SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 1–19

COMPUTING THE SVD OF A GENERAL MATRIX PRODUCT/QUOTIENT*

GENE GOLUB†, KNUT SØLNA‡, AND PAUL VAN DOOREN§

Abstract. In this paper we derive a new algorithm for constructing a unitary decomposition of a sequence of matrices in product or quotient form. The unitary decomposition requires only unitary left and right transformations on the individual matrices and amounts to computing the generalized singular value decomposition of the sequence. The proposed algorithm is related to the classical Golub–Kahan procedure for computing the singular value decomposition (SVD) of a single matrix in that it constructs a bidiagonal form of the sequence as an intermediate result. When applied to two matrices this new method is an alternative way of computing the quotient and product SVD and is more economical than current methods.

Key words. mumerical methods, generalized singular values, products of matrices, quotients of matrices

AMS subject classification. 65F15

PII. S0895479897325578

Introduction. The two basic unitary decompositions of a matrix A yielding some spectral information are the Schur form $A = UTU^*$—where U is unitary and T is upper triangular—and the singular value decomposition (SVD) $A = U\Sigma V^*$—where U and V are unitary and Σ is diagonal (for the latter A does not need to be square). It is interesting to note that both forms are usually computed by a QR-like iteration [7]. The SVD algorithm of Golub–Kahan [6] is indeed an implicit QR algorithm applied to the Hermitian matrix A^*A. When looking at unitary decompositions involving *two* matrices, say, A and B, a similar implicit algorithm was given in [10] and is known as the QZ algorithm. It computes $A = QT_aZ^*$ and $B = QT_bZ^*$, where Q and Z are unitary and T_a and T_b are upper triangular. This algorithm is in fact the QR algorithm again performed implicitly on the quotient $B^{-1}A$. The corresponding decomposition is therefore also known as the *generalized Schur form*.

When considering the generalized SVD of two matrices, appearing as a quotient $B^{-1}A$ or a product BA, the currently used algorithm is *not* of QR type but of a Jacobi type. The reason for this choice is that Jacobi methods easily extend to products and quotients. Unfortunately, the Jacobi algorithm typically has a (moderately) higher complexity than the QR algorithm. Yet, so far, nobody proposed an implicit QR-like method for the SVD of a product or quotient of two matrices.

*Received by the editors August 20, 1997; accepted for publication (in revised form) by L. Eldén November 15, 1999; published electronically May 31, 2000. This paper contains research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

http://www.siam.org/journals/simax/22-1/32557.html

†Computer Science Department, Stanford University, Stanford, CA 94305-9025 (golub@sccm.stanford.edu). This author was partially supported by the National Science Foundation under grants DMS-9105192 and DMS-9403899.

‡SC-CM, Stanford University, Stanford, CA 94305-9025 (solna@sccm.stanford.edu). This author was partially supported by The Research Council of Norway.

§Cesame, Université Catholique de Louvain, Louvain-la-Neuve B1348, Belgium (vdooren@anma.ucl.ac.be). This author was partially supported by the National Science Foundation under grant CCR-96-19596.

In this paper we show that, in fact, such an implicit algorithm is easy to derive and that it even extends straightforwardly to sequences of products/quotients of several matrices. Moreover, the complexity will be shown to be lower than for the corresponding Jacobi-like methods.

1. Implicit SVD. Consider the problem of computing the SVD of a matrix A that is an expression of the following type:

$$A = A_K^{s_K} \cdots A_2^{s_2} \cdot A_1^{s_1}, \tag{1}$$

where $s_i = \pm 1$, i.e., a sequence of products or quotients of matrices. For simplicity we assume that the A_i matrices are square $n \times n$ and invertible. It was pointed out in [3] that one can always perform a preliminary QR-like reduction that extracts from a sequence of matrices with compatible dimensions another sequence of square invertible matrices with the same generalized singular values as the original sequence. We refer to [3] for the details of this reduction and will treat here only the case of square invertible matrices. While it is clear that one has to perform left and right transformations on A to get $U^*AV = \Sigma$, these transformations will affect only A_K and A_1. Beyond this, one can insert an expression $Q_i^*Q_i = I_n$ between every pair $A_{i+1}^{s_{i+1}}A_i^{s_i}$ in (1). If we also define $Q_K \doteq U$ and $Q_0 \doteq V$, we arrive at the following expression:

$$U^*AV = (Q_K^* A_K^{s_K} Q_{K-1}) \cdots (Q_2^* A_2^{s_2} Q_1) \cdot (Q_1^* A_1^{s_1} Q_0). \tag{2}$$

With the degrees of freedom present in these $K+1$ unitary transformations Q_i at hand, one can now choose each expression $Q_i^* A_i^{s_i} Q_{i-1}$ to be upper triangular. Note that the expression $Q_i^* A_i^{s_i} Q_{i-1} = T_i^{s_i}$ with T_i upper triangular can be rewritten as

$$Q_i^* A_i Q_{i-1} = T_i \quad \text{for } s_i = 1\,, \qquad\qquad Q_{j-1}^* A_j Q_j = T_j \quad \text{for } s_j = -1. \tag{3}$$

From the construction of a normal QR decomposition, it is clear that while making the matrix A upper triangular, this "freezes" only one matrix Q_i per matrix A_i. The remaining unitary matrix leaves enough freedom to finally diagonalize the matrix A as well. Since (2) computes the singular values of (1), it is clear that such a result can be obtained only by an *iterative procedure*. On the other hand, one intermediate form that is used in the Golub–Kahan SVD algorithm [6] is the bidiagonalization of A and this can be obtained in a *finite recurrence*. We show in the next section that the matrices Q_i in (2) can be constructed in a finite number of steps in order to obtain a bidiagonal $Q_K^* A Q_0$ in (2). In carrying out this task one should try to do as much as possible implicitly. Moreover, one would like the total complexity of the algorithm to be comparable to, or less than, the cost of K singular value decompositions. This means that the complexity should be $\mathrm{O}(Kn^3)$ for the whole process.

2. Implicit bidiagonalization. We now derive such an implicit reduction to bidiagonal form. Below $\mathcal{H}(i,j)$ denotes the group of *Householder* transformations having (i,j) as the range of rows/columns they operate on. Similarly $\mathcal{G}(i,i+1)$ denotes the group of *Givens* transformations operating on rows/columns i and $i+1$. We first consider the case where all $s_i = 1$. We thus have only a product of matrices A_i and in order to illustrate the procedure we show its evolution operating on a product of three matrices only, i.e., $A_3A_2A_1$. Below is a sequence of displays of the matrix product that illustrates the evolution of the bidiagonal reduction. Each display indicates the pattern of zeros ("0") and nonzeros ("x") in the three matrices.

First perform a Householder transformation $Q_1^{(1)} \in \mathcal{H}(1,n)$ on the rows of A_1 and the columns of A_2. Choose $Q_1^{(1)}$ to annihilate all but one element in the first column of A_1:

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}.$$

Then perform a Householder transformation $Q_2^{(1)} \in \mathcal{H}(1,n)$ on the rows of A_2 and the columns of A_3. Choose $Q_2^{(1)}$ to annihilate all but one element in the first column of A_2:

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}.$$

Then perform a Householder transformation $Q_3^{(1)} \in \mathcal{H}(1,n)$ on the rows of A_3. Choose $Q_3^{(1)}$ to annihilate all but one element in the first column of A_3:

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}.$$

Note that this third transformation yields the same form also for the product of the three matrices:

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix} = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}.$$

At this stage we are interested in the *first row* of this product (indicated by boldface $\mathbf{x}$'s above). This row can be constructed as the product of the first row of A_3 with the matrices to the right of it, and this requires only $\mathrm{O}(Kn^2)$ flops. Once this row is constructed we can find a Householder transformation $Q_0^{(1)} \in \mathcal{H}(2,n)$ operating on the last $(n-1)$ elements which annihilates all but two elements (the colon, ":", is used as it is in MATLAB):

$$A_3(1,:)A_2A_1Q_0^{(1)} = \begin{bmatrix} x & x & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

This transformation is then applied to A_1 only and completes the first stage of the bidiagonalization since

$$Q_K^{(1)*} A Q_0^{(1)} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}.$$

The second stage of the bidiagonalization is analogous to the first; it differs only in that the transformations operate only on rows/columns 2 to n. The Householder transformations $Q_i^{(2)} \in \mathcal{H}(2,n)$ for $1 \le i \le 3$ are chosen to eliminate elements 3 to n in the second columns of A_i in the manner described above. The transformation $Q_0^{(2)} \in \mathcal{H}(3,n)$ operates on the last $(n-2)$ elements of the second row of the product and annihilates all but two elements:

$$A_3(2,:) A_2 A_1 Q_0^{(2)} = \begin{bmatrix} 0 & x & x & 0 & 0 \end{bmatrix}. \tag{5}$$

This transformation is applied to A_1 only, completing the second step of the bidiagonalization of A:

$$Q_K^{(2)*} Q_K^{(1)*} A Q_0^{(1)} Q_0^{(2)} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \end{bmatrix}.$$

It is now clear from the context how to proceed further with this algorithm to obtain after $n-1$ stages:

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix}.$$

Note that we never construct the whole product $A = A_3 A_2 A_1$, but rather compute one of its rows when needed for constructing the transformations $Q_0^{(i)}$. The only matrices that are kept in memory and updated are the A_i matrices and possibly Q_K and Q_0 if we require the singular vectors of A afterwards.

The complexity of this bidiagonalization step is easy to evaluate. Each matrix A_i gets pre- and postmultiplied with essentially n Householder transformations of decreasing range. For updating all A_i we therefore need $10Kn^3/3$ flops, and for updating Q_K and Q_0 we need $4n^3$ flops. For constructing the required row vectors of A we need $(K-1)n^3/3$ flops. Overall we thus need on the order of $11Kn^3/3$ flops for the construction of the triangular T_i and $4n^3$ for the outer transformations Q_K and Q_0. Essentially this is $11n^3/3$ flops per updated matrix.

If we now have some of the $s_i = -1$, we cannot use Householder transformations anymore on all matrices. Indeed, in order to construct the rows of A when needed, the matrices A_i for which $s_i = -1$ have to be triangularized first, say, with a QR factorization. The QR factorization is performed in an initial step and uses Householder transformations. From there on the same procedure as above is followed, but this implies using Givens rotations in certain steps of the bidiagonalization. For simplicity, we illustrate this on a matrix $A = A_3 A_2^{-1} A_1$. We first apply a left transformation

$Q_2^{(0)}$ (using a sequence of Householder transformations) that triangularizes A_2 from the left and also apply this to the rows of A_1. The resulting triple then has the form

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix}.$$

We then apply a unitary transformation $Q_1^{(1)}$ to the rows of A_1 to eliminate elements 2 to n in its first column using Givens transformations $G_1 \in \mathcal{G}(n-1, n)$ until $G_{n-1} \in \mathcal{G}(1, 2)$. Below we indicate in which order these zeros are created in the first column of A_1 by their index:

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0_4 & x & x & x & x \\ 0_3 & x & x & x & x \\ 0_2 & x & x & x & x \\ 0_1 & x & x & x & x \end{bmatrix}.$$

These transformations also have to be applied to the left of A_2, but the use of Givens rotations allows us to update the triangularized matrix A_2, while keeping it upper triangular: each time a Givens rotation applied to the left of A_2 destroys its triangular form, another Givens rotation is applied to the right of A_2 in order to restore its triangular form. (The same technique is used, for instance, in keeping the B matrix upper triangular in the QZ algorithm applied to $B^{-1}A$.) Let $Q_2^{(1)}$ be the product of the Givens rotations applied to the right of A_2, then $Q_2^{(1)}$ also has to be applied to the right of A_3. Finally, for the column transformation $Q_3^{(1)}$ of A_3 eliminating elements 2 to n of its first column, we can again use a Householder transformation $H_5 \in \mathcal{H}(1, n)$. After this fifth transformation, the resulting triple has the form

$$\begin{bmatrix} x & x & x & x & x \\ 0_5 & x & x & x & x \\ 0_5 & x & x & x & x \\ 0_5 & x & x & x & x \\ 0_5 & x & x & x & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix},$$

which clearly has a first column with only its leading element different from 0. Its first row can easily be constructed, and we then apply a Householder transformation $Q_0^{(1)} \in \mathcal{H}(2, n)$ annihilating all but two elements as indicated in (4). This completes the first stage of the bidiagonalization of $A = A_3 A_2^{-1} A_1$. Subsequent steps are similar but operate on matrices of decreasing dimensions.

Notice that all transformations Q_i and Q_{i-1} applied to a matrix A_i with index $s_i = -1$ have to be of Givens type, which require more flops than Householder transformations for the same number of annihilated elements. So the more negative exponents we have, the more expensive the overall algorithm becomes. Without loss of generality, we can assume that at most half of the indices are equal to -1, since otherwise we can compute the SVD of A^{-1} rather than that of A (all matrices A_i were assumed to be invertible). The situation with the highest computational complexity is thus when every other index s_i is negative, since then all transformations

but one have to be of Givens type. Let us analyze the case where K is even and $s_{2i} = -1,\ s_{2i-1} = 1,\ i = 1, \ldots, \frac{K}{2}$. The preliminary QR reduction of the matrices A_{2i} requires $\frac{4}{3}n^3\frac{K}{2}$ flops and $2n^3\frac{K}{2}$ flops for also updating the matrices A_{2i-1} with these transformations. From there on, each matrix undergoes $\frac{n(n-1)}{2}$ Givens rotations on the left and on the right. For the triangular matrices A_{2i} this requires a total of $3n^2\frac{K}{2}$ flops, whereas for the (originally dense) matrices A_{2i-1} this requires a total of $5n^3\frac{K}{2}$ flops. For constructing the required row vectors of A we need $(K-1)n^3/3$ flops as before. Finally, updating the matrix Q_0 via Givens transformations and Q_K via Householder transformations requires $2n^2$ and $3n^3$ flops, respectively. The worst-case complexity of the general case is thus $6n^3K$ flops for obtaining the triangular matrices T_i and $5n^3$ flops for the outer transformations Q_0 and Q_K. This is about 60% more than for the product case.

3. Error analysis. In the previous section we showed how to obtain an estimate of a bidiagonal decomposition of the matrix product/quotient. We now turn to the problem of obtaining accurate estimates of the singular values. This warrants a discussion of the errors committed in the bidiagonalization step.

The use of Householder and Givens transformations for all operations in the bidiagonalization step guarantees that the obtained matrices T_i in fact correspond to slightly perturbed data as follows:

$$T_i = Q_i^*(A_i + \delta A_i)Q_{i-1},\ s_i = 1, \qquad T_j = Q_{j-1}^*(A_j + \delta A_j)Q_j,\ s_j = -1, \tag{6}$$

where

$$\|\delta A_i\| \leq \epsilon c_n \|A_i\|\,, \qquad \|Q_i^* Q_i - I_n\| \leq \epsilon d_n, \tag{7}$$

with ϵ the machine precision and $c_n,\ d_n$ moderate constants depending on the problem size n. This is obvious since each element transformed to zero can indeed be put equal to zero without affecting the ϵ bound (see [11], [7]).

The situation is different for the elements of A since they are not stored explicitly in the computer. How does one proceed further to compute the generalized singular values of A? Once the triangular matrices T_i are obtained, it is easy and cheap to *reconstruct* the bidiagonal:

$$T_K^{s_k} \cdots T_2^{s_2} \cdot T_1^{s_1} = B = \begin{bmatrix} q_1 & e_2 & o_{1,3} & \cdots & o_{1,n} \\ & q_2 & e_3 & \ddots & \vdots \\ & & \ddots & \ddots & o_{n-2,n} \\ & & & \ddots & e_n \\ & & & & q_n \end{bmatrix}, \tag{8}$$

and then compute the singular values of the bidiagonal in a *standard* way. The diagonal elements q_i are indeed just a product of the corresponding diagonal elements of the T_j matrices, possibly inverted:

$$q_i = t_{K_{i,i}}^{s_K} \cdots t_{2_{i,i}}^{s_2} \cdot t_{1_{i,i}}^{s_1},$$

and the off-diagonal elements e_i can be computed from the corresponding 2×2 diagonal blocks (with index $i-1$ and i) of the T_j matrices. It is clear that the q_i can be computed in a backward stable way since all errors can be superimposed on the

diagonal elements $t_{j_{i,i}}$ of the matrices T_j. For the errors incurred when computing the e_i one needs a more detailed analysis. We show below that the backward errors can be superimposed on the off-diagonal elements $t_{j_{i-1,i}}$ of T_j without creating any conflicts with previously constructed backward errors, and we derive bounds for these backward errors. From the vector recurrence

$$\begin{bmatrix} e \\ q \end{bmatrix} := \begin{bmatrix} t_{j_{i-1,i-1}} & t_{j_{i-1,i}} \\ 0 & t_{j_{i,i}} \end{bmatrix}^{s_j} \cdot \begin{bmatrix} e \\ q \end{bmatrix} \tag{9}$$

we easily derive the following algorithm used for computing q_i and e_i for $i = 1, \ldots, n$.

$q := 1;\ e := 0;$
  for $j = 1 : K$
    if $s_j = 1$, **then** $e := e * t_{j_{i-1,i-1}} + q * t_{j_{i-1,i}};\ q := q * t_{j_{i,i}};$
      else $q := q/t_{j_{i,i}};\ e := (e - q * t_{j_{i-1,i}})/t_{j_{i-1,i-1}};$
  end
  $q_i := q;\ e_i := e;$

Note that for $i = 1$ the same recurrence holds without the expressions involving e. From these recurrences it is clear that the calculation of q_i involves one flop per step j and hence a total of K rounding errors which can be superimposed on the diagonal elements $t_{j_{i,i}}$:

$$\begin{aligned} q_i &= comp(t_{K_{i,i}}^{s_K} \cdots t_{1_{i,i}}^{s_1}) \\ &= \bar{t}_{K_{i,i}}^{s_K} \cdots \bar{t}_{1_{i,i}}^{s_1} \qquad \text{with} \quad \bar{t}_{j_{i,i}} = t_{j_{i,i}}(1 + \epsilon_{i,j}),\ \ |\epsilon_{i,j}| < \epsilon \end{aligned} \tag{10}$$

with *comp* denoting a floating point operator. For the calculation of e_i there are 3 flops per step j and hence a total of $3K$ roundings which have to be superimposed on the $t_{j_{i-1,i}}$ elements. Fortunately, e_j is a sum of K terms which contain each a *different* element $t_{j_{i-1,i}}$ as a factor. We illustrate this for $K = 4$ and $s_j = 1$, highlighting the relevant elements:

$$\begin{aligned} e_i = comp(&\mathbf{t}_{4_{i-1,i}} \cdot t_{3_{i,i}} \cdot t_{2_{i,i}} \cdot t_{1_{i,i}} \\ &+ t_{4_{i-1,i-1}} \cdot \mathbf{t}_{3_{i-1,i}} \cdot t_{2_{i,i}} \cdot t_{1_{i,i}} \\ &+ t_{4_{i-1,i-1}} \cdot t_{3_{i-1,i-1}} \cdot \mathbf{t}_{2_{i-1,i}} \cdot t_{1_{i,i}} \\ &+ t_{4_{i-1,i-1}} \cdot t_{3_{i-1,i-1}} \cdot t_{2_{i-1,i-1}} \cdot \mathbf{t}_{1_{i-1,i}}). \end{aligned} \tag{11}$$

The $3K$ rounding errors can thus easily be superimposed on these different elements $t_{j_{i-1,i}},\ j = 1, \ldots, K$. But since we have already superimposed errors on the all-diagonal elements $t_{j_{i,i}}$ we have to add these perturbations here as well. For $s_j = 1$ we thus have

$$\begin{aligned} e_i = &\bar{t}_{4_{i-1,i}} \cdot \bar{t}_{3_{i,i}} \cdot \bar{t}_{2_{i,i}} \cdot \bar{t}_{1_{i,i}} \\ &+ \bar{t}_{4_{i-1,i-1}} \cdot \bar{t}_{3_{i-1,i}} \cdot \bar{t}_{2_{i,i}} \cdot \bar{t}_{1_{i,i}} \\ &+ \bar{t}_{4_{i-1,i-1}} \cdot \bar{t}_{3_{i-1,i-1}} \cdot \bar{t}_{2_{i-1,i}} \cdot \bar{t}_{1_{i,i}} \\ &+ \bar{t}_{4_{i-1,i-1}} \cdot \bar{t}_{3_{i-1,i-1}} \cdot \bar{t}_{2_{i-1,i-1}} \cdot \bar{t}_{1_{i-1,i}}, \end{aligned} \tag{12}$$

where $(K-1)$ additional roundings are induced for each factor. Therefore, we have $\bar{t}_{j_{i-1,i}} = t_{j_{i-1,i}}(1 + \eta_{i,j}),\ \ |\eta_{i,j}| < (4K-1)\epsilon/(1-(4K-1)\epsilon)$. When some of the $s_j = -1$ the above expression is similar: the $t_{j_{i,i}}$ then appear as inverses, some $+$ signs change to $-$ signs, and an additional factor $1/(t_{j_{i-1,i-1}} t_{j_{i,i}})$ appears in the jth term if $s_j = -1$. So in the worst case $(K+1)$ additional roundings are introduced for each factor and the obtained bound is then $|\eta_{i,j}| < (4K+1)\epsilon/(1-(4K+1)\epsilon)$. In the worst case the errors yield a backward perturbation $\|\delta T_j\|$ which is thus bounded

by $5K\epsilon\|T_j\|$ and hence much smaller than the errors δA_j incurred in the triangularization process. The perturbation effect of computing the elements q_i and e_i is thus negligible compared to that of the triangularization. We thus showed that the computed bidiagonal corresponds *exactly* to the bidiagonal of the product of slightly perturbed triangular matrices T_j, that in turn satisfy the bounds (6)–(7). Unfortunately, nothing of the kind can be guaranteed for the elements $o_{i,j}$ in (8), which are supposed to be zero in exact arithmetic. Notice that the element e_{i+1} is obtained as the norm of the vector on which a Householder transformation is applied:

$$|e_{i+1}| = \|T_K^{s_K}(i, i:n)T_{K-1}^{s_{K-1}}(i:n, i:n)\cdots T_1^{s_1}(i:n, i+1:n)\|,$$

where we used the MATLAB notation for subarrays: $T_1^{s_1}(i:n, i+1:n)$ is thus the submatrix of $T_1^{s_1}$ with row indices i to n and column indices $i+1$ to n. If all $s_i = 1$ we can obtain by straightforward perturbation results of matrix vector products, a bound of the type

$$|o_{i,j}| \leq \epsilon c_n \|T_K(i, i:n)\| \cdot \|T_{K-1}(i:n, i:n)\| \cdots \|T_1(i:n, i:n)\|.$$

If not all $s_i = 1$ we need to also use perturbation results of solutions of systems of equations, since we need to evaluate the last $n-i$ components of the vector $e_i^* T_K^{s_K}(i : n, i:n)T_{K-1}^{s_{K-1}}(i:n, i:n)\cdots T_1^{s_1}(i:n, i:n)$ and this requires a solution of a triangular system of equations each time a power $s_j = -1$ is encountered. In this case the bound would become

$$|o_{i,j}| \leq \epsilon c_n \|T_K^{s_K}(i, i:n)\| \cdot \|T_{K-1}^{s_{K-1}}(i:n, i:n)\| \cdots \|T_1^{s_1}(i:n, i:n)\|\kappa,$$

where κ is the product of all condition numbers of the inverted triangular systems (and hence much larger than 1). These are much weaker bounds than asking the off-diagonal elements of A to be ϵ smaller than the ones on the bidiagonal. This would be the case, e.g., if instead we had

$$|o_{i,j}| \leq \epsilon c_n \|T_K^{s_K}(i, i:n)T_{K-1}^{s_{K-1}}(i:n, i:n)\cdots T_1^{s_1}(i:n, i+1:n)\| = \epsilon c_n |e_{i+1}|.$$

Such a bound would guarantee high relative accuracy in the singular values computed from the bidiagonal only [4]. Hence, this is the kind of result one would hope for. These two bounds can in fact be very different when significant cancellations occur between the individual matrices, e.g., if

$$\|A\| << \|A_K^{s_K}\| \cdots \|A_2^{s_2}\| \cdot \|A_1^{s_1}\|.$$

One could observe that the bidiagonalization procedure is in fact a Lanczos procedure [6]. Therefore, there is a tendency to find first the *dominant* directions of the expression $A_K^{s_K} \cdots A_1^{s_1}$ and hence also those directions where there is less cancellation between the different factors. We will see in the examples below that such a phenomenon indeed occurs which is a plausible explanation for the good accuracy obtained. One way to test the performance of the algorithm in cases with very small singular values is to generate powers of a symmetric matrix $A = S^K$. The singular values will be the powers of the absolute values of the eigenvalues of S:

$$\sigma_i(A) = |\lambda_i(S)|^K,$$

and hence will have a large dynamic range. The same should be true for the bidiagonal of A and the size of the $o_{i,j}$ will then become critical for the accuracy of the singular

values when computed from the bidiagonal elements $q_i,\ e_i$. This and several other examples are discussed in the next section. There we observe a very high relative accuracy even for the smallest singular values. The only explanation we can give for this is that as the bidiagonalization proceeds, it progressively finds the largest singular values first and creates submatrices that are of smaller norm. These then do not really have cancellation between them, but instead the decreasing size of the bidiagonal elements is the result of decreasing elements in each transformed matrix A_i. In other words, a grading is created in each of the transformed matrices. We believe this could be explained by the fact that the bidiagonalization is a Lanczos procedure and that such grading is often observed there when the matrix has a large dynamic range of eigenvalues. In practice it is of course always possible to evaluate bounds for the elements $|o_{i,j}|$ and thereby obtain estimates of the accuracy of the computed singular values.

The consequence of the above is that the singular values of such sequences can be computed (or better, "estimated") at high relative accuracy from the bidiagonal only! Notice that the bidiagonalization requires 4 to $6Kn^3$ flops but that the subsequent SVD of the bidiagonal is essentially free since it is $\mathrm{O}(n^2)$.

4. Singular vectors and iterative refinement. If one wants the singular vectors as well as the singular values at a guaranteed accuracy, one can start from the bidiagonal B as follows. First compute the bidiagonal,

$$B = Q_K^* A Q_0 = T_K^{s_K} \cdots T_2^{s_2} \cdot T_1^{s_1},$$

and then the SVD of B,

$$B = U \Sigma V^*,$$

where we choose the diagonal elements of Σ to be ordered in decreasing order. We then proceed by propagating the transformation U (or V) and updating each T_i so that they remain upper triangular. Since the neglected elements $o_{i,j}$ were small, the new form

$$\hat{Q}_K^* A \hat{Q}_0 = \hat{T}_K^{s_K} \cdots \hat{T}_2^{s_2} \cdot \hat{T}_1^{s_1}$$

will be upper triangular, and nearly diagonal. This is the ideal situation to apply one sweep of Kogbetliantz's algorithm. Since this algorithm is quadratically convergent when the diagonal is ordered [2], one sweep should be enough to obtain ϵ-small off-diagonal elements.

The complexity of this procedure is as follows. If we use only Givens transformations, we can keep all matrices upper triangular by a subsequent Givens correction. Such a pair takes $6n$ flops per matrix and we need to propagate $n^2/2$ of those. That means $3n^3$ per matrix. The cost of one Kogbetliantz sweep is exactly the same since we propagate the same amount of Givens rotations. We therefore arrive at the following total count for our algorithm:

4 to $6Kn^3$ for triangularizing $A_i \to T_i$,
4 to $5n^3$ for constructing Q_K and Q_0,
$8n^3$ for computing U and V,
$3Kn^3$ for updating $T_i \to \hat{T}_i$,
$3Kn^3$ for one last Kogbetliantz sweep.

The total amount of flops after the bidiagonalization is thus comparable to applying 2 Kogbetliantz sweeps, whereas the Jacobi-like methods typically require 5 to 10 sweeps!

Moreover, our method allows us to select a few singular values and only compute the corresponding singular vectors. The matrices Q_K and Q_0 can be stored in factored form and inverse iteration performed on B to find its selected singular vector pairs and then transformed back to pairs of A using Q_K and Q_0.

5. Numerical examples. Computing the SVD of a general product/quotient of matrices is, as suggested above, a delicate numerical problem. In this section we analyze the accuracy of the QR-like algorithm described in this paper using several examples of varying degree of difficulty. The examples are chosen in order to illustrate the following points already discussed in the paper.

(a) Implicit methods are more reliable than explicit methods. This is of course well known, but we illustrate it with some striking examples.
(b) The bidiagonal computed by the QR-like method yields singular values computed to high relative accuracy even when their dynamical range is very large.
(c) The bidiagonal has a typical "graded" structure when the singular values have a wide dynamical range and its "off-bidiagonal" elements are negligible with respect to the bidiagonal elements in the same row. This is due to its connection to the Lanczos procedure as discussed earlier.
(d) The connection with the Lanczos procedure also allows us to terminate the bidiagonalization early and yet has a good estimate of the dominant singular values.

Points (a)–(d) illustrate the good (relative) accuracy that can be obtained from this procedure even without using iterative refinement based on Kogbetliantz's algorithm. The following points now compare the QR-like and Kogbetliantz approaches.

(e) The bidiagonalization and Kogbetliantz methods have comparable accuracy in "difficult" examples with strong cancellation in the product.
(f) The typical number of Kogbetliantz steps (6 to 10) needed for convergence yields a much slower method than mere bidiagonalization. Moreover, the results are comparable, even when the Kogbetliantz iteration is continued further.
(g) The accuracy obtained from the bidiagonal only is already better on average than that of Kogbetliantz.

Finally, we illustrate that good (relative) accuracy is obtained also for a matrix *quotient*.

(h) The accuracy obtained by bidiagonalization of a matrix quotient is high, even when compared to the accuracy obtained if the inverted factors are explicitly known.

These points illustrate the power of this QR-like method. Note that in all examples we use only the basic part of the algorithm without the iterative refinement step. All calculations were carried out in MATLAB on a Silicon Graphics Indigo workstation with IEEE floating point standard. For computing the singular values of the computed bidiagonal we use the method due to Fernando and Parlett [5].

(a) *Implicit versus explicit.* Let us consider the following products:

$$A_1[n, m] = T_n^m,$$

where T_n is a $n \times n$ symmetric Toeplitz matrix whose first column is $[2, -1, 0, 0, \ldots, 0]$ (singular values and singular vectors of such matrices are known [8]). Since it contains only integers we can form these powers of T_n without any rounding errors, which is important for our comparison. The accuracy obtained by computing the SVD of this explicitly formed product is displayed in Figure 1. The interpretation of the figure is

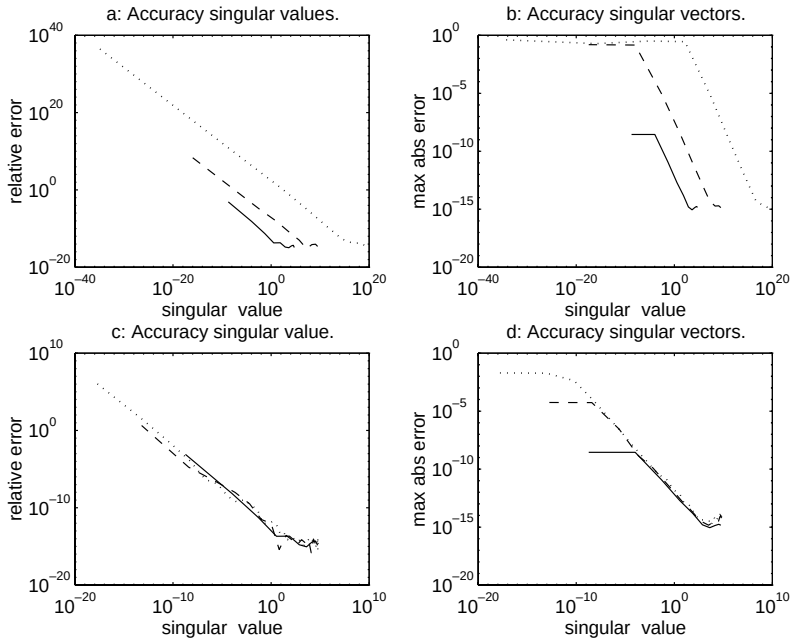


FIG. 1. *The relative accuracy obtained by computing the SVD for the explicitly formed product of Toeplitz matrices. In* (a) *and* (b) *solid, dashed, and dotted lines correspond to* $A_1[n,m]$ *for* $n = 10$ *and* $m \in \{8, 16, 32\}$*; in* (c) *and* (d), $n \in \{10, 20, 40\}$ *and* $m = 8$. *The lines interpolate the relative accuracies for the different singular values.*

as follows. Figures 1(a) and 1(b) correspond to $n = 10$ and $m \in \{8, 16, 32\}$, whereas Figures 1(c) and 1(d) correspond to $n \in \{10, 20, 40\}$ and $m = 8$. The associated results are indicated by the solid, dashed, and dotted lines, respectively. Notice that each line corresponds to a different range of singular values as expected for matrices T_n^m with different values of m and n. In Figures 1(a) and 1(c) we plot the relative accuracy of the singular values as a function of the actual magnitude of the singular value. The lines interpolate the observed relative accuracies, that is, the values $|\sigma_i - \hat{\sigma}_i|/\sigma_i$. In Figures 1(b) and 1(d) we plot the maximum absolute error in the left singular vector elements, that is, $\max_j[|u_{ji} - \hat{u}_{ji}|]$, with u_{ji} being the elements of the ith singular vector, also as a function of the magnitude of the corresponding singular value. From the figure it is clear that the relative accuracy of the computed decomposition is quickly lost as we form powers of the matrix. Moreover, the situation is aggravated as we increase the dimension, and hence the condition number, of the matrix. The explanation lies of course in the fact that roundoff errors in a matrix A are typically proportional to $\epsilon||A||$. For the product $A_1[n,m]$ this tends to have a catastrophic effect on the accuracy of the smallest singular values since they are smaller than $\epsilon||A_1||$.

Let us now use the QR-like SVD algorithm. The result is shown in Figure 2 and illustrates that we have obtained a relative accuracy which is essentially independent of the magnitude of the associated singular value. This shows the need for implicit methods.

(b) *Relative accuracy of implicit methods.* A nonsymmetric example along the same vein is given in Figure 3. The interpretation of the figure is as for Figure 2, but now we consider the product

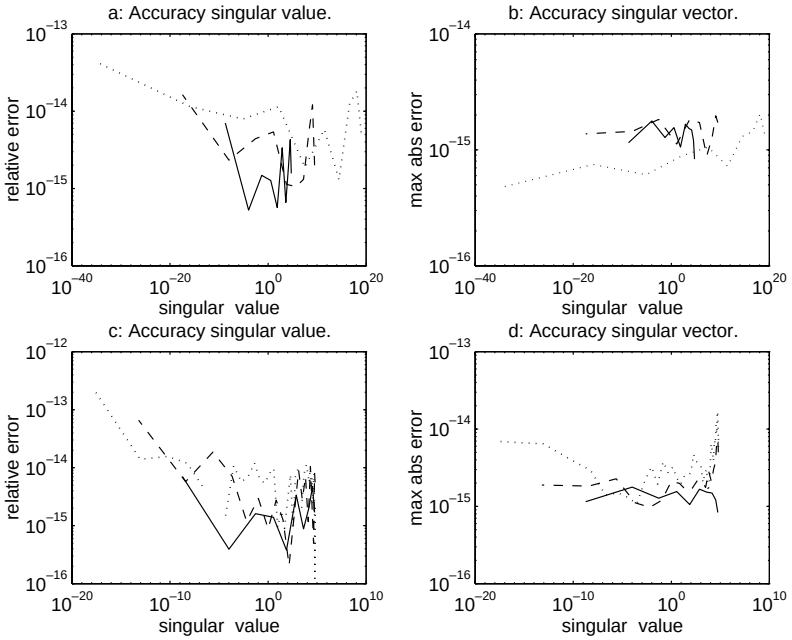


FIG. 2. *Relative accuracy of the SVD estimate obtained by the QR-like algorithm for the product of Toeplitz matrices. In* (a) *and* (b) *solid, dashed, and dotted lines correspond to* $A_1[n,m]$ *for* $n = 10$ *and* $m \in \{8, 16, 32\}$*; in* (c) *and* (d)*,* $n \in \{10, 20, 40\}$ *and* $m = 8$*. Note that also the smallest singular values are computed with high relative accuracy.*

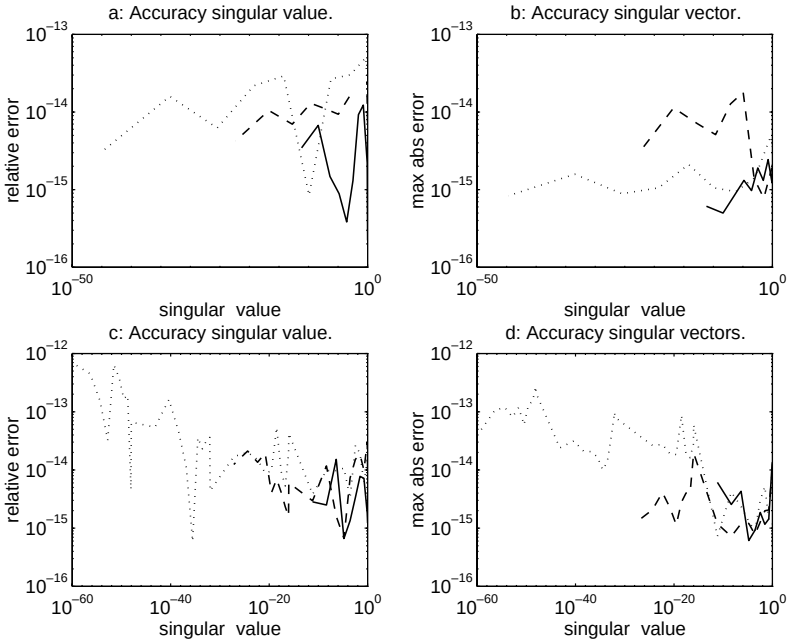


FIG. 3. *Relative accuracy of the SVD estimate obtained by the QR-like algorithm for the product of nonsymmetric matrices. In* (a) *and* (b) *solid, dashed, and dotted lines correspond to* $A_2[n,m]$ *for* $n = 10$ *and* $m \in \{8, 16, 32\}$*; in* (c) *and* (d)*,* $n \in \{10, 20, 40\}$ *and* $m = 8$*.*

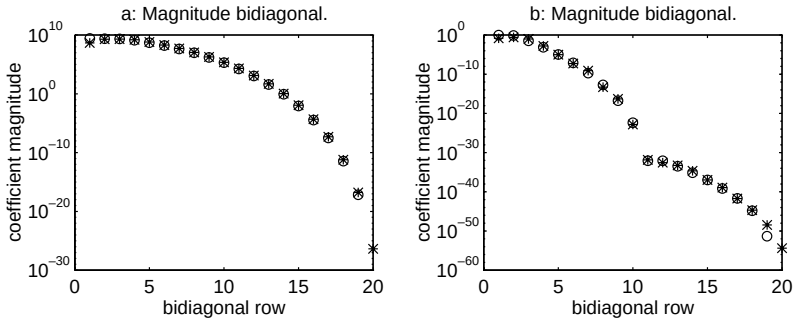


FIG. 4. *Grading in the bidiagonal decomposition computed by the QR-like algorithm. Figure* (a) *corresponds to* $A_1[20,16]$ *and* (b) *to* $A_2[20,16]$. *The* $*$*'s show the magnitude of the diagonal coefficients, the o's the magnitude of the upper bidiagonal coefficients.*

$$A_2[n,m] = (D_n\ D_n^*)^m.$$

Thus, there are $2m$ matrices in the matrix product. The matrix D_n is obtained by explicitly forming

$$D_n \equiv U_n\ \Sigma_n\ V_n^*, \tag{13}$$

where the matrices U_n and V_n are randomly chosen orthogonal matrices. They are defined by the singular vectors of a matrix with independent mean zero unit variance Gaussian entries. Furthermore, Σ_n is the leading part of a $10k \times 10k$ diagonal matrix with diagonal equal to the Kronecker product:

$$[10, 9.9, 9, 8, 7, 6, 5, 4, 3, 2] \otimes [10^{-1}, 10^{-2}, \ldots, 10^{-k}].$$

The motivation for choosing the product in this way is that we obtain an example in which the matrices involved are nonsymmetric and for which we "know" the actual singular values and can examine the obtained relative accuracy. The result is much as above. Using the implicit procedure for computing the singular values returns singular values whose relative accuracies are rather insensitive to the actual magnitude of the corresponding singular value.

(c) *Graded bidiagonal.* That the merits of the algorithm can be understood in terms of the Lanczos connection is confirmed by the next example explained in Figure 4. Here we have plotted the magnitude of the coefficients of the computed bidiagonal for the products $A_1[20,16]$ and $A_2[20,16]$, respectively, in Figures 4(a) and 4(b). The $*$'s show the absolute values of the diagonal coefficients and the o's the absolute values of the upper bidiagonal coefficients in the computed bidiagonal. We see that in both cases a grading has indeed been obtained. The algorithm picks out the dominant directions first, leading to a grading in the computed decomposition. The "effective condition number" of remaining subproblems are therefore successively reduced, and the associated singular values can apparently be obtained with high relative accuracy.

The high accuracy obtained above suggests that the "off-bidiagonal" elements in the transformed product are indeed small relative to the bidiagonal. This is confirmed by the next figure. In Figure 5 we plot, indicated by $*$, the norm of the off-bidiagonal elements normalized by the norm of the bidiagonal elements. That is, after the transformation to upper triangular form we explicitly form the product of the matrices in the product and compute for each row j, $||o_{j,(j+2):n}||/||o_{j,j:(j+1)}||$, with $o_{i,j}$ being the

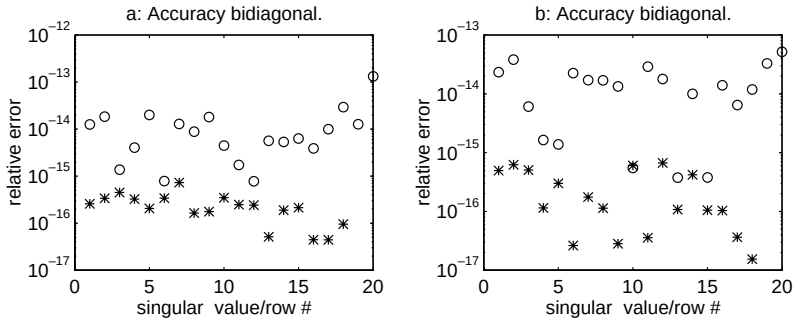


FIG. 5. *Relative accuracies of the bidiagonal elements* $(*)$ *and of the computed singular values* (o). *Figure* (a) *corresponds to* $A_1[20,16]$ *and* (b) *to* $A_2[20,16]$.

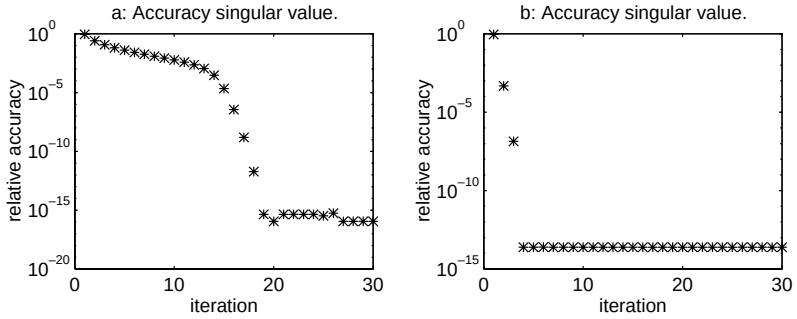


FIG. 6. *Accuracy of estimate of dominant singular value obtained from leading part of computed bidiagonal. The accuracy is plotted as a function of the dimension of the leading submatrix. Figure* (a) *corresponds to* $A_1[80,16]$ *and* (b) *to* $A_2[80,16]$.

elements in the computed product. In exact arithmetic this quantity should be zero. The o's in the figure are the relative accuracies in the computed singular values. Note that the grading and relative smallness of the off-bidiagonal elements make it possible to compute even the smallest singular values with high relative accuracy.

(d) *Dominant singular value.* A consequence of the Lanczos connection is furthermore that we can obtain good estimates for the dominant singular values of the product without computing the full bidiagonal. This is illustrated in Figure 6. Here we plot the estimate of the dominant singular value we obtain by computing the corresponding singular value for the leading parts of the computed bidiagonal, $\hat{B}(1:i,1:i)$. We plot the relative accuracy of this estimate as a function of i in Figures 6(a) and 6(b), corresponding to the products $A_1[80,16]$ and $A_2[80,16]$, respectively. The plots show that we need compute only a part of the bidiagonal in order to obtain a good estimate of the dominant singular value.

(e) *Examples with strong cancellation.* Here we consider examples with a significant cancellation in the product. That is, a subsequence of the matrices in the product is associated with a large dynamic range relative to that of the product whose associated singular values might be only mildly, or not at all, graded. The following example illustrates this:

$$A_3[10,m] = D_{10}^{m}\ D_{10}^{-m},$$
$$A_4[10,m] = (D_{10}\ D_{10}^{-1})^{m},$$

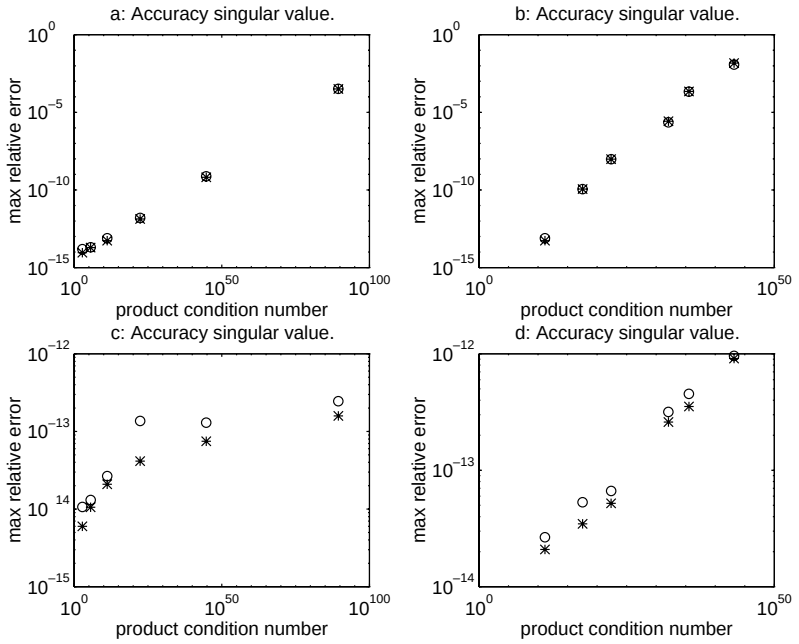


FIG. 7. *The figure compares the accuracy of the computed SVD using, respectively, the QR-like* $(*)$ *and the Kogbetliantz* (o) *algorithms. The considered matrix products exhibit strong cancellation. Figures* (a) *and* (b) *correspond to* $A_3[n,m]$ *and* (c) *and* (d) *to* $A_4[n,m]$*. In* (a) *and* (c)*,* $n = 10$ *and* $m \in \{2,4,8,16,32,64\}$*; in* (b) *and* (d)*,* $n \in \{10,14,18,22,26,30\}$ *and* $m = 8$*. The QR-like algorithm provides accurate singular value estimates at a lower computational cost than the Kogbetliantz algorithm.*

with D_{10} defined as in (13). Note that in this example we compute D_{10}^{-1} explicitly and use the product form of the algorithm. In Figure 7 subplots (a) and (b) correspond to A_3 and (c) and (d) to A_4. For the various product sizes we plot the maximum relative error over the computed singular values. We do so as a function of the "product condition number," defined as the product of the condition numbers of the matrices involved, in this case κ_D^{2m}. In Figures 7(a) and 7(c) we let $n = 10$ and $m \in \{2,4,8,16,32,64\}$, whereas in Figures 7(b) and 7(d) we let $n \in \{10,14,18,22,26,30\}$ and $m = 8$. Note that for both of the above matrix products the product condition number is much larger than its actual condition number. Figures 7(a) and 7(b) show that for the matrix products which are associated with a significant cancellation there is a loss in relative accuracy. The o's in the plot correspond to computing the decomposition by the Kogbetliantz algorithm, fixing the number of sweeps to 12 to avoid issues of convergence tests. Note that the accuracy obtained thereby is not much better than that obtained by the bidiagonalization part of the QR-like algorithm, that is, without iterative refinement.

(f) *Convergence and complexity.* We next turn to the special but important case when $m = 2$ and compare the performance of the algorithm with that of the Kogbetliantz algorithm. In Figures 8(a) and 8(b) we consider the product $A_2[40,1]$. The dashed lines correspond to the relative accuracy obtained by $2, 4, 6$, and 10 sweeps of the Kogbetliantz algorithm. The relatively slow convergence of some singular values corresponds to those being closely spaced. Note that even 10 sweeps of Kogbetliantz's algorithm do not return an approximation with accuracy beyond that obtained by the

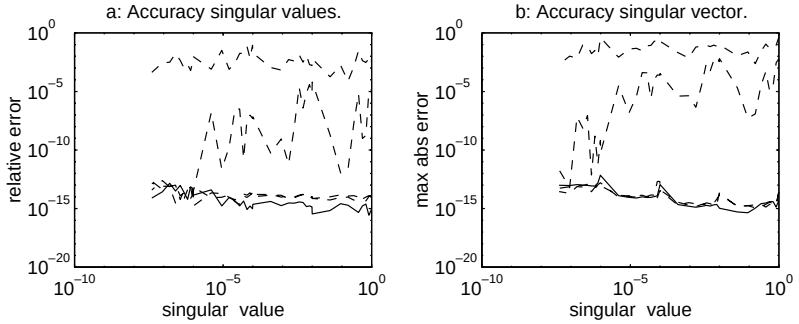


FIG. 8. *Convergence of Kogbetliantz algorithm when computing the SVD of the pair of matrices defined by* $A_2[40, 1]$. *The accuracies after* $2, 4, 6$, *and* 10 *sweeps are shown. The bottom solid line shows the accuracy obtained with the QR-like algorithm without iterative refinement.*

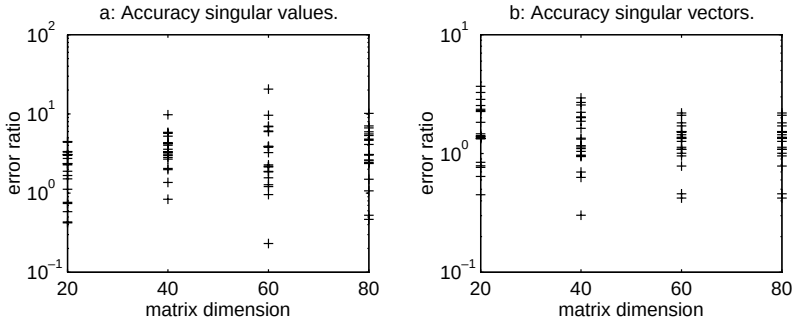


FIG. 9. *Comparison of accuracy of the computed SVD obtained, respectively, by the QR-like and the Kogbetliantz algorithms for the square of a collection of random matrices. For each matrix realization the cross is the maximum relative error with the Kogbetliantz algorithm over the maximum relative error for the QR-like algorithm.*

QR-like SVD algorithm without iterative refinement as shown by the solid line.

(g) *Comparison of accuracy.* The two plots in Figure 9 are obtained as follows. We consider the products defined by

$$A_5[n, 2] = N_n \; N_n^*$$

with N_n being an $n \times n$ random matrix whose coefficients are normally distributed and with $n \in \{20, 40, 60, 80\}$. The SVD was first computed (via MATLAB) and we then reconstructed N_n from this SVD. Hence it is reasonable to assume that the singular values of N_n and $A_5[n, 2]$ are known "exactly." Let $\hat{\sigma}_i$ and $\tilde{\sigma}_i$ represent the estimates of the singular values associated with, respectively, the QR-like and the Kogbetliantz algorithms. In the latter case we used 10 sweeps, whereas in the former we did not include iterative refinement. Similarly let $\hat{u}_{ij}$ and $\tilde{u}_{ij}$ represent the coefficients in the left singular vectors. We then plot the ratio $\max_i[|\sigma_i - \tilde{\sigma}_i|/\sigma_i]/\max_i[|\sigma_i - \hat{\sigma}_i|/\sigma_i]$ in Figure 9(a) and the ratio $\max_{ij}[|u_{ij} - \tilde{u}_{ij}|]/\max_{ij}[|u_{ij} - \hat{u}_{ij}|]$ in Figure 9(b). The +'s correspond to different realizations of the matrix N_n. We see that the QR-like algorithm typically yields a more accurate approximation despite its lower computational cost.

(h) *Example with matrix quotients.* In this last example we compute the SVD of matrix quotients involving inverted matrices. As described above, we then have

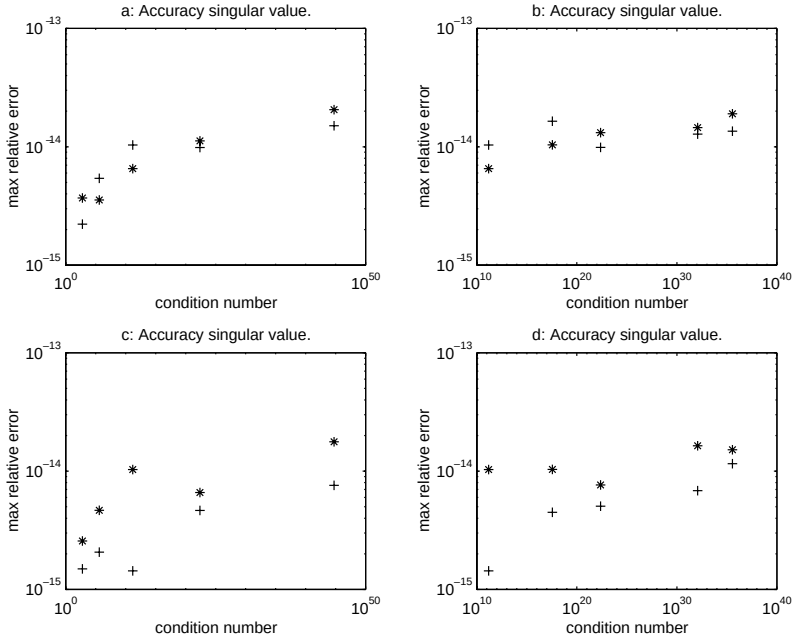


FIG. 10. *The figure compares the accuracy of the computed SVD when the QR-like algorithm based on, respectively, the quotient representation* (+) *and the product representation* (∗) *of the matrix is being used. Figures* (a) *and* (b) *correspond to* $A_6[n,m]$ *and* (c) *and* (d) *to* $A_7[n,m]$. *In* (a) *and* (c), $n = 10$ *and* $m \in \{2,4,8,16,32\}$ *and in* (b) *and* (d), $n \in \{10,14,18,22,26\}$ *and* $m = 8$. *Note that the accuracy obtained from the quotient representation is similar to the accuracy obtained from the product representation.*

to carry out an initial step in the bidiagonalization where the QR factorizations of the inverted matrices are computed. We construct the matrix quotients such that we explicitly can compute the inverses and compare the accuracy of the product version of the algorithm, based on these explicitly computed inverses, with the quotient version.

First, define the matrix quotient A_6:

$$A_6[n,m] = A_1^{-1}A_2^{-1}\cdots A_m^{-1}A_{m+1}A_{m+2}\cdots A_{2m},$$

where

$$\begin{aligned} A_i &= Q_n^{(i)}\Sigma_n^{-1}Q_n^{(i-1)*}, \\ A_{m+i} &= Q_n^{(m+i-1)}\Sigma_n Q_n^{(m+i)*} \end{aligned}$$

for $1 \le i \le m$ and with Σ_n defined as in (13). Moreover, the $Q_n^{(i)}$ are independent random orthogonal $n \times n$ matrices. As above these are defined by the singular vectors of a matrix with independent mean zero unit variance Gaussian entries.

Second, define the matrix quotient A_7:

$$A_7[n,m] = A_1^{-1}A_2A_3^{-1}\cdots A_{2m-2}A_{2m-1}^{-1}A_{2m}$$

with

$$\begin{aligned} A_{2i-1} &= Q_n^{(2i-1)}\Sigma_n^{-1}Q_n^{(2i-2)*}, \\ A_{2i} &= Q_n^{(2i-1)}\Sigma_n Q_n^{(2i)*} \end{aligned}$$

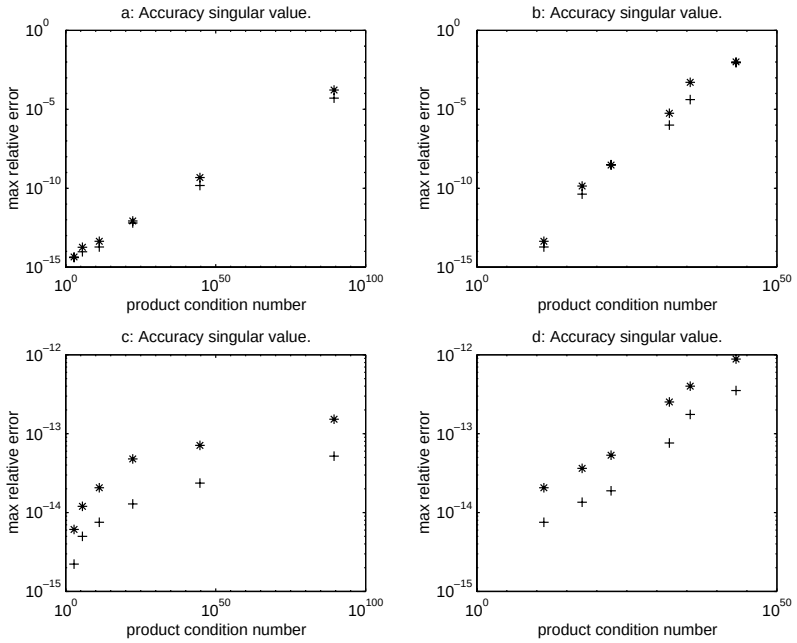


FIG. 11. *The figure compares the accuracy of the computed SVD when the QR-like algorithm based on, respectively, the quotient representation* (+) *and the product representation* (∗) *of the matrix is being used. The considered matrix quotients exhibit strong cancellation. Figures* (a) *and* (b) *correspond to* $A_3[n,m]$ *and* (c) *and* (d) *to* $A_4[n,m]$. *In* (a) *and* (c), $n = 10$ *and* $m \in \{2,4,8,16,32,64\}$ *and in* (b) *and* (d), $n \in \{10,14,18,22,26,30\}$ *and* $m = 8$.

for $1 \leq i \leq m$ and with Σ_n and $Q_n^{(i)}$ defined as above. Note that these quotients are associated with a large dynamic range.

The resulting relative accuracy obtained when varying the matrix dimension and the number of matrices in the quotient is shown in Figure 10. Figures 10(a) and 10(b) correspond to the quotient A_6 and Figures 10(c) and 10(d) correspond to the quotient A_7. In Figures 10(a) and 10(c), $n = 10$ and $m \in \{2,4,8,16,32\}$, whereas in Figures 10(b) and 10(d), $n \in \{10,14,18,22,26\}$ and $m = 8$. The +'s show the accuracy obtained with the QR-like algorithm based on the *quotient* and *without* iterative refinement. The $*$'s show the accuracy obtained with the product form of the QR-like algorithm *without* iterative refinement; note that in this case the inverses are explicitly computed. We see that the relative accuracy obtained when we do not assume knowledge of the inverses is comparable to, or even somewhat better than, the accuracy obtained if these are known!

Finally, reconsider the matrices of example (e) that exhibit strong cancellation. We compute as above the decomposition based on both the product form and the quotient form. The result is shown in Figure 11. Figures 11(a) and 11(b) correspond to the quotient A_3 and Figures 11(c) and 11(d) correspond to the quotient A_4. In Figures 11(a) and 11(c), $n = 10$ and $m \in \{2,4,8,16,32,64\}$, whereas in Figures 11(b) and 11(d), $n \in \{10,14,18,22,26,30\}$ and $m = 8$. The figure shows that computation based on the quotient form gives a relative accuracy that is in general somewhat better than the accuracy based on the product form, at the cost of a slightly higher flop count.

6. Concluding remarks. The algorithm presented in this paper nicely complements the unitary decompositions for sequences of matrices defined for the generalized QR [3] and Schur decompositions [1]. These decompositions find applications in sequences of matrices defined from discretizations of ordinary differential equations occurring, for instance, in 2-point boundary value problems [9] or control problems [1]. We expect that they will lead to powerful tools for analyzing as well as solving problems in these application areas.

We want to stress here that in all examples it turned out to be sufficient to compute the bidiagonal B of the expression $A^{s_K} \cdots A^{s_1}$ and then the singular values of B, without any further iterative refinement. This is rather surprising. The bounds obtained on the accuracy of the bidiagonal are much worse than what was observed in the examples. This point and the connection to the Lanczos process need further analysis. That we get accurate approximations for the leading order bidiagonals might be useful when solving ill-posed or inverse problems.

The main advantage of the new method lies exactly in the fact that this bidiagonal is so accurate. If no iterative refinement is needed, then the method requires 5 to 10 times less flops than Kogbetliantz! If iterative refinement is needed, then the method should still be superior since the work then amounts essentially to the work of two Kogbetliantz steps.

Finally, we point out that there is recent work on computing singular values to high relative accuracy via the Kogbetliantz algorithm. This work is based on extracting particular scalings from the factors. So far this has been applied to problems involving three factors only. Extensions to several matrices and whether these methods perhaps could be combined with the bidiagonal approach in an advantageous way are still open problems. Those methods and the ideas developed in this paper are, we believe, related. In both methods grading in the factors, obtained either explicitly or implicitly, is important.

REFERENCES

[1] A. Bojanczyk, G. Golub, and P. Van Dooren, *The periodic Schur form. Algorithms and applications*, in Proceedings of the SPIE Conference, San Diego, CA, 1992, pp. 31–42.

[2] J. P. Charlier and P. Van Dooren, *On Kogbetliantz's SVD algorithm in the presence of clusters*, Linear Algebra Appl., 95 (1987), pp. 135–160.

[3] B. De Moor and P. Van Dooren, *Generalizations of the singular value and QR decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 993–1014.

[4] J. Demmel, M. Gu, S. Eisenstat, I. Slapnicar, and K. Veselic, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., submitted.

[5] K. V. Fernando and B. N. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[6] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[7] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[8] R. Gregory and D. Karney, *A Collection of Matrices for Testing Computational Algorithms*, Wiley-Interscience, New York, 1969.

[9] R. M. Mattheij and S. J. Wright, *Parallel stabilized compactification for ODEs with parameters and multipoint conditions*, Appl. Numer. Math., 13 (1993), pp. 305–333.

[10] C. B. Moler and G. W. Stewart, *An algorithm for the generalized matrix eigenvalue problem*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[11] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 20–32

CONVERGENCE OF NESTED ITERATIVE METHODS FOR SYMMETRIC P-REGULAR SPLITTINGS*

ZHI-HAO CAO†

Abstract. We study the convergence of nested iterative methods and present conditions on the splittings corresponding to the iterative methods to guarantee convergence for any number of the inner iterations. In contrast to Lanzkron, Rose, and Szyld [*Numer. Math.*, 58 (1991), pp. 685–702], Frommer and Szyld [*Numer. Math.*, 63 (1992), pp. 345–356; Numer. Math., 69 (1994), pp. 141–153], and Cao [*Math. Numer. Sinica*, 17 (1995), pp. 98–109; *Linear Algebra Appl.*, 22 (1995), pp. 159–170], in which the coefficient matrices are either of monotone matrices or of H-matrices and the splittings they set relate to regular and weak regular ones, the coefficient matrices considered in this paper are symmetric positive definite and the splittings we set relate to P-regular ones.

Key words. solution of linear systems, iterative methods, splittings, P-regular splitting, symmetric positive definite

AMS subject classifications. 65F10, 65F15

PII. S0895479897331229

1. Introduction.

Consider the iterative solution of a large linear system of equations

$$Ax = b \tag{1.1}$$

on parallel computers, where A is an $n \times n$ nonsingular matrix. Lanzkron, Rose, and Szyld [5] (see also Cao [1, 2], Frommer and Szyld [3, 4]) studied the convergence of nested iterative methods for solving (1.1). The conditions they presented on the corresponding splittings to guarantee convergence are related to regular and weak regular splittings. As a result, they have implicitly assumed that the matrix A in (1.1) is either a monotone matrix (i.e., $A^{-1} \geq 0$) or an H-matrix. Their proofs are based on the theory of nonnegative matrices. In this paper, we assume the matrix A in (1.1) is symmetric positive definite (s.p.d.). The conditions we will assume on the corresponding splittings to guarantee convergence are related to P-regular splittings. Our proofs are based on properties of s.p.d. matrices. Recently, Migallón and Penadés [6] also considered the convergence of two-stage iterative methods for Hermitian positive definite matrices based on P-regular splittings.

Let us partition the s.p.d. matrix A in (1.1) into $q \times q$ blocks

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1q} \\ A_{21} & A_{22} & \cdots & A_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ A_{q1} & A_{q2} & \cdots & A_{qq} \end{pmatrix} \tag{1.2}$$

with diagonal blocks A_{ii} being square of order $n_i, i = 1, \ldots, q$, and $\sum_{i=1}^{q} n_i = n$. Parallel computation makes block Jacobi type methods particular attractive. In such

*Received by the editors December 5, 1997; accepted for publication (in revised form) by M. Eiermann June 4, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/33122.html

†Department of Mathematics, Fudan University, Shanghai 200433, People's Republic of China (zcao@fudan.edu.cn). This work was supported by China State Major Key Project for basic research, the Experimental Foundation of Laboratory of Computational Physics, and the Doctoral Point Foundation of China.

methods, a splitting $A = M - N$ of A is used, where M is block diagonal, denoted by $M = \mathrm{diag}(M_i)$, with the blocks M_i being nonsingular of order $n_i, i = 1, \ldots, q$. The vectors x, b and other intermediate vectors are partitioned in a way consistent with (1.2). If splittings $M_i = F_i - G_i, i = 1, \ldots, q$, are used, then the block two-stage iterative method is the following (cf. [4]).

ALGORITHM 1.1 (block two-stage).

Given an initial vector $x_0 = [x_0^{(1)^T}, \ldots, x_0^{(q)^T}]^T$

for $k = 1, 2, \ldots,$
  for $i = 1, \ldots, q,$
    $y_0^{(i)} = x_{k-1}^{(i)},$
    for $j = 1, \ldots, p_{k,i},$
      $F_i y_j^{(i)} = G_i y_{j-1}^{(i)} + (N x_{k-1} + b)^{(i)},$
    $x_k^{(i)} = y_{p_{k,i}}^{(i)},$

where positive integers $p_{k,i}, k = 1, 2, \ldots, i = 1, \ldots, q,$ *are the numbers of inner iterations, which may depend on* k *and* i.

If $q = 1$ and $p_{k,i} = p$ for all k, then Algorithm 1.1 is called a (stationary) two-stage iterative method; if $q = 1$, and $p_{k,i} = p_k$, then Algorithm 1.1 is called a nonstationary two-stage iterative method.

2. Preliminaries. We begin with some basic notation and preliminary results which we refer to later.

A matrix $A \in \mathcal{C}^{n,n}$ is called positive definite if for all $x \in \mathcal{C}^n, x \neq 0$, one has $Re(x^H A x) > 0$. Obviously, if $A \in \mathcal{R}^{n,n}$, then A is positive definite if and only if $x^T A x > 0$ for all $x \in \mathcal{R}^n, x \neq 0$. We will use the notation $A \succ 0 (A \succeq 0)$ for a matrix A to be either Hermitian or symmetric positive (semi-) definite (cf. [7]).

A representation $A = M - N$ is called a splitting of A if M is nonsingular. A splitting $A = M - N$ is called convergent if $\rho(M^{-1}N) < 1$; here $\rho(Q)$ denotes the spectral radius of a matrix Q. Ortega [8] called a splitting $A = M - N$ P-regular if $M + N$ is positive definite.

LEMMA 2.1 (see [5]). *Given a nonsingular matrix* $A \in \mathcal{R}^{n,n}$ *and* $H \in \mathcal{R}^{n,n}$ *such that* $I - H$ *is nonsingular, there exists a unique pair of matrices* F, G, *such that* $H = F^{-1}G$ *and* $A = F - G$, *where* F *is nonsingular.*

In context of this lemma, we say that the iterative matrix H induces the splitting $A = F - G$.

LEMMA 2.2 (see [9]). *Let* $A \in \mathcal{R}^{n,n}$ *be symmetric and let* $A = M - N$ *be a P-regular splitting of* A. *Then* $\rho(M^{-1}N) < 1$ *if and only if* A *is s.p.d. (i.e.,* $A \succ 0$*).*

LEMMA 2.3 (see [7]). *Let* $A \in \mathcal{C}^{n,n}$ *be positive definite, and let* $A = M - N$ *be a P-regular splitting of* A. *Then* M *is positive definite.*

LEMMA 2.4 (see [7]). *Let* $A \succ 0$, *and let* $(M_1, N_1), (M_2, N_2)$ *be two splittings of* A. *If* $0 \preceq N_1 \preceq N_2$, *then*

$$\rho(M_1^{-1}N_1) \leq \rho(M_2^{-1}N_2) < 1.$$

3. Convergence of (stationary and nonstationary) two-stage and nested iterations. We first consider stationary two-stage methods. The convergence results for these methods are used later to analyze nested iterative methods, nonstationary two-stage iterative methods, and in the next section, to analyze block two-stage iterative methods.

In the case $q = 1$ and $p_{k,i} = p$ for all k, we have $A = M - N$ and $M = F - G$; Algorithm 1.1 is simplified to the following algorithm.

ALGORITHM 3.1 (stationary two-stage).

Given an initial vector x_0,
  for $k = 1, 2, \ldots,$
    $y_0 = x_{k-1}$,
    for $j = 1, \ldots, p,$
      $Fy_j = Gy_{j-1} + Nx_{k-1} + b,$
    $x_k = y_p,$

where the positive integer p *is the number of inner iterations. From the algorithm above, we have*

$$\begin{aligned} x_k &= (F^{-1}G)^p x_{k-1} + \sum_{j=0}^{p-1} (F^{-1}G)^j F^{-1}(Nx_{k-1} + b) \\ &= H^p x_{k-1} + (I - H^p)(I - H)^{-1} F^{-1}(Nx_{k-1} + b), \end{aligned} \tag{3.1}$$

where $H = F^{-1}G$, *and we have assumed* $I - H$ *is nonsingular. If we assume* $I - H^p$ *is also nonsingular and use Lemma* 2.1 *to derive a pair of matrices* B, C *such that* $B^{-1}C = (F^{-1}G)^p \equiv H^p$ *and* $M = B - C$, *then it is easy to show that*

$$B = F(I - H)(I - H^p)^{-1}, \qquad C = F(I - H)(I - H^p)^{-1}H^p. \tag{3.2}$$

From (3.1) and (3.2) we obtain the total (i.e., two-stage) iterative matrix of Algorithm 3.1,

$$\begin{aligned} T_p &= H^p + (I - H^p)(I - H)^{-1} F^{-1} N \\ &= H^p + B^{-1}N = B^{-1}(C + N). \end{aligned} \tag{3.3}$$

Then it is easy to show, by using Lemma 2.1, that the (unique) pair of matrices M_{T_p} and N_{T_p} induced by the two-stage iterative matrix T_p on A are

$$M_{T_p} = B, \qquad N_{T_p} = C + N. \tag{3.4}$$

THEOREM 3.1. *Let* A *be s.p.d., i.e.,* $A \succ 0$, *let* $A = M - N$ *be a symmetric P-regular splitting, and let* $M = F - G$ *be a symmetric convergent splitting; then the two-stage iterative method converges (i.e.,* $\rho(T_p) < 1$*), provided the inner iteration number* p *is even. Moreover,* $A = M_{T_p} - N_{T_p}$ *is a symmetric P-regular splitting.*

Proof. Since $A \succ 0$, and $A = M - N$ is a symmetric P-regular splitting, we have

$$M + N \succ 0 \tag{3.5}$$

and Lemma 2.3 implies M is s.p.d. (i.e., $M \succ 0$). Since $\rho(H) \equiv \rho(F^{-1}G) < 1$, the matrix $I - H^p$ is nonsingular. Thus matrices B and C in (3.2) induced by matrix H^p are well defined. We rewrite matrix B as

$$\begin{aligned} B &= F(I - F^{-1}G)(I - (F^{-1}G)^p)^{-1} \\ &= F(I - F^{-1}G) \sum_{j=0}^{\infty} (F^{-1}G)^{pj} = M(I - H^p)^{-1}, \end{aligned} \tag{3.6}$$

and from (3.6) we can see that B and hence C are symmetric.

We now consider $M_{T_p} + N_{T_p}$:

$$M_{T_p} + N_{T_p} = B + C + N = B + BH^p + N$$

which is also symmetric. From (3.6) we have

$$\begin{aligned} M_{T_p} + N_{T_p} &= B(I + H^p) + N \\ &= M \sum_{j=0}^{\infty} (H^p)^j (I + H^p) + N \\ &= 2M \sum_{j=1}^{\infty} H^{pj} + M + N. \end{aligned} \tag{3.7}$$

Note that p is even and we have, for $j = 1, 2, \ldots,$

$$\begin{aligned} MH^{pj} &= (F - G)(F^{-1}G)^{pj} = G(F^{-1}G)^{pj-1} - G(F^{-1}G)^{pj} \\ &= G(F^{-1}G)^{\frac{pj}{2}-1} F^{-1} G (F^{-1}G)^{\frac{pj}{2}-1} \\ &\quad - G(F^{-1}G)^{\frac{pj}{2}-1} F^{-1} G F^{-1} G (F^{-1}G)^{\frac{pj}{2}-1} \\ &= G(F^{-1}G)^{\frac{pj}{2}-1} (F^{-1} - F^{-1}GF^{-1}) G (F^{-1}G)^{\frac{pj}{2}-1} \\ &= (GF^{-1})^{\frac{pj}{2}} M (F^{-1}G)^{\frac{pj}{2}}. \end{aligned} \tag{3.8}$$

(3.8) implies, for all positive integers j, MH^{pj} is symmetric positive semidefinite, i.e., $MH^{pj} \succeq 0$. From (3.5) and (3.7) we have shown that $M_{T_p} + N_{T_p} \succ 0$, i.e., $A = M_{T_p} - N_{T_p}$ is a symmetric P-regular splitting of the s.p.d. matrix A. Lemma 2.2 implies $\rho(T_p) < 1$. Thus the proof is completed. □

The assumption that the inner iteration number p is even is important. Otherwise, even though we assume that $M = F - G$ is a symmetric P-regular splitting, the resulting two-stage iterative method may not converge if p is odd, as the following simple example shows.

EXAMPLE 3.1.

$$A = 5I, \quad M = 3I, \quad N = -2I; \quad F = 2I, \quad G = -I.$$

Obviously, $A = M - N$ and $M = F - G$ are both P-regular splittings. If $p = 1$, then

$$H = F^{-1}G = -\frac{1}{2}I, \quad B = M(I - H)^{-1} = 2I, \quad C = -I,$$

$$M_{T_1} = B = 2I, \quad N_{T_1} = C + N = -3I.$$

Thus, $M_{T_1} + N_{T_1} = -I$ is not s.p.d. and $\rho(T_1) = \frac{3}{2} > 1$.

Note that in Example 3.1 we still have $B + C = I \succ 0$, i.e., $M = B - C$ is a P-regular splitting of M. In the following we will give convergence results in which the inner iteration number p can be any positive integer.

THEOREM 3.2. *Let A be s.p.d., i.e., $A \succ 0$, let $A = M - N$ be a symmetric splitting such that N is symmetric positive semidefinite, i.e., $N \succeq 0$, let $M = F - G$ be a P-regular splitting, and let the inner iteration number p be any positive integer. Then the two-stage iterative method converges. Moreover, $A = M_{T_p} - N_{T_p}$ is a P-regular splitting. If the splitting $M = F - G$ is symmetric P-regular, then $A = M_{T_p} - N_{T_p}$ is a symmetric P-regular splitting.*

Proof. Since $N \succeq 0$, we have $M = A + N \succ 0$ and $M + N = A + 2N \succ 0$. Thus, $A = M - N$ is a symmetric P-regular splitting. Lemma 2.2 implies $\rho(H) \equiv \rho(F^{-1}G) < 1$ and $I - H^p$ is nonsingular. Thus, B and C in (3.2) are well defined. Moreover, we have

$$\begin{aligned} M - H^T M H &= M - (I - F^{-1}M)^T M (I - F^{-1}M) \\ &= (F^{-1}M)^T (F^T + G)(F^{-1}M) \succ 0. \end{aligned} \tag{3.9}$$

From (3.9) we deduce, for $j = 1, 2, \ldots,$

$$(H^T)^j M H^j - (H^T)^{j+1} M H^{j+1} = (F^{-1}MH^j)^T (F^T + G)(F^{-1}MH^j) \succeq 0. \tag{3.10}$$

(3.9) and (3.10) imply

$$M - (H^p)^T M H^p = \sum_{j=0}^{p-1} (F^{-1}MH^j)^T (F^T + G)(F^{-1}MH^j) \succ 0. \tag{3.11}$$

However, we have

$$\begin{aligned} M - (H^p)^T M H^p &= M - (I - B^{-1}M)^T M (I - B^{-1}M) \\ &= (B^{-1}M)^T (B^T + C)(B^{-1}M). \end{aligned} \tag{3.12}$$

Combining (3.11) and (3.12) we have shown $M = B - C$ is a P-regular splitting, i.e., $B^T + C \equiv B + B^T - M \succ 0$.

We now have

$$M_{T_p} + N_{T_p} = B^T + C + N \succ 0.$$

Thus, $A = M_{T_p} - N_{T_p}$ is a P-regular splitting. If $M = F - G$ is a symmetric P-regular splitting, then B and C in (3.2) are both symmetric and $A = M_{T_p} - N_{T_p}$ is a symmetric P-regular splitting. The proof of Theorem 3.2 is finished. □

REMARK 3.1. *Migallón and Penadés have obtained the same result (Theorem* 2.1 *in* [6]*) as Theorem* 3.2*. We retain Theorem* 3.2 *because it is used below in Theorem* 3.4*, Corollary* 3.6*, and Theorem* 5.1*, and its proof is used in Theorem* 4.1*.*

THEOREM 3.3. *Let A be s.p.d., i.e., $A \succ 0$, let $A = M - N$ be a symmetric P-regular splitting, let $M = F - G$ be a symmetric splitting such that G is symmetric positive semidefinite, i.e., $G \succeq 0$, and let inner iteration number p be any positive integer. Then the two-stage iterative method converges. Moreover, $A = M_{T_p} - N_{T_p}$ is a symmetric P-regular splitting.*

Proof. Lemma 2.3 implies $M \succ 0$. Since $G \succeq 0$, we have $F = M + G \succ 0$ and $F + G = M + 2G \succ 0$. Thus, $M = F - G$ is a symmetric P-regular splitting, Lemma 2.2 implies $\rho(H) \equiv \rho(F^{-1}G) < 1$, and hence $I - H^p$ is nonsingular. Thus B and C in (3.2) are well defined and are symmetric.

We now consider the terms $MH^{pj}, j = 1, 2, \ldots,$ in (3.7). If pj is even, then (cf.(3.8))

$$MH^{pj} \succeq 0. \tag{3.13}$$

If pj is odd, then we have

$$\begin{aligned} MH^{pj} &= (F-G)(F^{-1}G)^{pj} \\ &= G(F^{-1}G)^{pj-1} - G(F^{-1}G)^{pj} \\ &= (GF^{-1})^{\frac{pj-1}{2}} G(F^{-1}G)^{\frac{pj-1}{2}} \\ &\quad - (GF^{-1})^{\frac{pj-1}{2}} GF^{-1}G(F^{-1}G)^{\frac{pj-1}{2}} \\ &= (GF^{-1})^{\frac{pj-1}{2}} (G - GF^{-1}G)(F^{-1}G)^{\frac{pj-1}{2}}. \end{aligned} \tag{3.14}$$

However, we have

$$G - GF^{-1}G = G^{\frac{1}{2}}(I - G^{\frac{1}{2}}F^{-1}G^{\frac{1}{2}})G^{\frac{1}{2}} \succeq 0 \tag{3.15}$$

since $G^{\frac{1}{2}}F^{-1}G^{\frac{1}{2}} \succeq 0$ and $\rho(G^{\frac{1}{2}}F^{-1}G^{\frac{1}{2}}) = \rho(F^{-1}G) < 1$.

Combining (3.13), (3.14), and (3.15) we deduce that, for $j = 1, 2 \ldots,$

$$MH^{pj} \succeq 0. \tag{3.16}$$

From (3.16) and (3.7) we now have $M_{T_p} + N_{T_p} \succ 0$. Thus, the proof is finished. ∎

Since matrices M_{T_p} in all three theorems above are s.p.d., i.e., $M_{T_p} \succ 0$, the theory we are developing can be extended to the case of recursive inner iterations. Then we get nested iterative methods. For formal recursive definition of nested iterative methods cf. [5]. On the convergence of nested iterative methods we have the following theorem the proof of which is analogous to Corollary 4.7 in [5].

THEOREM 3.4. *Let A be s.p.d., i.e., $A \succ 0$. At each level let the redefined $A = M - N$ and at the inner most level let $M = \widehat{F} - \widehat{G}$ be*

(i) *a symmetric P-regular splitting and a symmetric convergent splitting, respectively, and at each level the "inner" iteration number (p) is an even positive integer, or*

(ii) *a symmetric splitting such that $N \succeq 0$ and a symmetric P-regular splitting, respectively, or*

(iii) *a symmetric P-regular splitting and a symmetric splitting such that $\widehat{G} \succeq 0$.*

Then the corresponding nested iterative method converges.

We now consider convergence of nonstationary two-stage iterative methods. In this case $p_{k,i} = p_k$. We have the following result.

THEOREM 3.5. *Let $A \succ 0$. Assume that the stationary two-stage iterative method converges (for any positive inner iteration number p) and $\rho(F^{-1}G) < 1$, $\rho(M^{-1}N) < 1$, where $A = M - N$ and $M = F - G$ are the outer and inner symmetric splittings, respectively. Then the nonstationary iterative method with any positive inner iteration number sequence $\{p_k\}$ converges, too.*

Proof. In nonstationary two-stage algorithms (3.1) is replaced by

$$x_k = (F^{-1}G)^{p_k} x_{k-1} + \sum_{j=0}^{p_k-1} (F^{-1}G)^j F^{-1}(Nx_{k-1} + b), \quad k = 1, 2, \ldots. \tag{3.17}$$

Let $x_* = A^{-1}b$ and $\epsilon_k = x_k - x_*$; then we have (cf. (3.3) and (3.17))

$$\epsilon_k = T_{p_k}\epsilon_{k-1} = T_{p_k} \ldots T_{p_1}\epsilon_0, \tag{3.18}$$

where (cf. (3.3) and (3.4))

$$T_{p_j} = M_{T_{p_j}}^{-1} N_{T_{p_j}} = I - M_{T_{p_j}}^{-1} A, \tag{3.19}$$

while (cf. (3.2), (3.4), and note that $\rho(H) \equiv \rho(F^{-1}G) < 1$)

$$\begin{aligned} M_{T_{p_j}}^{-1} &= (I - H^{p_j})(I - H)^{-1}F^{-1} = (I - H^{p_j})\sum_{i=0}^{\infty} H^i F^{-1} \\ &= \sum_{i=0}^{p_j-1} H^i F^{-1} = \sum_{i=0}^{p_j-1} (F^{-1}G)^i F^{-1}. \end{aligned} \tag{3.20}$$

We now define a vector norm $\|.\|_{A^{\frac{1}{2}}}$ by using the s.p.d. matrix A as

$$\|x\|_{A^{\frac{1}{2}}} = \|A^{\frac{1}{2}}x\|_2 \text{ for all } x \in \mathcal{R}^n. \tag{3.21}$$

It is easy to show that the induced matrix norm is

$$\|Q\|_{A^{\frac{1}{2}}} = \|A^{\frac{1}{2}}QA^{-\frac{1}{2}}\|_2 \text{ for all } Q \in \mathcal{R}^{n,n}. \tag{3.22}$$

From (3.19) we have

$$\|T_{p_j}\|_{A^{\frac{1}{2}}} = \|A^{\frac{1}{2}}T_{p_j}A^{-\frac{1}{2}}\|_2 = \rho(A^{\frac{1}{2}}T_{p_j}A^{-\frac{1}{2}}) = \rho(T_{p_j}) \tag{3.23}$$

since $A^{\frac{1}{2}}T_{p_j}A^{-\frac{1}{2}}$ is symmetric (cf. (3.19) and (3.20)):

$$\begin{aligned} A^{\frac{1}{2}}T_{p_j}A^{-\frac{1}{2}} &= I - A^{\frac{1}{2}}M_{T_{p_j}}^{-1}A^{\frac{1}{2}} \\ &= I - A^{\frac{1}{2}}\sum_{i=0}^{p_j-1}(F^{-1}G)^i F^{-1}A^{\frac{1}{2}}. \end{aligned} \tag{3.24}$$

For any positive integer p, T_p can be rewritten as (cf. (3.19) and (3.20))

$$T_p = I - (I - (F^{-1}G)^p)(I - M^{-1}N). \tag{3.25}$$

From (3.25) immediately we have

$$\lim_{p\to\infty} \rho(T_p) = \rho(M^{-1}N) < 1. \tag{3.26}$$

Thus, $\frac{1}{2}(1 - \rho(M^{-1}N)) > 0$ and there exists a positive integer p_0 such that if $p \geq p_0$, then

$$\rho(T_p) \leq \rho(M^{-1}N) + \frac{1}{2}(1 - \rho(M^{-1}N)) = \frac{1}{2}(1 + \rho(M^{-1}N)). \tag{3.27}$$

Let

$$\theta = \max\left(\{\rho(T_j), j = 1, \ldots, p_0 - 1\} \cup \left\{\frac{1}{2}(1 + \rho(M^{-1}N))\right\}\right). \tag{3.28}$$

Obviously, $\theta \in (0, 1)$. Then we have for all j

$$\|T_{p_j}\|_{A^{\frac{1}{2}}} \equiv \rho(T_{p_j}) \leq \theta. \tag{3.29}$$

Hence, (3.18) implies

$$\|\epsilon_k\|_{A^{\frac{1}{2}}} \leq \theta^k \|\epsilon_0\|_{A^{\frac{1}{2}}} \to 0 (k \to \infty),$$

i.e., the nonstationary two-stage iterative method converges. Thus the proof of the theorem is finished. □

From Theorem 3.5 we have the following corollary the proof of which is obvious.

COROLLARY 3.6. *Let $A \succ 0$. If splittings $A = M - N$ and $M = F - G$ satisfy the conditions in Theorem* 3.2 *or Theorem* 3.3 *or Theorem* 3.1*, then the corresponding nonstationary two-stage iterative method converges. But in the last case* (*i.e., Theorem* 3.1) *the inner iteration numbers $p_k, k = 1, 2, \ldots,$ in the corresponding nonstationary two-stage iterative method have to be even positive integers.*

4. Convergence of block two-stage iterations. In this section we consider the convergence of Algorithm 1.1. From this algorithm we have (cf. (3.1) and (3.17))

$$x_k^{(i)} = (F_i^{-1}G_i)^{p_{k,i}} x_{k-1}^{(i)} + \sum_{j=0}^{p_{k,i}-1} (F_i^{-1}G_i)^j F_i^{-1}(Nx_{k-1} + b)^{(i)}, \tag{4.1}$$

$$i = 1, \ldots, q,$$

where $A = M - N, M = \operatorname{diag}(M_i), M_i = F_i - G_i, i = 1, \ldots, q$.

THEOREM 4.1. *Let $A \succ 0$, let $A = M - N$ be a symmetric splitting such that M is block diagonal: $M = \operatorname{diag}(M_i)$ and $N \succeq 0$, let $M_i = F_i - G_i, i = 1, \ldots, q$, be symmetric P-regular splittings, and let $p_{k,i}, k = 1, 2, \ldots, i = 1, \ldots, q$, be arbitrary bounded positive integers, i.e., there exists a positive integer p_0 such that $1 \le p_{k,i} \le p_0$, for $k = 1, 2, \ldots, i = 1, \ldots, q$. Then Algorithm* 1.1 *converges.*

Proof. Let $H_i = F_i - G_i$ and for a fixed positive integer k let $B_i^{(k)} = M_i(I - H_i^{p_{k,i}})^{-1}, i = 1, \ldots, q$, and

$$M(k) = \operatorname{diag}(B_i^{(k)}). \tag{4.2}$$

Then the kth iterative matrix of Algorithm 1.1 is (cf. (3.17), (3.19), and (4.1))

$$T(k) = I - M(k)^{-1}A. \tag{4.3}$$

Since $I - T(k) \equiv M(k)^{-1}A$ is nonsingular, Lemma 2.1 implies that

$$A = M(k) - N(k) \tag{4.4}$$

is the splitting induced by $T(k)$, where

$$N(k) = \operatorname{diag}(B_i^{(k)} H_i^{p_{k,i}}) + N \equiv \operatorname{diag}(C_i^{(k)}) + N. \tag{4.5}$$

As in the proof of Theorem 3.2 we have

$$\operatorname{diag}(B_i^{(k)}) + \operatorname{diag}(C_i^{(k)}) \succ 0. \tag{4.6}$$

Thus

$$M(k) + N(k) = \operatorname{diag}(B_i^{(k)}) + \operatorname{diag}(C_i^{(k)}) + N \succ 0 \tag{4.7}$$

from which we know that (4.4) is a P-regular splitting of the s.p.d. matrix A; therefore, by Lemma 2.2 we have

$$\rho(T(k)) < 1. \tag{4.8}$$

By using an analogous argument as in the proof of Theorem 3.5 and noting that $1 \le p_{k,i} \le p_0$ we can show that

$$\|T(k)\|_{A^{\frac{1}{2}}} = \rho(T(k)) \le \theta, \tag{4.9}$$

where $\theta \in (0,1)$ is a constant independent of k.

Let $\epsilon_k = x_k - x_*$, where $x_* = A^{-1}b$. Since we have

$$\|\epsilon_k\|_{A^{\frac{1}{2}}} \le \prod_{j=1}^{k} \|T(j)\|_{A^{\frac{1}{2}}} \|\epsilon_0\|_{A^{\frac{1}{2}}} \le \theta^k \|\epsilon_0\|_{A^{\frac{1}{2}}}, \tag{4.10}$$

then

$$\lim_{k\to\infty} \|\epsilon_k\|_{A^{\frac{1}{2}}} = 0.$$

Thus, the proof of the theorem is completed. □

5. Monotonicity. In this section we consider the following question: When does the spectral radius $\rho(T_p)$ (cf. (3.3)) of the stationary two-stage iterative method become a monotonically decreasing function of p? We give the following result.

THEOREM 5.1. *Let A be s.p.d., i.e., $A \succ 0$, let $A = M - N$ be a symmetric splitting of A such that N is symmetric positive semidefinite, i.e., $N \succeq 0$, and let $M = F - G$ be a symmetric splitting of M such that G is symmetric positive semidefinite, i.e., $G \succeq 0$. Let two inner iterative numbers p and q satisfy $q \ge p > 0$; then*

$$\rho(T_q) \le \rho(T_p).$$

Proof. We have

$$M = A + N \succ 0 \quad \text{and} \quad F = M + G \succ 0. \tag{5.1}$$

Thus, $M = F - G$ is a P-regular splitting of an s.p.d. matrix M, Lemma 2.2 implies that $\rho(H) \equiv \rho(F^{-1}G) < 1$, and hence $I - H^p$ and $I - H^q$ are nonsingular.

From Theorem 3.2 or Theorem 3.3 we have

$$\rho(T_p) < 1 \quad \text{and} \quad \rho(T_q) < 1. \tag{5.2}$$

For any positive integer $l \ge 1$ we have

$$\begin{aligned} M_{T_l} &= M(I - H^l)^{-1} = M\sum_{j=0}^{\infty} H^{lj} = M + \sum_{j=1}^{\infty} MH^{lj}, \\ N_{T_l} &= M\sum_{j=1}^{\infty} H^{lj} + N = \sum_{j=1}^{\infty} MH^{lj} + N. \end{aligned} \tag{5.3}$$

If lj is even, then (cf. (3.8))

$$MH^{lj} = (GF^{-1})^{\frac{lj}{2}} M (F^{-1}G)^{\frac{lj}{2}} \succeq 0. \tag{5.4}$$

If lj is odd, then (cf. (3.14) and (3.15))

$$\begin{aligned} MH^{lj} &= (GF^{-1})^{\frac{lj-1}{2}} (G - GF^{-1}G)(F^{-1}G)^{\frac{lj-1}{2}} \\ &= (GF^{-1})^{\frac{lj-1}{2}} G^{\frac{1}{2}} (I - G^{\frac{1}{2}}F^{-1}G^{\frac{1}{2}}) G^{\frac{1}{2}} (F^{-1}G)^{\frac{lj-1}{2}} \succeq 0. \end{aligned} \tag{5.5}$$

By (5.3)–(5.5) and $N \succeq 0$ we have

$$N_{T_l} \succeq 0. \tag{5.6}$$

Hence

$$M_{T_l} = A + N_{T_l} \succ 0. \tag{5.7}$$

From (5.3) we obtain

$$\begin{aligned} M_{T_l}^{-1} &= (I - H^l)(I - H)^{-1}F^{-1} \\ &= (I - H^l)\sum_{j=0}^{\infty} H^j F^{-1} \\ &= \sum_{j=0}^{l-1} H^j F^{-1}. \end{aligned} \tag{5.8}$$

Obviously, we have

$$M_{T_{l+1}}^{-1} = M_{T_l}^{-1} + H^l F^{-1} = M_{T_l}^{-1} + (F^{-1}G)^l F^{-1}. \tag{5.9}$$

If l is even, then

$$(F^{-1}G)^l F^{-1} = (F^{-1}G)^{\frac{l}{2}} F^{-1} (GF^{-1})^{\frac{l}{2}} \succ 0. \tag{5.10}$$

If l is odd, then

$$\begin{aligned} (F^{-1}G)^l F^{-1} &= (F^{-1}G)^{\frac{l-1}{2}} F^{-1} G (F^{-1}G)^{\frac{l-1}{2}} F^{-1} \\ &= (F^{-1}G)^{\frac{l-1}{2}} F^{-1} G F^{-1} (GF^{-1})^{\frac{l-1}{2}} \\ &\succeq 0. \end{aligned} \tag{5.11}$$

By (5.9)–(5.11) we have

$$M_{T_{l+1}}^{-1} \succeq M_{T_l}^{-1}.$$

Therefore, if two positive integers p and q satisfy $q \geq p \geq 1$, then

$$M_{T_q}^{-1} \succeq M_{T_p}^{-1}$$

which is equivalent to the expression

$$N_{T_q} \preceq N_{Tp}. \tag{5.12}$$

From (5.12) and Lemma 2.4 we have

$$\rho(T_q) \leq \rho(T_p).$$

Thus, the proof is completed. □

In contrast to Theorem 3.2 and Theorem 3.3, the assumptions of Theorem 5.1 seem to be restrictive. However, the following examples show that we cannot weaken them.

EXAMPLE 5.1. *Let*

$$A = 5I, \quad M = 3I, \quad N = -2I; \quad F = 4I, \quad G = I.$$

Obviously, $A = M - N$ and $M = F - G$ are both P-regular splittings.

If $p = 1$, then

$$H_1 = F^{-1}G = \frac{1}{4}I, \quad B_1 = M(I - H_1)^{-1} = 4I, \quad C_1 = B_1H_1 = I,$$

$$M_{T_1} = B_1 = 4I, \qquad N_{T_1} = C_1 + N = -I.$$

Therefore, $A = M_{T_1} - N_{T_1}$ is a P-regular splitting of A, since $M_{T_1} + N_{T_1} = 3I$. We have

$$\rho(T_1) = \frac{1}{4}.$$

If $p = 2$, then

$$H_2 = (F^{-1}G)^2 = \frac{1}{16}I, \quad B_2 = M(I - H_2)^{-1} = \frac{16}{5}I, \quad C_2 = B_2H_2 = \frac{1}{5}I,$$

$$M_{T_2} = B_2 = \frac{16}{5}I, \qquad N_{T_2} = C_2 + N = -\frac{9}{5}I.$$

Therefore, $A = M_{T_2} - N_{T_2}$ is a P-regular splitting of A, since $M_{T_2} + N_{T_2} = \frac{7}{5}I$. We have

$$\rho(T_2) = \frac{9}{16}.$$

Thus, we have $\rho(T_1) < \rho(T_2)$.

EXAMPLE 5.2. *Let*

$$A = 5I, \quad M = 6I, \quad N = I; \qquad F = 4I, \quad G = -2I.$$

Obviously, $A = M - N$ and $M = F - G$ are both P-regular splittings.

If $p = 1$, then

$$H_1 = F^{-1}G = -\frac{1}{2}I, \quad B_1 = M(I - H_1)^{-1} = 4I, \quad C_1 = B_1H_1 = -2I,$$

$$M_{T_1} = B_1 = 4I, \qquad N_{T_1} = C_1 + N = -I.$$

Therefore, $A = M_{T_1} - N_{T_1}$ is a P-regular splitting of A, since $M_{T_1} + N_{T_1} = 3I$. We have

$$\rho(T_1) = \frac{1}{4}.$$

If $p = 2$, then

$$H_2 = (F^{-1}G)^2 = \frac{1}{4}I, \quad B_2 = M(I - H_2)^{-1} = 8I, \quad C_2 = B_2H_2 = 2I,$$

$$M_{T_2} = B_2 = 8I, \qquad N_{T_2} = C_2 + N = 3I.$$

Therefore, $A = M_{T_2} - N_{T_2}$ is a P-regular splitting of A, since $M_{T_2} + N_{T_2} = 11I$. We have

$$\rho(T_2) = \frac{3}{8}.$$

Thus, we have $\rho(T_1) < \rho(T_2)$.

At the end of the paper we briefly discuss the choice of the outer and inner splittings.

An important outer splitting $A = M - N$ is the SSOR splitting

$$M = \frac{1}{\omega(2-\omega)}(D - \omega L)D^{-1}(D - \omega L^T),$$

$$N = \frac{1}{\omega(2-\omega)}[(1-\omega)D + \omega L]D^{-1}[(1-\omega)D + \omega L^T],$$

where $A = D - L - L^T, D$ is the diagonal of $A, -L$ is the lower triangular part of A. Obviously, if $\omega \in (0, 2)$, then $M \succ 0$ and $N \succeq 0$. Thus, the SSOR splitting is a symmetric P-regular splitting with $N \succeq 0$.

As pointed out in [6] a simple way to construct an outer splitting $A = M - N$ such that M is symmetric and $N \succeq 0$ is to split the s.p.d. matrix A into $A = M_1 - N_1$, where M_1 is symmetric, and let D_1 be a nonnegative diagonal matrix such that $D_1 + N_1 \succeq 0$. Then, the splitting $A = M - N$ with $M = M_1 + D_1$ and $N = N_1 + D_1$ is a symmetric P-regular splitting.

Since M is always s.p.d., the method above to construct the symmetric P-regular outer splittings of the s.p.d. matrix A can also be used to construct the symmetric P-regular inner splittings of the s.p.d. matrix M. For the nonsymmetric P-regular inner splitting (e.g., in Theorem 3.2) $M = F - G$ an important example is the SOR splitting:

$$F = \frac{1}{\omega}(\widetilde{D} - \omega\widetilde{L}),$$

$$G = \frac{1}{\omega}[(1-\omega)\widetilde{D} + \omega\widetilde{L}^T],$$

where $M = \widetilde{D} - \widetilde{L} - \widetilde{L}^T$, $\widetilde{D}$ is the diagonal of M, $\widetilde{L}$ is the lower triangular part of M. Obviously, if $\omega \in (0, 2)$, then $F^T + G = \frac{2-\omega}{\omega}\widetilde{D} \succ 0$. Thus, the SOR splitting is P-regular.

Acknowledgments. I am grateful to the referees and the editor for their helpful comments and suggestions and also for pointing out reference [6], which significantly improved the paper.

REFERENCES

[1] Z.-H. Cao, *Convergence of two-stage iterative methods for the solution of linear systems*, Math. Numer. Sinica, 17 (1995), pp. 98–109 (in Chinese).

[2] Z.-H. Cao, *On convergence of nested stationary iterative methods*, Linear Algebra Appl., 221 (1995), pp. 159–170.

[3] A. Frommer and D.B. Szyld, *H-splittings and two-stage iterative methods*, Numer Math., 63 (1992), pp. 345–356.

[4] A. Frommer and D.B. Szyld, *Asynchronous two-stage iterative methods*, Numer Math., 69 (1994), pp. 141–153.

[5] P.J. Lanzkron, D.J. Rose, and D.B. Szyld, *Convergence of nested classical iterative methods for linear systems*, Numer. Math., 58 (1991), pp. 685–702.
[6] V. Migallón and J. Penadés, *Convergence of two-stage iterative method for Hermitian positive definite matrices*, Appl. Math. Letters, 10 (1997), pp. 79–83.
[7] R. Nabben, *A note on comparison theorems for splittings and multisplittings of Hermitian positive definite matrices*, Linear Algebra Appl., 233 (1996), pp. 67–80.
[8] J.M. Ortega, *Numerical Analysis—A Second Course*, Academic Press, New York, 1972.
[9] J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 33–40

HOUSEHOLDER TRANSFORMATIONS REVISITED*

A. A. DUBRULLE†

Abstract. A new analysis of the two types of Householder reflections shows that the second type stably defined by Parlett has a better ability to propagate information borne by its driving vector. The results are extended to the Davis–Kahan rotation.

Key words. Householder matrices, elementary Hermitian matrices, elementary reflectors, rotators, orthogonalization

AMS subject classifications. 65F05, 65F15, 65F25, 65F30

PII. S0895479898338561

1. Introduction.

Householder matrices, or elementary reflectors, seem to have first appeared in [11], but they truly became a standard tool of numerical linear algebra with Householder's 1958 article on the triangularization of a nonsymmetric matrix [7]. An elementary reflector $\boldsymbol{H} : \mathbb{C}^n \leftrightarrow \mathbb{C}^n$ is a rank-one modification of the identity matrix and has the canonical form

$$\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^*, \qquad \|\boldsymbol{u}\| = \sqrt{2}, \qquad \boldsymbol{u} \in \mathbb{C}^n,$$

where $\|.\|$ designates the Euclidean norm. It is Hermitian, unitary, and involutory (each of these properties is the consequence of the other two). Elementary reflectors are used in numerical algorithms for the construction of orthogonal bases for which problems take forms amenable to simple solutions. From a computational viewpoint, such transformations originate in the annihilation of selected elements of vectors or matrices and are represented by the isometric mapping of a driving vector $\boldsymbol{z}$ into a stretching of a vector of the canonical basis:

$$\boldsymbol{H}_k\boldsymbol{z} = \beta\boldsymbol{e}_k, \qquad |\beta| = \|\boldsymbol{z}\|, \qquad \boldsymbol{z} \in \mathbb{C}^n.$$

The transformation matrix has the expression

$$\boldsymbol{H}_k = \boldsymbol{I} - \frac{1}{\bar{\beta}\,(\beta - z_k)}(\boldsymbol{z} - \beta\boldsymbol{e}_k)(\boldsymbol{z} - \beta\boldsymbol{e}_k)^*, \tag{1.1}$$

where β is defined for the moment only by its modulus and the hermiticity relation

$$\bar{\beta}z_k = \pm|z_k|\,\|\boldsymbol{z}\|. \tag{1.2}$$

For purposes of algorithm implementation, $\boldsymbol{H}_k$ is most commonly represented by one or two vectors $\boldsymbol{v}$ or $\boldsymbol{w}$ as follows:

$$\left.\begin{aligned} \boldsymbol{H}_k &= \boldsymbol{I} + \boldsymbol{v}\boldsymbol{w}^*, \\ \boldsymbol{H}_k &= \boldsymbol{I} + \frac{1}{w_k}\boldsymbol{w}\boldsymbol{w}^*, \\ \boldsymbol{H}_k &= \boldsymbol{I} + \bar{w}_k\boldsymbol{v}\boldsymbol{v}^*, \end{aligned}\right\} \quad \boldsymbol{v} = \frac{\boldsymbol{z} - \beta\boldsymbol{e}_k}{z_k - \beta}, \qquad \boldsymbol{w} = \frac{\boldsymbol{z} - \beta\boldsymbol{e}_k}{\beta}. \tag{1.3}$$

*Received by the editors May 11, 1998; accepted for publication (in revised form) by G. Golub December 6, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/33856.html
†215 Hillview Ave., Los Altos, CA 94022 (na.dubrulle@na-net.ornl.gov).

The above formulas verify

$$\|\boldsymbol{w}\|^2 = -2w_k, \qquad \|\boldsymbol{v}\|^2 = -\frac{2}{w_k}.$$

When square roots can be computed accurately and reasonably fast, the representation

$$\boldsymbol{H}_k = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^*, \qquad \boldsymbol{u} = \frac{\boldsymbol{z} - \beta\boldsymbol{e}_k}{\left[\bar{\beta}\,(\beta - z_k)\right]^{1/2}}, \qquad \|\boldsymbol{u}\| = \sqrt{2}, \tag{1.4}$$

has the advantage of substituting a square root for multiple scaling operations in the transformation of matrices.[1]

An important application of Householder reflectors is found in the implementation of the implicit QR algorithm for the solution of the Hessenberg eigenvalue problem [1], [2], [5]. There, multiple shifts of the spectrum are injected into the iteration and carried through the QR sweep by Householder similarity transformation. As the QR iteration makes the trailing components of the driving vectors of these transformations dwindle, it was conjectured in [5] that the inability of the common type of Householder matrix to carry the information borne by these small components was contributing to the instability of the algorithm for moderately large numbers of shifts. An investigation of this conjecture—so far inconclusive—led to the consideration of the uncommon type and the new analysis presented below.

Section 2 describes the two types of Householder matrices and summarizes known results. Section 3 analyzes their ability to propagate information. Section 4 extends the results of the analysis to the Davis–Kahan rotations.

2. The two types of elementary reflectors. Equation (1.2) prescribes that $\bar{\beta}z_k$ must be real and we have

$$\beta = \pm\frac{z_k}{|z_k|}\|\boldsymbol{z}\|, \qquad z_k \neq 0,$$

with the arbitrary choice $\beta = \|\boldsymbol{z}\|$ when $z_k = 0$. For the stability of formula (1.1), it is generally recommended that the minus sign be used. Parlett showed, however, that the other choice is not unstable at all if an appropriate formula is used for the computation of $(\beta - z_k)$, and he developed a norm error analysis of the corresponding reflector [8]. This work has been so far largely ignored, perhaps because of a lack of obvious software application.

We thus have two types of elementary reflectors determined by the sign of β. They are defined below.

(1) *First type:* $\beta = -\dfrac{z_k}{|z_k|}\|\boldsymbol{z}\|$.

This option maximizes $|\beta - z_k|$ and generates the matrix

$$\boldsymbol{H}_k^{(1)} = \boldsymbol{I} - \frac{1}{\|\boldsymbol{z}\|(\|\boldsymbol{z}\| + |z_k|)}(\boldsymbol{z} - \beta\boldsymbol{e}_k)(\boldsymbol{z} - \beta\boldsymbol{e}_k)^*.$$

This is the type commonly used, which is stable for the computation of $(\beta - z_k)$ by straightforward subtraction. $\boldsymbol{H}_k^{(1)}$ is the reflection in the outer bisector of

[1] Taking n extra square roots saves about $0.5n^2$ multiplications for the QR factorization of a matrix of order n, $1.5n^2$ multiplications for the reduction to Hessenberg form, and $2n^2$ multiplications if the associated transformation of the eigenvectors is performed.

the angle $\angle(\boldsymbol{z}, z_k\boldsymbol{e}_k)$. The corresponding vector $\boldsymbol{u}$ of representation (1.4) is expressed by

$$
\begin{aligned}
u_k^{(1)} &= \frac{z_k}{|z_k|}\left(1+\frac{|z_k|}{\|\boldsymbol{z}\|}\right)^{1/2},\\
u_i^{(1)} &= \frac{z_i}{\|\boldsymbol{z}\|}\left(1+\frac{|z_k|}{\|\boldsymbol{z}\|}\right)^{-1/2}, \qquad i\neq k,
\end{aligned}
\tag{2.1}
$$

with $\|\boldsymbol{u}\|_\infty = |u_k|$.

(2) *Second type:* $\beta = \dfrac{z_k}{|z_k|}\|\boldsymbol{z}\|$.

This option is practically never used. Parlett's analysis shows that the instability is not in the choice of sign for β but in the computation of $(\beta - z_k)$ by floating-point subtraction. The following formula is a stable substitute,

$$\beta - z_k = \frac{z_k}{|z_k|}\,\frac{\|\boldsymbol{z}-z_k\boldsymbol{e}_k\|^2}{\|\boldsymbol{z}\|+|z_k|},$$

where the squared norm in the numerator is computed as the sum of the squares of the components of $\boldsymbol{z}$ other than z_k (an intermediate step in the calculation of $\|\boldsymbol{z}\|$). The corresponding expression of the matrix is

$$\boldsymbol{H}_k^{(2)} = \boldsymbol{I} - \frac{1+\dfrac{|z_k|}{\|\boldsymbol{z}\|}}{\|\boldsymbol{z}-z_k\boldsymbol{e}_k\|^2}(\boldsymbol{z}-\beta\boldsymbol{e}_k)(\boldsymbol{z}-\beta\boldsymbol{e}_k)^*.$$

$\boldsymbol{H}_k^{(2)}$ is the reflection in the inner bisector of the angle $\angle(\boldsymbol{z}, z_k\boldsymbol{e}_k)$. The vector $\boldsymbol{u}$ of form (1.4) is expressed by

$$
\begin{aligned}
u_k^{(2)} &= -\frac{z_k}{|z_k|}\,\frac{\|\boldsymbol{z}-z_k\boldsymbol{e}_k\|}{\|\boldsymbol{z}\|}\left(1+\frac{|z_k|}{\|\boldsymbol{z}\|}\right)^{-1/2},\\
u_i^{(2)} &= \frac{z_i}{\|\boldsymbol{z}-z_k\boldsymbol{e}_k\|}\left(1+\frac{|z_k|}{\|\boldsymbol{z}\|}\right)^{1/2}, \qquad i\neq k.
\end{aligned}
\tag{2.2}
$$

The two types are backward stable in norm [8]. The second type requires a little more arithmetic work, which is usually negligible in the context of larger computations such as the solutions of linear equations and eigenvalue problems. Formulas (2.1) and (2.2) show major differences in the roles played by z_k and $\{z_i\}_{i\neq k}$ in the two types of transformations. The first type inflates the relative importance of z_k at the expense of $\{z_i\}_{i\neq k}$ through reciprocal scalings by $(1+|z_k|/\|\boldsymbol{z}\|)^{1/2}$ and normalization by $\|\boldsymbol{z}\|$. The opposite is true for the second type where the scaling is reversed and $\{z_i\}_{i\neq k}$ is normalized by $\|\boldsymbol{z}-z_k\boldsymbol{e}_k\|$. These differences constitute the main motivation behind the analysis of the next section.

3. Elementary reflectors as information carriers. The transformation of a vector or a matrix by a Householder reflector can be seen as the propagation of the information borne by the driving vector $\boldsymbol{z}$. This interpretation is best illustrated by the injection of shifts in the implicit QR iteration [2] through a similarity Householder

transformation. In the following, we examine how well elementary reflectors carry information.

Consider the reflector in form (1.4) defined by

$$\boldsymbol{H}_k = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^*, \qquad \boldsymbol{H}_k\boldsymbol{z} = \beta\boldsymbol{e}_k, \qquad |\beta| = \|\boldsymbol{z}\|, \qquad \|\boldsymbol{u}\| = \sqrt{2},$$

and its application to some nonzero arbitrary vector $\boldsymbol{x}$,

$$\boldsymbol{y} = \boldsymbol{x} - (\boldsymbol{u}^*\boldsymbol{x})\boldsymbol{u}. \tag{3.1}$$

We propose to assess the ability of $\boldsymbol{H}_k$ to carry to $\boldsymbol{y}$ the information contained in $\boldsymbol{z}$ by measuring the effects of the $\boldsymbol{z}$ components on $\boldsymbol{x}$ through $\boldsymbol{u}$. As needed, we use the notation $\boldsymbol{u} = \boldsymbol{u}^{(i)}$, $i = 1, 2$, according to the type of transformation, with

$$\boldsymbol{u}^{(1)*}\boldsymbol{u}^{(2)} = 0.$$

Componentwise, we have

$$y_i = x_i - (\boldsymbol{u}^*\boldsymbol{x})u_i, \tag{3.2}$$

and the floating-point invariance of x_i under transformation by $\boldsymbol{H}_k$ will be construed as a failure to transmit to y_i the information borne by u_i when $u_i \neq 0$. In the following, we develop conditions for which this loss occurs, and we shall see that these conditions substantially differ for the two types of elementary reflectors.

We first investigate the most obvious situations in which invariance takes place. When

$$\boldsymbol{u}^*\boldsymbol{x} = 0,$$

no information is propagated from $\boldsymbol{z}$ to $\boldsymbol{y}$. If the underlying transformation is of, say, type 1 with

$$\boldsymbol{u} = \boldsymbol{u}^{(1)},$$

then $\boldsymbol{x}$ and $\boldsymbol{u}^{(2)}$ are in the subspace orthogonal to $\boldsymbol{u}^{(1)}$. Moreover, since

$$x_k = \frac{1}{2}\left[(\boldsymbol{u}^{(1)*}\boldsymbol{x})u_k^{(1)} + (\boldsymbol{u}^{(2)*}\boldsymbol{x})u_k^{(2)}\right],$$

the condition $x_k \neq 0$ necessarily implies that $\boldsymbol{u}^{(2)}$ is not orthogonal to $\boldsymbol{x}$ and that the transformation of type 2 will carry as much information from $\boldsymbol{z}$ to $\boldsymbol{y}$ as allowed by other conditions to be examined later. If $x_k = 0$, total loss of information occurs for both types. Obviously, the same argument applies if we exchange the transformation types.

Having disposed of these cases, we assume that

$$\boldsymbol{u}^*\boldsymbol{x} \neq 0,$$

and we turn to the special case of y_k. Using the expression (1.1) of $\boldsymbol{H}_k$, we get the formula[2]

$$y_k = \frac{\boldsymbol{z}^*\boldsymbol{x}}{\beta}.$$

[2]This formula is shown in [10] to produce a smaller bound for the forward floating-point rounding error than expression (3.2).

Hence, to working precision, the two types equally propagate to y_k the information borne by $\boldsymbol{z}$ since the two results differ only by their signs.

For the other components, invariance to floating-point machine precision ε is expressed by

$$\frac{|y_i - x_i|}{|x_i|} \leq \varepsilon, \qquad x_i \neq 0, \qquad i \neq k,$$

which yields

$$|u_i| \leq \varepsilon \frac{|x_i|}{|\boldsymbol{u}^* \boldsymbol{x}|}, \qquad i \neq k,$$

from (3.2). Using the Cauchy–Schwarz inequality and $\|\boldsymbol{u}\| = \sqrt{2}$, we obtain the sufficient condition

$$|u_i| \leq \frac{\varepsilon}{\sqrt{2}} \frac{|x_i|}{\|\boldsymbol{x}\|}.$$

The substitution of definition (1.4) in this inequality gives the condition of invariance in terms of the $\boldsymbol{z}$ components:

$$\frac{|z_i|}{[\bar{\beta}\,(\beta - z_k)]^{1/2}} \leq \frac{\varepsilon}{\sqrt{2}} \frac{|x_i|}{\|\boldsymbol{x}\|}, \qquad i \neq k. \tag{3.3}$$

We examine below the implications of the invariance condition for the two types of reflections.

(1) *First type*: $\beta = -\dfrac{z_k}{|z_k|}$.

In this case,

$$\bar{\beta}(\beta - z_k) = \|\boldsymbol{z}\|(\|\boldsymbol{z}\| + |z_k|),$$

and the substitution of this expression in inequality (3.3) leads to the invariance condition

$$\frac{|z_i|}{\|\boldsymbol{z}\|} \leq \frac{\varepsilon}{\sqrt{2}} \sqrt{1 + \frac{|z_k|}{\|\boldsymbol{z}\|}} \frac{|x_i|}{\|\boldsymbol{x}\|}, \qquad i \neq k. \tag{3.4}$$

This inequality is satisfied for small components of $\boldsymbol{z}$ that are not matched by proportionately small components of $\boldsymbol{x}$. The size of z_i, $i \neq k$, is measured relatively to the norm of $\boldsymbol{z}$.

(2) *Second type*: $\beta = \dfrac{z_k}{|z_k|}$.

The identity

$$\bar{\beta}(\beta - z_k) = \frac{\|\boldsymbol{z}\|\,\|\boldsymbol{z} - z_k \boldsymbol{e}_k\|^2}{\|z\| + |z_k|}$$

applies, and its combination with inequality (3.3) yields the invariance criterion

$$\frac{|z_i|}{\|\boldsymbol{z} - z_k \boldsymbol{e}_k\|} \sqrt{1 + \frac{|z_k|}{\|\boldsymbol{z}\|}} \leq \frac{\varepsilon}{\sqrt{2}} \frac{|x_i|}{\|\boldsymbol{x}\|}, \qquad i \neq k.$$

```
function u=HVEC(z,k,type)

t=norm(z,inf);
u=z(:)/(t+(t==0));
if t~=0
   uk=u(k);
   u(k)=0;
   s2=u'*u;
   if s2~=0
      sig=sign(uk)+(uk==0);
      t=real(sig*conj(uk));
      s=sqrt(t^2+s2);
      t=t+s;
      if type==1
         u(k)=sig*t;
      else
         t=s2/t;
         u(k)=-sig*t;
      end
   end
   u=u/sqrt(s*t);
 end
end
return
```

FIG. 3.1. *MATLAB computation of the Householder vector* $\boldsymbol{u}$ *for either type of elementary reflector* $\boldsymbol{H}_k = \boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^*$ *such that* $\boldsymbol{H}_k\boldsymbol{z} = \beta \boldsymbol{e}_k$.

Here, the size of z_i, $i \neq k$, is measured relatively to the norm of $(\boldsymbol{z} - z_k\boldsymbol{e}_k)$. This condition is harder to satisfy than its first-type counterpart (3.4), particularly when z_k is the dominant component of $\boldsymbol{z}$. This advantage is maximum when the directions of $\boldsymbol{z}$ and $\boldsymbol{e}_k$ are nearly parallel and disappears when they are orthogonal.

In summary, the second type is better at propagating the information contained in each z_i for $i \neq k$. The following example where the Householder vectors are computed with the MATLAB function in Figure 3.1 illustrates this phenomenon:

$$k = 1, \qquad \boldsymbol{z} = [\,1 \quad 6\eta \quad 2\eta\,]^T, \qquad \eta = \varepsilon/8, \qquad \boldsymbol{x} = \boldsymbol{e}.$$

(1) *First type*:
The invariance criterion is satisfied for $i > 1$, and the information borne by z_2 and z_3 is lost in the floating-point representation $f_\varepsilon(\boldsymbol{y})$ of $\boldsymbol{y}$:

$$f_\varepsilon(\boldsymbol{y}) = [\,-1 \quad 1 \quad 1\,]^T.$$

(2) *Second type*:
The invariance test is not satisfied, and the transformation preserves all subspace information:

$$f_\varepsilon(\boldsymbol{y}) = [\,1+\varepsilon \quad -1.4 \quad 0.2\,]^T.$$

4. Application to the Davis–Kahan rotations. In this section, we show that the simple relationship between Davis–Kahan rotations and Householder reflections makes these transformations numerically equivalent, with identical properties to propagate information.

The Davis–Kahan rotation matrix [4], [9] in $\mathbb{C}^{n\times n}$ can be written in the form

$$\boldsymbol{R}_1 = \begin{bmatrix} c & -\bar{s}\,\boldsymbol{\omega}^* \\ s\,\boldsymbol{\omega} & \boldsymbol{I} - \theta\boldsymbol{\omega}\boldsymbol{\omega}^* \end{bmatrix}, \qquad \begin{cases} \theta = 1 - \bar{c}, & |c|^2 + |s|^2 = 1, \\ \boldsymbol{\omega} \in \mathbb{C}^{n-1}, & \|\boldsymbol{\omega}\| = 1. \end{cases}$$

The above representation conveniently expresses how the Davis–Kahan rotation generalizes the plane rotation

$$\begin{bmatrix} c & -\bar{s} \\ s & \bar{c} \end{bmatrix}.$$

It is the simple relationship between plane rotations and reflections that naturally leads to the equally simple connection shown below between Davis–Kahan and Householder matrices.

$\boldsymbol{R}_1$ is a rank-two modification of the identity matrix and not an elementary matrix. Its eigenvalues are unity, except for the pair $c \pm is$. For a vector $\boldsymbol{z}$ and its projection $\boldsymbol{z}_{2:n}$ on the subspace spanned by $\{\boldsymbol{e}_2, \boldsymbol{e}_3, \ldots, \boldsymbol{e}_n\}$,

$$\boldsymbol{R}_1\boldsymbol{z} = \beta\boldsymbol{e}_1, \qquad \beta = \sigma\|\boldsymbol{z}\|, \qquad |\sigma| = 1,$$

for the settings

$$c = \sigma\frac{\bar{z}_1}{\|z\|}, \qquad s\boldsymbol{\omega} = -\bar{\sigma}\,\frac{\boldsymbol{z}_{2:n}}{\|\boldsymbol{z}\|}, \qquad |s| = \frac{\|\boldsymbol{z}_{2:n}\|}{\|\boldsymbol{z}\|}.$$

Note that s is defined only by its modulus. Among the possible choices for σ,

$$\sigma = \pm\frac{z_1}{|z_1|}$$

generates rotations that parallel the two types of Householder reflections. In fact, $\boldsymbol{R}_1$ and $\boldsymbol{H}_1$ differ only by the signs of their first rows, and the transformations of a vector $\boldsymbol{x}$ by $\boldsymbol{R}_1$ and $\boldsymbol{H}_1$ produce two vectors that differ only by the signs of their first components.

More generally, $\boldsymbol{R}_k$ such that $\boldsymbol{R}_k\boldsymbol{z} = \beta\boldsymbol{e}_k$ derives from $\boldsymbol{H}_k$ by a sign change of row k,

$$\boldsymbol{e}_k^T\boldsymbol{R}_k = -\boldsymbol{e}_k^T\boldsymbol{H}_k.$$

This relation provides a simple way to define the rotation matrix and its representation by a single vector. It follows that the corresponding two types of rotations have exactly the same numerical properties as their reflection homologues, including ability to carry information. In most numerical algorithms, reflections and rotations are practically interchangeable, but the multiplicative-group property of the latter may make them preferable for certain applications.

5. Conclusion. Our analysis of Householder transformations reveals that the second type has a better ability to propagate the information borne by its driving vector. Cox and Higham [3] show, however, that the use of the second type in QR

factorizations with pivoting for size by row and column exchanges may lead to row-wise instability, although the algorithm is normwise and columnwise stable. This suggests that the two types of transformations serve different purposes and should be used accordingly.

The use of the criteria of invariance of section 3 for the dynamic adjustment of the iteration multiplicity of the QR algorithm is described in [6]. These tests are rather conservative, owing to the use of the Cauchy–Schwarz inequality to bound the scalar product. Tighter bounds that would produce laxer tests did not lead to formulas efficient enough for practical application.

Acknowledgment. Beresford Parlett is gratefully acknowledged for his insightful comments.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, PA, 1995.

[2] Z. Bai and J. Demmel, *On a block implementation of Hessenberg multishift QR iteration*, Int. J. High-Speed Comput., 62 (1989), pp. 209–226.

[3] A. Cox and N. Higham, *Stability of Householder QR Factorization for Weighted Least-Squares Problems*, Numerical Analysis TR 301, Dept. of Mathematics, University of Manchester, UK, 1997.

[4] C. Davis and W. Kahan, *Some new bounds on perturbation of subspaces*, Bull. Amer. Math. Soc., 75 (1969), pp. 863–868.

[5] A. Dubrulle, *The Multishift QR Algorithm: Is It Worth the Trouble?*, TR G320-3558, IBM Scientific Center, Palo Alto, CA, 1991 (revised 1992).

[6] A. Dubrulle, *A QR algorithm with variable iteration multiplicity*, J. Comp. Appl. Math., 86 (1997), pp. 125–139.

[7] A. Householder, *Unitary triangularization of a nonsymmetric matrix*, J. ACM, 5 (1958), pp. 339–342.

[8] B.N. Parlett, *Analysis of algorithms for reflections in bisectors*, SIAM Rev., 13 (1971), pp. 197–208.

[9] B.N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998.

[10] N.-K. Tsao, *A note on implementing the Householder transformation*, SIAM J. Numer. Anal., 12 (1975), pp. 53–58.

[11] H. Turnbull and A. Aitken, *An Introduction to the Theory of Canonical Matrices*, Blackie, London, Glasgow, 1932.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 41–55

ON LAGRANGIAN RELAXATION OF QUADRATIC MATRIX CONSTRAINTS*

KURT ANSTREICHER† AND HENRY WOLKOWICZ‡

Abstract. Quadratically constrained quadratic programs (QQPs) play an important modeling role for many diverse problems. These problems are in general NP hard and numerically intractable. Lagrangian relaxations often provide good approximate solutions to these hard problems. Such relaxations are equivalent to semidefinite programming relaxations.

For several special cases of QQP, e.g., convex programs and trust region subproblems, the Lagrangian relaxation provides the exact optimal value, i.e., there is a zero duality gap. However, this is not true for the general QQP, or even the QQP with two convex constraints, but a nonconvex objective.

In this paper we consider a certain QQP where the quadratic constraints correspond to the matrix orthogonality condition $XX^T = I$. For this problem we show that the Lagrangian dual based on relaxing the constraints $XX^T = I$ *and* the seemingly redundant constraints $X^TX = I$ has a zero duality gap. This result has natural applications to quadratic assignment and graph partitioning problems, as well as the problem of minimizing the weighted sum of the largest eigenvalues of a matrix. We also show that the technique of relaxing quadratic matrix constraints can be used to obtain a strengthened semidefinite relaxation for the max-cut problem.

Key words. Lagrangian relaxations, quadratically constrained quadratic programs, semidefinite programming, quadratic assignment, graph partitioning, max-cut problems

AMS subject classifications. 49M40, 52A41, 90C20, 90C27

PII. S0895479898340299

1. Introduction.

Quadratically constrained quadratic programs (QQPs) play an important modeling role for many diverse problems. They often provide a much improved model compared to the simpler linear relaxation of a problem. However, very large linear models can be solved efficiently, whereas QQPs are in general NP-hard and numerically intractable. Lagrangian relaxations often provide good approximate solutions to these hard problems. Moreover these relaxations can be shown to be equivalent to semidefinite programming (SDP) relaxations, and SDP problems can be solved efficiently, i.e., they are polynomial time problems; see, e.g., [31].

SDP relaxations provide a tractable approach for finding good bounds for many hard combinatorial problems. The best example is the application of SDP to the max-cut problem, where a 87% performance guarantee exists [11, 12]. Other examples include matrix completion problems [23, 22], as well as graph partitioning problems and the quadratic assignment problem (references given below).

In this paper we consider several quadratically constrained quadratic (nonconvex) programs arising from hard combinatorial problems. In particular, we look at the orthogonal relaxations of the quadratic assignment and graph partitioning problems. We show that the resulting well-known eigenvalue bounds for these problems can be obtained from the Lagrangian dual of the orthogonally constrained relaxations,

*Received by the editors June 9, 1998; accepted for publication (in revised form) by P. Van Dooren July 30, 1999; published electronically May 31, 2000.

http://www.siam.org/journals/simax/22-1/34029.html

†Department of Management Sciences, University of Iowa, Iowa City, IA 52242-1000 (kurt-anstreicher@uiowa.edu).

‡University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada (henry@orion.uwaterloo.ca). This author's research was supported by Natural Sciences and Engineering Research Council of Canada.

but only if the seemingly redundant constraint $X^TX = I$ is explicitly added to the orthogonality constraint $XX^T = I$. Our main analytical tool is a strong duality result for a certain nonconvex QQP, where the quadratic constraints correspond to the orthogonality conditions $XX^T = I$, $X^TX = I$. We also show that the technique of applying Lagrangian relaxation to quadratic matrix constraints can be used to obtain a strengthened SDP relaxation for the max-cut problem.

Our results show that current tractable (nonconvex) relaxations for the quadratic assignment and graph partitioning problems can, in fact, be found using Lagrangian relaxations. (A converse statement is well known, i.e., the Lagrangian dual is equivalent to an (tractable) SDP relaxation.) Our results here provide further evidence to the following conjecture: *the Lagrangian relaxation of an appropriate QQP provides the strongest tractable relaxation for QQPs.*

1.1. Outline. We complete this section with the notation used in this paper.

In section 2, we present several known results on QQPs. We start with convex QQPs where a zero duality gap always holds. Then we look at the minimum eigenvalue problem and the trust region subproblem, where strong duality continues to hold. We conclude with the two trust region subproblem, the max-cut problem, and general nonconvex QQPs where nonzero duality gaps can occur.

The main results are in section 3. We show that strong duality holds for a class of orthogonally constrained quadratic programs if we add seemingly redundant constraints before constructing the Lagrangian dual.

In section 4 we apply this result to several problems, i.e., relaxations of quadratic assignment and graph partitioning problems, and a weighted sum of eigenvalue problem. In section 5 we present strengthened semidefinite relaxations for the max-cut problem. In section 6 we summarize our results and describe some promising directions for future research.

1.2. Notation. We now describe the notation used in the paper.

Let $\mathcal{S}_n$ denote the space of $n \times n$ symmetric matrices equipped with the trace inner product, $\langle A, B\rangle = \operatorname{tr} AB$, and let $A \succeq 0$ (resp., $A \succ 0$) denote positive semidefiniteness (resp., positive definiteness) and $A \succeq B$ denote $A - B \succeq 0$, i.e., $\mathcal{S}_n$ is equipped with the Löwner partial order. We let $\mathcal{P}$ denote the cone of symmetric positive semidefinite matrices; $\mathcal{M}_{m,n}$ denotes the space of general $m \times n$ matrices also equipped with the trace inner product, $\langle A, B\rangle = \operatorname{tr} A^TB$, while $\mathcal{M}_m$ denotes the space of general $m \times m$ matrices; $\mathcal{O}$ denotes the set of orthonormal (orthogonal) matrices; Π denotes the set of permutation matrices.

We let $\operatorname{Diag}(v)$ be the diagonal matrix formed from the vector v; its adjoint operator is $\operatorname{diag}(M)$, which is the vector formed from the diagonal of the matrix M. For $M \in \mathcal{M}_{m,n}$, the vector $m = \operatorname{vec}(M) \in \Re^{mn}$ is formed (columnwise) from M.

The Kronecker product of two matrices is denoted $A \otimes B$, and the Hadamard product is denoted $A \circ B$.

We use e to denote the vector of all ones, and $E = ee^T$ to denote the matrix of all ones.

2. Some known results. The general QQP is

$$\begin{aligned} \text{QQP} \qquad & \min q_0(x) \\ & \text{s.t. } q_k(x) \le 0 \ (\text{or } = 0), \quad k = 1, \dots, m, \end{aligned}$$

where $q_i(x) = x^TQ_ix - 2g_i^Tx$. We now present several QQP problems where the Lagrangian relaxation is important and well known. In all these cases, the Lagrangian

dual provides an important theoretical tool for algorithmic development, even where the duality gap may be nonzero.

2.1. Convex quadratic programs. Consider the convex quadratic program

$$\begin{aligned} \text{CQP} \qquad \mu^* := \min\, & q_0(x) \\ \text{s.t. } & q_k(x) \le 0, \quad k = 1, \dots, m, \end{aligned}$$

where all $q_i(x)$ are convex quadratic functions. The dual is

$$\text{DCQP} \qquad \nu^* := \max_{\lambda \ge 0} \ \min_x \ q_0(x) + \sum_{k=1}^{m} \lambda_k q_k(x).$$

If ν^* is attained at λ^*, x^*, then a *sufficient* condition for x^* to be optimal for CQP is primal feasibility and complementary slackness, i.e.,

$$\sum_{k=1}^{m} \lambda_k^* q_k(x^*) = 0.$$

In addition, it is well known that the Karush–Kuhn–Tucker (KKT) conditions are sufficient for global optimality, and under an appropriate constraint qualification the KKT conditions are also necessary. Therefore strong duality holds if a constraint qualification is satisfied, i.e., there is no duality gap and the dual is attained.

However, surprisingly, *if the primal value of CQP is bounded, then it is attained and there is no duality gap*; see, e.g., [44, 36, 34, 35] and, more recently, [26]. However, the dual may not be attained, e.g., consider the convex program

$$0 = \min\{x : x^2 \le 0\}$$

and its dual

$$0 = \max_{\lambda \ge 0} \min_x x + \lambda x^2.$$

Algorithmic approaches based on Lagrangian duality appear in, e.g., [19, 25, 31].

2.2. Rayleigh quotient. Suppose that $A = A^T \in \mathcal{S}_n$. It is well known that the smallest eigenvalue λ_1 of A is obtained from the Rayleigh quotient, i.e.,

$$\lambda_1 = \min\{x^T A x : x^T x = 1\}. \tag{2.1}$$

Since A is not necessarily positive semidefinite, this is the minimization of a nonconvex function on a nonconvex set. However, the Rayleigh quotient forms the basis for many algorithms for finding the smallest eigenvalue, and these algorithms are very efficient. In fact, it is easy to see that there is no duality gap for this nonconvex problem, i.e.,

$$\lambda_1 = \max_{\lambda} \ \min_x \ x^T A x - \lambda(x^T x - 1). \tag{2.2}$$

To see this, note that the inner minimization problem in (2.2) is unconstrained. This implies that the outer maximization problem has the hidden semidefinite constraint (an ongoing theme in the paper)

$$A - \lambda I \succeq 0,$$

i.e., λ is at most the smallest eigenvalue of A. With λ set to the smallest eigenvalue, the inner minimization yields the eigenvector corresponding to λ_1. Thus, we have an example of a *nonconvex problem for which strong duality holds*. Note that the problem (2.1) has the special norm constraint and a homogeneous quadratic objective.

2.3. Trust region subproblem. We will next see that strong duality holds for a larger class of seemingly nonconvex problems. The trust region subproblem (TRS) is the minimization of a quadratic function subject to a norm constraint. No convexity or homogeneity of the objective function is assumed.

$$\begin{array}{ll} \text{TRS} & \mu^* := \min q_0(x) \\ & \qquad \text{s.t. } x^T x - \delta^2 \leq 0 \text{ (or } = 0). \end{array}$$

Assuming that the constraint in TRS is written "$\leq$," the Lagrangian dual is

$$\text{DTRS} \qquad \nu^* := \max_{\lambda \geq 0} \min_x \; q_0(x) + \lambda(x^T x - \delta^2).$$

This is equivalent to (see [43]) the (concave) nonlinear semidefinite program

$$\begin{array}{ll} \text{DTRS} & \nu^* := \max g_0^T (Q + \lambda I)^\dagger g_0 - \lambda \delta^2 \\ & \qquad \text{s.t. } Q + \lambda I \succeq 0, \\ & \qquad \lambda \geq 0. \end{array}$$

where $\cdot^\dagger$ denotes Moore–Penrose inverse. It is shown in [43] that strong duality holds for TRS, i.e., there is a zero duality gap $\mu^* = \nu^*$, and both the primal and dual are attained. Thus, as in the eigenvalue case, we see that this is an example of a nonconvex program where strong duality holds.

Extensions of this result to a two-sided general, possibly nonconvex constraint are discussed in [43, 28]. An algorithm based on Lagrangian duality appears in [40] and (implicitly) in [29, 41]. These algorithms are extremely efficient for the TRS problem, i.e., they solve this problem almost as quickly as they can solve an eigenvalue problem.

2.4. Two trust region subproblem. The two trust region subproblem (TTRS) consists of minimizing a (possibly nonconvex) quadratic function subject to a norm and a least squares constraint, i.e., two convex quadratic constraints. This problem arises in solving general nonlinear programs using a sequential quadratic programming approach and is often called the Celis–Dennis–Tapia (CDT) problem; see [4].

In contrast to the above single TRS, the TTRS can have a nonzero duality gap; see, e.g., [33, 47, 48, 49]. This is closely related to quadratic theorems of the alternative, e.g., [5]. In addition, if the constraints are not convex, then the primal may not be attained; see, e.g., [26].

In [27], Martinez shows that the TRS can have at most one local and nonglobal optimum, and the Lagrangian at this point has one negative eigenvalue. Therefore, if we have such a case and add another ball constraint that contains the local, nonglobal optimum in its interior and also makes this point the global optimum, we obtain a TTRS where we cannot close the duality gap due to the negative eigenvalue. It is uncertain what constraints could be added to close this duality gap. In fact, it is still an open problem whether TTRS is an NP-hard or a polynomial-time problem.

2.5. Max-cut problem. Suppose that $G = (V, E)$ is an undirected graph with vertex set $V = \{v_i\}_{i=1}^n$ and weights w_{ij} on the edges $(v_i, v_j) \in E$. The *max-cut problem* consists of finding the index set $\mathcal{I} \subset \{1, 2, \ldots, n\}$, in order to maximize the weight of the edges with one end point with index in $\mathcal{I}$ and the other in the complement. This is equivalent to the following discrete optimization problem with a quadratic objective:

$$\text{MC} \qquad \max \; \tfrac{1}{2} \textstyle\sum_{i<j} w_{ij}(1 - x_i x_j), \quad x \in \{\pm 1\}^n.$$

We equate $x_i = 1$ with $i \in \mathcal{I}$ and $x_i = -1$ otherwise. Define the homogeneous quadratic objective

$$q(x) := x^T Q x,$$

where Q is an $n \times n$ symmetric matrix. Then the MC problem is equivalent to the QQP

$$\begin{aligned} \mu^*_{MC} := & \max q(x) \\ & \text{s.t. } x_j^2 = 1, \quad j = 1, \ldots, n. \end{aligned}$$

This problem is NP-hard, i.e., intractable.

Since the above QQP has many nonconvex quadratic constraints, a duality gap for the Lagrangian relaxation is expected and does indeed occur most of the time. However, the Lagrangian dual is equivalent to the SDP relaxation (upper bound)

$$\mu^*_{MC} \le \mu^*_{MCSDP} := \begin{array}{ll} \max & \operatorname{tr} QX \\ \text{s.t.} & \operatorname{diag}(X) = e, \\ & X \succeq 0, \end{array} \tag{2.3}$$

which has proven to have very strong theoretical and practical properties, i.e., the bound has an 87% performance guarantee for the problem MC and a 97% performance in practice; see, e.g., [12, 18, 15]. Other theoretical results for general objectives and further relaxed constraints appear in [30, 46].

In [38], several unrelated, though tractable, bounds for MC are shown to be equivalent. These bounds include the box relaxation $-e \le x \le e$, the trust region relaxation $\sum_i x_i^2 = n$, and an eigenvalue relaxation. Furthermore, these bounds are all shown to be equivalent to the Lagrangian relaxation; see [37]. Thus we see that the Lagrangian relaxation is equivalent to the best of these tractable bounds.

2.6. General QQP. The general, possibly nonconvex QQP has many applications in modeling and approximation theory; see, e.g., the applications to SQP methods in [21]. Examples of approximations to QQPs also appear in [9].

The Lagrangian relaxation of a QQP is equivalent to the SDP relaxation and is sometimes referred to as the Shor relaxation; see [42]. The Lagrangian relaxation can be written as an SDP if one takes into the account the hidden semidefinite constraint, i.e., a quadratic function is bounded below only if the Hessian is positive semidefinite. The SDP relaxation is then the Lagrangian dual of this semidefinite program. It can also be obtained directly by *lifting* the problem into matrix space using the fact that $x^T Q x = \operatorname{tr} x^T Q x = \operatorname{tr} Q x x^T$ and relaxing xx^T to a semidefinite matrix X.

One can relate the geometry of the original feasible set of QQP with the feasible set of the SDP relaxation. The connection is through *valid quadratic inequalities*, i.e., nonnegative (convex) combinations of the quadratic functions; see [10, 20].

3. Orthogonally constrained programs with zero duality gaps. Consider the *orthonormal constraint*

$$X^T X = I, \qquad X \in \mathcal{M}_{m,n}.$$

(The set of such X is sometimes known as the Stiefel manifold; see, e.g., [7]. Applications and algorithms for optimization on orthonormal sets of matrices are discussed in [7].) In this section we will show that for $m = n$, strong duality holds for a certain

(nonconvex) quadratic program defined over orthonormal matrices. Because of the similarity of the orthonormality constraint to the norm constraint $x^T x = 1$, the result of this section can be viewed as a matrix generalization of the strong duality result for the Rayleigh quotient problem (2.1).

Let A and B be $n \times n$ symmetric matrices, and consider the orthonormally constrained homogeneous QQP

$$\text{QQP}_\text{O} \qquad \mu^O := \begin{array}{ll} \min & \operatorname{tr} AXBX^T \\ \text{s.t.} & XX^T = I. \end{array} \tag{3.1}$$

This problem can be solved exactly using Lagrange multipliers (see, e.g., [14]) or using the classical Hoffman–Wielandt inequality (see, e.g., [3]). We include a simple proof for completeness.

PROPOSITION 3.1. *Suppose that the orthogonal diagonalizations of A, B are $A = V\Sigma V^T$ and $B = U\Lambda U^T$, respectively, where the eigenvalues in Σ are ordered nonincreasing and the eigenvalues in Λ are ordered nondecreasing. Then the optimal value of* QQP_O *is $\mu^O = \operatorname{tr} \Sigma\Lambda$ and the optimal solution is obtained using the orthogonal matrices that yield the diagonalizations, i.e., $X^* = VU^T$.*

Proof. The constraint $G(X) := XX^T - I$ maps $\mathcal{M}_n$ to $\mathcal{S}_n$. The Jacobian of the constraint at X acting on the direction h is $J(X)(h) = Xh^T + hX^T$. The adjoint of the Jacobian acting on $S \in \mathcal{S}_n$ is $J^*(X)(S) = 2SX$, since

$$\operatorname{tr} SJ(X)(h) = \operatorname{tr} h^T J^*(X)(S).$$

But $J^*(X)(S) = 0$ implies $S = 0$, i.e., J^* is one-one for all X orthogonal. Therefore, J is onto, i.e., the standard constraint qualification holds at the optimum. It follows that the necessary conditions for optimality are that the gradient of the Lagrangian

$$L(X, S) = \operatorname{tr} AXBX^T - \operatorname{tr} S(XX^T - I) \tag{3.2}$$

is 0, i.e.,

$$AXB - SXI = 0.$$

Therefore,

$$AXBX^T = S = S^T,$$

i.e., $AXBX^T$ is symmetric, which means that A and XBX^T commute and so are mutually diagonalizable by the orthogonal matrix U. Therefore, we can assume that both A and B are diagonal and we choose X to be a product of permutations that gives the correct ordering of the eigenvalues. ∎

The Lagrangian dual of QQP_O is

$$\max_{S=S^T} \min_X \operatorname{tr} AXBX^T - \operatorname{tr} S(XX^T - I). \tag{3.3}$$

However, there can be a nonzero duality gap for the Lagrangian dual; see [50] for an example. The inner minimization in the dual problem (3.3) is an unconstrained quadratic minimization in the variables $\operatorname{vec}(X)$, with Hessian

$$B \otimes A - I \otimes S.$$

Clearly this minimization is unbounded if the Hessian is not positive semidefinite. In order to close the duality gap, we need a larger class of quadratic functions.

Note that in $\mathrm{QQP_O}$ the constraints $XX^T = I$ and $X^TX = I$ are equivalent. Adding the redundant constraints $X^TX = I$, we arrive at

$$\begin{aligned} \mathrm{QQP_{OO}} \qquad \mu^O := &\min \operatorname{tr} AXBX^T \\ &\text{s.t. } XX^T = I,\ X^TX = I. \end{aligned}$$

Using symmetric matrices S and T to relax the constraints $XX^T = I$ and $X^TX = I$, respectively, we obtain a dual problem

$$\begin{aligned} \mathrm{DQQP_{OO}} \qquad \mu^O \geq \mu^D := &\max \operatorname{tr} S + \operatorname{tr} T \\ &\text{s.t. } (I \otimes S) + (T \otimes I) \preceq (B \otimes A), \\ &\quad S = S^T,\ T = T^T. \end{aligned}$$

THEOREM 3.2. *Strong duality holds for* $\mathrm{QQP_{OO}}$ *and* $\mathrm{DQQP_{OO}}$, *i.e.,* $\mu^D = \mu^O$ *and both primal and dual are attained.*

Proof. Let $A = V\Sigma V^T, B = U\Lambda U^T$, where V and U are orthonormal matrices whose columns are the eigenvectors of A and B, respectively, σ and λ are the corresponding vectors of eigenvalues, and $\Sigma = \operatorname{Diag}(\sigma)$, $\Lambda = \operatorname{Diag}(\lambda)$. Then for any S and T,

$$(B \otimes A) - (I \otimes S) - (T \otimes I) = (U \otimes V)\left[(\Lambda \otimes \Sigma) - (I \otimes \bar{S}) - (\bar{T} \otimes I)\right](U^T \otimes V^T),$$

where $\bar{S} = V^TSV$, $\bar{T} = U^TTU$. Since $U \otimes V$ is nonsingular, $\operatorname{tr} S = \operatorname{tr} \bar{S}$, and $\operatorname{tr} T = \operatorname{tr} \bar{T}$, the dual problem $\mathrm{DQQP_{OO}}$ is equivalent to

$$\begin{aligned} \mu^D = &\max \operatorname{tr} S + \operatorname{tr} T \\ &\text{s.t. } (\Lambda \otimes \Sigma) - (I \otimes S) - (T \otimes I) \succeq 0, \\ &\quad S = S^T,\ T = T^T. \end{aligned} \tag{3.4}$$

However, since Λ and Σ are diagonal matrices, (3.4) is equivalent to the ordinary linear program:

$$\begin{aligned} \mathrm{LD} \qquad &\max e^Ts + e^Tt \\ &\text{s.t. } \lambda_i\sigma_j - s_j - t_i \geq 0, \quad i,j = 1,\dots,n. \end{aligned}$$

But LD is the dual of the linear assignment problem:

$$\begin{aligned} \mathrm{LP} \qquad &\min \sum_{i,j} \lambda_i\sigma_j x_{ij} \\ &\text{s.t. } \sum_{j=1}^n x_{ij} = 1, \quad i = 1,\dots,n, \\ &\quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1,\dots,n, \\ &\quad x_{ij} \geq 0, \quad i,j = 1,\dots,n. \end{aligned}$$

Assume without loss of generality that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Then LP can be interpreted as the problem of finding a permutation $\pi(\cdot)$ of $\{1,\dots,n\}$ so that $\sum_{i=1}^n \lambda_i\sigma_{\pi(i)}$ is minimized. But the minimizing permutation is then $\pi(i) = i$, $i = 1,\dots,n$, and from Proposition 3.1 the solution value μ^D is exactly μ^O. ∎

4. Applications. We now present three applications of the above strong duality result.

4.1. Quadratic assignment problem. Let A and B be $n \times n$ symmetric matrices, and consider the homogeneous quadratic assignment problem (QAP) (see, e.g., [32]),

$$\begin{array}{ll} \text{QAP} & \min \operatorname{tr} AXBX^T \\ & \text{s.t. } X \in \Pi, \end{array}$$

where Π is the set of $n \times n$ permutation matrices. The set of orthonormal matrices contains the permutation matrices, and the orthonormally constrained problem (3.1) is an important relaxation of QAP. The bounds obtained are usually called the eigenvalue bounds for QAP; see [8, 13]. Theorem 3.2 shows that the eigenvalue bounds are in fact obtained from a Lagrangian relaxation of (3.1) after adding the seemingly redundant constraint $XX^T = I$.

4.2. Weighted sums of eigenvalues. Consider the problem of minimizing the weighted sum of the k largest eigenvalues of an $n \times n$ symmetric matrix Y, subject to linear equality constraints. An SDP formulation for this problem involving $2k$ semidefiniteness constraints on $n \times n$ matrices is given in [1, section 4.3]. We will show that the result of section 3 can be applied to obtain a new SDP formulation of the problem having only $k+1$ semidefiniteness constraints on $n \times n$ matrices.

For convenience we consider the equivalent problem of maximizing the weighted sum of the k minimum eigenvalues of Y. Let $w_1 \geq w_2 \geq \cdots \geq w_k > w_{k+1} = w_{k+2} = \cdots = w_n = 0$, and let $W = \operatorname{Diag}(w)$. We are interested in the problem

$$\begin{array}{ll} \text{WEIG} & \max \displaystyle\sum_{i=1}^{k} w_i \lambda_i(Y) \\ & \text{s.t. } \mathcal{A} \operatorname{vec}(Y) = b, \\ & \quad Y = Y^T, \end{array}$$

where $\lambda_1(Y) \leq \lambda_2(Y) \leq \cdots \leq \lambda_n(Y)$ are the eigenvalues of Y, and $\mathcal{A}$ is a $p \times n^2$ matrix. From Proposition 3.1 it is clear that, for any Y,

$$\sum_{i=1}^{k} w_i \lambda_i(Y) = \min_{X^T X = I} \operatorname{tr} YXWX^T,$$

and therefore from Theorem 3.2 the problem WEIG is equivalent to the problem

$$\begin{array}{ll} \max & \operatorname{tr} S + \operatorname{tr} T \\ \text{s.t.} & (W \otimes Y) - (I \otimes S) - (T \otimes I) \succeq 0, \\ & \mathcal{A} \operatorname{vec}(Y) = b, \\ & S = S^T,\ T = T^T,\ Y = Y^T. \end{array} \tag{4.1}$$

Note that, for any Y, the matrix $W \otimes Y$ is block diagonal, with the final $n-k$ blocks identically zero. Since $I \otimes S$ is also block diagonal, and $\operatorname{tr} T$ is a function of the diagonal of T only, it is obvious that T can be assumed to be a diagonal matrix $T = \operatorname{Diag}(t)$. Writing the problem (4.1) in terms of t, and separating the block diagonal constraints,

results in the SDP

$$\begin{aligned}
&\max \operatorname{tr} S + \sum_{i=1}^{k} t_i + (n-k)t_{k+1} \\
&\text{s.t. } w_i Y - S - t_i I \succeq 0, \quad i = 1 \ldots, k, \\
&\qquad -S - t_{k+1} I \succeq 0, \\
&\qquad \mathcal{A} \operatorname{vec} Y = b, \\
&\qquad S = S^T.
\end{aligned}$$

We have thus obtained an SDP representation for the problem WEIG with $k+1$ semidefiniteness constraints on $n \times n$ matrices, as claimed.

4.3. Graph partitioning problem. Let $G = (N, E)$ be an edge-weighted undirected graph with node set $N = \{1, \ldots, n\}$, edge set E, and weights $w_{ij},\ ij \in E$. The graph partitioning (GP) problem consists of partitioning the node set N into k disjoint subsets $S_1, \ldots, S_k$ of specified sizes $m_1 \geq m_2 \geq \cdots \geq m_k,\ \sum_{j=1}^{k} m_j = n$, so as to minimize the total weight of the edges connecting nodes in distinct subsets of the partition. This problem is well known to be NP-hard. GP can be modeled as a quadratic problem

$$\begin{aligned}
z := &\min \operatorname{tr} X^T L X \\
&\text{s.t. } X \in P,
\end{aligned}$$

where L is the Laplacian of the graph and P is the set of $n \times k$ partition matrices (i.e., each column of X is the indicator function of the corresponding set; $X_{ij} = 1$ if node i is in set j and 0 otherwise).

The well-known *Donath–Hoffman* bound [6] $z_{DH} \leq z$ for GP is

$$z_{DH} := \max_{e^T u = 0} \sum_{i=1}^{k} m_i \lambda_i (L + U),$$

where $U = \operatorname{Diag}(u)$, and $\lambda_1(L+U) \leq \lambda_2(L+U) \leq \cdots \leq \lambda_n(L+U)$ are the eigenvalues of $L + U$. We will now show that the Donath–Hoffman bound can be obtained by applying Lagrangian relaxation to an appropriate QQP relaxation of GP. (An SDP formulation for this bound is given in [1].) Clearly, if P is a partition matrix, then $x_i^T x_i = 1$, $i = 1, \ldots, n$, where x_i^T is the ith row of X. Moreover, the columns of X are orthogonal with one another, and the norm of the jth column of X is $\sqrt{m_j}$. It follows that if X is a partition matrix, there is an $n \times n$ orthogonal matrix $\bar{X}$ such that

$$X = \bar{X} \begin{pmatrix} M^{1/2} \\ 0 \end{pmatrix},$$

where M is the $k \times k$ matrix $M = \operatorname{Diag}(m)$, and therefore

$$X X^T = \bar{X} \begin{pmatrix} M^{1/2} \\ 0 \end{pmatrix} (M^{1/2}, 0) \bar{X}^T = \bar{X} \bar{M} \bar{X}^T, \quad \text{where} \quad \bar{M} = \begin{pmatrix} M & 0 \\ 0 & 0 \end{pmatrix}.$$

In addition, note that $x_i^T x_i$ is the ith diagonal element of $X X^T$, so the constraint $x_i^T x_i = 1$ is equivalent to $\bar{x}_i^T \bar{M} \bar{x}_i = 1$, where $\bar{x}_i^T$ is the ith row of $\bar{X}$. Since $\operatorname{tr} X^T L X =$

$\operatorname{tr} LXX^T$, a lower bound $z_1 \leq z$ can be defined by

$$\begin{aligned} z_1 := \min \operatorname{tr} L\bar{X}\bar{M}\bar{X}^T & \\ \text{s.t. } \bar{X}^T\bar{X} = I,\ \bar{X}\bar{X}^T = I, & \\ \bar{x}_i^T\bar{M}\bar{x}_i = 1, i = 1, \ldots, n. & \end{aligned} \tag{4.2}$$

We will now obtain a second bound $z_2 \leq z_1$ by applying a Lagrangian procedure to all of the constraints in (4.2). Using symmetric matrices S and T for the constraints $\bar{X}\bar{X}^T = I$ and $\bar{X}^T\bar{X} = I$, respectively, and a vector of multipliers u_i for the constraints $\bar{x}_i^T\bar{M}\bar{x}_i = 1$, $i = 1, \ldots, n$, we obtain

$$\begin{aligned} z_2 := \max_{u,S,T} \min_{\bar{X}} \ & \operatorname{tr} L\bar{X}\bar{M}\bar{X}^T + \operatorname{tr} S(I - \bar{X}\bar{X}^T) + \operatorname{tr} T(I - \bar{X}^T\bar{X}) \\ & + \sum_{i=1}^n u_i(\bar{x}_i^T\bar{M}\bar{x}_i - 1). \end{aligned}$$

THEOREM 4.1. $z_2 = z_{DH}$.

Proof. Rearranging terms and using Kronecker product notation, the definition of z_2 can be rewritten as

$$\begin{aligned} z_2 = \max_{u,S,T} \ & \operatorname{tr} S + \operatorname{tr} T - e^T u \\ & + \min_{\bar{X}} \ \operatorname{vec}(\bar{X})^T([\bar{M} \otimes (L + U)] - (I \otimes S) - (T \otimes I))\operatorname{vec}(\bar{X}), \end{aligned}$$

where $U = \operatorname{Diag}(u)$, and we are using the fact that

$$\sum_{i=1}^n u_i\bar{x}_i^T\bar{M}\bar{x}_i = \operatorname{tr} U\bar{X}\bar{M}\bar{X}^T.$$

Clearly if $[\bar{M} \otimes (L + U)] - (I \otimes S) - (T \otimes I) \succeq 0$, then $\bar{X} = 0$ solves the implicit minimization problem in the definition of z_2, and if this constraint fails to hold, the minimum is $-\infty$. Using this hidden semidefinite constraint, we can write

$$\begin{aligned} z_2 = \max & \operatorname{tr} S + \operatorname{tr} T - e^T u \\ \text{s.t. } & [\bar{M} \otimes (L + U)] - (I \otimes S) - (T \otimes I) \succeq 0, \\ & S = S^T,\ T = T^T. \end{aligned}$$

Note that if $u' = u + \lambda e$ and $T' = T + \lambda\bar{M}$ for any scalar λ, then

$$\begin{aligned} \bar{M} \otimes (L + U') &= \bar{M} \otimes (L + U) + \lambda(\bar{M} \otimes I), \\ T' \otimes I &= T \otimes I + \lambda(\bar{M} \otimes I). \end{aligned}$$

In addition, $\operatorname{tr} T' = \operatorname{tr} T + \lambda n$ and $e^T u' = e^T u + \lambda n$. It follows that we may choose any normalization for $e^T u$ without affecting the value of z_2. Choosing $e^T u = 0$, we arrive at

$$\begin{aligned} z_2 = \max & \operatorname{tr} S + \operatorname{tr} T \\ \text{s.t. } & [\bar{M} \otimes (L + U)] - (I \otimes S) - (T \otimes I) \succeq 0, \\ & e^T u = 0,\ S = S^T,\ T = T^T. \end{aligned}$$

However, as in the previous section, Proposition 3.1 and Theorem 3.2 together imply that for any U, the solution value in the problem

$$\begin{aligned} &\max \operatorname{tr} S + \operatorname{tr} T \\ &\text{s.t. } [\bar{M} \otimes (L+U)] - (I \otimes S) - (T \otimes I) \succeq 0, \\ &\quad S = S^T,\ T = T^T, \end{aligned}$$

is exactly $\sum_{i=1}^{k} m_i \lambda_i(L+U)$. Therefore, we immediately have $z_{DH} = z_2$. □

SDP relaxations for the GP problem are obtained via Lagrangian relaxation in [45]. A useful corollary of Theorem 4.1 is that any Lagrangian relaxation based on a more tightly constrained problem than (4.2) will produce bounds that dominate the Donath–Hoffman bounds.

A problem closely related to the orthogonal relaxation of GP is the orthogonal Procrustes problem on the Stiefel manifold; see [7, section 3.5.2]. This problem has a linear term in the objective function, and there is no known analytic solution for the general case.

5. A strengthened relaxation for max-cut. As discussed above, the SDP relaxation for MC performs very well in practice and has strong theoretical properties. There have been attempts at further strengthening this relaxation. For example, a copositive relaxation is presented in [39]. Adding cuts to the SDP relaxation is discussed in [15, 16, 17, 18]. These improvements all involve heuristics, such as deciding which cuts to choose or solving a copositive problem, which is NP-hard in itself.

The relaxation in (2.3) is obtained by *lifting* the vector x into matrix space using $X = xx^T$. Though the matrix X in the lifting is not an orthogonal matrix, it is a *partial isometry* up to normalization, i.e.,

$$X^2 - nX = 0. \tag{5.1}$$

We will now show that we can improve the semidefinite relaxation presented in section 2.5 by considering Lagrangian relaxations using the matrix quadratic constraint (5.1). In particular, consider the relaxation of MC

$$\begin{aligned} \mu_1 := \max \quad & \operatorname{tr} \quad QX \\ \text{s.t.} \quad & \operatorname{diag}(X) = e, \\ & X^2 \quad -nX = 0, \end{aligned}$$

where X is a symmetric matrix. Note that if $X^2 = nX$, then $\operatorname{tr} QX = (1/n) \operatorname{tr} QX^2$, and $\operatorname{diag}(X^2) = ne$. As a result, the above relaxation is equivalent to the relaxation

$$\begin{aligned} \mu_1 = &\max \frac{1}{n} \operatorname{tr} QX^2 \\ &\text{s.t. } x_i^T x_i = n, \quad i = 1, \ldots, n, \\ &\quad X^2 - nx_0 X = 0, \\ &\quad x_0^2 = 1, \end{aligned} \tag{5.2}$$

where x_i^T, $i = 1, \ldots, n$, denotes the ith row of X, and x_0 is a scalar. (Note that if $x_0 = -1$, then changing x_0 to 1 and replacing X with $-X$ leaves the objective and constraints in (5.2) unchanged.) We will obtain an upper bound $\mu_2 \geq \mu_1$ by applying a Lagrangian procedure to all of the constraints in (5.2). Using multipliers u_i for the

constraints $x_i^T x_i = n$, $i = 1, \ldots, n$, u_0 for the constraint $x_0^2 = 1$, and a symmetric matrix S for the matrix equality $X^2 - nX = 0$, we obtain a Lagrangian problem

$$\mu_2 := \min_{u_0,u,S} u_0 + nu^T e + \max_{x_0,X} \frac{1}{n} \operatorname{tr} QX^2 - \operatorname{tr} UX^2 + \operatorname{tr} SX^2 - nx_0 \operatorname{tr} SX - u_0 x_0^2,$$

where $U = \operatorname{Diag}(u)$. Letting $\bar{x}^T = (x_0, \operatorname{vec}(X)^T)$, this problem can be written in Kronecker product form as

$$\mu_2 = \min_{u_0,u,S} \; u_0 + ne^T u + \max_{\bar{x}} \; \bar{x}^T \bar{Q} \bar{x},$$

where

$$\bar{Q} = \begin{pmatrix} -u_0 & -\frac{n}{2}\operatorname{vec}(S)^T \\ -\frac{n}{2}\operatorname{vec}(S) & I \otimes \left(\frac{1}{n}Q - U + S\right) \end{pmatrix}.$$

Applying the hidden semidefinite constraint $\bar{Q} \preceq 0$, we obtain an equivalent problem,

$$\begin{aligned} \mu_2 = & \min u_0 + ne^T u \\ & \text{s.t.} \begin{pmatrix} u_0 & \frac{n}{2}\operatorname{vec}(S)^T \\ \frac{n}{2}\operatorname{vec}(S) & I \otimes \left(-\frac{1}{n}Q + U - S\right) \end{pmatrix} \succeq 0, \\ & S = S^T. \end{aligned} \tag{5.3}$$

Note that if we take $S = 0$ in (5.3), then $u_0 = 0$ is clearly optimal and the problem reduces to

$$\begin{aligned} & \min e^T u \\ & \text{s.t.} \; -Q + U \succeq 0, \end{aligned}$$

which is exactly the dual of (2.3), the usual SDP relaxation for MC. It follows that we have obtained an upper bound μ_2 which is a strengthening of the usual SDP bound, i.e., $\mu_2 \leq \mu^*_{MCSDP}$.

The strengthened relaxation (5.3) involves a semidefiniteness constraint on a $(n^2+1) \times (n^2+1)$ matrix, as opposed to an $n \times n$ matrix in the usual SDP relaxation (2.3). This dimensional increase can be mitigated by taking note of the fact that X in (5.2) must be a symmetric matrix, and therefore (5.2) can actually be written as a problem over a vector x of dimension $n(n+1)/2$. In addition, alternative relaxations can be obtained by not making the substitutions based on (5.1) used to obtain the problem (5.2). The effect of these alternatives on the performance of strengthened SDP bounds for MC is the topic of ongoing research; for up-to-date developments, see the URL http://orion.uwaterloo.ca/~hwolkowi/henry/reports/strngthMC.ps.gz.

6. Conclusion. In this paper we have shown that a class of nonconvex quadratic problems with orthogonal constraints can satisfy strong duality if certain seemingly redundant constraints are added before the Lagrangian dual is formed. As applications of this result we showed that well-known eigenvalue bounds for QAP and GP problems can actually be obtained from the Lagrangian dual of QQP relaxations of these problems. We also showed that the technique of relaxing quadratic matrix constraints can be used to obtain strengthened SDP relaxations for the max-cut problem.

Adding constraints to close the duality gap is akin to adding valid inequalities in cutting plane methods for discrete optimization problems. In [2, 24] this approach, in

combination with a lifting procedure, is used to solve discrete optimization problems. In our case we add quadratic constraints. The idea of quadratic valid inequalities has been used in [10]; and closing the duality gap has been discussed in [20].

Our success in closing the duality gap for the $\mathrm{QQP_O}$ problem considered in section 3, where we have the special Kronecker product in the objective function, raises several interesting questions. For example, can the strong duality result for $\mathrm{QQP_O}$ be extended to the same problem with an added linear term in the objective, or are there some other special classes of objective functions where this is possible? Another outstanding question is whether it is possible to add quadratic constraints to close the duality gap for the TTRS.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] E. Balas, S. Ceria, and G. Cornuejols, *A lift-and-project cutting plane algorithm for mixed* 0-1 *programs*, Math. Programming, 58 (1993), pp. 295–324.

[3] R. Bhatia, *Perturbation Bounds for Matrix Eigenvalues*, Pitman Res. Notes Math. Ser. 162, Longman Scientific and Technical, Harlow, UK, 1987.

[4] M. Celis, J. Dennis Jr., and R. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Proceedings of the SIAM Conference on Numerical Optimization, Boulder, CO, 1984 , pp. 71–82. Also available as Technical Report TR84-1, Rice University, Houston, TX.

[5] J.-P. Crouzeix, J.-E. Martinez-Legaz, and A. Seeger, *An alternative theorem for quadratic forms and extensions*, Linear Algebra Appl., 215 (1995), pp. 121–134.

[6] W. Donath and A. Hoffman, *Lower bounds for the partitioning of graphs*, IBM J. Res. Develop., 17 (1973), pp. 420–425.

[7] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[8] G. Finke, R. Burkhard, and F. Rendl, *Quadratic assignment problems*, Ann. Discrete Math., 31 (1987), pp. 61–82.

[9] M. Fu, Z. Luo, and Y. Ye, *Approximation algorithms for quadratic programming*, J. Combinatorial Optim., 2 (1998), pp. 29–50.

[10] T. Fujie and M. Kojima, *Semidefinite programming relaxation for nonconvex quadratic programs*, J. Global Optim., 10 (1997), pp. 367–380.

[11] M. X. Goemans, *Semidefinite programming in combinatorial optimization*, Math. Programming, 79 (1997), pp. 143–162.

[12] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42 (1995), pp. 1115–1145.

[13] S. Hadley, F. Rendl, and H. Wolkowicz, *Bounds for the quadratic assignment problems using continuous optimization*, in Integer Programming and Combinatorial Optimization, University of Waterloo Press, Waterloo, Ontario, Canada, 1990, pp. 237–248.

[14] S. Hadley, F. Rendl, and H. Wolkowicz, *A new lower bound via projection for the quadratic assignment problem*, Math. Oper. Res., 17 (1992), pp. 727–739.

[15] C. Helmberg, *An Interior Point Method for Semidefinite Programming and Max-Cut Bounds*, Ph.D. thesis, Graz University of Technology, Austria, 1994.

[16] C. Helmberg, *Fixing Variables in Semidefinite Relaxations*, Lecture Notes in Comput. Sci. 1284, Springer, Berlin, 1997.

[17] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. Optim, 10 (2000), pp. 673–696.

[18] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[19] F. Jarre, *On the convergence of the method of analytic centers when applied to convex quadratic programs*, Math. Programming, 49 (1990), pp. 341–358.

[20] M. Kojima and L. Tunçel, *Cones of Matrices and Successive Convex Relaxations of Nonconvex Sets*, Technical Report B-338, Tokyo Institute of Technology, Tokyo, Japan, 1998.

[21] S. Kruk and H. Wolkowicz, *SQ^2P, sequential quadratic constrained quadratic programming*, in Advances in Nonlinear Programming, Y. Xiang Yuan, ed., Appl. Optim. 14, Kluwer Academic Publishers, Dordrecht, 1998, pp. 177–204.

[22] M. Laurent, *Cuts, matrix completions and graph rigidity*, Math. Programming, 79 (1997), pp. 255–284.

[23] M. Laurent, *A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems*, in Topics in Semidefinite and Interior-Point Methods, Fields Inst. Commun. 18, AMS, Providence, RI, 1998, pp. 51–76.

[24] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[25] Z.-Q. Luo and J. Sun, *An Analytic Center Based Column Generation Algorithm for Convex Quadratic Feasibility Problems*, Technical report, McMaster University, Hamilton, Ontario, Canada, 1995; also available from ftp://ftp.nus.sg/pub/NUS/opt/qcut.ps.gz.

[26] Z.-Q. Luo and S. Zhang, *On the Extension of Frank-Wolfe Theorem*, Technical report, Erasmus University Rotterdam, The Netherlands, 1997.

[27] J. M. Martinez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176.

[28] J. J. Moré, *Generalizations of the trust region problem*, Optim. Methods Softw., 2 (1993), pp. 189–209.

[29] J. Moré and D. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.

[30] Y. E. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optim. Methods Softw., 9 (1998), pp. 141–160.

[31] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, PA, 1994.

[32] P. Pardalos, F. Rendl, and H. Wolkowicz, *The quadratic assignment problem: A survey and recent developments*, in Proceedings of the DIMACS Workshop on Quadratic Assignment Problems, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 16, AMS, Providence, RI, 1994, pp. 1–41.

[33] J.-M. Peng and Y.-X. Yuan, *Optimality conditions for the minimization of a quadratic with two quadratic constraints*, SIAM J. Optim., 7 (1997), pp. 579–594.

[34] E. L. Peterson and J. G. Ecker, *Geometric programming: Duality in quadratic programming and l_p-approximation.* I, in Proceedings of the Princeton Symposium on Mathematical Programming, Princeton Univ., 1967, Princeton University Press, Princeton, NJ, 1970, pp. 445–480.

[35] E. L. Peterson and J. G. Ecker, *Geometric programming: Duality in quadratic programming and l_p-approximation.* II. *Canonical programs*, SIAM J. Appl. Math., 17 (1969), pp. 317–340.

[36] E. L. Peterson and J. G. Ecker, *Geometric programming: Duality in quadratic programming and l_p-approximation.* III. *Degenerate programs*, J. Math. Anal. Appl., 29 (1970), pp. 365–383.

[37] S. Poljak, F. Rendl, and H. Wolkowicz, *A recipe for semidefinite relaxation for* (0,1)*-quadratic programming*, J. Global Optim., 7 (1995), pp. 51–73.

[38] S. Poljak and H. Wolkowicz, *Convex relaxations of* 0-1 *quadratic programming*, Math. Oper. Res., 20 (1995), pp. 550–561.

[39] A. Quist, E. D. Klerk, C. Roos, and T. Terlaky, *Copositive relaxation for general quadratic programming*, Optim. Methods Softw., 9 (1998), pp. 185–208.

[40] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, 77 (1997), pp. 273–299.

[41] S. Santos and D. Sorensen, *A New Matrix-Free Algorithm for the Large-Scale Trust-Region Subproblem*, Technical Report TR95-20, Rice University, Houston, TX, 1995.

[42] N. Shor, *Quadratic optimization problems*, Izv. Akad. Nauk SSSR Tekhn. Kibernet., 222 (1987), pp. 128–139.

[43] R. J. Stern and H. Wolkowicz, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations*, SIAM J. Optim., 5 (1995), pp. 286–313.

[44] T. Terlaky, *On l_p programming*, European J. Oper. Res., 22 (1985), pp. 70–100.

[45] H. Wolkowicz and Q. Zhao, *Semidefinite relaxations for the graph partitioning problem*, Discrete Appl. Math., 96–97 (1999), pp. 461–479.

[46] Y. Ye, *Approximating quadratic programming with bound and quadratic constraints*, Math. Programming, 84 (1999), pp. 219–226.

[47] Y. Yuan, *Some Properties of Trust Region Algorithms for Nonsmooth Optimization*, Technical Report DAMTP 1983/NA14, University of Cambridge, Cambridge, UK, 1983.

[48] Y. Yuan, *On a subproblem of trust region algorithms for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.
[49] Y. Yuan, *A dual algorithm for minimizing a quadratic function with two quadratic constraints*, J. Comput. Math., 9 (1991), pp. 348–359.
[50] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, J. Combin. Optim., 2 (1998), pp. 71–109.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 56–65

CONDITIONING OF THE STABLE, DISCRETE-TIME LYAPUNOV OPERATOR*

MICHAEL K. TIPPETT†, STEPHEN E. COHN‡, RICARDO TODLING§, AND DAN MARCHESIN¶

Abstract. The Schatten p-norm condition of the discrete-time Lyapunov operator $\mathcal{L}_A$ defined on matrices $P \in \mathbb{R}^{n\times n}$ by $\mathcal{L}_A P \equiv P - APA^T$ is studied for stable matrices $A \in \mathbb{R}^{n\times n}$. Bounds are obtained for the norm of $\mathcal{L}_A$ and its inverse that depend on the spectrum, singular values, and radius of stability of A. Since the solution P of the discrete-time algebraic Lyapunov equation (DALE) $\mathcal{L}_A P = Q$ can be ill-conditioned only when either $\mathcal{L}_A$ or Q is ill-conditioned, these bounds are useful in determining whether P admits a low-rank approximation, which is important in the numerical solution of the DALE for large n.

Key words. Lyapunov matrix equation, condition estimates, large-scale systems, radius of stability

AMS subject classifications. 15A12, 93C55, 93A15, 47B65

PII. S0895479899354822

1. Introduction.

Properties of the solution P of the discrete algebraic Lyapunov equation (DALE), $P = APA^T + Q$, are closely related to the stability properties of A. For instance, the DALE has a unique solution $P = P^T > 0$ for any $Q = Q^T > 0$ if A is stable [11], a fact also true in infinite-dimensional Hilbert spaces [18]. In the setting treated here with A, Q, $P \in \mathbb{R}^{n\times n}$, A is stable if its eigenvalues $\lambda_i(A)$, $i = 1, \ldots, n$, lie inside the unit circle; the eigenvalues are ordered so that $|\lambda_1(A)| \geq |\lambda_2(A)| \geq \cdots \geq |\lambda_n(A)|$. Here A is always assumed to be stable.

In applications where the dimension n is very large, direct solution of the DALE or even storage of P is impractical or impossible. For instance, in numerical weather prediction applications A is the matrix that evolves atmospheric state perturbations. The DALE and its continuous-time analogues can be solved directly for simplified atmospheric models [6, 23], but in realistic models n is about $10^6 - 10^7$ and even the storage of P is impossible. Krylov subspace [5] and Monte Carlo [9] methods have been used to find low-rank approximations of the right-hand side of the DALE and of the solution of the DALE [10].

The solution P of the DALE can be well approximated by a rank-deficient matrix if P has some small singular values. Therefore, it is useful to identify properties of A

*Received by the editors April 23, 1999; accepted for publication (in revised form) by R. Bhatia November 1, 1999; published electronically May 31, 2000. This work was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico grants 91.0029/95-4, 381737/97-7, and 30.0204/83-3, Financiadora de Estudos e Projetos grant 77.97.0315.00, and the NASA EOS Interdisciplinary Project on Data Assimilation.

http://www.siam.org/journals/simax/22-1/35482.html

†IRI, Lamont–Doherty Earth Observatory of Columbia University, Palisades, NY 10964-8000 (tippett@iri.ldeo.columbia.edu). This work was done while the author was with the Centro de Previsão de Tempo e Estudos Climáticos, Cachoeira Paulista, SP, Brazil.

‡Data Assimilation Office, Code 910.3, NASA/GSFC, Greenbelt, MD 20771 (cohn@dao.gsfc.nasa.gov).

§General Sciences Corp./SAIC, Code 910.3, NASA/GSFC/DAO, Greenbelt, MD 20771 (todling@dao.gsfc.nasa.gov).

¶Instituto de Matemática Pura e Aplicada, Rio de Janeiro, RJ, Brazil (marchesi@impa.br).

or Q that lead to P being ill-conditioned. If A is normal, then

$$\frac{\lambda_1(P)}{\lambda_n(P)} \leq \frac{\lambda_1(Q)}{\lambda_n(Q)} \frac{1-|\lambda_n(A)|^2}{1-|\lambda_1(A)|^2}\,; \tag{1.1}$$

the conditioning of P is controlled by that of Q and by the spectrum of A. In the general case, the conditioning of Q and of the discrete-time Lyapunov operator $\mathcal{L}_A$ defined by $\mathcal{L}_A P \equiv P - APA^T$ determine when P may be ill-conditioned.

THEOREM 1.1. *Let A be a stable matrix and suppose that $\mathcal{L}_A P = Q$ for $Q = Q^T > 0$. Then*

$$\|P\|_p \, \|P^{-1}\|_p \leq \|\mathcal{L}_A\|_p \, \|\mathcal{L}_A^{-1}\|_p \, \|Q\|_p \, \|Q^{-1}\|_p\,, \qquad p = \infty\,, \tag{1.2}$$

where $\|\cdot\|_p$ is the Schatten p-norm (see (2.2)).

Theorem 1.1 (see proof in the appendix) follows from $\mathcal{L}_A^{-1}$ and its adjoint being positive operators. Therefore, the same connection between rank-deficient approximate solutions and operator conditioning exists for matrix equations such as the continuous algebraic Lyapunov equation. We note that Theorem 1.1 also holds for $1 \leq p < \infty$ if either A is singular or $\sigma_1^2(A) \geq 2$; $\sigma_1(A)$ is the largest singular value of A.

Here we characterize the Schatten p-norm condition of $\mathcal{L}_A$. The main results are the following. Theorem 3.1 bounds $\|\mathcal{L}_A\|_p$ in terms of the singular values of A. A lower bound for $\|\mathcal{L}_A^{-1}\|_p$ depending on $\lambda_1(A)$ is presented in Theorem 4.1, generalizing results of [7]. Theorem 4.2 gives lower bounds for $\|\mathcal{L}_A^{-1}\|_1$ and $\|\mathcal{L}_A^{-1}\|_\infty$ in terms of the singular values of A. Theorem 4.6 gives an upper bound for $\|\mathcal{L}_A^{-1}\|_p$ depending on the radius of stability of A and generalizes results in [20]. Three examples illustrating the results are included. The issue of whether $\mathcal{L}_A$ and $\mathcal{L}_A^{-1}$ achieve their norms on symmetric, positive definite matrices is addressed in the concluding remarks.

2. Preliminaries. We investigate the condition number $\kappa(\mathcal{L}_A) = \|\mathcal{L}_A\| \, \|\mathcal{L}_A^{-1}\|$, where $\|\cdot\|$ is a norm on $\mathbb{R}^{n^2 \times n^2}$ induced by a matrix norm on $\mathbb{R}^{n \times n}$. Specifically, for $\mathcal{M} \in \mathbb{R}^{n^2 \times n^2}$ we consider norms defined by

$$\|\mathcal{M}\|_p = \max_{S \neq 0 \in \mathbb{R}^{n \times n}} \frac{\|\mathcal{M}S\|_p}{\|S\|_p}\,, \quad 1 \leq p \leq \infty\,, \tag{2.1}$$

where the Schatten matrix p-norm for $S \in \mathbb{R}^{n \times n}$ is defined by

$$\|S\|_p = \left(\sum_{i=1}^{n} (\sigma_i(S))^p \right)^{1/p}\,; \tag{2.2}$$

$\sigma_i(S)$ are the singular values of S with ordering $\sigma_1(S) \geq \sigma_2(S) \geq \cdots \geq \sigma_n(S) \geq 0$. On $\mathbb{R}^{n \times n}$, $\|\cdot\|_2$ is the Frobenius norm and $\|\cdot\|_\infty = \sigma_1(\cdot)$. If $S = S^T \geq 0$, then $\|S\|_1 = \operatorname{tr} S$. The following lemma about the Schatten p-norms follows from their being unitarily invariant [1, p. 94].

LEMMA 2.1. *For any three matrices X, Y, and $Z \in \mathbb{R}^{n \times n}$,*

$$\|XYZ\|_p \leq \|X\|_\infty \|Y\|_p \|Z\|_\infty\,, \qquad 1 \leq p \leq \infty\,. \tag{2.3}$$

The $p = 2$ Schatten norm on $\mathbb{R}^{n \times n}$ is equivalently defined as $\|S\|_2^2 = (S, S)$, where $(\cdot, \cdot)$ is the inner product on $\mathbb{R}^{n \times n}$ defined by $(S_1, S_2) = \operatorname{tr} S_1^T S_2$. This norm

corresponds to the usual Euclidean norm on $\mathbb{R}^{n^2}$ since $\|S\|_2^2$ is equal to the sum of the squares of the entries of S. As a consequence $\kappa_2(\mathcal{L}_A) = \sigma_1(\mathcal{L}_A)/\sigma_{n^2}(\mathcal{L}_A)$, where $\sigma_1(\mathcal{L}_A)$ and $\sigma_{n^2}(\mathcal{L}_A)$ are, respectively, the largest and smallest singular values of $\mathcal{L}_A$. The adjoint of $\mathcal{L}_A$ is given by $\mathcal{L}_A^* S = \mathcal{L}_{A^T} S = S - A^T S A$.

We now state some lemmas about mappings $\mathcal{M} \in \mathbb{R}^{n^2 \times n^2}$ and about the spectra of $\mathcal{L}_A$ and A.

LEMMA 2.2 (see [2, equation (15)]). $\|\mathcal{M}\|_p \le \|\mathcal{M}\|_1^{1/p} \|\mathcal{M}\|_\infty^{1-1/p}, \quad 1 \le p \le \infty.$

LEMMA 2.3. $\|\mathcal{M}\|_1 = \|\mathcal{M}^*\|_\infty$.

LEMMA 2.4 (see [2, proof of Theorem 1]). *If $\mathcal{M}S > 0$ for all $S \in \mathbb{R}^{n\times n}$ such that $S > 0$, then $\|\mathcal{M}\|_\infty = \|\mathcal{M}I\|_\infty$.*

LEMMA 2.5 (see [13, 14]). *The n^2 eigenvalues of $\mathcal{L}_A$ are $1 - \lambda_i(A)\overline{\lambda_j(A)}$, $1 \le i, j \le n$.*

3. The norm of the Lyapunov operator. If A is normal, then $\mathcal{L}_A$ is normal, and its conditioning in the $p = 2$ Schatten norm depends only on its eigenvalues. Therefore, when A is normal,

$$\|\mathcal{L}_A^{-1}\|_2 = \frac{1}{\sigma_{n^2}(\mathcal{L}_A)} = \frac{1}{|\lambda_{n^2}(\mathcal{L}_A)|} = \frac{1}{1 - |\lambda_1(A)|^2} \tag{3.1}$$

and

$$\|\mathcal{L}_A\|_2 = \sigma_1(\mathcal{L}_A) = |\lambda_1(\mathcal{L}_A)| = \max_{i,j} |1 - \lambda_i(A)\overline{\lambda_j(A)}|\,. \tag{3.2}$$

For general A, the following theorem bounds $\|\mathcal{L}_A\|_p$ in terms of the singular values of A.

THEOREM 3.1.

$$|1 - \sigma_1^2(A)| \le \max_j |1 - \sigma_j^2(A)| \le \|\mathcal{L}_A\|_p \le 1 + \sigma_1^2(A)\,, \quad 1 \le p \le \infty\,. \tag{3.3}$$

Proof. Note that $\mathcal{L}_A v_j v_j^T = v_j v_j^T - \sigma_j^2 u_j u_j^T$, where u_j and v_j are, respectively, the jth left and right singular vectors of A such that $Av_j = \sigma_j u_j$. The lower bound follows from $\|u_j u_j^T\|_p = \|v_j v_j^T\|_p = 1$ and

$$\|\mathcal{L}_A\|_p \ge \|v_j v_j^T - \sigma_j^2 u_j u_j^T\|_p \ge \left|\|v_j v_j^T\|_p - \|\sigma_j^2 u_j u_j^T\|_p\right| = \left|1 - \sigma_j^2\right|\,. \tag{3.4}$$

The upper bound follows from

$$\|\mathcal{L}_A P\|_p \le \|P\|_p + \|APA^T\|_p \le \|P\|_p + \|A\|_\infty^2 \|P\|_p\,. \qquad \square \tag{3.5}$$

If A is normal, $\sigma_j(A)$ can be replaced by $|\lambda_j(A)|$ in Theorem 3.1, and $\|\mathcal{L}_A\|_p \le 1 + |\lambda_1(A)|^2$. If A is normal and $(-\overline{\lambda_1(A)})$ is an eigenvalue of A, then $1 + |\lambda_1(A)|^2$ is an eigenvalue of $\mathcal{L}_A$ and $\|\mathcal{L}_A\|_p = 1 + |\lambda_1(A)|^2$.

Theorem 3.1 shows that $\|\mathcal{L}_A\|_p$ is large and contributes to ill-conditioning if and only if $\sigma_1(A)$ is large, a situation that occurs in various applications [3, 22]. If $\sigma_1(A) \gg 1$ and $|\lambda_1(A)| < 1$, A is highly nonnormal [8, p. 314] and, as Corollary 4.8 will show, close to an unstable matrix.

4. The norm of the inverse Lyapunov operator. We first show that a sufficient condition for $\|\mathcal{L}_A^{-1}\|_p$ to be large is that $\lambda_1(A)$ be near the unit circle. The condition is necessary when A is normal.

THEOREM 4.1. *Let A be a stable matrix. Then*

$$\|\mathcal{L}_A^{-1}\|_p \geq \frac{1}{1-|\lambda_1(A)|^2}\,, \qquad 1 \leq p \leq \infty\,, \tag{4.1}$$

with equality holding if A is normal.

Proof. To obtain the lower bound, let z_1 be the leading eigenvector of A, $Az_1 = \lambda_1(A)z_1$, and note that $\mathcal{L}_A z_1 z_1^H = (1-|\lambda_1(A)|^2)z_1 z_1^H$, where $(\cdot)^H$ denotes conjugate transpose. Either $\operatorname{Re} z_1 z_1^H \neq 0$ or $\operatorname{Im} z_1 z_1^H \neq 0$ is an eigenvector of $\mathcal{L}_A$, and it follows that $\|\mathcal{L}_A^{-1}\|_p \geq (1-|\lambda_1(A)|^2)^{-1}$. Finally, if A is normal, then

$$\mathcal{L}_{A^T}^{-1} I = \mathcal{L}_A^{-1} I = \sum_{i=1}^{n} \frac{1}{1-|\lambda_i(A)|^2} z_i z_i^H\,, \tag{4.2}$$

and $\|\mathcal{L}_A^{-1}\|_\infty = \|\mathcal{L}_A^{-1}\|_1 = (1-|\lambda_1(A)|^2)^{-1}$. Using Lemma 2.2 gives $\|\mathcal{L}_A^{-1}\|_p \leq (1-|\lambda_1(A)|^2)^{-1}$ when A is normal, and therefore $\|\mathcal{L}_A^{-1}\|_p = (1-|\lambda_1(A)|^2)^{-1}$. ∎

When A is nonnormal, $\|\mathcal{L}_A^{-1}\|_p$ can be large without $\lambda_1(A)$ being near the unit circle. For instance, if $\sigma_1(A)$ is large or, more generally, if $\|A^k\|_\infty$ converges to zero slowly as a function of k, then $\|\mathcal{L}_A^{-1}\|_p$ is large. We show this fact first for $p = 1, \infty$.

THEOREM 4.2. *Let A be a stable matrix. For all $m \geq 1$,*

$$\|\mathcal{L}_A^{-1}\|_1 = \left\| \sum_{k=0}^{\infty} \left(A^k\right)^{\mathrm{T}} A^k \right\|_\infty \geq \left\| \sum_{k=0}^{m} \left(A^k\right)^{\mathrm{T}} A^k \right\|_\infty + \frac{\sigma_n^{2(m+1)}(A)}{1-\sigma_n^2(A)}\,, \tag{4.3}$$

$$\|\mathcal{L}_A^{-1}\|_\infty = \left\| \sum_{k=0}^{\infty} A^k \left(A^k\right)^{\mathrm{T}} \right\|_\infty \geq \left\| \sum_{k=0}^{m} A^k \left(A^k\right)^{\mathrm{T}} \right\|_\infty + \frac{\sigma_n^{2(m+1)}(A)}{1-\sigma_n^2(A)}\,. \tag{4.4}$$

In particular,

$$\|\mathcal{L}_A^{-1}\|_p \geq 1 + \sigma_1^2(A) + \frac{\sigma_n^4(A)}{1-\sigma_n^2(A)}\,, \qquad p = 1, \infty\,. \tag{4.5}$$

Proof. The operator $\mathcal{L}_A^{-1}$ applied to $S \in \mathbb{R}^{n\times n}$ can be expressed as [18]

$$\mathcal{L}_A^{-1} S = \sum_{k=0}^{\infty} A^k S \left(A^k\right)^{\mathrm{T}}. \tag{4.6}$$

Applying Lemma 2.4 gives $\|\mathcal{L}_A^{-1}\|_\infty = \|\mathcal{L}_A^{-1} I\|_\infty$, with the inequality in (4.4) being a consequence of

$$\left\| \sum_{k=0}^{\infty} A^k \left(A^k\right)^{\mathrm{T}} \right\|_\infty \geq \left\| \sum_{k=0}^{m} A^k \left(A^k\right)^{\mathrm{T}} \right\|_\infty + \lambda_n \left(\sum_{k=m+1}^{\infty} A^k \left(A^T\right)^k \right), \tag{4.7}$$

and

$$\lambda_n \left(\sum_{k=m+1}^{\infty} A^k \left(A^T\right)^k \right) \geq \sum_{k=m+1}^{\infty} \lambda_n \left(A^k \left(A^T\right)^k \right) \geq \sum_{k=m+1}^{\infty} \sigma_n^{2k}(A) = \frac{\sigma_n^{2(m+1)}(A)}{1-\sigma_n^2(A)}\,, \tag{4.8}$$

where we have used the facts that for matrices $W, X, Y \in \mathbb{R}^{n\times n}$ with X, Y being symmetric positive semidefinite, $\lambda_i(X+Y) \geq \lambda_i(X) + \lambda_n(Y)$, and $\lambda_i(WXW^T) \geq \sigma_n^2(W)\lambda_i(X)$ [17]. Likewise the $p = 1$ results follow from $\|\mathcal{L}_A^{-1}\|_1 = \|\mathcal{L}_{A^T}^{-1} I\|_\infty$. ∎

Lower bounds for $1 < p < \infty$ follow trivially, e.g.,

$$\|\mathcal{L}_A^{-1}\|_p \geq \frac{\|\mathcal{L}_A^{-1} I\|_p}{\|I\|_p} = \frac{\|\mathcal{L}_A^{-1} I\|_p}{n^{1/p}} \geq n^{-1/p} \|\mathcal{L}_A^{-1}\|_\infty \,, \tag{4.9}$$

but give little information when n is large. A lower bound for $1 \leq p \leq \infty$ depending on $\sigma_1(A)$ and independent of n is given in Corollary 4.9.

We now relate $\|\mathcal{L}_A^{-1}\|_p$ to the distance from A to the set of unstable matrices as measured by its *radius of stability* [15].

DEFINITION 4.3. *For any stable matrix $A \in \mathbb{R}^{n\times n}$ define the radius of stability $r(A)$ by*

$$r(A) \equiv \min_{0\leq\theta\leq 2\pi} \|(e^{i\theta} I - A)^{-1}\|_\infty^{-1} = \min_{0\leq\theta\leq 2\pi} \|R(e^{i\theta}, A)\|_\infty^{-1} \,, \tag{4.10}$$

where the resolvent of A is $R(\lambda, A) = (\lambda I - A)^{-1}$.

If A is normal and stable, then $r(A) = 1 - |\lambda_1(A)|$. However, if A is nonnormal and if its eigenvalues are *sensitive* to perturbations, then $r(A) \ll 1 - |\lambda_1(A)|$. The sensitivity of the eigenvalues of A is most completely described by its *pseudospectrum* [21]. The radius of stability $r(A)$ is the largest value of ϵ such that the ϵ-pseudospectrum of A lies inside the unit circle; $r(A)$ being small indicates that the ϵ-pseudospectrum of A is close to the unit circle for small ϵ. The following theorem shows that when $r(A)$ is small, $\|\mathcal{L}_A^{-1}\|_p$ must be large.

THEOREM 4.4 (proven for $p = \infty$ in [7]). *Let A be a stable matrix. Then*

$$\|\mathcal{L}_A^{-1}\|_p \geq \frac{1}{2r(A) + r^2(A)} \,, \qquad 1 \leq p \leq \infty \,. \tag{4.11}$$

Proof. There exists a matrix $E \in \mathbb{R}^{n\times n}$ with $|\lambda_1(A + E)| = 1$ and $\|E\|_\infty = r(A)$. Therefore, there exists a vector x with $x^H x = 1$ such that $(A + E)x = e^{i\theta} x$ for some $0 \leq \theta \leq 2\pi$. Using $\|xx^H\|_p = 1$ and Lemma 2.1 gives

$$\begin{aligned} \|\mathcal{L}_A xx^H\|_p &= \| - Exx^H E^T + e^{i\theta} xx^H E^T + e^{-i\theta} Exx^H \| \\ &\leq \|Exx^H E^T\|_p + \|xx^H E^T\|_p + \|Exx^H\|_p \\ &\leq \|E\|_\infty^2 + 2\|E\|_\infty = r^2(A) + 2r(A) \,, \end{aligned} \tag{4.12}$$

and we have

$$\|\mathcal{L}_A^{-1}\|_p \geq \frac{\|\mathcal{L}_A^{-1}\mathcal{L}_A xx^H\|_p}{\|\mathcal{L}_A xx^H\|_p} = \frac{1}{\|\mathcal{L}_A xx^H\|_p} \geq \frac{1}{2r(A) + r^2(A)} \,. \tag{4.13}$$

∎

A consequence of Theorem 4.4 is the following lower bound for $r(A)$ in terms of $\|\mathcal{L}_A^{-1}\|_p$.

COROLLARY 4.5. *Let A be a stable matrix. Then*

$$r(A) \geq \frac{\|\mathcal{L}_A^{-1}\|_p^{-1}}{1 + \sqrt{1 + \|\mathcal{L}_A^{-1}\|_p^{-1}}} \,, \qquad 1 \leq p \leq \infty \,. \tag{4.14}$$

Bounds for $r(A)$ are useful in robust stability [12] and in the study of perturbations of the discrete algebraic Riccati equation (DARE) [19]. In [19, Lemma 2.2] the bound

$$r(A) \geq \frac{\|\mathcal{L}_A^{-1}\|_\infty^{-1}}{\sigma_1(A) + \sqrt{\sigma_1^2(A) + \|\mathcal{L}_A^{-1}\|_\infty^{-1}}} \tag{4.15}$$

was used to formulate conditions under which a perturbed DARE has a unique, symmetric, positive definite solution. Since the lower bound in (4.14) with $p = \infty$ is sharper than that in (4.15) when $\sigma_1(A) > 1$, it can be used to show existence of a unique, symmetric, positive definite solution of the perturbed DARE for a larger class of perturbations [19, Theorem 4.1].

We generalize to Schatten p-norms the conjecture of [7] proven in [20] for the Frobenius norm.

THEOREM 4.6. *Let A be a stable matrix. Then*

$$\|\mathcal{L}_A^{-1}\|_p \le \frac{1}{r^2(A)}\,, \quad 1 \le p \le \infty\,. \tag{4.16}$$

Proof. $\mathcal{L}_A^{-1} I$ can be expressed as [20, 13]

$$\mathcal{L}_A^{-1} I = \frac{1}{2\pi}\int_0^{2\pi} R(e^{i\theta}, A) R(e^{i\theta}, A)^H \, d\theta. \tag{4.17}$$

Therefore, from Lemma 2.4,

$$\|\mathcal{L}_A^{-1}\|_\infty = \|\mathcal{L}_A^{-1} I\|_\infty \le \frac{1}{2\pi}\int_0^{2\pi} \|R(e^{i\theta}, A)\|_\infty^2 \, d\theta \le \frac{1}{r^2(A)}\,. \tag{4.18}$$

The inequality (4.16) for $p = 1$ follows from $\|\mathcal{L}_A^{-1}\|_1 = \|\mathcal{L}_{A^T}^{-1} I\|_\infty$ and $r(A) = r(A^T)$. The theorem follows from Lemma 2.2. ∎

As a consequence, any solution of the DALE can be used to obtain an upper bound for $r(A)$.

COROLLARY 4.7. *Let A be a stable matrix and let $\mathcal{L}_A P = Q$. Then*

$$r^2(A) \le \frac{\|Q\|_p}{\|P\|_p}\,, \qquad 1 \le p \le \infty\,. \tag{4.19}$$

Theorem 4.6 can be combined with any lower bound for $\|\mathcal{L}_A^{-1}\|_p$ to obtain an upper bound for $r(A)$. For instance, from Theorem 4.2 we get the following upper bound.

COROLLARY 4.8. *Let A be a stable matrix. Then*

$$r^2(A) \le \frac{1}{1 + \sigma_1^2(A)}\,. \tag{4.20}$$

Combining Corollary 4.8 and Theorem 4.4 gives a lower bound for $\|\mathcal{L}_A^{-1}\|_p$.

COROLLARY 4.9. *Let A be a stable matrix. Then*

$$\|\mathcal{L}_A^{-1}\|_p \ge \frac{1 + \sigma_1^2(A)}{1 + 2\sqrt{1 + \sigma_1^2(A)}}\,, \qquad 1 \le p \le \infty\,. \tag{4.21}$$

5. Examples. We present three examples that illustrate how ill-conditioning of $\mathcal{L}_A$ leads to low-rank approximate solutions of the DALE.

Example 1. *Almost unit eigenvalues.* Take $A = \lambda z z^T$, where λ and z are real, $0 < \lambda < 1$, and $z^T z = 1$. The matrix A is symmetric and $\mathcal{L}_A$ is self-adjoint. The eigenvalues of A are $(\lambda, 0, \ldots, 0)$. The operator $\mathcal{L}_A$ has singular values (and eigenvalues) $(1, \ldots, 1, 1 - \lambda^2)$. Therefore, $\|\mathcal{L}_A\|_2 = 1$ and $1 \le \|\mathcal{L}_A\|_p \le 1 + \lambda^2$ from Theorem 3.1. The norm of the inverse Lyapunov operator is

$$\|\mathcal{L}_A^{-1}\|_p = \frac{1}{1 - \lambda^2}\,, \qquad 1 \le p \le \infty\,, \tag{5.1}$$

according to Theorem 4.1. As the eigenvalue λ approaches the unit circle, $\mathcal{L}_A$ is increasingly poorly conditioned. The solution of the DALE for this choice of A is

$$P = \frac{\lambda^2}{1-\lambda^2}\left(z^T Q z\right) z z^T + Q\,. \tag{5.2}$$

A "natural" rank-1 approximation $\tilde{P}$ of P is $\tilde{P} = \lambda^2(1-\lambda^2)^{-1}(z^TQz)zz^T$. As the eigenvalue λ approaches the unit circle, if (z^TQz) is nonzero, P is increasingly well approximated by $\tilde{P}$ in the sense that $\|P-\tilde{P}\|_p/\|P\|_p$ approaches zero.

Example 2. *Large singular values.* Take $A = \sigma y z^T$, where $\sigma > 0$ and y and z are real unit n-vectors. The matrix A has at most one nonzero eigenvalue, namely, $\lambda = \sigma(y^Tz)$, taken to be less than one in absolute value. The sensitivity s of the eigenvalue λ is the cosine of the angle between y and z, i.e., $s = \lambda/\sigma$ for $\lambda \neq 0$, indicating that λ is sensitive to perturbations to A when σ is large [8].

Theorem 3.1 gives that $1+\sigma^2 \geq \|\mathcal{L}_A\|_p \geq |1-\sigma^2|$, showing that $\|\mathcal{L}_A\|_p$ is large when σ is large. From Lemmas 2.3 and 2.4,

$$\|\mathcal{L}_A^{-1}\|_1 = \|\mathcal{L}_A^{-1}\|_\infty = 1 + \frac{\sigma^2}{1-\lambda^2}\,, \tag{5.3}$$

and it follows from Lemma 2.2 that $\|\mathcal{L}_A^{-1}\|_p \leq 1+\sigma^2/(1-\lambda^2)$. A lower bound for the $(p=2)$-norm is

$$\|\mathcal{L}_A^{-1}\|_2 \geq \|\mathcal{L}_A^{-1} z z^T\|_2 = \sqrt{1 + 2\frac{\lambda^2}{1-\lambda^2} + \frac{\sigma^4}{(1-\lambda^2)^2}}\,. \tag{5.4}$$

The matrix A is near an unstable matrix either when $|\lambda|$ is near unity or when σ is large since

$$\left\|\left(e^{i\theta}I - \sigma y z^T\right)^{-1}\right\|_\infty = \left\|e^{-i\theta}I + \frac{\sigma e^{-2i\theta}}{1-\lambda e^{-i\theta}} y z^T\right\|_\infty \geq 1 + \frac{2|\lambda|}{1-|\lambda|} + \frac{\sigma^2}{(1-|\lambda|)^2}\,. \tag{5.5}$$

Therefore, $r(A) \leq (1-|\lambda|)/\sigma$ and a lower bound on $\|\mathcal{L}_A^{-1}\|_p$ follows from Theorem 4.4. When either $|\lambda|$ is close to unity or σ is large, $r(A)$ is small and $\kappa_p(\mathcal{L}_A)$ is large.

The solution of the DALE is

$$P = \frac{\sigma^2}{1-\lambda^2}\left(z^T Q z\right) y y^T + Q\,. \tag{5.6}$$

When $\mathcal{L}_A$ is ill-conditioned and $\left(z^TQz\right) \neq 0$, the rank-1 matrix $\tilde{P} = \sigma^2(1-\lambda^2)^{-1}\times (z^TQz)yy^T$ is a good approximation of P in the sense that $\|P-\tilde{P}\|_p/\|P\|_p$ is small.

Example 3. *Sensitive eigenvalues.* Consider the dynamics arising from the one-dimensional advection equation, $w_t + w_x = 0$ for $0 \leq x \leq n$, with boundary condition $w(0,t) = 0$. The matrix A that advances the n-vector $w(x = 1,2,\ldots,n, t = t_0)$ to $w(x = 1,2,\ldots,n, t = t_0+1)$ is the $n \times n$ matrix with ones on the subdiagonal and zero elsewhere, i.e., the transpose of an $n \times n$ Jordan block with zero eigenvalue. Adding stochastic forcing with covariance Q at unit time intervals leads to the DALE, $\mathcal{L}_A P = Q$, where P is the steady-state covariance of w.

Since $\sigma_1(A) = 1$, Theorem 3.1 yields $1 \leq \|\mathcal{L}_A\|_p \leq 2$. Further, since $\|\mathcal{L}_A\|_1 \geq \|\mathcal{L}_A e_1 e_1^T\|_1 = \|e_1 e_1^T - e_2 e_2^T\|_1 = 2$, where e_j is the jth column of the identity matrix,

$\|\mathcal{L}_A\|_1 = 2$. A similar argument with $\mathcal{L}_{A^T}$ gives $\|\mathcal{L}_A\|_\infty = 2$. Calculating $\mathcal{L}_A^{-1} I$ and $\mathcal{L}_{A^T}^{-1} I$ gives $\|\mathcal{L}_A^{-1}\|_\infty = \|\mathcal{L}_A^{-1}\|_1 = n$. Therefore, using Lemma 2.2, $\|\mathcal{L}_A^{-1}\|_p \le n$. Also,

$$\|\mathcal{L}_A^{-1}\|_2 \ge \frac{\|\mathcal{L}_A^{-1} e_1 e_1^T\|_2}{\|e_1 e_1^T\|_2} = \sqrt{n}\,. \tag{5.7}$$

A direct calculation shows that

$$\|(e^{i\theta} I - A)^{-1}\|_2^2 = \left\| \sum_{k=0}^{n-1} A^k e^{-i(k+1)\theta} \right\|_2^2 = \frac{n(n+1)}{2} \tag{5.8}$$

for any real θ. Since $\sqrt{n}\|(e^{i\theta} I - A)^{-1}\|_\infty \ge \|(e^{i\theta} I - A)^{-1}\|_2$, we have $r^2(A) \le 2/(n+1)$. Theorem 4.4 then gives a lower bound for $\|\mathcal{L}_A^{-1}\|_p$, $1 \le p \le \infty$. Thus as n becomes large; that is, as the domain becomes large with respect to the advection length scale, $\mathcal{L}_A$ is increasingly ill-conditioned.

The elements P_{ij} of the solution P of the DALE are

$$P_{ij} = e_i^T P e_j = \sum_{k=0}^{n-1} e_i^T A^k Q (A^T)^k e_j = \sum_{k=0}^{\min(i-1, j-1)} Q_{i-k, j-k}\,. \tag{5.9}$$

Therefore, if $Q = Q^T > 0$, a "natural" rank-m approximation of P is the matrix $\tilde{P}$ defined by

$$\tilde{P}_{i,j} = \begin{cases} P_{i,j}\,, & n - m < i, j \le n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

When Q is diagonal, P is also diagonal and

$$P_{ii} = \sum_{k=1}^{i} Q_{kk}\,. \tag{5.11}$$

In this case, each $Q_{kk} > 0$ and $\tilde{P}$ is the best rank-m approximation of P in the sense of minimizing $\|P - \tilde{P}\|_p$. We note that $\tilde{P}$ is associated with the left-most part of the domain $0 \le x \le n$.

6. Concluding remarks. Results about $\|\mathcal{L}_A^{-1}\|_p$ translate into bounds for solutions of the DALE. For instance, the solution P of the DALE for $Q = Q^T \ge 0$ satisfies

$$\operatorname{tr} P \le \|\mathcal{L}_A^{-1}\|_1 \operatorname{tr} Q\,, \tag{6.1}$$

and the upper bound is achieved for $Q = w_1 w_1^T$, where w_1 is the leading eigenvector of $\mathcal{L}_{A^T}^{-1} I$. In the $(p = \infty)$-norm, $\mathcal{L}_A^{-1}$ achieves its norm on the identity. In the $(p = 2)$-norm, $\mathcal{L}_A^{-1}$ does not in general achieve its norm on the identity, and the question arises whether it achieves its norm on any symmetric, positive semidefinite matrix. The forward operator $\mathcal{L}_A$ does not in general assume its norm on a symmetric, positive semidefinite matrix. The following theorem states that $\mathcal{L}_A^{-1}$ does achieve its $(p = 2)$-norm on a symmetric, positive semidefinite matrix.

THEOREM 6.1. *There exists a matrix $S = S^T \ge 0$ such that $\|\mathcal{L}_A^{-1} S\|_2 / \|S\|_2 = \|\mathcal{L}_A^{-1}\|_2$.*

Proof. Theorem 8 of [4] states that the inverse of the stable, continuous-time Lyapunov operator achieves its $(p = 2)$-norm on a symmetric matrix. The proof is easily adapted to give that $\mathcal{L}_A^{-1}$ achieves its $(p = 2)$-norm on a symmetric matrix. We now show that if $\mathcal{L}_A^{-1}$ achieves its $(p = 2)$-norm on a symmetric matrix, it does so on a symmetric, positive semidefinite matrix. Suppose that $\|\mathcal{L}_A^{-1}S\|_2/\|S\|_2 = \|\mathcal{L}_A^{-1}\|_2$ and S is symmetric with Schur decomposition $S = UDU^T$. Define the symmetric, positive semidefinite matrix $S^+ = U|D|U^T$. Then $\|S\|_2 = \|S^+\|_2$ and $-S^+ \leq S \leq S^+$. The positiveness of the stable, discrete-time inverse Lyapunov operator mapping implies that $-\mathcal{L}_A^{-1}S^+ \leq \mathcal{L}_A^{-1}S \leq \mathcal{L}_A^{-1}S^+$, which implies that $\|\mathcal{L}_A^{-1}S\|_2 \leq \|\mathcal{L}_A^{-1}S^+\|_2$. Therefore,

$$\frac{\|\mathcal{L}_A^{-1}S\|_2}{\|S\|_2} = \frac{\|\mathcal{L}_A^{-1}S\|_2}{\|S^+\|_2} \leq \frac{\|\mathcal{L}_A^{-1}S^+\|_2}{\|S^+\|_2}\,. \tag{6.2}$$ ∎

Additional information about the leading singular vectors of $\mathcal{L}_A^{-1}$ could be useful for determining low-rank approximations of P. The power method can be applied to $\mathcal{L}_{A^T}^{-1}\mathcal{L}_A^{-1}$ to calculate the leading right singular vector and singular value of $\mathcal{L}_A^{-1}$ [7]. However, this approach requires solving two DALEs at each iteration, which may be impractical for large n. If it is practical to store P and to apply $\mathcal{L}_A$ and $\mathcal{L}_{A^T}$, a Lanczos method could be used to compute the trailing eigenvectors of $\mathcal{L}_A\mathcal{L}_{A^T}$ while avoiding the cost of solving any DALEs.

Appendix. *Proof of Theorem* 1.1. By definition, $\|P\|_p \leq \|\mathcal{L}_A^{-1}\|_p\,\|Q\|_p$, and it remains to show that $\|P^{-1}\|_\infty \leq \|\mathcal{L}_A\|_\infty\,\|Q^{-1}\|_\infty$. Since $P = P^T > 0$, there is a nonzero $x \in \mathbb{R}^n$ such that

$$\|P^{-1}\|_\infty = \frac{1}{\lambda_n(P)} = \frac{x^Tx}{x^T\left(\mathcal{L}_A^{-1}Q\right)x} = \frac{\operatorname{tr} xx^T}{\operatorname{tr}\left(\mathcal{L}_A^{-1}Q\right)xx^T} = \frac{\operatorname{tr} xx^T}{\operatorname{tr}\left(\left(\mathcal{L}_{A^T}\right)^{-1}xx^T\right)Q}\,. \tag{A.1}$$

Let $B = \mathcal{L}_{A^T}^{-1}(xx^T)$ and note $B = B^T \geq 0$. Then using Lemma 2.3 and $\operatorname{tr} BQ \geq \lambda_n(Q)\operatorname{tr} B$ gives

$$\|P^{-1}\|_\infty = \frac{\operatorname{tr}\mathcal{L}_{A^T}B}{\operatorname{tr} BQ} \leq \frac{\operatorname{tr}\mathcal{L}_{A^T}B}{\operatorname{tr} B}\frac{1}{\lambda_n(Q)} \leq \|\mathcal{L}_{A^T}\|_1\|Q^{-1}\|_\infty = \|\mathcal{L}_A\|_\infty\|Q^{-1}\|_\infty\,. \tag{A.2}$$ ∎

Theorem 1.1 holds for $1 \leq p \leq \infty$ given some restrictions on A. From [16], $\lambda_i(P) \geq \lambda_i(Q) + \sigma_n^2(A)\lambda_n(P)$, and it follows that $\|P^{-1}\|_p \leq \|Q^{-1}\|_p$ for $1 \leq p \leq \infty$. From Theorem 3.1, $\|\mathcal{L}_A\|_p \geq 1$ if either A is singular or $\sigma_1^2(A) \geq 2$. Therefore, if either A is singular or $\sigma_1^2(A) \geq 2$,

$$\|P^{-1}\|_p \leq \|\mathcal{L}_A\|_p\|Q^{-1}\|_p\,, \qquad 1 \leq p \leq \infty\,. \tag{A.3}$$

Acknowledgments. The authors thank Greg Gaspari for valuable observations and notation suggestions and the reviewer for useful comments.

REFERENCES

[1] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[2] R. Bhatia, *A note on the Lyapunov equation*, Linear Algebra Appl., 259 (1997), pp. 71–76.

[3] K. M. Butler and B. F. Farrell, *Three-dimensional optimal perturbations in viscous shear flow*, Phys. Fluids A, 4 (1992), pp. 1637–1650.
[4] R. Byers and S. Nash, *On the singular "vectors" of the Lyapunov operator*, SIAM J. Alg. Discrete Methods, 8 (1987), pp. 59–66.
[5] S. E. Cohn and R. Todling, *Approximate data assimilation schemes for stable and unstable dynamics*, J. Meteor. Soc. Japan, 74 (1996), pp. 63–75.
[6] B. F. Farrell and P. J. Ioannou, *Generalized stability theory. Part* I: *Autonomous operators*, J. Atmospheric Sci., 53 (1996), pp. 2025–2040.
[7] P. M. Gahinet, A. J. Laub, C. S. Kenney, and G. A. Hewer, *Sensitivity of the stable discrete-time Lyapunov equation*, IEEE Trans. Automat. Control, 35 (1990), pp. 1209–1217.
[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[9] P. L. Houtekamer and H. L. Mitchell, *Data assimilation using an ensemble Kalman filter technique*, Monthly Weather Rev., 126 (1998), pp. 796–811.
[10] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[11] R. E. Kalman and J. E. Bertram, *Control system analysis and design via the "second method" of Lyapunov.* II. *Discrete-time systems*, Trans. ASME Ser. D. J. Basic Engrg., 82 (1960), pp. 394–400.
[12] S. R. Kolla, *Improved stability robustness bounds for digital control systems in state-space models*, Internat. J. Control, 64 (1996), pp. 991–994.
[13] P. Lancaster, *Explicit solutions of linear matrix equations*, SIAM Rev., 12 (1970), pp. 544–566.
[14] C. C. MacDuffee, *The Theory of Matrices*, Chelsea, New York, 1956.
[15] T. Mori, *On the relationship between the spectral radius and the stability radius of discrete systems*, IEEE Trans. Automat. Control, 35 (1990), p. 835.
[16] T. Mori, N. Fukuma, and M. Kuwahara, *Upper and lower bounds for the solution to the discrete Lyapunov matrix equation*, Internat. J. Control, 36 (1982), pp. 889–892.
[17] A. Ostrowski, *A quantitative formulation of Sylvester's law of inertia*, Proc. Natl. Acad. Sci. USA, 45 (1959), pp. 740–744.
[18] K. M. Przyluski, *The Lyapunov equation and the problem of stability for linear bounded discrete-time systems in Hilbert space*, Appl. Math. Optim., 6 (1980), pp. 97–112.
[19] J.-G. Sun, *Perturbation theory for algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 39–65.
[20] M. K. Tippett and D. Marchesin, *Upper bounds for the solution of the discrete algebraic Lyapunov equation*, Automatica, 35 (1999), pp. 1485–1489.
[21] L. N. Trefethen, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.
[22] L. N. Trefethen, A. E. Trefethen, and S. C. Reddy, *Hydrodynamic stability without eigenvalues*, Science, 261 (1993), pp. 578–584.
[23] J. S. Whitaker and P. D. Sardeshmukh, *A linear theory of extratropical synoptic Eddy statistics*, J. Atmospheric Sci., 55 (1998), pp. 237–258.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 66–85

POSITIVE SUBDEFINITE MATRICES, GENERALIZED MONOTONICITY, AND LINEAR COMPLEMENTARITY PROBLEMS*

J.-P. CROUZEIX†, A. HASSOUNI‡, A. LAHLOU‡, AND S. SCHAIBLE§

Abstract. Positive subdefinite matrices were introduced by Martos to characterize generalized convex quadratic functions. This concept is extended to nonsymmetric matrices. It leads to a study of pseudomonotone matrices and to new characterizations of generalized monotone affine maps. Finally, some properties of linear complementarity problems involving such maps are derived.

Key words. positive subdefiniteness, copositivity, copositivity star, generalized monotonicity, linear complementarity

AMS subject classifications. Primary, 26B25; Secondary, 90C26, 90C33

PII. S0895479897331849

1. Introduction and notation. Let $\mathcal{F} : \mathbb{R}^n_+ \to \mathbb{R}^n$, where $\mathbb{R}^n_+$ denotes the nonnegative orthant of $\mathbb{R}^n$. The complementarity problem associated with $\mathcal{F}$ is to

$$\text{find} \quad x \geq 0 \quad \text{such that} \quad \mathcal{F}(x) \geq 0 \quad \text{and} \quad \langle \mathcal{F}(x), x \rangle = 0.$$

If $\mathcal{F}$ is affine, then the problem is called a linear complementarity problem (LCP); see [1]. Complementarity problems can be considered as particular cases of variational inequality problems (VIP) which also include optimization problems. An important class of optimization problems assumes convexity of the objective function. In variational inequality problems, this assumption corresponds to monotonicity of the map $\mathcal{F}$.

We recall that, given a convex subset C of $\mathbb{R}^n$, a map $\mathcal{F} : C \mapsto \mathbb{R}^n$ is said to be *monotone* on C if

$$x, y \in C \quad \Rightarrow \quad \langle y - x, \mathcal{F}(y) - \mathcal{F}(x) \rangle \geq 0;$$

pseudomonotone on C if

$$x, y \in C, \quad \langle y - x, \mathcal{F}(x) \rangle > 0 \quad \Rightarrow \quad \langle y - x, \mathcal{F}(y) \rangle > 0,$$

or, equivalently, if

$$x, y \in C, \quad \langle y - x, \mathcal{F}(x) \rangle \geq 0 \quad \Rightarrow \quad \langle y - x, \mathcal{F}(y) \rangle \geq 0\,;$$

and *quasi-monotone* on C if

$$x, y \in C, \quad \langle y - x, \mathcal{F}(x) \rangle > 0 \quad \Rightarrow \quad \langle y - x, \mathcal{F}(y) \rangle \geq 0.$$

*Received by the editors December 8, 1997; accepted for publication (in revised form) by R. Cottle August 4, 1999; published electronically May 31, 2000. Part of this work was done while the third author was visiting Université Blaise Pascal. This visit was supported by Action Intégrée Franco-Marocaine 95/0849.

http://www.siam.org/journals/simax/22-1/33184.html

†LIMOS, Université Blaise Pascal, 63177 Aubière Cedex, France (crouzeix@ucfma.univ-bpclermont.fr).

‡Département de Mathématiques et d'Informatique, Faculté des Sciences, B.P. 1014, Rue Ibn Batouta, Rabat, Morocco (hassouni@fsr.ac.ma, amale_lahlou@usa.net).

§A. G. Anderson Graduate School of Management, University of California, Riverside, CA 92521 (siegfried.schaible@ucr.edu).

Monotonicity, pseudomonotonicity, and quasi monotonicity have been introduced by Minty [18], Karamardian [13], and Hassouni [11] and independently by Karamardian and Schaible [14], respectively. Complementarity problems, together with optimization problems, variational inequality problems, fixed point problems, saddle-point problems, and other classical problems, can be viewed as special realizations of an abstract equilibrium problem. Recently, the study of these various models has been enriched by relaxing monotonicity to generalized monotonicity. The usefulness of generalized monotonicity concepts among models of such diversity is demonstrated in the survey [10] which covers quasi-monotone and pseudomonotone variational inequality problems and equilibrium problems.

In the symmetric case, positive subdefinite matrices have been introduced by Martos [16] to characterize generalized convex quadratic functions on $\mathbb{R}^n_+$. Section 2 extends this concept to nonsymmetric matrices. Then, in section 3, new characterizations for generalized monotone affine maps on $\mathbb{R}^n_+$ are given using positive subdefinite matrices. Section 4 is devoted to linear complementarity problems involving generalized monotone affine maps. Finally, section 5 relates the results further to the existing literature.

We conclude this section with some comments on the notation used throughout this paper. Given $z \in \mathbb{R}^n$, z^+ and z^- are the vectors of $\mathbb{R}^n$ defined by $z_i^+ := \max\{z_i, 0\}$ and $z_i^- := \max\{-z_i, 0\}$ for all i, then $z = z^+ - z^-$.

Given $\emptyset \neq C \subseteq \mathbb{R}^n$, we denote by C^o the *polar cone* of C, i.e.,

$$C^o = \{x^* : \ \langle x, x^* \rangle \leq 0 \ \text{ for all } x \in C\}.$$

It is known that $C = C^{oo}$ if and only if C is a closed convex cone.

Given an $n \times n$ matrix M, the kernel of M and the rank of M are denoted by $\mathrm{Ker}(M)$ and $\mathrm{rank}(M)$, respectively. The (Moore–Penrose) *pseudoinverse* of M is the uniquely defined $n \times n$ matrix $M^\dagger$ which satisfies the following conditions:

$$MM^\dagger M = M, \ M^\dagger MM^\dagger = M^\dagger, \ (MM^\dagger)^t = MM^\dagger, \ \text{ and } \ (M^\dagger M)^t = M^\dagger M.$$

We denote by M^s the matrix

$$M^s = 2M^t(M + M^t)^\dagger M.$$

Here the symbol "s" indicates symmetrization: M^s is always symmetric and $M^s = M$ if and only if M is symmetric.

The matrix M is said to be *copositive* if $\langle x, Mx \rangle \geq 0$ for all $x \geq 0$ and *conegative* if $-M$ is copositive. M is said to be *copositive star* [7] if it is copositive with the additional condition that

$$M^t x \leq 0 \quad \text{whenever} \quad x \geq 0, \quad Mx \geq 0 \text{ and } \langle x, Mx \rangle = 0.$$

$M \leq 0$ means that all entries of M are nonpositive.

Given a symmetric $n \times n$ matrix B, its *inertia* is the triple

$$In(B) = (\nu_+(B), \nu_-(B), \nu_0(B)),$$

where $\nu_+(B)$, $\nu_-(B)$, and $\nu_0(B)$ denote the number of positive, negative, and zero eigenvalues of B, respectively. Then $\nu_+(B) + \nu_-(B) + \nu_0(B) = n$.

2. Positive subdefinite matrices. We call an $n \times n$ matrix M *positive subdefinite* (*PSBD*) if

$$\langle z, Mz\rangle < 0 \quad \text{implies either} \quad M^t z \le 0 \quad \text{or} \quad M^t z \ge 0.$$

The terminology is borrowed from Martos [16, 17], who characterized pseudoconvex and quasi-convex quadratic functions on the nonnegative orthant via such matrices. Since Martos was considering the Hessian of quadratic functions, he was concerned only about symmetric matrices. In this section, we study nonsymmetric positive subdefinite matrices. These matrices will be used in the next section to characterize pseudomonotone and quasi-monotone affine maps on the nonnegative orthant.

It is clear that a positive semidefinite (PSD) matrix is PSBD. According to Martos, a matrix M is said to be *merely positive subdefinite* ($MPSBD$) if M is PSBD but not *PSD*.

The particular case of PSBD matrices of rank 1 is easily studied with help of the definitions.

PROPOSITION 2.1. *Assume that $M = ab^t$ with $a, b \in \mathbb{R}^n$, $a, b \neq 0$. Then M is PSBD if and only if one of the following conditions holds:*

(i) *there is $t > 0$ so that $b = ta$;*

(ii) *$b \neq ta$ for all $t > 0$ and either $b \ge 0$ or $b \le 0$.*

Assume that M is MPSBD; then M is copositive if and only if either ($a \ge 0$ and $b \ge 0$) or ($a \le 0$ and $b \le 0$) and copositive star if and only if, in addition, $a_i = 0$ whenever $b_i = 0$.

Proof. Case (i) corresponds to M PSD. Let us consider the second case. Then M is PSBD if and only if

$$\langle a, z\rangle\langle b, z\rangle < 0 \quad \text{implies either} \quad \langle a, z\rangle b \ge 0 \quad \text{or} \quad \langle a, z\rangle b \le 0,$$

from which the result follows easily. The other results are obvious. □

For general matrices, we have the following result.

PROPOSITION 2.2. *Assume that M is MPSBD. Then*

(i) $\nu_-(M + M^t) = 1$;

(ii) $(M + M^t)z = 0 \Rightarrow Mz = M^t z = 0$.

Proof. Set $B = M + M^t$.

(i) B has at least one negative eigenvalue since M is not PSD. Assume, for contradiction, that λ_1, λ_2, z_1, and z_2 exist so that

$$Bz_1 = 2\lambda_1 z_1, \quad Bz_2 = 2\lambda_2 z_2, \quad \|z_1\|^2 = \|z_2\|^2 = 1,$$

$$\lambda_1 \le \lambda_2 < 0 \quad \text{and} \quad \langle z_1, z_2\rangle = 0.$$

Then both $\langle z_1, Mz_1\rangle$ and $\langle z_2, Mz_2\rangle$ are negative. Without loss of generality, we assume that $M^t z_1 \le 0$ and $M^t z_2 \ge 0$.

For $t \in [0, 1]$, define $z(t) = tz_1 + (1 - t)z_2$. Then

$$\langle z(t), Bz(t)\rangle = 2t^2\lambda_1 + 2(1 - t)^2\lambda_2 < 0.$$

Hence $0 \neq M^t z(t) = tM^t z_1 + (1 - t)M^t z_2 \in \mathbb{R}^n_+ \cup -\mathbb{R}^n_+$. Since $M^t z(0) \ge 0$ and $M^t z(1) \le 0$, there is $\bar{t} \in (0, 1)$ such that $M^t z(\bar{t}) = 0$, a contradiction.

(ii) Assume that z_1, λ_1 are defined as above and z_0 is so that $Bz_0 = 0$. For $t \in \mathbb{R}$, let us define $z(t) = z_1 + tz_0$. Then

$$\langle z(t), Bz(t)\rangle = 2\lambda_1 < 0.$$

Hence for all $t \in \mathbb{R}$

$$0 \neq M^t z(t) = M^t z_1 + tM^t z_0 \in \mathbb{R}^n_+ \cup -\mathbb{R}^n_+.$$

It follows that $M^t z_0 = 0$ because of $M^t z_1 \neq 0$. Then $Mz_0 = 0$ as well. □

It is clear that for any matrix M, the following two assertions are equivalent:

(i) $(M + M^t)z = 0 \Rightarrow Mz = M^t z = 0$;

(ii) $M(\mathbb{R}^n) \subseteq (M + M^t)(\mathbb{R}^n)$.

For such matrices, we have the following result (see Crouzeix and Schaible [5, Lemma 2]).

LEMMA 2.1. *Assume that $M(\mathbb{R}^n) \subseteq (M + M^t)(\mathbb{R}^n)$. Denote by k the dimension of the kernel of M. Then*

$$\begin{aligned}
\nu_+(M^s) &= \nu_+(M + M^t) + \nu_0(M + M^t) - k,\\
\nu_-(M^s) &= \nu_-(M + M^t) + \nu_0(M + M^t) - k,\\
\nu_0(M^s) &= 2k - \nu_0(M + M^t).
\end{aligned}$$

Although M itself is nonsymmetric, Proposition 2.2 shows that symmetric matrices having one and only one negative eigenvalue play a fundamental role. For these matrices, we have the following result.

PROPOSITION 2.3. *Assume that B is a symmetric $n \times n$ matrix and $\nu_-(B) = 1$. Then there exists a closed convex cone T such that*

$$T \cup -T = \{z \,:\, \langle Bz, z\rangle \leq 0\} \text{ and}$$

$$\text{int}(T) \cup -\text{int}(T) = \{z \,:\, \langle Bz, z\rangle < 0\}.$$

Furthermore, $T^o \cap -T^o = \{0\}$ and

$$T^o \cup -T^o = \{z^* \,:\, \langle B^\dagger z^*, z^*\rangle \leq 0\} \cap B(\mathbb{R}^n).$$

Proof. It follows from the assumptions that an $n \times n$ matrix P, a positive definite diagonal $q \times q$ matrix $\triangle_2$ with $q < n$ and $\lambda_1 < 0$ exist so that

$$P^t P = I \quad \text{and} \quad P^t B P = \begin{pmatrix} \lambda_1 & 0 & 0\\ 0 & \triangle_2 & 0\\ 0 & 0 & 0 \end{pmatrix}.$$

Then

$$T \cup -T = \{z \,:\, \langle Bz, z\rangle \leq 0\} = \left\{z = Py : \lambda_1 y_1^2 + \langle \triangle_2 y_2, y_2\rangle + \langle 0 y_3, y_3\rangle \leq 0\right\},$$

$$\text{where } T \,=\, \left\{z = Py \,:\, \sqrt{\frac{-1}{\lambda_1}\langle \triangle_2 y_2, y_2\rangle} \leq y_1\right\}.$$

Then T is a closed convex cone and

$$\text{int}(T) \,=\, \left\{z = Py : \sqrt{\frac{-1}{\lambda_1}\langle \triangle_2 y_2, y_2\rangle} < y_1\right\}.$$

Let us determine T^o. By definition, $z^* = Py^* \in T^o$ if and only if

$$0 = \sup_{z \in T} \langle z, z^* \rangle = \sup \left[y_1 y_1^* + \langle y_2, y_2^* \rangle + \langle y_3, y_3^* \rangle \ : \ \sqrt{\frac{-1}{\lambda_1} \langle \triangle_2 y_2, y_2 \rangle} \leq y_1 \right].$$

Hence $z^* = Py^* \in T^o$ if and only if

$$y_3^* = 0 \ ; \ y_1^* \leq 0 \ \text{ and } \ y_1^* \sqrt{\frac{-1}{\lambda_1} \langle \triangle_2 y_2, y_2 \rangle} + \langle y_2, y_2^* \rangle \leq 0 \text{ for all } y_2.$$

It follows that

$$T^o = \left\{ z^* = Py^* : \ y_3^* = 0, \ y_1^* \leq 0, \text{ and } \frac{1}{\lambda_1} {y_1^*}^2 + \langle \triangle_2^{-1} y_2^*, y_2^* \rangle \leq 0 \right\},$$

which gives the result. □

Now we deduce the following characterization of nonsymmetric PSBD matrices.

THEOREM 2.1. *Assume that M is an $n \times n$ matrix. Then M is PSBD if and only if exactly one of the following conditions holds:*

(i) *M is PSD;*

(ii) *$M = ab^t \neq 0$ with $a, b \in \mathbb{R}^n$, $a \neq tb$ for all $t > 0$ and either $b \geq 0$ or $b \leq 0$;*

(iii) *$\operatorname{rank}(M) \geq 2$, $\nu_-(M + M^t) = 1$, $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$ and the matrix M^s is conegative.*

Proof. Set $B = M + M^t$ and assume that $\nu_-(B) = 1$. Take T as in the last proposition. A necessary and sufficient condition for M to be MPSBD is that

$$\text{either } T \subseteq \{ z \ : \ M^t z \leq 0 \} \text{ or } \ -T \subseteq \{ z \ : \ M^t z \leq 0 \}. \tag{2.1}$$

By polarity on closed convex cones, the last condition is equivalent to

$$\text{either } T^o \supseteq M(\mathbb{R}^n_+) \text{ or } \ -T^o \supseteq M(\mathbb{R}^n_+). \tag{2.2}$$

(a) Assume that M is MPSBD. Then M^s is conegative because of condition (2.2). On the other hand, Proposition 2.2 implies that $\operatorname{rank}(M) \leq \operatorname{rank}(B)$. Hence Lemma 2.1 implies that either $k = 1 + \nu_0(B)$ or $k = \nu_0(B)$ since $\nu_-(M^s) \geq 0$.

Assume that $k = 1 + \nu_0(B)$. Then $\nu_-(M^s) = 0$ implies M^s is PSD, thus $M^s = 0$ since M^s is conegative. Hence $n = \nu_0(M^s) = 2k - \nu_0(B) = k + 1$ and M has rank 1. Apply Proposition 2.1.

Assume that $k = \nu_0(B)$. If the rank of M^s is 1, then apply again Proposition 2.1. If $\operatorname{rank}(M^s) \geq 2$, then condition (2.2) is equivalent to the condition

$$T^o \cup -T^o \supseteq M(\mathbb{R}^n_+). \tag{2.3}$$

Indeed T^o and $M(\mathbb{R}^n_+)$ are closed convex cones having the same dimension, and this dimension is greater than or equal to 2. Hence (iii) holds.

(b) We already know that M is PSBD in cases (i) or (ii). Assume that (iii) holds. Hence M is MPSBD since conditions (2.2) and (2.3) are equivalent in this case. □

From Theorem 2.1, we recover the characterization of symmetric MPSBD matrices by Martos [16, 17].

COROLLARY 2.1. *Assume that M is a symmetric $n \times n$ matrix and $\nu_-(M) = 1$. Then the following three conditions are equivalent:*

(i) *M is MPSBD;*
(ii) *M is conegative;*
(iii) *$M \leq 0$.*

Proof. The symmetry of M implies that $M^s = M$. Furthermore, if $M = ab^t$, then a and b are colinear. Hence, in view of Theorem 2.1, conditions (i) and (ii) above are equivalent. Condition (iii) obviously implies condition (ii). Conversely, assume that conditions (i) and (ii) hold. Then, for all $z \geq 0$, $\langle z, Mz\rangle \leq 0$, hence $Mz \leq 0$ or $Mz \geq 0$. It is then easily derived that M cannot have a positive entry. □

COROLLARY 2.2. *Assume that M is symmetric and copositive. Then M is PSBD if and only if M is PSD.*

Proof. A symmetric matrix which is both copositive and conegative is the null matrix. □

Theorem 2.1 sets in evidence the roles of the matrix M^s and the convex cone T such that $T \cup -T = \{z : \langle z, Mz\rangle \leq 0\}$ in the analysis of MPSBD matrices. For such matrices

$$\text{either} \quad T \subseteq \{z : M^t z \leq 0\} \quad \text{or} \quad T \subseteq \{z : M^t z \geq 0\}.$$

Henceforth, we assume that

$$T \subseteq \{z : M^t z \leq 0\} \quad \text{so that} \quad \text{int}\,(T) = \{z : \langle z, Mz\rangle < 0,\ M^t z \leq 0\}.$$

We begin with the analysis of M^s and T^o for matrices of rank 1.

PROPOSITION 2.4. *Assume that $M = ab^t$, $a, b \in \mathbb{R}^n$ with $a, b \neq 0$. Let u, v, and φ be such that*

$$a = \|a\|u, \quad b = \|b\|v, \quad \text{and} \quad \langle u, v\rangle = \cos 2\varphi.$$

(i) *Assume that a and b are colinear. Then*

$$2\|a\|^2\|b\|^2(M + M^t)^\dagger = M = M^s.$$

(ii) *Assume that a and b are noncolinear. Then $M^s = 0$ and*

$$8\ \|a\|\ \|b\|\ (M + M^t)^\dagger = \frac{1}{\cos^4\varphi}(u+v)(u+v)^t - \frac{1}{\sin^4\varphi}(u-v)(u-v)^t.$$

(iii) *If $b \geq 0$, then $T^o = \{q = \lambda a - \mu b : \lambda \geq 0, \mu \geq 0\}$.*

Proof. Case (i) is trivial. Henceforth, we assume that a and b are noncolinear. Then there is a matrix P such that

$$P^t P = I, \quad P^t u = \cos\varphi\, e_1 + \sin\varphi\, e_2, \quad \text{and} \quad P^t v = \cos\varphi\, e_1 - \sin\varphi\, e_2,$$

where e_1 and e_2 are the first two vectors of the canonical basis of $\mathbb{R}^n$. Set $R = uv^t$. Then $M = \|a\|\ \|b\|R$ and

$$P^t(R + R^t)^\dagger P = \frac{1}{2\cos^2\ \varphi}e_1 e_1^t - \frac{1}{2\sin^2\ \varphi}e_2 e_2^t.$$

Replace e_1 and e_2 by their expressions in terms of u, v, and φ. It holds that

$$(R + R^t)^\dagger = \left[\frac{1}{8\cos^4\varphi}(u+v)(u+v)^t - \frac{1}{8\sin^4\varphi}(u-v)(u-v)^t\right].$$

Next, it is easily seen that $R^s = 2R^t(R+R^t)^\dagger R = 0$. Then $M^s = 0$ as already shown in the proof of Theorem 2.1.

Finally, the expression of T^o is obtained by straight calculations. □

Next, we continue the analysis of M^s and T^o for matrices M such that

$$M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M+M^t)(\mathbb{R}^n).$$

The following lemma shows that for these matrices, the "s" symmetrization corresponds to the classical symmetrization on the (pseudo)inverses.

LEMMA 2.2. *Assume that* $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M+M^t)(\mathbb{R}^n)$. *Then*

(i) $M^s = (M^t)^s = 2\left(M^\dagger + (M^t)^\dagger\right)^\dagger$;

(ii) $2(M^s)^\dagger = M^\dagger + (M^t)^\dagger$.

Proof. Since $B = M + M^t$ is symmetric and $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = B(\mathbb{R}^n)$, there exist nonsingular matrices P, $\triangle$, and R so that

$$PP^t = I, \quad P^tBP = \begin{pmatrix} \triangle & 0 \\ 0 & 0 \end{pmatrix}, \quad \text{and} \quad P^tMP = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}.$$

Then $\triangle = R + R^t$ and

$$\frac{1}{2}P^tM^sP = (P^tM^tP)(P^tB^\dagger P)(P^tMP) = \begin{pmatrix} R^t\triangle^{-1}R & 0 \\ 0 & 0 \end{pmatrix}.$$

Next we observe that

$$(R^t\triangle^{-1}R)^{-1} = R^{-1}(R+R^t)(R^t)^{-1} = (R^t)^{-1} + R^{-1} = (R\triangle^{-1}R^t)^{-1}.$$

Hence the conclusions follow. □

Note that the formula for $(M^s)^\dagger$ is no longer true in case (ii) of Theorem 2.1 since then $M^s = 0$ according to Proposition 2.4. Now we reconsider condition (iii) of Theorem 2.1.

PROPOSITION 2.5. *Assume that* $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M+M^t)(\mathbb{R}^n)$ *and* $\nu_-(M+M^t) = 1$. *Then*

(a) M, M^t, *and* M^s *are PSBD if one of these matrices is so;*

(b) *assume that* M *is PSBD; then, either* $(M+M^t) \le 0$ *or* M *is copositive star. Furthermore,*

$$T^o = \{q = Mu : \langle u, M^su\rangle \le 0 \text{ and } M^su \le 0\}.$$

Proof. (a) Lemma 2.1 shows that M^s and $(M+M^t)$ have the same inertia. The conclusion follows from Theorem 2.1, Corollary 2.1, and Lemma 2.2.

(b) Both M and M^t are PSBD according to (a). Then one and only one of the following inclusions holds:

$$T \subseteq \{z : M^tz \le 0 \text{ and } Mz \le 0\}$$

or

$$T \subseteq \{z : M^tz \le 0 \text{ and } Mz \ge 0\}.$$

In the first case, $(M+M^t)$ is PSBD; hence $(M+M^t) \le 0$ in view of Corollary 2.1.

Assume that we are in the second case and M is not copositive. Then $u \ge 0$ exists so that $\langle u, Mu\rangle < 0$. Hence, either $u \in T$, $Mu \ge 0$ and then $\langle u, Mu\rangle \ge 0$ or

$u \in -T$, $M^t u \geq 0$ and then $\langle u, M^t u\rangle \geq 0$ as well. Both cases yield a contradiction. Finally, it is easy to see that in the second case M is copositive star.

Set $B = M + M^t$. Then

$$T \cup -T = \{z : \langle z, Mz\rangle \leq 0\} = \{z : \langle z, Bz\rangle \leq 0\}.$$

Hence, in view of Proposition 2.3 and since $B(\mathbb{R}^n) = M(\mathbb{R}^n)$, $q \in T^o \cup -T^o$ if and only if there is u such that $q = Mu$ and $\langle u, M^s u\rangle \leq 0$. Next, M^s is PSBD in view of (a) and symmetric. It follows that $M^s \leq 0$ and $T^o \cup -T^o = A \cup -A$, where A is the convex cone

$$A = \{q = Mu : \langle u, M^s u\rangle \leq 0 \text{ and } M^s u \leq 0\}.$$

Since $T^o \cap -T^o = \{0\}$, then either $A = T^o$ or $A = -T^o$. Let $u \geq 0$ and $z \in T$, then $M^s u \leq 0$, $\langle u, M^s u\rangle \leq 0$, $M^t z \leq 0$, and $\langle z, Mu\rangle = \langle u, M^t z\rangle \leq 0$. Hence, on the one hand $M(\mathbb{R}^n_+) \subseteq A$ and on the other hand $M(\mathbb{R}^n_+) \subseteq T^o$. Thus $A = T^o$. □

Next, Theorem 2.1 is restated as follows.

THEOREM 2.2. *Assume that M is an $n \times n$ matrix. Then M is PSBD if and only if exactly one of the following conditions holds:*

(i) *M is PSD;*

(ii) *$M = ab^t \neq 0$ with $a, b \in \mathbb{R}^n$, $a \neq tb$ for all $t > 0$ and either $b \geq 0$ or $b \leq 0$;*

(iii) *$\operatorname{rank}(M) \geq 2$, $\nu_-(M + M^t) = 1$, $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$, and $M^s \leq 0$.*

Proof. Only the conditions Theorem 2.1(iii) and Theorem 2.2(iii) differ. Assume that $\nu_-(M + M^t) = 1$ and $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$. Then M is PSBD if and only if M^s is PSBD and, according to Corollary 2.1, if and only if $M^s \leq 0$. □

Remarks. (1) $M = ab^t$ with a and b noncolinear is the only case, where M may be PSBD and M^t is not. Consider, for instance, $a^t = (1, -1)$ and $b^t = (0, 1)$.

(2) Assume that $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$, $\nu_-(M + M^t) = 1$, and M is PSBD. Then, according to Proposition 2.5, M is copositive (and copositive star as well) if and only if the matrix $(M + M^t)$ has a positive entry. An example of such a matrix is

$$M = \begin{pmatrix} 0 & 11 \\ -1 & 0 \end{pmatrix} \text{ for which } M^s = -\frac{11}{5}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

(3) An invertible matrix can be PSBD while its inverse may not be. Consider the matrix M of remark (2); then

$$(M^{-1})^s = \frac{1}{5}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

and hence M^{-1} is not PSBD.

3. Generalized monotone affine maps on the nonnegative orthant. In this section, we consider the affine map $\mathcal{F}(x) = Mx + q$, where M is an $n \times n$ matrix and $q \in \mathbb{R}^n$. A first result is as follows.

PROPOSITION 3.1. *$\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$ if and only if*

$$z \in \mathbb{R}^n \text{ and } \langle z, Mz\rangle < 0 \Rightarrow \begin{cases} M^t z \geq 0 \quad \text{and} \quad \langle z, q\rangle \geq 0, \\ \qquad \text{or} \\ M^t z \leq 0,\ \langle z, q\rangle \leq 0, \ \text{and} \ \langle z, Mz^- + q\rangle < 0. \end{cases}$$

Proof. It follows from the definition that $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$ if and only if for all $z \in \mathbb{R}^n$ we have

$$(C_z) \qquad \left.\begin{array}{l} x \geq 0,\, t \geq 0,\;\; x + tz \geq 0, \\ \langle Mx + q, z\rangle \geq 0 \end{array}\right\} \Rightarrow \langle Mx + q, z\rangle + t\langle Mz, z\rangle \geq 0.$$

Set

$$Z_1 = \{z : \langle Mz, z\rangle \geq 0\},\; Z_2 = \{z : x \geq 0, t \geq 0,\; x + tz \geq 0 \;\Rightarrow \langle Mx + q, z\rangle < 0\}.$$

By duality in linear programming, we have

$$Z_2 = \{z \;\; : \langle q, z\rangle < 0 \text{ and } M^t z \leq 0\}.$$

Since condition (C_z) holds when $z \in Z_1 \cup Z_2$, we assume now that $z \notin Z_1 \cup Z_2$. Then condition (C_z) is equivalent to the condition

$$0 \leq \langle q, z\rangle + \inf_{x\geq 0, t\geq 0}[\langle M^t z, x\rangle + t\langle Mz, z\rangle \;\; : \;\; x + tz \geq 0,\; \langle M^t z, x\rangle \geq -\langle q, z\rangle].$$

The linear program is feasible. Hence, applying duality in linear programming again, the condition is equivalent to

$$0 \leq \langle q, z\rangle + \sup_{y\geq 0, r\geq 0}[-r\langle q, z\rangle \;\; : \;\; y + rM^t z \leq M^t z,\; \langle z, y\rangle \leq \langle Mz, z\rangle].$$

This, in turn, is equivalent to $0 \leq \max(M_1, M_2, M_3)$, where for $i = 1, 2, 3$

$$M_i = \sup_{r\in R_i, y\geq 0}[(1 - r)\langle q, z\rangle \;\; : \;\; y \leq (1 - r)M^t z,\; \langle z, y\rangle \leq \langle Mz, z\rangle],$$

with $R_1 = \{1\},\; R_2 = [0, 1)$, and $R_3 = (1, +\infty)$.

Now $M_1 = -\infty$ since $z \notin Z_1$ implies $\langle z, Mz\rangle < 0$.

M_2 is nonnegative if and only if $\langle q, z\rangle \geq 0$, $M^t z \geq 0$, and $r \in [0, 1),\; y$ exist so that $0 \leq y \leq (1 - r)M^t z$, and $\langle z, y\rangle \leq \langle z, Mz\rangle < 0$. Take $r = 0$, $y_i = 0$ when $z_i \geq 0$ and $y_i = (1 - r)(M^t z)_i$ when $z_i < 0$. Then $M^t z \geq 0$ implies

$$\langle z, y\rangle = -\langle M^t z, z^-\rangle \leq \langle M^t z, z^+\rangle - \langle M^t z, z^-\rangle = \langle M^t z, z\rangle.$$

Hence,

$$M_2 \geq 0 \Longleftrightarrow \langle q, z\rangle \geq 0 \text{ and } M^t z \geq 0.$$

In the same manner, M_3 is nonnegative if and only if $\langle q, z\rangle \leq 0$, $M^t z \leq 0$ and $r > 1$ exists so that $(r - 1)\langle M^t z, z^-\rangle \leq \langle Mz, z\rangle < 0$. Finally,

$$M_3 \geq 0 \Longleftrightarrow \langle q, z\rangle \leq 0,\; M^t z \leq 0 \text{ and } \langle M^t z, z^-\rangle < 0.$$

Summarizing, $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$ if and only if

$$z \in \mathbb{R}^n, \quad \langle z, Mz\rangle < 0 \Rightarrow \left\{\begin{array}{llll} M^t z \geq 0 & \text{and} & \langle z, q\rangle \geq 0, & \\ \text{or} & & & \\ M^t z \leq 0, & \langle z, q\rangle \leq 0, & \text{and} & \langle M^t z, z^-\rangle < 0, \\ \text{or} & & & \\ M^t z \leq 0 & \text{and} & \langle z, q\rangle < 0. & \end{array}\right.$$

Taking into account that $\langle M^t z, z^- \rangle \leq 0$ when $M^t z \leq 0$, the last two conditions are summarized in

$$M^t z \leq 0, \quad \langle z, q \rangle \leq 0, \quad \text{and} \quad \langle z, Mz^- + q \rangle < 0.$$

This ends the proof. ∎

A necessary and sufficient condition for $\mathcal{F}$ to be quasi-monotone on $\mathbb{R}^n_+$ is easily derived.

PROPOSITION 3.2. *$\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ if and only if*

$$z \in \mathbb{R}^n, \quad \langle z, Mz \rangle < 0 \Rightarrow \begin{cases} M^t z \geq 0 \text{ and } \langle z, q \rangle \geq 0, \\ \quad \text{or} \\ M^t z \leq 0 \text{ and } \langle z, q \rangle \leq 0. \end{cases}$$

Proof. It is known [5] that an affine map $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ when it is pseudomonotone on the interior of $\mathbb{R}^n_+$. Let e be the vector of $\mathbb{R}^n_+$ with all its entries equal to 1. It is clear that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ if and only, for all $\varepsilon > 0$, $\mathcal{F}$ is pseudomonotone on $K_\varepsilon = \{x \ : \ x \geq \varepsilon e\}$. Set $x' = x - \varepsilon e$ and $\mathcal{F}_\varepsilon(x') = Mx + q = Mx' + (\varepsilon Me + q)$. Then $\mathcal{F}$ is pseudomonotone on K_ε if and only if $\mathcal{F}_\varepsilon$ is so on $\mathbb{R}^n_+$ or, equivalently, if and only if for all z so that $\langle z, Mz \rangle < 0$ it follows that

$$\begin{cases} M^t z \geq 0 \quad \text{and} \quad \varepsilon \langle M^t z, e \rangle + \langle z, q \rangle \geq 0, \\ \quad \text{or} \\ M^t z \leq 0, \ \varepsilon \langle M^t z, e \rangle + \langle z, q \rangle \leq 0, \ \text{and} \ \langle Mz^- + q, z \rangle + \varepsilon \langle M^t z, e \rangle < 0. \end{cases}$$

If $M^t z \geq 0$, then $\langle M^t z, e \rangle > 0$ and if $M^t z \leq 0$, then $\langle M^t z, e \rangle < 0$ and $\langle M^t z, z^- \rangle \leq 0$. Now let $\varepsilon \to 0_+$. ∎

This proposition shows that M is PSBD when $\mathcal{F}$ is quasi-monotone (and, a fortiori, pseudomonotone) on $\mathbb{R}^n_+$. Next, we analyze the condition on the sign of $\langle z, q \rangle$.

THEOREM 3.1. *Assume that M is not PSD. Then $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ if and only if exactly one of the following conditions holds:*

(i) *$M = ab^t$, $b \neq ta$ for all $t > 0$, and either ($b \geq 0$ and $q = \lambda a - \mu b$ with $\lambda, \mu \geq 0$) or ($b \leq 0$ and $q = \mu b - \lambda a$ with $\lambda, \mu \geq 0$);*

(ii) *$\operatorname{rank}(M) \geq 2$, $q \in M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$, $\nu_-(M + M^t) = 1$, and $M^s \leq 0$. Let $\bar{x}$ be such that $q = M\bar{x}$. Then $\langle M^s \bar{x}, \bar{x} \rangle \leq 0$ and $M^s \bar{x} \leq 0$.*

Proof. It follows from Proposition 3.2 that a nonmonotone map $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ if and only if M is MPSBD and

$$\langle z, Mz \rangle \leq 0 \text{ and } M^t z \leq 0 \Rightarrow \langle z, q \rangle \leq 0.$$

The last condition is equivalent to $q \in T^o$. Then, the results follow from Theorem 2.2 and Propositions 2.4 and 2.5. ∎

Remark. Assume that $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M + M^t)(\mathbb{R}^n)$, $\nu_-(M + M^t) = 1$, and $\operatorname{rank}(M) \geq 2$. Let $\mathcal{F}(x) = M(x + \bar{x})$ and $\mathcal{G}(x) = M^t(x + \bar{x})$. Since $M^s = (M^t)^s$, then quasi monotonicity of $\mathcal{F}$ on $\mathbb{R}^n_+$ is equivalent to quasi monotonicity of $\mathcal{G}$ on $\mathbb{R}^n_+$.

Next, we return to pseudomonotonicity. The first result concerns the case where $\mathcal{F}$ is affine, but nonlinear.

THEOREM 3.2. *Assume that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ and $q \neq 0$. Then $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$.*

Proof. Assume, for contradiction, that $\mathcal{F}$ is not pseudomonotone on $\mathbb{R}^n_+$. Then M is MPSBD and there exists z such that

$$\langle z, Mz\rangle < 0, \quad M^t z \le 0, \quad \langle q, z\rangle \le 0, \quad \text{and} \quad \langle M^t z, z^-\rangle + \langle q, z\rangle = 0.$$

Hence

$$\langle M^t z, z^-\rangle = \langle q, z\rangle = 0.$$

On the other hand

$$z \in \operatorname{int}(T) \subseteq T \subseteq \{v : M^t v \le 0, \quad \langle q, v\rangle \le 0\}.$$

Thus $z + tq \in \operatorname{int}(T)$ for $t > 0$ small enough. Then

$$0 \ge \langle q, z + tq\rangle = t\,\|q\|^2 > 0,$$

a contradiction. ∎

We are left with the linear case. Following Gowda [7], we define a matrix M to be *pseudomonotone* if the map $\mathcal{F}(x) = Mx$ is pseudomonotone on the nonnegative orthant.

The following lemma gives a first characterization of pseudomonotone matrices.

LEMMA 3.1. *A matrix M is pseudomonotone if and only if M is PSBD and*

$$\langle z, Mz\rangle < 0, \quad M^t z \le 0 \quad \Rightarrow \quad Mz^- \ne 0.$$

Proof. Assume that M is pseudomonotone. Then Proposition 3.1 implies that M is PSBD and

$$\langle z, Mz\rangle < 0 \quad \text{and} \quad M^t z \le 0 \quad \Rightarrow \quad \langle z, Mz^-\rangle = \langle M^t z, z^-\rangle < 0.$$

Then $Mz^- = 0$ cannot occur.

Conversely, assume that M is PSBD and not pseudomonotone. Then z exists so that

$$\langle z, Mz\rangle < 0 \quad \text{and} \quad M^t z \le 0 \quad \Rightarrow \quad \langle z, Mz^-\rangle = \langle M^t z, z^-\rangle = 0.$$

For any $v \in \mathbb{R}^n$, there exists $t(v) > 0$ so that

$$\langle z + tv, M(z + tv)\rangle < 0 \quad \text{for all } t \in [-t(v), t(v)].$$

Hence, for such t

$$z + tv \in \operatorname{int}(T) = \{u : \langle Mu, u\rangle < 0, \quad M^t u \le 0\},$$

and therefore

$$0 \ge M^t z + tM^t v.$$

Since $z^- \ge 0$, for all $t \in [-t(v), t(v)]$, it holds that

$$0 \ge \langle M^t z, z^-\rangle + t\langle M^t v, z^-\rangle = t\langle v, Mz^-\rangle.$$

Hence $\langle v, Mz^-\rangle = 0$ for all $v \in \mathbb{R}^n$ and therefore $Mz^- = 0$. ∎

It is clear that a PSD matrix is always pseudomonotone. Also, a pseudomonotone matrix is necessarily copositive. Indeed $\langle M0, x-0\rangle = 0$ and $\mathcal{F}(x) = Mx$ pseudomonotone on $\mathbb{R}^n_+$ imply $\langle Mx, x-0\rangle \geq 0$ for all $x \geq 0$. Now, in view of the remarks above, we consider only matrices which are both MPSBD and copositive. We begin with matrices of rank 1.

PROPOSITION 3.3. *Assume that a and b are two linearly independent and nonnegative vectors. Then $M = ab^t$ is pseudomonotone if and only if*

$$(C) \qquad b_i = 0 \quad \Rightarrow \quad a_i = 0.$$

Proof. Assume that M is not pseudomonotone and condition (C) holds. Then some z exists such that

$$\langle z, Mz\rangle < 0, \quad M^t z \leq 0, \quad \text{and} \quad Mz^- = 0.$$

Hence

$$\langle b, z\rangle > 0, \quad \langle a, z\rangle < 0, \quad \text{and} \quad \langle b, z^-\rangle = 0.$$

Condition (C), together with $\langle b, z^-\rangle = 0$, implies that $\langle a, z^-\rangle = 0$ and therefore

$$0 \leq \langle a, z^+\rangle < \langle a, z^-\rangle = 0.$$

Conversely, assume that there is some index i such that $b_i = 0$ and $a_i > 0$. Since b is nonnull, there is j with $b_j > 0$. Take $t > 0$ so that $a_j - ta_i < 0$ and let $z \in \mathbb{R}^n$ with $z_i = -t$, $z_j = 1$, and $z_l = 0$ for all $l \neq i, j$. Then

$$\langle a, z\rangle\langle b, z\rangle = (a_j - ta_i)b_j < 0 \quad \text{and} \quad Mz^- = \langle b, z^-\rangle a = 0.$$

Hence M is not pseudomonotone. □

Next, the case of MPSBD matrices with rank greater than or equal to 2 is analyzed below.

PROPOSITION 3.4. *Assume that $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M+M^t)(\mathbb{R}^n)$ and $\nu_-(M+M^t) = 1$. Then M is pseudomonotone if and only if M is PSBD and copositive.*

Proof. We have only to prove the sufficiency. Assume that M is PSBD and copositive, but not pseudomonotone. In view of Lemma 3.1, there exists z such that

$$\langle z, Mz\rangle < 0, \quad M^t z \leq 0, \quad \text{and} \quad Mz^- = 0.$$

Since $M(\mathbb{R}^n) = M^t(\mathbb{R}^n)$, then $\mathrm{Ker}(M) = \mathrm{Ker}(M^t)$. Hence, $M^t z^- = 0$. Then

$$0 > \langle z, Mz\rangle = \langle z^+ - z^-, M(z^+ - z^-)\rangle = \langle z^+, Mz^+\rangle \geq 0. \qquad \square$$

All the above results are summarized in the following theorem.

THEOREM 3.3. *A matrix M is pseudomonotone if and only if it is PSBD and copositive with the additional condition in case $M = ab^t$ that ($a_i = 0$ whenever $b_i = 0$). In fact, the class of pseudomonotone matrices coincides with the class of matrices which are both PSBD and copositive star.*

Remark. Gowda [9] conjectured that pseudomonotonicity of M implies pseudomonotonicity of M^t. This is true when M is PSD. This is also true when $\mathrm{rank}(M) \geq 2$ in view of the above proposition and Proposition 2.5. However, it is no longer true for matrices of rank 1. Take $M = ab^t$ with $a = (1, 0)^t$ and $b = (1, 1)^t$. Then M and M^t are PSBD and copositive, M is pseudomonotone, but M^t is not.

4. The LCP. In this section, we consider the LCP

$$\text{(LCP)} \qquad \text{find } x \geq 0 \text{ so that } Mx+q \geq 0 \text{ and } \langle Mx+q, x\rangle = 0,$$

and the quadratic program associated with (LCP):

$$\text{(QP)} \qquad m = \inf[\, f(x) = \langle Mx+q, x\rangle : x \geq 0, Mx+q \geq 0].$$

We introduce the following sets:
$E = \{x : x \geq 0,\ Mx+q = 0\}$,
$F = \{x : x \geq 0,\ Mx+q \geq 0\}$,
$S = \{x : x \geq 0,\ Mx+q \geq 0,\ \langle x, Mx+q\rangle = 0\}$,
$Q = \{x : x \geq 0,\ Mx+q \geq 0,\ \langle x, Mx+q\rangle = m\}$.
Obviously $E \subseteq S \subseteq F$, m is nonnegative and $x \in S$ if and only if $x \in Q$ and $m = 0$.

F, S, and Q are called the *feasible* set of (LCP) (and also of (QP)), the *solution* set of (LCP), and the *optimal solution* set of (QP), respectively.

(LCP) and (QP) are said to be *feasible* if $F \neq \emptyset$ and (LCP) is said to be *solvable* if $S \neq \emptyset$. As before, we set $\mathcal{F}(x) = Mx+q$. We show the following.

PROPOSITION 4.1. (a) *(LCP) is feasible if and only if*

$$u \geq 0, \quad M^t u \leq 0 \quad \Rightarrow \quad \langle q, u\rangle \geq 0.$$

(b) *Assume that (LCP) is feasible. Then $E = F$ (hence $S = E = F$) if and only if there exists $u > 0$ such that $M^t u \leq 0$ and $\langle q, u\rangle = 0$.*

(c) *Assume that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ and M is not PSD. Then (LCP) is feasible if and only if*

$$u \geq 0, \quad M^t u \leq 0 \quad \Rightarrow \quad \langle q, u\rangle = 0.$$

(d) *Assume that (LCP) is feasible, $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$, and M is not PSD. Then $E = F$ (and hence $E = F = S$) if and only if there exists $u > 0$ such that $M^t u \leq 0$.*

Proof. (a) As shown in ([1, p. 111]), the result is a direct consequence of Farkas' lemma.

(b) Let $e = (1, \ldots, 1)^t \in \mathbb{R}^n$. Then $F \neq \emptyset$ and $Mx+q = 0$ for all $x \in F$ if and only if

$$0 = \sup_x \left[\langle Mx+q, e\rangle : x \geq 0, \quad -Mx \leq q\right].$$

Hence, by duality in linear programming again, if and only if

$$0 = \inf_y \left[\langle q, y+e\rangle : y \geq 0, \quad M^t(y+e) \leq 0\right].$$

Take $u = y+e$.

(c) It is easily seen from Proposition 3.2 that

$$u \geq 0, \quad M^t u \leq 0 \quad \Rightarrow \quad \langle q, u\rangle \leq 0.$$

Combine with (a).

(d) Combine (b) and (c). □

Now, we assume that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$, (LCP) is feasible, and $F \neq E$. The conjunction of these assumptions has strong implications, as shown below.

PROPOSITION 4.2. *Assume that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ and $x \geq 0$ exists so that $Mx + q \geq 0$, $Mx + q \neq 0$. Then M is copositive.*

Proof. Let $u > 0$ and $t > 0$. Then,

$$\langle Mx + q, x + tu - x\rangle > 0.$$

Hence, since $\mathcal{F}$ is quasi-monotone,

$$\langle M(x + tu) + q, x + tu - x\rangle = t\langle Mx + q, u\rangle + t^2\langle Mu, u\rangle \geq 0.$$

Let $t \to +\infty$. Then M is copositive on the positive orthant and, by continuity, on the nonnegative orthant as well. □

We continue the analysis of (LCP) in the special case where the matrix M has rank 1. In this case, the solution set can be completely described. Assume that $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$ and M is copositive. Then, without loss of generality, we consider the case $\mathcal{F}(x) = Mx + q$, where

$$(H) \qquad \left.\begin{array}{l} M = ab^t \quad \text{with } a, b \in \mathbb{R}^n_+,\ a, b \neq 0, \\ 0 \neq q = \lambda a - \mu b, \quad \lambda \geq 0, \mu \geq 0. \end{array}\right\}$$

PROPOSITION 4.3. *Assume that (H) holds. Then (LCP) is feasible if and only if either ($\mu = 0$) or ($a_i > 0$ whenever $b_i > 0$).*

Proof. According to Proposition 4.1, F is nonempty if and only if

$$y \geq 0, \quad \langle a, y\rangle b \leq 0 \quad \Rightarrow \quad \langle q, y\rangle \geq 0.$$

Hence, since $0 \neq b \geq 0$ and $\langle a, y\rangle \geq 0$ for all $y \geq 0$, F is nonempty if and only if

$$y \geq 0, \quad \langle a, y\rangle = 0 \quad \Rightarrow \quad \mu\langle b, y\rangle \leq 0.$$

On the other hand, $\mu\langle b, y\rangle \geq 0$ for all $y \geq 0$. The conclusion follows. □

THEOREM 4.1. *Assume that (H) holds and (LCP) is feasible. Let $\gamma = \max[b_i/a_i :$ i such that $a_i > 0]$. Then $S = \mathbb{R}^n_+ \cap \tilde{S}$, where*

$$x \in \tilde{S} \Leftrightarrow \begin{cases} \langle a, x\rangle = 0 & \text{if } \mu = 0, \\ \langle a, x\rangle = \langle b, x\rangle = 0 & \text{if } \gamma\mu \leq \lambda, \\ \gamma\langle a, x\rangle = \langle b, x\rangle = \gamma\mu - \lambda & \text{if } 0 \leq \lambda < \mu\gamma. \end{cases}$$

Proof. Note that for $\mu \neq 0$, γ is positive in view of Proposition 4.3. Set $\xi = \min\{b_i/a_i : i \text{ such that } a_i > 0\}$. From the definition, $x \in S$ if and only if

$$x \geq 0, \quad (\langle b, x\rangle + \lambda)a \geq \mu b \quad \text{and} \quad \langle a, x\rangle\langle b, x\rangle + \lambda\langle a, x\rangle - \mu\langle b, x\rangle = 0.$$

For $x \geq 0$, we analyze the following cases in succession.

(i) $\langle a, x\rangle = 0$. Then, by Proposition 4.3, $\mu\langle b, x\rangle = 0$. Hence, $x \in S$ if and only if either $\mu = 0$ or ($\mu\gamma \leq \lambda$ and $\langle b, x\rangle = 0$).

(ii) $\langle a, x\rangle \neq 0$ and $\langle b, x\rangle = 0$. Then, $x \in S$ if and only if $\mu = \lambda = 0$.

(iii) $\langle a, x\rangle \neq 0$ and $\langle b, x\rangle \neq 0$. Then, $x \in S$ if and only if $\mu \neq 0$ and

$$\langle b, x\rangle \geq \gamma\mu - \lambda \quad \text{and} \quad \frac{\mu}{\langle a, x\rangle} - \frac{\lambda}{\langle b, x\rangle} = 1.$$

For $r > 0$ such that $1 \geq r(\gamma\mu - \lambda)$, let us define

$$S_r = \left\{ x : x \geq 0, \quad \langle b, x\rangle = \frac{1}{r}, \quad \text{and} \quad \frac{\mu}{\langle a, x\rangle} - \frac{\lambda}{\langle b, x\rangle} = 1 \right\}.$$

Then $x \in S$ if and only if $x \in S_r$ for some r such that $1 \geq r(\gamma\mu - \lambda)$. Thus, we are led to analyze the nonemptyness of S_r.

By Farkas' lemma ([1, p. 109])

$$S_r = \left\{ x : x \geq 0, \langle b, x\rangle = \frac{1}{r}, \ \langle a, x\rangle = \frac{\mu}{1 + \lambda r} \right\}$$

is nonempty if and only if

$$y \in \mathbb{R}^2 \text{ and } by_1 + ay_2 \leq 0 \Rightarrow \frac{1}{r} y_1 + \frac{\mu}{1 + \lambda r} y_2 \leq 0.$$

If $y_2 > 0$, then only $y_1 < 0$ is to be considered, and the implication yields

$$\frac{1}{r} \geq (\xi\mu - \lambda).$$

For this recall that $b_i = 0$ when $a_i = 0$. If $y_2 < 0$, then the above implication yields

$$\frac{1}{r} \leq (\gamma\mu - \lambda).$$

Since, we are considering only real r such that $1 \geq r(\gamma\mu - \lambda)$, there is only one r such that $1 = r(\gamma\mu - \lambda)$.

Therefore in case (iii), $x \in S$ if and only if

$$x \geq 0 \quad \text{and} \quad \gamma\langle a, x\rangle = \langle b, x\rangle = \gamma\mu - \lambda > 0.$$

Summarizing, we get the expression for $\tilde{S}$. □

Now, we consider the case where $\operatorname{rank}(M) \geq 2$. We begin with the relationship between (LCP) and (QP). The quadratic function $f(x) = \langle Mx + q, x\rangle$ is convex if the map $F(x) = Mx + q$ is monotone. But f is not necessarily pseudoconvex on $\mathbb{R}^n_+$ if F is only pseudomonotone on $\mathbb{R}^n_+$, as shown below.

Example.

$$M = \begin{pmatrix} 1 & 2 \\ -1 & 0 \end{pmatrix}, \quad \bar{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \text{ and } q = M\bar{x} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

Then

$$M^s = \begin{pmatrix} -8 & -4 \\ -4 & 0 \end{pmatrix}, \quad M^s\bar{x} = \begin{pmatrix} -4 \\ -4 \end{pmatrix}, \text{ and } \langle M^s\bar{x}, \bar{x}\rangle = 0.$$

It follows that $F(x) = Mx + q$ is pseudomonotone on $\mathbb{R}^n_+$ while $f(x) = \langle Mx + q, x\rangle$ is not pseudoconvex on $\mathbb{R}^n_+$ since the matrix $(M + M^t)$ has a positive entry, in contradiction with Martos' characterization in [16].

Assume that $F \neq \emptyset$, i.e., (QP) is feasible. Since the optimal value m of the quadratic program (QP) is finite, (QP) has an optimal solution in view of the Frank–Wolfe theorem [6]. Also, an optimal solution of (QP) is a Karush–Kuhn–Tucker (KKT) point of the program. Denote by K the set of the KKT points of (QP). Then,

$$K = \left\{ x : \begin{array}{c} x \geq 0, Mx + q \geq 0, \text{ and there are } u \geq 0, v \geq 0 \text{ so that} \\ (M + M^t)x + q = u + M^t v, \ \langle x, u\rangle = \langle Mx + q, v\rangle = 0 \end{array} \right\}.$$

For $x \in K$, let us define

$$K(x) = \left\{ (u,v): \quad \begin{array}{c} u \geq 0, v \geq 0,\ \langle x,u\rangle = \langle Mx+q, v\rangle = 0, \\ \text{and } (M+M^t)x + q = u + M^t v \end{array} \right\}.$$

Two questions are of interest for an (LCP) [2]: "Do the set of the solutions of (LCP), the set of the optimal solutions of (QP), and the set of the KKT points of (QP) coincide?"; and "Is the set of the solutions of (LCP) (polyhedral) convex?" We will see below that both of these questions receive positive answers if the map $\mathcal{F}$ is pseudomonotone. We first present the following three lemmas.

LEMMA 4.1. *Assume that $0 \neq a \in \mathbb{R}^n$ and A is an $n \times n$ symmetric matrix which is not PSD.*

(i) *The condition*

$$\langle a,h\rangle = 0 \Longrightarrow \langle Ah,h\rangle \geq 0 \tag{4.1}$$

holds if and only if $a \in A(\mathbb{R}^n)$ and $\langle A^\dagger a, a\rangle \leq 0$.

(ii) *Assume that* (4.1) *holds and $\langle a,d\rangle = 0$. Then*

$$\langle Ad,d\rangle = 0 \Longleftrightarrow Ad = \lambda a \text{ for some } \lambda.$$

Proof. For the first statement, see Crouzeix and Ferland [3]; for the second one, see Crouzeix, Marcotte, and Zhu [4]. □

LEMMA 4.2. *Assume that $\mathcal{F}(x) = Mx + q$ is quasi-monotone on $\mathbb{R}^n_+$ and M is not PSD. Then the implication*

$$\langle Mx+q,h\rangle = 0 \Longrightarrow \langle Mh,h\rangle \geq 0$$

holds for all $x \geq 0$ such that $Mx + q \neq 0$.

Proof. The map is quasi-monotone (and pseudomonotone) on the positive orthant if and only if the condition holds for all $x > 0$ [5, 15]. We are left with the points on the boundary. Let $x \geq 0$ such that $Mx + q \neq 0$. Consider a sequence $\{x_k\}$ of positive vectors converging to x. Then, by Lemma 4.1, $Mx_k + q \in (M+M^t)(\mathbb{R}^n)$ and $\langle (M+M^t)^\dagger (Mx_k+q), Mx_k+q\rangle \leq 0$. Now we pass to the limit and apply Lemma 4.1 again. □

LEMMA 4.3. *Let M be an MPSBD matrix with* $\operatorname{rank}(M) \geq 2$. *Assume that $\langle M^s u,u\rangle = \langle M^s v,v\rangle = \langle M^s u,v\rangle = 0$. Then Mv and Mu are colinear.*

Proof. Set $B = M + M^t$. As in the proof of Proposition 2.3, we consider P invertible, $\triangle_2$ positive definite diagonal, and $\lambda_1 < 0$ such that

$$P^tP = I \quad \text{and} \quad P^tBP = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \triangle_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Set $x = P^tMu$ and $y = P^tMv$. The condition on the ranges of M, M^t, and $(M+M^t)$ imply that $x_3 = y_3 = 0$. Then

$$\lambda_1^{-1}x_1^2 + \langle \triangle_2^{-1}x_2, x_2\rangle = \lambda_1^{-1}y_1^2 + \langle \triangle_2^{-1}y_2, y_2\rangle = 0.$$

Since $\triangle_2$ is positive definite, $x = 0$ ($y = 0$) if and only if $x_1 = 0$ ($y_1 = 0$). Hence, without loss of generality, we assume that $x_1 = y_1 = 1$. For $t \in [0,1]$, set $w = u + t(v-u)$ and $z = P^tMw$. Then $z_1 = 1$, $\langle M^s w,w\rangle = 0$, and

$$\lambda_1^{-1} + \langle \triangle_2^{-1}z_2, z_2\rangle = \lambda_1^{-1} + \langle \triangle_2^{-1}(x_2 + t(y_2-x_2)), x_2 + t(y_2-x_2)\rangle = 0.$$

Since the equality is true for all $t \in [0,1]$ and $\triangle_2$ is positive definite, we have $y_2 = x_2$. The conclusion follows. ∎

THEOREM 4.2. *Assume that $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$, $\operatorname{rank}(M) \geq 2$, and (LCP) is feasible. Then $S = K = Q \neq \emptyset$.*

Proof. (a) Let $x \in K$ and $u, v \in K(x)$. Then

$$Mx + q = u + M^t(v - x). \tag{4.2}$$

Hence,

$$\langle Mx + q, v - x\rangle = \langle u, v - x\rangle + \langle M^t(v - x), (v - x)\rangle,$$

$$\langle Mx + q, x\rangle + \langle u, v\rangle + \langle M(v - x), (v - x)\rangle = 0. \tag{4.3}$$

Equation (4.2) implies also that

$$Mv + q = u + (M + M^t)(v - x), \tag{4.4}$$

$$\langle Mv + q, v\rangle = \langle u, v\rangle + \langle (M + M^t)(v - x), v\rangle. \tag{4.5}$$

Again in view of (4.2), we have

$$\begin{aligned} \langle Mx + q, x\rangle &= \langle u, x\rangle + \langle M^t(v - x), x\rangle, \\ \langle Mx + q, x\rangle &= \langle Mx + q, v\rangle - \langle Mx + q, x\rangle + \langle q, x - v\rangle, \\ 2\langle Mx + q, x\rangle &= \langle q, x - v\rangle. \end{aligned} \tag{4.6}$$

(b) First, we consider the case where M is PSD. Then all three quantities in (4.3) are nonnegative; hence they are null and $x \in S$.

(c) Next, we consider the case where M is MPSBD. Assume, for contradiction, that $x \notin S$. Then, (4.3) and (4.6) imply that both quantities $\langle M(v - x), v - x\rangle$ and $\langle q, v - x\rangle$ are negative. Since $\mathcal{F}$ is pseudomonotone, Proposition 3.1 implies that $M^t(v - x) \leq 0$. Hence (4.2) implies

$$0 < \langle Mx + q, x\rangle = \langle M^t(v - x), x\rangle \leq 0,$$

a contradiction. ∎

THEOREM 4.3. *Assume that $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$, (LCP) is feasible, and $\operatorname{rank}(M) \geq 2$. Then S is a polyhedral convex set. In particular, let $x \in S$ be given such that $Mx + q \neq 0$. Then*

$$S = \left\{ y : \begin{array}{c} y \geq 0, My + q \geq 0, \langle Mx + q, y\rangle = \langle q, y - x\rangle = 0, \\ (M + M^t)(y - x) = \lambda(Mx + q) \text{ for some } \lambda \end{array} \right\}.$$

Furthermore, if $\lambda \neq 0$, then M is MPSBD, $\langle (M + M^t)^\dagger q, q\rangle = 0$, and Mx, My, and q are colinear.

Proof. (i) Assume that $x, y \in S$. Then $\langle Mx + q, y - x\rangle = \langle Mx + q, y\rangle \geq 0$ and $\langle My + q, x - y\rangle = \langle My + q, x\rangle \geq 0$. Since $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$, it follows that $\langle My + q, y - x\rangle \geq 0$ and $\langle Mx + q, x - y\rangle \geq 0$. Hence,

$$\langle Mx + q, y - x\rangle = \langle My + q, y - x\rangle = 0.$$

It follows that

$$\langle M(y-x),(y-x)\rangle = 0.$$

First, assume that M is PSD. Then $(M+M^t)(y-x) = 0 = 0(Mx+q)$.

Next, assume that M is MPSBD. Then Lemmas 4.1 and 4.2 ensure the existence of λ and μ such that

$$(M+M^t)(y-x) = \lambda(Mx+q) = \mu(My+q). \tag{4.7}$$

Obviously, (4.7) holds if M is PSD.

Furthermore, the equalities

$$\langle Mx+q,x\rangle = \langle Mx+q,y\rangle = \langle My+q,x\rangle = \langle My+q,y\rangle = 0$$

imply that for all t,

$$r(t) = f(x+t(y-x)) = \langle (1-t)(Mx+q)+t(My+q),(1-t)x+ty\rangle = 0.$$

Hence

$$0 = r'(0) = \langle (M+M^t)x+q,y-x\rangle = \langle x,(M+M^t)(y-x)\rangle + \langle q,y-x\rangle.$$

Then (4.7) yields

$$0 = \lambda\langle x,Mx+q\rangle + \langle q,y-x\rangle = \langle q,y-x\rangle.$$

We have proved that

$$S \subseteq T = \left\{ y: \begin{array}{c} y\geq 0, My+q\geq 0, \langle Mx+q,y\rangle = \langle q,y-x\rangle = 0, \\ (M+M^t)(y-x) = \lambda(Mx+q) \text{ for some } \lambda \end{array} \right\}.$$

Next, assume that $(M+M^t)(y-x)\neq 0$. Then M is MPSBD and λ and μ are nonzero. If $q=0$, then Mx, My, and q are obviously colinear in view of (4.7). Assume that $q\neq 0$. We know by Theorem 3.1 that $\bar{x}$ exists so that $M\bar{x}=q$, $\langle M^s\bar{x},\bar{x}\rangle \leq 0$, and $M^s\bar{x}\leq 0$. Since $M(\mathbb{R}^n) = M^t(\mathbb{R}^n) = (M+M^t)(\mathbb{R}^n)$, there is $w\in \mathrm{Ker}(M) = \mathrm{Ker}(M^t) = \mathrm{Ker}((M+M^t))$ such that

$$y-x = \lambda(M+M^t)^\dagger M(x+\bar{x})+w.$$

Then

$$0 = 2\langle q,y-x\rangle = 2\langle \bar{x},M^t(y-x)\rangle = \lambda\langle \bar{x},M^s(x+\bar{x})\rangle = \lambda(\langle M^s\bar{x},x\rangle + \langle \bar{x},M^s\bar{x}\rangle).$$

Recall that $\lambda\neq 0$, M^s is symmetric, $\langle M^s\bar{x},\bar{x}\rangle\leq 0$, $x\geq 0$, and $M^s\bar{x}\leq 0$. Hence $0 = \langle M^s\bar{x},x\rangle = \langle M^s\bar{x},\bar{x}\rangle = 2\langle (M+M^t)^\dagger q,q\rangle$. On an other hand,

$$0 = 2\langle Mx+q,y-x\rangle = 2\langle x,M^t(y-x)\rangle = \lambda\langle M^s(x+\bar{x}),x\rangle.$$

It follows that $\langle M^sx,x\rangle = 0$ and, by Lemma 4.3, that Mx and $q=M\bar{x}$ are colinear. In the same manner, it can be proved that My and q are colinear too.

(ii) Now we prove that $T\supseteq S$. Let $y\in T$. Then

$$f(y) = \langle My+q,y\rangle = f(x) + \langle (M+M^t)x+q,y-x\rangle + \frac{1}{2}\langle (M+M^t)(y-x),(y-x)\rangle.$$

$f(x) = 0$ since $x \in S$. $\langle Mx + q, y - x\rangle = 0$ since $x, y \in T$. Hence, the second and the third terms are also zero and $y \in S$.

(iii) It is clear that T is a polyhedral convex set. □

We give an example for the case where λ is not zero.

$$M = \begin{pmatrix} 0 & 11 \\ -1 & 0 \end{pmatrix},\ M^s = -\frac{11}{5}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ q = \begin{pmatrix} 11 \\ 0 \end{pmatrix}.$$

Then $S = \{0\} \times [0, \infty)$.

In the special case where M is PSD, the characterization of S in Theorem 4.3 is given already in ([1, p. 142]). From [2] it is known that convexity of the set of solutions of (LCP) implies polyhedrality of this set. Theorem 4.3 gives an explicit characterization of the set if one solution is known.

Gowda [8] proved that feasibility implies solvability if $\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$. Theorem 3.2 shows that this is the case if $q \neq 0$ and $\mathcal{F}$ is quasi-monotone on $\mathbb{R}^n_+$.

5. Concluding remarks. Positive subdefiniteness appears to be the good concept to characterize generalized monotone affine maps on the nonnegative orthant as well as pseudomonotone matrices. It leads to new characterizations that recover all previous ones and are more simple.

When M is symmetric, then $M^s = M$ and Theorem 3.1 corresponds to the characterization of quasi-convex quadratic functions on the nonnegative orthant first given by Martos [16, 17]. Pseudomonotone maps on $\mathbb{R}^n_+$ and pseudomonotone matrices have been studied by Gowda [7, 8, 9]. Gowda's characterizations are related to our Proposition 3.1. For instance, Gowda's characterization of pseudomonotonicity of affine maps is as follows [8, Theorem 2].

PROPOSITION 5.1. *$\mathcal{F}$ is pseudomonotone on $\mathbb{R}^n_+$ if and only if*

(a) $$z \in \mathbb{R}^n, \quad \langle z, Mz\rangle < 0 \Rightarrow \begin{cases} M^t z \geq 0 \text{ and } \langle z, Mz^- + q\rangle \geq 0, \\ \quad \text{or} \\ M^t z \leq 0 \text{ and } \langle z, Mz^- + q\rangle \leq 0; \end{cases}$$

(b) $\langle z, Mz^- + q\rangle \geq 0 \Rightarrow \langle z, Mz^+ + q\rangle \geq 0$.

Crouzeix and Schaible [5] have derived characterizations of generalized monotone affine maps on a convex set from the first-order conditions of pseudomonotonicity on an open convex set [15]. When specialized to the nonnegative orthant, their result is as follows.

PROPOSITION 5.2. *F is quasi-monotone on $\mathbb{R}^n_+$ (and pseudomonotone on $int(\mathbb{R}^n_+)$) if and only if one of the following conditions holds:*

(i) *M is PSD;*

(ii) *M has rank 1, $q \in (M + M^t)(\mathbb{R}^n) \supseteq M(\mathbb{R}^n)$, $\langle q, (M + M^t)^\dagger q\rangle \leq 0$, and $M^t(M + M^t)^\dagger q \leq 0$;*

(iii) *$\nu_-(M + M^t) = 1$, $(M + M^t)(\mathbb{R}^n) = M(\mathbb{R}^n)$, M^s is conegative, there is $\bar{x}$ so that $M\bar{x} = q$, and either ($\mathbb{R}^n_+ \subseteq W$ and $\bar{x} \in W$) or ($\mathbb{R}^n_+ \subseteq -W$ and $\bar{x} \in -W$), where W is a closed convex cone such that $W \cup -W = \{w : \langle w, M^s w\rangle \leq 0\}$.*

Finally, the different results in this paper show that nonsymmetric matrices of rank 1 require a separate treatment.

Acknowledgments. The authors would like to thank the referees for their suggestions which helped to improve the presentation of our results. The fourth author gratefully acknowledges the hospitality and the support of Université Blaise Pascal for his visit in June 1997.

REFERENCES

[1] R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[2] 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.

[3] J.-P. Crouzeix and J. A. Ferland, *Criteria for quasiconvexity and pseudoconvexity: Relationships and comparisons*, Math. Programming, 23 (1982), pp. 193–205.

[4] J.-P. Crouzeix, P. Marcotte, and D. Zhu, *Conditions Ensuring the Applicability of Cutting Plane Methods for Solving Variational Inequalities*, Technical report, Université de Montréal, Montreal, Canada, 1997.

[5] J.-P. Crouzeix and S. Schaible, *Generalized monotone affine maps*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 992–997.

[6] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist. Quart., 3 (1956), pp. 95–110.

[7] M. S. Gowda, *Pseudomonotone and copositive star matrices*, Linear Algebra Appl., 113 (1989), pp. 107–118.

[8] M. S. Gowda, *Affine pseudomonotone mappings and the linear complementarity problem*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 373–380.

[9] M. S. Gowda, *On the transpose of a pseudomonotone matrix and the LCP*, Linear Algebra Appl., 140 (1990), pp. 129–137.

[10] N. Hadjisavvas and S. Schaible, *Quasimonotonicity and pseudomonotonicity in variational inequalities and equilibrium problems*, in Generalized Convexity, Generalized Monotonicity, Nonconvex Optim. Appl. 27, J.-P. Crouzeix, J.-E. Martinez-Legaz, and M. Volle, eds., Kluwer Academic Publishers, Dordrecht, Boston, London, 1998, pp. 257–275.

[11] A. Hassouni, *Sous-differentiels des fonctions quasiconvexes*, Thèse de 3$^{\text{ème}}$ cycle de l'Université Paul Sabatier, Toulouse, France, 1983.

[12] S. Karamardian, *The complementarity problem*, Math. Programming, 2 (1972), pp. 107–129.

[13] S. Karamardian, *Complementarity problems over cones with monotone and pseudomonotone maps*, J. Optim. Theory Appl., 18 (1976), pp. 445–454.

[14] S. Karamardian and S. Schaible, *Seven kinds of monotone maps*, J. Optim. Theory Appl., 66 (1990), pp. 37–46.

[15] S. Karamardian, S. Schaible, and J.-P. Crouzeix, *Characterizations of generalized monotone maps*, J. Optim. Theory Appl., 76 (1993), pp. 399–413.

[16] B. Martos, *Subdefinite matrices and quadratic forms*, SIAM J. Appl. Math., 17 (1969), pp. 1215–1223.

[17] B. Martos, *Nonlinear Programming: Theory and Methods*, North–Holland, Amsterdam, 1975.

[18] G. J. Minty, *On the monotonicity of the gradient of a convex function*, Pacific J. Math., 14 (1964), pp. 243–247.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 86–105

HARTLEY TRANSFORM REPRESENTATIONS OF SYMMETRIC TOEPLITZ MATRIX INVERSES WITH APPLICATION TO FAST MATRIX-VECTOR MULTIPLICATION*

GEORG HEINIG† AND KARLA ROST‡

Abstract. Representations for inverses of real symmetric Toeplitz matrices involving discrete Hartley transformations are presented which can be used for fast matrix-vector multiplication. In this way, multiplication of a column vector by an inverse real symmetric Toeplitz matrix can be carried out with the help of six Hartley transformations plus two for preprocessing. Besides complexity, stability issues will also be discussed. In particular, it is shown that, for positive definite Toeplitz matrices, the relative error of the computed vector can be estimated by a multiple of the condition number of the matrix.

Key words. Toeplitz matrix, Bezoutian, Hartley transform, fast algorithm, stability, complexity

AMS subject classifications. 47B35, 15A09, 15A23

PII. S089547989833961X

1. Introduction.

This paper is the third, after [21] and [22], in a series dedicated to representations of Toeplitz-like matrices with the help of trigonometric transformations that can be used for fast matrix-vector multiplication. In the first paper real Toeplitz and Toeplitz-plus-Hankel matrices were considered; inverses of complex Toeplitz and Toeplitz-plus-Hankel matrices were considered in the second. The representations in [22] involve only complex DFTs. Since the DFT for real data and related real transformations requires only half (or even less) of the amount of a complex DFT it is natural to ask whether in the case of a real matrix real transformations can be used. In the present paper we discuss this for inverses of $n \times n$ real symmetric Toeplitz matrices $T_n = [\, t_{|i-j|} \,]_{i,j=0}^{n-1}$ and discrete Hartley transformations. Other trigonometric transformations like sine and cosine transformations and inverses of Toeplitz-plus-Hankel matrices will be considered in a forthcoming paper.

Let us briefly describe the history of the problem and mention some related work. The starting point of the investigation is the Gohberg–Semencul formula [15] that represents the inverse of a Toeplitz matrix with the help of triangular Toeplitz matrices. The original formula is valid only under some conditions but there are modifications working for an arbitrary nonsingular Toeplitz matrix (see [20]). Since multiplication by triangular Toeplitz matrices can be carried out with fast Fourier transforms (FFTs), matrix-vector multiplication by inverses of $n \times n$ Toeplitz matrices can be done with $O(n \log n)$ complexity. For matrix-vector multiplication with the help of the Gohberg–Semencul formula, six DFTs of length $2n$ plus two more DFTs of this length for preprocessing are required.

It was observed in [2] (see also [1], [13]) that there are also formulas involving circulant matrices rather than triangular Toeplitz matrices. These formulas are more

*Received by the editors June 1, 1998; accepted for publication (in revised form) by D. Calvetti September 29, 1999; published electronically May 31, 2000. This work was partly supported by research project SM-161, Kuwait University.

http://www.siam.org/journals/simax/22-1/33961.html

†Kuwait University, Department of Mathematics and Computer Science, P.O. Box 5969, Safat 13060, Kuwait (georg@mcs.sci.kuniv.edu.kw).

‡Technische Universität Chemnitz, Fakultät für Mathematik, D-09107 Chemnitz, Germany (krost@mathematik.tu-chemnitz.de).

efficient because they require only the computation of DFTs with length n rather than $2n$. Note that a circulant formula is already contained in [29]. More formulas involving circulants can be found in [13], [30], [4], [18].

In the series of papers [11], [6], [7], [12], and [5] matrix algebras generated by a Hessenberg matrix were investigated. For special choices of the Hessenberg matrix these algebras are associated with discrete trigonometric transformations. The main results concern representations of matrices with a displacement structure, in particular inverses of Toeplitz matrices, with the help of matrices from these algebras. These representations can be used then for fast matrix-vector multiplication. Besides formulas matching the best known results for complex matrices, the papers [11], [7], and [5] contain formulas for inverse real symmetric Toeplitz matrices involving cosine transforms which admit matrix-vector multiplication by six transformations of length n plus four for preprocessing.

The problem of fast matrix-vector multiplication by Toeplitz-like matrices was also discussed in [13], [14], and [25]. The second of these references is especially worth mentioning because it contains a general approach to trigonometric transform–based representations of Toeplitz-like matrices using the displacement structure and transformation into Cauchy matrices. The latter was proposed in [17].

The formulas designed in the present paper can also be used for matrix-vector multiplication with six transformations, namely, Hartley transformations, but only two are required for preprocessing. Note that the Hartley transformation does not fit into the framework of Hessenberg algebras since the corresponding generating matrix is equal to the sum of the cyclic shift and its transpose, which is not Hessenberg. Hartley transformation–based formulas are also not considered in [25].

Moreover, let us note that in [23] it is shown that matrix-vector multiplication by inverses of general Toeplitz-plus-Hankel matrices can also be carried out with the help of six DHTs only. However, the number of transformations in the preprocessing phase will be greater, namely, eight in this case.

Our approach differs from those of the above-mentioned papers. We believe that it is easier and more straightforward. Furthermore, no additional conditions on the matrix are required, as they are in the other papers. It is based on the fact that inverses of Toeplitz matrices are Bezoutians. For positive definite Toeplitz matrices this follows from Szegö's Christoffel–Darboux-type formula for orthogonal polynomials on the unit circle. For the general case this was observed in [28]. Let us note that the fact that a matrix is a Bezoutian is actually stronger than the fact that it has displacement rank 2. Therefore, it can be expected that the Bezoutian approach provides a deeper insight and stronger results than the displacement approach in [17] and [25].

Let us explain the contents of the paper. In section 2 we quote a well-known inversion formula for Toeplitz matrices and specify it for the symmetric case by showing that the inverse of a symmetric Toeplitz matrix is the Bezoutian of a symmetric and a skewsymmetric polynomial and how these polynomials can be characterized as solutions of special equations. This result might be new in the presented form. We also show that any symmetric Bezoutian, including a singular one, is the Bezoutian of a symmetric and a skewsymmetric polynomial.

The definition of the DHT will be recalled in section 3. Concerning properties of this transformation and fast algorithms for its computation, we refer to [3], [9], [34], [31]. It is convenient to introduce some modifications of the classical DHT. According to a suggestion of M. Tasche, we denote them by DHT-II, DHT-III, and DHT-IV.

The main results of the paper, which are representations of Toeplitz Bezoutians involving only Hartley and diagonal matrices, are contained in sections 4 and 5. We present four different formulas. The first two, which will be presented in section 4, are convenient if the order n of the matrix is odd. The general structures of them are

$$B = \mathcal{H}^I(D_1\mathcal{H}^{II}\mathcal{H}^{IV}D_2 + D_3\mathcal{H}^{II}\mathcal{H}^{IV}D_4)\mathcal{H}^{III}$$

and

$$B = \mathcal{H}^I(D_5\mathcal{H}^{II}W\mathcal{H}^{III}D_6 + D_7\mathcal{H}^{II}W\mathcal{H}^{III}D_8 + D_9)\mathcal{H}^I,$$

where $\mathcal{H}^X$ $(X = I, II, III, IV)$ denote the Hartley transforms defined in section 3, D_i $(i = 1, \dots, 9)$ are diagonal matrices, and W is defined by (4.7). The inverse of the Toeplitz matrix can be identified as a restriction of B. The third and fourth formulas are convenient if the order of the Toeplitz matrix is even. The structure of the formulas is similar, only other types of Hartley transforms appear.

With the help of these representations, matrix-vector multiplication by an inverse of a Toeplitz matrix can be carried out with six Hartley transformations of length N for any $N > n$ if n is odd and $N > n + 1$ if n is even and some Hadamard products of vectors. For preprocessing, only two Hartley transformations are required. The enlargement of n is convenient in order to have all transformations of the same length.

Of course, the relevance of an algorithm in finite arithmetic depends not only on its complexity but also on its stability. In section 6 we discuss this problem for the algorithms emerging from the representation formulas. So far, very little can be found in the literature concerning this topic. There are some remarks in [8] pointing out that the Gohberg–Semencul formula might lead to an unstable algorithm. In [16] some positive results concerning stability of the Gohberg–Semencul and related formulas are obtained. However, in this paper stability is understood in the rather weak sense that the forward error cannot be *arbitrarily* large.

In the present paper we show that the relative error of the computed vector can be estimated by the condition numbers of the matrix itself and its $(n-1) \times (n-1)$ principal section or an $(n+1) \times (n+1)$ extension of it. In the case of a positive definite Toeplitz matrix T_n we obtain, comparing the exact vector $\xi = T_n^{-1}b$ with the computed vector $\widetilde{\xi}$, the estimation

$$\frac{\|\widetilde{\xi} - \xi\|}{\|\xi\|} \le 48(n+2)\kappa(T_n)(\mathbf{u} + O(\mathbf{u}^2)),$$

where $\kappa(T_n)$ is the condition number of T_n and $\mathbf{u}$ is the machine precision.

That means that the algorithm emerging from one of the formulas is forward stable in the sense of [24, p. 142], at least for positive definite Toeplitz matrices. As far as we know, this is the first time that stability in this sense is shown for an inversion formula for Toeplitz matrices. A more complete analysis for complex inversion formulas is contained in [19]. Section 7 is devoted to a more detailed investigation of the complexity of fast matrix-vector multiplication.

Let us agree upon some notations that will be used throughout the paper. We denote by I_n the identity of order n and by J_n and J'_n the following permutation matrices of order n:

$$J_n = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \quad \text{and} \quad J'_n = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & J_{n-1} \end{bmatrix}.$$

2. Toeplitz matrix inverses and Bezoutians. It is convenient to introduce the concept of a Bezoutian in terms of generating functions. Let $\ell(\lambda)$ denote the column vector $\ell(\lambda) = [\,\lambda^k\,]_0^{n-1}$, $\lambda \in \mathbb{C}$. The generating function of a (column) vector $x \in \mathbb{C}^n$ is, by definition, the polynomial $x(\lambda) = \ell(\lambda)^T x$. In case of an $n \times n$ matrix A the generating function is the bivariate polynomial

$$A(\lambda, \mu) = \ell(\lambda)^T A\, \ell(\mu).$$

DEFINITION 2.1. *Suppose that $a, b \in \mathbb{C}^{n+1}$. The* Toeplitz Bezoutian *or* T-Bezoutian *of the polynomials $a(\lambda)$ and $b(\lambda)$ or of the corresponding coefficient vectors a and b is the $n \times n$ matrix $B = \mathrm{Bez}(a, b)$ with the generating function*

$$B(\lambda, \mu) = \frac{a(\lambda)(J_{n+1}b)(\mu) - b(\lambda)(J_{n+1}a)(\mu)}{1 - \lambda\,\mu}\,. \tag{2.1}$$

Besides Toeplitz Bezoutians, there exist also Hankel, Toeplitz-plus-Hankel, and more general Bezoutians (see [20], [22], [27]).

The basis of our approach is the following.

THEOREM 2.2.[1] *Let $T_n = [t_{i-j}]_0^{n-1}$ be a nonsingular Toeplitz matrix and let z and x be the solutions of*

$$T_n\, z = [\,t_{-n}\; t_{1-n}\; t_{2-n}\; \cdots\; t_{-1}\,]^T, \quad T_n\, x = [\,1\; 0\; \cdots\; 0\,]^T, \tag{2.2}$$

respectively, where t_{-n} is an arbitrary but fixed number. Then the inverse of T_n is given by

$$T_n^{-1} = \mathrm{Bez}(x, y)\,, \tag{2.3}$$

where $y(\lambda) = \lambda^n - z(\lambda)$.

In what follows we shall not distinguish between the solution vector $x \in \mathbb{C}^n$ and the vector $\left[\begin{smallmatrix} x \\ 0 \end{smallmatrix}\right] \in \mathbb{C}^{n+1}$.

Let us note that the converse is also true: The inverse of a nonsingular T-Bezoutian is Toeplitz (see [20], [28]). Let us also mention the fact that $\{x, y\}$ forms a basis of the kernel of the $(n-1) \times (n+1)$ matrix $\widetilde{T}_n = [t_{i-j}]_{i=1,\,j=0}^{n-1,\;n}$.

In this paper we shall not discuss how to obtain the solutions x and z. Let us only mention that they can be computed with $O(n^2)$ complexity using classical Levinson-type or Schur-type algorithms or with "superfast solvers" with $O(n \log^2 n)$ complexity (see, for example, [4], [20], [26]).

We are going to discuss now the specifics of symmetric Toeplitz matrices. For this let us agree upon some concepts.

DEFINITION 2.3. *A vector $x \in \mathbb{C}^n$ (as well as the corresponding polynomial $x(\lambda) = \ell(\lambda)^T x$) is called* symmetric, skewsymmetric, even, odd, *respectively, if*

$$J_n x = x, \quad J_n x = -x, \quad J_n' x = x, \quad J_n' x = -x.$$

An $n \times n$ matrix A will be called centro-symmetric, centro-skewsymmetric, centro-even, centro-odd, *respectively, if*

$$A\, J_n = J_n A, \quad A\, J_n = -J_n A, \quad A\, J_n' = J_n' A, \quad A\, J_n' = -J_n' A.$$

A is called persymmetric *if $J_n A = A^T J_n$.*

[1] See [20, Theorem 1.1 and formula (1.47)].

For example, Toeplitz matrices and T-Bezoutians are persymmetric. Consequently, symmetric Toeplitz matrices and T-Bezoutians are centro-symmetric. A matrix of the form

$$\begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix},$$

where B is centro-symmetric, is centro-even.

Assume now that T_n is symmetric, i.e., $t_{-i} = t_i$ $(i = 1, 2, \dots, n-1)$. We also set $t_n = t_{-n}$. For our further considerations it is important to have only symmetric and skewsymmetric vectors in the representation of T_n^{-1}.

But first we show that in the symmetric case the inverse of T_n can be represented with the help of the solution of only one of the equations of (2.2). In this concern we have to consider the cases $x_0 \neq 0$ and $x_0 = 0$, where x_0 denotes the first component of the solution vector x. According to Cramer's rule, $x_0 \neq 0$ if and only if the matrix T_{n-1} is nonsingular.

We consider first the case $x_0 \neq 0$. In this case we have

$$y = \frac{1}{x_0} J_{n+1} x + \alpha x$$

for some number α. Hence

$$T_n^{-1} = \frac{1}{x_0} \operatorname{Bez}(x, J_{n+1}x). \tag{2.4}$$

This is just the Gohberg–Semencul formula for the symmetric case.

Now we discuss how to represent T_n^{-1} with the help of the solution of the first equation in (2.2). This will cover also the case $x_0 = 0$. We consider the $(n+1)\times(n+1)$ matrix $T_{n+1}(t) = [\, t_{i-j} \,]_0^n$ with $t_n = t$. This matrix is nonsingular if the Schur complement of T_n is nonzero. This Schur complement is equal to $-s(t)$, where

$$s(t) = x_0 t^2 + 2qt + r, \tag{2.5}$$

with $q = g^T T_n^{-1} e_1 = g^T x$, $r = g^T T_n^{-1} g - t_0$ and $g = [\, 0 \; t_{n-1} \cdots t_1 \,]^T$, $e_1 = [\, 1 \; 0 \; \cdots \; 0 \,]^T$.

If $x_0 = 0$, then $q \neq 0$. In fact, if we had $x_0 = q = 0$, then the backward-shifted vector $[\, x_1 \cdots x_{n-1} \; 0 \,]^T$ would belong to the kernel of T_n. That means that $T_{n+1}(t)$ is singular for one or two values of t, which are just the zeros of (2.5).

If z is a solution of the first equation of (2.2) and $y = [y_i]_0^n = [\, -z^T \; 1 \,]^T$, then

$$T_{n+1}(t_n) y = \alpha \, [\, 0 \; \cdots \; 0 \; 1 \,]^T,$$

where $\alpha = \sum_{k=0}^{n} t_{n-k} y_k$. Clearly, $\alpha \neq 0$ if $s(t_n) \neq 0$. Thus, in this case the vector y is neither symmetric nor skewsymmetric, since otherwise we would obtain $T_{n+1}(t_n)y = 0$. This means that the vectors y and $J_{n+1}y$, which both belong to the kernel of $\widetilde{T_n}$, are linearly independent. Since also x belongs to this kernel and the kernel is two-dimensional, there are numbers β and γ such that

$$x = \beta y - \gamma J_{n+1} y.$$

The coefficients are given by

$$\gamma = -1/\alpha = -\left(\sum_{k=0}^{n} t_k y_{n-k}\right)^{-1}, \quad \beta = \gamma y_0. \tag{2.6}$$

Hence the following is true.

PROPOSITION 2.4. *For all values of t_n with the exception of at most two, the inverse of T_n can be represented as*

$$T_n^{-1} = \gamma \operatorname{Bez}(y, J_{n+1}y), \tag{2.7}$$

where γ is given by (2.6).

From (2.4) and (2.7) we conclude the following.

THEOREM 2.5. *The inverse of a symmetric Toeplitz matrix T_n can be represented in the form*

$$T_n^{-1} = \gamma \operatorname{Bez}(a_-, a_+), \tag{2.8}$$

where a_+ is symmetric and a_- is skewsymmetric. In particular, these vectors and the factor γ are given by the following:

1. *If the first component x_0 of the solution of the second equation of* (2.2) *is nonzero, then $a_\pm = x \pm J_{n+1}x$ and $\gamma = 1/x_0$.*
2. *If t_n is chosen such that $s(t_n) \neq 0$, where $s(t)$ is given by* (2.5), *then $a_\pm = y \pm J_{n+1}y$, where $y = [\,-z^T\ 1\,]^T$ and z is the solution of the first equation of* (2.2) *and γ is given by* (2.6).

Proof. We have $x = (a_+ + a_-)/2$ and $J_{n+1}x = (a_+ - a_-)/2$. Inserting this into (2.4) we obtain the assertion for the first case. The second case is analogous. □

The theorem claims that any nonsingular symmetric T-Bezoutian is the Bezoutian of a symmetric and a skewsymmetric vector, since the inverse of a T-Bezoutian is Toeplitz. We show that this is also true for singular Bezoutians, even for a more general Bezoutian concept. A matrix B will be called *generalized T-Bezoutian* if the matrix with the generating function $(1 - \lambda\mu)B(\lambda, \mu)$ has rank less than or equal to 2. In [20, Part I, section 2.3], it is shown that any square generalized Bezoutian is actually the (classical) T-Bezoutian of two polynomials. Moreover, the following is true.

PROPOSITION 2.6. *If B is a symmetric Bezoutian, then there are a symmetric vector a_+ and a skewsymmetric one a_- such that $B = \operatorname{Bez}(a_-, a_+)$.*

Proof. Suppose that $B = \operatorname{Bez}(a, b)$ and $R = a\widehat{b}^T - b\widehat{a}^T$, where $\widehat{u}$ means $J_{n+1}u$. Then R has the generating function $(1 - \lambda\mu)B(\lambda, \mu)$. The range of R is spanned by a and b. First we observe that these vectors cannot be both symmetric or skewsymmetric. In fact, assuming that a and b are both symmetric or skewsymmetric, then $R^T = -R$, but R is symmetric. If one of the vectors a, b is symmetric and the other one is skewsymmetric, then nothing has to be proved. Assume now that one of the vectors, say, a, is neither symmetric nor skewsymmetric. Then a and $\widehat{a}$ are linearly independent and, since R is centro-symmetric, $\widehat{a}$ belongs to the range of R. Hence there are numbers α and β such that $b = \alpha a + \beta\widehat{a}$. We obtain that $R = \beta(aa^T - \widehat{a}\widehat{a}^T)$. Thus for a multiple c of a or $\widehat{a}$ we have $B = \operatorname{Bez}(c, \widehat{c})$. Setting $a_\pm = \frac{1}{2}(c \pm \widehat{c})$ we obtain $B = \operatorname{Bez}(a_-, a_+)$. □

3. Discrete Fourier and Hartley transforms. We recall the definition of the classical DHTs and introduce some modifications of it that will be used in what follows. Let

$$\begin{aligned}
\mathcal{S}_n^I &= \left[\ \sin\tfrac{2jk\pi}{n}\ \right]_0^{n-1}, & \mathcal{C}_n^I &= \left[\ \cos\tfrac{2jk\pi}{n}\ \right]_0^{n-1},\\
\mathcal{S}_n^{II} &= \left[\ \sin\tfrac{j(2k+1)\pi}{n}\ \right]_0^{n-1}, & \mathcal{C}_n^{II} &= \left[\ \cos\tfrac{j(2k+1)\pi}{n}\ \right]_0^{n-1},\\
\mathcal{S}_n^{III} &= (\mathcal{S}_n^{II})^T, & \mathcal{C}_n^{III} &= (\mathcal{C}_n^{II})^T,\\
\mathcal{S}_n^{IV} &= \left[\ \sin\tfrac{(2j+1)(2k+1)\pi}{2n}\ \right]_0^{n-1}, & \mathcal{C}_n^{IV} &= \left[\ \cos\tfrac{(2j+1)(2k+1)\pi}{2n}\ \right]_0^{n-1}.
\end{aligned}$$

We define

$$\mathcal{F}_n^X = \mathcal{C}_n^X + \mathbf{i}\,\mathcal{S}_n^X, \quad \mathcal{H}_n^X = \mathcal{C}_n^X + \mathcal{S}_n^X \quad (X = I, II, III, IV),$$

where $\mathbf{i}$ is the imaginary unit. If there is no danger of misunderstanding, we will omit the subscript n.

$\mathcal{F}^I$ is the classical DFT and $\mathcal{H}^I$ the classical DHT. Obviously, the modified DFTs are connected with the classical one via

$$\mathcal{F}^{II} = D_n\mathcal{F}^I, \quad \mathcal{F}^{III} = \mathcal{F}^I D_n, \quad \mathcal{F}^{IV} = D_n\mathcal{F}^I D_n, \tag{3.1}$$

where $D_n = \operatorname{diag}\left(e^{\frac{\pi j\mathbf{i}}{n}}\right)_{j=0}^{n-1}$. For the modified DHTs the following relations are valid:

$$\mathcal{H}^{II} = K_n\mathcal{H}^I, \quad \mathcal{H}^{III} = \mathcal{H}^I K_n^T, \quad \mathcal{H}^{IV} = K_n\mathcal{H}^I K_n^T, \tag{3.2}$$

where $K_n = \operatorname{diag}\left(\cos\frac{\pi j}{n}\right)_{j=0}^{n-1} + \operatorname{diag}\left(\sin\frac{\pi j}{n}\right)_{j=0}^{n-1} J_n'$. The relations (3.2) follow from (3.1) taking into account that $J_n'\mathcal{H}^I = \mathcal{C}^I - \mathcal{S}^I$. We can conclude from the relations (3.2) that the amount for computing the modified DHTs of a vector is approximately the same as for the classical DHT.

Applying a DFT to a vector $x \in \mathbb{C}^n$ means, in principle, evaluating the values of its generating function at unit roots. More precisely, we have for all four DFTs

$$\mathcal{F}^I x = \left[x(\omega_n^{4j})\right]_0^{n-1}, \quad \mathcal{F}^{II} x = \operatorname{diag}\left(\omega_n^{2j}\right)_0^{n-1}\left[x(\omega_n^{4j})\right]_0^{n-1},$$

$$\mathcal{F}^{III} x = \left[x(\omega_n^{4j+2})\right]_0^{n-1}, \quad \mathcal{F}^{IV} x = \operatorname{diag}\left(\omega_n^{2j+1}\right)_0^{n-1}\left[x(\omega_n^{4j+2})\right]_0^{n-1},$$

where $\omega_n = \exp(\frac{\mathbf{i}\pi}{2n})$, which is a primitive $4n$th root of unity.

Note that $(\mathcal{H}^X)^{-1} = \frac{1}{n}(\mathcal{H}^X)^T$ for $X = I, II, III, IV$. This is well known for $\mathcal{H}^I$ and can easily be checked for the other transformations. That means that these transformations are almost unitary.

In view of the symmetry properties of the rows or columns of $\mathcal{S}^X$ and $\mathcal{C}^X$ the following intertwining relations are valid:

$$\mathcal{H}^I J_n' = J_n'\mathcal{H}^I,\ \mathcal{H}^{II} J_n = -J_n'\mathcal{H}^{II},\ \mathcal{H}^{III} J_n' = -J_n\mathcal{H}^{III},\ \mathcal{H}^{IV} J_n = J_n\mathcal{H}^{IV}, \tag{3.3}$$

which means, in particular, that $\mathcal{H}^I$ is centro-even and $\mathcal{H}^{IV}$ is centro-symmetric. From these relations we conclude the following properties of the Hartley transforms.

LEMMA 3.1.

(1) *$\mathcal{H}^I$ transforms an even vector a_e into an even vector and an odd vector a_o into an odd one. Moreover,*

$$\mathcal{H}^I a_e = \mathcal{F}^I a_e = \mathcal{C}^I a_e, \quad \mathcal{H}^I a_o = -\mathbf{i}\mathcal{F}^I a_o = \mathcal{S}^I a_o, \quad \mathcal{S}^I a_e = 0, \quad \mathcal{C}^I a_o = 0.$$

(2) *$\mathcal{H}^{II}$ transforms a symmetric vector a_+ into an odd vector and a skewsymmetric vector a_- into an even one. Moreover,*

$$\mathcal{H}^{II} a_+ = \mathcal{F}^{II} a_+ = \mathcal{C}^{II} a_+, \quad \mathcal{H}^{II} a_- = -\mathbf{i}\mathcal{F}^{II} a_- = \mathcal{S}^{II} a_-,$$

$$\mathcal{S}^{II} a_+ = 0, \quad \mathcal{C}^{II} a_- = 0.$$

(3) *$\mathcal{H}^{III}$ transforms an even vector a_e into a skewsymmetric vector and an odd vector a_o into a symmetric one. Moreover,*

$$\mathcal{H}^{III} a_e = -\mathbf{i}\mathcal{F}^{III} a_e = \mathcal{S}^{III} a_e, \ \mathcal{H}^{III} a_o = \mathcal{F}^{III} a_o = \mathcal{C}^{III} a_o,$$

$$\mathcal{S}^{III} a_o = 0, \ \mathcal{C}^{III} a_e = 0.$$

(4) *$\mathcal{H}^{IV}$ transforms a symmetric vector a_+ into a symmetric vector and a skewsymmetric vector a_- into a skewsymmetric one. Moreover,*

$$\mathcal{H}^{IV} a_+ = -\mathbf{i}\mathcal{F}^{IV} a_+ = \mathcal{S}^{IV} a_+, \ \mathcal{H}^{IV} a_- = \mathcal{F}^{IV} a_- = \mathcal{C}^{IV} a_-,$$

$$\mathcal{S}^{IV} a_- = 0, \ \mathcal{C}^{IV} a_+ = 0.$$

Let us note that for any centro-symmetric matrix A the vector $A\,a_+$ is symmetric, whereas $A\,a_-$ is skewsymmetric. In case of a centro-skewsymmetric matrix A we obtain $A\,a_+$ is skewsymmetric, $A\,a_-$ symmetric. Analogous observations hold for centro-even or centro-odd matrices and the vectors a_e and a_o.

Lemma 3.1 has some important consequences which will be exploited in the following two sections. The first one is that for an even vector $a_e \in \mathbb{R}^n$ the vector $\left[a_e(\omega_n^{4k})\right]_{k=0}^{n-1}$ is real and equal to $\mathcal{H}^I a_e$ and the vector $-\mathbf{i}\left[a_e(\omega_n^{4k+2})\right]_{k=0}^{n-1}$ is real and equal to $\mathcal{H}^{III} a_e$. Furthermore, for a symmetric vector a_+, the vector $\left[\omega^{2k} a_+(\omega_n^{4k})\right]_{k=0}^{n-1}$ is real and equal to $\mathcal{H}^{II} a_+$, and $-\mathbf{i}\left[\omega^{2k+1} a_+(\omega_n^{4k+2})\right]_{k=0}^{n-1}$ is real and equal to $\mathcal{H}^{IV} a_+$.

A second consequence of Lemma 3.1 is that, for an even vector a_e and an odd vector a_o, the even part of $\mathcal{H}^I(a_e + a_o)$ is equal to $\mathcal{H}^I a_e$ and its odd part is equal to $\mathcal{H}^I a_o$. That means that only one transformation is needed to compute both $\mathcal{H}^I a_e$ and $\mathcal{H}^I a_o$. This observation will be utilized in what follows. The other DHTs have analogous properties. We collect them in the following lemma.

LEMMA 3.2.

(1) *Let a_e be even, a_o odd, and $b^I = \mathcal{H}^I(a_e + a_o)$, $b^{III} = \mathcal{H}^{III}(a_e + a_o)$. Then*

$$\mathcal{H}^I a_{e,o} = \frac{1}{2}\,(I \pm J_n')b^I, \quad \mathcal{H}^{III} a_{e,o} = \frac{1}{2}\,(I \mp J_n)b^{III}.$$

(2) *Let a_+ be symmetric, a_- skewsymmetric, $b^{II} = \mathcal{H}^{II}(a_+ + a_-)$, $b^{IV} = \mathcal{H}^{IV}(a_+ + a_-)$. Then*

$$\mathcal{H}^{II} a_\pm = \frac{1}{2}\,(I \mp J_n')b^{II}, \quad \mathcal{H}^{IV} a_\pm = \frac{1}{2}\,(I \pm J_n)b^{IV}.$$

Finally, the relations in the following lemma are also consequences of the relations in Lemma 3.1. Here we present only those identities which will be used in the remainder of this paper. Some other similar equalities can also be derived.

LEMMA 3.3. *Let A be centro-symmetric, A^0 centro-even, and W centro-skewsymmetric. Then*

$$\begin{aligned} \mathcal{H}^{II} A \mathcal{H}^{IV} &= -\mathbf{i}\mathcal{F}^{II} A \mathcal{F}^{IV}, & \mathcal{H}^{IV} A \mathcal{H}^{IV} &= \mathcal{F}^{IV} A (\mathcal{F}^{IV})^*, \\ \mathcal{H}^{I} A^0 \mathcal{H}^{I} &= \mathcal{F}^{I} A^0 (\mathcal{F}^{I})^*, & \mathcal{H}^{I} A^0 \mathcal{H}^{II} &= -\mathbf{i}\mathcal{F}^{I} A^0 \mathcal{F}^{II}, \\ \mathcal{H}^{II} W J_n \mathcal{H}^{III} &= -\mathbf{i}\mathcal{F}^{II} W (\mathcal{F}^{II})^*. \end{aligned}$$

Proof. The proof of all relations is straightforward and relies on the symmetry properties of the rows and columns of $\mathcal{S}^X$ and $\mathcal{C}^X$ and the fact that the inner product of a symmetric and a skewsymmetric vector, as well as that of an odd and an even vector, vanishes.

Let us prove, as an example, the last relation. We have on one hand

$$\begin{aligned} \mathcal{H}^{II} W J_n \mathcal{H}^{III} &= (\mathcal{C}^{II} + \mathcal{S}^{II}) W (\mathcal{C}^{III} - \mathcal{S}^{III}) \\ &= \mathcal{S}^{II} W \mathcal{C}^{III} - \mathcal{C}^{II} W \mathcal{S}^{III}. \end{aligned}$$

Here we took into account that the columns of $\mathcal{C}^{III}$ and the rows of $\mathcal{C}^{II}$ are symmetric and the columns of $\mathcal{S}^{III}$ and the rows of $\mathcal{S}^{II}$ are skewsymmetric; thus $J_n \mathcal{H}^{III} = \mathcal{C}^{III} - \mathcal{S}^{III}$, and the fact that W transforms symmetric into skewsymmetric and skewsymmetric into symmetric vectors. On the other hand,

$$\begin{aligned} \mathcal{F}^{II} W (\mathcal{F}^{II})^* &= (\mathcal{C}^{II} + \mathbf{i}\mathcal{S}^{II}) W (\mathcal{C}^{III} - \mathbf{i}\mathcal{S}^{III}) \\ &= \mathbf{i}(\mathcal{S}^{II} W \mathcal{C}^{III} - \mathcal{C}^{II} W \mathcal{S}^{III}). \end{aligned}$$

This completes the proof. □

4. DHT-representations of symmetric T-Bezoutians: Centro-even version. In this and the next section we present matrix representations of real symmetric T-Bezoutians given in the form $B = \mathrm{Bez}\,(a_-, a_+)$, where a_+ is symmetric and a_- is skewsymmetric, involving only Hartley transformations and diagonal matrices.

In this section we extend the centro-symmetric matrix B to a centro-even matrix. The simplest of these extensions is

$$B^0 = \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}. \tag{4.1}$$

We could also add more zero columns and rows to all sides of B. In general, it is convenient to do this in order to obtain a matrix the order N of which is a power of 2, because in this case the algorithms for DHT computing are most efficient. This can be achieved if the order n of B is odd, since $N - n$ must be odd. In the next section we present another version which is convenient if n is even.

Throughout this section all transformations have length $n+1$. Therefore, we omit this subscript. Furthermore, we abbreviate in this section

$$\omega = \omega_{n+1} = \exp\left(\frac{\mathbf{i}\pi}{2(n+1)}\right).$$

The generating function of the matrix (4.1) is given by

$$B^0(\lambda,\mu) = \lambda\mu\, B(\lambda,\mu) = \frac{\lambda a_+(\lambda)\mu a_-(\mu) + \lambda a_-(\lambda)\mu a_+(\mu)}{1-\lambda\,\mu}\,. \tag{4.2}$$

The function $B^0(\lambda,\mu)$ is defined by (4.2) only if $\lambda\mu \neq 1$. For $\mu \to \lambda^{-1}$ we obtain using l'Hospital's rule

$$B^0(\lambda,\lambda^{-1}) = -(\lambda a'_+(\lambda)a_-(\lambda^{-1}) + \lambda a'_-(\lambda)a_+(\lambda^{-1})).$$

It is convenient to write this relation in a different form. To that aim let us adopt the following notation: For a polynomial $a(\lambda)$, $\deg a \le n$, we denote by $\widetilde{a}(\lambda)$ the polynomial

$$\widetilde{a}(\lambda) = \frac{n}{2}\, a(\lambda) - \lambda a'(\lambda). \tag{4.3}$$

Now we can write

$$B^0(\lambda,\lambda^{-1}) = \widetilde{a}_+(\lambda)a_-(\lambda^{-1}) + \widetilde{a}_-(\lambda)a_+(\lambda^{-1}). \tag{4.4}$$

Note that the coefficient vector $\widetilde{a}_+$ is skewsymmetric and $\widetilde{a}_-$ is symmetric.

The construction of the representations for B^0 are carried out in two steps. In the first step we transform B^0 into a Cauchy-like matrix with DHTs, and in the second step we represent the Cauchy matrix by DHTs. The first step is done in two versions in the following lemma.

LEMMA 4.1.

(1) $\mathcal{H}^I B^0 \mathcal{H}^{II} = [p_{jk}]_{j,k=0}^n$ *with*

$$p_{jk} = \frac{a^{II}_{+,j}a^{IV}_{-,k} + a^{II}_{-,j}a^{IV}_{+,k}}{2\,\mathrm{Im}\,\omega^{2j+2k+1}}\,, \tag{4.5}$$

where

$$\left[\, a^X_{\pm,j}\,\right]_{j=0}^n = \mathcal{H}^X a_\pm \quad (X = II, IV)\,.$$

(2) $\mathcal{H}^I B^0 \mathcal{H}^I = [q_{jk}]_{j,k=0}^n$ *with*

$$q_{jk} = \begin{cases} \dfrac{a^{II}_{+,j}a^{II}_{-,k} - a^{II}_{-,j}a^{II}_{+,k}}{2\,\mathrm{Im}\,\omega^{2(j-k)}} & : \quad j \neq k, \\[2ex] \widetilde{a}^{II}_{+,j}a^{II}_{-,j} + \widetilde{a}^{II}_{-,j}a^{II}_{+,j} & : \quad j = k, \end{cases} \tag{4.6}$$

where $\widetilde{a}_\pm$ *is defined by* (4.3) *and* $\left[\, \widetilde{a}^{II}_{\pm,j}\,\right]_{j=0}^n = \mathcal{H}^{II}\widetilde{a}_\pm$.

Proof. From (4.2) we obtain that the entries $\mathbf{i}p_{jk}$ of $\mathcal{F}^I B^0 \mathcal{F}^{II} = \mathbf{i}\mathcal{H}^I B^0 \mathcal{H}^{II}$ are given by

$$\begin{aligned} \mathbf{i}p_{jk} &= \frac{\omega^{4j+4k+2}[a_+(\omega^{4j})a_-(\omega^{4k+2}) + a_-(\omega^{4j})a_+(\omega^{4k+2})]}{1-\omega^{4j+4k+2}} \\ &= \frac{\omega^{2j}a_+(\omega^{4j})\omega^{2k+1}a_-(\omega^{4k+2}) + \omega^{2j}a_-(\omega^{4j})\omega^{2k+1}a_+(\omega^{4k+2})}{\omega^{-2j-2k-1} - \omega^{2j+2k+1}}. \end{aligned}$$

Since a_+ is symmetric and a_- is skewsymmetric we have, according to Lemma 3.1,

$$[\,\omega^{2j}a_+(\omega^{4j})\,]_0^n = \mathcal{F}^{II}a_+ = \mathcal{H}^{II}a_+, \quad [\,\omega^{2j}a_-(\omega^{4j})\,]_0^n = \mathcal{F}^{II}a_- = \mathbf{i}\,\mathcal{H}^{II}a_-,$$

$$[\,\omega^{2k+1}a_+(\omega^{4k+2})\,]_0^n = \mathcal{F}^{IV}a_+ = \mathbf{i}\,\mathcal{H}^{IV}a_+, \quad [\,\omega^{2k+1}a_-(\omega^{4k+2})\,]_0^n = \mathcal{F}^{IV}a_- = \mathcal{H}^{IV}a_-.$$

From this (4.5) is immediately obtained.

In the second version the entries q_{jk} of $\mathcal{H}^I B^0 \mathcal{H}^I = \mathcal{F}^I B^0 (\mathcal{F}^I)^*$ for $j \neq k$ are given by

$$\begin{aligned} q_{jk} &= \frac{\omega^{4j-4k}[a_+(\omega^{4j})a_-(\omega^{-4k}) + a_-(\omega^{4j})a_+(\omega^{-4k})]}{1-\omega^{4j-4k}} \\ &= \frac{\omega^{2j}a_+(\omega^{4j})\omega^{-2k}a_-(\omega^{-4k}) + \omega^{2j}a_-(\omega^{4j})\omega^{-2k}a_+(\omega^{-4k})}{\omega^{-2j+2k}-\omega^{2j-2k}}. \end{aligned}$$

It remains again to remember the facts of section 2 and to take into account that

$$[\,\omega^{-2k}a_+(\omega^{-4k})\,]_{k=0}^n = \overline{\mathcal{F}^{II}a_+} = \mathcal{H}^{II}a_+, \quad [\,\omega^{-2k}a_-(\omega^{-4k})\,]_{k=0}^n = \overline{\mathcal{F}^{II}a_-} = -\mathbf{i}\mathcal{H}^{II}a_-$$

to obtain the first relation in (4.6).

We consider now the case $j = k$. According to (4.4),

$$\begin{aligned} q_{jj} &= \omega^{2j}\widetilde{a}_+(\omega^{4j})\omega^{-2j}a_-(\omega^{-4j}) + \omega^{2j}\widetilde{a}_-(\omega^{4j})\omega^{-2j}a_+(\omega^{-4j}) \\ &= \mathbf{i}\,\widetilde{a}^{II}_{+,j}(-\mathbf{i}\,a^{II}_{-,j}) + \widetilde{a}^{II}_{-,j}a^{II}_{+,j}, \end{aligned}$$

from which the second relation in (4.6) follows. □

Now we are going to represent the matrices obtained after transformation. For this we need the representation of some standard Cauchy-like matrices, which are deduced in the next lemma. We introduce the matrix

$$W_m = \frac{1}{m}\,\mathrm{diag}\left(\frac{m-1}{2}, \frac{m-3}{2}, \dots, \frac{3-m}{2}, \frac{1-m}{2}\right) J_m. \tag{4.7}$$

The matrix $W_m J_m$ is centro-skewsymmetric and, therefore, the last relation in Lemma 3.3 is valid.

LEMMA 4.2.

(1)

$$\mathcal{H}^{II}\mathcal{H}^{IV} = \left[\frac{1}{\mathrm{Im}\,\omega^{2j+2k+1}}\right]_{j,k=0}^n.$$

(2)

$$\mathcal{H}^{II}W_{n+1}\mathcal{H}^{III} = \left[\,(2\,\mathrm{Im}\,\omega^{2j-2k})^\dagger\right]_{j,k=0}^n,$$

where $a^\dagger$ *means* $1/a$ *if* $a \neq 0$ *and* 0 *if* $a = 0$.

Proof. 1. The generating function of the identity I_{n+1} is given by

$$I_{n+1}(\lambda,\mu) = \ell(\lambda)^T\ell(\mu) = \frac{1-\lambda^{n+1}\mu^{n+1}}{1-\lambda\mu}.$$

This implies

$$\mathcal{F}^{II}\mathcal{F}^{IV} = \left[\omega^{2j}\,\frac{1-(\omega^{4j})^{n+1}(\omega^{4k+2})^{n+1}}{1-\omega^{4j}\omega^{4k+2}}\,\omega^{2k+1}\right] = \left[\frac{2}{\omega^{-2j-2k-1}-\omega^{2j+2k+1}}\right].$$

It remains to take the first relation in Lemma 3.3 into account.

2. Differentiating the identity $(1-\lambda\mu)\ell(\lambda)^T\ell(\mu) = 1-\lambda^{n+1}\mu^{n+1}$ by λ and multiplying the result by λ, we obtain

$$(1-\lambda\mu)\lambda\ell'(\lambda)^T\ell(\mu) = -(n+1)(\lambda\mu)^{n+1} + \lambda\mu\ell(\lambda)^T\ell(\mu).$$

In particular, we have for $\lambda = \omega^{4j}$ and $\mu = \omega^{-4k}$, $j \neq k$,

$$(1-\omega^{4j-4k})\omega^{4j}\ell'(\omega^{4j})^T\ell(\omega^{-4k}) = -(n+1).$$

Furthermore, direct computation yields

$$\omega^{4j}\ell'(\omega^{4j})^T\ell(\omega^{-4j}) = \frac{1}{2}\,n(n+1).$$

These two relations lead to

$$\mathcal{F}^I W_{n+1}J_{n+1}(\mathcal{F}^I)^* = \left[\,(1-\omega^{4j-4k})^\dagger\,\right]_{j,k=0}^{n}.$$

Consequently,

$$\mathcal{F}^{II} W_{n+1}J_{n+1}(\mathcal{F}^{II})^* = \left[\,\omega^{2j-2k}(1-\omega^{4j-4k})^\dagger\,\right]_{j,k=0}^{n} = \left[\,\mathbf{i}\,(2\mathrm{Im}\,\omega^{2j-2k})^\dagger\,\right]_{j,k=0}^{n}.$$

Since $W_{n+1}J_{n+1}$ is centro-skewsymmetric we conclude from this the assertion using the last relation in Lemma 3.3.

In order to reduce representations for B^0 to representations for B we introduce the $n\times(n+1)$ matrix

$$P_{10} = \left[\begin{array}{cc} 0 & I_n \end{array}\right],$$

where I_n denotes the identity matrix of order n.

Combining Lemma 4.1 and Lemma 4.2 we obtain two kinds of representations of real symmetric Bezoutians which are presented in the following theorem.

THEOREM 4.3. *The real symmetric Bezoutian* $B = \mathrm{Bez}\,(a_-, a_+)$ *admits the representations*

$$B = \frac{1}{2(n+1)^2}\,P_{10}\mathcal{H}^I\left(D_+^{II}\mathcal{H}^{II}\mathcal{H}^{IV}D_-^{IV} + D_-^{II}\mathcal{H}^{II}\mathcal{H}^{IV}D_+^{IV}\right)\mathcal{H}^{III}P_{10}^T, \tag{4.8}$$

where $D_\pm^X = \mathrm{diag}\,\left(\mathcal{H}^X a_\pm\right)$ *for* $X = II, IV$, *and*

$$B = \frac{1}{(n+1)^2}P_{10}\mathcal{H}^I\left(D_+^{II}\mathcal{H}^{II}W_{n+1}\mathcal{H}^{III}D_-^{II} - D_-^{II}\mathcal{H}^{II}W_{n+1}\mathcal{H}^{III}D_+^{II} + D\right)\mathcal{H}^I P_{10}^T, \tag{4.9}$$

where $D = \mathrm{diag}\,(d_j)_{j=0}^n$, $d_j = \widetilde{a}_{+,j}^{II}a_{-,j}^{II} + \widetilde{a}_{-,j}^{II}a_{+,j}^{II}$.

5. DHT-representations of symmetric T-Bezoutians: Centro-symmetric version. In this section we present two representations for $B = \mathrm{Bez}\,(a_-, a_+)$ that use the fact that B itself is centro-symmetric. In order to have all Hartley transformations of the same length we extend B to a larger centro-symmetric matrix bordering it by the same number of zero rows and columns from all sides, i.e., we consider instead of B a matrix of the form

$$B_0^0 = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 0 \end{array}\right].$$

If the order of B is even, then it can be achieved that the order of B_0^0 is a power of 2, which is a convenient length for the DHT algorithms. For simplicity we restrict ourselves to the case that B_0^0 has order $n+2$, which is no restriction of generality.

All transformations in this sections will have length $n+2$. We omit the corresponding subscript. Furthermore, in this section

$$\omega = \omega_{n+2} = \exp\left(\frac{\mathbf{i}\pi}{2(n+2)}\right).$$

Instead of the polynomials $a_\pm(\lambda)$ we deal with the polynomials $a_e(\lambda) = \lambda a_+(\lambda)$ and $a_o(\lambda) = \lambda a_-(\lambda)$. The coefficient vectors of these polynomials are, as vectors in $\mathbb{R}^{n+2}$, even and odd, respectively. Obviously,

$$B_0^0(\lambda,\mu) = \frac{a_e(\lambda)a_o(\mu) + a_o(\lambda)a_e(\mu)}{1-\lambda\mu}$$

for $\lambda\mu \neq 1$ and

$$B_0^0(\lambda,\lambda^{-1}) = -\lambda(a_e'(\lambda)a_o(\lambda^{-1}) + a_o'(\lambda)a_e(\lambda^{-1})) = \widetilde{a}_e(\lambda)a_o(\lambda^{-1}) + \widetilde{a}_o(\lambda)a_e(\lambda^{-1}),$$

where here (as opposed to in the previous section)

$$\widetilde{a}(\lambda) = \frac{n+2}{2}\,a(\lambda) - \lambda a'(\lambda). \tag{5.1}$$

Note that $\widetilde{a}_e$ is odd whereas $\widetilde{a}_o$ is even.

First we transform the matrix B_0^0 into a Cauchy-like matrix. The following lemma can be proved along the same lines as Lemma 4.1.

LEMMA 5.1.

(1) $\mathcal{H}^{II}B_0^0\mathcal{H}^{IV} = [p_{jk}]_{j,k=0}^{n+1}$ *with*

$$p_{jk} = \frac{a_{e,j}^I a_{o,k}^{III} + a_{o,j}^I a_{e,k}^{III}}{2\operatorname{Im}\omega^{2j+2k+1}}, \tag{5.2}$$

where

$$\left[\,a_{e,j}^X\,\right]_{j=0}^{n+1} = \mathcal{H}^X a_e\,, \quad \left[\,a_{o,j}^X\,\right]_{j=0}^{n+1} = \mathcal{H}^X a_o \quad (X = I, III).$$

(2) $\mathcal{H}^{IV}B^0\mathcal{H}^{IV} = [q_{jk}]_{j,k=0}^{n+1}$ *with*

$$q_{jk} = \begin{cases} \dfrac{a_{e,j}^{III}a_{o,k}^{III} - a_{o,j}^{III}a_{e,k}^{III}}{2\operatorname{Im}\omega^{2(j-k)}} & : \quad j \neq k, \\[2ex] \widetilde{a}_{e,j}^{III}a_{o,j}^{III} + \widetilde{a}_{o,j}^{III}a_{e,j}^{III} & : \quad j = k, \end{cases} \tag{5.3}$$

where $\widetilde{a}$ *is defined by* (5.1) *and* $[\widetilde{a}_j^{III}]_{j=0}^{n+1} = \mathcal{H}^{III}\widetilde{a}\,$.

Taking Lemmas 5.1 and 4.2 together we obtain representations for B_0^0. In order to reduce them to representations for B we introduce the $n \times (n+2)$ matrix

$$P_{11} = \left[\begin{array}{ccc} 0 & I_n & 0 \end{array}\right].$$

THEOREM 5.2. *The real symmetric Bezoutian* $B = \operatorname{Bez}(a_-, a_+)$ *admits the representations*

$$B = \frac{1}{2(n+2)^2}\,P_{11}\mathcal{H}^{III}\left(D_e^I\mathcal{H}^{II}\mathcal{H}^{IV}D_o^{III} + D_o^I\mathcal{H}^{II}\mathcal{H}^{IV}D_e^{III}\right)\mathcal{H}^{IV}P_{11}^T\,, \tag{5.4}$$

where $D^X_{e,o} = \operatorname{diag}\left(\mathcal{H}^X a_{e,o}\right)$ *for* $X = I, III$, *and*

$$B = \frac{1}{(n+2)^2} P_{11}\mathcal{H}^{IV}\left(D_o^{III}\mathcal{H}^{II}W_{n+2}\mathcal{H}^{III}D_e^{III} - D_e^{III}\mathcal{H}^{II}W_{n+2}\mathcal{H}^{III}D_e^{III} + D\right)\mathcal{H}^{IV}P_{11}^T, \tag{5.5}$$

where $D = \operatorname{diag}(d_j)_0^{n+1}$, $d_j = \widetilde{a}^{III}_{e,j}a^{III}_{o,j} + \widetilde{a}^{III}_{o,j}a^{III}_{e,j}$.

6. Stability. In this section we study the stability of the algorithms for matrix-vector multiplication by the inverse of a real symmetric Toeplitz matrix emerging from the representations of symmetric T-Bezoutians presented in the previous two sections.

First we recall that these representations involve only Hartley transforms and Hadamard products. The Cooley–Tukey algorithm for the Hartley transformation (see [34]) is forward stable in the sense that the relative forward error can be made arbitrarily small if the unit roots are given with sufficient accuracy. A proof for this was provided to us by Tasche [33]. The proof is similar to that for the FFT given in [24, section 23.1]. Since DHT computation can be made arbitrarily accurate and the DHTs are almost unitary, we neglect all errors that arise from DHT computation.

It remains to consider Hadamard products. If we use the standard model of floating point arithmetic, in which $fl(\alpha \operatorname{op} \beta) = (\alpha \operatorname{op} \beta)(1+\delta)$ holds for numbers α and β and any arithmetic operation "op," where $|\delta| \le \mathbf{u}$ and $\mathbf{u}$ is the machine precision, then we have for the Hadamard product of two vectors α and β for which approximations $\widetilde{\alpha}$ and $\widetilde{\beta}$ satisfying $\|\widetilde{\alpha}-\alpha\|/\|\alpha\| < k\mathbf{u}$ and $\|\widetilde{\beta}-\beta\|/\|\beta\| < l\mathbf{u}$ are given[2]

$$\|fl(\widetilde{\alpha}\circ\widetilde{\beta}) - \alpha\circ\beta\| \le (k+l+1)\|\alpha\|\,\|\beta\|(\mathbf{u}+O(\mathbf{u}^2)).$$

The analysis of the computation of $T^{-1}b$ via the formulas (4.8) or (5.4) reduces to the computation of a vector ξ by

$$\xi = p_+ \circ (\Omega\,(q_-\circ\eta)) + p_-\circ(\Omega\,(q_+\circ\eta)),$$

where Ω is unitary and the product of a DHT and an inverse DHT.

We have actually still a factor $\gamma/2$, but it is convenient to let the factor be absorbed by one of the vectors a_+ or a_-. We choose a_+ for it. That means, in the representations of Theorems 4.3 and 5.2, we redefine D_+^X as $D_+^X = \operatorname{diag}\left(\frac{\gamma}{2}\,a^X_{+,j}\right)_{j=0}^{N-1}$ ($N = n+1$ or $N = n+2$).

If we assume that all vectors are given with machine precision in the sense that instead of any vector w appearing in the formula we have a vector $\widetilde{w}$ satisfying $\|w-\widetilde{w}\|/\|w\| \le \mathbf{u}$, then

$$\|fl(\widetilde{p}_+\circ fl(\Omega\,(\widetilde{q}_-\circ\widetilde{\eta}))) + p_+\circ(\Omega(q_-\circ\eta))\| \le 5\|q_-\|\,\|p_+\|\|\eta\|(\mathbf{u}+O(\mathbf{u}^2)).$$

Together with the corresponding estimation for the second term, this leads to

$$\|\widetilde{\xi}-\xi\| \le 6\left(\|p_-\|\,\|q_+\| + \|q_-\|\,\|p_+\|\right)\|\eta\|(\mathbf{u}+O(\mathbf{u}^2)), \tag{6.1}$$

[2]For simplicity, we consider all estimations in the euclidean norm. Stronger estimates are available in other norms.

where $\widetilde{\xi}$ is the computed vector.

Note that $\|p_-\| \le \sqrt{N}\,\|a_-\|$ and $\|p_+\| \le \sqrt{N}\,\|\frac{\gamma}{2}\,a_+\|$, and the same for $q_\pm$. The estimation (6.1) shows that in order to show stability we have to find bounds for the norms of the vectors a_- and $\frac{\gamma}{2}a_+$.

We discuss the first version of Theorem 2.5, in which T_{n-1} is assumed to be nonsingular. The following lemma is crucial.

LEMMA 6.1. *Let x be the solution of the second equation in* (2.2) *and let $x_0 \neq 0$. Then*

$$\frac{1}{|x_0|}\|x\|^2 \le \|T_n^{-1}\| + \|T_{n-1}^{-1}\|.$$

In particular, if T_n is positive definite, then

$$\frac{1}{|x_0|}\|x\|^2 \le 2\|T_n^{-1}\|.$$

Proof. We form the matrix

$$X = \begin{bmatrix} x_0 & & 0 \\ x_1 & 1 & \\ \vdots & & \ddots \\ x_{n-1} & & 1 \end{bmatrix}.$$

Then

$$X^T T_n X = \begin{bmatrix} x_0 & 0 \\ 0 & T_{n-1} \end{bmatrix}.$$

Hence

$$T_n^{-1} = X \begin{bmatrix} 1/x_0 & 0 \\ 0 & T_{n-1}^{-1} \end{bmatrix} X^T = \frac{1}{x_0} x x^T + \begin{bmatrix} 0 & 0 \\ 0 & T_{n-1}^{-1} \end{bmatrix}.$$

The assertion is now immediate. ∎

Since the operators $\frac{1}{2}(I_{n+1} \pm J_{n+1})$ are orthogonal projections, we have

$$\|a_-\| \left\|\frac{\gamma}{2}\,a_+\right\| \le 2\,(\|T_n^{-1}\| + \|T_{n-1}^{-1}\|). \tag{6.2}$$

Inserting this into (6.1) we obtain

$$\begin{aligned}\|\widetilde{\xi} - \xi\| &\le 24N(\|T_n^{-1}\| + \|T_{n-1}^{-1}\|)\|b\|(\mathbf{u} + O(\mathbf{u}^2)) \\ &\le 24N(\kappa(T_n) + \kappa(T_{n-1}) + |t_{n-1}|\|T_{n-1}^{-1}\|)\|\xi\|(\mathbf{u} + O(\mathbf{u}^2)).\end{aligned}$$

We arrive at the following theorem.

THEOREM 6.2. *Let $\xi = T_n^{-1}b$ be computed by the first formula of Theorem* 4.3 *or Theorem* 5.2, *where $a_\pm$ are given according to the first version of Theorem* 2.5 *and all vectors are given in floating point machine precision* $\mathbf{u}$. *If all errors caused by DHT computation are neglected, then the relative error of the computed vector $\widetilde{\xi}$ can be estimated as*

$$\frac{\|\widetilde{\xi} - \xi\|}{\|\xi\|} \le 24N(\kappa(T_{n-1}) + \kappa(T_n) + |t_{n-1}|\|T_{n-1}^{-1}\|)(\mathbf{u} + O(\mathbf{u}^2)).$$

In particular, if T_n is positive definite, then

$$\frac{\|\widetilde{\xi} - \xi\|}{\|\xi\|} \leq 48N\kappa(T_n)(\mathbf{u} + O(\mathbf{u}^2)).$$

This means that we have a small relative forward error if both T_n and T_{n-1} are well conditioned. For positive definite T_n this estimations means stability in the sense of [24, p. 142].

Let us point out that stability of a Toeplitz matrix inversion formula in this strong sense, i.e., with an estimation proportional to the condition number, is, to the best of our knowledge, proved here for the first time. In [16] a linear estimate for the relative error of the inverse matrix computed by a formula of the first inversion variant in [20] is presented. This leads, however, to quadratic estimates for the error in the computed solution.

Note that one can show with the same arguments that the algorithm emerging from the Gohberg–Semencul formula is stable for positive definite Toeplitz matrices. On the other hand, it can be shown that the formula of the first inversion variant in [20], which is the second version of Theorem 2.5, may be not stable (in our sense) for positive definite Toeplitz matrices (see [19, Example 2]). This is remarkable, because the investigations in [16] seem to indicate that this formula is favorable compared with the Gohberg–Semencul formula. For more discussion on this issue, we refer to [19].

Now we consider the second version of Theorem 2.5. For this we mention first that

$$T_{n+1}(t_n)y = -\frac{1}{\gamma}e_{n+1},$$

where e_{n+1} is the last unit vector, y is given in Theorem 2.5, and γ is given by (2.6).

LEMMA 6.3.

$$|\gamma|\|y\|^2 \leq \|T_n^{-1}\| + \|T_{n+1}(t_n)^{-1}\|.$$

Proof. We form the matrix

$$Y = \begin{bmatrix} 1 & & -z_0 \\ & \ddots & \vdots \\ & 1 & -z_{n-1} \\ 0 & & 1 \end{bmatrix}.$$

Then

$$Y^T T_{n+1}(t_n) Y = \begin{bmatrix} T_n & 0 \\ 0 & -1/\gamma \end{bmatrix}.$$

Hence

$$T_{n+1}(t_n)^{-1} = Y \begin{bmatrix} T_n^{-1} & 0 \\ 0 & -\gamma \end{bmatrix} Y^T = -\gamma y y^T + \begin{bmatrix} T_n^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

The assertion is now immediate. ∎

From this lemma we conclude that

$$\|a_-\| \left\| \frac{\gamma}{2}\, a_+ \right\| \le 2\,(\|T_n^{-1}\| + \|T_{n+1}(t_n)^{-1}\|). \tag{6.3}$$

Inserting this into (6.1) we obtain the estimation

$$\frac{\|\widetilde{\xi} - \xi\|}{\|\xi\|} \le 24N(\kappa(T_n) + \|T_{n+1}(t_n)^{-1}\|\,\|T_n\|)(\mathbf{u} + O(\mathbf{u}^2))\ .$$

That means that we have a small relative forward error if T_n is well conditioned and has a well-conditioned extension $T_{n+1}(t_n)$. In [19] it is shown that such a well-conditioned extension, even an extension with $\|T_{n+1}(t_n)^{-1}\| \le \|T_n^{-1}\|$, always exists. For example, if

$$T_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

then for any t_2 away from 0 and not too large the matrix $T_3(t_2)$ is well conditioned.

7. Complexity. In this section we give an account of the number of operations required for matrix-vector multiplication by a symmetric Bezoutian B, in particular an inverse symmetric Toeplitz matrix, with the help of the formulas in Theorems 4.3 and 5.2. For this we recall first that the application of these formulas includes only Hartley transforms, Hadamard products, i.e., multiplications by diagonal matrices, and additions of vectors. Next we mention that we have to distinguish two phases. In the first one, the preprocessing phase, the data involved in the formula are evaluated. If matrix-vector products Bb have to be computed for several vectors b, this has to be done only once. The second phase is then the computation of Bb according to the formula using the precomputed data. Since preprocessing has to be done only once, we put as many calculations as possible in this phase.

Let us discuss the complexity of the Hartley transforms. In [10], [31], [32] fast algorithms for the classical DHT-I $\mathcal{H}_N^I$ are presented that require $N \log_2 N$ multiplications and $\frac{3}{2}\, N \log_2 N$ additions, provided that N is a power of 2. The other DHTs can be computed using formulas (3.2) and the algorithm for the computation of the DHT-I. One has then $2N$ more multiplications for DHT-II and DHT-III, and $4N$ more multiplications for DHT-IV. Possibly, there are algorithms for DHT-II, DHT-III, and DHT-IV without this overhead complexity.

Since most of the DHT algorithms in the literature are designed for vectors, the length N of which is a power of 2, we extend a given symmetric $n \times n$ Bezoutian B to a matrix of order $N = 2^t \ge n$ via bordering B by zeros in such a way that the extended matrix is centro-symmetric or centro-even, depending on whether n is even or odd. In this sense, the formulas in Theorem 4.3 are convenient for odd n and the formulas in Theorem 5.2 for even n.

We assume, moreover, that $N > n$ if n is odd and $N > n+1$ if n is even, in order to guarantee that all DHTs occurring in the representations of B have length N.

Table 1 shows the amount of the second phase. In this table we consider the different DHTs as equal.

In the preprocessing phase the diagonal matrices occurring in the representations have to be computed. For this we can take advantage of Lemma 3.2, which states that

TABLE 1
Amount for computation of Bb.

	# of DHTs	Multiplications	Additions
(4.8) or (5.4)	6	$4N$	N
(4.9) or (5.5)	6	$7N$	$2N$

TABLE 2
Amount for preprocessing.

	# of DHTs	Multiplications	Additions
(4.8) or (5.4)	2	N	$3N$
(4.9) or (5.5)	2	$3N$	$9N/2$

the DHT of two vectors, a symmetric and a skewsymmetric one, can be computed with one transformation only. Thus only two DHTs of length N are necessary for preprocessing in both formulas of Theorems 4.3 and 5.2. This is a complexity gain compared with the representations given in [11], [7], [5], where four transformations in the preprocessing phase are needed.

The operations are counted as follows. We assume that B is given as $B = \mathrm{Bez}\,(a_-, a_+)$. In the case of formula (4.8) or (5.4), N additions are required to compute $a_+ + a_-$. Then two Hartley transformations produce b^X for $X = II$ and $X = IV$, and $2N$ additions give $\mathcal{H}^X a_\pm$. Here we utilize the symmetry properties of the vectors. Finally two diagonals have to be multiplied by a factor which requires, due to symmetry, N multiplications. Similarly the operation account for the formulas (4.9) and (5.5) can be explained.

Tables 1 and 2 suggest that the first formulas in Theorems 4.3 and 5.2 are more efficient, from complexity point of view, than the second ones. However, the second formulas better reflect the symmetry of the original matrix B. This can be utilized if moments $b^T B\, b$, rather than vectors Bb, have to be computed. We obtain the following.

COROLLARY 7.1. *Using formulas* (4.9) *and* (5.5) *for a symmetric Bezoutian B, the moment $b^T B\, b$, where $b \in \mathbb{R}^n$, can be evaluated with the help of three DHTs and two DHTs for preprocessing plus $O(n)$ operations.*

As the FFT algorithms, the fast algorithms for DHT computation are highly parallelizable. They require only $O(\log N)$ operations on a parallel computer with N processors. Since Hadamard products and vector addition have only $O(1)$ parallel complexity, we conclude that matrix-vector multiplication by B can be carried out with $O(\log N)$ parallel complexity if N processors are available.

Iterative refinement is a useful tool to improve the accuracy of the solution of a linear system. In the case of a Toeplitz system $T_n\xi = b$ it is desirable to carry out the iteration with the help of a few fast transformations. Let B be a Bezoutian approximating T_n^{-1}. Then we replace the system $T_n\xi = b$ by the equivalent system $\mathcal{H}BT_n\xi = \mathcal{H}Bb$, where $\mathcal{H} = \mathcal{H}^I$ or $\mathcal{H} = \mathcal{H}^{II}$. Then one needs five DHTs to compute $\mathcal{H}Bb$. Each iteration step consists mainly in a matrix-vector product by $\mathcal{H}BT_n$. In [21] representations of Toeplitz matrices using Hartley transforms are presented that allow matrix-vector multiplication with the help of only four DHTs. That would mean that multiplication by $\mathcal{H}BT_n$ can be done with nine DHTs. However, it can easily be seen that eight DHTs are sufficient. The question is now whether this amount can further be reduced.

REFERENCES

[1] G. Ammar and P. Gader, *A variant of the Gohberg–Semencul formula involving circulant matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 534–540.

[2] G. Ammar and P. Gader, *New decompositions of the inverse of a Toeplitz matrix*, in Signal Processing, Scattering and Operator Theory, Progr. Systems Control Theory 5, Birkhäuser Boston, Cambridge, MA, 1990, pp. 421–428.

[3] D. Bini and P. Favati, *On a matrix algebra related to the discrete Hartley transform*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 500–507.

[4] D. Bini and V. Pan, *Polynomial and Matrix Computations,* I. *Fundamental Algorithms*, Birkhäuser Boston, Cambridge, MA, 1994.

[5] E. Bozzo, *Matrix Algebras and Discrete Transforms*, Ph.D. thesis TD-1/94, Dipartimento di Informatica, Universita di Pisa, Italy, 1994.

[6] E. Bozzo, *Algebras of higher dimension for displacement decompositions and computations with Toeplitz-plus-Hankel matrices*, Linear Algebra Appl., 230 (1995), pp. 127–150.

[7] E. Bozzo and C. Di Fiore, *On the use of certain matrix algebras associated with discrete trigonometric transforms in matrix displacement decompositions*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 312–326.

[8] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 349–364.

[9] O. Buneman, *Conversion of FFT's to fast Hartley transform*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 624–638.

[10] P. Duhamel and M. Vetterli, *Improved Fourier and Hartley transform algorithms and application to cyclic convolution of real data*, IEEE Trans. Acoust. Speech Signal Process., 35 (1987), pp. 818–824.

[11] C. Di Fiore and P. Zellini, *Matrix decompositions using displacement rank and classes of commutative matrix algebras*, Linear Algebra Appl., 229 (1995), pp. 49–99.

[12] C. Di Fiore and P. Zellini, *Matrix displacement decompositions and applications to Toeplitz linear systems*, Linear Algebra Appl., 268 (1998), pp. 197–226.

[13] I. Gohberg and V. Olshevsky, *Circulants, displacements and decomposition of matrices*, Integral Equations and Operator Theory, 15 (1992), pp. 730–743.

[14] I. Gohberg and V. Olshevsky, *Complexity of multiplication with vectors for structured matrices*, Linear Algebra Appl., 202 (1994), pp. 163–192.

[15] I. Gohberg and A. Semencul, *On the inversion of finite-section Toeplitz matrices and their continuous analogues*, Mat. Issled., 7 (1972), pp. 201–224 (in Russian).

[16] M. Gutknecht and M. Hochbruck, *The stability of inversion formulas for Toeplitz matrices*, Linear Algebra Appl., 223/224 (1995), pp. 307–324.

[17] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in Linear Algebra for Signal Processing, IMA Vol. Math. Appl. 69, 1994, pp. 95–114.

[18] G. Heinig, *Matrix representation of Bezoutians*, Linear Algebra Appl., 223/224 (1995), pp. 337–354.

[19] G. Heinig, *Stability of Toeplitz matrix inversion formulas*, in Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing, Contemp. Math., AMS, Providence, RI, submitted.

[20] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-Like Matrices and Operators*, Akademie-Verlag, Berlin, and Birkhäuser Basel, Boston, MA, 1984.

[21] G. Heinig and K. Rost, *Representations of Toeplitz-plus-Hankel matrices using trigonometric transformations with application to fast matrix-vector multiplication*, Linear Algebra Appl., 275/276 (1998), pp. 225–248.

[22] G. Heinig and K. Rost, *DFT representations of Toeplitz-plus-Hankel Bezoutians with application to fast matrix-vector multiplication*, Linear Algebra Appl., 284 (1998), pp. 157–175.

[23] G. Heinig and K. Rost, *Hartley transform representations of inverses of real Toeplitz-plus-Hankel matrices*, Numer. Funct. Anal. Optim., 2000, to appear.

[24] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.

[25] T. Kailath and V. Olshevsky, *Displacement structure approach to discrete transform based preconditioners of G. Strang type and of T. Chan type*, Calcolo, 33 (1996), pp. 191–208.

[26] T. Kailath and A. H. Sayed, *Displacement structure: Theory and applications*, SIAM Rev., 37 (1995), pp. 297–386.

[27] M. Krein and M. Naimark, *The method of symmetric and Hermitian forms in the theory of the separation of the roots of algebraic equations*, Linear and Multilinear Algebra, 56 (1974), pp. 69–87.

[28] F. I. Lander, *The Bezoutian and the inversion of Hankel and Toeplitz matrices*, Mat. Issled.,

9 (1974), pp. 69–87 (in Russian).
[29] L. Lerer and M. Tismenetsky, *Generalized Bezoutian and the inversion problem for block matrices,* I. *General scheme*, Integral Equations Operator Theory, 9 (1986), pp. 790–819.
[30] K. Rost, *Generalized companion matrices and matrix representations for generalized Bezoutians*, Linear Algebra Appl., 193 (1993), pp. 151–172.
[31] H. Sorensen, D. Jones, M. Heidman, and C. Burrus, *On computing the discrete Hartley transform*, IEEE Trans. Acoust. Speech Signal Process., 33 (1985), pp. 1231–1238.
[32] H. Sorensen, D. Jones, M. Heidman, and C. Burrus, *Real valued fast Fourier transform algorithms*, IEEE Trans. Acoust. Speech Signal Process., 35 (1987), pp. 849–863.
[33] M. Tasche, *Why are the Hartley Transforms Stable?*, manuscript.
[34] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, PA, 1992.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 106–113

SPECIAL ULTRAMETRIC MATRICES AND GRAPHS*

MIROSLAV FIEDLER†

Abstract. Special ultrametric matrices are, in a sense, extremal matrices in the boundary of the set of ultrametric matrices introduced by Martínez, Michon, and San Martín [*SIAM J. Matrix Anal. Appl.*, 15 (1994), pp. 98–106]. We show a simple construction of these matrices, if of order n, from nonnegatively edge-weighted trees on n vertices, or, equivalently, from nonnegatively edge-weighted paths. A general ultrametric matrix is then the sum of a nonnegative diagonal matrix and a special ultrametric matrix, with certain conditions fulfilled. The rank of a special ultrametric matrix is also recognized and it is shown that its Moore–Penrose inverse is a generalized diagonally dominant M-matrix. Some results on the nonsymmetric case are included.

Key words. ultrametric matrix, weighted graph, M-matrix

AMS subject classifications. 15A48, 05C50

PII. S0895479899350988

1. Introduction.

Strictly ultrametric matrices were defined in [3] as nonnegative symmetric matrices $A = (a_{ik})$ whose entries satisfy the following inequalities:

$$a_{ik} \geq \min(a_{ij}, a_{jk}) \quad \text{for all} \quad i, j, k, \tag{1}$$

$$a_{ii} > a_{ik} \quad \text{for all} \quad i, k, i \neq k. \tag{2}$$

The following theorem was proved in [3] and a linear algebra proof was given in [5].

THEOREM A. *Every strictly ultrametric matrix is nonsingular and its inverse is a diagonally dominant M-matrix.*

A construction also was given in [5] to describe all such ultrametric matrices. Later, nonsymmetric ultrametric matrices were independently defined in [4] and [6], and equality was allowed in (2) (with additional constraints). We shall call such matrices *ultrametric matrices*.

In the present note, we first deal with symmetric ultrametric matrices. We shall say that a symmetric ultrametric matrix is *special* if the following equality holds for all i:

$$a_{ii} = \max_{k \neq i} a_{ik}. \tag{3}$$

Such a matrix is in fact never nonsingular but is always, as we shall show, the limit of a convergent sequence of matrices that are inverses of (weakly) diagonally dominant M-matrices. For the sake of brevity, we shall denote as $\mathcal{Q}$ the class of all inverses of weakly diagonally dominant nonsingular M-matrices and as $\overline{\mathcal{Q}}$ the *closure* of $\mathcal{Q}$, i.e., the set of limits of convergent sequences of matrices in $\mathcal{Q}$.

Now let T be a tree on n vertices $V_1, \ldots, V_n$, let $E_1, \ldots, E_{n-1}$ be its edges. We say that T is *nonnegatively edge-weighted* if a nonnegative number w_i is assigned to each

*Received by the editors April 30, 1999; accepted for publication (in revised form) by R. Brualdi November 16, 1999; published electronically May 31, 2000. This research was supported by grant GACR 201/98/0222.

http://www.siam.org/journals/simax/22-1/35098.html

†Academy of Sciences of the Czech Republic, Institute of Computer Science, Pod vodárenskou věží 2, 187 02 Praha 8, The Czech Republic (fiedler@math.cas.cz).

edge E_i, $i = 1, \ldots, n-1$. We then assign to every such nonnegatively edge-weighted tree T a nonnegative $n \times n$ symmetric matrix $C(T) = (c_{ik})$ as follows.

For each k,

$$c_{kk} = \max\{w_i;\ E_i \text{ is incident with } V_k\}; \tag{4}$$

if $i \neq k$, then

$$c_{ik} = \min\{w_j; E_j \text{ is in the path from } V_i \text{ to } V_k\}. \tag{5}$$

The main result of the first part of this note is that the class of all matrices $C(T)$ just defined coincides with the class of special symmetric ultrametric matrices, and this is true even if we restrict ourselves to paths.

In addition, every special symmetric matrix belongs to $\overline{\mathcal{Q}}$, and every symmetric ultrametric matrix is a sum of a special ultrametric matrix and a nonnegative diagonal matrix.

In connection with this assertion, we shall use the following well-known result which was (in a slightly different form) proved in [2].

THEOREM B. *If $A \in \overline{\mathcal{Q}}$ and D is a diagonal matrix of the same order with positive diagonal entries, then $A + D \in \mathcal{Q}$. If $A \in \overline{\mathcal{Q}}$ and D is a nonnegative diagonal matrix of the same order, then $A + D \in \overline{\mathcal{Q}}$. If for some real square matrix B, $B + D \in \mathcal{Q}$ for every diagonal matrix D of the same order with positive diagonal entries, then $B \in \overline{\mathcal{Q}}$.*

In the second part, we shall discuss the nonsymmetric case and introduce two new classes of ultrametric matrices.

2. Symmetric ultrametric matrices.

THEOREM 2.1. *Every special symmetric ultrametric matrix is singular and belongs to $\overline{\mathcal{Q}}$.*

Proof. Let $A = (a_{ik})$ be a special symmetric ultrametric matrix. Let a_{pq}, $p \neq q$, be a maximal off-diagonal entry of A. We shall show that the pth and qth rows of A are identical. Clearly, $a_{pp} = a_{qq} = a_{pq}$. If $p \neq j \neq q$, then $a_{pj} \geq \min(a_{pq}, a_{qj}) = a_{qj} \geq \min(a_{qp}, a_{pj}) = a_{pj}$.

If now $\{D_k\}$, $k = 1, 2, \ldots$, is a sequence of diagonal matrices with positive diagonal entries converging to the zero matrix, then all matrices $A + D_k$ are ultrametric in the original sense. Therefore, by Theorem A the inverses of the matrices $A + D_k$ are diagonally dominant M-matrices so that $\{A + D_k\}$ is a convergent sequence of matrices in $\mathcal{Q}$ with the limit $A \in \overline{\mathcal{Q}}$. □

THEOREM 2.2. *Let A be an $n \times n$ nonnegative symmetric matrix. Then the following are equivalent:*

(1) *A is a special ultrametric matrix.*

(2) *There exists a nonnegatively edge-weighted path L with n vertices such that $A = C(L)$.*

(3) *There exists a nonnegatively edge-weighted tree T with n vertices such that $A = C(T)$.*

Proof. $1 \to 2$. We shall use induction with respect to n. The assertion is trivial for $n = 1$ and $n = 2$. Now let A be a special ultrametric $n \times n$ matrix and suppose the implication holds for all matrices of smaller order. We shall proceed similarly as in [5]. Let τ be the smallest off-diagonal entry of A. Then there exists a permutation matrix P such that

$$PAP^T = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} + \tau J,$$

where J is the $n \times n$ matrix of all ones. Both B_1 and B_2 are again special ultrametric matrices. By the induction hypothesis, $B_k = C(L_k)$ for some path L_k, $k = 1, 2$. Let now L be a path obtained by adding to $L_1 \cup L_2$ a new edge weighted by τ joining one end-vertex (it does not matter which) of L_1 with one end-vertex of L_2 and adding τ to all weights of edges in both L_1 and L_2. Since the permutation P corresponds to renumbering of the vertices, it follows that after such renumbering in L, we shall have $A = C(L)$.

$2 \to 3$. The proof is trivial.

$3 \to 1$. If $A = C(T)$ for some tree T, then by (4) and (5), both conditions (1) and (3) are fulfilled and A is thus special ultrametric. □

Remark 2.3. An analogous approach to ultrametric distance matrices was already mentioned in [1]. A different approach using distances in trees was considered in [4].

In the next theorem, we shall investigate the rank of a special matrix given by a weighted path.

THEOREM 2.4. *Let $L = (1, 2, \ldots, n)$ be a path with nonnegative weights w_i assigned to edges $(i, i+1)$, $i = 1, \ldots, n-1$. Set $w_0 = 0$, $w_n = 0$. Let S denote the set of those indices $k \in \{1, \ldots, n-1\}$ for which $w_k \geq \max(w_{k-1}, w_{k+1})$. Then the nullity $\nu(C(L))$ of $C(L)$ satisfies the inequality*

$$\nu(C(L)) \geq |S| + r, \tag{6}$$

where $r = 0, 1,$ or 2, according to whether none, one, or two of the numbers w_1 and w_{n-1} equal zero.

In (6), equality holds if all numbers $w_1, \ldots, w_{n-1}$ are distinct. In such a case, the matrix of order $n - \nu(C(L))$ obtained from $C(L)$ by deleting all rows and columns with indices in S (and eventually the zero row and column if $r = 1$) is nonsingular and its inverse is a weakly diagonally dominant M-matrix.

Proof. Let $k \in S$. Then clearly $(C(L))_{ik} = (C(L))_{i,k+1}$ for every $i = 1, \ldots, n$ so that the kth and $(k+1)$st rows of $C(L)$ coincide. If, in addition, one or both numbers w_1, w_{n-1} are equal to zero, one or both of the first and last rows of $C(L)$ are zero. It means that in the first case, the n-vector $(0, \ldots, 0, 1, -1, 0, \ldots, 0)^T$ with 1 as the kth coordinate belongs to the null-space N of $C(L)$; in the latter case one or both of the vectors $(1, 0, \ldots, 0)$, $(0, \ldots, 0, 1)$ belong to N. This implies (6) since all of these vectors are linearly independent.

To prove the last assertion, we shall use induction with respect to n. For $n = 2$ and $n = 3$, the assertion is correct. Now let $n > 3$ and suppose the assertion is true for paths with less than n vertices. Let m be the smallest of the weights in L. If $m = 0$, then either the first or the last row is zero, and after removing it, the shorter path satisfies the induction hypothesis and we are through, or this is not the case. Then the path L can be considered as a union of two disjoint shorter paths L_1 and L_2, each of which satisfies the induction hypothesis. The matrix $C(L)$ is then a direct sum of the matrices $C(L_1)$ and $C(L_2)$. The reduced matrix of $C(L)$ is then again a direct sum of the reduced matrices of $C(L_1)$ and $C(L_2)$. The assertion for L thus follows from the assertions for L_1 and L_2.

Now let m be positive. Subtracting m from each weight, we obtain a nonnegatively edge-weighted path $\widehat{L}$ with one weight zero. Clearly, $C(L) = C(\widehat{L}) + mJ$, where J is the matrix of ones. Let $w_1 = m$. Then the reduced matrix $\widehat{C}(L)$ of $C(L)$ has the form

$$\widehat{C}(L) = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{C}(\widehat{L}) \end{pmatrix} + m\widehat{J},$$

where $\widehat{C}(\widehat{L})$ and $\widehat{J}$ mean the reduced matrix of $C(\widehat{L})$ and the corresponding matrix of ones. The formula

$$\begin{pmatrix} m & me^T \\ me & Z + mee^T \end{pmatrix} \begin{pmatrix} m^{-1} + e^T Z^{-1} e & -e^T Z^{-1} \\ -Z^{-1} e & Z^{-1} \end{pmatrix} = I,$$

where e is the column vector of ones, applied to $Z = \widehat{C}(\widehat{L})$ shows that the inverse of the matrix $\widehat{C}(L)$ is a weakly diagonally dominant M-matrix since by the induction hypothesis this holds for $\widehat{C}(\widehat{L})$.

Since the case $w_{n-1} = m$ leads to a similar result, it remains to suppose that $m = w_k$ for some $k \in \{2, \ldots, n-2\}$. Then the path $\widehat{L}$ can be split into two nontrivial positively edge-weighted paths L_1 and L_2 satisfying the assumptions of the induction hypothesis. Applying the Woodbury formula

$$(P + mee^T)^{-1} = P^{-1} - m(1 + me^T P^{-1} e)^{-1} P^{-1} ee^T P^{-1}$$

to the case that P is the direct sum of the reduced matrices of $C(L_1)$ and $C(L_2)$ we again obtain that the inverse of the reduced matrix of $C(L)$ is a weakly diagonally dominant M-matrix. □

Remark 2.5. The example $n = 4$, $w_1 = 1$, $w_2 = 0$, $w_3 = 1$, where $\nu(L) = |S|$ but the numbers w_k are not distinct, shows that the distinctness of the w_k's is not necessary for the equality in (6). The example $n = 5$, $w_1 = 1$, $w_2 = w_3 = 0$, $w_4 = 2$, where $\nu(L) > |S|$, shows that equality in (6) is not always attained.

Remark 2.6. It is immediate that every strictly ultrametric matrix is a sum of a special ultrametric matrix and a diagonal matrix with positive diagonal entries. Theorem 2.4 allows one to decide in some cases under which conditions the sum of a special ultrametric matrix and a nonnegative diagonal matrix is nonsingular.

For instance, this will be the case when, in the notation of Theorem 2.4, all w_i are distinct, $w_1 w_{n-1} \neq 0$, and we add to $C(L)$ a nonnegative diagonal matrix with $|S|$ positive diagonal entries in the positions d_k for $k \in S$.

Remark 2.7. One can easily show that if the weights of edges of the path L are mutually distinct and different from zero, the Moore–Penrose generalized inverse of $C(L)$ is obtained from the inverse of the reduced matrix $\widehat{C}(L)$ by what is in a sense an inverse operation to reducing repeated rows and columns: where we had reduced the kth row and column and left the $(k+1)$st, the corresponding row and column of $(\widehat{C}(L))^{-1}$ appearing like

$$\begin{pmatrix} * & a_1 & * \\ a_1^T & a_{22} & a_3^T \\ * & a_3 & * \end{pmatrix},$$

we put in $(C(L))^+$

$$\begin{pmatrix} * & \frac{1}{2}a_1 & \frac{1}{2}a_1 & * \\ \frac{1}{2}a_1^T & \frac{1}{4}a_{22} & \frac{1}{4}a_{22} & \frac{1}{2}a_3^T \\ \frac{1}{2}a_1^T & \frac{1}{4}a_{22} & \frac{1}{4}a_{22} & \frac{1}{2}a_3^T \\ * & \frac{1}{2}a_3 & \frac{1}{2}a_3 & * \end{pmatrix}.$$

Thus this Moore–Penrose inverse is not an M-matrix but can be considered as a generalized M-matrix (the diagonal blocks are positive matrices of rank-one), again weakly diagonally dominant.

3. The nonsymmetric case. In this section, we shall try to generalize the approach of the first section to nonsymmetric ultrametric matrices. Let us mention that in this sense generalized ultrametric matrices were introduced in [4] and [6] by adding further restrictions to (1) and (2). Let us first give an example.

Example 3.1. Let $a < b < c < d$ be positive numbers. Then the nonsymmetric matrix

$$A = \begin{pmatrix} d & a & a \\ d & d & b \\ c & c & c \end{pmatrix}$$

is ultrametric (in the sense that only (1) and (2) with possible equality are supposed) and its inverse is

$$A^{-1} = \frac{1}{\Delta} \begin{pmatrix} c(d-b) & 0 & -a(d-b) \\ -c(d-b) & c(d-a) & -d(b-a) \\ 0 & -c(d-a) & d(d-a) \end{pmatrix},$$

where $\Delta = c(d-a)(d-b)$.

Thus A^{-1} is an M-matrix; however, it is not row-diagonally dominant since the sum in the second row is $-(d-c)(b-a)$.

Remark 3.2. Example 3.1 shows that it does not suffice to restrict oneself only to (1) and (2) to obtain a similar result as in Theorem A.

Remark 3.3. Observe that the matrix A from Example 3.1 arises from a similar construction as $A(\overrightarrow{G})$ from the directed graph $\overrightarrow{G}$ having the vertices $1, 2, 3$ and edges $(1, 2)$ with weight a, $(2, 1)$ with weight d, $(2, 3)$ with weight c, and $(3, 2)$ with weight b.

However, in the following two cases this construction works.

THEOREM 3.4. *Let $n \geq 1$ and let $a_1, \ldots, a_{n-1}$ be nonnegative numbers not exceeding one. Define an $n \times n$ matrix $U = (u_{ik})$ by*

$$\begin{aligned} u_{ii} &= 1,\ i = 1, \ldots, n, \\ u_{ik} &= \min_{j, i \leq j < k} a_j \text{ for } i < k, \\ u_{ik} &= 0 \text{ for } i > k. \end{aligned}$$

Then the upper triangular nonsingular matrix U is ultrametric and its inverse is an M-matrix which is weakly diagonally dominant in both rows and columns.

In addition, if e is the column vector of n ones, then

$$(\min_i a_i) e^T U^{-1} e \leq 1.$$

Proof. It is easily checked that the matrix U is a generalized ultrametric matrix in the sense of [6, Definition 2.5]. Therefore, the result follows from [6, Theorem 3.6]. □

THEOREM 3.5. *Let $n \geq 2$ and let $a_1, \ldots, a_n$ be nonnegative numbers. Let $\overrightarrow{G}$ be the directed graph with vertices $1, \ldots, n$ and edges $(1, 2), (2, 3), \ldots, (n-1, n), (n, 1)$. Assign to the edge $(k, k+1)$ the weight a_k, $k = 1, \ldots, n-1$, and to $(n, 1)$ the weight a_n. Define an $n \times n$ matrix $B = (b_{ik})$ by $b_{ii} = \max_j a_j$ for all i and b_{ik} as the minimum of all weights in the path from i to k in $\overrightarrow{G}$ if $i \neq k$.*

Then B is an ultrametric matrix and is nonsingular with the exception of the case $a_1 = a_2 = \cdots = a_n$, and in the case of nonsingularity, the inverse of B is an M-matrix which is both row- and column-diagonally dominant.

In addition,

$$\det B \geq (\max_k a_k - \min_k a_k)^n,$$

and if B is nonsingular and e is the vector of all ones, then

$$(\min_k a_k) e^T B^{-1} e \leq 1.$$

Proof. It is evident that in the case $a_1 = \cdots = a_n$ the matrix B is singular. Let this not be the case. By a cyclic transformation of the indices, we can arrange that $a_1 = \min_i a_i$. We can also assume that $\max_i a_i = 1$, so that $a_1 < 1$. Now define a matrix $C = (c_{ij})$, $i, j = 0, \ldots, n-1$, by

$$C = (1 - a_1)^{-1}(B - a_1 e e^T). \tag{7}$$

Then for some column vector u with $n-1$ coordinates $u_1, \ldots, u_{n-1}$ and some $(n-1) \times (n-1)$ matrix A,

$$C = \begin{pmatrix} 1 & 0 \\ u & A \end{pmatrix}.$$

The matrix A satisfies the assumptions of Theorem 3.4. Thus, A^{-1} is an upper triangular M-matrix with all row-sums and all column-sums nonnegative. Let $v = (v_1, \ldots, v_{n-1})^T$ be the last column of A.

Then

$$\begin{aligned} (1 - a_1)u_k &= \min(a_{k+1}, a_{k+2}, \ldots, a_n) - a_1 \quad \text{for } k = 1, \ldots, n-1, \\ (1 - a_1)v_k &= \min(a_{k+1}, \ldots, a_{n-1}) - a_1 \quad \text{for } k = 1, \ldots, n-2, \\ v_{n-1} &= 1. \end{aligned} \tag{8}$$

It follows that $0 \leq u \leq v$.

If $a_n < a_{n-1}$, let t be the index for which $a_n \geq a_t$ but $a_n < a_k$ for $k = t+1, \ldots, n-1$. Thus, $t < n-1$. If $a_n \geq a_{n-1}$, set $t = n-1$.

We have then $u_i = v_i$ for $i = 1, \ldots, t-1$, but if $t < n-1$, then $u_t < v_t$, as well as $u_t = u_{t+1} = \cdots = u_{n-1}$.

Let w be the vector $w = (0, \ldots, 0, 1)^T$ with $n-1$ coordinates. Then $v = Aw$, which implies

$$(A^{-1}u)_k \geq (A^{-1}v)_k \geq 0 \quad \text{for } k = 1, \ldots, t-1,$$

whereas, because of the row-diagonal dominance of A^{-1},

$$(A^{-1}u)_k = u_{n-1}(A^{-1}e)_k \geq 0 \quad \text{for } k = t, t+1, \ldots, n-1.$$

Thus $A^{-1}u \geq 0$ and the matrix

$$C^{-1} = \begin{pmatrix} 1 & 0 \\ -A^{-1}u & A^{-1} \end{pmatrix}$$

is an M-matrix.

We shall show that C^{-1} is column-diagonally dominant. This is true for the last $n-1$ columns by Theorem 3.4. To show that $1 - e^T A^{-1} u \geq 0$ as well, observe that from $e^T A^{-1} \geq 0$ and $v - u \geq 0$ we have $(e^T A^{-1} v) \geq (e^T A^{-1} u)$, which implies

$$\begin{aligned} 1 - (e^T A^{-1} u) &\geq 1 - (e^T A^{-1} v) \\ &= 1 - e^T w \\ &= 0. \end{aligned}$$

Let us show that B^{-1} is also a column-diagonally dominant M-matrix. By (7), if we denote $(1 - a_1)C$ as Q,

$$B^{-1} = Q^{-1} - \frac{a_1}{1 + a_1 e^T Q^{-1} e} Q^{-1} e e^T Q^{-1}. \tag{9}$$

Thus

$$\begin{aligned} e^T B^{-1} &= \frac{1}{1 + a_1 e^T Q^{-1} e} e^T Q^{-1} \\ &= \frac{1}{(1 - a_1)(1 + a_1 (1 - a_1)^{-1} e^T C^{-1} e)} e^T C^{-1}, \end{aligned}$$

which is indeed a nonnegative row vector.

Let us show now that B^{-1} is also row-diagonally dominant. This follows from the fact that the matrix B^T satisfies the assumptions of the theorem and is thus column-diagonally dominant.

It remains to prove the last two inequalities. By (6),

$$BC^{-1} = (1 - a_1) I + a_1 e e^T C^{-1}.$$

Thus

$$\begin{aligned} \det BC^{-1} &= (1 - a_1)^n + (1 - a_1)^{n-1} a_1 \operatorname{tr} e e^T C^{-1} \\ &= (1 - a_1)^n + (1 - a_1)^{n-1} a_1 e^T C^{-1} e \\ &\geq (1 - a_1)^n \end{aligned}$$

since the last term is nonnegative. Observing that $\det C = 1$, the first inequality follows.

To prove the second inequality, we can assume that $a_1 > 0$. By (9), if we denote $Z = e^T Q^{-1} e$,

$$\begin{aligned} e^T B^{-1} e &= Z - \frac{a_1}{1 + a_1 Z} Z^2 \\ &= \frac{Z}{1 + a_1 Z} \\ &\leq \frac{1}{a_1} \end{aligned}$$

since Z is nonnegative. □

Remark 3.6. The last inequality also follows from the previous part of the theorem and from Theorem 3.6 in [4].

REFERENCES

[1] M. Fiedler, *Ultrametric sets in Euclidean point spaces*, Electron. J. Linear Algebra, 3 (1998), pp. 23–30.

[2] M. Fiedler, C. R. Johnson, and T. L. Markham, *Notes on inverse M-matrices*, Linear Algebra Appl., 91 (1987), pp. 75–81.

[3] S. Martínez, G. Michon, and J. San Martín, *Inverse of strictly ultrametric matrices are of Stieltjes type,* SIAM J. Matrix Anal. Appl., 15 (1994), pp. 98–106.

[4] J. J. McDonald, M. Neumann, H. Schneider, and M. J. Tsatsomeros, *Inverse M-matrix inequalities and generalized ultrametric matrices*, Linear Algebra Appl., 220 (1995), pp. 321–341.

[5] R. Nabben and R. S. Varga, *A linear algebra proof that the inverse of a strictly ultrametric matrix is a strictly diagonally dominant Stieltjes matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 107–113.

[6] R. Nabben and R. S. Varga, *Generalized ultrametric matrices—A class of inverse M-matrices*, Linear Algebra Appl., 220 (1995), pp. 365–390.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 114–144

FRACTION-FREE COMPUTATION OF MATRIX RATIONAL INTERPOLANTS AND MATRIX GCDs*

BERNHARD BECKERMANN† AND GEORGE LABAHN‡

Abstract. We present a new set of algorithms for computation of matrix rational interpolants and one-sided matrix greatest common divisors. Examples of these interpolants include Padé approximants, Newton–Padé, Hermite–Padé, and simultaneous Padé approximants, and more generally M-Padé approximants along with their matrix generalizations. The algorithms are fast and compute all solutions to a given problem. Solutions for all (possibly singular) subproblems along offdiagonal paths in a solution table are also computed by stepping around singular blocks on a path corresponding to "closest" regular interpolation problems.

The algorithms are suitable for computation in exact arithmetic domains where growth of coefficients in intermediate computations is a central concern. This coefficient growth is avoided by using fraction-free methods. At the same time, the methods are fast in the sense that they are at least an order of magnitude faster than existing fraction-free methods for the corresponding problems. The methods make use of linear systems having a special striped Krylov structure.

Key words. Hermite–Padé approximant, simultaneous Padé approximant, striped Krylov matrices, fraction-free arithmetic

AMS subject classifications. 65D05, 41A21

CR subject classification. G.1.2

PII. S0895479897326912

1. Introduction.

A number of methods are available for the computation of various rational interpolation problems. Consider, for example, the simplest case of rational interpolation, that of Padé approximation. One can compute a Padé approximant by setting up a linear system of equations and using Gaussian elimination to solve the system. The number of operations in this case is $O(n^3)$, where n is the number of equations in the system. However, since the coefficient matrix of this system has a special Hankel or Toeplitz structure, there exist more efficient algorithms for these computations. Examples include fast $O(n^2)$ algorithms and even superfast $O(n\ log^2 n)$ algorithms (cf. Brent, Gustavson, and Yun [15] or Cabay and Choi [18] in addition to many others). A similar statement can also be made for other matrix-like Padé approximation problems. Here one finds fast or superfast algorithms for computing Hermite–Padé and simultaneous Padé approximants, e.g., Van Barel and Bultheel [51, 52], Cabay, Labahn, and Beckermann [19], Cabay and Labahn [22], and Beckermann and Labahn [7, 8, 9]. In all the examples above the algorithms both are fast and avoid problems associated to the existence of singular blocks in an associated solution table. Alternatively, one may obtain fast algorithms for Padé approximation by translating to polynomial language some of the algorithms developed for structured matrices having a small displacement rank, for example, those found in Heinig and Rost [36].

*Received by the editors September 9, 1997; accepted for publication (in revised form) by G. Golub October 6, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/32691.html

†Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France (bbecker@ano.univ-lille1.fr).

‡Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (glabahn@daisy.uwaterloo.ca).

However, at the implementation level, these algorithms have drawbacks that limit their effectiveness. For example, suppose one is working in a floating point environment. Since the previously mentioned algorithms assume exact arithmetic, implementations in floating point domains do not take into consideration roundoff error. In these cases the computations all suffer from some degree of numerical instability. It is only recently that a number of new algorithms have appeared that are both fast and stable in a numerical setting, for example, [6, 20, 23, 34, 53] for Padé problems, [13, 14, 35, 27, 28] for Toeplitz and Hankel systems, [24, 31, 33] and further references mentioned in [32] for systems with displacement structure.

The roundoff problems encountered when implementing in floating point domains do not appear when implementing in exact arithmetic environments, for example, in computer algebra systems such as Maple or Mathematica. However, even in these cases it turns out that most existing algorithms have problems that also limit their usefulness. In the case of numerical arithmetic, the efficient algebraic algorithms are fast but sometimes suffer from a lack of accuracy. In exact domains these algorithms are accurate but often lack efficiency. For example, in Czapor and Geddes [26], it is shown that a minor modification of Gaussian elimination computes Padé approximants more efficiently than Levinson's algorithm, that is, in this case an $O(n^3)$ algorithm is faster than an $O(n^2)$ algorithm. The reason for this is simple to explain: in exact arithmetic domains, operations such as addition or multiplication do not have a constant cost. Rather, the arithmetic cost depends on the size of the components and so we need to measure bit complexity rather than operations complexity. The (possibly exponential) growth in the cost of intermediate arithmetic operations may be observed in particular when the domain of coefficients is a field of quotients $\mathbf{Q}(a_1, \ldots, a_n)$, where $\mathbf{Q}$ is the field of rational numbers (or an algebraic extension of the rational numbers) and $a_1, \ldots, a_n$ are indeterminants, a typical situation for symbolic computation in computer algebra systems. In order to compute in these domains one must try for a low complexity while at the same time keeping the components of the arithmetic operations at a small size. In addition, the cost of keeping the components of the arithmetic operations at a small size must be done in an efficient manner.

In this paper we present a new fast algorithm for efficiently computing *all* solutions to a variety of matrix rational interpolation problems along with one-sided matrix greatest common divisors. The interpolation problems covered include the partial realization problem for matrix power series and Padé, Newton–Padé, Hermite–Padé, simultaneous Padé, M-Padé, and multipoint-Padé approximation problems and their matrix generalizations. The connection between rational interpolation and greatest common divisor problems has been known for a long time and has been successfully exploited in the scalar case.

The algorithm is recursive, providing also solutions for all (possibly singular) subproblems along offdiagonal paths in a solution table. Here singular subproblems are not skipped over via pseudodivisions or look-ahead techniques, but by following [5, 8, 51] we step around singular blocks on some path corresponding to "closest" regular interpolation problems. This leads to an additional gain in complexity if there are only few regular subproblems, a rather typical situation for GCD computations.

Rather than present the algorithm for a field, we assume that the coefficient domain is an integral domain and give a *fraction-free* algorithm for efficiently computing solutions to these interpolation problems. The concept of fraction-free implies that arithmetic operations remain inside the integral domain, rather than requiring that one does arithmetic in its quotient field. This avoids the need for costly great-

est common divisor computations required for such rational operations, making the algorithm suitable for implementation in computer algebra systems. This allows for efficient computation of matrix interpolation problems in the case of parameterized data. Such computations also appear in such diverse applications as the Gfun package of Salvy and Zimmermann [49] for determining recurrences relations, factorization of linear differential operators [54], and computation of matrix normal forms [11, 55].

The algorithm presented here is at least an order of magnitude faster than applying the fraction-free algorithm of Bareiss [2], which is based on Gaussian elimination. This is the only known fraction-free method that will also work for the rational interpolation problems studied here. However, there have been fraction-free algorithms that are faster than Bareiss's algorithm in some special cases. For Padé approximation the algorithm of Cabay and Kossowski [21] makes use of the close relationship between Padé approximation and polynomial remainder sequences to obtain an improved fraction-free algorithm. For matrix Padé approximation the algorithm of Beckermann, Cabay, and Labahn [10] uses a recursive procedure based on modified Schur complements of the associated linear equations to improve on Gaussian elimination. Finally the subresultant GCD algorithm of Brown and Traub [16] and Collins [25] gives a fast greatest common divisor algorithm in the case of scalar polynomials. In all cases our algorithm is also faster or at least as fast as those mentioned in special cases.

In terms of linear algebra, we can view our problem as determining nullspaces of rectangular striped Krylov matrices and their principal submatrices in a fast and fraction-free manner. Notice that this task includes the fraction-free computation of vectors required for explicit inversion formulas, for example, for Hankel, Toeplitz, or Sylvester matrices and their block counterparts [36] (for an interpretation in terms of Padé problems see, for instance, [40, 42]). In the regular case where all principal submatrices are nonsingular, it is possible to look for fraction-free counterparts of known algorithms for structured matrices, for example, the fast Gaussian elimination scheme, that is, Schur type algorithms, or Levinson type methods [32, 36]. Recent extensions [24, 31, 32, 33] also allow for pivoting in order to overcome problems with singularities. However, in our setting, additional transformation techniques would be necessary in order to allow for pivoting. Our alternate approach is motivated by the fact that transformations may lead to a significant increase of complexity of the input data and, in any case, this cannot be done in a fraction-free manner. In addition, the kind of pivoting used in these extensions does not allow us to solve subproblems of our initial interpolation problem.

The rest of the paper is organized as follows. Section 2 introduces the rational interpolation problems defined in terms of a "special rule." Section 3 shows that the rational interpolation problems can be interpreted in linear algebra terms as solving a linear system of equations having a striped Krylov matrix as a coefficient matrix. Some regularity properties are studied in section 4, while section 5 introduces Mahler systems, a matrix of determinant polynomials which give a basis for our solution spaces. Section 6 gives a fraction-free recursion along a so-called perfect path, while section 7 considers the more difficult nonperfect case. Section 8 shows how the algorithm from the previous section can be used to compute the one-sided GCD of two matrix polynomials. The last section includes a conclusion along with a discussion of future research directions.

2. Rational interpolation and their linear systems. Let $\mathbb{D}$ be an integral domain with $\mathbb{Q}$ its quotient field. Let $\mathcal{V}$ be an infinite dimensional vector space over $\mathbb{Q}$ having a basis $(\omega_u)_{u=0,1,\ldots}$ with $(c_u)_{u=0,1,\ldots}$ its dual basis (i.e., a set of linear

functionals on $\mathcal{V}$ satisfying $c_u(\omega_v) = \delta_{u,v}$). Thus every element f of $\mathcal{V}$ can be written as

$$f = f_0 \cdot \omega_0 + f_1 \cdot \omega_1 + f_2 \cdot \omega_2 + \cdots \tag{2.1}$$

with $c_u(f) = f_u$. We define the *order* of a nontrivial element f of $\mathcal{V}$ by

$$ord(f) = n \quad \text{iff} \quad c_0(f) = \cdots = c_{n-1}(f) = 0 \quad \text{and} \quad c_n(f) \neq 0,$$

and $ord(0) = +\infty$.

We also assume that we have a *special* element z that acts on $\mathcal{V}$ via a special multiplication rule

$$c_u(z \cdot f) = c_{u,0} \cdot c_0(f) + \cdots + c_{u,u} \cdot c_u(f) \text{ with } c_{u,v} \in \mathbb{D}. \tag{2.2}$$

This special rule can be viewed as a type of Leibniz chain rule. The special rule allows us to define a multiplication $p(z) \cdot f$ for any polynomial $p \in \mathbb{Q}[z]$ and $f \in \mathcal{V}$, making $\mathcal{V}$ an infinite dimensional module over $\mathbb{Q}[z]$.

In this paper we will study the following interpolation problem with polynomial linear combinations of functions $f^{(1)}, \ldots, f^{(m)}$, where $m \geq 2$.

DEFINITION 2.1 (rational interpolation problem). *Let $f = [f^{(1)}, \ldots, f^{(m)}]$ be a vector of m elements from $\mathcal{V}$, σ a positive integer, and $\vec{n} = (\vec{n}^{(1)}, \ldots, \vec{n}^{(m)})$ a multi-index. Determine a vector $p(z) = [p^{(1)}(z), \ldots, p^{(m)}(z)]^T$ of polynomials in z, with each $p^{(\ell)}(z)$ having degree bounded by $\vec{n}^{(\ell)} - 1$, and satisfying the order condition*

$$\text{ord}\,(f \cdot p(z)) = \text{ord}\,(f^{(1)} \cdot p^{(1)}(z) + \cdots + f^{(m)} \cdot p^{(m)}(z)) \geq \sigma. \tag{2.3}$$

In this case, $p(z)$ will be referred to as solution of type $(\sigma, \vec{n})$. ∎

Example 2.1 (Hermite–Padé approximants [45, 46, 47, 48, 51]). Let $\mathcal{V}$ be the space $\mathbb{Q}[[z]]$ of formal power series around 0 with basis $(z^u)_{u=0,1,\ldots}$ and let the $c_{i,j}$ be defined by $c_{i,j} = \delta_{i-1,j}$. Then the special multiplication rule is simply the standard multiplication by z. With $\sigma = |\vec{n}| - 1$, where $|\vec{n}| := \vec{n}^{(1)} + \cdots + \vec{n}^{(m)}$, the interpolation problem (2.3) is the Hermite–Padé approximation problem of type $\vec{n}$, introduced by Hermite in 1873. When $m = 2$ and $f^{(2)} = -1$, this gives the classical Padé approximant. Hermite–Padé approximation also includes other classical approximation problems such as algebraic approximants ($f = (1, g, g^2, \ldots, g^{m-1})$) and G^3J approximants ($m = 3, f = (g', g, 1)$). We refer the reader to [1] for some additional examples. ∎

Before giving further examples for the rational interpolation problem of Definition 2.1, let us have a closer look at the underlying system of linear equations. Notice first that we may rewrite the special multiplication rule (2.2) in terms of linear algebra. We denote by $\mathbf{C} = (c_{u,v})_{u,v=0,1,\ldots}$ the lower triangular infinite matrix determined by the coefficients of (2.2) and by $\mathbf{C}_\sigma$, $\sigma \geq 0$ its principal submatrix of order σ. Furthermore, for each $f \in \mathcal{V}$ and nonnegative integer σ we associate a vector of coefficients

$$\mathbf{F}_\sigma = [c_0(f), \ldots, c_{\sigma-1}(f)]^T, \qquad \mathbf{F} = [c_0(f), c_1(f), c_2(f), \ldots]^T. \tag{2.4}$$

Note that we begin our row and column enumeration at 0. Then in matrix terms the special multiplication rule can be interpreted as

$$\mathbf{C}_\sigma \cdot \mathbf{F}_\sigma = [c_0(z \cdot f), \ldots, c_{\sigma-1}(z \cdot f)]^T$$

and more generally

$$p(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma = [c_0(p(z) \cdot f), \ldots, c_{\sigma-1}(p(z) \cdot f)]^T$$

for any polynomial $p(z) \in \mathbb{Q}[z]$ and for any nonnegative integer σ.

For our rational interpolation problem we can associate as in (2.4) to f the vectors of values $\mathbf{F}_\sigma = (\mathbf{F}_\sigma^{(1)}, \ldots, \mathbf{F}_\sigma^{(m)})$, $\mathbf{F}_\sigma^{(i)} = [c_0(f^{(i)}), \ldots, c_{\sigma-1}(f^{(i)})]^T$, $i = 1, \ldots, m$. Then the order condition (2.3) in Definition 2.1 may be rewritten as

$$p^{(1)}(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma^{(1)} + \cdots + p^{(m)}(\mathbf{C}_\sigma) \cdot \mathbf{F}_\sigma^{(m)} = 0.$$

In order to obtain explicitly a system of equations, we introduce

$$\mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma) = \left[\begin{array}{cccc|c|cccc} \mathbf{F}_\sigma^{(1)} & \mathbf{C}_\sigma\mathbf{F}_\sigma^{(1)} & \cdots & \mathbf{C}_\sigma^{\vec{n}^{(1)}-1}\mathbf{F}_\sigma^{(1)} & \cdots & \mathbf{F}_\sigma^{(m)} & \cdots & \mathbf{C}_\sigma^{\vec{n}^{(m)}-1}\mathbf{F}_\sigma^{(m)} \end{array}\right],$$

a striped Krylov matrix of size $\sigma \times |\vec{n}|$. Furthermore, we identify a vector polynomial $p(z) = [p^{(1)}(z), \ldots, p^{(m)}(z)]^T$ of the form $p^{(i)}(z) = \sum_{j=0}^{\vec{n}^{(i)}-1} p_j^{(i)} z^j$, $i = 1, \ldots, m$, with its *coefficient vector*

$$\mathbf{P} = [p_0^{(1)}, \ldots, p_{\vec{n}^{(1)}-1}^{(1)} | \cdots\cdots | p_0^{(m)}, \ldots, p_{\vec{n}^{(m)}-1}^{(m)}]^T.$$

Then $p(z)$ is a solution of type $(\sigma, \vec{n})$ iff its coefficient vector $\mathbf{P}$ satisfies

$$\mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma) \cdot \mathbf{P} = 0. \tag{2.5}$$

In the remaining part of this section, further special cases of the interpolation problem of Definition 2.1 are discussed.

Example 2.2 (vector and power Hermite–Padé approximants [7, 8, 52]). Let $\mathcal{V}$ be the space $\mathbb{Q}^s[[z]]$ of $1 \times s$ vectors of formal power series around 0. A basis for $\mathcal{V}$ is given by $\omega_u = \omega_{n \cdot s+k} = z^n \cdot \vec{e}_{k+1}$ with $0 \le k < s$, where $\vec{e}_k$ denotes the kth unit vector. Let $c_{i,j}$ be defined by $c_{i,j} = \delta_{i-s,j}$. Then the special multiplication rule is again the standard scalar multiplication by z, viewed as a scalar. In this case, problem (2.3) with $\sigma = |\vec{n}| - 1$ is the vector Hermite–Padé approximation problem of type $\vec{n}$. This interpolation problem appears, for example, in the new Van Hoeij algorithm for the factorization of differential operators [54].

We can also let $\mathcal{V}$ be the space $\mathbb{Q}[[x]]$ of formal power series around 0 with basis $\tilde{\omega}_u = \omega_u(x^s) \cdot [1, x, ..., x^{s-1}]^T$ with the ω_u from above. Let the $c_{i,j}$ again be defined by $c_{i,j} = \delta_{i-s,j}$. Then the special rule is multiplication by $z = x^s$. In this case, problem (2.3) is then to find polynomials $p^{(i)}$ in z with the correct degree bounds (with respect to z of course) and satisfying the equation

$$f^{(1)} \cdot p^{(1)}(x^s) + \cdots + f^{(m)} \cdot p^{(m)}(x^s) = r_\sigma x^\sigma + r_{\sigma+1} x^{\sigma+1} + \cdots.$$

This is the power Hermite–Padé approximation problem. Note that this problem is the same as the first part of our example obtained by multiplying both sides of every basis equation (2.1) by the vector $[1, x, \ldots, x^{s-1}]^T$. This is the "$s$-trick" described in [7, 8]. Besides vector Hermite–Padé approximants, power Hermite–Padé approximation can be used to represent (and hence to compute) matrix Padé approximants [41] and simultaneous Padé approximants [45] along with their matrix generalizations [40].

For instance, solutions of type $(|\vec{n}| - s, \vec{n})$ are required as building blocks for matrix Padé approximants (see [8]). □

Example 2.3 (linearized rational interpolation). Suppose that we have a sequence of not necessarily distinct knots $x_i \in \mathbb{D}$ and a function g being sufficiently smooth in a neighborhood of these knots. The linearized rational interpolation problem of type $[L/M]$ (see, e.g., [1]) consists of finding polynomials p and q of degree at most L and M, respectively, such that $-p + g \cdot q = [-1, g] \cdot [p, q]^T$ vanishes at $x_0, \ldots, x_{L+M}$, counting multiplicities.

Let $\mathcal{V}$ be the space of all formal Newton series in z with respect to the given knots $x_0, x_1, \ldots$. Note that a basis of $\mathcal{V}$ (or some finite dimensional counterpart) may be constructed using either Newton, Lagrange, or Hermite polynomials. Therefore, there are several choices for the sequence of linear functionals $(c_u)_{u=0,1,2,\ldots}$ in order to reformulate the linearized rational interpolation problem using the formalism of Definition 2.1. For instance, one may take as c_v the vth divided difference $[x_0, \ldots, x_v]$. It is easy to verify that for these linear functionals the special multiplication rule (2.2) holds, with $c_{i,j} = \delta_{i,j} \cdot x_i + \delta_{i-1,j}$, $i > 0$, and $c_{0,0} = x_0$.

If the knots $x_0, x_1, \ldots$ are distinct, then the simpler choice $c_v(g) = g(x_v)$ leads to the special multiplication rule (2.2) with $c_{i,j} = \delta_{i,j} \cdot x_i$. In the case of not necessarily distinct knots, we may more generally consider the values of the successive derivatives, i.e., $c_v(g) = g^{(\rho_v)}(x_v)/(\rho_v!)$, where ρ_v denotes the multiplicity of x_v in $(x_0, x_1, \ldots, x_{v-1})$. Here the components $c_{i,j}$ for the special multiplication rule is based on (some permutation of) a Jordan normal form matrix $\mathbf{C}$. □

In Example 2.3 we mentioned the case $m = 2$ with $f = [-1, g]$. The case of general f has also been discussed by several authors.

Example 2.4 (M-Padé approximants; see [3, 4, 5, 44, 45]). Suppose that we have a sequence of not necessarily distinct knots $x_i \in \mathbb{D}$. Let again $\mathcal{V}$ be the space of all formal Newton series in z with basis elements $\omega_u = (z - x_0) \cdots (z - x_{u-1})$, with the dual basis consisting of the vth divided difference $c_v = [x_0, \ldots, x_v]$, $v \geq 0$ (the corresponding special multiplication rule is given in Example 2.3). Solutions of type $(|\vec{n}| - 1, \vec{n})$ of our interpolation problem of Definition 2.1 are known as M-Padé approximants of type $\vec{n}$. One can also obtain a vector M-Padé problem using the same method as described in Example 2.2.

An important application for M-Padé approximation is the generalized Richardson extrapolation process (GREP) where one tries to approximate the limit of some sequence $(g(x_j))_{j=0,1,\ldots}$ with distinct $x_0, x_1, \ldots$ by interpolating with help of the function 1 and polynomial linear combinations of some functions $g_1, \ldots, g_m$ [50]. Here the sequence of knots and the functions $g_1, \ldots, g_m$ are chosen such that $(x_j^\ell \cdot g_i(x_j))_{j=0,1,\ldots}$ tends to zero for all i, ℓ. Thus the (scalar) ratio between the first and the second component of an M-Padé approximant of type $[1, 1, n_1, \ldots, n_m]$ with respect to the system $[-1, g, g_1, \ldots, g_m]$ is used for approximating the desired limit. Note that, due to the available data, the linear functionals $c_v(f) = f(x_v)$ may be preferable. □

3. The linear algebra background. For the remainder of this paper we will assume that we have a fixed lower triangular infinite matrix $\mathbf{C}$ and a fixed $\mathbf{F} = [\mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(m)}]$ of infinite coefficient vectors for elements $f^{(1)}, \ldots, f^{(m)}$ of $\mathcal{V}$. Let $\vec{n}$ be a multi-index and σ a positive integer. In order to simplify notation, we will simply drop $\mathbf{C}_\sigma$ and $\mathbf{F}_\sigma$ from our notation when using the striped Krylov matrices, i.e., we will write $\mathbf{K}(\vec{n}, \sigma) = \mathbf{K}(\vec{n}, \mathbf{C}_\sigma, \mathbf{F}_\sigma)$ for the associated striped Krylov matrix of size $\sigma \times |\vec{n}|$. Note that since $\mathbf{C}$ is lower triangular, the matrix $\mathbf{K}(\vec{n}, j)$ for $j < \sigma$ consists of the first j rows of $\mathbf{K}(\vec{n}, \sigma)$.

We have seen in section 2 that finding a solution p of type $(|\vec{n}|-1,\vec{n})$ of the interpolation problem of Definition 2.1 with exact order $|\vec{n}|-1$ is equivalent to solving the system of linear equations

$$\mathbf{K}(\vec{n},|\vec{n}|)\cdot\bar{\mathbf{P}}=[0,\ldots,0,1]^T \tag{3.1}$$

for the corresponding coefficient vector $\bar{\mathbf{P}}$. In our case, we look for solutions with coefficients not in the fraction field $\mathbb{Q}$ but in the integral domain $\mathbb{D}$. This is accomplished by means of Cramer's rule over $\mathbb{Q}$, giving a solution

$$\mathbf{K}(\vec{n},|\vec{n}|)\cdot\mathbf{P}=\det\left(\mathbf{K}(\vec{n},|\vec{n}|)\right)\cdot[0,\ldots,0,1]^T, \tag{3.2}$$

with $\mathbf{P}$ being a vector having only coefficients from $\mathbb{D}$. Here, the determinant representation of $\mathbf{P}$ furnished by Cramer's rule is quite useful and will be studied in section 5. For instance, this representation enables us to obtain bounds for the size (in bits) of such a solution in terms of the initial size of the components of the series using Hadamard's inequality [30, p. 299]

$$|\det(a_{j,k})|\le\prod_j\left[\sum_k|a_{j,k}|^2\right]^{1/2}. \tag{3.3}$$

In fact, Cramer solutions are also furnished by applying fraction-free Gaussian elimination [2, 30] on (3.1). Our contribution is to show in the second part of this paper that Cramer solutions may be obtained in a more efficient way.

It seems that in general Cramer solutions may be considered as the "simplest" solutions of (3.1) with coefficients in $\mathbb{D}$. Of course, one may construct examples where additional simplifications occur, but it can be quite expensive to detect such further simplifications. To illustrate this statement, take for instance the problem of computing a scalar GCD. Here several methods exist which avoid fractions (for a summary, see, e.g., [30, section 7.2]), for instance, the reduced polynomial remainder sequence (PRS) algorithm. However, only the subresultant GCD algorithm of Brown and Traub [16] and Collins [25] gives "maximal" Cramer solutions.

We recall that, depending on the matrix $\mathbf{C}$ defined by our special rule (2.2), we may obtain a system of equations with a matrix of coefficients having a quite particular structure, for instance, the following.

Example 3.1 (Toeplitz and generalized Sylvester matrices). Let $\mathbf{C}$ be the classical lower shift matrix, that is, $c_{i,j}=\delta_{i-1,j}$. Then $\mathbf{K}(\vec{n},\sigma)$ is a generalized Sylvester matrix [40] with each stripe a lower triangular Toeplitz matrix. If $m=2$ and

$$\mathbf{F}=\begin{bmatrix}p_0 & \cdots & p_k & 0 & \cdots & 0\\ q_0 & \cdots & \cdots & q_n & 0\cdots & 0\end{bmatrix}^T,$$

then

$$\mathbf{K}((n,k),n+k) = \left[\begin{array}{cccc|cccc} p_0 & 0 & \cdots & 0 & q_0 & 0 & \cdots & 0 \\ & p_0 & \ddots & \vdots & & q_0 & \ddots & \vdots \\ \vdots & & \ddots & 0 & \vdots & & \ddots & 0 \\ \vdots & & & p_0 & \vdots & & & q_0 \\ p_k & & & \vdots & q_n & & & \vdots \\ 0 & \ddots & & \vdots & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & p_k & 0 & \cdots & 0 & q_n \end{array}\right]$$

is the classical Sylvester matrix for the polynomials $p(z) = \sum_{i=0}^{k} p_{k-i} z^i$ and $q(z) = \sum_{i=0}^{n} q_{n-i} z^i$. Sylvester's matrix is heavily used in the fraction-free computation of the GCD of two polynomials (cf. [30]). □

Besides (striped) Toeplitz or Sylvester matrices associated to (Hermite–)Padé approximation, striped Krylov matrices with lower triangular $\mathbf{C}$ may be used to represent other well-known structured matrices. For instance, for vector or power Hermite–Padé approximants (Example 2.2) we may choose as $\mathbf{C}$ the sth power of the lower shift matrix. Then $\mathbf{K}(\vec{n},\sigma)$ is a generalized vector Sylvester matrix with each stripe a vector Toeplitz matrix having $s \times 1$ vector entries. If all the stripes have equal length k, then this is, up to permutation, the same as a block triangular Toeplitz matrix with blocks of size $s \times k$. We can also consider the case where $\mathbf{C}$ is a matrix made up of diagonal blocks of (possibly different sized) shift matrices, leading to mosaic generalized Sylvester matrices.

In case of the rational interpolation problems discussed in Examples 2.3 and 2.4, one is left with matrices $\mathbf{C}$ consisting of diagonal blocks of the form

$$\left[\begin{array}{cccc} x_0 & 0 & \cdots & 0 \\ 0 & x_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & x_k \end{array}\right] \quad \text{or} \quad \left[\begin{array}{ccccc} x_0 & 0 & \cdots & \cdots & 0 \\ 1 & x_1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & x_k \end{array}\right],$$

the first in the case of function evaluations, the second if one uses divided differences (or simply successive derivatives at a point different from zero). For the first choice, $\mathbf{K}(\vec{n},\sigma)$ consists of stripes, each of them a rectangular Vandermonde matrix multiplied on the left by a diagonal matrix.

A powerful formalism for solving structured systems is the concept of displacement operators (see, for example, [36]), that is, matrices $\mathbf{M}$ where for some given matrices $\mathbf{A}_1, \mathbf{A}_2$, the quantity $\mathbf{A}_1\mathbf{M} - \mathbf{M}\mathbf{A}_2$ has a much smaller rank than the size of $\mathbf{M}$. In our case we have

$$\begin{aligned} &\mathbf{C}_\sigma \cdot \mathbf{K}(\vec{n},\sigma) - \mathbf{K}(\vec{n},\sigma) \cdot \mathbf{Z} \\ &= \left[\begin{array}{ccccc|c|ccccc} 0 & \cdots & 0 & \mathbf{C}_\sigma^{\vec{n}^{(1)}} \mathbf{F}_\sigma^{(1)} & & \cdots & & 0 & \cdots & 0 & \mathbf{C}_\sigma^{\vec{n}^{(m)}} \mathbf{F}_\sigma^{(m)} \end{array}\right], \end{aligned}$$

where $\mathbf{Z}$ is block diagonal consisting of lower shift matrices of size $\vec{n}^{(j)}$, $j = 1, 2, \ldots, m$. Thus our striped Krylov matrices $\mathbf{K}(\vec{n},\sigma)$ have displacement rank m.

A significant number of fast (but not fraction-free) algorithms have been suggested in the last years for factoring or inverting matrices with small displacement rank, or for solving corresponding structured systems. For instance, we mention Levinson-type methods based on bordering techniques and Schur type algorithms (also called fast Gaussian elimination) based on the fact that Schur complements verify similar displacement equations [32, 36]. In our case, we wish to have a (fraction-free) description of the nullspace of all principal submatrices of $\mathbf{K}' := \mathbf{K}(\vec{n}, \sigma)\mathbf{P}$, where $\mathbf{P}$ is some permutation matrix such that $\mathbf{Z}' := \mathbf{P}^T\mathbf{Z}\mathbf{P}$ remains strictly lower triangular (that is, we follow some path in the corresponding solution table). Notice that the displacement equation for Schur complements (cf. [32, Lemma 3.1]) becomes quite involved since $\mathbf{Z}'$ is no longer upper triangular. Also, in case of singularities, one has to use look-ahead, or one needs to add a technique of pivoting [24, 31, 32, 33] which for general displacement operators $\mathbf{A}_1\mathbf{M} - \mathbf{M}\mathbf{A}_2$ seems to be feasible only if one of the matrices $\mathbf{A}_1$ (for row pivoting) or $\mathbf{A}_2$ (for column pivoting) is diagonal. However, our matrix $\mathbf{C}$ will be diagonal only in special cases,[1] and $\mathbf{Z}'$ is never diagonal.[2] One usually overcomes this drawback by using transformation techniques, that is, by multiplying $\mathbf{K}'$ on the left and/or on the right by suitable matrices (e.g., FFT matrices in the Toeplitz case) which changes the displacement operator but keeps the displacement rank essentially invariant [31, 32].

In the present paper we use neither transformation techniques nor look-ahead. In both cases these methods may present major inconveniences. Transformations can lead to a significant increase of complexity of the input data, and look-ahead is less efficient for large jumps (a common occurrence in GCD problems). In addition, both approaches do not allow us to keep track of all interpolation subproblems corresponding to principal submatrices of $\mathbf{K}'$. Our contribution in section 7 is to show that a very particular column pivoting still enables us to solve all corresponding subproblems. Here we generalize polynomial recurrences presented by several authors [8, 45, 46, 47, 51], and thus a polynomial language is more appropriate in our context.

4. Normality and controllable data. For the solvability of system (3.1) we require some regularity assumptions. The aim of this section is to discuss several such concepts.

DEFINITION 4.1 (multigradients, normality). *The scalar*

$$d(\vec{n}) = \det(\mathbf{K}(\vec{n}, |\vec{n}|))$$

is called the multigradient of $\mathbf{F}$ *of type* $\vec{n}$. *The multi-index* $\vec{n}$ *is called a* normal *point if* $d(\vec{n}) \neq 0$. *Finally, the data* $(\mathbf{C}, \mathbf{F})$ *is called* perfect *if every multi-index is normal.* □

We use the convention that the determinant of an empty matrix equals one. Of course, given $\sigma > 0$, the existence of a normal point $\vec{n}$ with $|\vec{n}| = \sigma$ requires that the linear functionals $c_0, \ldots, c_{\sigma-1}$ are linearly independent with respect to the set $\mathcal{V}_0 := \{f \cdot p(z) : p(z) \in \mathbb{Q}[z]^m\}$ (considered as a vector space over $\mathbb{Q}$), in general a

[1] See Examples 2.3 and 2.4. Here a row pivoting corresponds to a permutation of interpolation points (which need to be distinct), a classical technique in rational interpolation or M-Padé approximation.

[2] By using block pivoting, it seems to be possible to allow also for matrices A_1 and A_2, which are block diagonal. However, this does not apply for our setting. For example, consider the problem of Hermite–Padé approximation with three scalar functions following an offdiagonal path to $\vec{n} = [k, k, k]$ (these paths are included in section 7). Then $\mathbf{C}$ is the lower shift and $\mathbf{Z}'$ is the third power of the upper shift.

proper subset of $\mathcal{V}$. In terms of linear algebra, this is equivalent to saying that the data $(\mathbf{C}_\sigma, \mathbf{F}_\sigma)$ are *controllable*, i.e., for large k, the columns of $\mathbf{F}_\sigma, \mathbf{C}_\sigma \cdot \mathbf{F}_\sigma, \ldots, \mathbf{C}_\sigma^k \cdot \mathbf{F}_\sigma$ generate the whole space $\mathbb{Q}^\sigma$. Moreover, from system theory (see, e.g., [37, p. 426 ff, p. 481 ff]) it is well known that this necessary condition is also sufficient for the existence of a normal point $\vec{n}$ with $|\vec{n}| = \sigma$.

We will say that $(\mathbf{C}, \mathbf{F})$ is controllable if $(\mathbf{C}_\sigma, \mathbf{F}_\sigma)$ are controllable for all $\sigma \geq 0$. One easily verifies the equivalent condition that, for each $\sigma \geq 0$, there exists an element of $\mathcal{V}_0$ having exact order σ. Such a regularity assumption has been imposed for several algorithms for solving the approximation problems mentioned in the examples of section 2. Also, equivalent characterizations have been established: in the case of M-Padé approximation (see Example 2.4), it is shown in [5, Lemma 3.1] that $(\mathbf{C}, \mathbf{F})$ is controllable iff the vector of functions $f = [f_1, \ldots, f_m]$ does not vanish identically at any of the involved knots. In particular, for Hermite–Padé approximation we have the equivalent requirement $f(0) \neq 0$. Moreover [9, Lemma 2.7], for vector Hermite–Padé approximants (see Example 2.2), $(\mathbf{C}, \mathbf{F})$ is controllable iff the $s \times m$ matrix $f(0)$ has maximal rank.

Though such a condition allows us to simplify notation, for an application of our theory to the matrix-GCD problem we need to also allow for noncontrollable $(\mathbf{C}, \mathbf{F})$. One possibility to remedy this drawback is to introduce additional functions $f^{(m+1)}, f^{(m+2)}, \ldots$, and thus to consider a suitable extension $\mathbf{F}^*$ of $\mathbf{F}$. Instead, we prefer to consider a particular maximal subsequence of linear functionals being linearly independent with respect to $\mathcal{V}_0$. The symbol $*$ will be used to identify the resulting Krylov matrices and multigradients.

We define a unique sequence of integers $(\sigma(j))_{j=0,1,\ldots}$ being the indices of our maximal subsequence of linearly independent linear functionals by the following requirements: for all nonnegative integers j there holds

(4.1) $c_{\sigma(0)}, c_{\sigma(1)}, \ldots, c_{\sigma(j)}$ are linearly independent $\mathcal{V}_0$,

(4.2) $c_{\sigma(0)}, \ldots, c_{\sigma(j-1)}, c_\sigma$ are linearly dependent $\mathcal{V}_0$ for all $0 \leq \sigma < \sigma(j)$.

DEFINITION 4.2 (paranormality, σ-normality). *Let $\vec{n}$ be a multi-index, and let j, σ be nonnegative integers. We denote by $\mathbf{K}^*(\vec{n}, j)$ the matrix of size $j \times |\vec{n}|$ obtained by taking the rows labeled $\sigma(0), \ldots, \sigma(j-1)$ of the ordinary striped Krylov matrix $\mathbf{K}(\vec{n}, \sigma(j))$. The scalar*

$$d^*(\vec{n}) = \det(\mathbf{K}^*(\vec{n}, |\vec{n}|))$$

will be referred to as the modified multigradient *of $\mathbf{F}$ of type $\vec{n}$. The multi-index $\vec{n}$ is called* paranormal *if $d^*(\vec{n}) \neq 0$, and called* σ-normal *if it is paranormal and $\sigma(|\vec{n}| - 1) < \sigma \leq \sigma(|\vec{n}|)$ (where $\sigma(-1) := -1$).* □

Note that the concepts of paranormality and of normality (in the sense of Definition 4.1) coincide exactly in the case of controllable $(\mathbf{C}, \mathbf{F})$. Moreover, $\vec{n}$ is $|\vec{n}|$-normal iff it is a normal point. This implies in particular that $\sigma(j) = j$ for $j = 0, 1, \ldots, |\vec{n}| - 1$, that is, $(\mathbf{C}_{|\vec{n}|}, \mathbf{F}_{|\vec{n}|})$ is controllable. Also, by exploiting the dependency relations (4.2) one gets a special multiplication rule of the form (2.2) connecting only the linearly independent linear functionals

$$c_{\sigma(j)}(z \cdot f) = c^*_{j,0} \cdot c_{\sigma(0)}(f) + \cdots + c^*_{j,j} \cdot c_{\sigma(j)}(f)$$

for all $f \in \mathcal{V}_0$ and for all $j \geq 0$, with $c^*_{j,k} \in \mathbb{Q}$. Hence modified striped Krylov matrices $\mathbf{K}^*(\vec{n}, j)$ may be represented themselves as striped Krylov matrices with controllable

data $(\mathbf{C}^*, \mathbf{F}^*)$. However, in what follows we will not make use of this result. A final characterization is mentioned in the following lemma.

LEMMA 4.3. *The multi-index $\vec{n}$ is σ-normal iff any striped Krylov matrix $\mathbf{K}(\vec{n}', \sigma)$ containing the submatrix $\mathbf{K}(\vec{n}, \sigma)$ has rank $|\vec{n}|$. In this case, a maximal invertible submatrix is given by $\mathbf{K}^*(\vec{n}, |\vec{n}|)$.*

Proof. Apply Gaussian elimination with column pivoting to $\mathbf{K}(\vec{n}', \sigma)$. ∎

5. Mahler systems. In this section we introduce the notion of a Mahler system. These systems are generalizations of the Padé and matrix-type Padé systems of [19, 40, 41] and, up to a constant factor, have already been considered by Mahler [45] in the case of perfect systems for Hermite–Padé and simultaneous Padé approximants. They are also the fundamental building blocks that we use for the fraction-free algorithm presented in the later sections.

For a given multi-index $\vec{n}$ define $r(\vec{n}, z)$ and $p^{(\ell)}(\vec{n}, z)$ by $r(\vec{0}, z) = 0$, $p^{(\ell)}(\vec{n}, z) = 0$ in the case $\vec{n}^{(\ell)} = 0$ and otherwise by the determinant formulas

$$r(\vec{n}, z) = \det \left[\begin{array}{c} \mathbf{K}^*(\vec{n}, |\vec{n}| - 1) \\ \hline \mathbf{E}(z) \end{array} \right],$$

where

$$\mathbf{E}(z) = [f^{(1)}, \ldots, z^{\vec{n}^{(1)}-1} f^{(1)} | \ldots\ldots | f^{(m)}, \ldots, z^{\vec{n}^{(m)}-1} f^{(m)}]$$

and

$$p^{(\ell)}(\vec{n}, z) = \det \left[\begin{array}{c} \mathbf{K}^*(\vec{n}, |\vec{n}| - 1) \\ \hline \mathbf{E}^{(\ell)}(z) \end{array} \right]$$

with

$$\mathbf{E}^{(\ell)}(z) = \mathbf{E}^{(\ell)}(\vec{n}, z) = [0, \ldots, 0 | 1, z, \ldots, z^{\vec{n}^{(\ell)}-1} | 0, \ldots, 0]. \tag{5.1}$$

The nonzero entries in $\mathbf{E}^{(\ell)}(z)$ occur in the ℓth stripe. In addition, we let $p(\vec{n}, z) = [p^{(1)}(\vec{n}, z), \ldots, p^{(m)}(\vec{n}, z)]^T$ be the column vector of the polynomials defined above.

LEMMA 5.1. *For a multi-index $\vec{n}$ we have*

(a) $f \cdot p(\vec{n}, z) = r(\vec{n}, z) \in \mathcal{V}_0$;

(b) $\operatorname{ord}(r(\vec{n}, z)) \geq \sigma(|\vec{n}| - 1)$ *and* $c_{\sigma(|\vec{n}|-1)}(r(\vec{n}, z)) = d^*(\vec{n})$;

(c) $deg(p^{(\ell)}(\vec{n}, z)) \leq \vec{n}^{(\ell)} - 1$. *Moreover, if $\vec{n}^{(\ell)} > 0$, then the $\vec{n}^{(\ell)} - 1$st coefficient is*

$$p^{(\ell)}_{\vec{n}^{(\ell)}-1} = \epsilon^{(\ell)}(\vec{n}) \cdot d^*(\vec{n} - \vec{e}_\ell), \quad \epsilon^{(\ell)}(\vec{n}) := (-1)^{\vec{n}^{(\ell+1)} + \cdots + \vec{n}^{(m)}};$$

(d) $p(\vec{n}, z)$ *is not identically zero iff, up to multiplication by a scalar from $\mathbb{Q}$, there exists exactly one solution of type $(\sigma(|\vec{n}| - 2) + 1, \vec{n})$ (being given by $p(\vec{n}, z)$).*

Proof. Part (a) follows by expanding determinants with respect to the last row. In order to show part (b) notice that, for $i = \sigma(j)$, $0 \leq j < |\vec{n}| - 1$, $c_i(r(\vec{n}, z))$ is a determinant of a matrix with two equal rows and hence is zero. In the case $i \in \{0, \ldots, \sigma(|\vec{n}| - 1) - 1\} \setminus \{\sigma(0), \ldots, \sigma(|\vec{n}| - 2)\}$ we obtain $c_i(r(\vec{n}, z)) = 0$ according

to (4.2). The first potential case where a possibly linearly independent row occurs is when $i = \sigma(|\vec{n}| - 1)$, and thus $c_i(r(\vec{n}, z)) = d^*(\vec{n})$. Part (c) follows by expanding out the determinant definition of $p^{(\ell)}(\vec{n}, z)$ along the last row. The coefficient is, at least up to sign, the same as taking determinants of the matrix determined by eliminating the last row and column $\vec{n}^{(1)}+\cdots+\vec{n}^{(\ell)}$, which is just $d^*(\vec{n}-\vec{e}_\ell)$. The sign is determined by counting the number of columns from the bottom right corner of the matrix. Finally, the assertion of part (d) is a consequence of Cramer's rule applied to the homogeneous system of linear equations $\mathbf{K}^*(\vec{n}, |\vec{n}| - 1) \cdot \mathbf{P} = 0$, since in fact $p(\vec{n}, z) \neq 0$ iff the rank of the matrix $\mathbf{K}^*(\vec{n}, |\vec{n}| - 1)$ of size $(|\vec{n}| - 1) \times |\vec{n}|$ is maximal. □

Lemma 5.1 says that $p(\vec{n}, z)$ is a solution in $\mathbb{D}^m[z]$ to our interpolation problem of Definition 2.1 of type $(\sigma, \vec{n})$, $\sigma \leq \sigma(|\vec{n}| - 1)$. However, one rarely wants to use this definition in order to compute this solution. Rather, it is better to use systems of linear equations for this computation. For instance, suppose that $\vec{n}$ is a normal point. Then solving the system (3.1) using Cramer's rule over $\mathbb{Q}$ gives a solution $\mathbf{P}$ of problem (3.2) with $\mathbf{P}$ being a vector having only coefficients from $\mathbb{D}$. From Lemma 5.1 (b), (d) one sees that $\mathbf{P}$ provides the coefficients of the polynomials $p(\vec{n}, z)$ via partitioning the coefficient vector as

$$\mathbf{P} = [p_0^{(1)}, \ldots, p_{\vec{n}^{(1)}-1}^{(1)} | \ldots\ldots | p_0^{(m)}, \ldots, p_{\vec{n}^{(m)}-1}^{(m)}].$$

Similarly, suppose that $\vec{n}$ is paranormal (see Definition 4.2) and choose σ such that $\vec{n}$ is σ-normal. By Lemma 4.3 we have rank $\mathbf{K}(\vec{n}, \sigma) = \text{rank } \mathbf{K}(\vec{n}+\vec{e}_i, \sigma) = |\vec{n}|$ for all $i = 1, \ldots, m$, with a square submatrix of maximal rank being given by $\mathbf{K}^*(\vec{n}, |\vec{n}|)$. Therefore we may find unique solutions for the systems of equations (usually referred to as *fundamental equations* [36] or *Yule–Walker equations* of the corresponding striped Krylov matrix)

$$\mathbf{K}(\vec{n}, \sigma) \cdot \bar{\mathbf{P}}^{(i)} = -\mathbf{C}_\sigma^{\vec{n}^{(i)}} \cdot \mathbf{F}_\sigma^{(i)}, \qquad 1 \leq i \leq m.$$

Again using Cramer's rule (with respect to $\mathbf{K}^*(\vec{n}, |\vec{n}|)$), we obtain solutions $\tilde{\mathbf{P}}^{(i)}$ of elements from the domain $\mathbb{D}$ to the systems

$$\mathbf{K}(\vec{n}, \sigma) \cdot \tilde{\mathbf{P}}^{(i)} = -d^*(\vec{n}) \cdot \mathbf{C}_\sigma^{\vec{n}^{(i)}} \cdot \mathbf{F}_\sigma^{(i)}, \qquad 1 \leq i \leq m. \tag{5.2}$$

Thus, by part (c) of Lemma 5.1, the vector $\tilde{\mathbf{P}}^{(i)}$ consists of the coefficients of the vector of determinant polynomials $\epsilon^{(i)}(\vec{n}) \cdot p(\vec{n} + \vec{e}_i, z)$.

We are interested in recursively or iteratively computing solutions of equation (2.3). However to do this we need a larger collection of solutions to the problem. One can think of the scalar GCD problem as an example—there one needs two remainders at every step to get the next remainder. In our case we need to look for the m solutions described already by (5.2).

Definition 5.2 (Mahler systems). *The $m \times m$ matrix of polynomials*

$$\mathbf{M}(\vec{n}, z) = [\mathbf{M}^{(\lambda,j)}(\vec{n}, z)]_{\lambda,j=1}^m, \qquad \mathbf{M}^{(\lambda,j)}(\vec{n}, z) := \epsilon^{(j)}(\vec{n}) \cdot p^{(\lambda)}(\vec{n} + \vec{e}_j, z),$$

is called the Mahler system *of type $\vec{n}$. We shall denote its jth column by $\mathbf{M}^{(\cdot,j)}(\vec{n}, z)$.* □

Some Mahler systems for Hermite–Padé approximation may be found in Example 6.1 below. For the particular case of M-Padé approximation at a normal point $\vec{n}$, our Mahler system coincides with that proposed by Mahler [45] (up to the common scalar factor $d^*(\vec{n})$). In what follows, we will consider only Mahler systems at paranormal points for which we may establish several equivalent characterizations.

LEMMA 5.3. *Let $\vec{n}$ be a multi-index, and $\lambda \in \{1, \ldots, m\}$. The following assertions are pairwise equivalent:*

(a) *$\vec{n}$ is a paranormal point.*
(b) $\deg p^{(\lambda)}(\vec{n} + \vec{e}_\lambda, z) = \vec{n}^{(\lambda)}$.
(c) *A solution of type $(\sigma(|\vec{n}| - 1) + 1, \vec{n} + \vec{e}_\lambda)$ is unique up to multiplication with an element from $\mathbb{Q}$, with its λth component having exact degree $\vec{n}^{(\lambda)}$.*
(d) *For any $\sigma > \sigma(|\vec{n}| - 1)$, a solution of type $(\sigma, \vec{n})$ is necessarily trivial.*
(e) *The columns of the Mahler system $\mathbf{M}(\vec{n}, z)$ are linearly independent over $\mathbb{Q}[z]$.*

Proof. The equivalence of assertion (a) and any of the assertions (b) or (c) follows from Lemma 5.1 and the following remarks. In order to establish equivalence between (a) and (d), notice that the coefficient vector $\mathbf{P}$ of a solution $p(z)$ of type $(\sigma(|\vec{n}|-1)+1, \vec{n})$ necessarily satisfies $\mathbf{K}(\vec{n}, \sigma(|\vec{n}| - 1) + 1) \cdot \mathbf{P} = 0$. By definition (4.1), (4.2), we obtain the equivalent system of equations $\mathbf{K}^*(\vec{n}, |\vec{n}|) \cdot \mathbf{P} = 0$, with a square matrix of coefficients. Thus $\mathbf{K}^*(\vec{n}, |\vec{n}|)$ is nonsingular or, in other words, $d^*(\vec{n}) \neq 0$ iff each such solution $\mathbf{P}$ is trivial.

For the equivalence between (a) and (e) it is sufficient to show that $\det \mathbf{M}(\vec{n}, z) \neq 0$ iff $d^*(\vec{n}) \neq 0$. Notice that the elements of $\mathbf{M}(\vec{n}, z)$, namely, $\mathbf{M}^{(\lambda,j)}(\vec{n}, z)$, $\lambda, j = 1, \ldots, m$, are determinants of matrices of size $(|\vec{n}| + 1) \times (|\vec{n}| + 1)$. These matrices are obtained by bordering the matrix $\mathbf{K}^*(\vec{n}, |\vec{n}|)$ on the bottom by one additional row and on the right by one additional column. Let $\vec{e} := (1, 1, \ldots, 1)$, and let $\mathbf{E}^{(\lambda)}(\vec{n}, z)$ be defined as in (5.1). Then, by the Sylvester determinantal identity, we have

$$\det \mathbf{M}(\vec{n}, z) = (\det \mathbf{K}^*(\vec{n}, |\vec{n}|))^{m-1} \cdot \beta(z),$$

where $\beta(z)$ denotes the determinant of the augmented matrix

$$\beta(z) = \pm \det \left[\begin{array}{c} \mathbf{K}^*(\vec{n} + \vec{e}, |\vec{n}|) \\ \hline \mathbf{E}^{(1)}(\vec{n} + \vec{e}, z) \\ \vdots \\ \mathbf{E}^{(m)}(\vec{n} + \vec{e}, z) \end{array}\right].$$

Expanding $\beta(z)$ with respect to the last m rows shows that $\beta(z)$ is a polynomial in z, and that, more precisely[3],

$$\beta(z) = \pm d^*(\vec{n}) \cdot z^{|\vec{n}|} + \alpha(z), \qquad \deg \alpha < |\vec{n}|.$$

Here we have taken into account that the coefficient of $z^{|\vec{n}|}$ in $\beta(z)$ is obtained by the cofactor of $\operatorname{diag}(z^{\vec{n}^{(1)}}, \ldots, z^{\vec{n}^{(m)}})$ in $\beta(z)$. Consequently, $\det \mathbf{M}(\vec{n}, z) = \pm d^*(\vec{n})^{m-1} \cdot (d^*(\vec{n}) \cdot z^{|\vec{n}|} \pm \alpha)$. Therefore the two quantities $\det \mathbf{M}(\vec{n}, z)$ and $d^*(\vec{n})$ only simultaneously become zero. ∎

Given a paranormal multi-index $\vec{n}$, we will mostly apply Lemma 5.3 in order to verify that a given matrix polynomial is a Mahler system of type $\vec{n}$. Here we just have

[3] One shows that, for controllable $(\mathbf{C}, \mathbf{F})$

$$\det \mathbf{M}(\vec{n}, z) = \pm d^*(\vec{n})^m \cdot \prod_{k=0}^{|\vec{n}|-1} (z - c_{k,k}).$$

(For the approximation problems of section 2, see [46, p. 42], [3, p. 90–91], or [9, Lemma 2.7].)

to check that, for $\lambda = 1, \ldots, m$, the λth column is a solution of type $(\sigma(|\vec{n}|-1)+1, \vec{n}+\vec{e}_\lambda)$ with the correct normalization, i.e., the coefficient of $z^{\vec{n}^{(\lambda)}}$ of the λth component equals $d^*(\vec{n})$.

To the end of this section, we state a further equivalent characterization of paranormal multi-indices. This statement will be proved at the end of section 7 where additional results are available. For the remainder of this paper we will use the abbreviation $z^{\vec{\nu}}$ for denoting the diagonal matrix $\mathrm{diag}(z^{\vec{\nu}^{(1)}}, \ldots, z^{\vec{\nu}^{(m)}})$.

COROLLARY 5.4. *Let $\vec{n}$ be a multi-index, and $\sigma > \sigma(|\vec{n}|-1)$. Then $\vec{n}$ is σ–normal iff there exists a matrix polynomial $\mathbf{M}(z)$ with columns having order $\geq \sigma$ which satisfies the degree constraints*

$$z^{-\vec{n}} \cdot \mathbf{M}(z) = c \cdot \mathbf{I}_m + \mathcal{O}(z^{-1})_{z\to\infty}, \qquad c \in \mathbb{Q} \setminus \{0\}.$$

In this case, $\mathbf{M}(z) = \frac{c}{d^*(\vec{n})} \cdot \mathbf{M}(\vec{n}, z)$.

6. Computing Mahler systems along perfect paths. For a given multi-index $\vec{n}$, we are interested in computing a solution of type $(|\vec{n}|-1, \vec{n})$ to the interpolation problem of Definition 2.1 in a fraction-free way. By Lemma 5.1, the polynomial vector $p(\vec{n}, z)$ defined in the previous section provides one solution to this problem. Of course, to compute these polynomials one does not want to use the determinant definition, except perhaps for small problems. In this section we give a fast method to compute the solution to our rational interpolation problem using only polynomial operations over the integral domain $\mathbb{D}$. However, for the algorithm presented in this section we require some regularity assumptions, which are no longer necessary for the algorithm presented in the next section.

In the case where we are at a normal point $\vec{n}$ the next theorem tells us (in a more general setting) how to compute a Mahler system at a subsequent normal point $\vec{n}+\vec{e}_\lambda$ from the Mahler system at $\vec{n}$. A similar recurrence relation for Hermite–Padé approximation has been established earlier by Paszkowski [46, 47, 48] and generalized by one of the authors [3, Kapitel 3.3] without, however, noticing that this is the key for fraction-free computations.

THEOREM 6.1. *Suppose that $\vec{n}$ is paranormal. Furthermore, let $\sigma(|\vec{n}|-1) < \sigma \leq \sigma(|\vec{n}|)$, and for $\ell = 1, \ldots, m$ set*

$$r^{(\ell)} := c_\sigma \left(f \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z) \right).$$

(a) *$\vec{n}$ is also $(\sigma+1)$-normal (i.e., $\sigma < \sigma(|\vec{n}|)$) iff $r^{(1)} = r^{(2)} = \ldots = r^{(m)} = 0$.*
(b) *$\vec{n}+\vec{e}_\lambda$ is a paranormal point iff $r^{(\lambda)} \neq 0$.*
(c) *In the case $r^{(\lambda)} \neq 0$, we define also for $\ell = 1, \ldots, m$, $\ell \neq \lambda$*

$$p^{(\ell)} := \mathrm{coefficient}(\mathbf{M}^{(\ell,\lambda)}(\vec{n}, z), z^{\vec{n}^{(\ell)}-1}).$$

Then $\mathbf{M}(\vec{n}+\vec{e}_\lambda, z)$ can be computed from $\mathbf{M}(\vec{n}, z)$ as follows:

$$\mathbf{M}^{(\cdot,\ell)}(\vec{n}+\vec{e}_\lambda, z) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) = \mathbf{M}^{(\cdot,\ell)}(\vec{n}, z) \cdot r^{(\lambda)} - \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot r^{(\ell)} \tag{6.1}$$

for $\ell = 1, 2, \ldots, m$, $\ell \neq \lambda$, and

$$\begin{aligned} \mathbf{M}^{(\cdot,\lambda)}(\vec{n}+\vec{e}_\lambda, z) \cdot p^{(\lambda)} \cdot \epsilon^{(\lambda)}(\vec{n}) = {} & (z - c_{\sigma,\sigma}) \cdot \mathbf{M}^{(\cdot,\lambda)}(\vec{n}, z) \cdot r^{(\lambda)} \\ & - \sum_{\ell \neq \lambda} \mathbf{M}^{(\cdot,\ell)}(\vec{n}+\vec{e}_\lambda, z) \cdot p^{(\ell)} \cdot \epsilon^{(\lambda)}(\vec{n}). \end{aligned} \tag{6.2}$$

Proof. For a proof of part (a), set

$$B := \mathbf{K}(\vec{n}+[\sigma+1,\sigma+1,\ldots,\sigma+1],\sigma), \quad B' := \mathbf{K}(\vec{n}+[\sigma+1,\sigma+1,\ldots,\sigma+1],\sigma+1).$$

By Lemma 4.3 along with our assumptions, we have that rank $B = |\vec{n}|$, and from the Cayley–Hamilton theorem we know that rank $B' \geq$ rank $\mathbf{K}(\vec{n}',\sigma+1)$ for any multi-index $\vec{n}'$. Hence from definition (4.1), (4.2) we obtain the characterization $\sigma < \sigma(|\vec{n}|)$ iff rank $B =$ rank B'. The $m\cdot(\sigma+1)$ coefficient vectors of the polynomial vectors

$$(z-c_{\sigma,\sigma})^j \cdot \mathbf{M}^{(\cdot,\ell)}(\vec{n},z), \qquad \ell = 1,\ldots,m, \qquad j = 0,\ldots,\sigma,$$

are easily shown to be elements of the kernel of B, and are linearly independent over $\mathbb{Q}$ by Lemma 5.3(e). Thus we have found a basis of the kernel of B. Notice also that, according to (2.2), the order of $f\cdot(z-c_{\sigma,\sigma})^j\cdot\mathbf{M}^{(\cdot,\ell)}(\vec{n},z)$ is larger than σ if $j>0$. As a consequence, we have established the equivalencies $\sigma<\sigma(|\vec{n}|)$ iff the kernels of B and B' coincide iff $f\cdot\mathbf{M}^{(\cdot,\ell)}(\vec{n},z)$ has order $\geq\sigma+1$ for $\ell=1,\ldots,m$, as claimed in part (a).

Assertion (b) follows from part (a) together with Lemma 5.1(b).

In order to show the recurrence relation (6.1) for the case $\ell\neq\lambda$, let

$$q(z) := \mathbf{M}^{(\cdot,\ell)}(\vec{n}+\vec{e}_\lambda,z)\cdot p^{(\lambda)}\cdot\epsilon^{(\lambda)}(\vec{n}) - \mathbf{M}^{(\cdot,\ell)}(\vec{n},z)\cdot r^{(\lambda)} + \mathbf{M}^{(\cdot,\lambda)}(\vec{n},z)\cdot r^{(\ell)}.$$

We claim that $q(z)=0$. First by construction we get $ord\,(f\cdot q(z))\geq\sigma+1$. Furthermore, $deg\,q^{(\mu)}(z)\leq\vec{n}^{(\mu)}-1+\delta_{\mu,\ell}+\delta_{\mu,\lambda}$. More precisely, the coefficient of $z^{\vec{n}^{(\ell)}}$ of the ℓth component of $q(z)$ is given by

$$d^*(\vec{n}+\vec{e}_\lambda)\cdot p^{(\lambda)}\cdot\epsilon^{(\lambda)}(\vec{n}) - d^*(\vec{n})\cdot r^{(\lambda)} = 0$$

since $p^{(\lambda)} = d^*(\vec{n})$ due to Lemma 5.1(c), and $r^{(\lambda)} = \epsilon^{(\lambda)}(\vec{n})\cdot d^*(\vec{n}+\vec{e}_\ell)$ due to Lemma 5.1(b). Hence $q(z)$ is a solution of type $(\vec{n}+\vec{e}_\lambda,\sigma+1)$, and thus by Lemma 5.3(d) is identically zero.

Identity (6.2) is shown in a similar manner; let

$$q(z) := (z-c_{\sigma,\sigma})\cdot\mathbf{M}^{(\cdot,\lambda)}(\vec{n},z)\cdot d^*(\vec{n}+\vec{e}_\lambda) - \sum_{\ell=1}^{m}\mathbf{M}^{(\cdot,\ell)}(\vec{n}+\vec{e}_\lambda,z)\cdot p^{(\ell)}.$$

Since $d^*(\vec{n}+\vec{e}_\lambda) = r^{(\lambda)}\cdot\epsilon^{(\lambda)}(\vec{n})$, it is sufficient to prove that $q(z)$ vanishes identically, which follows again by Lemma 5.3(d) by checking order and degree of $q(z)$. First notice that $ord\,(f\cdot(z-c_{\sigma,\sigma})\cdot\mathbf{M}^{(\cdot,\lambda)}(\vec{n},z))\geq\sigma+1$ by (2.2). Moreover, all terms in the sum have order at least $\sigma+1$, and thus $ord\,(f\cdot q(z))\geq\sigma+1$. Also, by definition, the μth component of $q(z)$ contains only powers z^j with $j=0,1,\ldots,\vec{n}^{(\mu)}+\delta_{\lambda,\mu}=:j_\mu$. By using Lemma 5.1 (c), one verifies that the factors in the sum have been chosen such that the coefficient before z^{j_μ} in $q^{(\mu)}(z)$ vanishes, and hence $deg\,q^{(\mu)}(z)\leq\vec{n}^{(\mu)}-1+\delta_{\mu,\lambda}$ for all μ. Thus $q(z)=0$. □

Theorem 6.1 leads to an algorithm to compute solutions to the rational interpolation problem on staircases under the assumption that all intermediate problems are at normal points. Here we denote by staircase a sequence $(\vec{n}_k)_{k=0,1,\ldots}$ of multi-indices with the properties that

(6.3) $\quad\vec{n}_0=\vec{0}$, $\vec{n}_{|\vec{n}|}=\vec{n}$, and for all $k\geq0\,\exists\lambda_k$ such that $\vec{n}_{k+1}-\vec{n}_k=\vec{e}_{\lambda_k}$.

At every step $\vec{n}_k$ we find a λ such that $\vec{n}_{k+1}=\vec{n}_k+\vec{e}_\lambda$ is normal (which is, for instance, the case when the vector f is perfect; see Definition 4.1). Then, using the

TABLE 1
Algorithm FFFGnormal.

ALGORITHM FFFGnormal (on arbitrary staircases consisting of normal points)

INPUT: a vector of formal series f,
a staircase $(\vec{n}_k)_{k=0,\ldots,K}$ of normal points.

OUTPUT: For $k = 0, 1, 2, \ldots, K$ with $\epsilon_k \in \{-1, 1\}$:
Mahler systems $\mathbf{M}_k = \epsilon_k \cdot \mathbf{M}(\vec{n}_k, z)$,
multigradients $d_k = \epsilon_k \cdot d^*(\vec{n}_k)$.

INITIALIZATION: $\mathbf{M}_0 \leftarrow \mathbf{I}_m$, $d_0 \leftarrow 1$

ITERATIVE STEP: For $k = 0, 1, 2, \ldots, K-1$:
Define $\lambda \in \{1, \ldots, m\}$ by $\vec{n}_{k+1} - \vec{n}_k = \vec{e}_\lambda$.

Calculate for $\ell = 1, \ldots, m$:
first term of residuals $r^{(\ell)} \leftarrow c_k(f \cdot \mathbf{M}_k^{(\cdot,\ell)})$,
leading coefficients $p^{(\ell)} \leftarrow \mathit{coefficient}(\mathbf{M}_k^{(\ell,\lambda)}, z^{\vec{n}_k^{(\ell)}-1})$.

Increase order for $\ell = 1, \ldots, m$, $\ell \neq \lambda$:
$\mathbf{M}_{k+1}^{(\cdot,\ell)} \leftarrow [\mathbf{M}_k^{(\cdot,\ell)} \cdot r^{(\lambda)} - \mathbf{M}_k^{(\cdot,\lambda)} \cdot r^{(\ell)}]/d_k$,
$\mathbf{M}_{k+1}^{(\cdot,\lambda)} \leftarrow (z - c_{k,k}) \cdot \mathbf{M}_k^{(\cdot,\lambda)}$

Adjust degree constraints:
$\mathbf{M}_{k+1}^{(\cdot,\lambda)} \leftarrow [\mathbf{M}_{k+1}^{(\cdot,\lambda)} \cdot r^{(\lambda)} - \sum_{\ell\neq\lambda} \mathbf{M}_{k+1}^{(\cdot,\ell)} \cdot p^{(\ell)}]/d_k$

New multigradient: $d_{k+1} = r^{(\lambda)}$

iteration given by Theorem 6.1 with $\sigma = |\vec{n}_k| = k$, we see that we can remove the scalar common factor $p^{(\mu)} = d^*(\vec{n}_k)$ before we proceed with our next iteration. This scalar is determined as the leading coefficient of the (λ, λ) term of the kth Mahler system.

Therefore, not only the representations (3.2), (5.2) of the solutions but also recurrence (6.1) remind one of the well-known recurrence relations of fraction-free Gaussian elimination [2, 30]. On the other hand, relation (6.2) gives a significant gain in complexity in comparison with the classical Gaussian elimination, obtained by taking into account the particular structure of our block Krylov matrices. This serves as motivation to refer to our algorithm proposed in Table 1 as fraction-free fast Gaussian elimination.

From Theorem 6.1 one can see that the iteration is best done in two stages. If we have the Mahler system of type $\vec{n}_k$ and wish to compute the Mahler system of type $\vec{n}_{k+1} = \vec{n}_k + \vec{e}_{\lambda_k}$, then we first increase the order of all the columns of $\mathbf{M}(\vec{n}_k, z)$. This is done by using column λ_k to increase the orders of all the other columns using (6.1) of Theorem 6.1. The λ_kth column itself has its order increased by multiplication by $z - c_{|\vec{n}_k|,|\vec{n}_k|}$. At this stage all the columns except λ_k are constant multiples of the corresponding columns of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$. We pull out the constant from these columns to make them the same as the corresponding columns of the new Mahler system. Finally, column λ_k does not have the correct degree structure as required for our new Mahler system. We then use all the other columns to return this degree structure to the desired form. This gives column λ_k as a constant multiple of the λ_kth column of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$. Removing this constant gives the correct λ_kth column

of $\mathbf{M}(\vec{n}_k + \vec{e}_{\lambda_k}, z)$ and hence the new Mahler system.

In Algorithm FFFGnormal stated in Table 1, one may find a slight simplification of relations (6.1), (6.2). In fact, we prefer to compute Mahler systems only up to sign, namely $\mathbf{M}_k = \epsilon_k \cdot \mathbf{M}(\vec{n}_k, z)$ with

$$\epsilon_0 = 1, \qquad \epsilon_{k+1} = \epsilon^{(\lambda_k)}(\vec{n}_k) \cdot \epsilon_k, \qquad k \geq 0, \tag{6.4}$$

since then all terms $\epsilon^{(\lambda_k)}(\vec{n}_k)$ in (6.1), (6.2) may be dropped.

In Table 1, we have not discussed in detail how to compute efficiently the first term of the residuals, namely $r^{(\ell)}$, $\ell = 1, \ldots, m$. One possibility (mainly applicable for Hermite–Padé approximation and its vector counterpart) is to compute explicitly $c_k(f \cdot \mathbf{M}_\sigma^{(\cdot,\ell)})$ by determining a particular coefficient of the scalar product $f \cdot \mathbf{M}_\sigma$. Another approach, which may be preferable for more complicated special multiplication rules (2.2), is to simultaneously compute all required values of the residuals, i.e., to compute the (nontrivial part of the) *residual vectors*

$$\mathbf{R}_k^{(\ell)} = [c_\sigma(f \cdot \mathbf{M}_k^{(\cdot,\ell)})]_{\sigma=0,\ldots,K-1}.$$

Here we use the initializations $\mathbf{R}_0^{(\ell)} = \mathbf{F}^{(\ell)}$ and obtain according to Table 1 and (2.2) the recurrences

$$\mathbf{R}_{k+1}^{(\ell)} = \begin{cases} \left[\mathbf{R}_k^{(\ell)} \cdot r^{(\lambda)} - \mathbf{R}_k^{(\lambda)} \cdot r^{(\ell)}\right]/d_k & \text{for } \ell \neq \lambda, \\ \left[(\mathbf{C}_K - c_{k,k} \cdot \mathbf{I}_K) \cdot \mathbf{R}_k^{(\lambda)} \cdot r^{(\lambda)} - \sum_{\ell \neq \lambda} \mathbf{R}_{k+1}^{(\ell)} \cdot p^{(\ell)}\right]/d_k & \text{for } \ell = \lambda. \end{cases}$$

We again observe close relationships to the recurrence relations of the classical one-step fraction-free Gaussian elimination [2, 30]. We also mention that multistep elimination schemes may be given. However, due to our special rule, the formalism becomes more complicated.

Example 6.1. *Let f be the vector of power series*[4] *whose first six terms are*

$$\left[1 - z + 19\,z^2 + 3\,z^3 - 5\,z^5, \quad 9 + 6\,z - 5\,z^2 + 5\,z^3 + 4\,z^5, \quad 1 + 9\,z^2 + 9\,z^3 - 4\,z^5\right].$$

Then the Mahler systems of f of type $[1,0,0]$, $[1,1,0]$, $[1,1,1]$, and $[2,1,1]$ generated by the preceding algorithm are given by

$$\mathbf{M}_1 = \mathbf{M}(\vec{n}_1, z) = \begin{bmatrix} z & -9 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{M}_2 = \mathbf{M}(\vec{n}_2, z) = \begin{bmatrix} 15\,z + 9 & 81 & -6 \\ -1 & 15\,z - 9 & -1 \\ 0 & 0 & 15 \end{bmatrix},$$

$$\mathbf{M}_3 = \mathbf{M}(\vec{n}_3, z) = \begin{bmatrix} 26\,z + 80 & 810 & 86 \\ 9 & 26\,z + 96 & 10 \\ -161 & -1674 & 26\,z - 176 \end{bmatrix},$$

and

$$\mathbf{M}_4 = \mathbf{M}(\vec{n}_4, z) = \begin{bmatrix} -670\,z^2 + 138\,z + 270 & 12286\,z + 16930 & 1042\,z + 990 \\ 22 & -670\,z + 1779 & 103 \\ -468 & -32941 & -670\,z - 1917 \end{bmatrix}.$$

[4] Since $f(0) = [1, 9, 1] \neq 0$, we have controllable data, and thus we may drop the asterisk.

The residuals determined by $f \cdot \mathbf{M}_4$ *are given by*

$$\left[\ -12316\,z^4 + O\left(z^5\right),\quad 33508\,z^4 + O\left(z^5\right),\quad -2904\,z^4 + O\left(z^5\right)\ \right]$$

so that in this step $r^{(1)} = d([3,1,1]) = -12316$, $r^{(2)} = -d([2,2,1]) = 33508$, *and* $r^{(3)} = d([2,1,2]) = -2904$ *(see Lemma* 5.1(b)*). Also, the leading coefficients of the polynomials on the diagonal of the Mahler system* $\mathbf{M}_4$ *are equal to* $d_4 = d([2,1,1]) = -670$. *In order to generate the Mahler system* $\mathbf{M}_5 = -\mathbf{M}(\vec{n}_5, z)$ *of type* $[2,2,1]$, *the algorithm first increases the orders of all the columns by combining column* ℓ *with column* 2, $\ell = 1, 3$, *and by multiplying the second column by* z. *This gives*

$$\tilde{\mathbf{P}} = \begin{bmatrix} 33508\,z^2 - 232744\,z - 324712 & 12286\,z^2 + 16930z & -105364\,z - 122892 \\ 12316\,z - 33802 & -670\,z^2 + 1779z & 2904\,z - 12862 \\ 628930 & -32941z & 33508\,z + 238650 \end{bmatrix}.$$

Note that the common multiplier $d_4 = -670$ *has been removed from the computations of columns* 1 *and* 3. *The algorithm then uses the values* $p^{(1)} = 12286 = d([1,2,1])$ *with the first column and* $p^{(3)} = -32941 = -d([2,2,0])$ *with the third column in order to return the second column to the degree bounds needed for a Mahler system of type* $[2,2,1]$ *(see Lemma* 5.1(c)*). This then gives* $\mathbf{M}_5$ *as*

$$\begin{bmatrix} 33508\,z^2 - 232744\,z - 324712 & 65690\,z + 87722 & -105364\,z - 122892 \\ 12316\,z - 33802 & 33508\,z^2 - 5906z + 12531 & 2904\,z - 12862 \\ 628930 & -200501 & 33508\,z + 238650 \end{bmatrix}.$$

□

We remark that our use of the integers as a coefficient domain in Example 6.1 is mainly for ease of presentation. A more typical domain would be $\mathbb{Q}[\epsilon]$, where ϵ denotes an indeterminant (for example, ϵ may be a symbolic representation of an allowable error for numeric input).

An asymptotic cost analysis of computing a Mahler system by Algorithm FFFG-normal is provided in the following theorem. Here we assume following [10] that, for $a, b \in \mathbb{D}$,

$$\begin{aligned} \text{size}(a+b) &= \mathcal{O}(\max\{\text{size}(a), \text{size}(b)\}), \\ \text{size}(a \cdot b) &= \mathcal{O}(\text{size}(a) + \text{size}(b)), \\ \text{cost}(a+b) &= \mathcal{O}(1), \\ \text{cost}(a \cdot b) &= \mathcal{O}(\text{size}(a) \cdot \text{size}(b)), \end{aligned}$$

where the function "size" measures the total storage required for its arguments and the function "cost" estimates the number of boolean operations (machine cycles) required to perform the indicated arithmetic. These assumptions are justified for large operands where, for example, the cost of addition is negligible in comparison to the cost of multiplication. Notice that a smaller complexity may be expected if fast multiplication algorithms (e.g., Schönhage–Strassen) can be applied (cf. Knuth [38]).

THEOREM 6.2. *Let* κ *be an upper bound for the size of any element occurring in* $\mathbf{C}$ *or in* $\mathbf{C}^j \cdot \mathbf{F}$, $j \geq 0$, *and suppose that only* $\mathcal{O}(1)$ *entries in a row of* $\mathbf{C}$ *are different from zero. Then for computing a Mahler system of order* K *by Algorithm FFFGnormal we have the cost estimate* $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$.

Proof. Let $0 \le k \le K$. We obtain a bound for the size of the $m \times (k+1)$ coefficients of the components of $\mathbf{M}_k$ by using the determinantal representation of Definition 5.2: applying the Hadamard inequality (3.3) and taking into account the above assumptions, we get for their size the upper bound $\mathcal{O}(k \cdot \kappa)$. The same size estimate is valid for the $m \cdot (K-k)$ nontrivial components of the residual vectors $\mathbf{R}_k^{(\ell)}$, $\ell = 1, \ldots, m$.

In step k of the algorithm, we have to perform essentially $2m$ operations of the form

$$\mathbf{P}_3 = [a_1 \cdot \mathbf{P}_1 + a_2 \cdot \mathbf{P}_2]/a_3,$$

where $a_j \in \mathbb{D}$, and $\mathbf{P}_j \in \mathbb{D}[z]^m$ having $\mathcal{O}(k)$ nontrivial coefficients. In addition, for computing the residual vectors we again have essentially $2m$ operations of the above form, but now $\mathbf{P}_j$ stands for some vector having $\mathcal{O}(K-k)$ nontrivial components (by assumption on $\mathbf{C}$, the cost of multiplying $(\mathbf{C}_K - c_{k,k} \cdot \mathbf{I}_K)$ with $\mathbf{R}_k^{(\lambda)}$ is negligible). As a consequence, in step k we have $\mathcal{O}(m \cdot K)$ multiplications (and additions) of two elements of $\mathbb{D}$, each being of size bounded by $\mathcal{O}(k \ldots \kappa)$. Summing over $k = 0, \ldots, K-1$ gives the cost estimate as claimed above. ∎

The cost estimate $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$ of Algorithm FFFGnormal has to be compared with solving (5.2) by fraction-free Gaussian elimination, with cost given by $\mathcal{O}(K^5 \cdot \kappa^2)$. For the special case of matrix–Padé approximation, we gain a factor m in comparison with the method proposed in [10]. Let us mention already in this context that a modification of Algorithm FFFGnormal presented in the following section will have the same complexity in case of singularities, whereas the complexity may increase by a factor K for look-ahead methods such as [10].

7. The general recurrence: Nonperfect systems. In this section we present an algorithm that avoids nonnormal points by traveling around them along a path of "closest paranormal points." We will show that this path of closest paranormal points is separated for each order by at most one unit. The recurrence from section 6 will then be valid for this problem.

Let $\vec{n} = (\vec{n}^{(1)}, \ldots, \vec{n}^{(m)})$ be a multi-index. We will construct a sequence of multi-indices $(\vec{n}_k)_{k=0,\ldots,|\vec{n}|}$ with $|\vec{n}_k| = k$ and $\vec{n}_{|\vec{n}|} = \vec{n}$ along an offdiagonal path of indices, namely, a particular staircase of the form (6.3). At the same time we will construct a sequence of multi-indices $(\vec{\nu}_k)_{k=0,\ldots,|\vec{n}|}$ together with the corresponding Mahler systems $\mathbf{M}(\vec{\nu}_k, z)$. These points have the property that $\vec{\nu}_k = \vec{n}_k$ iff $\vec{n}_k$ is a normal point. Otherwise, the multi-index $\vec{\nu}_k$ is a k-normal point having a kind of "minimal distance" to the sequence $(\vec{n}_j)_j$ as specified below (see Theorem 7.3 and the subsequent remarks).

In order to simplify the presentation, we first introduce some properties for $m \times m$ polynomials which will hold for the Mahler systems computed below.

DEFINITION 7.1 ($\vec{n}$-Popov form, Popov-basis). *An $m \times m$ matrix polynomial $\mathbf{M}(z) \in \mathbb{Q}^{m\times m}[z]$ is in $\vec{n}$-Popov form (with row degree $\vec{\nu}$) if there exists a multi-index $\vec{\nu}$ such that $\mathbf{M}(z)$ satisfies the degree constraints*

$$z^{-\vec{\nu}} \cdot \mathbf{M}(z) = c \cdot \mathbf{I}_m + \mathcal{O}(z^{-1})_{z\to\infty}, \quad c \in \mathbb{Q} \setminus \{0\}, \tag{7.1}$$

$$z^{-\vec{n}} \cdot \mathbf{M}(z) \cdot z^{\vec{n}-\vec{\nu}} = \mathbf{T} + \mathcal{O}(z^{-1})_{z\to\infty}, \quad \mathbf{T} \in \mathbb{Q}^{m\times m} \text{ being upper triangular.} \tag{7.2}$$

If, in addition, the columns of $\mathbf{M}(z)$ have order $\ge \sigma$ with $\sigma \ge \sigma(|\vec{\nu}|)$, then $\mathbf{M}(z)$ will be referred to as a Popov-basis *of type $(\sigma, \vec{n})$.* ∎

Notice that the matrix $\mathbf{T}$ in (7.2) is necessarily nonsingular because of (7.1). Also, by multiplying with an appropriate constant we may suppose that $\mathbf{M}(z)$ has coefficients in $\mathbb{D}$ (in fact, we will only encounter Mahler systems). Up to a (unique) permutation of columns, we find the classical Popov normal form [37, subsection 6.7.2, p. 481] in the case $c = 1$ and $\vec{n} = \vec{0}$ (or $\vec{n} = [N, N, \ldots, N]$ since (7.2) is invariant under adding a constant to all components of $\vec{n}$). Here the row degree $\vec{\nu}$ is usually referred to as the vector of *controllability* or *Kronecker indices*. It is known [37, p. 484] that any square nonsingular matrix polynomial may be transformed to Popov normal form by multiplication on the right by a unimodular matrix polynomial and that the resulting polynomial is unique.[5] The introduction of an additional parameter $\vec{n}$ is natural in the context of the approximation problems of section 2. Also, by an appropriate choice of $\vec{n}$ we may force the matrix $\mathbf{M}(z)$ to be upper triangular, allowing us to include the Hermite normal form in our framework (see, e.g., [37, subsection 6.7.1, p. 476]).

The notion basis will become clear from Theorem 7.3(a) since any solution of order at least σ may be rewritten as a polynomial linear combination of the columns of a Popov-basis of type $(\sigma, \vec{n})$. For solutions of type $(\sigma, \vec{n})$ or, more generally, of type $(\sigma, \vec{n}_k)$ we may even be more precise. In fact, it is easy to see that the set of polynomial vectors of order $\geq \sigma$ forms a submodule over $\mathbb{Q}[z]$ of the module $\mathbb{Q}[z]^m$. Bases of such modules have already been successfully computed (not in a fraction-free way) by several authors [3, 5, 8, 9, 19, 20, 22, 40, 17, 51, 52, 53]. Here we may distinguish between two different kinds of algorithms (for a summary, see, e.g., [9]). For the hybrid (or look-ahead) methods in [19, 20, 22, 40, 53] one uses only order bases corresponding to normal or perfect points. In this case additional degree constraints are simple to describe (see, e.g., Corollary 5.4). In contrast, for the single step methods given in [3, 5, 8, 51, 52] only weaker degree constraints are imposed (for example, there is no longer uniqueness). A rather detailed study of the fine structure of degrees of bases in case of singular matrix–Padé approximation has been given in [17], based on a different computational path and a different normalization of bases. The approach used in this paper of combining order bases with Popov normal forms seems to be conceptionally simpler than that of [17], and easily extends to fraction-free computations.

In Algorithm FFFG (see Table 2) we compute a sequence of paranormal multi-indices $(\vec{\nu}_\sigma)_{\sigma=0,\ldots,K}$ together with the corresponding Mahler systems (up to a sign which may be determined by adapting (6.4)), using the fraction-free recurrence relation of Theorem 6.1. The efficient computation of the quantities $r^{(\ell)}$ is not specified. It can be implemented as described before Example 6.1. We establish in Theorem 7.2 below the connection to Popov-bases. In Theorem 7.3, we show in particular that we have solved the interpolation problem of Definition 2.1.

We remark that the algorithm has been implemented in the Maple computer algebra system with the code available from either author.

THEOREM 7.2 (feasibility of Algorithm FFFG). *Algorithm FFFG of Table* 2 *is well defined and gives the specified results (see also Theorem* 7.3(a)*): for any* $\sigma \geq 0$, *the multi-index* $\vec{\nu}_\sigma$ *is* σ*-normal, and* $\mathbf{M}_\sigma$ *coincides up to a sign with the Mahler system* $\mathbf{M}(\vec{\nu}_\sigma, z)$. *Furthermore,* $\mathbf{M}_\sigma$ *is a Popov-basis of type* $(\sigma, \vec{n})$ *with row-degree* $\vec{\nu}_\sigma$. *Finally,*

[5] These properties remain valid for the more general $\vec{n}$-Popov form [11]. As a consequence, we obtain uniqueness (up to a constant factor) of Popov-bases of a given type. A constructive proof of existence will be given in Theorem 7.2 below. In addition, it follows from Theorem 7.2 that a Popov-basis with row degree $\vec{\nu}$ coincides up to a constant with the (nontrivial) Mahler system $\mathbf{M}(\vec{\nu}, z)$.

TABLE 2
Algorithm FFFG.

ALGORITHM FFFG (on offdiagonal staircases)

INPUT: a vector of formal series f, a multi-index $\vec{n}$.

OUTPUT: For $\sigma = 0, 1, 2, \ldots, K$ with $\epsilon_\sigma \in \{-1, 1\}$:
$\vec{\nu}_\sigma$, a closest σ–normal point to $(\vec{n}_k)_{k=0,1,\ldots}$ defined by (6.3), (7.3),
Mahler systems $\mathbf{M}_\sigma = \epsilon_\sigma \cdot \mathbf{M}(\vec{\nu}_\sigma, z)$,
multigradients $d_\sigma = \epsilon_\sigma \cdot d^*(\vec{\nu}_\sigma)$,
basis for set of solutions of type $(\sigma, \vec{n}_k)$, $k \geq 0$:
$\{z^\ell \cdot \mathbf{M}_\sigma^{(\cdot,\mu)} : \ell = 0, 1, ..., \vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} - 1,\ \mu = 1, ..., m\}$.

INITIALIZATION: $\mathbf{M}_0 \leftarrow \mathbf{I}_m$, $d_0 \leftarrow 1$, $\vec{\nu}_0 \leftarrow \vec{0}$

ITERATIVE STEP: For $\sigma = 0, 1, 2, \ldots, K-1$:
Calculate for $\ell = 1, \ldots, m$:
first term of residuals $r^{(\ell)} \leftarrow c_\sigma(f \cdot \mathbf{M}_\sigma^{(\cdot,\ell)})$
Define set $\Lambda = \Lambda_\sigma = \{\ell \in \{1, \ldots, m\} : r^{(\ell)} \neq 0\}$.

If $\Lambda = \{\}$ **then** $\mathbf{M}_{\sigma+1} \leftarrow \mathbf{M}_\sigma$, $d_{\sigma+1} \leftarrow d_\sigma$, $\vec{\nu}_{\sigma+1} \leftarrow \vec{\nu}_\sigma$
else

Next closest para–normal point: $\vec{\nu}_{\sigma+1} \leftarrow \vec{\nu}_\sigma + \vec{e}_\pi$, where $\pi = \pi_\sigma \in \Lambda$ satisfies
$\pi = \min\{\ell \in \Lambda : \vec{n}^{(\ell)} - \vec{\nu}_\sigma^{(\ell)} = \max_{\mu\in\Lambda}\{\vec{n}^{(\mu)} - \vec{\nu}_\sigma^{(\mu)}\}\}$.

Calculate for $\ell = 1, \ldots, m$, $\ell \neq \pi$:
leading coefficients $p^{(\ell)} \leftarrow \mathit{coefficient}(\mathbf{M}_\sigma^{(\ell,\pi)}, z^{\vec{\nu}_\sigma^{(\ell)}-1})$.

Increase order for $\ell = 1, \ldots, m$, $\ell \neq \pi$:
$\mathbf{M}_{\sigma+1}^{(\cdot,\ell)} \leftarrow [\mathbf{M}_\sigma^{(\cdot,\ell)} \cdot r^{(\pi)} - \mathbf{M}_\sigma^{(\cdot,\pi)} \cdot r^{(\ell)}]/d_\sigma$
$\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \leftarrow (z - c_{\sigma,\sigma}) \cdot \mathbf{M}_\sigma^{(\cdot,\pi)}$

Adjust degree constraints:
$\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \leftarrow [\mathbf{M}_{\sigma+1}^{(\cdot,\pi)} \cdot r^{(\pi)} - \sum_{\ell\neq\pi} \mathbf{M}_{\sigma+1}^{(\cdot,\ell)} \cdot p^{(\ell)}]/d_\sigma$

New multigradient: $d_{\sigma+1} = r^{(\pi)}$
endif

with the assumptions of Theorem 6.2*, computing a Mahler system of order K has a cost of $\mathcal{O}(m \cdot K^4 \cdot \kappa^2)$.*

Proof. The first sentence of the assertion follows by recurrence on σ from Theorem 6.1, where in contrast to Algorithm FFFGnormal we also apply Theorem 6.1(a) in order to include the case $\Lambda = \{\}$. Let us verify (again by induction on σ) the link to Popov-bases. The assertion is trivially true for $\sigma = 0$ since $\mathbf{M}_0 = \mathbf{I}_m$. Suppose therefore that the statement holds for $\sigma \geq 0$, and let $\Lambda \neq \{\}$ (the case $\Lambda = \{\}$ is trivial). In what follows of the proof we write $\vec{\nu} = \vec{\nu}_\sigma$ and recall that $\vec{\nu}_{\sigma+1} = \vec{\nu} + \vec{e}_\pi$. We know already that $\mathbf{M}_{\sigma+1}$ is a Mahler system of type $\vec{\nu} + \vec{e}_\pi$, implying that the conditions on the order of the columns as well as the degree constraints (7.1) hold. To verify (7.2), notice that the recurrence of Theorem 6.1(c) may be rewritten in the form

$$\mathbf{M}_{\sigma+1} = \mathbf{M}_\sigma \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot d,$$

with some $d \in \mathbb{Q}$, and

$$\mathbf{P}_1 = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ a_1 & \cdots & a_{\pi-1} & z-a_\pi & a_{\pi+1} & \cdots & a_m \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 1 & & & b_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & b_{\pi-1} & & & \\ & & & 1 & & & \\ & & & b_{\pi+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & b_m & & & 1 \end{bmatrix}$$

with some $a_j, b_j \in \mathbb{Q}$, where we have not displayed the zero entries. By hypothesis we know that the limit for $z \to \infty$ of $z^{-\vec{n}} \cdot \mathbf{M}_\sigma \cdot z^{\vec{n}-\vec{\nu}}$ is upper triangular. Also, the limit of $z^{-\vec{n}+\vec{\nu}} \cdot \mathbf{P}_1 \cdot z^{\vec{n}-\vec{\nu}-\vec{e}_\pi}$ is upper triangular since, by construction of π, for $\ell = 1, \ldots, m$ we have either $a_\ell = 0$ or $\vec{n}^{(\pi)} - \vec{\nu}^{(\pi)} \geq \vec{n}^{(\ell)} - \vec{\nu}^{(\ell)}$, with equality only if $\ell \geq \pi$. In addition, $b_\ell \neq 0$ implies that $deg \mathbf{M}_\sigma^{(\ell,\pi)} = \vec{\nu}^{(\ell)} - 1$. Comparing with (7.2) for $\mathbf{M}_\sigma$, we get for $\ell = 1, \ldots, m$ either $b_\ell = 0$ or $\vec{n}^{(\pi)} - \vec{\nu}^{(\pi)} - 1 \leq \vec{n}^{(\ell)} - \vec{\nu}^{(\ell)}$, with equality only if $\ell < \pi$. Thus the limit of $z^{-\vec{n}+\vec{\nu}+\vec{e}_\pi} \cdot \mathbf{P}_2 \cdot z^{\vec{n}-\vec{\nu}-\vec{e}_\pi}$ is upper triangular, and we have established relation (7.2) for $\mathbf{M}_{\sigma+1}$. Finally, for a cost analysis we may fully apply the reasoning of the proof of Theorem 6.2 □

In order to describe further properties of the output of Algorithm FFFG, let us specify the offdiagonal path mentioned in the introduction of this section. The sequence $(\vec{n}_k)_{k=0,1,2,\ldots}$ is constructed at each step by increasing the index that has the furthest to go to reach $\vec{n}$, with ties broken by index order. That is, given an $\vec{n}_k$ we determine $\vec{n}_{k+1}$ by increasing the λth component by one where λ is chosen as $\vec{n}^{(\lambda)} - \vec{n}_k^{(\lambda)} = \max_\mu\{\vec{n}^{(\mu)} - \vec{n}_k^{(\mu)}\}$. If there is more than one choice of λ, then λ is the minimum index satisfying the maximality condition. In other words, we use the construction (6.3), where in each step $\lambda = \lambda_k$ satisfies

$$\lambda := \min\left\{\ell \in \{1, \ldots, m\} : \vec{n}^{(\ell)} - \vec{n}_k^{(\ell)} = \max_{\mu \in \{1,\ldots,m\}} \left\{\vec{n}^{(\mu)} - \vec{n}_k^{(\mu)}\right\}\right\}. \tag{7.3}$$

Thus, for example, if $\vec{n} = [1, 3, 3]$, then the sequence of 8 vectors are $[0, 0, 0]$, $[0, 1, 0]$, $[0, 1, 1]$, $[0, 2, 1]$, $[0, 2, 2]$, $[1, 2, 2]$, $[1, 3, 2]$, and $[1, 3, 3]$. In particular, notice that $\vec{n}_{|\vec{n}|} = [1, 3, 3]$, the multi-index of our original problem. The choice (7.3) of our particular staircase can also be understood as an elimination strategy in an extrapolation process with respect to an asymptotic scale of comparison. In fact, because of some numerical and theoretical results, this ordering was also preferred for GREP [29].

THEOREM 7.3 (properties of Algorithm FFFG).

(a) *For all $k, \sigma \geq 0$, the set of solutions of type $(\sigma, \vec{n}_k)$ (and thus the kernel of the matrix $\mathbf{K}(\vec{n}_k, \sigma)$) is spanned by*

$$z^j \cdot \mathbf{M}_\sigma^{(\cdot,\mu)}, \qquad j = 0, 1, \ldots, \vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} - 1, \qquad \mu = 1, \ldots, m.$$

(b) *For all $k, \sigma \geq 0$ we have*[6] *rank $\mathbf{K}(\vec{n}_k, \sigma) = |\min(\vec{\nu}_\sigma, \vec{n}_k)|$.*

(c) *A multi-index $\vec{\nu}$ verifying rank $\mathbf{K}(\vec{n}_k, \sigma) = |\min(\vec{\nu}, \vec{n}_k)|$ for $k \geq 0$ necessarily coincides with $\vec{\nu}_\sigma$.*

(d) *The multi-index $\vec{n}_k$ is σ-normal iff $\vec{n}_k = \vec{\nu}_\sigma$. In particular, $\vec{n}_k$ is normal iff $\vec{n}_k = \vec{\nu}_k$.*

Proof. For a proof of (a), (b), let us first mention that the $K := \max(\vec{0}, \vec{n}_k - \vec{\nu}_\sigma) = |\vec{n}_k| - |\min(\vec{\nu}_\sigma, \vec{n}_k)|$ polynomial vectors given in the assertion are linearly independent

[6] In what follows, the operations max, min for integer vectors are defined on a component basis.

over $\mathbb{Q}$ by Lemma 5.3(e) since $\mathbf{M}_\sigma$ essentially is a Mahler system. Let us show that they are all solutions of type $(\sigma, \vec{n}_k)$. From Theorem 7.2 we know that the order is correct. Furthermore, from (7.2) we have the degree constraints

$$deg\, M_\sigma^{(\ell,\mu)} \le \vec{n}^{(\ell)} - \vec{n}^{(\mu)} + \vec{\nu}_\sigma^{(\mu)} - \eta_{\ell,\mu}, \quad \ell, \mu = 1, \ldots, m,$$

with $\eta_{\ell,\mu} = 1$ if $\ell > \mu$ and $\eta_{\ell,\mu} = 0$ otherwise. Notice also that our offdiagonal staircase verifies

$$\vec{n}_k^{(\mu)} > 0 \quad \Longrightarrow \quad \vec{n}^{(\mu)} - \vec{n}_k^{(\mu)} \ge \vec{n}^{(\ell)} - \vec{n}_k^{(\ell)} - \eta_{\ell,\mu}, \quad \ell, \mu = 1, \ldots, m.$$

Hence in the case $\vec{n}_k^{(\mu)} - \vec{\nu}_\sigma^{(\mu)} > 0$ (and thus $\vec{n}_k^{(\mu)} > 0$) we get $deg\, M_\sigma^{(\ell,\mu)} \le \vec{n}_k^{(\ell)} - \vec{n}_k^{(\mu)} + \vec{\nu}_\sigma^{(\mu)}$, as required to show that the polynomial vectors of (a) are solutions of type $(\sigma, \vec{n}_k)$. Consequently, we have found K linearly independent elements of the kernel of $\mathbf{K}(\vec{n}_k, \sigma)$, showing that rank $\mathbf{K}(\vec{n}_k, \sigma) \le |\min(\vec{\nu}_\sigma, \vec{n}_k)|$. On the other hand, by Lemma 4.3 and the paranormality of $\vec{\nu}_\sigma$, a nonsingular submatrix of $\mathbf{K}(\vec{n}_k, \sigma)$ is given by $\mathbf{K}^*(\min(\vec{\nu}_\sigma, \vec{n}_k), |\min(\vec{\nu}_\sigma, \vec{n}_k)|)$, implying the assertions (a), (b).

In order to show (c), notice that by assumption and part (b) we have that $|\min(\vec{\nu}_\sigma, \vec{n}_k)| = |\min(\vec{\nu}, \vec{n}_k)|$ for all $k \ge 0$. If now $\vec{\nu} \ne \vec{\nu}_\sigma$, say, $\vec{\nu}^{(\ell)} < \vec{\nu}_\sigma^{(\ell)}$, then we may find a k such that $\vec{n}_k^{(\ell)} = \vec{\nu}^{(\ell)}$, and $\vec{n}_{k+1}^{(\mu)} = \vec{n}_k^{(\mu)} + \delta_{\ell,\mu}$, a contradiction.

Finally, for part (d) it is sufficient to prove the first sentence since $|\vec{n}_k| = k$. The σ-normality of $\vec{\nu}_\sigma$ has been already established in Theorem 7.2, and the final implication is a consequence of Lemma 4.3 and part (c). □

Notice that, for any k, and for any σ-normal multi-index $\vec{\nu}$, the matrix $\mathbf{K}(\vec{a}, \sigma)$, $\vec{a} := \min\{\vec{\nu}, \vec{n}_k\}$, is a submatrix both of $\mathbf{K}(\vec{n}_k, \sigma)$, and of $\mathbf{K}(\vec{\nu}, \sigma)$. Moreover, by Lemma 4.3, the latter matrix has maximal column rank. Therefore the Krylov matrix $\mathbf{K}(\vec{a}, \sigma)$ has full column rank, and, by Theorem 7.3(b),

$$|\max\{\vec{0}, \vec{n}_k - \vec{\nu}\}| = |\vec{n}_k| - \text{rank } \mathbf{K}(\vec{a}, \sigma) \ge |\vec{n}_k| - \text{rank } \mathbf{K}(\vec{n}_k, \sigma) = |\max\{\vec{0}, \vec{n}_k - \vec{\nu}_\sigma\}|. \tag{7.4}$$

Thus one may consider $\vec{\nu}_\sigma$ as the *closest σ-normal point* to the sequence $(\vec{n}_k)_k$, and in addition such a multi-index is unique according to Theorem 7.3(c). In order to illustrate this statement, we have drawn in Figure 1 in the classical C-table (i.e., Padé approximation, $m = 2$) an offdiagonal path together with the path of closest paranormal points. We also remark that the classic block structure of the Padé table is easily shown using Theorem 7.3(a) (cf. [9, Example 4.2]).

We may now establish the equivalent characterization of paranormal points as claimed in Corollary 5.4

Proof of Corollary 5.4. If $\vec{n}$ is σ-normal, then $\mathbf{M}(z) = \mathbf{M}(\vec{n}, z)$ has the required properties by construction. Let therefore $\mathbf{M}(z)$ be given as described above. We have shown in Theorem 7.2 that $\vec{\nu} := \vec{\nu}_\sigma$ is σ-normal, and hence $\sigma(|\vec{\nu}| - 1) < \sigma \le \sigma(|\vec{\nu}|)$, implying that $|\vec{\nu}| \ge |\vec{n}|$. On the other hand, the columns of $\mathbf{M}(z)$ are all solutions of type $(\sigma, \vec{n})$. Thus from Theorem 7.3(a) we know that there exists a matrix polynomial $\mathbf{P}(z)$ such that $\mathbf{M}(z) = \mathbf{M}_\sigma \cdot \mathbf{P}(z)$. Taking into account the degree assumption on $\mathbf{M}(z)$ and (7.2), we may conclude that the limit of $z^{\vec{\nu}-\vec{n}} \cdot \mathbf{P}(z)$ for $z \to \infty$ exists. Hence the components of $\vec{n} - \vec{\nu}$ have to be nonnegative, which together with $|\vec{\nu}| \ge |\vec{n}|$ implies that $\vec{\nu} = \vec{n}$. Consequently, $\vec{n}$ is σ-normal, and the representation of Corollary 5.4 follows from Lemma 5.3(c). □

Besides solving the approximation problems of section 2, we mention one further application for Algorithm FFFG.

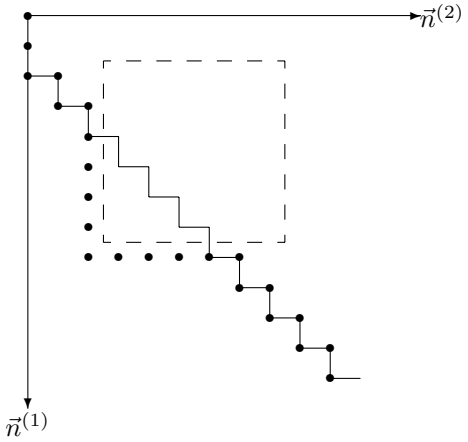


FIG. 1. *An example of singular Padé approximation. We have drawn the corresponding C-table of bigradients; here the dashed square indicates a singular block of zero-entries. By a straight line we denote the offdiagonal path induced by* $\vec{n} = (7, 6)$, *with the dots characterizing the corresponding closest paranormal points.*

Example 7.1 (fraction-free Hankel matrix solver). Suppose that we want to solve a system of linear equations

$$\mathbf{H} \cdot x = b, \qquad \mathbf{H} = [h_{i+j}]_{i,j=0\ldots n-1}, \qquad b = [b_j]_{j=n-1,\ldots,2n-2},$$

with a Hankel matrix of coefficients. If $h_j, b_j \in \mathbb{D}$, we may apply Algorithm FFFG in two different ways to obtain the Cramer solution $x^* := x \cdot \det \mathbf{H} \in \mathbb{D}^n$: First, as mentioned already in the context of (2.5), we may consider a (homogenous) Hermite–Padé approximation problem with $m = 3$, $f_1 = -1$, $f_2(z) = \sum h_j z^j$, $f_3(z) = -\sum b_j z^j$. It is easily shown that the resulting Sylvester matrix $\mathbf{K}(\vec{n}, 2n-1)$, $\vec{n} := [n-1, n, 0]$, is upper block triangular, with the left upper block being equal to the identity of order $n-1$ and the right lower block being equal to $\mathbf{H}$ up to a permutation of the columns. Hence $\det \mathbf{H} = \pm d(\vec{n})$ and $\mathbf{H}$ is nonsingular if and only if $\vec{n}$ is normal. In this case, $\mathbf{M}^{(3,3)}(\vec{n}, z) = \epsilon \cdot \det \mathbf{H}$, $\epsilon \in \{\pm 1\}$, and thus the coefficient vector of $\mathbf{M}^{(2,3)}(\vec{n}, z)$ is ϵ times the Cramer solution x^* of our Hankel system.

If one wants to solve the above system for multiple right-hand sides, it may be more interesting to get explicitly the adjoint $\det \mathbf{H} \cdot \mathbf{H}^{-1} \in \mathbb{D}^{n \times n}$. Again this can be done by Algorithm FFFG, using a well-known inversion formula for Hankel matrices: we compute the Mahler system $\mathbf{M}([n-1, n], z)$ corresponding to the Padé approximation problem $f = [f_1, f_2]$ and denote by $[q_j]_{j=0,\ldots,n-1}$ and $[v_j]_{j=0,\ldots,n}$, respectively, the coefficient vectors of $\mathbf{M}^{(2,1)}([n-1, n], z)$ and of $\mathbf{M}^{(2,2)}([n-1, n], z)$, $q_n := 0$. Then $v_n = \pm \det \mathbf{H}$, and from [36, section 1] we obtain (up to a sign) the adjoint of $\mathbf{H}$ by

$$v_n \cdot \mathbf{H}^{-1} = \tfrac{1}{v_n} \cdot \begin{bmatrix} v_n & & \\ \vdots & \ddots & \\ v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} q_{n-1} & \cdots & q_0 \\ \vdots & . & \\ q_0 & & \end{bmatrix}$$
$$- \tfrac{1}{v_n} \cdot \begin{bmatrix} q_n & & \\ \vdots & \ddots & \\ q_1 & \cdots & q_n \end{bmatrix} \begin{bmatrix} v_{n-1} & \cdots & v_0 \\ \vdots & . & \\ v_0 & & \end{bmatrix}.$$

Thus our algorithm gives a fraction-free method of solving systems of equations having

Hankel coefficient matrices. By reversing the order of columns one can state a similar result for Toeplitz systems. Note that in this case we have a Hankel, rather than a Toeplitz solver, since the algorithm solves all subproblems for nonsingular matrices along the principal diagonal of a Hankel matrix (principal antidiagonal of a Toeplitz matrix). The complexity of this solver is $\mathcal{O}(n^4 \cdot \kappa^2)$, which is faster than $\mathcal{O}(n^5 \cdot \kappa^2)$ required for fraction-free Gaussian elimination. One can also use our algorithm for fast fraction-free solving of linear systems having coefficient matrices that are block Hankel or block Hankel-like [40]. □

8. Fraction-free matrix GCD computations. Given two matrix polynomials A, B having s rows, with elements in $\mathbb{D}[z]$, the aim of this section is to show that Algorithm FFFG of section 7 enables us to compute a greatest common left divisor (GCLD) of A, B in a fraction-free way. Here it is convenient to combine A, B in a larger matrix $G = [A, B] \in \mathbb{D}[z]^{s\times m}$, where we suppose[7] that the rows of G are linearly independent over $\mathbb{Q}[z]$. We recall the well-known fact (see, e.g., [37, Lemma 6.3-3, p. 377]) that from a decomposition

$$G \cdot U = [A, B] \cdot U = [R, 0], \qquad R \in \mathbb{Q}[z]^{s\times s}, \qquad U \in \mathbb{Q}[z]^{m\times m}, \tag{8.1}$$

with U being unimodular (i.e., $\det U \in \mathbb{Q} \setminus \{0\}$) we may read off the solution of the matrix GCD problem: the matrix R is a GCLD (over $\mathbb{Q}[z]$) of A, B, and it is unique up to multiplication on the right by a unimodular matrix (in particular, the degree of its determinant is unique). Note that, by multiplying with a suitable element from $\mathbb{D}$, the matrices R, U of (8.1) may be chosen to have elements only from $\mathbb{D}[z]$. Algorithm FFFG will not only provide $R \in \mathbb{D}[z]^{s\times s}$ but also the cofactor matrix U.

The link to the interpolation problems of section 2 is given by reversing coefficients, i.e., z is replaced by $1/z$. We are then left with a vector Hermite–Padé approximation problem, with the corresponding system of functions $f \in \mathbb{D}[z]^{s\times m}$ being polynomial. However, the corresponding $(\mathbf{C}, \mathbf{F})$ is in general not controllable. Some results for the recursive solution of such a problem have been mentioned (without complete proofs) in [8, section 4] by exploiting the connections to power Hermite–Padé approximants.

In order to describe the complexity of our approach, we will make use of a result of Kung, Kailath, and Morf [39], [12, Theorem 1] on the rank of certain block Sylvester matrices. Here we require some definitions from the theory of matrix polynomials: the *degree* of a (rectangular) matrix polynomial C is the smallest integer N allowing a representation of the form $C(z) = C_0 + C_1 z + \cdots + C_N z^N$. The *McMillan degree* of C is the maximum of the degrees of the determinant of a maximal square submatrix of C (see, e.g., [37, 12, 51, 52]). We also need the concept of *minimal indices* [37, section 6.5.4] which are closely related to the controllability and Kronecker indices mentioned previously. The solutions $h \in \mathbb{Q}[z]^m$ of the equation $G \cdot h = 0$ form a submodule $\mathcal{M}$ of $\mathbb{Q}[z]^m$ of dimension $m - s$. We may find a basis of $\mathcal{M}$ given by the columns of $H = [h_1, \ldots, h_{m-s}] \in \mathbb{Q}[z]^{m\times(m-s)}$ such that H is column-reduced and irreducible [37, Theorem 6.5-10, p. 458]. Denoting by $\vec{\alpha}^{(j)}$ the degree of h_j, $j = 1, \ldots, m - s$, and $\vec{\alpha} = (\vec{\alpha}^{(1)}, \ldots, \vec{\alpha}^{(m-s)})$, it is known that $\vec{\alpha}$ is unique (up to a permutation) [37, Lemma 6.3-14] and that $|\vec{\alpha}|$ equals the McMillan degree of G minus the degree of the determinant of an GLCD.

We state the main result of this section in the following theorem.

[7] This restriction is natural since otherwise we may have GCDs with arbitrarily high degree [12], [37, p. 376 ff].

THEOREM 8.1 (GCLD via FFFG). *Let $G = [A, B] \in \mathbb{D}[z]^{s\times m}$ with degree N and* McMillan *degree $N^{\#}$. In addition, let $\vec{\alpha}$ be the vector of minimal indices of G, with its largest component*[8] *denoted by N^*. If we apply Algorithm FFFG to the data $f(z) = z^N \cdot G(1/z)$, $\vec{n} = (N, N, \ldots, N)$, $c_{j,k} = \delta_{j-s,k}$, with stopping criterion $\sigma = \sigma^*$ such that $f \cdot \mathbf{M}_\sigma$ is* reduced, *i.e., $f \cdot \mathbf{M}_\sigma$ contains only s columns different from zero, then we have the following:*

(a) *For $\sigma \geq \sigma^*$, the matrix $U_\sigma(z)$, a column permutation of $\mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$, is unimodular and verifies* (8.1). *Thus we have solved the extended GLCD problem.*

(b) *We have that $\sigma^* \leq \sigma' := s \cdot (N + N^* + 1)$. For N^* we have the worst case*[9] *estimate $N^* \leq N^{\#} \leq s \cdot N$.*

(c) *Suppose the coefficients occurring in G are all bounded in size by the constant κ. Then computing the GLCD by Algorithm FFFG has a worst case complexity of $\mathcal{O}(m\ (\sigma')^4\ \kappa^2)$.*

Proof. Denote by I the set of indices of the columns in $f \cdot M_{\sigma^*}$ which are different from zero. Note that I contains s elements by assumption on G and σ^*. Let $j \in \{1, \ldots, m\} \setminus I$, and $\sigma \geq \sigma^*$. Since $ord\,(f \cdot \mathbf{M}_{\sigma^*}^{(\cdot,j)}) = \infty$, one easily verifies by induction that the jth column of $\mathbf{M}_\sigma$ coincides with that of $\mathbf{M}_{\sigma^*}$ (up to some constant). In particular, $f \cdot \mathbf{M}_\sigma$ is reduced. For the assertion of part (a) it remains to show that $U(z) := \mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$ is a unimodular matrix polynomial. In fact, U is a matrix polynomial according to (7.2), and one easily verifies that $\det \mathbf{M}_\sigma = d \cdot z^{|\vec{\nu}_\sigma|}$ with $d \in \mathbb{D} \setminus \{0\}$. Therefore $\det U \in \mathbb{D}$, as claimed in part (a).

The set Λ appearing in Algorithm FFFG is a subset of I in any step where $\sigma \geq \sigma^*$. Therefore the components of $\vec{\nu}_\sigma$ with indices not in I remain invariant for $\sigma \geq \sigma^*$. For the remainder of the proof it will be convenient to reorder the columns of G (and thus simultaneously the rows and columns of $\mathbf{M}_\sigma$) such that $I = \{1, 2, \ldots, s\}$. Thus $U_\sigma(z) = \mathbf{M}_\sigma(1/z) \cdot z^{\vec{\nu}_\sigma}$ and $\vec{\nu}_\sigma = (\vec{c}_\sigma, \vec{a})$ for $\sigma \geq \sigma^*$, with some multi-index $\vec{a}$ having $m - s$ components.

In [12, Theorem 1] (see also [39]), the authors discuss the rank of transposed block Sylvester matrices which are given by $S_k := \mathbf{K}([k, \ldots, k], s \cdot (k + N))^T$ using our notation. It is shown that

$$\text{rank } S_k = |\min([\underbrace{k, \ldots, k}_{m}], [\underbrace{k, \ldots, k}_{s}, \vec{\alpha}])|, \quad k \geq 0,$$

where $\vec{\alpha}$ is the vector of minimal indices of G. Notice that $\mathbf{K}([k, \ldots, k], s \cdot (k+N))$ has a rhombus block structure, and that $\mathbf{K}([k, \ldots, k], \sigma)$ is obtained from $\mathbf{K}([k, \ldots, k], s \cdot (k + N))$ for $\sigma \geq s \cdot (k + N)$ by bordering $\sigma - s \cdot (k + N)$ zero rows. Consequently, with $\sigma = s \cdot (N + \ell)$, we get for $\ell = 0, 1, 2, \ldots$ and for $k = 0, 1, \ldots, \ell$

$$\text{rank } \mathbf{K}([k, \ldots, k], \sigma) = |\min([\underbrace{k, \ldots, k}_{m}], [\underbrace{k, \ldots, k}_{s}, \vec{\alpha}])| = |\min([\underbrace{k, \ldots, k}_{m}], \vec{\nu}_\sigma)|,$$

the final equality following from Theorem 7.3(b). We may conclude that the one partition of $\vec{\nu}_{\sigma'}$, namely $\vec{a}$, coincides up to a permutation with the vector $\vec{\alpha}$ of minimal indices, and that the other partition $\vec{c}_{\sigma'}$ contains only components strictly larger than N^*. Consider now $P(z) := H(1/z) \cdot z^{\vec{\alpha}}$, with $H \in \mathbb{Q}[z]^{m\times(m-s)}$ constituting a minimal

[8] If (without loss of generality) the McMillan degree of G is attained for $\det A$, then N^* is the minimal degree of a matrix polynomial $[C^T, D^T]$ allowing a representation $A^{-1} \cdot B = C \cdot D^{-1}$.

[9] As seen from the proof, it is more likely that N^* has the same magnitude as $N^{\#}/(m - s)$. In this case, σ' is at most of order $(N + 1) \cdot s \cdot m/(m - s)$.

basis as described before Theorem 8.1. Then $P \in \mathbb{Q}[z]^{m\times(m-s)}$, with its jth column having the degree $\vec{\alpha}^{(j)}$, and $f \cdot P = 0$. By Theorem 7.3(a), the columns of P may be represented as a polynomial linear combination of the columns of $\mathbf{M}_{\sigma'}$, that is, there exists a $Q \in \mathbb{Q}[z]^{m\times(m-s)}$ such that $P = \mathbf{M}_{\sigma'} \cdot Q$, and $\vec{z}^{\vec{\nu}_{\sigma'}} \cdot Q \cdot z^{-\vec{\alpha}}$ has a finite limit for $z \to \infty$. According to the special form of $\vec{\nu}_{\sigma'}$ we may conclude that the first s rows of Q vanish. Moreover, denoting by Q^* the (square) submatrix obtained from the last $m - s$ rows of Q, we know that $\vec{z}^{\vec{a}} \cdot Q^* \cdot z^{-\vec{\alpha}}$ has a finite limit. In addition, as with P, the columns of Q^* are also linearly independent over $\mathbb{Q}[z]$, and $|\vec{a}| = |\vec{\alpha}|$. Thus Q^* is unimodular, showing that $f \cdot \mathbf{M}_{\sigma'}$ is reduced, and hence $\sigma' \geq \sigma^*$. For a proof of part (b), it remains to establish the (rough) bound for N^*. Notice that $N^* \leq |\vec{\alpha}|$, with the latter quantity being bounded above by $N^{\#}$, the McMillan degree of G (see the remark before Theorem 8.1). The final estimate $N^{\#} \leq s \cdot N$ of part (b) is trivial. Finally, part (c) is a consequence of Theorem 7.2. □

Example 8.1. Let

$$A^*(z) := \begin{bmatrix} -3\,z+1 & 4\,z \\ 1 & -2 \end{bmatrix},\ B^*(z) := \begin{bmatrix} 2\,z+1 & -4\,z \\ z^2 & 3 \end{bmatrix},\ C(z) := \begin{bmatrix} 3\,z+1 & -3\,z \\ z^2 & z^2-z \end{bmatrix}.$$

We will compute the GCLD of the two matrix polynomials $A = C \cdot A^*$ and $B = C \cdot B^*$ using the method described in Theorem 8.1. The matrix polynomials A^* and B^* are shown to be left coprime, and so GCLDs of A and B are obtained by multiplying C on the right by some unimodular matrix. Here the combined matrix $G(z) = [A(z), B(z)]$ is given by

$$\begin{bmatrix} -9\,z^2-3\,z+1 & 12\,z^2+10\,z & -3\,z^3+6\,z^2+5\,z+1 & -12\,z^2-13\,z \\ -3\,z^3+2\,z^2-z & 4\,z^3-2\,z^2+2\,z & z^4+z^3+z^2 & -4\,z^3+3\,z^2-3\,z \end{bmatrix},$$

with $m = 4$, $s = 2$, and $N = 4$. We compute that $N^{\#} = 6 < s \cdot N$, while the vector of minimal indices is given by $\vec{\alpha} = (1, 2)$ (see below), and thus $N^* = 2$. Notice that $f(z) = z^N \cdot G(1/z)$ leads to a vector Hermite–Padé approximation problem where the data is not controllable (in fact, the first row of f is divisible by z^2). From Theorem 8.1 we know that Algorithm FFFG gives us a reduced basis (and thus a GCLD) at iteration σ^*, with $\sigma^* \leq \sigma' = 14$.

Using Algorithm FFFG we find that $\sigma^* = 11$, and $\vec{\nu}_{\sigma^*} = [3, 3, 2, 1]$ and hence we have computed $|\vec{\nu}_{11}| = 9$ different Mahler systems. It is quite instructive to have a look at the sequence of closest paranormal points $(\vec{\nu}_\sigma)_{0\leq\sigma\leq\sigma^*}$ which are given by

$$\begin{aligned}&[0,0,0,0],\ [0,0,0,0],\ [0,0,1,0],\ [0,0,1,0],\ [1,0,1,0],\ [1,0,2,0],\\ &[1,1,2,0],\ [1,1,2,1],\ [2,1,2,1],\ [2,2,2,1],\ [3,2,2,1],\ [3,3,2,1].\end{aligned}$$

Clearly, this staircase differs significantly from the offdiagonal staircase induced by $\vec{n} = [4, 4, 4, 4]$, that is, the "ideal" staircase contains only two paranormal points, and $[0, 0, 0, 0]$ is the only (trivially) normal point (the linear functionals c_0 and c_2 have been rejected). This illustrates why the reliable version of Algorithm FFFG as presented in section 7 is in fact needed.

We note some interesting points about the output of Algorithm FFFG. By reversing coefficients in $f \cdot \mathbf{M}_{11}$ and by eliminating the last two zero columns, we get the GCLD of A and B as the answer,

$$C^*(z) := \begin{bmatrix} -20736 & -124416\,z \\ -41472\,z^2+20736\,z & 41472\,z^2-41472\,z \end{bmatrix} = -20736 \cdot C(z) \cdot \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix},$$

with the factor on the right being unimodular. We observe in this example that the coefficients of the GCLD computed by Algorithm FFFG still have a common factor $d_{11} = -20736$. However, the prediction of such common factors (which also occur for Cramer solutions in other contexts) seems to be quite a difficult problem to solve. Also, notice that during our intermediate computations we have already factored out $\prod_{j=0}^{10} d_j$, a quantity which is of much bigger size than d_{11}. Finally, we observe that by partitioning

$$U(z) = \mathbf{M}_{11}(1/z) \cdot z^{\vec{\nu}_{11}} = \begin{bmatrix} U_1 & U_2 \\ U_3 & U_4 \end{bmatrix}$$

with blocks of size 2×2 we have found the cofactors in the diophantine equation $A \cdot U_1 + B \cdot U_2 = C^*$. Furthermore, $U_4 \cdot U_3^{-1}$ is the (irreducible) right coprime matrix fraction description of the rational function $B^{-1}A$.

For presentation purposes our example uses coefficients from the integers. A similar example could easily be constructed where the problem has parameters, for example, having coefficients from the domain $\mathbb{Q}[\epsilon]$, with ϵ an unknown. □

The significance of Mahler systems for the scalar GCD problem has been discussed in some detail in [10, section 6]. Here A, B are scalar polynomials, i.e., $m = 2$, $s = 1$, $N = \max(deg\, A, deg\, B) = N^{\#}$, and $N - N^*$ is the degree of the GCD C of A and B. The dimension of the largest Sylvester matrix encountered in Algorithm FFFG will be $N + N^*$, which may be larger than $deg\, A + deg\, B - deg\, C$, the dimension of the well-known critical Trudi submatrix. In fact, for a more efficient implementation one may choose instead of $\vec{n} = [N, \ldots, N]$ the "smallest" multi-index $\vec{n}$ such that $f(z) := G(1/z) \cdot z^{\vec{n}}$ is polynomial. Here the corresponding unimodular matrix is obtained by $z^{\vec{n}} \cdot \mathbf{M}_\sigma(1/z) \cdot z^{-\vec{n}+\vec{\nu}_\sigma}$.

9. Conclusions. In this paper we have presented algorithms for the computation of matrix rational interpolants and one-sided matrix greatest common divisors. The algorithms are fraction-free and designed to work in exact arithmetic domains where coefficient growth is a primary concern. The algorithms require no restrictions on input and are at least an order of magnitude faster than existing methods that compute solutions to the general problem. When specialized to cases such as Padé and matrix Padé approximation and scalar greatest common divisor computation, our approach is at least as efficient as existing fast fraction-free algorithms that work for these particular cases [10, 16, 21, 25].

Our method finds a basis for the $\mathbb{Q}[z]$-submodule of polynomial vectors of a given order, by recursively computing all bases of lower order. As such we find all possible solutions to the above interpolation problems. The methods also illustrate the advantages of considering the "closest normal points" of a given offdiagonal staircase of multi-indices which may contain nonnormal points. The approach taken in this paper differs from the method proposed in [10], which computes matrix Padé approximation by also using Mahler systems as its fundamental computation tool, but only at normal points. Problems corresponding to nonnormal points are "jumped" using fraction-free Gaussian elimination.[10] As a result, in cases where there are significant sized jumps their algorithm is potentially an order of magnitude less efficient than the one presented in this paper.[11]

[10] The method of "jumping" over singularities by some look-ahead strategy has been shown to be very useful in a numerical setting; see [6, 20, 23, 53]. Also, as shown in [10], there is a nice interpretation of such jumps in terms of modified Schur complements.

[11] Jumps of larger size are quite typical for matrix-GCD computations; see Example 8.1.

In the case of computing a scalar GCD, we do not use pseudodivisions in order to jump over problems associated to multi-indices being not (para)normal. This is in contrast to classical fraction-free methods for solving such problems [30]. In fact, we do not believe that our algorithm can be easily converted to recover the subresultant algorithm. Instead it is probably the case that one would have to choose bases different from Mahler systems ("comonic" instead of "monic" bases in the terminology of [9]), leading to some fraction-free variant of the algorithm of [17]. However, notice that, for large jumps, the size of the intermediate quantities in the subresultant algorithm [16, 25] (as well as in the algorithms of [10, 21]) may become significant. Our method, using closest normal points, does not have this drawback.

For some applications, it is of interest to follow computational paths different than the offdiagonal paths used in this paper. For example, it is of interest to obtain a Toeplitz instead of a Hankel solver. If this path consists of normal points, then one may apply the fraction-free algorithm of section 6. However, we are interested in giving a version that allows us to drop any regularity assumptions. Here, it might be possible to adapt the method of [5] to fraction-free arithmetic (or, alternatively, the methods in [9, 52]). In addition, in some applications such as Padé–Chebyshev approximation or state-space realizations in the theory of linear systems, one is interested in the case where the matrices $\mathbf{C}$ are lower Hessenberg instead of lower triangular. The corresponding special multiplication rule has the drawback that one decreases by one the order while multiplying by z. It is possible to adapt Algorithm FFFGnormal, but a generalization to singular cases is still an open problem.

As mentioned toward the end of section 2, the computation of matrix rational interpolants are related to the computation of both Popov and Hermite normal forms for matrices of polynomials. We plan to develop efficient fraction-free algorithms for these important computations, by combining our Algorithm FFFG with methods presented in [55]. Similarly it is of interest to see if our methods can be extended to Ore domains as done by Li [43] in the case of greatest common divisor computations of differential and difference operators.

Fraction-free algorithms are often important for theoretic reasons since they form the basis for generating exact algorithms based on modular reduction. We plan to investigate such algorithms for computing rational interpolants and matrix greatest common divisors. That these methods ultimately provide improved practical algorithms has been noted by Li [43] in the case of computing greatest common divisors of differential operators.

REFERENCES

[1] G.A. Baker and P.R. Graves-Morris, *Padé Approximants* , 2nd ed., Cambridge University Press, Cambridge, UK, 1995.

[2] E. Bareiss, *Sylvester's identity and multistep integer-preserving Gaussian elimination*, Math. Comp., 22 (1968), pp. 565–578.

[3] B. Beckermann, *Zur Interpolation mit polynomialen Linearkombinationen beliebiger Funktionen*, Ph.D. Thesis, Department of Applied Mathematics, University of Hannover, Hannover, Germany, 1990.

[4] B. Beckermann, *The structure of the singular solution table of the M-Padé approximation problem*, J. Comput. Appl. Math., 32 (1990), pp. 3–15.

[5] B. Beckermann, *A reliable method for computing M-Padé approximants on arbitrary staircases*, J. Comput. Appl. Math., 40 (1992), pp. 19–42.

[6] B. Beckermann, *The stable computation of formal orthogonal polynomials*, Numer. Algorithms, 11 (1996), pp. 1–23.

[7] B. Beckermann and G. Labahn, *A uniform approach for Hermite Padé and simultaneous Padé approximants and their matrix generalizations*, Numer. Algorithms, 3 (1992), pp. 45–54.

[8] B. Beckermann and G. Labahn, *A uniform approach for the fast, computation of matrix-type Padé approximants*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 804–823.

[9] B. Beckermann and G. Labahn, *Recursiveness in matrix rational interpolation problems*, J. Comput. Appl. Math., 77 (1997), pp. 5–34.

[10] B. Beckermann, S. Cabay, and G. Labahn, *Fraction–free Computation of Matrix Padé Systems*, in Proceedings of ISSAC'97, Maui, HI, ACM Press, New York, 1997, pp. 125–132.

[11] B. Beckermann, G. Labahn, and G. Villard, *Shifted Normal Forms of Polynomial Matrices*, in Proceedings of ISSAC'99, Vancouver, BC, ACM Press, New York, 1999, pp. 189–196.

[12] R.R. Bitmead, S.Y. Kung, B.D.O. Anderson, and T. Kailath, *Greatest common divisors via generalized Sylvester and Bezout matrices*, IEEE Trans. Automat. Control, 23 (1978), pp. 1043–1046.

[13] A.W. Bojanczyk, R.P. Brent, and F.R. de Hoog, *Stability analysis of a general Toeplitz systems solver*, Numer. Algorithms, 10 (1995), pp. 225–244.

[14] A.W. Bojanczyk, R.P. Brent, F.R. de Hoog, and D.R. Sweet, *On the stability of the Bareiss and related Toeplitz factorization algorithms*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 40–57.

[15] R. Brent, F.G. Gustavson, and D.Y.Y. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.

[16] W. Brown and J.F. Traub, *On Euclid's algorithm and the theory of subresultants*, J. ACM, 18 (1971), pp. 505–514.

[17] A. Bultheel and M. Van Barel, *A matrix Euclidean algorithm and the matrix minimal Padé approximation problem*, in Continued Fractions and Padé Approximants, C. Brezinski, ed., North–Holland, Amsterdam, pp. 11–51, 1990.

[18] S. Cabay and D.K. Choi, *Algebraic computations of scaled Padé fractions*, SIAM J. Comput., 15 (1986), pp. 243–270.

[19] S. Cabay, G. Labahn, and B. Beckermann, *On the theory and computation of non-perfect Padé-Hermite approximants*, J. Comput. Appl. Math., 39 (1992), pp. 295–313.

[20] S. Cabay, A.R. Jones, and G. Labahn, *Computation of numerical Padé-Hermite and simultaneous Padé systems* II*: A weakly stable algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 268–297.

[21] S. Cabay and P. Kossowski, *Power series remainder sequences and Padé fractions over an integral domain*, J. Symbolic Comput., 10 (1990), pp. 139–163.

[22] S. Cabay and G. Labahn, *A superfast algorithm for multidimensional Padé systems*, Numer. Algorithms, 2 (1992), pp. 201–224.

[23] S. Cabay and R. Meleshko, *A weakly stable algorithm for Padé approximants and the inversion of Hankel matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 735–765.

[24] S. Chandrasekaran and A.H. Sayed, *Stabilizing the generalized Schur algorithm*, SIAM J. Matrix Anal. Appl., 14 (1996), pp. 950–983.

[25] G. Collins, *Subresultant and reduced polynomial remainder sequences*, J. ACM, 14 (1967), pp. 128–142.

[26] S.R. Czapor and K.O. Geddes, *A comparison of algorithms for the symbolic computation of Padé approximants*, in Proceedings of EUROSAM'84, J. Fitch, ed., Lecture Notes in Comput. Sci. 174, Springer-Verlag, Berlin, 1984, pp. 248–259.

[27] R.W. Freund and H. Zha, *Formally biorthogonal polynomials and a look–ahead Levinson algorithm for general Toeplitz systems*, Linear Algebra Appl., 188/89 (1993), pp. 255–303.

[28] R.W. Freund and H. Zha, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math., 64 (1993), pp. 295–321.

[29] W.F. Ford and A. Sidi, *An algorithm for a generalization of the Richardson extrapolation process*, SIAM J. Numer. Anal., 24 (1987), pp. 1212–1232.

[30] K.O. Geddes, S.R. Czapor, and G. Labahn, *Algorithms for Computer Algebra*, Kluwer, Boston, MA, 1992.

[31] I. Gohberg, T. Kailath, and V. Olshevski, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1567.

[32] G. Golub and V. Olshevski, *Pivoting for Structured Matrices, with Applications*, http://www-isl.stanford.edu/∼olshevsk, 1997.

[33] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 279–306.

[34] M.H. Gutknecht, *Stable row recurrences for the Padé table and generically superfast look-ahead solvers for non-Hermitian Toeplitz systems*, Linear Algebra Appl., 188/89 (1993), pp. 351–421.
[35] M.H. Gutknecht and M. Hochbruck, *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz Systems*, Numer. Math., 70 (1995), pp. 181–227.
[36] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Oper. Theory Adv. Appl. 13, Birkhäuser Verlag, Basel, Boston, 1984.
[37] T. Kailath, *Linear Systems*, Prentice–Hall Englewood Cliffs, NJ, 1980.
[38] D. Knuth, *The Art of Computer Programming, Vol.* 2, Addison-Wesley, Reading, MA, 1981.
[39] S.Y. Kung, T. Kaliath, and M. Morf, *A generalized resultant matrix for polynomial matrices*, in Proceedings of the IEEE Conference on Decision and Control, Florida, 1976, pp. 892–895.
[40] G. Labahn, *Inversion Components of block Hankel-like matrices*, Linear Algebra Appl., 177 (1992), pp. 7–48.
[41] G. Labahn and S. Cabay, *Matrix Padé fractions and their computation*, SIAM J. Comput., 18 (1989), pp. 639–657.
[42] G. Labahn, D.K. Choi, and S. Cabay, *The inverses of block Hankel and block Toeplitz matrices*, SIAM J. Comput., 19 (1990), pp. 98–123.
[43] Z. Li, *A Subresultant Theory for Linear Differential, Linear Difference and Ore Polynomials, with Applications*, Ph.D. Thesis, Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria, 1996.
[44] W. Lübbe, *Über ein allgemeines Interpolationsproblem—Lineare Identitäten zwischen benachbarten Lösungssystemen*, Ph.D. Thesis, Department of Applied Mathematics, University of Hannover, Hannover, Germany, 1983.
[45] K. Mahler, *Perfect systems*, Composit. Math., 19 (1968), pp. 95–166.
[46] S. Paszkowski, *Quelques Algorithmes de l'Approximation de Padé–Hermite*, Publication ANO 89, University of Science and Technology at Lille, Lille, France, 1982.
[47] S. Paszkowski, *Recurrence relations in Padé-Hermite approximation*, J. Comput. Appl. Math., 19 (1987), pp. 99–107.
[48] S. Paszkowski, *Hermite Padé approximation: Basic notions and theorems*, J. Comput. Appl. Math., 32 (1990), pp. 229–236.
[49] B. Salvy and P. Zimmermann, *Gfun: A Maple package for the manipulation of generating and holonomic functions in one variable*, ACM Trans. Math. Software, 20 (1994), pp. 163–177.
[50] A. Sidi, *On a generalization of the Richardson extrapolation process*, Numer. Math., 57 (1990), pp. 365–377.
[51] M. Van Barel and A. Bultheel, *The computation of non-perfect Padé-Hermite approximants*, Numer. Algorithms, 1 (1991), pp. 285–304.
[52] M. Van Barel and A. Bultheel, *A general module theoretic framework for vector M-Padé and matrix rational interpolation*, Numer. Algorithms, 3 (1992), pp. 451–462.
[53] M. Van Barel and A. Bultheel, *A look–ahead algorithm for the solution of block Toeplitz systems*, Linear Algebra Appl., 266 (1997), pp. 291–335.
[54] M. Van Hoeij, *Factorization of differential operators with rational function coefficients*, J. Symbolic Comput., 24 (1997), pp. 537–561.
[55] G. Villard, *Computing Popov and Hermite forms of polynomial matrices*, in Proceedings of ISSAC'96, Zurich, ACM Press, New York, 1996, pp. 250–258.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 145–154

RATIONAL MATRIX FUNCTIONS AND RANK-1 UPDATES*

DANIEL S. BERNSTEIN† AND CHARLES F. VAN LOAN‡

Abstract. Suppose $f = p/q$ is a quotient of two polynomials and that p has degree r_p and q has degree r_q. Assume that $f(A)$ and $f(A+uv^T)$ are defined where $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^n$ are given and set $r = \max\{r_p, r_q\}$. We show how to compute $f(A+uv^T)$ in $O(rn^2)$ flops assuming that $f(A)$ is available together with an appropriate factorization of the "denominator matrix" $q(A)$. The central result can be interpreted as a generalization of the well-known Sherman–Morrison formula. For an application we consider a Jacobian computation that arises in an inverse problem involving the matrix exponential. With certain assumptions the work required to set up the Jacobian matrix can be reduced by an order of magnitude by making effective use of the rank-1 update formulae developed in this paper.

Key words. matrix functions, Sherman–Morrison

AMS subject classifications. 65F05, 65F99, 65L05

PII. S0895479898333636

1. Introduction.

Suppose $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^n$ are given and that f is a prescribed rational function that is defined on the spectrum of A and $A+uv^T$. In this paper we are concerned with the efficient computation of $f(A+uv^T)$ assuming that $f(A)$ is available.

The special case $f(z) = 1/z$ is well known:

$$(A+uv^T)^{-1} = A^{-1} - \alpha yz^T, \qquad y = A^{-1}u,\ z = A^{-T}v,\ \alpha = 1/(1+z^Tu).$$

This is the *Sherman–Morrison formula* and it can be used to compute $(A+uv^T)^{-1}$ from A^{-1} in $O(n^2)$ flops. See [2, p. 50]. In a linear equation setting, the Sherman–Morrison result, together with a QR factorization of A, makes it possible to solve $(A+uv^T)x = b$ in $O(n^2)$ flops.

Interest in a "fast" $f(A+uv^T)$ result can arise in several settings. For example, Benzi and Golub [1] present a Lanczos-based process for bounding certain matrix functions. Our generalized Sherman–Morrison result widens the class of problems that can be efficiently handled by their technique. Kenney and Laub [3] discuss condition estimation for matrix functions. With our results it is possible to compute more specialized sensitivity measures, e.g., how a rational function of a matrix A changes when a particular a_{ij} is varied.

The paper is organized as follows. In the next section we derive the generalized Sherman–Morrison formula and a closed-form expression for the derivative of $f(A)$ with respect to a_{ij}. Numerical results, stability issues, and a Jacobian evaluation application are discussed in section 3.

2. A generalized Sherman–Morrison result.

We first show that if A changes by a rank-1 matrix, then A^j changes by a rank-j matrix. Krylov matrices and exchange permutation matrices are involved. For $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$, and $j > 0$, the

*Received by the editors February 3, 1998; accepted for publication (in revised form) by R. Freund October 28, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/33363.html
†Department of Computer Science, University of Massachusetts, 140 Governor's Drive, Amherst, MA 01003 (bern@cs.umass.edu).
‡Department of Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853 (cv@cs.cornell.edu).

Krylov matrix $\mathrm{Kry}(A,x,j) \in \mathbb{R}^{n\times j}$ is defined by

$$\mathrm{Kry}(A,x,j) = \left[x,\ Ax,\ldots,A^{j-1}x\right] \in \mathbb{R}^{n\times j}.$$

We adopt the convention that $\mathrm{Kry}(A,x,j)$ is the empty matrix if $j=0$. The *exchange permutation matrix* $E_j \in \mathbb{R}^{j\times j}$ is just the identity matrix I_j with its columns in reverse order, i.e.,

$$E_j = I_j(:, j\!:\!-1\!:\!1)\,.$$

LEMMA 1. *If* $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, *and* $j > 0$, *then*

$$(A+uv^T)^j \;=\; A^j + K_j E_j L_j^T,$$

where $K_j = \mathrm{Kry}(A,u,j)$ *and* $L_j = \mathrm{Kry}(A^T+vu^T,v,j)$.

Proof. We use induction. The lemma is true for $j=1$ since $K_1 = u$, $E_1 = I_1$, and $L_1 = v$. Assume that the lemma holds for some $j \geq 1$. It follows that

$$\begin{aligned}
(A+uv^T)^{j+1} &= (A+uv^T)^j(A+uv^T) \;=\; (A^j + K_jE_jL_j^T)(A+uv^T)\\
&= A^{j+1} + K_jE_jL_j^T(A+uv^T) + A^juv^T\\
&= A^{j+1} + K_j((A^T+vu^T)L_jE_j)^T + (A^ju)v^T\\
&= A^{j+1} + K_j\left[\,(A^T+vu^T)^jv\,,\ldots,\,(A^T+vu^T)v\,\right]^T \;+\; (A^ju)v^T\\
&= A^{j+1} + \left[K_j\,,\ A^ju\right]\begin{bmatrix} v^T(A+uv^T)^j \\ \vdots \\ v^T(A+uv^T) \\ v^T \end{bmatrix}\\
&= A^{j+1} + K_{j+1}E_{j+1}L_{j+1}^T. \qquad \square
\end{aligned}$$

The computation of $(A+uv^T)^j$ from A^j requires approximately $6jn^2$ flops if the lemma is carefully exploited.

The next result shows that if A changes by a rank-1 matrix, then a degree-r polynomial in A changes by a rank-r matrix. *Hankel* matrices are involved and for $\alpha = (\alpha_1,\ldots,\alpha_r)$ we define

$$\mathrm{Hank}(\alpha) \;=\; \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_r \\ \alpha_2 & \alpha_3 & \cdots & \cdots & 0 \\ \alpha_3 & \vdots & \ddots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_r & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{r\times r}.$$

LEMMA 2. *If* $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, *and* $p(z) = \alpha_0 + \alpha_1 z + \cdots + \alpha_r z^r$ *with* $r > 0$, *then*

$$p(A+uv^T) \;=\; p(A) + K_rH_\alpha L_r^T,$$

where $K_r = \mathrm{Kry}(A,u,r)$, $L_r = \mathrm{Kry}(A^T+vu^T,v,r)$, $\alpha = (\alpha_1,\ldots,\alpha_r)$, *and* $H_\alpha = \mathrm{Hank}(\alpha)$.

Proof. Using Lemma 1 we have

$$p(A+uv^T) = \sum_{j=0}^{r} \alpha_j (A+uv^T)^j \;=\; \alpha_0 I \;+\; \sum_{j=1}^{r} \alpha_j (A^j + K_j E_j L_j^T)$$
$$= p(A) + \sum_{j=1}^{r} \alpha_j K_j E_j L_j^T,$$

where $K_j = \mathrm{Kry}(A,u,j)$, $L_j = \mathrm{Kry}(A^T + vu^T, v, j)$, and E_j is the j-by-j exchange permutation. Note that K_j and L_j are the first j columns of $K_r = \mathrm{Kry}(A,u,r)$ and $L_r = \mathrm{Kry}(A^T + vu^T, v, r)$, i.e., $K_j = K_r(:, 1{:}j)$ and $L_j = L_r(:, 1{:}j)$. Let `zeros(m,n)` denote the m-by-n zeros matrix as in MATLAB with the convention that it is the empty matrix if either argument is zero. It follows that

$$K_j E_j L_j^T = K_r \begin{bmatrix} E_j & 0 \\ 0 & \texttt{zeros}(r-j, r-j) \end{bmatrix} L_r^T$$

and so

$$p(A+uv^T) = p(A) \;+\; K_r \left(\sum_{j=1}^{r} \alpha_j \begin{bmatrix} E_j & 0 \\ 0 & \texttt{zeros}(r-j, r-j) \end{bmatrix} \right) L_r^T \,.$$

This proves the lemma since the matrix inside the parentheses is precisely the Hankel matrix H_α defined above. □

The computation of $p(A+uv^T)$ from $p(A)$ involves about $6n^2r + nr^2$ flops if the lemma is carefully exploited.

We are now set to prove that if A changes by a rank-1 matrix, then a rational function of A changes by a rank-r matrix where r is the maximum degree of the numerator and denominator polynomials.

THEOREM 3. *Suppose $f(z) = p(z)/q(z)$, where*

$$p(z) \;=\; \sum_{i=0}^{r_p} \alpha_i z^i \qquad \textit{and} \qquad q(z) \;=\; \sum_{i=0}^{r_q} \beta_i z^i.$$

Let $r = \max\{r_p, r_q\}$ and define the r-vectors

$$\tilde{\alpha} \;=\; (\, \alpha_1, \ldots, \alpha_{r_p}, \underbrace{0, \ldots, 0}_{r-r_p} \,) \qquad \textit{and} \qquad \tilde{\beta} \;=\; (\, \beta_1, \ldots, \beta_{r_q}, \underbrace{0, \ldots, 0}_{r-r_q} \,).$$

Suppose $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ and that $f(A)$ and $f(A+uv^T)$ are defined. Set $H_{\tilde{\alpha}} = \mathrm{Hank}(\tilde{\alpha})$ and $H_{\tilde{\beta}} = \mathrm{Hank}(\tilde{\beta})$. If

$$K_r = \mathrm{Kry}(A,u,r), \tag{1}$$
$$L_r = \mathrm{Kry}(A^T + vu^T, v, r), \tag{2}$$
$$Y_\alpha^T = H_{\tilde{\alpha}} L_r^T, \tag{3}$$
$$Y_\beta^T = H_{\tilde{\beta}} L_r^T, \tag{4}$$

then

$$f(A+uv^T) \;=\; f(A) + XY^T, \tag{5}$$

where

$$X = q(A)^{-1}K_r, \tag{6}$$

$$Y^T = Y_\alpha^T - M^{-1}Y_\beta^T(f(A) + XY_\alpha^T), \tag{7}$$

where

$$M = I_r + Y_\beta^T X. \tag{8}$$

Proof. Assume for clarity that both r_p and r_q are positive. The proof is easily adapted to handle the cases $r_p = 0$ and/or $r_q = 0$. (Various matrices and vectors below turn out to be empty.)

Set $\alpha = (\alpha_1, \ldots, \alpha_{r_p})$ and $\beta = (\beta_1, \ldots, \beta_{r_q})$ and define $H_\alpha = \text{Hank}(\alpha)$ and $H_\beta = \text{Hank}(\beta)$. Noting that $K_{r_p} = K_r(:, 1{:}r_p)$, $L_{r_p} = L_r(:, 1{:}r_p)$, $K_{r_q} = K_r(:, 1{:}r_q)$, and $L_{r_q} = L_r(:, 1{:}r_q)$, we see from Lemma 2 that

$$\begin{aligned} p(A + uv^T) = P + K_{r_p} H_\alpha L_{r_p}^T &= P + K_r H_{\tilde{\alpha}} L_r^T, \\ q(A + uv^T) = Q + K_{r_q} H_\beta L_{r_q}^T &= Q + K_r H_{\tilde{\beta}} L_r^T, \end{aligned}$$

where $P = p(A)$ and $Q = q(A)$. Thus,

$$\begin{aligned} f(A + uv^T) &= q(A + uv^T)^{-1} p(A + uv^T) \\ &= \left(Q + K_r H_{\tilde{\beta}} L_r^T\right)^{-1} \left(P + K_r H_{\tilde{\alpha}} L_r^T\right) \\ &= \left(I_n + Q^{-1} K_r H_{\tilde{\beta}} L_r^T\right)^{-1} Q^{-1} \left(P + K_r H_{\tilde{\alpha}} L_r^T\right) \\ &= \left(I_n + XY_\beta^T\right)^{-1} \left(F + XY_\alpha^T\right), \end{aligned}$$

where $F = f(A) = Q^{-1}P$. By the Sherman–Morrison–Woodbury formula [2, p. 50],

$$\left(I_n + XY_\beta^T\right)^{-1} = I_n - XM^{-1}Y_\beta^T,$$

where $M = I_r + Y_\beta^T X$ and so

$$\begin{aligned} f(A + uv^T) &= \left(I_n - XM^{-1}Y_\beta^T\right)\left(F + XY_\alpha^T\right) \\ &= F + X\left(Y_\alpha^T - M^{-1}Y_\beta^T(F + XY_\alpha^T)\right) \\ &= F + X\left(Y_\alpha^T - (I_r + Y_\beta^T X)^{-1}Y_\beta^T(F + XY_\alpha^T)\right) \\ &= F + XY^T. \qquad \square \end{aligned}$$

The computation of $f(A + uv^T)$ from $f(A)$ via (1)–(4) and (6)–(8) requires $\text{O}(n^2 r)$ flops. Indeed, if we assume that $r = r_p = r_q$ and that a QR factorization of the denominator polynomial $q(A)$ is available, then the work is distributed as follows:

Calculation	Flops
K_r	$2n^2r$
L_r	$2n^2r$
Y_α	nr^2
Y_β	nr^2
X	$3n^2r$
M	$2nr^2$
Y	$4n^2r + 2nr^2$

This totals to $11n^2r$ flops if we assume that $r \ll n$. As with any Sherman–Morrison-type computation, the condition numbers of $q(A)$ and the matrix M defined by (8) should be monitored because they shed light on the accuracy of the computed update.

Note that Theorem 3 can be generalized in the direction of the Sherman–Morrison–Woodbury formula (see [2, p. 50]). In particular, if UV^T is a rank-d matrix and $f(A + UV^T)$ exists, then it can be shown that $f(A)$ and $f(A + UV^T)$ differ by a matrix that has rank dr.

We conclude this section by using Theorem 3 to develop an expression for the partial derivative of $f(A)$ with respect to a particular matrix element a_{ij}, i.e.,

$$\frac{\partial}{\partial a_{ij}} f(A) \;=\; \lim_{\delta \to 0} \frac{f(A + \delta e_i e_j^T) - f(A)}{\delta},$$

where $I_n = [\, e_1, \ldots, e_n \,]$. The idea is to use Theorem 3 to simplify the difference between $f(A)$ and $f(A + \delta e_i e_j^T)$.

COROLLARY 4. *Suppose* $f(z) = p(z)/q(z)$, *where*

$$p(z) \;=\; \sum_{i=0}^{r_p} \alpha_i z^i \qquad \textit{and} \qquad q(z) \;=\; \sum_{i=0}^{r_q} \beta_i z^i.$$

Let $r = \max\{r_p, r_q\}$ *and define the* r*-vectors*

$$\tilde{\alpha} \;=\; (\, \alpha_1, \ldots, \alpha_{r_p}, \underbrace{0, \ldots, 0}_{r - r_p} \,) \qquad \textit{and} \qquad \tilde{\beta} \;=\; (\, \beta_1, \ldots, \beta_{r_q}, \underbrace{0, \ldots, 0}_{r - r_q} \,).$$

Assume that $A \in \mathbb{R}^{n \times n}$, $f(A)$ *is defined, and* $1 \le i, j \le n$. *If* $H_{\tilde{\alpha}} = \mathrm{Hank}(\tilde{\alpha})$, $H_{\tilde{\beta}} = \mathrm{Hank}(\tilde{\beta})$, $Q = q(A)$, $K^{(i)} = \mathrm{Kry}(A, e_i, r)$, *and* $L^{(j)} = \mathrm{Kry}(A^T, e_j, r)$, *then*

$$\frac{\partial}{\partial a_{ij}} f(A) \;=\; X^{(i)} Z^{(j)T},$$

where

$$X^{(i)} = Q^{-1} K^{(i)}, \tag{9}$$

$$Z^{(j)T} = H_{\tilde{\alpha}} L^{(j)T} - H_{\tilde{\beta}} L^{(j)T} f(A). \tag{10}$$

Proof. Since $f(A)$ is defined, then the matrix $f(A + \delta e_i e_j^T)$ is also defined for all δ less than or equal to some sufficiently small δ_0. Assuming that $\delta \le \delta_0$, we apply Theorem 3 with $u = e_i$ and $v = \delta e_j$. It follows that if

$$\begin{aligned} L(\delta) &= \mathrm{Kry}(A^T + \delta e_j e_i^T, \delta e_j, r), \\ Y_\alpha(\delta)^T &= H_{\tilde{\alpha}} L(\delta)^T, \\ Y_\beta(\delta)^T &= H_{\tilde{\beta}} L(\delta)^T, \end{aligned}$$

then

$$f(A + \delta e_i e_j^T) = f(A) + X^{(i)} Y(\delta)^T,$$

where

$$Y(\delta)^T \;=\; Y_\alpha(\delta)^T - (I_r + Y_\beta(\delta)^T X^{(i)})^{-1} Y_\beta(\delta)^T (f(A) + X^{(i)} Y_\alpha(\delta)^T).$$

Since

$$L(\delta) \;=\; \delta \cdot \mathrm{Kry}(A^T + \delta e_j e_i^T, e_j, r),$$

we see that

$$\lim_{\delta\to 0} \frac{Y_\alpha(\delta)}{\delta} = H_{\tilde{\alpha}} L^{(j)T} \qquad \text{and} \qquad \lim_{\delta\to 0} \frac{Y_\beta(\delta)}{\delta} = H_{\tilde{\beta}} L^{(j)T}.$$

Since $Y_\alpha(\delta)$ and $Y_\beta(\delta)$ both converge to zero as $\delta \to 0$, it follows that

$$\lim_{\delta\to 0} Y(\delta)^T = \lim_{\delta\to 0} \left(\frac{Y_\alpha(\delta)^T}{\delta} \;-\; (I_r + Y_\beta(\delta)^T X^{(i)})^{-1} \frac{Y_\beta(\delta)^T}{\delta} (f(A) + X^{(i)} Y_\alpha(\delta)^T) \right)$$

$$= H_{\tilde{\alpha}} L^{(j)T} - H_{\tilde{\beta}} L^{(j)T} f(A) \;\equiv\; Z^{(j)T}. \qquad \square$$

Note that if f is a polynomial, then $H_{\tilde{\beta}} = 0$.

3. Discussion. We have written a MATLAB function

```
[XP,YP,XQ,YQ,XF,YF,condM] = Rational_Update(A,u,v,alfa,P,beta,Q_cell)
```

that implements the update formulae derived in the proof of Theorem 3. The vectors `alfa` and `beta` contain the coefficients of the numerator and denominator polynomials p and q, respectively. The QR factorization of $Q = q(A)$ is passed via a cell array representation `Q_cell`. If `A` is n-by-n and r is the larger of $\deg(p)$ and $\deg(q)$, then the output matrices `XP`, `YP`, `XQ`, `YQ`, `XF`, and `YF` are n-by-r and relate $p(A)$ to $p(A + uv^T)$, $q(A)$ to $q(A + uv^T)$, and $f(A)$ to $f(A + uv^T)$ as follows:

$$\begin{aligned} p(A + uv^T) &= P + X_P Y_P^T, \\ q(A + uv^T) &= Q + X_Q Y_Q^T, \\ f(A + uv^T) &= F + X_F Y_F^T. \end{aligned}$$

The r-by-r matrix M defined by (8) has an important role to play in the calculation of Y_F, and so its condition number is returned.

Table 1 reports on the quality of $f(A + uv^T)$ when f is the diagonal Pade approximation to the exponential function:

$$f(A) = \left(\sum_{\mu=0}^{r} \beta_\mu A^\mu \right)^{-1} \left(\sum_{\mu=0}^{r} \alpha_\mu A^\mu \right), \quad \alpha_\mu = \frac{(2r-\mu)!r!}{(2r)!\mu!(r-\mu)!}, \quad \beta_\mu = (-1)^\mu \alpha_\mu. \tag{11}$$

See [2, p. 572]. `Rational_Update` is used to update $f(A)$. The matrix A is a randomly generated 100-by-100 example with eigenvalues in the left-half plane and $\| A \|_2 = 15$. The denominator matrix $q(A)$ is well conditioned. The rank-1 correction uv^T has unit 2-norm in the test case. Define

$$\begin{aligned} \tilde{F}_0 &= \texttt{expm(A)}, \\ F_0 &= \text{the } (p,p) \text{ Pade approximation to } \exp(A), \\ \tilde{F}_1 &= \texttt{expm(A+u*v')}, \\ F_1 &= \text{the } (p,p) \text{ Pade approximation to } \exp(A + uv^T), \\ F_1^{(up)} &= \text{an estimate of } F_1 \text{ obtained by updating } F_0, \end{aligned}$$

TABLE 1
Errors associated with the update of the (r,r)-Pade approximation to e^A.

r	$\frac{\| F_0 - \tilde{F}_0 \|_2}{\| \tilde{F}_0 \|_2}$	$\frac{\| F_1 - \tilde{F}_1 \|_2}{\| \tilde{F}_1 \|_2}$	$\frac{\| F_1 - F_1^{(up)} \|_2}{\| F_1 \|_2}$	`cond(Q)`	`cond(M)`
0	2.00e+001	1.99e+001	0.00e+000	1.0e+000	1.0e+000
1	1.56e+001	1.55e+001	2.01e−015	3.6e+000	1.0e+000
2	9.42e+000	9.40e+000	3.17e−015	8.9e+000	1.0e+000
3	4.36e+000	4.35e+000	7.66e−015	1.7e+001	1.5e+000
4	1.52e+000	1.52e+000	2.31e−014	2.7e+001	4.7e+000
5	4.01e−001	4.03e−001	3.95e−014	3.9e+001	9.9e+000
6	8.08e−002	8.14e−002	4.46e−014	5.0e+001	3.5e+002
7	1.27e−002	1.28e−002	4.97e−014	6.2e+001	1.2e+004
8	1.59e−003	1.61e−003	5.35e−014	7.3e+001	2.7e+005
9	1.62e−004	1.64e−004	5.80e−014	8.3e+001	4.9e+006
10	1.37e−005	1.39e−005	6.15e−014	9.3e+001	7.9e+007
11	9.76e−007	9.88e−007	6.36e−015	1.0e+002	1.1e+009
12	5.94e−008	6.00e−008	6.88e−014	1.1e+002	1.5e+010
13	3.12e−009	3.14e−009	6.93e−014	1.2e+002	1.8e+011
14	1.43e−010	1.43e−010	7.46e−014	1.2e+002	2.0e+012
15	5.77e−012	5.75e−012	7.80e−014	1.3e+002	2.1e+013
16	2.06e−013	2.05e−013	7.89e−014	1.4e+002	2.0e+014

where `expm` is the built-in MATLAB function for the matrix exponential. In this study we treat `expm` as exact, thereby enabling us to report on the relative error in F_0 and F_1. We are well aware of the difficulties associated with e^A calculations, but this example is not numerically challenging for `expm`.

An interesting aspect of Table 1 is that the 2-norm relative error in $F_1^{(up)}$ is consistently small even as the condition of the matrix M deteriorates with increasing r. We have no analysis or informal explanation for this phenomena, which (it turns out) is quite typical. The Krylov matrices that "make up" the matrix M and the "right-hand-side matrix" $Y_\beta^T(f(A) + XY_\alpha^T)$ in (7) are notoriously ill conditioned, especially for large values of r. But somehow the effect of this ill-conditioning is muted. It is perhaps worth noting that Krylov matrices are always involved (at least implicitly) with *any* matrix polynomial calculation because

$$\sum_{i=0}^{r} \alpha_i A^i \;=\; [I, A, \ldots, A^r] \begin{bmatrix} \alpha_0 I \\ \alpha_1 I \\ \vdots \\ \alpha_r I \end{bmatrix}.$$

Apparently there is not a simple connection between the condition of a matrix polynomial computation and the condition of the Krylov matrices that lurk in the background. Clearly, more research in this area is required, especially if high-degree polynomials are involved.

We now proceed to a discussion of Corollary 4 and how it can be applied in a systems identification context. Suppose the value of a scalar-valued function

$$y(t) = c^T e^{At} b, \qquad b, c \in \mathbb{R}^n,\ A \in \mathbb{R}^{n\times n},$$

is known at $t = t_1, \ldots, t_m$ with $m \geq n^2$ and that from that data we want to reconstruct A. Defining

$$y_k = y(t_k), \qquad k = 1{:}m,$$

we see that y_k is a snapshot of the solution to the initial value problem

$$\dot{x} = Ax,\ x(0) = b$$

"as seen through" the observation vector c, i.e., $y_k = c^T x(t_k)$.

There are several ways to attack this difficult identification problem (see [4, 5]). One approach is to approximate e^{At_k} with a rational function $f_k(A)$ for $k = 1{:}m$ and then to minimize $\| \phi(A) - y \|_2$, where

$$\phi(A) = \begin{bmatrix} c^T f_1(A)b \\ c^T f_2(A)b \\ \vdots \\ c^T f_m(A)b \end{bmatrix} \qquad \text{and} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

This is a nonlinear least square problem in the n^2 entries that define the matrix $A = (a_{ij})$. If the Levenberg–Marquardt algorithm is applied, then (among other things) at each step we need to compute the m-by-n^2 Jacobian J of the vector-valued function $\phi(A)$.

Suppose an ODE solver is used to generate $f_k(A)b \approx x(t_k)$ for $k = 1{:}m$. If each of these vectors requires $\mathrm{O}(n^2)$ flops to compute, then a ϕ-evaluation costs $\mathrm{O}(mn^2)$ flops and a finite-difference approximation to J would involve $\mathrm{O}(mn^4)$ flops. We show how this flop count can be reduced by a factor that ranges from n to n^2.

Note that the kth row of the Jacobian involves partial derivatives of the scalar-valued function $c^T f_k(A)b$ with respect to each of the matrix entries a_{ij}:

$$\frac{\partial}{\partial a_{ij}} \left(c^T f_k(A)b \right) \ = \ c^T \left(\frac{\partial}{\partial a_{ij}} f_k(A) \right) b.$$

Corollary 4 is therefore applicable. Let us see how we might use the corollary to simplify the evaluation of these Jacobian entries. Assume that f_k is the (r_p, r_q) Pade approximation to e^{At_k}, i.e.,

$$f_k(A) = q_k(A)^{-1} p_k(A),$$

$$p_k(A) = \left(\sum_{\mu=0}^{r_p} \alpha_\mu^{(k)} A^\mu \right),$$
$$q_k(A) = \left(\sum_{\mu=0}^{r_q} \beta_\mu^{(k)} A^\mu \right)$$

with

$$\alpha_\mu^{(k)} = \frac{(r_p + r_q - \mu)! r_p!}{(r_p + r_q)! \mu! (r_p - \mu)!} t_k^\mu \quad \text{and} \quad \beta_\mu^{(k)} = \frac{(r_p + r_q - \mu)! r_q!}{(r_p + r_q)! \mu! (r_q - \mu)!} t_k^\mu.$$

Using Corollary 4,

$$c^T \left(\frac{\partial}{\partial a_{ij}} f_k(A) \right) b \ = \ c^T X_k^{(i)} Z_k^{(j)T} b, \tag{12}$$

where

$$X_k^{(i)} = q_k(A)^{-1}K^{(i)}, \tag{13}$$

$$Z_k^{(i)} = H_{\alpha^{(k)}}L^{(j)T} - H_{\beta^{(k)}}L^{(j)T}q_k(A)^{-1}p_k(A). \tag{14}$$

Note that if $r = \max\{r_p, r_q\}$ and we precompute and store the matrix powers $A^2, \ldots, A^r$, then the Krylov matrices $K^{(i)}$ and $L^{(j)}$ are easily retrieved and the matrices $p_k(A)$ and $q_k(A)$ can be set up in $O(n^2 r)$ flops. For a given Jacobian row index k, we must compute the vectors $c_{k1}, \ldots, c_{kn} \in \mathbb{R}^r$ defined by

$$c_{ki}^T = c^T q_k(A)^{-1}K^{(i)},\ i = 1{:}n, \tag{15}$$

and the vectors $b_{k1}, \ldots, b_{kn} \in \mathbb{R}^r$ defined by

$$b_{ki} = H_{\alpha^{(k)}}L^{(i)T}b - H_{\beta^{(k)}}L^{(i)T}q_k(A)^{-1}p_k(A)b,\ i = 1{:}n. \tag{16}$$

The kth row of the Jacobian is then made up of the inner products $c_{ki}^T b_{kj}$, where i and j each range from 1 to n. Summarizing the overall Jacobian evaluation we get

Compute and save the matrix powers $A^2, \ldots, A^r$.
For $k = 1{:}m$
  Set up $q_k(A)$ and $p_k(A)$, and compute the QR factorization of the former.
  For $i = 1{:}n$
    Compute the vectors c_{ki} and b_{ki} defined by (15) and (16).
  end
  Establish the kth row of the Jacobian by computing the inner products
    $c_{ki}^T b_{kj}$, $i = 1{:}n, j = 1{:}n$.
end

The powers of A cost $\mathrm{O}(rn^3)$. As we mentioned above, once these powers are available, then the Krylov matrices $K^{(i)}$ and $L^{(j)}$ that figure in the computations are "free." For each k there is an $O(n^3)$ QR factorization. The vectors $c_{k1}, \ldots, c_{kn}$ and $b_{k1}, \ldots, b_{kn}$ can altogether be computed in $\mathrm{O}(rn^2)$ flops. The n^2 inner products $c_{ki}^T b_{kj}$ cost another $O(rn^2)$ flops. Thus, each row of the Jacobian requires $O(rn^2 + n^3)$ flops. The total Jacobian evaluation as outlined is therefore an $O(rmn^2 + mn^3)$ computation, a factor of n less than the ODE finite-difference approach.

We note that if the Pade approximation is a polynomial ($r_q = 0$), then a number of simplifications result:

Compute and save the matrix powers $A^2, \ldots, A^r$.
For $i = 1{:}n$
  Compute $c_i^T = c^T K^{(i)}$ and $b_i = L^{(i)T}b$.
end
For $k = 1{:}m$
  Compute the inner products $c_i^T H_{\alpha^{(k)}} b_j$, $i = 1{:}n, j = 1{:}n$.
end

The flop count now is just $\mathrm{O}(rmn^2)$, which is less than the ODE finite-difference approach by a factor of n^2.

A numerical issue here concerns the size of r. As we mentioned earlier, the Krylov-related computations can be problematic if this parameter is too large. However, in some contexts it is possible to base the identification process on observations of $c^T e^{At} b$ at values $t = t_1, \ldots, t_m$ that are small in value. In this case a low-order Pade approximation to e^{At_k} can suffice.

MATLAB software for carrying out the updates discussed in this paper is available from http://www.cs.cornell.edu/cv.

Acknowledgments. The authors became interested in this problem as a result of discussions with Professor Martha Contreras of the Biometrics Unit at Cornell University. Mike Todd read an earlier draft of this paper, spotted some errors, and showed us that the exact Jacobian could be obtained in section 2.

REFERENCES

[1] M. Benzi and G.H. Golub, *Bounds for the entries of matrix functions with applications to preconditioning*, BIT, 39 (1999), pp. 417–438.
[2] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1997.
[3] C.S. Kenney and A.J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.
[4] R.I. Jennrich and P.B. Bright, *Fitting systems of linear differential equations using computer generated exact derivatives*, Technometrics, 18 (1976), pp. 385–392.
[5] L. Ljung, *System Identification—Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 155–172

OPTIMAL KRONECKER PRODUCT APPROXIMATION OF BLOCK TOEPLITZ MATRICES*

JULIE KAMM† AND JAMES G. NAGY‡

Abstract. This paper considers the problem of finding $n \times n$ matrices A_k and B_k that minimize $||T - \sum A_k \otimes B_k||_F$, where $\otimes$ denotes Kronecker product and T is a banded $n \times n$ block Toeplitz matrix with banded $n \times n$ Toeplitz blocks. It is shown that the optimal A_k and B_k are banded Toeplitz matrices, and an efficient algorithm for computing the approximation is provided. An image restoration problem from the Hubble Space Telescope (HST) is used to illustrate the effectiveness of an approximate SVD preconditioner constructed from the Kronecker product decomposition.

Key words. block Toeplitz matrix, conjugate gradient method, Kronecker product, image restoration, preconditioning, singular value decomposition

AMS subject classifications. 65F20, 65F30

PII. S0895479898345540

1. Introduction.

A Toeplitz matrix is characterized by the property that its entries are constant on each diagonal. Toeplitz and block Toeplitz matrices arise naturally in many signal and image processing applications; see, for example, Bunch [4] and Jain [17] and the references therein. In image restoration [21], for instance, one needs to solve large, possibly ill-conditioned linear systems in which the coefficient matrix is a banded block Toeplitz matrix with banded Toeplitz blocks (BTTB).

Iterative algorithms, such as conjugate gradients (CGs), are typically recommended for large BTTB systems. Matrix-vector multiplications can be done efficiently using fast Fourier transforms [14]. In addition, convergence can be accelerated by preconditioning with block circulant matrices with circulant blocks (BCCB). A circulant matrix is a Toeplitz matrix in which each column (row) can be obtained by a circular shift of the previous column (row), and a BCCB matrix is a natural extension of this structure to two dimensions; cf. Davis [10].

Circulant and BCCB approximations are used extensively in signal and image processing applications, both in direct methods which solve problems in the "Fourier domain" [1, 17, 21] and as preconditioners [7]. The *optimal circulant preconditioner* introduced by Chan [8] finds the closest circulant matrix in the Frobenius norm. Chan and Olkin [9] extend this to the block case; that is, a BCCB matrix C is computed to minimize

$$||T - C||_F .$$

BCCB approximations work well for certain kinds of BTTB matrices [7], especially if the unknown solution is almost periodic. If this is not the case, however, the performance of BCCB preconditioners can degrade [20]. Moreover, Serra-Capizzano and Tyrtyshnikov [6] have shown recently that it may not be possible to construct a BCCB preconditioner that results in superlinear convergence of CGs.

*Received by the editors October 5, 1998; accepted for publication (in revised form) by L. Reichel November 6, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/34554.html
†Raytheon Systems Company, Dallas, TX 75266 (j-kamm@ti.com).
‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (nagy@mathcs.emory.edu).

Here we consider an alternative approach: optimal Kronecker product approximations. A Kronecker product $A \otimes B$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{bmatrix}.$$

In particular, we consider the problem of finding matrices A_k, B_k to minimize

$$\left\| T - \sum_{k=1}^{s} A_k \otimes B_k \right\|_F, \tag{1.1}$$

where T is an $n^2 \times n^2$ banded BTTB matrix, and A_k, B_k are $n \times n$ banded Toeplitz matrices. A general approach for constructing such an optimal approximation was proposed by Van Loan and Pitsianis [25] (see also Pitsianis [23]). Their approach, which we describe in more detail in section 2, requires computing principal singular values and vectors of an $n^2 \times n^2$ matrix related to T.

An alternative approach for computing a Kronecker product approximation $T \approx A \otimes B$ for certain deconvolution problems was proposed by Thirumalai [24]. A similar approach for banded BTTB matrices was considered by Nagy [22]. As opposed to the method of Van Loan and Pitsianis, the schemes described in [22, 24] require computing principal singular values and vectors of an array having dimension at most $n \times n$, and thus can be substantially less expensive. Moreover, Kamm and Nagy [20] show how these approximations can be used to efficiently construct approximate SVD preconditioners.

Numerical examples in [20, 22, 24] indicate that this more efficient approach can lead to preconditioners that perform better than BCCB approximations. However, theoretical results establishing optimality of the approximations, such as in (1.1), were not given. In this paper, we provide these results. In particular, we show that some modifications to the method proposed in [22, 24] are needed to obtain an approximation of the form (1.1). Our theoretical results lead to an efficient algorithm for computing Kronecker product approximations of banded BTTB matrices.

This paper is organized as follows. Some notation is defined, and a brief review of the method proposed by Van Loan and Pitsianis is provided in section 2. In section 3 we show how to exploit the banded BTTB structure to obtain an efficient scheme for computing terms in the Kronecker product decomposition. A numerical example from image restoration is given in section 4.

2. Preliminaries and notation. In this section we establish some notation to be used throughout the paper and describe some previous work on Kronecker product approximations. To simplify notation, we assume T is an $n \times n$ block matrix with $n \times n$ blocks.

2.1. Banded BTTB matrices. We assume that the matrix T is a BTTB, so it can be uniquely determined by a single column $\mathbf{t}$ which contains all of the nonzero values in T; that is, some central column. It will be useful to define an $n \times n$ array P as $\mathbf{t} = \text{vec}(P^T)$, where the **vec** operator transforms matrices into vectors by stacking columns as follows:

$$A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix} \quad \Leftrightarrow \quad \mathrm{vec}(A) = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}.$$

Suppose further that the entry of P corresponding to the diagonal of T is known.[1] For example, suppose that

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}, \tag{2.1}$$

where the diagonal of T is located at $(i, j) = (2, 3)$. Then $\mathbf{t} = \mathrm{vec}(P^T)$ is the sixth column of T, and we write

$$T = \mathrm{toep2}[\mathbf{t}, 2, 3] = \left[\begin{array}{ccc|ccc|ccc} p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} & 0 & 0 & 0 \\ 0 & p_{23} & p_{22} & 0 & p_{13} & p_{12} & 0 & 0 & 0 \\ 0 & 0 & p_{23} & 0 & 0 & p_{13} & 0 & 0 & 0 \\ \hline p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} \\ 0 & p_{33} & p_{32} & 0 & p_{23} & p_{22} & 0 & p_{13} & p_{12} \\ 0 & 0 & p_{33} & 0 & 0 & p_{23} & 0 & 0 & p_{13} \\ \hline 0 & 0 & 0 & p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} \\ 0 & 0 & 0 & 0 & p_{33} & p_{32} & 0 & p_{23} & p_{22} \\ 0 & 0 & 0 & 0 & 0 & p_{33} & 0 & 0 & p_{23} \end{array}\right]. \tag{2.2}$$

In general, if the diagonal of T is p_{ij}, then the upper and lower block bandwidths of T are $i-1$ and $n-i$, respectively. The upper and lower bandwidths of each Toeplitz block are $j-1$ and $n-j$, respectively.

In a similar manner, the notation $X = \mathrm{toep}(\mathbf{x}, i)$ is used to represent a banded point Toeplitz matrix X constructed from the vector $\mathbf{x}$, where x_i corresponds to the diagonal entry. For example, if the second component of the vector $\mathbf{x} = [\, x_1 \; x_2 \; x_3 \; x_4 \,]^T$ corresponds to the diagonal element of a banded Toeplitz matrix X, then

$$X = \mathrm{toep}(\mathbf{x}, 2) = \begin{bmatrix} x_2 & x_1 & 0 & 0 \\ x_3 & x_2 & x_1 & 0 \\ x_4 & x_3 & x_2 & x_1 \\ 0 & x_4 & x_3 & x_2 \end{bmatrix}.$$

2.2. Kronecker product approximations. In this subsection we review the work of Van Loan and Pitsianis. We require the following properties of Kronecker products:

- $(A \otimes B)^T = A^T \otimes B^T$.
- $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.
- If U_1 and U_2 are orthogonal matrices, then $U_1 \otimes U_2$ is also orthogonal.
- $(A \otimes B)\mathbf{x} = \mathrm{vec}(BXA^T)$, $\mathrm{vec}(X) = \mathbf{x}$.

A more complete discussion and additional properties of Kronecker products can be found in Horn and Johnson [16] and Graham [13].

[1] In image restoration, P is often referred to as a "point spread function," and the diagonal entry is the location of the "point source." See section 4 for more details.

Van Loan and Pitsianis [25] (see also Pitsianis [23]) propose a general technique for an approximation involving Kronecker products where $||T - \sum_k (A_k \otimes B_k)||_F$ is minimized. By defining the transformation to *tilde space* of a block matrix T,

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & & \vdots \\ T_{n1} & T_{n2} & \cdots & T_{nn} \end{bmatrix},$$

as

$$\tilde{T} = \text{tilde}(T) = \begin{bmatrix} \text{vec}(T_{11})^T \\ \vdots \\ \text{vec}(T_{n1})^T \\ \vdots \\ \text{vec}(T_{1n})^T \\ \vdots \\ \text{vec}(T_{nn})^T \end{bmatrix};$$

it is shown in [23, 25] that

$$\left\| T - \sum_{k=1}^{s} (A_k \otimes B_k) \right\|_F = \left\| \tilde{T} - \sum_{k=1}^{s} (\tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T) \right\|_F,$$

where $\tilde{\mathbf{a}}_k = \text{vec}(A_k)$ and $\tilde{\mathbf{b}}_k = \text{vec}(B_k)$. Thus, the Kronecker product approximation problem is reduced to a rank-s approximation problem. Given the SVD of $\tilde{T}$, $\tilde{T} = \sum_{k=1}^{r} \tilde{\sigma}_k \tilde{\mathbf{u}}_k \tilde{\mathbf{v}}_k^T$, $\text{rank}(\tilde{T}) = r$, it is well known [12] that the rank-s approximation $\tilde{T}_s$, $s \le r$, which minimizes $||\tilde{T} - \tilde{T}_s||_F$ is $\tilde{T}_s = \sum_{k=1}^{s} \tilde{\sigma}_k \tilde{\mathbf{u}}_k \tilde{\mathbf{v}}_k^T$. Choosing $\tilde{\mathbf{a}}_k = \sqrt{\tilde{\sigma}_k}\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{b}}_k = \sqrt{\tilde{\sigma}_k}\tilde{\mathbf{v}}_k$ minimizes $||\tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T||_F$ over all rank-s approximations, and thus one can construct an approximation $\hat{T} = \sum_{k=1}^{s} (A_k \otimes B_k)$ which minimizes $||T - \hat{T}||_F$.

This general technique requires computing the largest s singular triplets of an $n^2 \times n^2$ matrix, which may be expensive for large n. Thirumalai [24] and Nagy [22] show that a Kronecker product approximation of a banded BTTB matrix T can be found by computing the largest s singular triplets of the $n \times n$ array P. However, this method does not find the Kronecker product which minimizes the Frobenius norm approximation problem in (1.1). In the next section we show that if T is a banded BTTB matrix, then this optimal approximation can be computed from an SVD of a weighted version of the $n \times n$ array P.

3. BTTB optimal Kronecker product approximation. Recall that the Van Loan and Pitsianis approach minimizes $||T - \sum_{k=1}^{s} (A_k \otimes B_k)||_F$ for a general (unstructured) matrix T by minimizing $||\tilde{T} - \sum_{k=1}^{s} (\tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T)||_F$. If it is assumed that A_k and B_k are banded Toeplitz matrices, then the array P associated with the central column of T can be weighted and used to construct an approximation which minimizes $||\tilde{T} - \sum_{k=1}^{s} (\tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T)||_F$.

THEOREM 3.1. *Let T be the $n^2 \times n^2$ banded BTTB matrix constructed from P, where p_{ij} is the diagonal element of T. (Therefore, the upper and lower block bandwidths of T are $i-1$ and $n-i$, and the upper and lower bandwidths of each Toeplitz block are $j-1$ and $n-j$.) Further, let A_k be an $n \times n$ banded Toeplitz matrix*

with upper bandwidth $i-1$ and lower bandwidth $n-i$, and let B_k be an $n \times n$ banded Toeplitz matrix with upper bandwidth $j-1$ and lower bandwidth $n-j$. Define $\mathbf{a}_k$ and $\mathbf{b}_k$ such that $A_k = toep(\mathbf{a}_k, i)$ and $B_k = toep(\mathbf{b}_k, j)$, and define

$$\begin{aligned}
\tilde{T} &= tilde(T),\\
\tilde{\mathbf{a}}_k &= vec(A_k),\\
\tilde{\mathbf{b}}_k &= vec(B_k),\\
W_a &= \mathrm{diag}(\sqrt{n-i+1}, \sqrt{n-i+2}, \ldots, \sqrt{n-1}, \sqrt{n}, \sqrt{n-1}, \ldots, \sqrt{i+1}, \sqrt{i}),\\
W_b &= \mathrm{diag}(\sqrt{n-j+1}, \sqrt{n-j+2}, \ldots, \sqrt{n-1}, \sqrt{n}, \sqrt{n-1}, \ldots, \sqrt{j+1}, \sqrt{j}),\\
P_w &= W_a P W_b.
\end{aligned}$$

Then for $s \le r = \mathrm{rank}(P)$,

$$\left\| \tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T \right\|_F = \left\| P_w - \sum_{k=1}^{s} (W_a \mathbf{a}_k)(W_b \mathbf{b}_k)^T \right\|_F .$$

Proof. See section 3.1. ∎

Therefore, if A_k and B_k are constrained to be banded Toeplitz matrices, then $||T - \sum_{k=1}^{s}(A_k \otimes B_k)||_F$ can be minimized by finding $\mathbf{a}_k$, $\mathbf{b}_k$ which minimize $||P_w - \sum_{k=1}^{s}(W_a\mathbf{a}_k)(W_b\mathbf{b}_k)^T||_F$. This is a rank-$s$ approximation problem, involving a matrix of relatively small dimension, which can be constructed using the SVD of P_w. Noting that W_a and W_b are diagonal matrices which do not need to be formed explicitly, the construction of $\hat{T} = \sum_{k=1}^{s} A_k \otimes B_k$ which minimizes $||T - \hat{T}||_F$, where A_k and B_k are banded Toeplitz matrices, can be computed as follows:

- Define the weight vectors w_a and w_b based on the (i, j) location (in P) of the diagonal entry of T:

$$\begin{aligned}
\mathbf{w}_a &= \left[\begin{array}{ccccccc} \sqrt{n-i+1} & \cdots & \sqrt{n-1} & \sqrt{n} & \sqrt{n-1} & \cdots & \sqrt{i} \end{array} \right]^T,\\
\mathbf{w}_b &= \left[\begin{array}{ccccccc} \sqrt{n-j+1} & \cdots & \sqrt{n-1} & \sqrt{n} & \sqrt{n-1} & \cdots & \sqrt{j} \end{array} \right]^T.
\end{aligned}$$

- Calculate $P_w = (\mathbf{w}_a \mathbf{w}_b^T) .* P$ and its SVD $P_w = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T$, where ".*" denotes point-wise multiplication.
- Calculate

$$\begin{aligned}
\mathbf{a}_k &= (\sqrt{\sigma_k}\mathbf{u}_k)./\mathbf{w}_a,\\
A_k &= \mathrm{toep}(\mathbf{a}_k, i),\\
\mathbf{b}_k &= (\sqrt{\sigma_k}\mathbf{v}_k)./\mathbf{w}_b,\\
B_k &= \mathrm{toep}(\mathbf{b}_k, j)
\end{aligned}$$

  for $k = 1, \ldots, s$, $s \le r$, where "./" denotes point-wise division.

The proof of Theorem 3.1 is based on observing that $\tilde{T}$ has at most n unique rows and n unique columns, which consist precisely of the rows and columns of P. This observation will become clear in the following subsection.

3.1. Proof of Theorem 3.1. To prove Theorem 3.1, we first observe that if a matrix has one row which is a scalar multiple of another row, then a rotator can be constructed to zero out one of these rows, i.e.,

$$Q \left[\begin{array}{c} \alpha \mathbf{x}^T \\ \mathbf{x}^T \end{array} \right] = \left[\begin{array}{cc} \frac{\alpha}{\sqrt{\alpha^2+1}} & \frac{1}{\sqrt{\alpha^2+1}} \\ \frac{-1}{\sqrt{\alpha^2+1}} & \frac{\alpha}{\sqrt{\alpha^2+1}} \end{array} \right] \left[\begin{array}{c} \alpha \mathbf{x}^T \\ \mathbf{x}^T \end{array} \right] = \left[\begin{array}{c} \sqrt{\alpha^2+1}\mathbf{x}^T \\ \mathbf{0}^T \end{array} \right]. \tag{3.1}$$

If this is extended to the case where more than two rows are repeated, then a simple induction proof can be used to establish the following lemma.

LEMMA 3.2. *Suppose an $n \times n$ matrix X has k identical rows:*

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_{n-k+1}^T \end{bmatrix}.$$

Then a sequence of $k-1$ orthogonal plane rotators $Q_1, Q_2, \ldots, Q_{k-1}$ can be constructed such that

$$QX = Q_{k-1}Q_{k-2}\cdots Q_1 X = \begin{bmatrix} \sqrt{k}\mathbf{x}_1^T \\ \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_{n-k+1}^T \end{bmatrix},$$

thereby zeroing out all the duplicate rows.

It is easily seen that this result can be applied to the columns of a matrix as well, using the transpose of the plane rotators defined in Lemma 3.2.

LEMMA 3.3. *Suppose an $n \times n$ matrix X contains k identical columns:*

$$X = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_1 & \cdots & \mathbf{x}_1 & \mathbf{x}_2 & \cdots \mathbf{x}_{n-k+1} \end{bmatrix}.$$

Then an orthogonal matrix Q can be constructed from a series of plane rotators such that

$$XQ^T = \begin{bmatrix} \sqrt{k}\mathbf{x}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{x}_2 & \cdots & \mathbf{x}_{n-k+1} \end{bmatrix}.$$

The above results illustrate the case where the first occurrence of a row (column) is modified to zero out the remaining occurrences. However, this is for notational convenience only. By appropriately constructing the plane rotators, any one of the duplicate rows (columns) may be selected for modification, and the remaining rows (columns) zeroed out. These rotators can now be applied to the matrix $\tilde{T}$.

LEMMA 3.4. *Let T be the $n^2 \times n^2$ banded BTTB matrix constructed from P, where p_{ij} is the diagonal entry of T. In other words, $T = toep2[vec(P^T), i, j]$. Further, define*

$$\begin{aligned} \tilde{T} &= tilde(T), \\ W_a &= \operatorname{diag}(\sqrt{n-i+1}, \sqrt{n-i+2}, \ldots, \sqrt{n-1}, \sqrt{n}, \sqrt{n-1}, \ldots, \sqrt{i+1}, \sqrt{i}), \\ W_b &= \operatorname{diag}(\sqrt{n-j+1}, \sqrt{n-j+2}, \ldots, \sqrt{n-1}, \sqrt{n}, \sqrt{n-1}, \ldots, \sqrt{j+1}, \sqrt{j}). \end{aligned}$$

Then orthogonal matrices Q_1 and Q_2 can be constructed such that

$$Q_1 \tilde{T} Q_2^T = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & W_a P W_b & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Proof. By definition,

$$T = \begin{bmatrix} T_i & & T_1 & & 0 \\ & \ddots & & \ddots & \\ T_n & & T_i & & T_1 \\ & \ddots & & \ddots & \\ 0 & & T_n & & T_i \end{bmatrix}.$$

Defining $\tilde{\mathbf{t}}_i^T = \mathrm{vec}(T_i)^T$, and representing $\tilde{T}$ using the $n \times n^2$ submatrices $\tilde{T}_i$,

$$\tilde{T} = \begin{bmatrix} \tilde{T}_1 \\ \vdots \\ \tilde{T}_i \\ \vdots \\ \tilde{T}_n \end{bmatrix},$$

it is clear that $\tilde{T}$ contains only n unique rows, which are $\tilde{\mathbf{t}}_1^T, \ldots, \tilde{\mathbf{t}}_n^T$, and that the ith submatrix, $\tilde{T}_i$, contains all the unique rows, i.e.,

$$\tilde{T}_i = \begin{bmatrix} \tilde{\mathbf{t}}_1^T \\ \tilde{\mathbf{t}}_2^T \\ \vdots \\ \tilde{\mathbf{t}}_n^T \end{bmatrix}.$$

Furthermore, it can be seen that there are $n - i + 1$ occurrences of $\tilde{\mathbf{t}}_1^T$, $\ldots$, $n - 1$ occurrences of $\tilde{\mathbf{t}}_{i-1}^T$, n occurrences of $\tilde{\mathbf{t}}_i^T$, $n - 1$ occurrences of $\tilde{\mathbf{t}}_{i+1}^T$, $\ldots$, and i occurrences of $\tilde{\mathbf{t}}_n^T$. Therefore, a sequence of orthogonal plane rotators can be constructed

to zero out all rows of $\tilde{T}$ except those in the submatrix $\tilde{T}_i$, i.e.,

$$Q_1\tilde{T} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ W_a\tilde{T}_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \\ \sqrt{n-i+1}\,\tilde{\mathbf{t}}_1^T \\ \vdots \\ \sqrt{n-1}\,\tilde{\mathbf{t}}_{i-1}^T \\ \sqrt{n}\,\tilde{\mathbf{t}}_i^T \\ \sqrt{n-1}\,\tilde{\mathbf{t}}_{i+1}^T \\ \vdots \\ \sqrt{i}\,\tilde{\mathbf{t}}_n^T \\ \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix}.$$

Now, partitioning $\tilde{T}_i$,

$$\tilde{T}_i = \begin{bmatrix} \tilde{T}_{i1} & \cdots & \tilde{T}_{ij} & \cdots & \tilde{T}_{in} \end{bmatrix},$$

where each $\tilde{T}_{ij}$ is an $n \times n$ submatrix, it can be seen that $\tilde{T}_i$ contains only n unique columns, which are the columns of P, $\mathbf{p}_1, \ldots, \mathbf{p}_n$, and that the jth submatrix $\tilde{T}_{ij}$ contains all the unique columns, i.e.,

$$\tilde{T}_{ij} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix} = P.$$

Furthermore, the matrix $\tilde{T}_i$ contains $n-j+1$ occurrences of $\mathbf{p}_1$, $\ldots$, $n-1$ occurrences of $\mathbf{p}_{j-1}$, n occurrences of $\mathbf{p}_j$, $n-1$ occurrences of $\mathbf{p}_{j+1}$, $\ldots$, and j occurrences of $\mathbf{p}_n$. Therefore, a sequence of orthogonal plane rotators can be constructed such that

$$Q_1\tilde{T}Q_2^T = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & W_aPW_b & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \qquad \square$$

The following properties involving the **vec** and **toep2** operators are needed.

LEMMA 3.5. *Let T, $\tilde{T}$, and P be defined as in Lemma* 3.4. *Further, let A_k be an $n \times n$ banded Toeplitz matrix with upper bandwidth $i-1$ and lower bandwidth $n-i$, and let B_k be an $n \times n$ banded Toeplitz matrix with upper bandwidth $j-1$ and lower bandwidth $n-j$. Define $\mathbf{a}_k$ and $\mathbf{b}_k$ such that $A_k = toep(\mathbf{a}_k, i)$ and $B_k = toep(\mathbf{b}_k, j)$. Then*

(1) *$vec(X) - vec(Y) = vec(X - Y)$, where X and Y are any two matrices of the same size,*
(2) *$toep2(\mathbf{x}, i, j) - toep2(\mathbf{y}, i, j) = toep2(\mathbf{x} - \mathbf{y}, i, j)$, where $\mathbf{x}$ and $\mathbf{y}$ are any two vectors of the same length,*

(3) $toep2\{vec[(\sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T)^T], i, j\} = \sum_{k=1}^{s} A_k \otimes B_k$, *and*
(4) $toep2\{vec[(P - \sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T)^T], i, j\} = T - \sum_{k=1}^{s} A_k \otimes B_k$.

Proof. Properties 1 and 2 are clear from the definitions of the vec and toep2 operators. Property 3 can be seen by considering the banded Toeplitz matrices $A = \text{toep}(\mathbf{a}, i)$ and $B = \text{toep}(\mathbf{b}, j)$ and noting that the central column of $A \otimes B$ containing all the nonzero entries is

$$\text{vec}[(\mathbf{a}\mathbf{b}^T)^T] = \begin{bmatrix} a_1 b_1 \\ \vdots \\ a_1 b_n \\ \vdots \\ a_n b_1 \\ \vdots \\ a_n b_n \end{bmatrix}.$$

Therefore, property 3 holds when $k = 1$ since both sides are banded BTTB matrices constructed from the same central column, and can be extended to $k = 1, \ldots, s$ by applying property 2. Property 4 follows from properties 2 and 3. ∎

Using these properties, Lemma 3.4 can be extended to the matrix $\tilde{T} - \sum_k \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T$.

LEMMA 3.6. *Let T be the $n^2 \times n^2$ banded BTTB matrix constructed from P, where p_{ij} is the diagonal entry of T. Further, let A_k be an $n \times n$ banded Toeplitz matrix with upper bandwidth $i - 1$ and lower bandwidth $n - i$, and let B_k be an $n \times n$ banded Toeplitz matrix with upper bandwidth $j - 1$ and $n - j$. Define $\mathbf{a}_k$ and $\mathbf{b}_k$ such that $A_k = toep(\mathbf{a}_k, i)$ and $B_k = toep(\mathbf{b}_k, j)$, and define $\tilde{\mathbf{a}}_k = vec(A_k)$ and $\tilde{\mathbf{b}}_k = vec(B_k)$. Let $\tilde{T}$, W_a, and W_b be defined as in Lemma 3.4. Then orthogonal matrices Q_1 and Q_2 can be constructed such that*

$$Q_1 \left(\tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T \right) Q_2^T = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & W_a(P - \sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T) W_b & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Proof. Using Lemma 3.5,

$$T - \sum_{k=1}^{s} A_k \otimes B_k = \text{toep2} \left\{ \text{vec} \left[\left(P - \sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T \right)^T \right], i, j \right\}.$$

By definition of the transformation to tilde space,

$$\text{tilde} \left(T - \sum_{k=1}^{s} A_k \otimes B_k \right) = \tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T.$$

Applying Lemma 3.4 to $T - \sum_{k=1}^{s} A_k \otimes B_k$ yields

$$Q_1 \left(\tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T \right) Q_2^T = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & W_a(P - \sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T) W_b & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \qquad \square$$

The proof of Theorem 3.1 follows directly from Lemma 3.6 by noting that

$$\begin{aligned} \left\| \tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T \right\|_F &= \left\| Q_1 \left(\tilde{T} - \sum_{k=1}^{s} \tilde{\mathbf{a}}_k \tilde{\mathbf{b}}_k^T \right) Q_2^T \right\|_F \\ &= \left\| W_a \left(P - \sum_{k=1}^{s} \mathbf{a}_k \mathbf{b}_k^T \right) W_b \right\|_F \\ &= \left\| P_w - \sum_{k=1}^{s} (W_a \mathbf{a}_k)(W_b \mathbf{b}_k)^T \right\|_F. \end{aligned}$$

3.2. Further analysis. It has been shown how to minimize $||T - \hat{T}||_F$ when the structure of $\hat{T}$ is constrained to be a sum of Kronecker products of banded Toeplitz matrices. We now show that if T is a banded BTTB matrix, then the matrix $\hat{T} = \sum_i A_i \otimes B_i$ minimizing $||T - \hat{T}||_F$ must adhere to this structure. Therefore, the approximation minimizes $||T - \hat{T}||_F$ over all matrices $\hat{T} = \sum_i A_i \otimes B_i$ when T is a banded BTTB matrix.

If T is a banded BTTB matrix, then the rows and columns of $\tilde{T}$ have a particular structure. To represent this structure, using an approach similar to Van Loan and Pitsianis [25], we define the constraint matrix $S_{n,\omega}$. Given an $n \times n$ banded Toeplitz matrix T, with upper and lower bandwidths $\omega = \begin{bmatrix} \omega_u, & \omega_l \end{bmatrix}$, $S_{n,\omega}$ is an $n^2 \times (n^2 - (\omega_u + \omega_l + 1))$ $\{-1, 0, 1\}$ matrix such that $S_{n,\omega}^T \text{vec}(T) = 0$. For example, let T be a 4×4 banded Toeplitz matrix with bandwidths $\omega_u = 2$ and $\omega_l = 1$. Then

$$T = \begin{bmatrix} t_2 & t_1 & t_0 & 0 \\ t_3 & t_2 & t_1 & t_0 \\ 0 & t_3 & t_2 & t_1 \\ 0 & 0 & t_3 & t_2 \end{bmatrix},$$

and

$$S^T_{4,[2,1]} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1\\ \hline 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \end{array}\right].$$

Note that $S^T_{n,\omega}$ clearly has full row rank. Given the matrix T in (2.2),

$$\tilde{T} = \left[\begin{array}{ccc|ccc|ccc} p_{22}&p_{23}&0&p_{21}&p_{22}&p_{23}&0&p_{21}&p_{22}\\ p_{32}&p_{33}&0&p_{31}&p_{32}&p_{33}&0&p_{31}&p_{32}\\ 0&0&0&0&0&0&0&0&0\\ \hline p_{12}&p_{13}&0&p_{11}&p_{12}&p_{13}&0&p_{11}&p_{12}\\ p_{22}&p_{23}&0&p_{21}&p_{22}&p_{23}&0&p_{21}&p_{22}\\ p_{32}&p_{33}&0&p_{31}&p_{32}&p_{33}&0&p_{31}&p_{32}\\ \hline 0&0&0&0&0&0&0&0&0\\ p_{12}&p_{13}&0&p_{11}&p_{12}&p_{13}&0&p_{11}&p_{12}\\ p_{22}&p_{23}&0&p_{21}&p_{22}&p_{23}&0&p_{21}&p_{22} \end{array}\right],$$

$$S^T_{3,[1,1]} = \left[\begin{array}{ccc|ccc|ccc} 1&0&0&0&-1&0&0&0&0\\ 0&1&0&0&0&-1&0&0&0\\ 0&0&1&0&0&0&0&0&0\\ \hline 0&0&0&1&0&0&0&-1&0\\ 0&0&0&0&1&0&0&0&-1\\ \hline 0&0&0&0&0&0&1&0&0 \end{array}\right],$$

and the rows and columns of $\tilde{T}$ satisfy

$$S^T_{3,[1,1]}\tilde{T}(:,i) = 0,$$
$$S^T_{3,[1,1]}\tilde{T}(i,:)^T = 0$$

for $i = 1, \ldots, n^2$. Using the structure of $\tilde{T}$, the matrix $\hat{T} = \sum_{i=1}^{k} A_i \otimes B_i$ minimizing $||T - \hat{T}||_F$ must be structured such that A_i and B_i are banded Toeplitz matrices, as the following sequence of results illustrates.

LEMMA 3.7. *Let* $A = \left[\begin{array}{cccc} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right]$ *be the* $n \times n$ *matrix whose structure is constrained by* $S^T_{n,\omega}\mathbf{a}_i = 0$, $\mathbf{a}_i \neq 0$, *for* $i = 1, \ldots, n$. *Further, let* $A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ *be the SVD of* A, *where* $r = \text{rank}(A)$. *Then* $\mathbf{u}_i$ *satisfies* $S^T_{n,\omega}\mathbf{u}_i = 0$ *for* $i = 1, \ldots, r$.

Proof. Given the SVD of A

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

for $i = 1, \ldots, n$, and subsequently

$$S^T_{n,\omega} A\mathbf{v}_i = \sigma_i S^T_{n,\omega}\mathbf{u}_i.$$

By definition, $S_{n,\omega}^T A = 0$ and $\sigma_i > 0$ for $i = 1, \ldots, r$. Therefore, $S_{n,\omega}^T \mathbf{u}_i = 0$ for $i = 1, \ldots, r$. ∎

Applying this result to A^T, it is clear that the right singular vectors of A satisfy $S_{n,\omega}^T \mathbf{v}_i = 0$ for $i = 1, \ldots, r$ if the rows of A are structured in the same manner.

LEMMA 3.8. *Let*

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix}$$

be the $n \times n$ matrix whose structure is constrained by $S_{n,\omega}^T \mathbf{a}_i = 0$ for $i = 1, \ldots, n$. Further, let $A = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ be the SVD of A, where $r = \text{rank}(A)$. Then $\mathbf{v}_i$ satisfies $S_{n,\omega}^T \mathbf{v}_i = 0$ for $i = 1, \ldots, r$.

THEOREM 3.9. *Let T be an $n \times n$ banded block Toeplitz matrix with $n \times n$ banded Toeplitz blocks, where the upper and lower block bandwidths of T are $\omega = \begin{bmatrix} \omega_u & \omega_l \end{bmatrix}$, and the upper and lower bandwidths of each Toeplitz block are $\gamma = \begin{bmatrix} \gamma_u & \gamma_l \end{bmatrix}$. Then the matrices A_i and B_i minimizing*

$$\left\| T - \sum_{i=1}^{k} (A_i \otimes B_i) \right\|_F$$

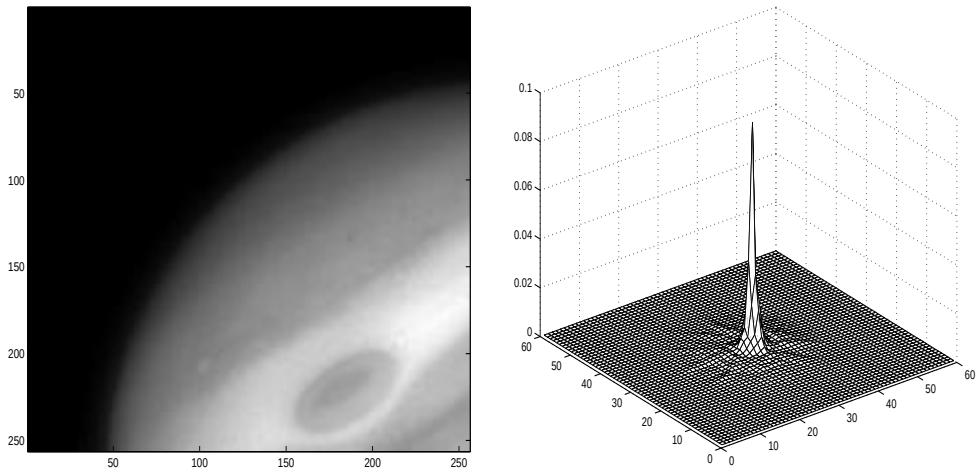
for $k \leq n$ are $n \times n$ banded Toeplitz matrices, where the upper and lower bandwidths of A_i are given by ω, and the upper and lower bandwidths of B_i are given by γ.

Proof. Recall that

$$\left\| T - \sum_{i=1}^{k} (A_i \otimes B_i) \right\|_F = \left\| \tilde{T} - \sum_{i=1}^{k} (\tilde{\mathbf{a}}_i \tilde{\mathbf{b}}_i^T) \right\|_F,$$

where $\text{vec}(A_i) = \tilde{\mathbf{a}}_i$ and $\text{vec}(B_i) = \tilde{\mathbf{b}}_i$. The structure of T results in $\text{rank}(\tilde{T}) = r \leq n$ and $S_{n,\omega}^T \tilde{T}(:, i) = S_{n,\gamma}^T \tilde{T}(i, :)^T = 0$ for $i = 1, \ldots, n^2$. Letting $\tilde{T} = \sum_{i=1}^r \tilde{\sigma}_i \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^T$ be the SVD of $\tilde{T}$, $||\tilde{T} - \sum_{i=1}^k (\tilde{\mathbf{a}}_i \tilde{\mathbf{b}}_i^T)||_F$, $k \leq r$, is minimized by $\tilde{\mathbf{a}}_i = \sqrt{\tilde{\sigma}_i} \tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{b}}_i = \sqrt{\tilde{\sigma}_i} \tilde{\mathbf{v}}_i$, where $S_{n,\omega}^T \tilde{\mathbf{u}}_i = S_{n,\gamma}^T \tilde{\mathbf{v}}_i = 0$. Therefore, A_i is an $n \times n$ banded Toeplitz matrix with upper and lower bandwidths given by ω, and B_i is an $n \times n$ banded Toeplitz matrix with upper and lower bandwidths given by γ. ∎

3.3. Remarks on optimality. The approach outlined in this section results in an optimal Frobenius norm Kronecker product approximation to a banded BTTB matrix. The approximation is obtained from the principal singular components of an array $P_w = W_a P W_b$. It might be interesting to consider whether it is possible to compute approximations which are optimal in another norm. In particular, the method considered in [20, 22, 24] uses a Kronecker product approximation computed from the principal singular components of P. Unfortunately we are unable to show that this leads to an optimal norm approximation. However, there is a very close relationship between the approaches. Since W_a and W_b are full rank, well-conditioned diagonal matrices, P and P_w have the same rank. Although it is possible to establish bounds on the singular values of products of matrices (see, for example, Horn and Johnson [15]), we have not been able to determine a precise relationship between the Kronecker product approximations obtained from the two methods. However,



(a) Observed, blurred image. (b) PSF, P.

FIG. 1. *Observed HST image and point spread function.*

we have found through extensive numerical results that both methods give similarly good approximations. Since numerical comparisons do not provide any additional insight into the quality of the approximation, we omit such results. Instead, in the next section we provide an example from an application that motivated this work and illustrate how a Kronecker product approximation might be used in practice. We note that further comparisons with BCCB approximations can be found in [20, 24].

4. An image restoration example. In this section we consider an image restoration example, and show how the Kronecker product approximations can be used to construct an approximate SVD preconditioner. Image restoration is often modeled as a linear system:

$$\mathbf{b} = T\mathbf{x} + \mathbf{n}\,,$$

where $\mathbf{b}$ is an observed blurred, noisy image, T is a large, often ill-conditioned matrix representing the blurring phenomena, $\mathbf{n}$ is noise, and $\mathbf{x}$ is the desired true image. If the blur is assumed to be *spatially invariant*, then T is a banded BTTB matrix [1, 21]. In this case, the array P corresponding to a central column of T is called a *point spread function* (PSF).

The test data we use consists of a partial image of Jupiter taken from the Hubble Space Telescope (HST) in 1992, before the mirrors in the Wide Field Planetary Camera were fixed. The data was obtained via anonymous ftp from ftp.stsci.edu in the directory pub/stsdas/testdata/restore/data/jupiter. Figure 1 shows the observed image. Also shown in Figure 1 is a mesh plot of the PSF, P, where the peak corresponds to the diagonal entry of T. The observed image is 256×256, so T is $65{,}536 \times 65{,}536$.

We mention that if T is ill conditioned, which is often the case in image restoration, then regularization is needed to suppress noise amplification in the computed solution [21]. Although T is essentially too large to compute its condition number, certain properties of the data indicate that T is fairly well conditioned. For instance, we observe that the PSF is not very smooth (smoother PSFs typically indicate more ill-conditioned T). Another indication comes from the fact that the optimal circulant

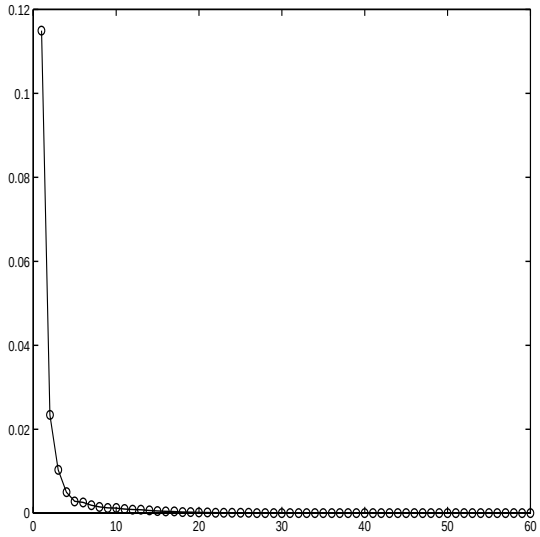


FIG. 2. *Singular values of the PSF, P.*

approximation of T, as well as our approximate SVD of T (to be described below), is well conditioned; specifically these approximations have condition numbers that are approximately 20.

We also mention that if the PSF can be expressed as $P = \sigma \mathbf{u}\mathbf{v}^T$ (i.e., it has rank 1), then the matrix T is separable. Using Theorem 3.1, $T = A \otimes B$, where $A = \text{toep}(\sqrt{\sigma}\mathbf{u})$ and $B = \text{toep}(\sqrt{\sigma}\mathbf{v})$. Efficient numerical methods that exploit the Kronecker product structure of T (e.g., [2, 5, 11]) can then be used.

However, as can be seen from the plot of the singular values of P in Figure 2, for this data, P is not rank 1, and so T is not separable. We therefore suggest constructing an approximate SVD to use as a preconditioner and solve the least squares problem $T\mathbf{x} \approx \mathbf{b}$ using a CG algorithm, such as CGLS; see Björck [3]. This preconditioning idea was proposed in [20] and can be described as follows. Given

$$T \approx \sum_{k=1}^{s} A_k \otimes B_k, \tag{4.1}$$

an SVD approximation of T can be constructed as

$$\begin{aligned} T &\approx U\Sigma V^T, \\ U &= U_A \otimes U_B, \\ V &= V_A \otimes V_B, \\ \Sigma &= \text{diag}(U^T T V) \\ &= \text{diag}(U^T(A_1 \otimes B_1 + A_2 \otimes B_2 + \cdots + A_k \otimes B_k)V), \end{aligned}$$

where $A_1 = U_A \Sigma_A V_A^T$ and $B_1 = U_B \Sigma_B V_B^T$. Note that the number of terms s only affects the setup cost of calculating Σ. For $s \geq 1$, $\Sigma = \text{diag}(U^T T V)$ clearly solves the minimization problem

TABLE 4.1
Number of CGLS and preconditioned CGLS (PCGLS) iterations needed for convergence.

CGLS, no prec.	PCGLS, circulant prec.	PCGLS, SVD prec.
43	12	4

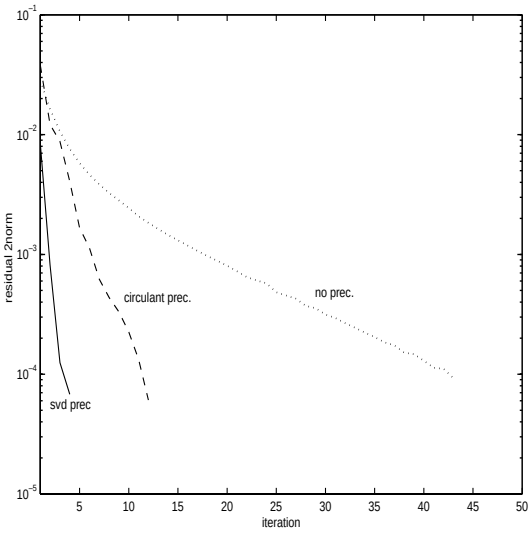


FIG. 3. *Plot of the residuals at each iteration.*

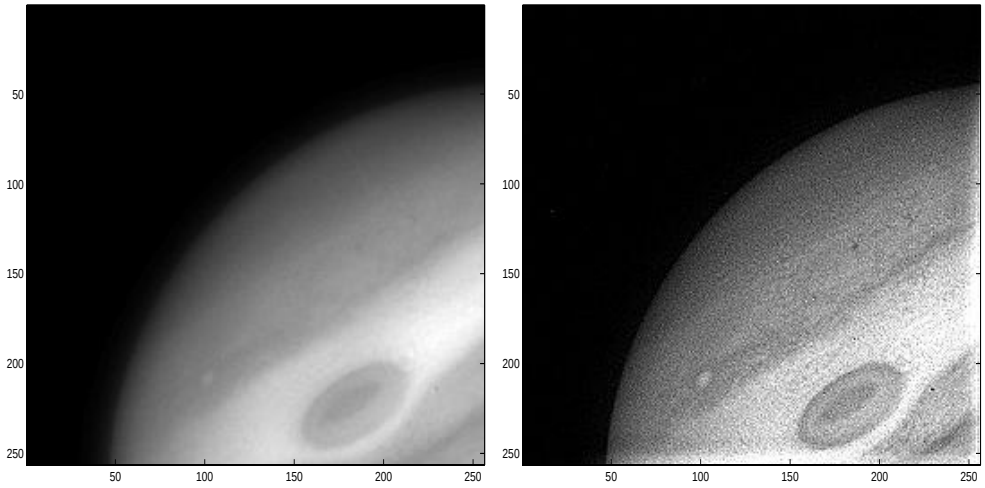
$$\min_{\Sigma} ||\Sigma - U^T T V||_F = \min_{\Sigma} ||U\Sigma V^T - T||_F$$

over all diagonal matrices Σ and therefore produces an optimal SVD approximation, given a fixed $U = U_A \otimes U_B$ and $V = V_A \otimes V_B$. This is analogous to the circulant and BCCB approximations discussed earlier, which provide an optimal eigendecomposition given a fixed set of eigenvectors (i.e., the Fourier vectors).

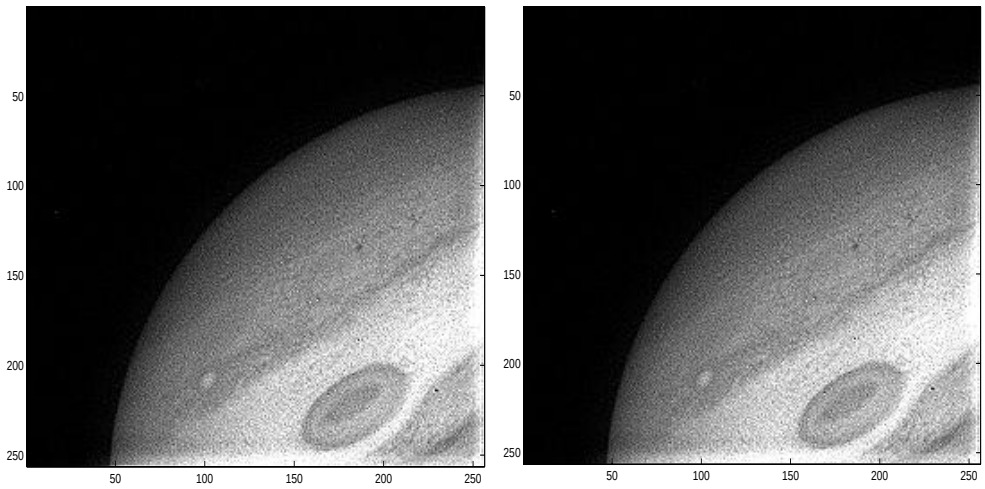
In our tests, we use CGLS to solve the least squares problem $T\mathbf{x} \approx \mathbf{b}$ using no preconditioner, our approximate SVD preconditioner (with $s = 3$ terms in equation (4.1)), and the optimal circulant preconditioner. Although we observed that T is fairly well conditioned, we should still be cautious about noise corrupting the computed restorations. Therefore, we use the conservative stopping tolerance $||T^T\mathbf{b} - T^T T\mathbf{x}||_2/||T^T\mathbf{b}||_2 < 10^{-4}$.

Table 4.1 shows the number of iterations needed for convergence in each case, and in Figure 3 we plot the corresponding residuals at each iteration. The computed solutions are shown in Figure 4, along with the HST observed, blurred image for comparison.

5. Concluding remarks. Because the image and PSF used in the previous section come from actual HST data, we cannot get an analytical measure on the accuracy of the computed solutions. However, we observe from Figure 4 that all solutions appear to be equally good restorations of the image, and from Figure 3 we see that the approximate SVD preconditioner is effective at reducing the number of iterations needed to obtain the solutions. Additional numerical examples comparing the accuracy of computed solutions, as well as computational cost of BCCB and



(a) HST blurred image. (b) CGLS solution, 43 iterations.



(c) PCGLS solution, circ. prec., 12 its. (d) PCGLS solution, SVD prec., 4 its.

FIG. 4. *The observed image, along with computed solutions from CGLS and PCGLS.*

the approximation SVD preconditioner, can be found in [19, 20]. A comparison of computational complexity between BCCB preconditioners and the approximate SVD preconditioner depends on many factors. For example,

- What is the dimension of P (i.e., the bandwidths of T)?
- Is a Lanczos scheme used to compute SVDs of P, A_1, and B_1?
- Do we take advantage of band and Toeplitz structure when forming matrix-matrix products involving U_A, U_B, V_A, V_B, and A_k, B_k, $k = 2, \ldots, s$?
- How many terms, s, do we take in the Kronecker product approximation?
- For BCCB preconditioners, is n a power of 2?

If we assume T is $n^2 \times n^2$, and $s = O(1)$, then set up and application of the approximate SVD preconditioner is *at most* $O(n^3)$. If we further assume that n is a power of 2, then the corresponding cost for BCCB preconditioners is *at least* $O(n^2 \log_2 n)$. It should be noted that the approximate SVD preconditioner does not require complex arithmetic, does not require n to be a power of 2, or does not require any zero padding. Moreover, decomposing T into a sum of Kronecker products, whose terms are banded Toeplitz matrices, might lead to other fast algorithms (as has occurred over many years of studying displacement structure [18]). In this case, the work presented in this paper provides an algorithm for efficiently computing an optimal Kronecker product approximation.

REFERENCES

[1] H. Andrews and B. Hunt, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[2] E. S. Angel and A. K. Jain, *Restoration of images degraded by spatially varying pointspread functions by a conjugate gradient method*, Applied Optics, 17 (1978), pp. 2186–2190.

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

[4] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 349–364.

[5] D. Calvetti and L. Reichel, *Application of ADI iterative methods to the image restoration of noisy images*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 165–186.

[6] S. Serra-Capizzano and E. Tyrtyshnikov, *Any circulant-like preconditioner for multilevel matrices is not superlinear*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 431–439.

[7] R. H. Chan and M. K. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[8] T. F. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.

[9] T. F. Chan and J. A. Olkin, *Preconditioners for Toeplitz-block matrices*, Numer. Algorithms, 6 (1993), pp. 89–101.

[10] P. J. Davis, *Circulant Matrices*, John Wiley, New York, 1979.

[11] L. Eldén and I. Skoglund, *Algorithms for the Regularization of Ill-Conditioned Least Squares Problems with Tensor Product Structure, and Application to Space-Variant Image Restoration*, Tech. Report LiTH-MAT-R-82-48, Department of Mathematics, Linköping University, Sweden, 1982.

[12] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[13] A. Graham, *Kronecker Products and Matrix Calculus: With Applications*, Halsted Press, John Wiley, New York, 1981.

[14] M. Hanke and J. G. Nagy, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques*, Inverse Problems, 12 (1996), pp. 157–173.

[15] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[16] R. A. Horn and C. A. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[17] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[18] T. Kailath and A. H. Sayed, *Displacement structure: Theory and applications*, SIAM Rev., 37 (1995), pp. 297–386.

[19] J. Kamm, *Singular Value Decomposition-Based Methods for Signal and Image Restoration*, Ph.D. thesis, Southern Methodist University, Dallas, TX, 1998.

[20] J. Kamm and J. G. Nagy, *Kronecker product and SVD approximations in image restoration*, Linear Algebra Appl., 284 (1998), pp. 177–192.

[21] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1991.

[22] J. G. Nagy, *Decomposition of Block Toeplitz Matrices into a Sum of Kronecker Products with Applications in Image Restoration*, Tech. Report 96-1, Department of Mathematics, Southern Methodist University, Dallas, TX, 1996.

[23] N. P. Pitsianis, *The Kronecker Product in Approximation and Fast Transform Generation*, Ph.D. thesis, Cornell University, Ithaca, NY, 1997.
[24] S. Thirumalai, *High Performance Algorithms to Solve Toeplitz and Block Toeplitz Matrices*, Ph.D. thesis, University of Illinois, Urbana, IL, 1996.
[25] C. F. Van Loan and N. P. Pitsianis, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real Time Applications, M. S. Moonen and G. H. Golub, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993, pp. 293–314.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 173–194

ON PRINCIPAL ANGLES BETWEEN SUBSPACES OF EUCLIDEAN SPACE*

ZLATKO DRMAČ†

Abstract. The cosines of the principal angles between the column spaces of full column rank matrices $X \in \mathbf{R}^{m\times p}$ and $Y \in \mathbf{R}^{m\times q}$ are efficiently computed, using the Björck–Golub algorithm, as the singular values of $Q_x^T Q_y$, where Q_x and Q_y are orthonormal matrices computed by the QR factorizations of X and Y, respectively. This paper shows that the Björck–Golub algorithm is mixed stable in the following sense: the computed singular values approximate with small relative error the exact cosines of the principal angles between the column spaces of $X+\Delta X$ and $Y+\Delta Y$, where ΔX, ΔY are small backward errors. Further, theoretical analysis and numerical evidence show that the algorithm becomes more robust if the QR factorizations are computed with the complete pivoting scheme of Powell and Reid. Moreover, it is shown that Gaussian elimination with complete pivoting can be used as an efficient preconditioner in computation and as a useful tool in analysis of the sensitivity of the QR factorization.

Key words. canonical correlations, principal angles, singular values

AMS subject classifications. 65F15, 65G05, 65F25

PII. S0895479897320824

1. Introduction.

Let $X \in \mathbf{R}^{m\times p}$, $Y \in \mathbf{R}^{m\times q}$ be full column rank matrices with $p \geq q$ and let $\mathcal{X} = \mathrm{span}(X)$, $\mathcal{Y} = \mathrm{span}(Y)$ be the corresponding column spaces. The minimal angle $\vartheta_1 \in [0, \pi/2]$ between $\mathcal{X}$ and $\mathcal{Y}$ is defined by

$$\cos\vartheta_1 = \max_{\substack{x\in\mathcal{X},\\ \|x\|_2=\|y\|_2=1}} \max_{y\in\mathcal{Y}} x^T y, \qquad \text{where } \|x\|_2 = \sqrt{x^T x}.$$

If $\mathcal{P}_\mathcal{X}$ and $\mathcal{P}_\mathcal{Y}$ are the orthogonal projectors onto $\mathcal{X}$, $\mathcal{Y}$, respectively, then $\sigma_1 = \cos\vartheta_1$ is the largest singular value of $P_\mathcal{X} P_\mathcal{Y}$. If $\sigma_1 \geq \cdots \geq \sigma_q$ are the singular values of $P_\mathcal{X} P_\mathcal{Y}$, then the principal angles $\vartheta_i \in [0, \pi/2]$, $1 \leq i \leq q$, between $\mathcal{X}$ and $\mathcal{Y}$ are defined by $\sigma_i = \cos\vartheta_i$. Note that $\sigma_1 = \|P_\mathcal{X} P_\mathcal{Y}\|_2$, where $\|\cdot\|_2$ denotes the operator norm induced by the Euclidean vector norm $\|\cdot\|_2$. If $p = q$, then the angle (distance) $\angle(\mathcal{X}, \mathcal{Y})$ between $\mathcal{X}$ and $\mathcal{Y}$ is defined by $\sin\angle(\mathcal{X}, \mathcal{Y}) = \|P_\mathcal{X} - P_\mathcal{Y}\|_2 = \sin\vartheta_p$.

The cosines of the principal angles are also known as canonical correlations and have important applications, e.g., in statistics, econometrics, and geology. Golub and Zha [25] discuss various equivalent characterizations and applications of the principal angles. For instance, the principal angles can be used to solve constrained optimization problems such as $\max_{A,B} \mathrm{Trace}(A^T X^T Y B)$, where $A^T X^T X A = I_p$, $B^T Y^T Y B = I_q$, or to solve the orthogonal Procrustes minimization problem $\min_{U^T U = I} \|Q_x - Q_y U\|_F$ with the Frobenius matrix norm $\|\cdot\|_F$ and orthonormal matrices Q_x and Q_y.

Björck and Golub [10] have shown that the principal angles can be efficiently computed via the singular value decomposition (SVD) of $Q_x^T Q_y$, where $X = Q_x R_x$ and

*Received by the editors April 30, 1997; accepted for publication (in revised form) by F. T. Luk December 18, 1998; published electronically May 31, 2000. This research was supported by National Science Foundation grants ACS–9357812 and ASC–9625912, Department of Energy grant DE–FG03–94ER25215, and Croatian Ministry of Science and Technology grant 037012. A preliminary version of this paper appeared as technical report CU–CS–838–97, Department of Computer Science, University of Colorado at Boulder.

http://www.siam.org/journals/simax/22-1/32082.html

†Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (drmac@math.hr).

$Y = Q_y R_y$ are the QR factorizations of X and Y, respectively. Taking $\mathcal{P}_{\mathcal{X}} \equiv Q_x Q_x^T$, $\mathcal{P}_{\mathcal{Y}} \equiv Q_y Q_y^T$ and writing the SVD of $Q_x^T Q_y$ as $Q_x^T Q_y = W \Sigma V^T$ we obtain the SVD $\mathcal{P}_{\mathcal{X}} \mathcal{P}_{\mathcal{Y}} = (Q_x W) \Sigma (Q_y V)^T$.

In this paper, we analyze the numerical stability of the Björck–Golub algorithm. In section 2, we show that the algorithm is mixed stable: the computed approximations of the singular values of $\mathcal{P}_{\mathcal{X}} \mathcal{P}_{\mathcal{Y}}$ approximate with small relative error the singular values of $\mathcal{P}_{\tilde{\mathcal{X}}} \mathcal{P}_{\tilde{\mathcal{Y}}}$, where $\tilde{\mathcal{X}} = \mathrm{span}(X + \Delta X)$, $\tilde{\mathcal{Y}} = \mathrm{span}(Y + \Delta Y)$ and $\max_{1 \le i \le p} \|\Delta X e_i\|_2 / \|X e_i\|_2$, $\max_{1 \le i \le q} \|\Delta Y e_i\|_2 / \|Y e_i\|_2$ are, up to factors of the dimensions, of the order of the machine precision $\boldsymbol{\varepsilon}$. (Here e_i denotes the ith column of the identity matrix I.) From this estimate, we conclude that the Björck–Golub algorithm has equally small backward error angles $\angle(\mathcal{X}, \tilde{\mathcal{X}})$, $\angle(\mathcal{Y}, \tilde{\mathcal{Y}})$ for all bases XD_1, YD_2 of $\mathcal{X}$, $\mathcal{Y}$, where D_1, D_2 are arbitrary diagonal nonsingular matrices.

We also show that the backward error bound in the Björck–Golub algorithm can be improved to $|\Delta X_{ij}| \le f(m,p) \mu_i \boldsymbol{\varepsilon} \max_j |X_{ij}|$, where $f(\cdot)$ is modest polynomial and μ_i is a certain pivot growth factor. The values of μ_i are moderate if the QR factorization is computed using the complete pivoting of Powell and Reid [29]. Numerical evidence shows that in this case the QR factorization is more robust in computing an orthonormal basis for $\mathrm{span}(X)$. To explain this high accuracy and to explore possibilities of further improvement, we devise a new stable algorithm for computing the QR factorization.

The new algorithm is defined and analyzed in section 3. The main novelty is that we use Gaussian elimination with complete pivoting as a preconditioner for the QR factorization. If $P_1 X P_2 = L_x U_x$ is the LU factorization with complete pivoting, and if $L_x = Q_x R_x$ is the QR factorization, then $XP_2 = P_1^T Q_x (R_x U_x)$ is the QR factorization of XP_2. Due to pivoting, L_x is well-conditioned and the matrix Q_x can be efficiently computed by a variant of the modified Gram–Schmidt algorithm. Error analysis shows that the new algorithm is mixed stable. The backward error changes $\mathcal{X}$ to $\tilde{\mathcal{X}} = \mathrm{span}(P_1^T (L_x + \Delta L_x))$, where $\max_{1 \le i \le p} \|\Delta L_x e_i\|_2 / \|L_x e_i\|_2$ depends on the accuracy of Gaussian elimination with pivoting. In other words, columnwise backward error is introduced in the basis L_x of $\mathcal{X}$, rather than in X. The corresponding perturbation in X is elementwise small with a bound similar to the one for QR factorization with complete pivoting.

An analysis shows that the new algorithm computes with nearly the same accuracy orthonormal bases for column spaces of all matrices of the form $D_1 X D_2$, where D_1, D_2 are arbitrary diagonal nonsingular matrices. Similar accuracy is observed in the case of the QR factorization with complete pivoting. This fact is related to the similar form of the elementwise backward errors in the QR factorization and Gaussian elimination. However, the bounds seem to be sharper in the case of Gaussian elimination. In section 4, we give numerical examples that illustrate the benefits of complete pivoting and, in some cases, higher accuracy if the QR factorization is preconditioned using Gaussian elimination.

2. Analysis of the Björck–Golub algorithm. Golub and Zha [24] show that the Björck–Golub algorithm has the same forward error bounds as a backward stable algorithm. In this section, we prove that the algorithm is mixed stable: the computed canonical correlations are close approximations of the exact canonical correlations of certain matrices $\tilde{X} \approx X$ and $\tilde{Y} \approx Y$. Detailed analysis, presented in section 2.1, shows that the backward errors are independent of column scalings of X and Y. In section 2.2, we show that the algorithm achieves much higher accuracy if the QR factorization is computed with complete pivoting of Powell and Reid [29], and in section 2.3 we

analyze the elementwise structure of the backward error.

2.1. Mixed stability. The Björck–Golub algorithm follows a three-step scheme: (i) compute the orthonormal QR factors Q_x, Q_y of the data matrices X, Y; (ii) compute the matrix product $S = Q_x^T Q_y$; (iii) compute the SVD of S. In the standard model of floating-point arithmetic, this algorithm is mixed stable.

THEOREM 2.1. *Let $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_q$ be the singular values computed by the Björck–Golub algorithm. Then there exist $\tilde{X} = X + \Delta X \in \mathbf{R}^{m\times p}$, $\tilde{Y} = Y + \Delta Y \in \mathbf{R}^{m\times q}$ with the following two properties:*

(i) *The values $\max_{1\leq i\leq p} \|\Delta X e_i\|_2/\|Xe_i\|_2$ and $\max_{1\leq i\leq q} \|\Delta Y e_i\|_2/\|Ye_i\|_2$ are of the order of machine precision times a moderate polynomial of the corresponding matrix dimensions.*

(ii) *If $\sigma'_1 \geq \cdots \geq \sigma'_q$ are the exact cosines of the principal angles between $\mathrm{span}(\tilde{X})$ and $\mathrm{span}(\tilde{Y})$, then, for all i, either $\tilde{\sigma}_i = \sigma'_i = 0$ or $|\tilde{\sigma}_i - \sigma'_i|/\sigma'_i$ is less than machine precision times a moderate polynomial of the matrix dimensions.*

Proof. Let $\tilde{Q}_x$, $\tilde{Q}_y$, $\tilde{R}_x$, $\tilde{R}_y$ be the computed approximations of Q_x, Q_y, R_x, R_y, respectively. Then there exist backward perturbations δX, δY and an $\eta_1 \ll 1$ such that

(2.1)
$$X + \delta X = \tilde{Q}_x \tilde{R}_x, \quad Y + \delta Y = \tilde{Q}_y \tilde{R}_y, \quad \max\left\{ \max_{1\leq i\leq p} \frac{\|\delta X e_i\|_2}{\|Xe_i\|_2}, \max_{1\leq i\leq q} \frac{\|\delta Y e_i\|_2}{\|Ye_i\|_2} \right\} \leq \eta_1.$$

To prove relation (2.2), recall that there exists a backward error $\delta_0 X$ and an exactly orthonormal matrix $\hat{Q}_x$ such that $X+\delta_0 X = \hat{Q}_x \tilde{R}_x$, where $\max_i \|\delta_0 X e_i\|_2/\|Xe_i\|_2$ and $\|\tilde{Q}_x - \hat{Q}_x\|_2$ are small multiples of machine precision (cf. [15], [26, section 18.3]). Note that computation of the orthogonal factors is generally not backward stable unless the computed matrices $\tilde{Q}_x$ and $\tilde{Q}_y$ are exactly orthonormal. (We generally cannot say that the computed matrix $\tilde{Q}_x$ is an exact orthogonal factor of some $\tilde{X} \approx X$.) The best we can prove is mixed stability: $\tilde{Q}_x$ and $\tilde{Q}_y$ are close to exact orthogonal factors of $X + \delta X$ and $Y + \delta Y$, respectively. This is ensured since there exists an $\eta_2 \ll 1$ such that

$$\max\{\|\tilde{Q}_x^T \tilde{Q}_x - I\|_F, \|\tilde{Q}_y^T \tilde{Q}_y - I\|_F\} \leq \eta_2.$$

(Here we assume that we use a QR factorization algorithm that ensures near orthogonality of the computed matrices $\tilde{Q}_x$ and $\tilde{Q}_y$.) If $\tilde{Q}_x = Q'_x(I + T'_x)$ is the exact QR factorization of $\tilde{Q}_x$ and if $\eta_2 < 1/4$, then the upper triangular matrix T'_x satisfies (cf. [20, Theorem 2.1]) $\|T'_x\|_2 \leq \eta_2$ and the mixed stability of the computation of the orthonormal QR factor follows from the relation

$$\tilde{X} = X + \Delta X \equiv X + \delta X = Q'_x \left((I + T'_x)\tilde{R}_x \right), \qquad \|\tilde{Q}_x - Q'_x\|_2 \leq \|T'_x\|_2.$$

Similarly, $Y + \delta Y = Q'_y(I + T'_y)\tilde{R}_y$, $\|T'_y\|_2 \leq \eta_2$. Let $\tilde{S} = \mathrm{fl}(\tilde{Q}_x^T \tilde{Q}_y)$ be the computed matrix product. Then $\tilde{S} = \tilde{Q}_x^T \tilde{Q}_y + E_S$, where $\|E_S\|_2 \leq \eta_3$. The computed singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_q$ of $\tilde{S}$ are the exact singular values of $\tilde{S} + \delta\tilde{S}$, where $\|\delta\tilde{S}\|_2 \leq \eta_4$. Here the values of $\eta_3 \ll 1$, $\eta_4 \ll 1$ depend on the details of computation (cf. [23, sections 2.4.8 and 8.3.2]). We can write the matrix $\tilde{S} + \delta\tilde{S}$ as

$$\begin{aligned} \tilde{S} + \delta\tilde{S} &= \tilde{Q}_x^T \tilde{Q}_y + E_S + \delta\tilde{S} = \tilde{Q}_x^T (\tilde{Q}_y + (\tilde{Q}_x^T)^\dagger (E_S + \delta\tilde{S})) \\ &= (I + T'_x)^T \left((Q'_x)^T Q''_y \right) (I + T''_y), \end{aligned}$$

where $(\tilde{Q}_x^T)^\dagger = Q_x'(I+T_x')^{-T}$ is the generalized inverse and $\tilde{Q}_y + (\tilde{Q}_x^T)^\dagger(E_S+\delta\tilde{S}) = Q_y''(I+T_y'')$ is the QR factorization of an almost orthonormal matrix with $\|T_y''\|_2 \leq \eta_5 \ll 1$. Note that

$$\tilde{Y} = Y + \Delta Y \equiv Y + \delta Y + (\tilde{Q}_x^T)^\dagger(E_S+\delta\tilde{S})\tilde{R}_y$$

satisfies $\tilde{Y} = Q_y''(I+T_y'')\tilde{R}_y$ and that, for all $1 \leq i \leq q$,

$$\begin{aligned}\left\| \left((\tilde{Q}_x^T)^\dagger(E_S+\delta\tilde{S})\tilde{R}_y \right) e_i \right\|_2 &\leq \frac{\|E_S\|_2 + \|\delta\tilde{S}\|_2}{1-\|T_x'\|_2} \frac{1+\eta_1}{1-\|T_y'\|_2} \|Ye_i\|_2 \\ &\leq (1+\eta_1)\frac{\eta_3+\eta_4}{(1-\eta_2)^2}\|Ye_i\|_2.\end{aligned}$$

The singular values of $(Q_x')^T Q_y''$ are the cosines of the principal angles between $\mathrm{span}(\tilde{X})$ and $\mathrm{span}(\tilde{Y})$. The proof is completed by noting that the relative difference between the singular values of $\tilde{S}+\delta\tilde{S}$ and $(Q_x')^T Q_y''$ is at most $\|T_x'\|_2 + \|T_y''\|_2 + \|T_x'\|_2\|T_y''\|_2$ (cf. [21, Theorem 3.1]). ∎

The backward errors described in Theorem 2.1 are small normwise relative errors in the columns of the bases X and Y of $\mathcal{X}$, $\mathcal{Y}$, respectively. These estimates, however, do not guarantee that the backward perturbations of $\mathcal{X}$ and $\mathcal{Y}$ are small in the angle metric. Consider now the angle $\angle(\mathcal{X},\tilde{\mathcal{X}})$, where $\mathcal{X} = \mathrm{span}(X)$, $\tilde{\mathcal{X}} = \mathrm{span}(X+\Delta X)$, and X, ΔX are as in Theorem 2.1. If $D_X = \mathrm{diag}(\|Xe_i\|_2)$, $X = X_c D_X$, $\Delta X_c = \Delta X D_X^{-1}$, then $\mathcal{X} = \mathrm{span}(X_c)$, $\tilde{\mathcal{X}} = \mathrm{span}(X_c + \Delta X_c)$, and, for sufficiently small ΔX, an estimate of Wedin [42] yields

$$\sin\angle(\mathcal{X},\tilde{\mathcal{X}}) \leq \|X_c^\dagger\|_2\sqrt{p}\max_{1\leq i\leq p}\frac{\|\Delta X e_i\|_2}{\|Xe_i\|_2}.$$

Hence, we have the following corollary of Theorem 2.1.

COROLLARY 2.1. *Let the assumptions of Theorem* 2.1 *hold. Then there exist subspaces* $\tilde{\mathcal{X}}$, $\tilde{\mathcal{Y}}$ *and a modest polynomial* $f(m,p,q)$ *such that*

$$\sin\angle(\mathcal{X},\tilde{\mathcal{X}}) \leq \|X_c^\dagger\|_2\sqrt{p}\max_{1\leq i\leq p}\frac{\|\Delta X e_i\|_2}{\|Xe_i\|_2}, \qquad \sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}) \leq \|Y_c^\dagger\|_2\sqrt{q}\max_{1\leq i\leq q}\frac{\|\Delta Y e_i\|_2}{\|Ye_i\|_2}$$

and such that the computed singular values $(\tilde{\sigma}_i)_{i=1}^q$ *are, up to a relative error of order* $f(m,p,q)\boldsymbol{\varepsilon}$, *the exact singular values of* $\mathcal{P}_{\tilde{\mathcal{X}}}\mathcal{P}_{\tilde{\mathcal{Y}}}$.

The following perturbation results for the canonical correlations shows that the angles $\angle(\mathcal{X},\tilde{\mathcal{X}})$, $\angle(\mathcal{Y},\tilde{\mathcal{Y}})$ are a natural metric for measuring backward errors.

THEOREM 2.2. *Let* $\mathcal{X}$, $\tilde{\mathcal{X}}$, $\mathcal{Y}$, $\tilde{\mathcal{Y}}$ *be subspaces of* $\mathbf{R}^m$ *(or* $\mathbf{C}^m$*) with* $\dim(\mathcal{X}) = \dim(\tilde{\mathcal{X}})$, $\dim(\mathcal{Y}) = \dim(\tilde{\mathcal{Y}})$, *and let*

$$\eta = \sin\angle(\mathcal{X},\tilde{\mathcal{X}}) + \sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}) + \sin\angle(\mathcal{X},\tilde{\mathcal{X}})\cdot\sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}).$$

Let $\Sigma = \mathrm{diag}(\sigma_i)$, $\Xi = \mathrm{diag}(\xi_j)$, $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_i)$, $\tilde{\Xi} = \mathrm{diag}(\tilde{\xi}_j)$ *be the singular values of* $\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}$, $(I-\mathcal{P}_{\mathcal{X}})\mathcal{P}_{\mathcal{Y}}$, $\mathcal{P}_{\tilde{\mathcal{X}}}\mathcal{P}_{\tilde{\mathcal{Y}}}$, $(I-\mathcal{P}_{\tilde{\mathcal{X}}})\mathcal{P}_{\tilde{\mathcal{Y}}}$, *respectively. Then*

$$\|\Sigma-\tilde{\Sigma}\|_2 \leq \eta, \qquad \|\Xi-\tilde{\Xi}\|_2 \leq \eta. \tag{2.2}$$

Furthermore, let $\Theta = \mathrm{diag}(\vartheta_i)$, $\tilde{\Theta} = \mathrm{diag}(\tilde{\vartheta}_i)$ *be the principal angles between* $\mathcal{X}$ *and* $\mathcal{Y}$, *and between* $\tilde{\mathcal{X}}$ *and* $\tilde{\mathcal{Y}}$, *respectively. Then* $\eta < 1-\sqrt{2}/2$ *implies*

$$\|\Theta-\tilde{\Theta}\|_2 \leq \frac{\eta}{\sqrt{1-(\eta+1/\sqrt{2})^2}}. \tag{2.3}$$

Proof. The assumptions about the dimensions of the subspaces imply $\sin\angle(\mathcal{X},\tilde{\mathcal{X}}) = \|\mathcal{P}_{\mathcal{X}} - \mathcal{P}_{\tilde{\mathcal{X}}}\|_2$ and $\sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}) = \|\mathcal{P}_{\mathcal{Y}} - \mathcal{P}_{\tilde{\mathcal{Y}}}\|_2$. Since the error in the singular values of a perturbed matrix is not larger than the spectral norm of the perturbation (cf. [23, Corollary 8.3.2]), we have

$$\begin{aligned}
&\| \Sigma - \tilde{\Sigma}\|_2 \\
&\le \|\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}} - \mathcal{P}_{\tilde{\mathcal{X}}}\mathcal{P}_{\tilde{\mathcal{Y}}}\|_2 = \|(\mathcal{P}_{\mathcal{X}} - \mathcal{P}_{\tilde{\mathcal{X}}})\mathcal{P}_{\mathcal{Y}} + \mathcal{P}_{\mathcal{X}}(\mathcal{P}_{\mathcal{Y}} - \mathcal{P}_{\tilde{\mathcal{Y}}}) - (\mathcal{P}_{\mathcal{X}} - \mathcal{P}_{\tilde{\mathcal{X}}})(\mathcal{P}_{\mathcal{Y}} - \mathcal{P}_{\tilde{\mathcal{Y}}})\|_2 \\
&\le \sin\angle(\mathcal{X},\tilde{\mathcal{X}}) + \sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}) + \sin\angle(\mathcal{X},\tilde{\mathcal{X}}) \cdot \sin\angle(\mathcal{Y},\tilde{\mathcal{Y}}).
\end{aligned}$$

Similarly, the second inequality in (2.2) follows from $\|\Xi - \tilde{\Xi}\|_2 \le \|(I - \mathcal{P}_{\mathcal{X}})\mathcal{P}_{\mathcal{Y}} - (I - \mathcal{P}_{\tilde{\mathcal{X}}})\mathcal{P}_{\tilde{\mathcal{Y}}}\|_2$ and from the identity $\angle(\mathcal{X},\tilde{\mathcal{X}}) = \angle(\mathcal{X}^{\perp},\tilde{\mathcal{X}}^{\perp})$. To prove (2.3), we first note that

$$|\vartheta_i - \tilde{\vartheta}_i| = \int_{\min\{\sigma_i,\tilde{\sigma}_i\}}^{\max\{\sigma_i,\tilde{\sigma}_i\}} \frac{dt}{\sqrt{1-t^2}} = \int_{\min\{\xi_i,\tilde{\xi}_i\}}^{\max\{\xi_i,\tilde{\xi}_i\}} \frac{dt}{\sqrt{1-t^2}} \tag{2.4}$$

and that $\min\{\max\{\sigma_i,\tilde{\sigma}_i\},\max\{\xi_i,\tilde{\xi}_i\}\} \le 1/\sqrt{2} + \eta$. Then we estimate the integrals in (2.4). ∎

Corollary 2.1 shows that the backward error angles in the Björck–Golub algorithm are independent of column scalings of the bases X and Y and that these angles might be large only if $\min_{D=\mathrm{diag}}\kappa_2(XD)$ and $\min_{D=\mathrm{diag}}\kappa_2(YD)$ are large. (Here we recall the near optimality of the spectral condition number $\kappa_2(X_c) = \|X_c\|_2\|X_c^{\dagger}\|_2$: $\kappa_2(X_c) \le \sqrt{p}\min_{D=\mathrm{diag}}\kappa_2(XD)$; see [39].) In that case, certain, even very small, normwise relative changes of the columns of the ill-conditioned basis X might cause arbitrarily large flutter of the corresponding subspace. The following example illustrates this situation. Let

$$X = \begin{bmatrix} 1 & 1 \\ \epsilon & -\epsilon \\ \epsilon & \epsilon \end{bmatrix}, \qquad Y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \tilde{X} = \begin{bmatrix} 1 & 1 \\ \epsilon & -\epsilon \\ \epsilon & -\epsilon \end{bmatrix}, \qquad |\epsilon| \ll 1, \tag{2.5}$$

and let $\mathcal{X} = \mathrm{span}(X)$, $\tilde{\mathcal{X}} = \mathrm{span}(\tilde{X})$, $\mathcal{Y} = \mathrm{span}(Y)$. The angle between $\mathcal{X}$ and $\mathcal{Y}$ is fairly large ($\mathcal{X}$ is close to $\mathrm{span}(e_1, e_2)$) and the corresponding columns of X and $\tilde{X}$ differ by small ($O(\epsilon)$) angles. However, it holds that $\mathcal{Y} \subset \tilde{\mathcal{X}}$. Using MATLAB with $\epsilon = 1000 * \mathtt{eps} \approx 2.22 \cdot 10^{-13}$, we compute the orthogonal factors of X and $\tilde{X}$, respectively, as

$$\tilde{Q}_x \approx \begin{bmatrix} -1.00 & 2.22\cdot 10^{-13} \\ -2.22\cdot 10^{-13} & -1.00 \\ -2.22\cdot 10^{-13} & 2.46\cdot 10^{-26} \end{bmatrix}, \qquad \widetilde{\tilde{Q}_x} \approx \begin{bmatrix} -1.00 & 3.14\cdot 10^{-13} \\ -2.22\cdot 10^{-13} & -7.07\cdot 10^{-1} \\ -2.22\cdot 10^{-13} & -7.07\cdot 10^{-1} \end{bmatrix}.$$

Hence, the principal angles between $\mathcal{X}$ and $\mathcal{Y}$ are poorly determined in the presence of such errors. This behavior is also captured by the following theorem of Golub and Zha [24].

THEOREM 2.3. *Let $X \in \mathbf{R}^{m\times p}$ and $Y \in \mathbf{R}^{m\times q}$ be full column rank matrices and let $X = X_c D_X$, $Y = Y_c D_Y$, where $D_X = \mathrm{diag}(\|Xe_i\|_2)$, $D_Y = \mathrm{diag}(\|Ye_i\|_2)$. Let $\tilde{X} = X + \Delta X$, $\tilde{Y} = Y + \Delta Y$ be full column rank matrices such that $|\Delta X| \le \epsilon G_X|X|$, $|\Delta Y| \le \epsilon G_Y|Y|$, where $0 \le \epsilon \ll 1$ and G_X, G_Y are matrices with nonnegative elements. Let $\mathcal{X} = \mathrm{span}(X)$, $\mathcal{Y} = \mathrm{span}(Y)$, $\tilde{\mathcal{X}} = \mathrm{span}(\tilde{X})$, $\tilde{\mathcal{Y}} = \mathrm{span}(\tilde{Y})$. Let $\mathcal{C}(\mathcal{X},\tilde{\mathcal{X}})$ be the orthogonal complement of $\mathcal{X}\bigcap\tilde{\mathcal{X}}$ in $\mathcal{X}+\tilde{\mathcal{X}}$, and let ξ be the minimal angle between $\mathcal{C}(\mathcal{X},\tilde{\mathcal{X}})$ and $\mathcal{Y}$. Similarly, let ζ be defined as the minimal angle between*

$\mathcal{C}(\mathcal{Y},\tilde{\mathcal{Y}})$ and $\tilde{\mathcal{X}}$. If Σ, $\tilde{\Sigma}$ are the diagonal matrices of the cosines of the principal angles between $\mathcal{X}$ and $\mathcal{Y}$, and between $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$, respectively, then

$$\|\tilde{\Sigma}-\Sigma\|_2 \leq \epsilon\left(\sqrt{p(m-p)}\|G_X\|_2\kappa_2(X_c)\cos\xi+\sqrt{q(m-q)}\|G_Y\|_2\kappa_2(Y_c)\cos\zeta\right).$$

This theorem shows that the accuracy of the singular values of $\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}$ depends on the condition numbers of the column scaled matrices X_c and Y_c. It can be shown that the dimension factors $\sqrt{p(m-p)}$ and $\sqrt{q(m-q)}$ can be improved to $\sqrt{p}$ and $\sqrt{q}$, respectively.

We conclude our analysis of the Björck–Golub algorithm with an experimental illustration of the bounds in Theorem 2.1 and Corollary 2.1.

Example 2.1. We generate test pairs (X,Y) as follows. We write X, Y as $X=X_cD_X$, $Y=Y_cD_Y$, where $D_X=\mathrm{diag}(\|Xe_i\|_2)$, $D_Y=\mathrm{diag}(\|Ye_i\|_2)$, and $\kappa_2(X_c),\kappa_2(Y_c)\in\{10^i,\ i=2,\ldots,6\}$, $\kappa_2(D_X)\in\{10^8,10^{12},10^{16}\}$, $\kappa_2(D_Y)\in\{10^9,10^{13},10^{15}\}$. For fixed values of the condition numbers $(\kappa_2(X_c),\kappa_2(D_X),\kappa_2(Y_c),\kappa_2(D_Y))$ we generate X_c, D_X, Y_c, D_Y with different distributions of singular values. We use the procedure `DLATM1()` from [14], and we choose the values of the parameter `MODE` so that the distributions of the singular values of X_c, D_X, Y_c, D_Y are from the set $\{5,3\}\times\{5\}\times\{5,-4\}\times\{5\}$. In this way, we generate 900 test pairs (X,Y), divided into 25 classes, where the pairs from the same class $\mathcal{C}_{ij}$ have nearly the same values of $(\kappa_2(X_c),\kappa_2(Y_c))\approx(10^i,10^j)$, $2\leq i,j\leq 6$. We measure the backward error angles in the following way. We use single precision floating-point arithmetic ($\boldsymbol{\varepsilon}\approx 10^{-8}$) to find approximate orthonormal bases $\tilde{Q}_x$ and $\tilde{Q}_x^{\perp}$ for $\mathcal{X}$ and $\mathcal{X}^{\perp}$, respectively. Then, we use double precision computation to compute the sine of the angle between $\mathrm{span}(\tilde{Q}_x)$ and $\mathcal{X}$. This is accomplished by an application of the Björck–Golub algorithm to the matrices $\tilde{Q}_x^{\perp}$ and X. The same procedure is applied to $\tilde{Q}_y$ and Y. (It is clear from the proof of Theorem 2.1 that the computation of the orthonormal bases introduces the major part of the error. Hence, this experiment gives a useful insight into the overall accuracy of the algorithm.) The QR factorizations are computed using the LAPACK [1] procedure `SGEQRF()`. The results of the test with $m=200$, $p=100$, $q=50$ are given in Figure 2.1, where

$$e_{ij}=\max_{(X,Y)\in\mathcal{C}_{ij}}\max\{\sin\angle(\mathrm{span}(\tilde{Q}_x),\mathcal{X}),\sin\angle(\mathrm{span}(\tilde{Q}_y),\mathcal{Y})\}.$$

From Corollary 2.1, it follows that e_{ij} is bounded (roughly) by $f(m,p,q)\boldsymbol{\varepsilon}\max\{\kappa_2(X_c),\kappa_2(Y_c)\}\approx f(m,p,q)10^{\max\{i,j\}-8}$, where $f(\cdot)$ is a modest function of the dimensions. Figure 2.1 indicates that this bound is almost attainable.

2.2. Effects of complete pivoting in the QR factorization. If the matrix X is changed to $X+\Delta X$, where $\|\Delta X\|_2/\|X\|_2$ is small, the relevant condition number that determines the sensitivity of the QR factorization of X is $\kappa_2(X)$ (cf. [31]). If it is known that ΔX is a small columnwise perturbation of X as in relation (2.2), we can write $X+\Delta X=(X_c+\Delta X_c)D_X$, where $D_X=\mathrm{diag}(\|Xe_i\|_2)$, $X=X_cD_X$, $\Delta X=\Delta X_cD_X$, and $\|\Delta X_c\|_2/\|X_c\|_2$ is small. Thus, the sensitivity of the QR factorization is determined by $\kappa_2(X_c)$, which can be much smaller than $\kappa_2(X)$. Hence, the finer structure of the perturbation ΔX makes it possible to eliminate artificial ill-conditioning, in this case represented by large $\kappa_2(X)$, which is the result of different lengths of matrix columns.

In some situations, artificial ill-conditioning is the result of heavy row weighting. For example, if $X=D'X_1$, where D' is an ill-conditioned diagonal matrix and X_1

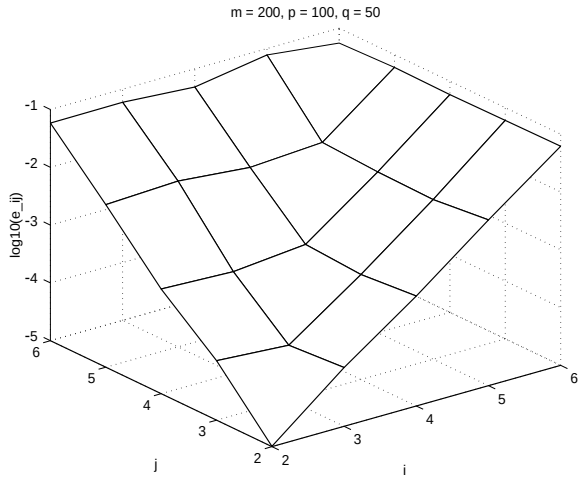


FIG. 2.1. *The values of* $\log_{10} e_{ij}$, $2 \le i, j \le 6$ *in Example* 2.1. *Note that* $\log_{10} e_{ij} \approx \max\{i, j\} - 7$, *almost as predicted by the theory.*

is a well-conditioned matrix, then it may not be possible to scale the columns of X to obtain a well-conditioned matrix X_c. In that case, both $\kappa_2(X)$ and $\kappa_2(X_c)$ are large, and numerical experiments show that the QR factorization is computed with large errors. This stability problem of the QR factorization of matrices with heavily weighted rows is well known (e.g., in solving weighted least squares problems) and there exist computational experience and satisfactory backward error bounds which show that the factorization is more robust if it is computed with column and row interchanges; see Powell and Reid [29], Barlow [3], Van Loan [40], Björck [8, section 4.4.2]. The QR factorization with complete (column and row) pivoting is described in the following result of Powell and Reid [29].

PROPOSITION 2.1. *Let the QR factorization of* X *be computed in floating-point arithmetic by a sequence of* p *Householder reflections, and let* $\tilde{X} = X + \delta X$, *where* δX *is the backward error. Let* $\tilde{X}^{(k)}$, $k \in \{1, \ldots, p\}$, *denote the floating-point matrix computed in the* k*th step of the algorithm, let* $X^{(k)}$, $k \in \{1, \ldots, p\}$, *denote the matrix in the* k*th step of exact computation, and let*

$$\rho_i(\tilde{X}) = \max_{j,k} |(\tilde{X}^{(k)})_{ij}|, \qquad \rho_i(X) = \max_{j,k} |(X^{(k)})_{ij}|, \qquad i = 1, \ldots, m. \tag{2.6}$$

Then there exists a modest polynomial $h(p)$ *such that* $|\delta X_{ij}| \le h(p)\varepsilon\rho_i(\tilde{X})$. *Furthermore, if the columns of* X *are permuted following the pivoting of Golub* [22], *and if, in addition, the rows of the matrices* $X^{(k)}$ *are permuted so that, for all* k, $|(X^{(k)})_{kk}| = \max_{i \ge k} |(X^{(k)})_{ik}|$, *then*

$$\max_j |(X^{(k+1)})_{k,j}| \le \sqrt{m}|(X^{(k)})_{kk}|, \qquad \textit{and} \quad \rho_i(X) \le (1+\sqrt{2})^{i-1}\sqrt{m}\max_j |X_{ij}|.$$

The pivot growth factors $\mu_i(X) = \frac{\rho_i(X)}{\max_j |X_{ij}|}$, $1 \le i \le m$, *are usually moderate and the exponential growth is attained only in pathological cases.*

Barlow [3] shows that the G-algorithm of Bareiss [2] has a backward error bound similar to the one in Proposition 2.1. Cox and Higham [12] show that the row pivoting

in Proposition 2.1 can be replaced by the initial sorting of matrix rows by decreasing ℓ_∞ norms. Hence, complete pivoting reduces to column pivoting and it can be implemented in high performance software; see Quintana-Orti, Sun, and Bischof [30].

From Proposition 2.1, it follows that complete pivoting ensures an elementwise backward error bound which is invariant under row scalings. Before we analyze theoretical implications of this fact, we illustrate the effects of pivoting in numerical computation. We first show that row weighting does not increase the backward error angle if the QR factorization is computed with complete pivoting.

Example 2.2. We follow a test procedure similar to the one in Example 2.1. The only difference is that instead of $X = X_s D_X$ we generate $X = D'_X X_s D''_X$, where D'_X, D''_X are diagonal matrices generated in the same way as D_X. We generate 900 matrices divided into 5 groups with $\kappa_2(X_s) = 10^i$, $i = 2, 3, 4, 5, 6$. For each generated matrix X we also compute $\kappa_2(X_c)$, where X_c is obtained by normalizing the columns of X. The matrices X_s, D'_X, and D''_X are generated in a sequence of nested loops with different choices of parameters (cf. Example 2.1). For the orthonormal basis $\tilde{Q}_x$, computed with the QR factorization with complete pivoting, we compute $e_x = \sin\angle(\tilde{Q}_x, \mathcal{X})$. We also compute $f_x = \sin\angle(\tilde{\tilde{Q}}_x, \mathcal{X})$, where $\tilde{\tilde{Q}}_x$ is computed without pivoting. The values of f_x, e_x, $\kappa_2(X_c)$ shown in Figure 2.2 demonstrate that the error angle f_x increases with large $\kappa_2(X_c)$, while the values of e_x depend only on $\kappa_2(X_s)$. Note that $e_x/\kappa_2(X_c)$ is much smaller than the roundoff $\boldsymbol{\varepsilon} \approx 10^{-8}$ and that $e_x \approx \kappa_2(X_s)\boldsymbol{\varepsilon}$. (The almost periodic behavior of $\kappa_2(X_c)$ is due to the fact that the matrices are generated in a sequence of nested loops.) In this example, we have observed similar accuracy if the row pivoting is replaced by the initial row ordering.

In the next two examples, we show the difference in the forward errors for the two variants of the Björck–Golub algorithm.

Example 2.3. In this example we show that QR-based computation of the orthogonal bases can introduce large error and it can fail to detect that, for example, one of the principal angles is close to $\pi/2$. We take the bases X and Y to be

$$X \approx \begin{bmatrix} 0.57378941 \cdot 10^{17} & -0.74737239 \cdot 10^{09} & -0.10439621 \cdot 10^{02} \\ -0.75415686 \cdot 10^{29} & 0.25173789 \cdot 10^{22} & -0.11089462 \cdot 10^{14} \\ -0.52912208 \cdot 10^{19} & 0.51559708 \cdot 10^{12} & -0.63842515 \cdot 10^{04} \\ 0.26020839 \cdot 10^{26} & -0.72667785 \cdot 10^{18} & 0.14745371 \cdot 10^{10} \\ 0.21463361 \cdot 10^{22} & -0.76107815 \cdot 10^{14} & 0.39906168 \cdot 10^{06} \\ 0.13388386 \cdot 10^{26} & -0.48858418 \cdot 10^{19} & 0.75605997 \cdot 10^{11} \\ -0.43084490 \cdot 10^{20} & 0.33985776 \cdot 10^{13} & -0.38962076 \cdot 10^{05} \end{bmatrix},$$

$$Y \approx \begin{bmatrix} 0.12378225 \cdot 10^{+00} & -0.17331250 \cdot 10^{+13} \\ 0.84008590 \cdot 10^{-09} & 0.17773952 \cdot 10^{+05} \\ -0.26428604 \cdot 10^{-14} & -0.98536731 \cdot 10^{-01} \\ 0.13059467 \cdot 10^{-12} & -0.80072369 \cdot 10^{+00} \\ 0.18943973 \cdot 10^{-11} & -0.20708348 \cdot 10^{+01} \\ -0.16178360 \cdot 10^{+01} & -0.33048027 \cdot 10^{+13} \\ 0.40286435 \cdot 10^{-06} & 0.10409793 \cdot 10^{+09} \end{bmatrix},$$

where the entries of the corresponding double precision arrays are shown to eight decimal places. In Table 2.1, we show the computed approximations of the cosines of the principal angles. (π-Björck–Golub refers to the Björck–Golub algorithm with pivoting suggested by Powell and Reid. Algorithm 3.1 is described in section 3. At this point, the only purpose of the last row in Table 2.1 is to give a second set of double precision reference values (singular value approximations).) Since $\tilde{\sigma}_2 \approx 3.7\boldsymbol{\varepsilon}$, it is determined

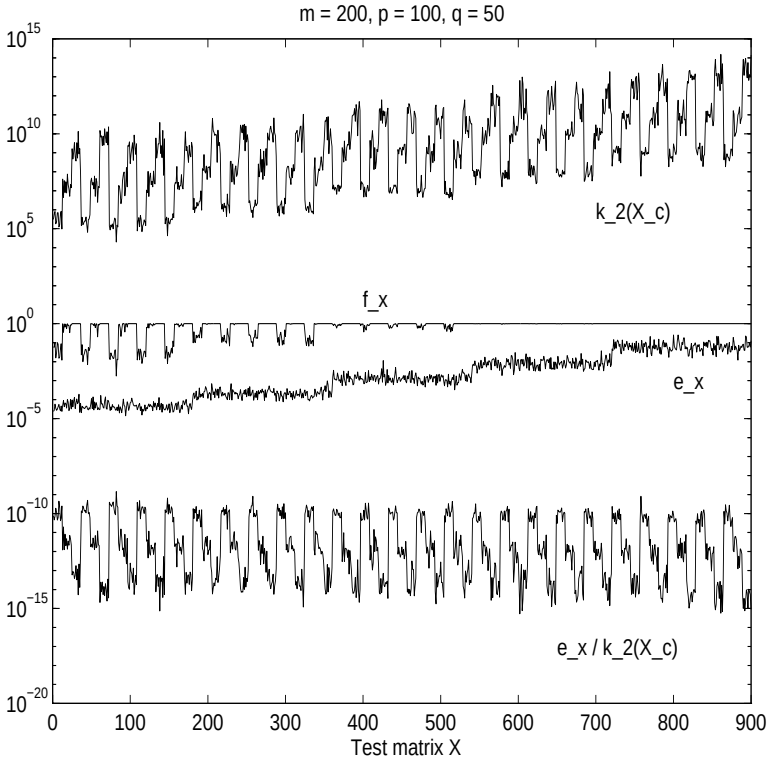


FIG. 2.2. *The values of* e_x, f_x, $\kappa_2(X_c)$, $e_x/\kappa_2(X_c)$ *in Example* 2.2.

TABLE 2.1
The computed singular values in Example 2.3.

$\tilde{\sigma}_i$	Björck–Golub (single)	Björck–Golub (double)	π-Björck–Golub (single)
$\tilde{\sigma}_1$	$0.10000002 \cdot 10^1$	0.9999999910693745	$0.10000000 \cdot 10^1$
$\tilde{\sigma}_2$	0.91120803	$0.2269574724944604 \cdot 10^{-6}$	$0.21987161 \cdot 10^{-6}$
Algorithm 3.1 (double): $\tilde{\sigma}_1 \approx 0.9999999910693748$, $\tilde{\sigma}_2 \approx 0.2219392787298458 \cdot 10^{-6}$.			

only to an absolute uncertainty of order $\boldsymbol{\varepsilon}$. To illustrate this, we multiply the entries of X and Y by randomly chosen numbers $1 \pm \epsilon_{ij}$ with $|\epsilon_{ij}| \leq 10^{-4}$. The single precision Björck–Golub algorithm and π-Björck–Golub algorithm compute, respectively, $\tilde{\sigma}_2 \approx 0.99112201$ and $\tilde{\sigma}_2 \approx 0.75409122 \cdot 10^{-6}$. The double precision computation gives $\tilde{\sigma}_2 \approx 0.7685475597770073 \cdot 10^{-6}$. The maximal principal angle is not sensitive to this change since $\tilde{\vartheta}_2 \approx \arccos(0.21987161 \cdot 10^{-6})$ and $\tilde{\tilde{\vartheta}}_2 \approx \arccos(0.75409122 \cdot 10^{-6})$ satisfy $\tilde{\vartheta}_2/\tilde{\tilde{\vartheta}}_2 \approx 1.0000003$ and $(\pi/2)/\min\{\tilde{\vartheta}_2, \tilde{\tilde{\vartheta}}_2\} \approx 1.0000005$. This is obvious from the formula

$$\vartheta_i = \frac{\pi}{2} - \int_0^{\sigma_i} \frac{dt}{\sqrt{1-t^2}}.$$

REMARK 2.1. *The value of* $\tilde{\sigma}_1 = 0.10000002 \cdot 10^1$ *in Table* 2.1 *shows that mixed stability is the right framework for the numerical analysis of principal angle computation. No backward perturbation can in exact arithmetic lead to* $0.10000002 \cdot 10^1$ *as*

TABLE 2.2
The computed singular values in Example 2.4.

$\tilde{\sigma}_i$	Björck–Golub (single)	Björck–Golub (double)	π-Björck–Golub (single)
$\tilde{\sigma}_1$	0.99059296	$0.5015345317976148 \cdot 10^{-2}$	$0.48222207 \cdot 10^{-2}$
$\tilde{\sigma}_2$	$0.25136729 \cdot 10^{-9}$	$0.2510846255712600 \cdot 10^{-9}$	$0.22261035 \cdot 10^{-9}$
Algorithm 3.1 (double): $\tilde{\sigma}_1 \approx 0.5015345505648627 \cdot 10^{-2}$, $\tilde{\sigma}_2 \approx 0.2510846257369576 \cdot 10^{-9}$.			

TABLE 2.3
The results in Example 2.4 *with perturbed data.*

$\tilde{\sigma}_i$	Björck–Golub (single)	Björck–Golub (double)	π-Björck–Golub (single)
$\tilde{\sigma}_1$	0.99053305	$0.5013070874085607 \cdot 10^{-2}$	$0.48202435 \cdot 10^{-2}$
$\tilde{\sigma}_2$	$0.24991473 \cdot 10^{-9}$	$0.2502365149198476 \cdot 10^{-9}$	$0.22178702 \cdot 10^{-9}$
Algorithm 3.1 (double): $\tilde{\sigma}_1 \approx 0.5013070984780922 \cdot 10^{-2}$, $\tilde{\sigma}_2 \approx 0.2502365153154889 \cdot 10^{-9}$.			

the cosine of a principal angle. (Strictly speaking, the Björck–Golub algorithm and Algorithm 3.1 *in section* 3 *are not backward stable.)*

Example 2.4. The most critical part of the principal angle computation is the computation of the orthonormal bases of the given spaces. If that computation introduces large errors that "rotate" the initial spaces, there is no way to tell which singular value of $\mathcal{P}_{\mathcal{X}}\mathcal{P}_{\mathcal{Y}}$ will suffer the largest perturbation. In this example, we show that the largest error might be in the largest singular value, while the smallest one is computed very accurately. Let

$$X \approx \begin{bmatrix} 0.81909804 \cdot 10^{01} & -0.85610022 \cdot 10^{02} & -0.19108842 \cdot 10^{12} \\ -0.31793150 \cdot 10^{11} & 0.15111104 \cdot 10^{13} & 0.26747300 \cdot 10^{22} \\ -0.51921289 \cdot 10^{12} & 0.32394455 \cdot 10^{13} & 0.74985519 \cdot 10^{22} \\ -0.12806811 \cdot 10^{16} & 0.32962115 \cdot 10^{16} & 0.11506216 \cdot 10^{26} \\ 0.11302525 \cdot 10^{03} & -0.85968597 \cdot 10^{03} & -0.16852694 \cdot 10^{13} \\ 0.85886880 \cdot 10^{16} & -0.89292760 \cdot 10^{17} & -0.17015941 \cdot 10^{27} \\ 0.14028936 \cdot 10^{05} & -0.69895642 \cdot 10^{06} & -0.11412105 \cdot 10^{16} \end{bmatrix},$$

$$Y \approx \begin{bmatrix} -0.77654567 \cdot 10^{-4} & -0.42605337 \cdot 10^{-06} \\ -0.52320495 \cdot 10^{-7} & -0.42627118 \cdot 10^{-09} \\ -0.12184166 \cdot 10^{-6} & -0.47657759 \cdot 10^{-09} \\ 0.34901023 \cdot 10^{-6} & 0.19476305 \cdot 10^{-08} \\ 0.22741771 \cdot 10^{+4} & 0.86991999 \cdot 10^{+01} \\ 0.15964494 \cdot 10^{-8} & 0.15686126 \cdot 10^{-10} \\ 0.75523679 \cdot 10^{-9} & 0.46711879 \cdot 10^{-11} \end{bmatrix}.$$

The computed singular values are given in Table 2.2. (As in Example 2.3, the last row in Table 2.2 and in Table 2.3 is used only as a second set of reference values.) To illustrate how well σ_1 and σ_2 are determined by the data, we introduce random rounding errors of order 10^{-4} into the entries of X and Y and we run the test again. The results are shown in Table 2.3.

2.3. Elementwise structure of the backward error. In this section, we try to gain further understanding of the structure of the backward error in the QR factorization with complete pivoting and its implications about the forward perturbation of the orthogonal QR factor. We use the following useful observation:

(i) Let $X = X_c' D$, where D is a diagonal matrix of powers of the base of the floating-point arithmetic. Then, in the absence of underflow and overflow, the

QR factorizations of X and X'_c are numerically equivalent in the sense that both compute the same floating-point approximation $\tilde{Q}_x \approx Q_x$. (Q_x denotes the exact orthogonal factor of X.)

To simplify the notation, we assume that the initial matrix is permuted so that no column or row interchanges are necessary in the Powell–Reid QR factorization with pivoting. We also assume that we can write $X = D'X_sD = X'_cD$, where D, D' are nonsingular diagonal scalings and the diagonals of D are powers of the base of the floating-point arithmetic, and that no column interchanges are necessary to compute the QR factorization with column pivoting of X'_c. In that case, neither the column or the row interchanges are necessary in the Powell–Reid row pivoting, and we assume that the pivot growth factors $\mu_i(\tilde{X}'_c)$ are moderate. We let $\tilde{Q}_x$ denote the computed approximation of the orthonormal basis Q_x. Using Proposition 2.1, we conclude that there exist an exactly orthonormal matrix Q'_x and a backward error δX such that the computed triangular factor $\tilde{R}_x$ satisfies

$$X + \delta X = Q'_x\tilde{R}_x, \qquad |\delta X_{ij}| \le h(p)\boldsymbol{\varepsilon}\mu_i(\tilde{X}) \max_j |X_{ij}|. \tag{2.7}$$

By observation (i), we can also write

$$\tilde{X}'_c \equiv X'_c + \delta(X'_c) = Q'_x(\tilde{R}_xD^{-1}), \qquad |(\delta(X'_c))_{ij}| \le h(p)\boldsymbol{\varepsilon}\mu_i(\tilde{X}'_c) \max_j |(X'_c)_{ij}|. \tag{2.8}$$

We can rewrite relation (2.8) to

$$D'(X_s + D'^{-1}\delta(X'_c))D = Q'_x\tilde{R}_x, \qquad \left|\frac{(\delta(X'_c))_{ij}}{D'_{ii}}\right| \le h(p)\boldsymbol{\varepsilon}\mu_i(\tilde{X}'_c) \max_j |(X_s)_{ij}|. \tag{2.9}$$

Since the computed matrix $\tilde{Q}_x$ is nearly orthonormal and since $\|\tilde{Q}_x - Q'_x\|_2$ is (up to a factor of the dimensions m, p) of order $\boldsymbol{\varepsilon}$, the main issue in the perturbation of $\mathcal{X} = \mathrm{span}(X)$ is how the matrix Q_x changes in the presence of the following perturbation:

$$X \equiv D'X_sD \longmapsto X + \delta X = D'(X_s + \delta X_s)D, \quad |(\delta X_s)_{ij}| \le h(p)\boldsymbol{\varepsilon}\mu_i(\tilde{X}'_c) \max_j |(X_s)_{ij}|. \tag{2.10}$$

The existing perturbation results for the QR factorization $X = Q_xR_x$ can be roughly divided into two groups. In the first group, we have error bounds in terms of $\|\delta X\|_F/\|X\|_2$ and a typical estimate is of the form

$$\|\delta Q_x\|_F \le \sqrt{2}\kappa_2(X)\frac{\|\delta X\|_F}{\|X\|_2}, \tag{2.11}$$

as in [37] (derived using fixed-point and operator theory; see also [31], [35]) or

$$\|\delta Q_x\|_F \le \sqrt{2} \max_{0\le t\le 1} \|(X + t\cdot(\delta X))^{-1}\|_2\|\delta X\|_F, \tag{2.12}$$

as in [5] (derived for $m \times m$ nonsingular matrices using calculus on the manifold $\mathbf{GL}(m)$). In the second group are the results of Sun [36] and Zha [43]. These results are best represented by the following theorem due to Zha: if $|\delta X| \le \epsilon G_X|X|$, with $G_X \ge 0$, $\|G_X\|_\infty = 1$, and with sufficiently small ϵ, then

$$\|\delta Q_x\|_\infty \le z(m,p)\epsilon\| \, |R_x| \cdot |R_x^{-1}| \, \|_\infty, \tag{2.13}$$

where $z(m,p)$ is a modestly growing function. Zha has shown that the bound (2.13) is sharp. (Here the matrix absolute values and the inequalities involving matrices are understood elementwise; $\|\cdot\|_\infty$ is the matrix norm induced by the ℓ_∞ vector norm.) An important feature of the bound (2.13) is that it is invariant under replacing $X+\delta X$ with $(X+\delta X)D_x$, where D_x is an arbitrary diagonal nonsingular matrix. Hence, the size of the error in the case of columnwise perturbations ($\|\delta X e_i\|_2 \ll \|Xe_i\|_2$ for all i) is essentially determined by $\mathrm{cond}(X) = \min_{D_x=\mathrm{diag}} \kappa_2(XD_x)$. However, $\mathrm{cond}(X)$ may be large if X has heavily weighted rows, for example, if X is composed as $X = D'X_sD$, where D and D' are ill-conditioned diagonal scalings, and X_s has moderate (say) $\kappa_2(X_s)$. Thus, the bound (2.13) is not sharp in the case of perturbation (2.10).

It does not seem simple to deal with row scaling in the perturbation analysis of the QR factorization. In the case of column scaling, we use the fact that both $X = Q_xR_x$ and $XD_x = Q_x(R_xD_x)$ are the essentially unique QR factorizations, and we can take advantage of the fact that $\kappa_2(XD_x)$ might be much smaller than $\kappa_2(X)$. In other words, if the ill-conditioning can be "filtered out" by column scaling, it is artificial and it does not affect the accuracy of the computation. On the other hand, the relation between the orthonormal QR factors of the matrices X, X_s, $X+\delta X$, $X_s+\delta X_s$ in relation (2.10) is not obvious. (For an asymptotic analysis see [32].) We discuss the solution to this problem in the next section, where we describe a new algorithm that is based on another fundamental matrix factorization, namely, the LU factorization.

3. The new algorithm. The main difference between our new algorithm and the algorithm of Björck and Golub is in the computation of the orthonormal bases of $\mathcal{X} = \mathrm{span}(X)$ and $\mathcal{Y} = \mathrm{span}(Y)$. Instead of the QR factorization applied directly to the matrices X and Y, we first compute the LU factorizations of X and Y using Gaussian elimination with complete (or partial) pivoting. Then we use the computed unit lower trapezoidal LU factors as new bases for $\mathcal{X}$, $\mathcal{Y}$. (Note that the numbers of parameters in the unit lower trapezoidal LU factors of X and Y are equal to the dimensions of the corresponding Stiefel manifolds of $m\times p$ and $m\times q$ orthonormal matrices.)

ALGORITHM 3.1. CC(X,Y).

Input $X\in\mathbf{R}^{m\times p}$, $Y\in\mathbf{R}^{m\times q}$ full column rank matrices with $p\geq q$.

Step 1 Compute the LU factorizations with pivoting, $P_1XP_2 = L_xU_x$, $P_3YP_4 = L_yU_y$. (For partial pivoting, $P_2 = I_p$, $P_4 = I_q$.)

Step 2 Compute the QR factorizations $L_x = Q_xR_x$, $L_y = Q_yR_y$, using the modified Gram–Schmidt algorithm.

Step 3 Compute the matrix $S = Q_x^T((P_1P_3^T)Q_y)$ and the SVD of S, $S = W\Sigma V^T$.

Output Return the matrices Σ, $P_1^TQ_xW$, $P_3^TQ_yV$.

Since L_x and L_y are lower trapezoidal, the cost of the modified Gram–Schmidt orthogonalization can be reduced using the following algorithm.

ALGORITHM 3.2. MGS_LT(L).

for $j = p, p-1, \ldots, 1$
  $L_x(j:m,j) := (1/\|L_x(j:m,j)\|_2)L_x(j:m,j)$
  for $i = j-1, j-2, \ldots, 1$
    $L_x(j:m,i) := L_x(j:m,i) - \left((L_x(j:m,j))^T L_x(j:m,i)\right) L_x(j:m,j)$
  end for
end for

This algorithm overwrites L_x with a lower trapezoidal orthonormal basis of span(L_x). The QR factorization of L_x can be also computed using orthogonal transformations, but the modified Gram–Schmidt algorithm is simpler.

3.1. Error analysis. The first and the most important fact used in the analysis is that L_x and L_y are well-conditioned bases for $\mathcal{X}$ and $\mathcal{Y}$. The matrices L_x and L_y are lower trapezoidal with unit diagonal and with off-diagonal elements less than one in modulus. Further, the spectral condition numbers of L_x and L_y are bounded by a function of the dimensions, independent of X, Y. Although the theoretical bound of the condition numbers is an exponential function of the dimension, the values of $\kappa_2(L_x)$ and $\kappa_2(L_y)$ are almost always moderate (cf., e.g., [38], [34], [41]). Hence, we can safely use the modified Gram–Schmidt algorithm to compute nearly orthogonal bases for span(L_x) and span(L_y).

We begin the analysis by pointing out an important difference between the LU and the QR factorizations. Namely, the LU factorization and the backward error from its floating-point computation are, under certain assumptions, invariant under row and column scalings. To simplify the notation, we assume that in Step 1 of Algorithm 3.1 the rows and the columns of X are permuted so that $X \equiv P_1 X P_2 = L_x U_x$ is the LU factorization with complete pivoting. Then the computed matrices $\tilde{L}_x$, $\tilde{U}_x$ satisfy (cf. [26, Theorem 9.3])

$$X + \delta X = \tilde{L}_x \tilde{U}_x, \qquad |\delta X| \le \varepsilon_{LU}(p) |\tilde{L}_x| \cdot |\tilde{U}_x|, \qquad \varepsilon_{LU}(p) \le \frac{p\varepsilon}{1 - p\varepsilon}, \tag{3.1}$$

where the matrix absolute values and inequalities are understood elementwise.

Let $X = D_1 Z D_2$, where D_1 and D_2 are diagonal scalings, and let δZ be defined by the relation

$$X + \delta X = D_1 (Z + \delta Z) D_2, \tag{3.2}$$

that is, $\delta Z = D_1^{-1} \delta X D_2^{-1}$. If $Z = L_z U_z$ and $Z + \delta Z = \tilde{L}_z \tilde{U}_z$ are the LU factorizations, then

$$Z = (D_1^{-1} L_x D_1)(D_1^{-1} U_x D_2^{-1}), \qquad Z + \delta Z = (D_1^{-1} \tilde{L}_x D_1)(D_1^{-1} \tilde{U}_x D_2^{-1}),$$

and, by the uniqueness of the LU factorization,

$$L_z = D_1^{-1} L_x D_1, \qquad U_z = D_1^{-1} U_x D_2^{-1}, \qquad \tilde{L}_z = D_1^{-1} \tilde{L}_x D_1, \qquad \tilde{U}_z = D_1^{-1} \tilde{U}_x D_2^{-1}.$$

Furthermore, from relations (3.1) and (3.2) it follows that

$$\tilde{L}_z \tilde{U}_z = Z + \delta Z, \qquad |\delta Z| \le \varepsilon_{LU}(p) |\tilde{L}_z| \cdot |\tilde{U}_z|. \tag{3.3}$$

Note that $L_x - \tilde{L}_x = D_1 (L_z - \tilde{L}_z) D_1^{-1}$ and that

$$\frac{\|(L_x - \tilde{L}_x) e_i\|_2}{\|L_x e_i\|_2} \le \max_{j>i} \left| \frac{(D_1)_{jj}}{(D_1)_{ii}} \right| \frac{\|(L_z - \tilde{L}_z) e_i\|_2}{\|L_x e_i\|_2} \le \max_{j>i} \left| \frac{(D_1)_{jj}}{(D_1)_{ii}} \right| \|(L_z - \tilde{L}_z) e_i\|_2. \tag{3.4}$$

Hence, if $X \equiv P_1 X P_2$ can be written as $X = D_1 Z D_2$, where the diagonal entries of the diagonal matrix $|D_1|$ are graded from large to small and Z in (3.3) admits an accurate LU factorization with moderate $\|L_z\|_2$, then the computed matrix $\tilde{L}_x$ has columnwise small relative error. (See [13], [16], [19], [33]. Similar analysis applies to

$U_x - \tilde{U}_x = (D_1 D_2) D_2^{-1} (U_z - \tilde{U}_z) D_2$, where we can derive rowwise bounds.) Since $\tilde{L}_x$ is also well-conditioned, its QR factorization can be computed with the modified Gram–Schmidt algorithm.

The numerical properties of the modified Gram–Schmidt algorithm are well understood; see [6], [11], [7]. The two most important facts are summarized in the following theorem due to Higham [26, section 18.7, Theorem 8.12].

THEOREM 3.1. *Let the modified Gram–Schmidt algorithm be applied to* $A \in \mathbf{R}^{m\times p}$ *of* rank p, *and let* $A = A_c \operatorname{diag}(\|Ae_i\|_2)$. *If* $\tilde{Q}$ *and* $\tilde{R}$ *are the computed matrices, then there exist a backward perturbation* δA *and moderate polynomials* $\wp_{MGS}(m,p)$, $\wp'_{MGS}(m,p)$ *such that*

$$A + \delta A = \tilde{Q}\tilde{R}, \qquad \|\delta A e_i\|_2 \le \varepsilon \wp_{MGS}(m,p)\|Ae_i\|_2, \tag{3.5}$$

$$\|\tilde{Q}^T\tilde{Q} - I\|_2 \le \varepsilon \wp'_{MGS}(m,p)\kappa_2(A_c) + O((\varepsilon \wp'_{MGS}(m,p)\kappa_2(A_c))^2). \tag{3.6}$$

The next result describes Algorithm 3.1 in floating-point arithmetic.

THEOREM 3.2. *Let* $\tilde{L}_x = L_x + \delta L_x$, $\tilde{L}_y = L_y + \delta L_y$ *be the computed lower triangular factors in Step* 1 *of Algorithm* 3.1. *Let* $\operatorname{rank}(L_x + \delta L_x) = \operatorname{rank}(L_x)$, $\operatorname{rank}(L_y + \delta L_y) = \operatorname{rank}(L_y)$, *and let*

$$\eta_x \equiv \max_{1\le i\le p} \frac{\|\delta L_x e_i\|_2}{\|L_x e_i\|_2} < 1, \qquad \eta_y \equiv \max_{1\le i\le q} \frac{\|\delta L_y e_i\|_2}{\|L_y e_i\|_2} < 1.$$

Further, let the approximations $\tilde{Q}_x \approx Q_x$, $\tilde{Q}_y \approx Q_y$, *computed in Step* 2, *satisfy*

$$\omega \equiv \max\{\|\tilde{Q}_x^T\tilde{Q}_x - I_p\|_F, \|\tilde{Q}_y^T\tilde{Q}_y - I_q\|_F\} < 1, \tag{3.7}$$

where ω *is derived from Theorem* 3.1. *Then there exist subspaces* $\hat{\mathcal{X}}$, $\hat{\mathcal{Y}}$, *and a moderate function* $f(m,p,q)$ *such that the following hold:*

(i) *The subspaces* $\hat{\mathcal{X}}$ *and* $\hat{\mathcal{Y}}$ *are close approximations of* $\mathcal{X}$ *and* $\mathcal{Y}$, *respectively. More precisely, if we define* $\eta'_y = \eta_y + \varepsilon\wp_{MGS}(m,q)(1+\eta_y)$, *then it holds that*

$$\sin\angle(\mathcal{X},\hat{\mathcal{X}}) \le \sqrt{p}(\eta_x + \varepsilon\wp_{MGS}(m,p)(1+\eta_x))\|(L_x)_c^\dagger\|_2, \tag{3.8}$$

$$\sin\angle(\mathcal{Y},\hat{\mathcal{Y}}) \le \sqrt{q}\left(\eta'_y + \frac{\varepsilon f(m,p,q)(1+\eta'_y)}{(1-\omega)^2}\right)\|(L_y)_c^\dagger\|_2. \tag{3.9}$$

(ii) *If* $\sigma'_1 \ge \cdots \ge \sigma'_q$ *are the exact cosines of the principal angles between* $\hat{\mathcal{X}}$ *and* $\hat{\mathcal{Y}}$, *then, for all* i, *either* $\tilde{\sigma}_i = \sigma'_i = 0$ *or* $|\tilde{\sigma}_i - \sigma'_i|/\sigma'_i$ *is less than* ω *plus* ε *times a moderate polynomial of the space dimension.*

Proof. The floating-point QR factorization of $\tilde{L}_x$ can be represented as $\tilde{Q}_x\tilde{R}_x = \tilde{L}_x + \delta\tilde{L}_x$, where (cf. Theorem 3.1) $\|\delta\tilde{L}_x e_i\|_2 \le \varepsilon\wp_{MGS}(m,p)\|\tilde{L}_x e_i\|_2$, $1 \le i \le p$. Let $\Delta L_x = \delta L_x + \delta\tilde{L}_x$, and let, as in Theorem 2.1, $\tilde{Q}_x = Q'_x(I + T'_x)$ be the QR factorization of $\tilde{Q}_x$. (Q'_x is exactly orthonormal and T'_x is upper triangular with $\|T'_x\|_2 \le \omega$.) Then $L_x + \Delta L_x = Q'_x(I + T'_x)\tilde{R}_x$. Note that $\operatorname{rank}(L_x + \Delta L_x) = \operatorname{rank}(L_x)$. Define $\hat{\mathcal{X}} = \operatorname{span}(L_x + \Delta L_x)$ and note that the sine of the angle between $\mathcal{X}$ and $\hat{\mathcal{X}}$ equals $\sin\angle(\mathcal{X},\hat{\mathcal{X}}) = \|((Q'_x)^\perp)^T Q_x\|_2$, where $(Q'_x)^\perp$ is the orthonormal basis of the orthogonal complement of $\hat{\mathcal{X}}$. An easy calculation shows that

$$((Q'_x)^\perp)^T Q_x = -((Q'_x)^\perp)^T(\Delta L_x)R_x^{-1}, \qquad \sin\angle(\mathcal{X},\hat{\mathcal{X}}) \le \|\Delta L_x R_x^{-1}\|_2.$$

Similarly, we can write $L_y + \Delta' L_y = Q'_y(I + T'_y)\tilde{R}_y$, where $\Delta' L_y = \delta L_y + \delta\tilde{L}_y$. As in the proof of Theorem 2.1, we write $\tilde{S} \equiv \mathrm{fl}(\tilde{Q}_x^T\tilde{Q}_y) = \tilde{Q}_x^T\tilde{Q}_y + E_S$ and we represent

the computed singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_q$ of $\tilde{S}$ as exact singular values of $\tilde{S} + \delta\tilde{S}$, where both $\delta\tilde{S}$ and E_S are small in the spectral norm ($\|E_S\|_2 \ll 1$, $\|\delta\tilde{S}\|_2 \ll 1$). Then we write

$$\begin{aligned}\tilde{S} + \delta\tilde{S} &= \tilde{Q}_x^T\tilde{Q}_y + E_S + \delta\tilde{S} = \tilde{Q}_x^T(\tilde{Q}_y + (\tilde{Q}_x^T)^\dagger(E_S + \delta\tilde{S})) \\ &= (I + T_x')^T\left((Q_x')^T Q_y''\right)(I + T_y''),\end{aligned}$$

where $\tilde{Q}_y + (\tilde{Q}_x^T)^\dagger(E_S + \delta\tilde{S}) = Q_y''(I + T_y'')$ is the QR factorization of an almost orthonormal matrix with $\|T_y''\|_2 \ll 1$. Define

$$L_y + \Delta L_y \equiv L_y + \Delta' L_y + (\tilde{Q}_x^T)^\dagger(E_S + \delta\tilde{S})\tilde{R}_y = Q_y''(I + T_y'')\tilde{R}_y,$$

and $\hat{\mathcal{Y}} = \operatorname{span}(L_y + \Delta L_y)$. The proof is completed by an elementary calculation of the upper bounds for $\|\Delta L_x e_i\|_2/\|L_x e_i\|_2$, $\|\Delta L_y e_j\|_2/\|L_y e_j\|_2$, $1 \leq i \leq p$, $1 \leq j \leq q$, and by comparing the singular values of $\tilde{S} + \delta\tilde{S}$ and $(Q_x')^T Q_y''$, as in the proof of Theorem 2.1. ∎

REMARK 3.1. *The backward error bounds* (3.8) *and* (3.9) *can be improved as follows. Note that it also holds that* $\sin\angle(\mathcal{X}, \hat{\mathcal{X}}) = \|(Q_x^\perp)^T Q_x'\|_2$, *where* $Q_x^\perp$ *is the orthonormal basis of the orthogonal complement of* $\mathcal{X}$. *An easy calculation shows that* $(Q_x^\perp)^T Q_x' = (Q_x^\perp)^T(\Delta L_x)\tilde{R}_x^{-1}(I + T_x')^{-1}$. *Let now* $\Delta L_x = Q_{\Delta_x} R_{\Delta_x}$ *be the QR factorization of* ΔL_x *and let* $\mathcal{L}_{\Delta_x} = \operatorname{span}(\Delta L_x)$. *Then*

$$\begin{aligned}(Q_x^\perp)^T Q_x' &= \left((Q_x^\perp)^T Q_{\Delta_x}\right) R_{\Delta_x}\tilde{R}_x^{-1}(I + T_x')^{-1}, \\ \sin\angle(\mathcal{X}, \hat{\mathcal{X}}) &\leq \sin\angle(\mathcal{X}, \mathcal{L}_{\Delta_x})\frac{\|\Delta L_x \tilde{R}_x^{-1}\|_2}{1 - \|T_x'\|_2}.\end{aligned}$$

Hence, if $\operatorname{span}(\Delta L_x) \subset \mathcal{X}$, *then* $\sin\angle(\mathcal{X}, \hat{\mathcal{X}}) = 0$.

REMARK 3.2. *In this paper, we consider only the classical partial and complete pivoting in the Gaussian elimination. Other choices include, for example, the pivoting for stability and sparsity due to Björck and Duff* [9], *the maximal transversal pivoting due to Olschowka and Neumaier* [27], *and the pivoting for forward stable Gaussian elimination due to Demmel et al.* [13].

3.2. Applications to the QR factorization with complete pivoting. The conclusions about the sensitivity of the LU factorization can be used to understand the high accuracy of the QR factorization with complete pivoting. Recall relation (2.10),

$$X \equiv D' X_s D \longmapsto X + \delta X = D'(X_s + \delta X_s)D, \qquad |(\delta X_s)_{ij}| \leq h(p)\boldsymbol{\varepsilon}\mu_i(\tilde{X}_c')\max_j |(X_s)_{ij}|,$$

and assume that the diagonals of D, D' are graded ($|D_{ii}| \geq |D_{i+1,i+1}|$, $|D_{ii}'| \geq |D_{i+1,i+1}'|$) and that X_s admits an accurate LU factorization in the presence of the perturbation δX_s. (Note that pivoting ensures that D, D' nearly meet the ordering assumption.) In that case, the LU factorization of $X = L_x U_x$ is accurate as well and $\max_i \|\delta L_x e_i\|_2/\|L_x e_i\|_2 \ll 1$. Now note that from $X = L_x U_x = Q_x R_x$ it follows that $L_x = Q_x(R_x U_x^{-1})$ is the QR factorization of L_x. In other words, the orthonormal QR factors of X and L_x are essentially the same (up to the orientation of the columns of Q_x, depending on the signs of the pivots). Similarly, if $X + \delta X = (L_x + \delta L_x)(U_x + \delta U_x) = (Q_x + \delta Q_x)(R_x + \delta R_x)$, then $Q_x + \delta Q_x$ is the orthonormal QR factor of $L_x + \delta L_x$. This means that we can develop a perturbation theory for δQ_x as a function of L_x and δL_x. The good news is that δL_x

is from the columnwise class of perturbations and the relevant condition number is $\min_{D_L=\mathrm{diag}} \kappa_2(L_x D_L)$. This condition number is moderate if the unit lower trapezoidal LU factor of X_s is well-conditioned. In that case, we can derive sharp perturbation estimates for the QR factorization of the perturbed matrix X from relation (2.10). For example, we can prove the following proposition.

PROPOSITION 3.1. *Let $X = Q_x R_x$, $X + \delta X = (Q_x + \delta Q_x)(R_x + \delta R_x)$ be the QR factorizations of X and $X + \delta X$, respectively. Let L_x and $L_x + \delta L_x$ be the unit lower triangular factors of X and $X + \delta X$, and let $(L_x)_c = L_x \,\mathrm{diag}(1/\|L_x e_i\|_2)$, $(\delta L_x)_c = \delta L_x \,\mathrm{diag}(1/\|L_x e_i\|_2)$, $\|(\delta L_x)L_x^\dagger\|_2 < 1/2$. There exists an upper triangular matrix E such that*

$$\left(I + (\delta L_x)L_x^\dagger\right) Q_x = (Q_x + \delta Q_x)(I + E),$$

$$\|E\|_F \le \sqrt{2}\| \left((\delta L_x)L_x^\dagger\right)^T + (\delta L_x)L_x^\dagger + \left((\delta L_x)L_x^\dagger\right)^T \left((\delta L_x)L_x^\dagger\right) \|_F, \text{ and}$$

$$\|\delta Q_x\|_F \le \frac{\|(\delta L_x)L_x^\dagger\|_F + \|E\|_F}{1 - \|E\|_2}.$$

Furthermore, $\sin\angle(\mathrm{span}(Q_x + \delta Q_x), \mathrm{span}(X)) \le \|(L_x)_c^\dagger\|_2 \|(\delta L_x)_c\|_2$.

3.3. Comments on the SVD computation in principal angle algorithms. In the proofs of Theorem 2.1 and Theorem 3.2 we do not explicitly mention which algorithm is used to compute the SVD of the matrix $\tilde{S} = \mathrm{fl}(\tilde{Q}_x^T \tilde{Q}_y) = \tilde{Q}_x^T \tilde{Q}_y + E_S$. We only use the fact that the algorithm is backward stable where the backward error $\delta\tilde{S}$ is small in the spectral matrix norm, $\|\delta\tilde{S}\|_2 \le f(p,q)\boldsymbol{\varepsilon}\|\tilde{S}\|_2$. If we use the Jacobi SVD algorithm, then we can estimate $\delta\tilde{S}$ by (cf. [18]) $\|\delta\tilde{S}e_i\|_2 \le g(p,q)\boldsymbol{\varepsilon}\|\tilde{S}e_i\|_2$, $1 \le i \le q$, where $g(p,q)$ is a modest polynomial. Here, the backward error in small columns is correspondingly small and the Jacobi SVD algorithm computes small singular values with higher relative accuracy. However, this higher accuracy of the Jacobi SVD algorithm does not improve the overall accuracy. If for some j the column $\tilde{S}e_j$ is small (of order $\boldsymbol{\varepsilon}$, say), then $\tilde{S}e_j$ is generally accurate to an absolute uncertainty of order $m\boldsymbol{\varepsilon}$ ($\|E_S e_j\|_2 \le O(m\boldsymbol{\varepsilon})$) and the relative error in that column might be large. In that case, even an exact SVD computation would not be able to compute the singular values of $\tilde{S}$ with a high relative accuracy. Hence, the SVD algorithm of choice is the fastest algorithm which ensures small $\|\delta\tilde{S}\|_2/\|\tilde{S}\|_2$.

Example 3.1. We illustrate the above discussion in the case $p = q = 1$. Let $X = [x]$, $Y = [y]$, where $x, y \in \mathbf{R}^m$ are unit vectors, and let δx, δy be small perturbations ($\|\delta x\|_2 \ll 1$, $\|\delta y\|_2 \ll 1$). Then $\angle(\mathcal{X}, \mathcal{Y}) = \arccos(x^T y)$ and $|(x+\delta x)^T(y+\delta y) - x^T y| \le \|\delta x\|_2 + \|\delta y\|_2 + \|\delta x\|_2\|\delta y\|_2$. Hence, the relative accuracy of the singular value $\sigma_1 = x^T y$ in the presence of errors δx and δy is determined by $(\|\delta x\|_2 + \|\delta y\|_2)/(x^T y)$. If $\|\delta x\|_2$ and $\|\delta y\|_2$ are of the order of the machine precision $\boldsymbol{\varepsilon}$, then we see that floating-point computation of σ_1 is feasible to only (roughly) $-\lfloor \log_{10}(\boldsymbol{\varepsilon}/(x^T y))\rfloor$ decimal places. In other words, small singular values are poorly determined if the corresponding subspaces are nearly orthogonal (cf. [24]).

REMARK 3.3. *If X and Y are normally scaled ($p = q$, $X^T Y = I_p$), then the canonical correlations of X and Y are the singular values of XY^T. In that case the canonical correlations are determined to high relative accuracy if the condition numbers $\kappa_2(X_c)$, $\kappa_2(Y_c)$ of column scaled matrices X, Y are moderate (cf.* [24]*). For accurate SVD computation of XY^T, see* [16], [17].

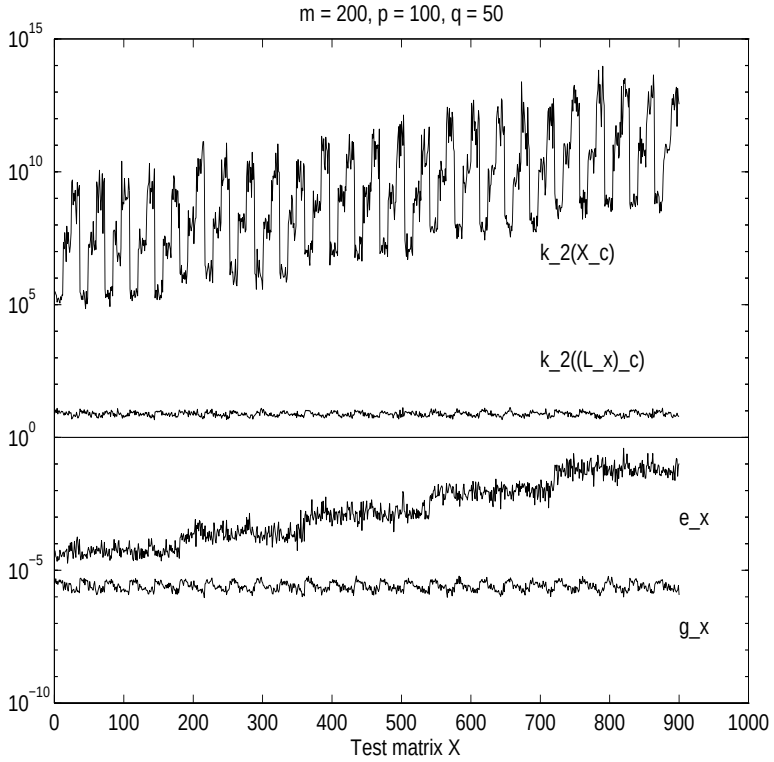


FIG. 4.1. *The values of* e_x, $\mathbf{g}_x$, $\kappa_2(X_c)$, $\kappa_2((\tilde{L}_x)_c)$ *for* 900 *test in Example* 4.1. *The LU factorizations are computed with complete pivoting. Similar results are obtained if the LU factorizations are computed with the partial (row) pivoting.*

3.4. Cross-product implementation. In this subsection, we briefly discuss an implementation of Algorithm 3.1 that may be an efficient alternative in the case $m \gg \max\{p, q\}$.

ALGORITHM 3.3. X–CC(X, Y).

Input $X \in \mathbf{R}^{m\times p}$, $Y \in \mathbf{R}^{m\times q}$ full column rank matrices with $p \geq q$.

Step 1 Compute the LU factorizations with pivoting, $P_1 X P_2 = L_x U_x$, $P_3 Y P_4 = L_y U_y$. (For partial pivoting, $P_2 = I_p$, $P_4 = I_q$.)

Step 2 Compute the matrices $H_{xx} = L_x^T L_x$, $H_{yy} = L_y^T L_y$, $H_{xy} = L_x^T((P_1 P_3^T) L_y)$, and the Cholesky factorizations $H_{xx} = R_x^T R_x$, $H_{yy} = R_y^T R_y$. Exploit symmetry as much as possible.

Step 3 Compute the matrix $S = R_x^{-T} H_{xy} R_y^{-1}$ and the SVD of S, $S = W\Sigma V^T$.

Output Return the matrix Σ.

The use of the Cholesky factors of the cross-product matrices is similar to the Peters–Wilkinson [28] algorithm for least squares solution using the normal equations. (Recall that the principal angle problem in the case $q = 1$ is closely related to the classical least squares problem; cf. [10]. Also note that in the case of sparse X and Y we may use the complete pivoting of Björck and Duff [9] which is designed to preserve as much of the original sparsity as possible. For related results see also Barlow [3] and Barlow and Handy [4].) Perturbation analysis of Algorithm 3.3 can be done using a technique from [17]. We omit the details for the sake of brevity.

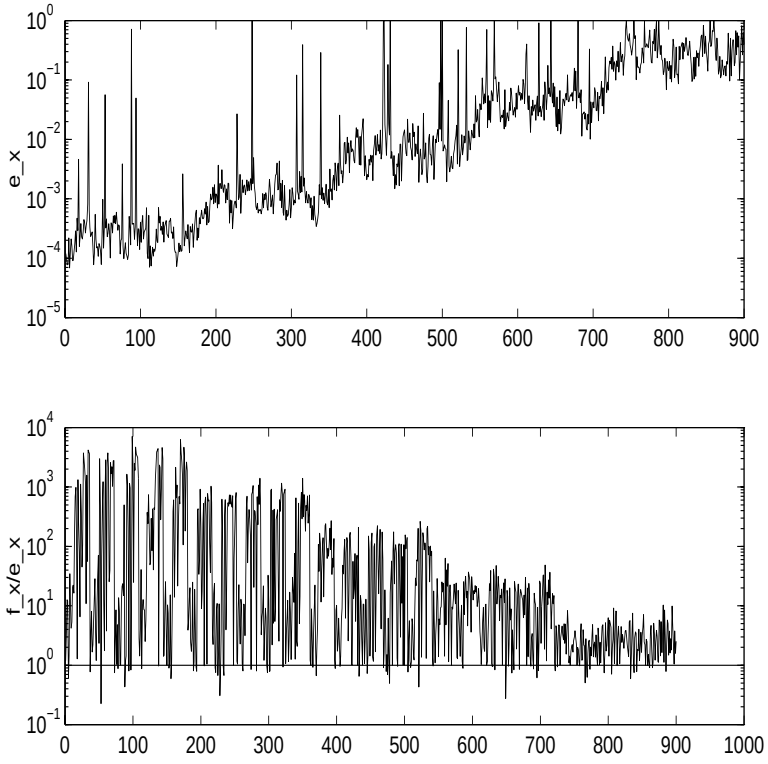


FIG. 4.2. *The values of e_x and f_x/e_x for all* 900 *matrices $\{X\}$ in Example* 4.2. *The LU and the QR factorizations are computed with complete pivoting. Note that in most cases $f_x \gg e_x$.*

4. Numerical examples. We conclude this work with several numerical examples. We show that complete pivoting in the QR factorization improves the accuracy of the Björck–Golub algorithm and that in some cases Gaussian elimination with pivoting can be used as an accurate preconditioner for the QR factorization.

Example 4.1. In this example, we generate test matrices X as in Example 2.1, and we measure the errors in the computed orthonormal bases $\tilde{Q}_x$ of $\mathcal{X} = \mathrm{span}(X)$. For each generated matrix X we compute $\kappa_2(X_c)$ ($X_c = X\,\mathrm{diag}(1/\|Xe_i\|_2)$), $\kappa_2((\tilde{L}_x)_c)$, where $\tilde{L}_x$ is the computed lower trapezoidal factor of X and $(\tilde{L}_x)_c = \tilde{L}_x\,\mathrm{diag}(1/\|\tilde{L}_x e_i\|_2)$, and

$$e_x = \sin\angle(\mathrm{span}(\tilde{Q}_x), \mathcal{X}), \qquad g_x = \|\tilde{Q}_x^T\tilde{Q}_x - I_p\|_2.$$

The results for all 900 values of X are given in Figure 4.1. Recall that the test matrices $\{X \equiv D'X_sD\}$ are divided into five classes (180 examples each) with fixed $\kappa_2(X_s) = 10^2, 10^3, 10^4, 10^5, 10^6$. These classes are clearly recognizable in Figure 4.1 if one follows the growth of e_x. The accuracy is determined by $\kappa_2(X_s)$ and not by $\kappa_2(X_c)$. Also note that the deviation from orthonormality of $\tilde{Q}_x$ is of the order of $m\boldsymbol{\varepsilon}$ and that $\kappa_2((\tilde{L}_x)_c) \approx O(1)$. A similar accuracy is observed in a variant of Algorithm 3.1 with LU factorizations with the partial (row) pivoting.

Example 4.2. In this example, we generate a set of rather ill-conditioned bases. We first generate an X as in Example 2.1, and then we partition X as $X = [X_1, X_2]$, and we introduce heavy weighting into the rows of X_2. Both Algorithm 3.1 and the Björck–

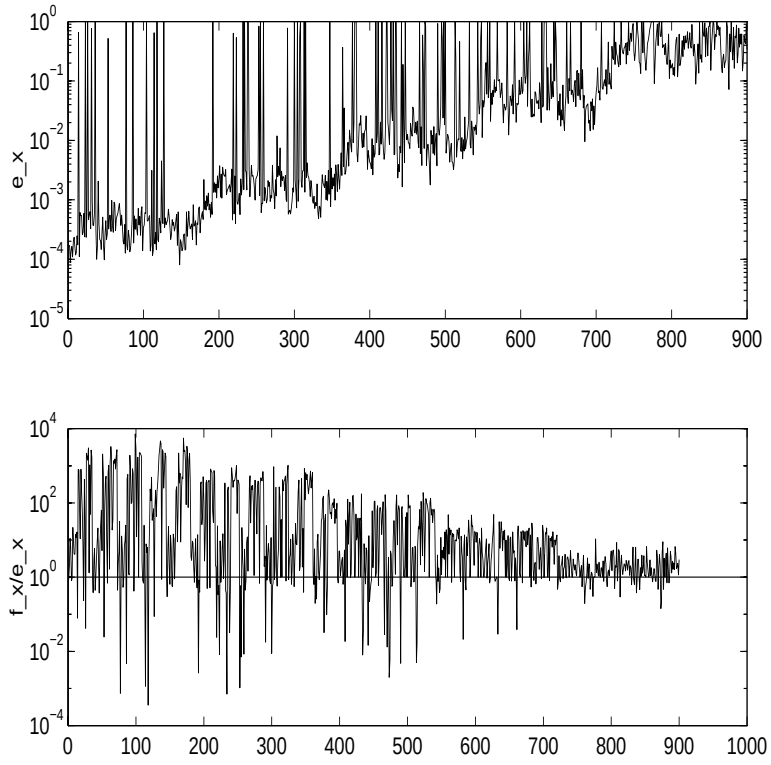


FIG. 4.3. *The values of e_x and f_x/e_x for all matrices $\{X\}$ in Example* 4.2. *The LU factorizations are computed with partial pivoting and the QR factorizations are computed with complete pivoting.*

Golub algorithm are sensitive to ill-conditioning introduced in this way. However, Algorithm 3.1 retains its accuracy properties in most of the cases, while the QR-based approach results in much larger errors. In Figure 4.2, e_x is defined as in Example 4.1 ($\tilde{Q}_x$ computed by Algorithm 3.1) and $f_x = \sin\angle(\mathrm{span}(\tilde{\tilde{Q}}_x), \mathcal{X})$, where $\tilde{\tilde{Q}}_x$ is computed by the QR factorization with complete pivoting. The variant of Algorithm 3.1 with partial pivoting is also less accurate; see Figure 4.3.

Example 4.3. Examples where Algorithm 3.1 is guaranteed to achieve high accuracy include structured matrices where various combinatorial and algebraic conditions (sparsity, sign pattern) ensure forward stable Gaussian elimination with complete pivoting. These include Cauchy and Vandermonde matrices and many others; see [13]. (For further references on highly accurate Gaussian elimination, see [26, *Chapter* 9].) In such cases, Algorithm 3.1 has an advantage over the straightforward use of orthogonal QR factorization.

Example 4.4. In this example, we measure the forward error in the computed canonical correlations. As reference values we use the approximate canonical correlations $\sigma_1^{(D)} \geq \cdots \geq \sigma_q^{(D)}$ computed by the double precision Algorithm 3.1. The test problems are generated as in Example 4.1. We test the accuracy of the Björck–Golub algorithm with complete pivoting, Algorithm 3.1 with complete and partial pivoting, and Algorithm 3.3 with complete pivoting. For single precision approximations

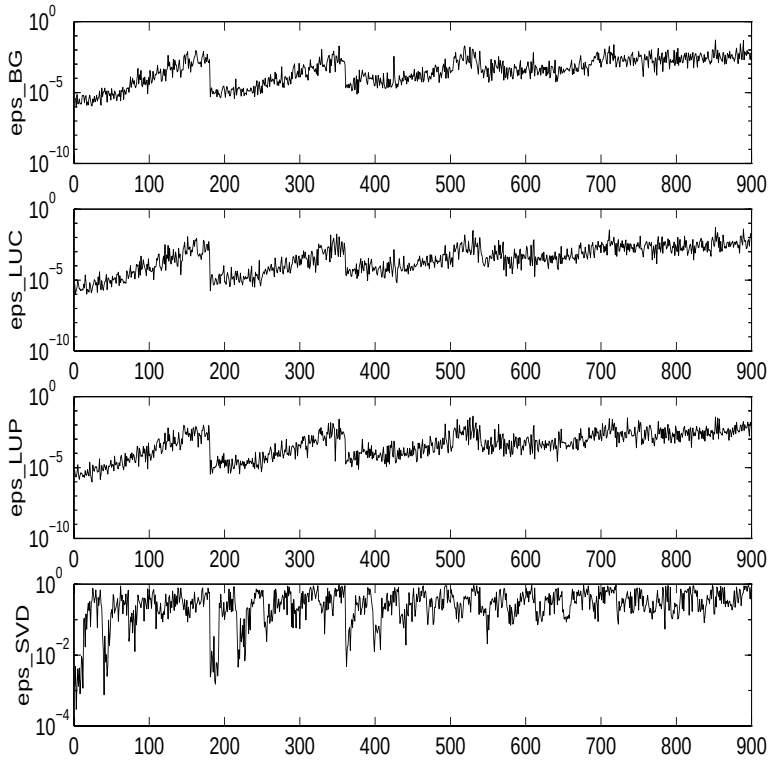


FIG. 4.4. *The computed forward errors in Example* 4.5.

$\sigma_1^{(S)} \geq \cdots \geq \sigma_q^{(S)}$ computed by each of the four algorithms, we compute

$$\epsilon_{CC} = \frac{\max_{1\leq i\leq q} |\sigma_i^{(D)} - \sigma_i^{(S)}|}{\max\{\kappa_2(X_s), \kappa_2(Y_s)\}}.$$

The expected values of ϵ_{CC} are of order of the machine precision ε. The maximal computed values of ϵ_{CC} for all four algorithms are between $2.6 \cdot 10^{-7}$ and $3.5 \cdot 10^{-7}$ which, in this example, shows nearly the same accuracy. A high performance implementation of the QR factorization with complete pivoting (using [12] and [30]) is the method of choice in this example.

Example 4.5. In our last example, we compare the algorithms based on the QR and the LU factorizations with pivoting with the algorithm based on the use of the SVD in the computation of the orthonormal bases for span(X), span(Y). (The use of the SVD in the principal angle computation is discussed in [10] in connection with ill-conditioned and rank deficient cases.) In this example, we compute the SVD using the *LAPACK* procedure `SGESVD()`. The test is performed as in Example 4.4 and with the dimensions $m = 100$, $p = q = 50$. For each of the 900 examples, we compute the maximal forward errors ϵ_{BG} (for the Björck–Golub algorithm with complete pivoting), ϵ_{LUC} (for Algorithm 3.1 with complete pivoting), ϵ_{LUP} (for Algorithm 3.1 with partial pivoting), and ϵ_{SVD} for the computation based on the SVD. The results shown in Figure 4.4 show that the SVD approach is less accurate than the QR- and LU-based algorithms with complete pivoting. The reason is the high sensitivity of the bidiagonalization based SVD algorithms to differently scaled matrix columns and/or

rows. (We conjecture that a similar situation occurs in the weighted least squares computation if we compare the Peters–Wilkinson algorithm, the QR approach with complete pivoting, and the algorithm based on a SVD procedure that is sensitive to column and row weighting.)

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide,* 2nd ed., SIAM, Philadelphia, PA, 1995.

[2] E. H. Bareiss, *Numerical Solution of the Weighted Linear Least Squares Problem by G-Transformations*, Technical Report 82–03–NAM–03, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 1982.

[3] J. Barlow, *Stability analysis of the G-algorithm and a note on its application to sparse least squares problems*, BIT, 25 (1985), pp. 507–520.

[4] J. Barlow and S. Handy, *The direct solution of weighted and equality constrained least-squares problems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 704–716.

[5] R. Bhatia and K. Mukherjea, *Variation of the unitary part of a matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1007–1014.

[6] A. Björck, *Solving linear least squares problems by Gram–Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[7] Å. Björck, *Numerics of Gram–Schmidt orthogonalization*, Linear Algebra Appl., 197/198 (1994), pp. 297–316.

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

[9] Å. Björck and I. S. Duff, *A direct method for the solution of sparse linear least squares*, Linear Algebra Appl., 34 (1980), pp. 43–67.

[10] Å. Björck and G. H. Golub, *Numerical methods for computing angles between linear subspaces*, Math. Comp., 27 (1973), pp. 579–594.

[11] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[12] A. J. Cox and N. J. Higham, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, Pitman Res. Notes Math. Ser. 380, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., Addison-Wesley-Longman, Harlow, Essex, UK, 1998, pp. 57–73.

[13] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.

[14] J. Demmel and A. McKenney, *A Test Matrix Generation Suite*, LAPACK Working Note 9, Courant Institute, New York, 1989.

[15] Z. Drmač, *Computing the Singular and the Generalized Singular Values*, Ph.D. thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Hagen, Germany, 1994.

[16] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.

[17] Z. Drmač, *New accurate algorithms for singular value decomposition of matrix triplets*, SIAM J. Matrix Anal. Appl., to appear.

[18] Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal., 35 (1998), pp. 1804–1832.

[19] Z. Drmač and E. R. Jessup, *On Accurate Generalized Singular Value Computation in Floating-Point Arithmetic*, Department of Computer Science, University of Colorado at Boulder; SIAM J. Matrix Anal. Appl., submitted.

[20] Z. Drmač, M. Omladič, and K. Veselić, *On the perturbation of the Cholesky factorization*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1319–1332.

[21] S. Eisenstat and I. Ipsen, *Relative perturbation techniques for singular value problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1972–1988.

[22] G. H. Golub, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[24] G. H. Golub and H. Zha, *Perturbation analysis of the canonical correlations of matrix pairs*, Linear Algebra Appl., 210 (1994), pp. 3–28.

[25] G. H. Golub and H. Zha, *The canonical correlations of matrix pairs and their numerical computation*, in Linear Algebra for Signal Processing, IMA Vol. Math. Appl., A. Bojanczyk and G. Cybenko, eds., Springer, New York, 1995, pp. 27–49.
[26] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
[27] M. Olschowka and A. Neumaier, *A new pivoting strategy for Gaussian elimination*, Linear Algebra Appl., 240 (1996), pp. 131–151.
[28] G. Peters and J. H. Wilkinson, *The least squares problem and pseudoinverses*, Comput. J., 13 (1970), pp. 309–316.
[29] M. J. D. Powell and J. K. Reid, *On applying Householder transformations to linear least squares problems*, in Information Processing 68, Proc. International Federation of Information Processing Congress, Edinburgh, 1968, North-Holland, Amsterdam, 1969, pp. 122–126.
[30] G. Quintana-Orti, X. Sun, and C. H. Bischof, *A BLAS 3 Version of the QR Factorization with Column Pivoting*, Argonne Preprint MCS–P551–1295 and PRISM Working Note 26, Argonne National Laboratory, Argonne, IL, 1990.
[31] G. W. Stewart, *Perturbation bounds for the QR decomposition of a matrix*, SIAM J. Numer. Anal., 14 (1977), pp. 509–518.
[32] G. W. Stewart, *On the asymptotic behavior of scaled singular value and QR decompositions*, Math. Comp, 43 (1984), pp. 483–489.
[33] G. W. Stewart, *On the Perturbation of LU and Cholesky Factors*, Technical Report TR–3535, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1995.
[34] G. W. Stewart, *The Triangular Matrices of Gaussian Elimination and Related Decompositions*, Technical Report TR–3533, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1995.
[35] J.-G. Sun, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.
[36] J.-G. Sun, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.
[37] J.-G. Sun, *On perturbation bounds for the QR factorization*, Linear Algebra Appl., 215 (1995), pp. 95–111.
[38] L. N. Trefethen and R. Schreiber, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.
[39] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[40] C. F. Van Loan, *A generalized SVD analysis of some weighting methods for equality constrained least squares*, in Matrix Pencils, B. Kagstrom and A. Ruhe, eds., Lecture Notes in Math. 973, Springer-Verlag, New York, 1983, pp. 245–262.
[41] D. Viswanath and L. N. Trefethen, *Condition Numbers of Random Triangular Matrices*, http://simon.cs.cornell.edu/home/lnt/, 1996.
[42] P. A. Wedin, *On angles between subspaces of a finite dimensional inner product space*, in Matrix Pencils, B. Kagstrom and A. Ruhe, eds., Lecture Notes in Math. 973, Springer-Verlag, New York, 1983, pp. 263–285.
[43] H. Zha, *A componentwise perturbation analysis of the QR decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1124–1131.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 195–212

ON GMRES-EQUIVALENT BOUNDED OPERATORS*

LEONID KNIZHNERMAN†

Abstract. Given a bounded linear operator A in a Hilbert space $\mathcal{H}$ and a nonzero vector $\mathfrak{r} \in \mathcal{H}$, we construct a unitary operator U and (under some conditions) bounded self-adjoint operators P and T (nonnegative definite and indefinite, respectively) such that all the residual Krylov subspaces of $(A, \mathfrak{r})$, $(U, \mathfrak{r})$, $(P, \mathfrak{r})$, and $(T, \mathfrak{r})$ of the same dimension for the equation $Ax = \mathfrak{r}$ are equal. When possible (for example, for U and P, provided 0 is outside the field of values of A), we estimate a gap in the spectrum of U and the condition numbers of P and T. Some attainability results are also established.

It is shown that some analogous matrix assertions are valid, which can be obtained by means of degenerating the operator case. Numerical examples are presented for the finite-dimensional case.

Key words. GMRES, Arnoldi's method, orthonormal polynomials, bounded operators, Hilbert space, matrices

AMS subject classifications. 65F10, 47A99

PII. S0895479897325748

1. Introduction.

GMRES [17] is a popular Arnoldi-based method for solving systems of linear equations $A\mathfrak{x} = \mathfrak{r}$. At a kth step of Arnoldi's algorithm [4] with A and $\mathfrak{r}$, GMRES produces the approximate solution which minimizes the residual norm $\|A\mathfrak{t} - \mathfrak{r}\|$ over the k-dimensional Krylov subspace span $\{A^0\mathfrak{r}, A^1\mathfrak{r}, \dots, A^{k-1}\mathfrak{r}\}$.

To simplify theoretical investigation of GMRES, Greenbaum and Strakoš proposed in [9] to study the GMRES process with $(B, \mathfrak{r})$ instead of $(A, \mathfrak{r})$, where B is a matrix generating the same residual subspaces as A, but having a simpler structure. Greenbaum and Strakoš showed that one can take a unitary matrix as B and that if 0 is outside the field of values of A, then B can be taken Hermitian positive definite.[1]

Their constructions are based on the following assertion. Let n be the dimension of the space, let $(\mathfrak{w}_1, \dots, \mathfrak{w}_k)$ be an orthonormal basis of span $\{A^1\mathfrak{r}, \dots, A^k\mathfrak{r}\}$ for $k = 1, \dots, n$, and let one consider representations of (finite-dimensional) operators in the basis $(\mathfrak{w}_1, \dots, \mathfrak{w}_n)$. There exists a Hessenberg matrix H such that $AW = WH$, and operators, generating the same residual subspaces with respect to $\mathfrak{r}$ as A, have the form

$$RH, \qquad R \text{ is upper triangular.} \tag{1.1}$$

Since the behavior of GMRES applied to a normal matrix can be well described in terms of the (discrete) spectral measure, Greenbaum and Strakoš tried to find normal matrices of sort (1.1). Evidently, if $H = R_1 N$ with $R_1 = R^{-1}$ upper triangular (the inverse of that in (1.1)) and N normal, then N is a desirable normal matrix such that $GMRES(A, \mathfrak{r}) = GMRES(N, \mathfrak{r})$.

For example, considering an RQ-decomposition of H, i.e., $H = R_1 Q$ with upper triangular R_1 and unitary Q, they constructed an equivalent unitary matrix $N = Q$.

*Received by the editors August 21, 1997; accepted for publication (in revised form) by A. Greenbaum December 20, 1999; published electronically May 31, 2000.
http://www.siam.org/journals/simax/22-1/32574.html

†Central Geophysical Expedition, Narodnogo Opolcheniya St., House 40, Bldg. 3, Moscow 123298, Russia (mmd@cge.ru).

[1] Under a special condition, they also built an equivalent Hermitian matrix, not necessarily positive definite.

Provided 0 is outside the numerical range of A, they considered the decomposition $H = UL = (UL^{-*})(L^*L)$ with upper triangular U and lower triangular L, which gave tridiagonal Hermitian $N = L^*L$.

Greenbaum and Strakoš also listed a few open questions; e.g., they did not bound a possible gap in the spectrum of the unitary matrix and the condition number of the Hermitian one. It was proved in [10] that any nonincreasing convergence curve (of a finite length) is attained in GMRES at a matrix with any desired eigenvalues; the authors concluded that eigenvalues are not the relevant quantities in determining the behavior of GMRES.

The aim of this paper is partially analogous to that of [9] and [10], but we go further: we try to estimate the mentioned spectral gap and condition numbers. We mainly work with bounded operators in Hilbert spaces instead of matrices and use the technique of linear operators [2] and orthonormal polynomials [18, 14]. We consider the finite-dimensional (i.e., matrix) case as degenerated one.

Positive points and some difficulties in working with infinite spectra by means of potential theory are described in detail in [6]. Also, it is worth mentioning that [13] is entirely devoted to bounded operators in Krylov subspace processes. Polynomials, orthonormal on the unit circumference, were earlier exploited in [1, 7].

In section 2 we construct a special orthonormal "basis"[2] $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$ of the Hilbert space in terms of which ("basis") we shall define some operators. A similar (finite) construction can be found in [9, 10, 3]; it is just briefly described. However, the normalization of vectors is important for our considerations, so we fix a particular normalization from the variety suggested in [9, 10], and we cannot omit some details.[3] We mention that a particular normalization is also fixed in [12], where some results "dual" to our own are established.

In section 3 we build a GMRES-equivalent (i.e., generating the same infinite sequence of the residual Krylov subspaces) unitary operator U. We prove that 1 is not an eigenvalue of U and, moreover, under some condition we bound a gap in the spectrum of U around 1 on the unit circumference; if 0 lies outside the closure of the numerical range (= field of values) of A, the condition is satisfied. Further, we prove that any *infinite* nonincreasing and asymptotically vanishing residual sequence is attained at a unitary operator; a consequence is derived for the full orthogonalization method (FOM). Note that we cannot replace a unitary operator with one, having a prescribed spectrum, in the last result: unlike the spectrum of a matrix, the spectrum of an operator determines the worst possible (when $\mathfrak{r}$ varies) asymptotical speed of convergence of GMRES [13, Theorem 3.4.9] and an analogous upper bound for the spectral Arnoldi decomposition method (SADM) [11, Theorem 1] in terms of the generalized Green function of the spectrum.[4]

In section 4, under some condition, we construct a GMRES-equivalent bounded self-adjoint injective[5] nonnegative definite operator P; under a stronger condition, we bound the condition number of P; this is the case if 0 is outside the closure of the numerical range of A.

In section 5 we analogously construct an indefinite self-adjoint equivalent operator.

---

[2] We shall use the term *basis* in its usual linear algebra sense without quotes and in the sense of Hilbert spaces (a countable set with a dense envelope) in quotes.

[3] A few related points will be italicized in the text.

[4] Rigorously speaking, we should add to the spectrum all points separated by it from infinity.

[5] That is, one having a trivial kernel.

In section 6 we formulate and shortly prove results for matrices. We show that the matrix theorems are trivial consequences of the analogous operator assertions. The derivation is mainly performed by means of cutting off infinite series. A few figures illustrating the proved theorems are drawn.

In section 7 we list some questions which still remain open.

Calligraphic letters denote vector spaces, capital roman letters denote operators and matrices, small letters of the font Euler Fraktur denote vectors, small roman letters denote complex scalars, Greek letters denote functions and measures. The symbol $\uparrow\downarrow$ denotes collinearity of vectors, $\equiv$ equality by definition, $\mathcal{K}^m$ the mth Krylov subspace, span the linear envelope, and the angular brackets $\langle,\rangle$ the scalar product. Notation introduced in proofs is local.

2. Constructing an orthonormal "basis". Let A be a bounded injective linear operator in a Hilbert space $\mathcal{H}$ and $\mathfrak{r}$ be a vector in $\mathcal{H}\backslash\{0\}$. We shall consider the GMRES process [17] with A and $\mathfrak{r}$. Set $\mathfrak{q} = A\mathfrak{r}$. Excluding trivial cases and without loss of generality, we can reckon that the union of the Krylov subspaces $\cup_{k=0}^{\infty}\mathcal{K}^k(A,\mathfrak{r})$ is dense in $\mathcal{H}$ and that $\dim\mathcal{K}^k(A,\mathfrak{r}) = \dim\mathcal{K}^k(A,\mathfrak{q}) = k$.

Recall that for each $k \in \mathbf{N}$ GMRES finds the vector $\mathfrak{t}_k \in \mathcal{K}^k(A,\mathfrak{r})$ realizing $\min_{\mathfrak{x}_k\in\mathcal{K}^k(A,\mathfrak{r})}\|\mathfrak{r} - A\mathfrak{x}_k\|$. It follows from the minimality (see [2, section 7]) that the residues $\mathfrak{r}_k = \mathfrak{r} - A\mathfrak{t}_k$ satisfy the orthogonality property

$$\mathfrak{r}_k \perp A\mathcal{K}^k(A,\mathfrak{r}) = \mathcal{K}^k(A,\mathfrak{q}). \tag{2.1}$$

Let us construct a special orthonormal "basis," in terms of which we shall later define some operators in $\mathcal{H}$. Let the Arnoldi process with A and $\mathfrak{q}$ produce Arnoldi's vectors $\mathfrak{w}_1$, $\mathfrak{w}_2$,.... As, due to (2.1), $\mathcal{K}^{k+1}(A,\mathfrak{q}) \ni \mathfrak{r}_k - \mathfrak{r}_{k+1} \perp \mathcal{K}^k(A,\mathfrak{q})$, we have $\mathfrak{r}_k - \mathfrak{r}_{k+1} \uparrow\downarrow \mathfrak{w}_{k+1}$ or $\mathfrak{r}_k - \mathfrak{r}_{k+1} = 0$. Putting $g_{k+1} \equiv \|\mathfrak{r}_k - \mathfrak{r}_{k+1}\|$, *renormalize* $\mathfrak{w}_{k+1}$ *so that* $\mathfrak{r}_k - \mathfrak{r}_{k+1} = g_{k+1}\mathfrak{w}_{k+1}$ ($k \in \mathbf{N}$). So, the desirable "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$ has been obtained.

It follows from the density of $\cup_{k=0}^{\infty}\mathcal{K}^k(A,\mathfrak{r})$ in $\mathcal{H}$ that

$$\mathfrak{r}_k \to 0 \quad \text{as} \quad k \to \infty. \tag{2.2}$$

We can decompose a residue in $\{\mathfrak{w}_i\}$. Really, for $k \in \mathbf{N}$,

$$\sum_{l=k+1}^{\infty} g_l\mathfrak{w}_l = \lim_{m\to\infty}\sum_{l=k+1}^{k+m} g_l\mathfrak{w}_l = \lim_{m\to\infty}\sum_{l=k+1}^{k+m}(\mathfrak{r}_{l-1} - \mathfrak{r}_l) = \mathfrak{r}_k - \lim_{m\to\infty}\mathfrak{r}_{k+m} = \mathfrak{r}_k$$

owing to (2.2). In particular,

$$\mathfrak{r} = \mathfrak{r}_0 = \sum_{l=1}^{\infty} g_l\mathfrak{w}_l. \tag{2.3}$$

Similarly to [10], denote

$$f_k \equiv \|\mathfrak{r}_k\| = \sqrt{\sum_{l=k+1}^{\infty} g_l^2}.$$

Slightly modifying a definition from [9, section 3], we shall write $GMRES(A,\mathfrak{r}) = GMRES(B,\mathfrak{r})$, where B is a bounded linear operator in $\mathcal{H}$, if $A\mathcal{K}^k(A,\mathfrak{r}) = B\mathcal{K}^k(B,\mathfrak{r})$, $k \in \mathbf{N}$.

LEMMA 2.1. *If a bounded linear operator B in $\mathcal{H}$ is presented by an unreduced upper Hessenberg infinite matrix in the "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$ and $B\mathfrak{r} \uparrow\downarrow \mathfrak{w}_1$, then GMRES$(A, \mathfrak{r})$ = GMRES$(B, \mathfrak{r})$.*

Proof. We have

$$B\mathcal{K}^k(B, \mathfrak{r}) = \mathcal{K}^k(B, \mathfrak{w}_1) = \operatorname{span}\{\mathfrak{w}_1, \ldots, \mathfrak{w}_k\} = A\mathcal{K}^k(A, \mathfrak{r})$$

due to the Hessenbergness of B and by the construction of the "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$. □

3. A GMRES-equivalent unitary operator. Define the numbers

$$a_n = \|\mathfrak{r}\|/f_n, \qquad b_n = \sqrt{a_n^2 - a_{n-1}^2} \qquad (n \in \mathbf{N},\ a_{-1} \equiv 0) \tag{3.1}$$

and the infinite matrix U whose $(n+1)$st column $(n \in \mathbf{N})$ is

$$\left(-\frac{b_{n+1}}{a_n a_{n+1}} \cdot b_0 \quad -\frac{b_{n+1}}{a_n a_{n+1}} \cdot b_1 \quad \ldots \quad -\frac{b_{n+1}}{a_n a_{n+1}} \cdot b_n \quad \frac{a_n}{a_{n+1}} \quad 0 \ldots \right)^T. \tag{3.2}$$

We shall see later that the components of (3.2) are the coefficients in the recurrence for some system of polynomials with *positive leading coefficients* orthonormal on the unit circumference and that the numbers a_n and b_n are the coefficients in the recurrences (3.8) and (3.9) for those orthonormal polynomials and the dual ones.

One can derive from (3.1) that columns (3.2) form an orthonormal system of vectors in l_2. Since $\sum_{k=0}^n b_k^2 = a_n^2$, the squared norm of the $(n+1)$st column equals $\frac{b_{n+1}^2 + a_n^2}{a_{n+1}^2} = 1$ and the scalar product of the $(m+1)$st and $(n+1)$st columns $(m > n)$ is

$$\frac{b_{m+1}}{a_m a_{m+1}} \cdot \frac{b_{n+1}}{a_n a_{n+1}} \cdot a_n^2 - \frac{b_{n+1} b_{m+1}}{a_m a_{m+1}} \cdot \frac{a_n}{a_{n+1}} = 0.$$

Now, it follows from [2, section 26, the second theorem] that U presents a bounded operator in $\mathcal{H}$ if one refers U to the "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$. Moreover, $U^*U = I$, so U is unitary.

We shall identify the matrix U and the correspondent operator.

THEOREM 3.1. *The assertion GMRES$(A, \mathfrak{r})$ = GMRES$(U, \mathfrak{r})$ holds for the unitary operator U just constructed.*

Proof. Compute

$$U^{-1}\mathfrak{w}_1 = U^*\mathfrak{w}_1 = -\sum_{n=1}^{\infty} \frac{b_n}{a_{n-1}a_n} \mathfrak{w}_n.$$

As, due to *the nonnegativity of* g_n,

$$\frac{b_n}{a_{n-1}a_n} = \sqrt{\frac{a_n^2 - a_{n-1}^2}{a_{n-1}^2 a_n^2}} = \sqrt{a_{n-1}^{-2} - a_n^{-2}} = \|\mathfrak{r}\|^{-1} g_n, \tag{3.3}$$

we deduce by virtue of (2.3) $U^{-1}\mathfrak{w}_1 \uparrow\downarrow \mathfrak{r}$, whence $U\mathfrak{r} \uparrow\downarrow \mathfrak{w}_1$. It remains to apply Lemma 2.1. □

There exists a positive bounded measure σ on the unit circumference such that

$$\left\langle U^k \mathfrak{w}_1, \mathfrak{w}_1 \right\rangle = \frac{1}{2\pi} \int_{|z|=1} z^k d\sigma(z), \qquad k \in \mathbf{Z} \tag{3.4}$$

(see [2, section 62]). The measure σ may be considered as the spectral measure of the normal operator U (see [16, chap. 13, section 33]). It follows from (3.4) and the unitariness of U that

$$\left\langle U^k \mathfrak{w}_1, U^l \mathfrak{w}_1 \right\rangle = \left\langle U^{k-l} \mathfrak{w}_1, \mathfrak{w}_1 \right\rangle = \frac{1}{2\pi} \int_{|z|=1} z^k \bar{z}^l d\sigma(z), \qquad k, l \in \mathbf{Z}. \tag{3.5}$$

Extending (3.5) by linearity, we can define the following scalar product on $\mathbf{C}[z]$:

$$\langle \alpha, \beta \rangle = \frac{1}{2\pi} \int_{|z|=1} \alpha(z)\overline{\beta(z)} d\sigma(z) = \langle \alpha(U)\mathfrak{w}_1, \beta(U)\mathfrak{w}_1 \rangle, \qquad \alpha, \beta \in \mathbf{C}[z]. \tag{3.6}$$

Let the polynomials φ_k ($k \in \mathbf{N}$) satisfy the recurrence

$$z\varphi_n(z) = \frac{a_n}{a_{n+1}} \varphi_{n+1}(z) - \frac{b_{n+1}}{a_n a_{n+1}} \sum_{k=0}^{n} b_k \varphi_k(z), \qquad \varphi_0(z) = 1. \tag{3.7}$$

We easily derive from $\varphi_0(U)\mathfrak{w}_1 = \mathfrak{w}_1$ and (3.2) by induction that $\varphi_k(U)\mathfrak{w}_1 = \mathfrak{w}_{k+1}$. Therefore, $\{\varphi_k\}$ is the family of orthonormal polynomials corresponding to scalar product (3.6) (see [14, chap. 3, section 1]). The leading coefficient of φ_k is a_k; it is proved in [14] that $b_k = \varphi_k(0)$.

Introduce also the dual polynomials $\varphi_n^*(z) = z^n \varphi_n\left(z^{-1}\right)$, $n \in \mathbf{N}$, and put $\varphi_{-1} = \varphi_{-1}^* \equiv 0$. The relations

$$a_n \varphi_{n+1}(z) = a_{n+1} z \varphi_n(z) + b_{n+1} \varphi_n^*(z), \qquad n \in \mathbf{N}, \tag{3.8}$$

and

$$b_{n+1} \varphi_{n+1}(z) = a_{n+1} \varphi_{n+1}^*(z) - a_n \varphi_n^*(z), \qquad n \geq -1, \tag{3.9}$$

take place (see [14, chap. 3, section 1]).

THEOREM 3.2. *The number* 1 *is not an eigenvalue of* U.

Proof. Since $U\mathfrak{x} = \mathfrak{x}$ implies $U^*\mathfrak{x} = \mathfrak{x}$, it is sufficient to prove that the adjoint equation $U^*\mathfrak{x} - \mathfrak{x} = 0$ has no nontrivial solution $\mathfrak{x}$ in $\mathcal{H}$. Decompose $\mathfrak{x} = \sum_{l=0}^{\infty} x_l \mathfrak{w}_{l+1}$, $x_l \in \mathbf{C}$. Writing the equation componentwise, we have

$$-\frac{b_{n+1}}{a_n a_{n+1}} \sum_{k=0}^{n} b_k x_k + \frac{a_n}{a_{n+1}} x_{n+1} - x_n = 0, \qquad n \in \mathbf{N}. \tag{3.10}$$

If $x_0 \neq 0$, then, comparing (3.10) with (3.7), we see that $x_n = \varphi_n(1) x_0$.

From the equality $\varphi_n^*(1) = \varphi_n(1)$ and formula (3.8) we deduce

$$\frac{\varphi_{n+1}(1)}{\varphi_n(1)} = \frac{a_{n+1} + b_{n+1}}{a_n} = \frac{a_{n+1}}{a_n} \left(1 + \frac{b_{n+1}}{a_{n+1}}\right),$$

whence

$$\varphi_n(1) = a_n \prod_{l=1}^{n} \left(1 + \frac{b_l}{a_l}\right) \geq a_n \to \infty \quad \text{as} \quad n \to \infty,$$

which contradicts the assumption that $\mathfrak{x} \in \mathcal{H}$. Therefore, $x_0 = 0$ and $\mathfrak{x} = 0$. □

Theorem 3.2 means that 1 does not belong to the discrete spectrum of U. However, 1 may belong to the continuous spectrum. For example, this is the case if a_k

tends to $+\infty$ slower than any exponential. (It follows from [14, chap. 2, formula (7.19) and chap. 3, formula (6.3)] that in this case the spectrum of U is the whole unit circumference.) The next theorem asserts that if all steps of the GMRES process are uniformly far from stagnation, then there is a gap in the spectrum of U around 1.

THEOREM 3.3. *If the residual norms satisfy the inequality*

$$f_l/f_{l+1} \geq p > 1, \qquad l \in \mathbf{N}, \tag{3.11}$$

then 1 *is a regular point of* U*; namely,*

$$\left\|(I-U)^{-1}\right\| \leq \frac{1}{2}\sqrt{\frac{p+1}{p-1}}\left(\frac{1}{\sqrt{1-p^{-2}}}+\frac{1}{\sqrt{p^2-1}}\right). \tag{3.12}$$

Proof. For $a_l \to +\infty$ as $l \to +\infty$, the measure σ violates the Szegö condition (see [14, chap. 3, Theorem 2.1]). This implies that φ_l form a complete "basis" in the space $L_{2,\sigma}$ of functions whose squared modulus is integrable on the unit circumference with respect to σ (see [14, chap. 3, Theorem 2.2]).

By virtue of (3.2) and (3.7), U can be considered as the operator of multiplication by the independent variable z in $L_{2,\sigma}$.

We shall show that the operator $I-U$ is continuously invertible. First, set

$$\psi(z) = \sum_{l=0}^{\infty} u_l \varphi_l(z) \in L_{2,\sigma} \qquad \text{with} \quad \|\psi\|^2_{L_{2,\sigma}} = \sum_{l=0}^{\infty} |u_l|^2 = 1.$$

Using (3.9), perform the redecomposition from φ_l in φ_l^*:

$$\sum_{l=0}^{\infty} u_l \varphi_l = \sum_{l=0}^{\infty} v_l \varphi_l^* \qquad \text{with} \quad v_l = a_l\left(\frac{u_l}{b_l} - \frac{u_{l+1}}{b_{l+1}}\right),$$

the series in φ_l^* being convergent in $L_{2,\sigma}$, because $\|\varphi_l^*\|^2_{L_{2,\sigma}} = a_l^{-2}\sum_{k=0}^{l} b_k^2 = 1$ (see [14, chap. 3, section 1, identity (1.3)]),

$$\frac{a_l}{b_l} = \frac{1}{\sqrt{1-\left(\frac{a_{l-1}}{a_l}\right)^2}} \leq \frac{1}{\sqrt{1-p^{-2}}}, \tag{3.13}$$

and

$$\frac{a_l}{b_{l+1}} = \frac{1}{\sqrt{\left(\frac{a_{l+1}}{a_l}\right)^2-1}} \leq \frac{1}{\sqrt{p^2-1}}, \tag{3.14}$$

where we used (3.11) in the form $a_{l+1}/a_l \geq p$. Again exploiting (3.13) and (3.14), we get

$$\begin{aligned}\sqrt{\sum_{l=0}^{\infty}|v_l|^2} &\leq \sqrt{\sum_{l=0}^{\infty}\left(\frac{a_l}{b_l}|u_l|\right)^2} + \sqrt{\sum_{l=0}^{\infty}\left(\frac{a_l}{b_{l+1}}|u_{l+1}|\right)^2} \\ &\leq \max_{l\in\mathbf{N}}\frac{a_l}{b_l} + \max_{l\in\mathbf{N}}\frac{a_l}{b_{l+1}} \leq \frac{1}{\sqrt{1-p^{-2}}} + \frac{1}{\sqrt{p^2-1}}.\end{aligned} \tag{3.15}$$

Now we shall find a function $\chi(z)=\sum_{l=0}^{\infty}q_l\varphi_l(z)\in L_{2,\sigma}$ such that $(1-z)\chi(z)=\psi(z)$. In view of (3.8) and (3.9), we have

$$\begin{aligned}z\varphi_n&=a_{n+1}^{-1}\left(a_n\frac{a_{n+1}\varphi_{n+1}^*-a_n\varphi_n^*}{b_{n+1}}-b_{n+1}\varphi_n^*\right)\\&=\frac{a_n}{b_{n+1}}\varphi_{n+1}^*-\left(\frac{a_n^2}{a_{n+1}b_{n+1}}+\frac{b_{n+1}}{a_{n+1}}\right)\varphi_n^*=\frac{a_n}{b_{n+1}}\varphi_{n+1}^*-\frac{a_{n+1}}{b_{n+1}}\varphi_n^*\end{aligned}$$

and

$$(1-z)\varphi_n=-\frac{a_{n-1}}{b_n}\varphi_{n-1}^*+\left(\frac{a_n}{b_n}+\frac{a_{n+1}}{b_{n+1}}\right)\varphi_n^*-\frac{a_n}{b_{n+1}}\varphi_{n+1}^*,$$

whence

$$\begin{aligned}(1-z)\chi(z)&=\sum_{n=0}^{\infty}q_n\left[-\frac{a_{n-1}}{b_n}\varphi_{n-1}^*+\left(\frac{a_n}{b_n}+\frac{a_{n+1}}{b_{n+1}}\right)\varphi_n^*-\frac{a_n}{b_{n+1}}\varphi_{n+1}^*\right]\\&=\sum_{n=0}^{\infty}\varphi_n^*\left[-\frac{a_{n-1}}{b_n}q_{n-1}+\left(\frac{a_n}{b_n}+\frac{a_{n+1}}{b_{n+1}}\right)q_n-\frac{a_n}{b_{n+1}}q_{n+1}\right].\end{aligned}$$

The infinite tridiagonal matrix H with the rows

$$\left(\dots-\frac{a_{n-1}}{b_n}\quad\frac{a_n}{b_n}+\frac{a_{n+1}}{b_{n+1}}\quad-\frac{a_n}{b_{n+1}}\quad\dots\right)$$

is symmetric and presents a bounded self-adjoint operator in l_2. Since

$$\begin{aligned}\frac{a_n}{b_n}+\frac{a_{n+1}}{b_{n+1}}-\frac{a_{n-1}}{b_n}-\frac{a_n}{b_{n+1}}&=\frac{a_n-a_{n-1}}{\sqrt{a_n^2-a_{n-1}^2}}+\frac{a_{n+1}-a_n}{\sqrt{a_{n+1}^2-a_n^2}}\\&=\sqrt{\frac{a_n-a_{n-1}}{a_n+a_{n-1}}}+\sqrt{\frac{a_{n+1}-a_n}{a_{n+1}+a_n}}\ge 2\sqrt{\frac{p-1}{p+1}},\end{aligned}$$

we observe diagonal dominance in H:

$$H\ge 2\sqrt{\frac{p-1}{p+1}}I. \tag{3.16}$$

It follows from (3.15) and (3.16) that χ exists and

$$\|\chi\|\le\frac{1}{2}\sqrt{\frac{p+1}{p-1}}\left(\frac{1}{\sqrt{1-p^{-2}}}+\frac{1}{\sqrt{p^2-1}}\right).$$

This means that the resolvent $(I-U)^{-1}$ exists, (3.12) holding. □

Liesen [12] gave upper bounds for a residue of GMRES, applied to a unitary matrix, in terms of a gap in the spectrum. This is partially "dual" to what we just did.

As p tends to $+\infty$, the right-hand side of (3.12) tends to $1/2$; this means that $\mathrm{Sp}(U)$ concentrates around the point -1. On the other hand, as $p\to 1+0$, the right-hand side of (3.12) is asymptotically equivalent to $1/(p-1)$, so $\mathrm{Sp}(U)$ can approach 1.

The condition of the following assertion was considered in [9], but without deducing quantitative results.

COROLLARY 3.4. *If* 0 *does not belong to the closure of the numerical range of* A*, then* 1 *is a regular point of* U*.*

Proof. As in the proof of the error estimate for the steepest descent method [8, section 2.2], it can be shown that

$$\|\mathfrak{r}_{k+1}\|^2 \le \|\mathfrak{r}_k\|^2 - \frac{|\langle A\mathfrak{r}_k, \mathfrak{r}_k\rangle|^2}{\|A\mathfrak{r}_k\|^2}. \tag{3.17}$$

According to the condition, there exists a positive constant d such that $|\langle A\mathfrak{u}, \mathfrak{u}\rangle| \ge d\|\mathfrak{u}\|^2$ for any $\mathfrak{u} \in \mathcal{H}$. Combined with (3.17), this implies

$$f_{k+1} \le \sqrt{1 - (d/\|A\|)^2}\, f_k, \tag{3.18}$$

and it only remains to apply Theorem 3.3. □

Evidently, with greater d (i.e., the distance from 0 to the numerical range of A), a larger gap in the spectrum of U around 1 can be guaranteed.

The next theorem directly generalizes the main result of [10].

THEOREM 3.5. *Let* f_k *be a nonincreasing sequence of real numbers converging to* 0*. Then there exist a unitary operator* U *and a nonzero vector* $\mathfrak{r}$ *in a separable Hilbert space* $\mathcal{H}$ *such that* $f_k = \|\mathfrak{r}_k\|$ *for GMRES with* $(A, \mathfrak{r})$*.*

Proof. Let $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$ be an orthonormal "basis" of $\mathcal{H}$. Define the numbers g_k by the equality $g_k = \sqrt{f_{k-1}^2 - f_k^2}$ $(k \ge 1$; note that $\sum_{k=1}^{\infty} g_k^2 < +\infty)$, the vector $\mathfrak{r}$ by (2.3), the numbers a_n and b_n by (3.1), and the operator U by (3.2). As in the proofs of Theorem 3.1 and Lemma 2.1, it can be shown that $U^{-1}\mathfrak{w}_1 \uparrow\downarrow \mathfrak{r}$ and $U\mathcal{K}^k(U, \mathfrak{r}) = \mathcal{K}^k(U, \mathfrak{w}_1) = \operatorname{span}\{\mathfrak{w}_1, \ldots, \mathfrak{w}_k\}$, from which the formula for the residue is instantly derived. □

COROLLARY 3.6. *Let a sequence* $h_k \in]0, +\infty]$ *of elements of the extended real axis satisfy the condition* $\liminf_{k\to\infty} h_k < +\infty$*. Then there exist a unitary operator* U *and a nonzero vector* $\mathfrak{r}$ *in a separable Hilbert space* $\mathcal{H}$ *such that GMRES with* $(U, \mathfrak{r})$ *converges and* $\|\mathfrak{r}_k^A\| = h_k$*, where* $\mathfrak{r}_k^A$ *is the* kth *residue of FOM with* $(U, \mathfrak{r})$*.*

Proof. Set $f_0 = h_0$ and, inductively for $k \ge 1$,

$$f_k = \begin{cases} \frac{h_k}{\sqrt{1+(h_k/f_{k-1})^2}} & \text{if } h_k < +\infty, \\ f_{k-1} & \text{otherwise.} \end{cases}$$

As $0 < f_k \le f_{k-1}$, there exists $f = \lim_{k\to\infty} f_k \ge 0$. If $f > 0$, then

$$h_k = \begin{cases} \frac{f_k}{\sqrt{1-(f_k/f_{k-1})^2}} & \text{if } f_k \ne f_{k-1}, \\ +\infty & \text{otherwise} \end{cases} \to +\infty \quad \text{as} \quad k \to \infty,$$

which contradicts the assumption. Therefore, $f_k \to 0$.

According to Theorem 3.5, there exist an operator U and vector $\mathfrak{r}$ for which $\|\mathfrak{r}_k\| = f_k$. It is sufficient to use Theorem 3 in [5], asserting that

$$\|\mathfrak{r}_k^A\| = \begin{cases} \frac{\|\mathfrak{r}_k\|}{\sqrt{1-(\|\mathfrak{r}_k\|/\|\mathfrak{r}_{k-1}\|)^2}} & \text{if } \|\mathfrak{r}_k\| \ne \|\mathfrak{r}_{k-1}\|, \\ +\infty & \text{otherwise.} \end{cases}$$ □

4. A GMRES-equivalent self-adjoint nonnegative definite operator. Under the assumption

$$g_i \ne 0, \qquad \sup_{i\ge 1} g_{i+1}/g_i < +\infty,$$

define the infinite symmetric tridiagonal matrix

$$P = \begin{pmatrix} h_1 & e_1 & & \\ e_1 & h_2 & e_2 & \\ & \ddots & \ddots & \ddots \end{pmatrix} \qquad \text{with} \quad h_i = 1 + \frac{g_{i+1}^2}{g_i^2}, \quad e_i = -\frac{g_{i+1}}{g_i} \quad (i \geq 1).$$

The matrix P evidently presents a bounded self-adjoint operator in the "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$. The condition $g_i \neq 0$ is natural, because GMRES applied to a self-adjoint nonnegative definite operator cannot stagnate.

THEOREM 4.1. *The assertion GMRES$(A, \mathfrak{r}) =$ GMRES$(P, \mathfrak{r})$ holds for the self-adjoint operator P just constructed.*

Proof. Direct calculation based on (2.3) demonstrates that $P\mathfrak{r} = g_1\mathfrak{w}_1 \uparrow\downarrow w_1$. Then refer to Lemma 2.1. □

THEOREM 4.2. *The operator P is nonnegative definite, and 0 is not its eigenvalue.*

Proof. For any vector

$$\mathfrak{x} = \sum_{i=1}^{\infty} x_i \mathfrak{w}_i \in \mathcal{H} \qquad \left(\text{so that} \quad \sum_{i=1}^{\infty} |x_i|^2 < +\infty \right)$$

we derive

$$\begin{aligned} \langle P\mathfrak{x}, \mathfrak{x} \rangle &= \sum_{i=1}^{\infty} \left(1 + \frac{g_{i+1}^2}{g_i^2} \right) |x_i|^2 - \sum_{i=1}^{\infty} \frac{g_{i+1}}{g_i} \left(x_i \overline{x_{i+1}} + x_{i+1} \overline{x_i} \right) \\ &= |x_1|^2 + \sum_{i=1}^{\infty} \left| \frac{g_{i+1}}{g_i} x_i - x_{i+1} \right|^2 \geq 0, \end{aligned} \tag{4.1}$$

so P is nonnegative definite. Besides that, if $P\mathfrak{x} = 0$, then $x_1 = 0$ and, inductively, $x_2 = x_3 = \cdots = 0$. □

Theorem 4.2 shows, in particular, that P is injective, but does not guarantee that P is continuously invertible. The next theorem gives a sufficient condition for P to be so.

THEOREM 4.3. *If the numbers g_k satisfy the inequality*

$$g_i / g_j \leq c p^{i-j}, \qquad c > 0, \quad 0 < p < 1, \quad i \geq j \geq 1, \tag{4.2}$$

then the operator P is positive definite. Its condition number is estimated by

$$\operatorname{cond} P \leq \left[\frac{c(1+cp)}{1-p} \right]^2. \tag{4.3}$$

Proof. Let again $\mathfrak{x} = \sum_{i=1}^{\infty} x_i \mathfrak{w}_i \in \mathcal{H}$. Define the quantities $y_1 = x_1$, $y_{i+1} = g_{i+1}/g_i \cdot x_i - x_{i+1}$ $(i \geq 1)$ and the vector $\mathfrak{y} = \sum_{i=1}^{\infty} y_i \mathfrak{w}_i$. In view of (4.1), we have $\langle P\mathfrak{x}, \mathfrak{x} \rangle = \sum_{i=1}^{\infty} |y_i|^2 = \|\mathfrak{y}\|^2$.

Prove by induction that $|x_i| \leq \sum_{k=1}^{i} \frac{g_i}{g_k} |y_k|$. For $i = 1$ this assertion is reduced to $|x_1| \leq |y_1|$. Move from i to $i+1$:

$$|x_{i+1}| \leq |y_{i+1}| + \frac{g_{i+1}}{g_i} |x_i| \leq |y_{i+1}| + \frac{g_{i+1}}{g_i} \sum_{k=1}^{i} \frac{g_i}{g_k} |y_k| = \sum_{k=1}^{i+1} \frac{g_{i+1}}{g_k} |y_k|.$$

Thus,

$$\|\mathfrak{x}\|^2 \le \sum_{i=1}^{\infty} \left(\sum_{m=1}^{i} \frac{g_i}{g_m} |y_m| \right)^2 .$$

The terms with $|y_k| \cdot |y_l|$ $(k \ge l)$ have the summarized coefficient

$$\sum_{i=k}^{\infty} \frac{g_i^2}{g_k g_l} \le c^2 \sum_{i=k}^{\infty} p^{2i-k-l} = \frac{c^2 p^{k-l}}{1-p^2},$$

which is to be doubled if $k \ne l$ (we have used (4.2) here). Since for $m \ge 0$

$$\sum_{l=1}^{\infty} \frac{c^2 p^m}{1-p^2} |y_l| \cdot |y_{l+m}| \le \frac{c^2 p^m}{1-p^2} \|\mathfrak{y}\|^2,$$

then

$$\begin{aligned} \|\mathfrak{x}\|^2 \le \left(\frac{c^2}{1-p^2} + 2 \sum_{m=1}^{\infty} \frac{c^2 p^m}{1-p^2} \right) \|\mathfrak{y}\|^2 &= \left[\frac{c^2}{1-p^2} + \frac{2c^2 p}{(1-p)(1-p^2)} \right] \|\mathfrak{y}\|^2 \\ &= \frac{c^2}{(1-p)^2} \|\mathfrak{y}\|^2 = \frac{c^2}{(1-p)^2} \langle P\mathfrak{x}, \mathfrak{x} \rangle , \end{aligned}$$

whence

$$P \ge \left(\frac{1-p}{c} \right)^2 I. \tag{4.4}$$

In view of the estimate

$$\|P\| \le \max_i \{ |e_{i-1}| + |h_i| + |e_i| \} \le (1+cp)^2 \qquad (e_{-1} \equiv 0)$$

and (4.4), we can obtain bound (4.3). □

COROLLARY 4.4. *If* 0 *does not belong to the closure of the numerical range of* A*, then* 0 *is a regular point of* P*, i.e.,* P *is positive definite.*

Proof. We shall prove the applicability of Theorem 4.3 by demonstrating that a partial case of (4.2) holds. One can extract the inequality $f_k^2 - f_{k+1}^2 \ge (\frac{d}{\|A\|})^2 f_k^2$ from (3.18), whence $g_{k+1} \ge \frac{d}{\|A\|} f_k$ $(k \in \mathbf{N})$. If $i \ge j \ge 1$, then we deduce

$$\frac{g_i}{g_j} \le \frac{f_{i-1}}{\frac{d}{\|A\|} f_{j-1}} \le \frac{\|A\|}{d} \left(\sqrt{1 - \left(\frac{d}{\|A\|} \right)^2} \right)^{i-j},$$

where we again use inequality (3.18). □

5. A GMRES-equivalent self-adjoint indefinite operator. Put

$$u_k = \frac{f_k}{f_{k-1}}, \qquad v_k = (-1)^{k-1} \frac{f_{k-1}^2}{g_k f_{k-1} + g_{k+1} f_k}, \qquad k \ge 1.$$

Under the condition

$$|v_k| \le c_1, \qquad k \ge 1, \qquad c_1 > 0, \tag{5.1}$$

define the infinite matrices

$$L = \begin{pmatrix} 1 & & & \\ u_1 & 1 & & \\ & u_2 & 1 & \\ & & \ddots & \ddots \end{pmatrix}, \qquad D = \operatorname{diag}(v_1, v_2, \ldots).$$

In particular, condition (5.1) means that g_k and g_{k+1} cannot vanish simultaneously. This is natural, because a GMRES process with a self-adjoint operator cannot stagnate at two consecutive steps.

Owing to the inequality $u_k \le 1$ and condition (5.1), we can derive from [2, section 26, the second theorem] that L and D present bounded operators. Form the self-adjoint operator $T = LDL^*$. (As usual, we identify infinite matrices and operators in $\mathcal{H}$ by means of the "basis" $(\mathfrak{w}_1, \mathfrak{w}_2, \ldots)$.)

THEOREM 5.1. *The operator T is injective, and the pair $(T, \mathfrak{r})$ is GMRES-equivalent to $(A, \mathfrak{r})$.*

Proof. The injectivity of L and D is evident, and the one of L^* follows from the definition of u_k and the fact $\lim_{k\to\infty} f_k = 0$.

The matrix presenting T is tridiagonal. Since $v_k(g_k + u_k g_{k+1}) = (-1)^{k-1} f_{k-1}$, we have with use of (2.3) $T\mathfrak{r} = LDL^*\mathfrak{r} = L\sum_{k=1}^{\infty}(-1)^{k-1} f_{k-1}\mathfrak{w}_k = f_0\mathfrak{w}_1 \upuparrows \mathfrak{w}_1$. This completes the proof. ∎

THEOREM 5.2. *If the numbers f_k additionally satisfy the inequality*

$$f_i/f_j \le c_2 p^{i-j}, \qquad c_2 > 0, \qquad 0 < p < 1, \qquad i \ge j \ge 0, \tag{5.2}$$

then the operator T is continuously invertible; namely,

$$\operatorname{cond} T \le 2c_1(1 + c_2 p)^2 \left(1 + \frac{c_1 p}{1 - p}\right)^2. \tag{5.3}$$

Proof. First,

$$\left\|D^{-1}\right\| = \max_k \left|v_k^{-1}\right| = \max_k \left|\frac{g_k}{f_{k-1}} + \frac{g_{k+1}}{f_{k-1}} \cdot \frac{f_k}{f_{k-1}}\right| \le 2. \tag{5.4}$$

Second, due to (5.2) the infinite matrix

$$\begin{pmatrix} 1 & & & & \\ -u_1 & 1 & & & \\ u_1u_2 & -u_2 & 1 & & \\ -u_1u_2u_3 & u_2u_3 & -u_3 & 1 & \\ & \cdots & & & \ddots \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -\frac{f_1}{f_0} & 1 & & & \\ \frac{f_2}{f_0} & -\frac{f_2}{f_1} & 1 & & \\ -\frac{f_3}{f_0} & \frac{f_3}{f_1} & -\frac{f_3}{f_2} & 1 & \\ & \cdots & & & \ddots \end{pmatrix}$$

represents a bounded operator, and this operator is directly shown to be L^{-1}. Third, the continuous invertibility of L implies that of L^*. Thus, T is the product of three continuously invertible operators.

It follows from (5.1), (5.2), and (5.4) that $\|T\| \le c_1(1 + c_2 p)^2$ and

$$\left\|T^{-1}\right\| \le 2\left(1 + \frac{c_1 p}{1 - p}\right)^2,$$

which implies (5.3). ∎

6. Results and examples for matrices. Let $m \geq 1$, A_m be a linear operator in $\mathbf{C}^m$, $\mathfrak{r} \in \mathbf{C}^m$. Denote by $\mathfrak{r}_k$ $(0 \leq k \leq m-1)$ the residues of the GMRES process with $(A_m, \mathfrak{r})$ and by f_k the residual norms $f_k = \|\mathfrak{r}_k\|$. Assume that $\mathfrak{r}_{m-1} \neq 0$. Define also the numbers

$$g_k = \begin{cases} \sqrt{f_{k-1}^2 - f_k^2} & \text{if } 1 \leq k \leq m-1, \\ f_{m-1} & \text{if } k = m. \end{cases}$$

Let B_m be a linear operator in $\mathbf{C}^m$. Following [9, 10], we write

$$GMRES(A_m, \mathfrak{r}) = GMRES(B_m, \mathfrak{r}),$$

if $A_m \mathcal{K}^k(A_m, \mathfrak{r}) = B_m \mathcal{K}^k(B_m, \mathfrak{r})$ for $k = 0, \ldots, m-1$.

We introduce an orthonormal basis $(\mathfrak{w}_1, \ldots, \mathfrak{w}_m)$ of $\mathbf{C}^m$ by

$$\mathfrak{r}_{i-1} - \mathfrak{r}_i = g_i \mathfrak{w}_i, \qquad g_i = \|\mathfrak{r}_{i-1} - \mathfrak{r}_i\|, \qquad i = 1, \ldots, m, \qquad \mathfrak{r}_m \equiv 0$$

(note *the nonnegativity of* g_i; refer, e.g., to [15] for other details).

6.1. An equivalent unitary matrix. Define the numbers

$$a_n = \|\mathfrak{r}\|/f_n, \qquad b_n = \sqrt{a_n^2 - a_{n-1}^2} \qquad (n = 0, \ldots, m-1,\ a_{-1} \equiv 0),$$

and the Hessenberg matrix

$$U_m = \begin{pmatrix} -b_0 \frac{b_1}{a_0 a_1} & -b_0 \frac{b_2}{a_1 a_2} & \cdots & -b_0 \frac{b_{m-1}}{a_{m-2} a_{m-1}} & -\frac{b_0}{a_{m-1}} \\ \frac{a_0}{a_1} & -b_1 \frac{b_2}{a_1 a_2} & \cdots & -b_1 \frac{b_{m-1}}{a_{m-2} a_{m-1}} & -\frac{b_1}{a_{m-1}} \\ & \frac{a_1}{a_2} & \cdots & -b_2 \frac{b_{m-1}}{a_{m-2} a_{m-1}} & -\frac{b_2}{a_{m-1}} \\ & & & \vdots & \vdots \\ & & & -b_{m-2} \frac{b_{m-1}}{a_{m-2} a_{m-1}} & -\frac{b_{m-2}}{a_{m-1}} \\ & & & \frac{a_{m-2}}{a_{m-1}} & -\frac{b_{m-1}}{a_{m-1}} \end{pmatrix}. \tag{6.1}$$

We shall identify U_m and the linear operator in $\mathbf{C}^m$ defined by the matrix U_m in terms of the basis $(\mathfrak{w}_1, \ldots, \mathfrak{w}_m)$.

THEOREM 6.1. *The matrix* U_m *is unitary, and the assertion*

$$GMRES(A_m, \mathfrak{r}) = GMRES(U_m, \mathfrak{r})$$

holds in the sense of [9, 10].

Proof. The proof of this theorem (and also Theorems 6.4 and 6.7) is analogous to that of Theorem 3.1 (4.1 and 5.1, respectively). The only new (trivial) point is to check out that the mth component of $U_m^{-1} \mathfrak{w}_1$ is what we need. □

THEOREM 6.2. *The number* 1 *is not an eigenvalue of* U_m.

Proof. Modifying the proof of Theorem 3.2, we obtain the formula

$$x_k = a_k \prod_{l=1}^{k} \left(1 + \frac{b_l}{a_l}\right) x_0, \qquad k = 0, \ldots, m-1.$$

At that, formula (3.10) with $n = m-1$ becomes

$$-\sum_{k=0}^{m-1} \frac{b_k}{a_{m-1}} x_k - x_{m-1} = 0,$$

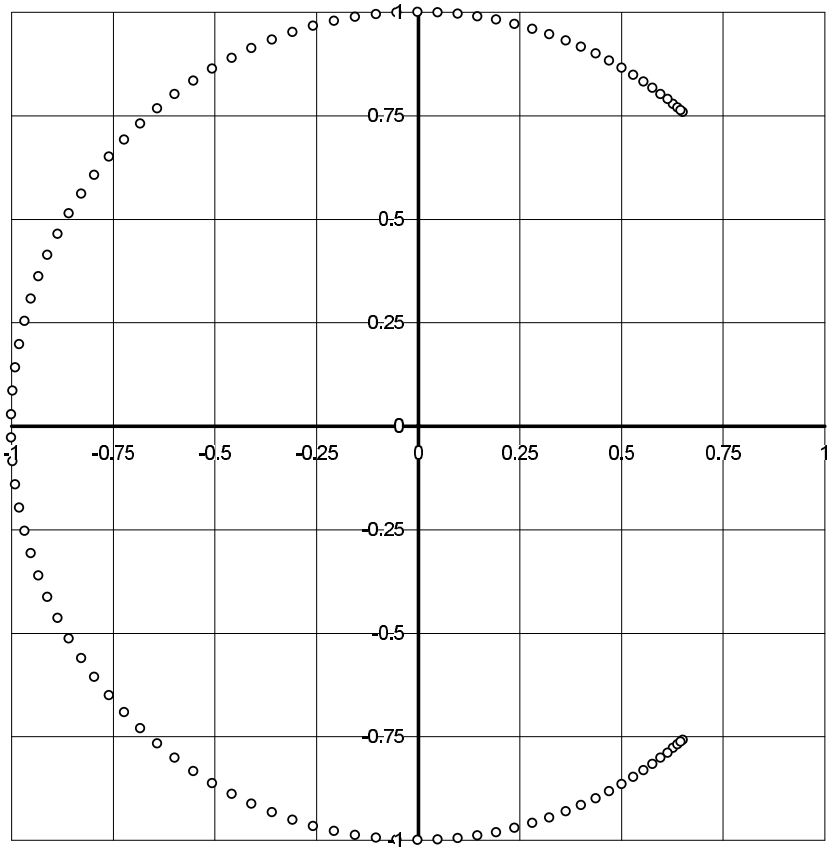


FIG. 6.1. $\mathrm{Sp}(U_m)$ *for* $m = 100$, $a_i = p^{i-1}$, $p = 1.1$.

which is possible only if $x_0 = 0$, because otherwise all the terms in the left-hand side are of the same sign. □

This theorem may also be derived from [1, Lemma 1].

THEOREM 6.3. *If the residual norms satisfy the inequality*

$$f_l/f_{l+1} \geq p > 1, \qquad l = 0, \ldots, m-2, \tag{6.2}$$

then

$$\left\|(I - U_m)^{-1}\right\| \leq \frac{1}{2}\sqrt{\frac{p+1}{p-1}}\left(\frac{1}{\sqrt{1-p^{-2}}} + \frac{1}{\sqrt{p^2-1}}\right). \tag{6.3}$$

Proof. The infinite series in the proof of Theorem 3.3 become finite by means of omitting the terms with φ_k, φ_k^* for $k \geq m$. The identity (3.8) for $n = m-1$ turns into $z\varphi_{m-1}(z) = -\varphi_{m-1}^*(z)$. □

We draw the spectra of U_m with $m = 100$ for two sequences f_k in Figures 6.1 and 6.2. Estimate (6.3) gives the lower bounds $\mathrm{dist}(1, \mathrm{Sp}(U_m)) \geq 0.09524$ and 0.00995, respectively. The clear gaps around the point 1 in the figures evidently satisfy the mentioned inequalities.

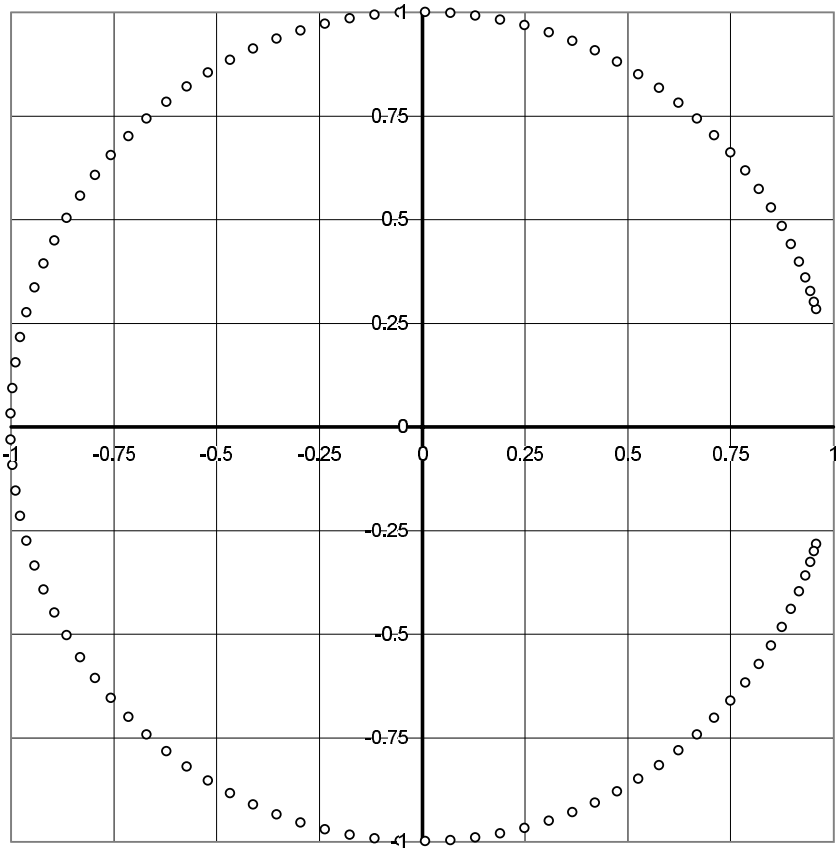


FIG. 6.2. $\mathrm{Sp}(U_m)$ *for* $m = 100$, $a_i = p^{i-1}$, $p = 1.01$.

6.2. An equivalent Hermitian positive definite matrix. Under the assumption

$$g_i \neq 0, \qquad i = 1, \ldots, m, \tag{6.4}$$

define the $m \times m$ symmetric tridiagonal matrix

$$P_m = \begin{pmatrix} h_1 & e_1 & & \mathrm{O} \\ e_1 & h_2 & e_2 & \\ & & \ddots & \\ \mathrm{O} & & e_{m-1} & h_m \end{pmatrix}$$

with

$$h_i = 1 + \frac{g_{i+1}^2}{g_i^2}, \quad e_i = -\frac{g_{i+1}}{g_i} \quad (i = 1, \ldots, m, \quad g_{m+1} \equiv 0).$$

THEOREM 6.4. *The assertion* GMRES$(A_m, \mathfrak{r}) =$ GMRES$(P_m, \mathfrak{r})$ *holds in the sense of* [9, 10].

THEOREM 6.5. *The matrix* P_m *is positive definite.*

Proof. The infinite series in the proof of Theorem 4.2 are truncated: the numbers x_i, $i > m$, vanish. ∎

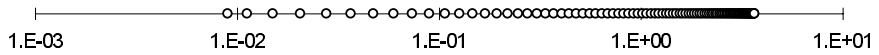


FIG. 6.3. $\mathrm{Sp}(P_m)$ *for* $m = 100$, $g_i = p^{i-1}$, $p = 1/1.1$.

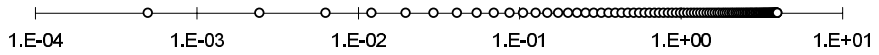


FIG. 6.4. $\mathrm{Sp}(P_m)$ *for* $m = 100$, $g_i = p^{i-1}$, $p = 1/1.01$.

THEOREM 6.6. *If the numbers* g_k *satisfy the inequality*

$$g_i/g_j \le cp^{i-j}, \qquad c > 0, \quad 0 < p < 1, \quad m \ge i \ge j \ge 1, \tag{6.5}$$

then the condition number of P_m *is estimated by*

$$\operatorname{cond} P_m \le \left[\frac{c(1+cp)}{1-p}\right]^2. \tag{6.6}$$

Proof. The infinite series in the proof of Theorem 4.3 are to be properly truncated. □

The initial assumption (6.4) is natural, because a GMRES process with an Hermitian positive definite matrix has no stagnation steps.

We draw the spectra of P_m with $m = 100$ for two sequences f_k in Figures 6.3 and 6.4. Estimate (6.6) gives the upper bounds $\operatorname{cond}(P_m) \le 441.00$ and 40401.0, respectively. In fact, the correspondent values of $\operatorname{cond}(P_m)$ are 405.01 and 7895.8.

6.3. An equivalent Hermitian indefinite matrix. Under the condition

$$\max(g_{k-1}, g_k) > 0, \qquad k = 2, \dots, m, \tag{6.7}$$

define the numbers

$$u_k = \frac{f_k}{f_{k-1}}, \qquad v_k = (-1)^{k-1}\frac{f_{k-1}^2}{g_k f_{k-1} + g_{k+1} f_k}, \qquad k = 1, \dots, m, \quad u_m \equiv 0,$$

and the $m \times m$ matrices

$$L_m = \begin{pmatrix} 1 & & & & \\ u_1 & 1 & & & \\ & u_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & u_{m-1} & 1 \end{pmatrix}, \qquad D_m = \operatorname{diag}(v_1, \dots, v_m).$$

Form the Hermitian matrix $T_m = L_m D_m L_m^*$.

THEOREM 6.7. *The matrix* T_m *is regular, and the pair* $(T_m, \mathfrak{r})$ *is GMRES-equivalent to* $(A_m, \mathfrak{r})$ *in the sense of* [9, 10].

THEOREM 6.8. *If the numbers* f_k *obey the inequality*

$$f_i/f_j \le c_2 p^{i-j}, \qquad c_2 > 0, \qquad 0 < p < 1, \qquad 0 \le j \le i \le m-1,$$

then the condition number of the matrix T_m *is estimated as*

$$\operatorname{cond} T_m \le 2c_1(1+c_2 p)^2 \left(1 + \frac{c_1 p}{1-p}\right)^2, \tag{6.8}$$

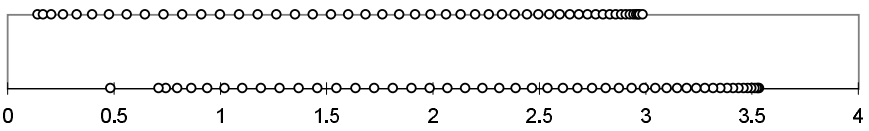


FIG. 6.5. $|\mathrm{Sp}(T_m)|$ *for* $m = 100$ *and stagnation at every other step. The two horizontal bars correspond to the positive and negative eigenvalues, respectively.*

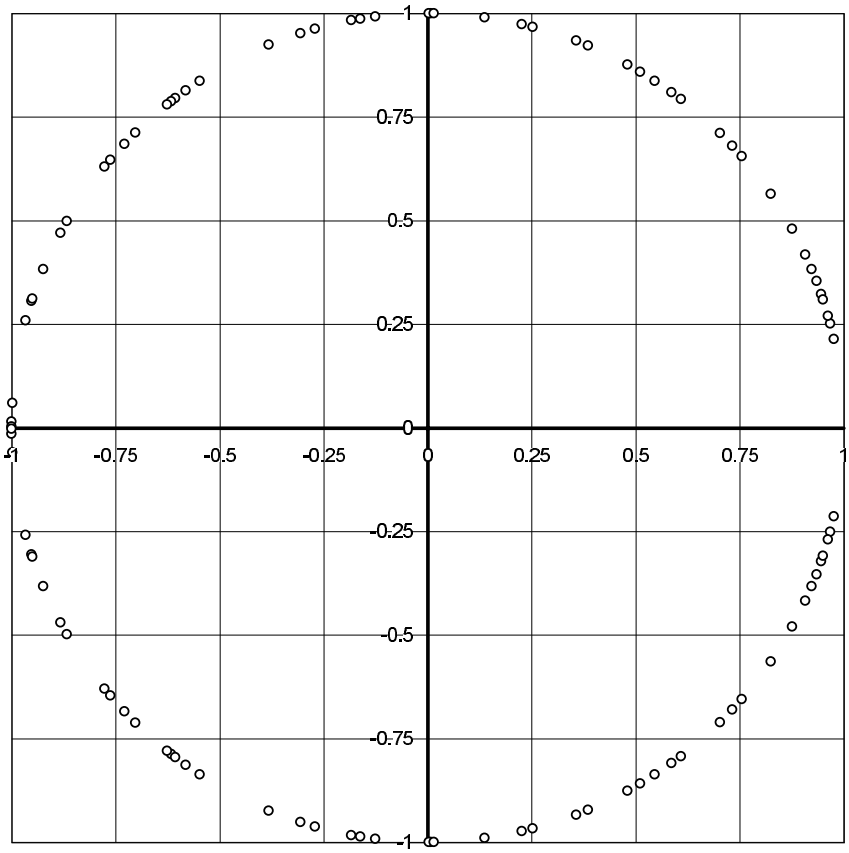


FIG. 7.1. $\mathrm{Sp}(U_m)$ *for* $m = 100$, $a_2 = a_3 = p^2$, $a_4 = \cdots = a_6 = p^5$, $a_{11} = \cdots = a_{15} = p^{14}$, $a_{16} = \cdots = a_{21} = p^{20}$, $a_{22} = \cdots = a_{28} = p^{27}$, $a_{29} = \cdots = a_{36} = p^{35}$, $a_{37} = \cdots = a_{45} = p^{44}$, $a_{46} = \cdots = a_{55} = p^{54}$, $a_{56} = \cdots = a_{64} = p^{63}$, $a_{65} = \cdots = a_{76} = p^{75}$, $a_{77} = \cdots = a_{89} = p^{88}$, $a_{90} = \cdots = a_{99} = p^{98}$, $p = 1.1$.

where

$$c_1 = \max_{k=1,\ldots,m-1} \frac{f_{k-1}^2}{g_k f_{k-1} + g_{k+1} f_k}.$$

Again, assumption (6.7) is obligatory, because a GMRES process with an Hermitian matrix cannot stagnate at two consecutive steps.

The proofs of Theorems 5.1 and 5.2 obviously induce those of Theorems 6.7 and 6.8.

We draw the modulus of the eigenvalues of T_m with $m = 100$ for the sequence

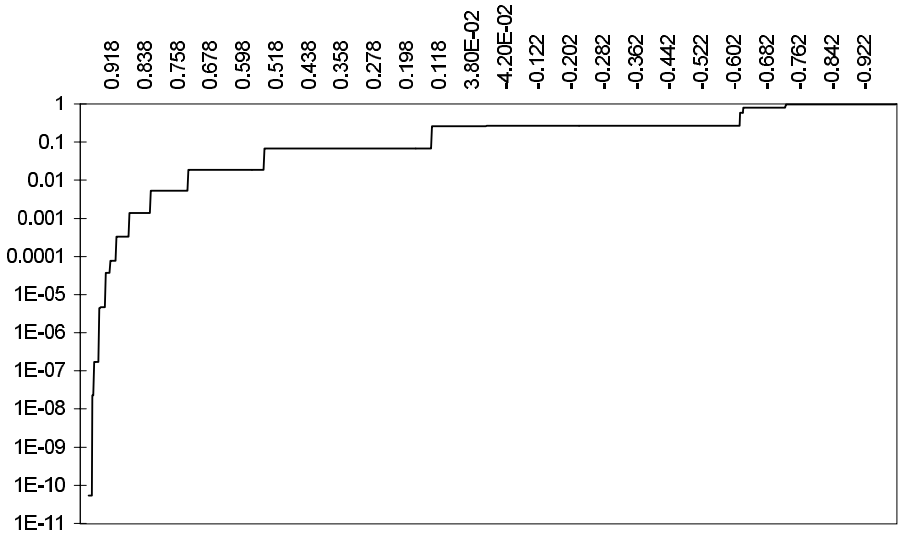


FIG. 7.2. *The "spectral function" of the pair* $(U_m, \mathfrak{r})$ *with the matrix* U_m *inherited from Figure* 7.1. *The abscissas axis is directed to the left.*

$f_k = 1.1^{-2[(k+1)/2]}$ in Figure 6.5. Estimate (6.8) gives the upper bound $\operatorname{cond} T_m \le 2500.7$. In fact, the value of $\operatorname{cond} T_m$ equals 24.545.

7. Open questions. We do not know if inequalities (3.12), (4.3), and (5.3) are precise in any sense. Besides that, there may be other gaps in the spectrum of U and gaps in those of P and T, also influencing the speed of convergence of GMRES.

Neither do we know how to interpret conditions (3.11), (4.2), and (5.2) in a natural way—i.e., not so roughly, as in Corollaries 3.4 and 4.4.

It is sometimes more suitable to characterize convergence in terms of the spectral measure instead of the spectrum. In Figure 7.1 we draw the spectrum of the matrix U_m with $m = 100$ for a residual sequence majorized by the one from the example presented in Figure 6.1. We can see that the gap around 1 in Figure 7.1 is much less than in Figure 6.1, notwithstanding the fact that convergence of the correspondent GMRES process is not worse. However, we plotted the graph of the function

$$x \mapsto \|\mathfrak{r}\|^{-2} \sum_{\Re\theta_i > x} |\langle \mathfrak{r}, \mathfrak{z}_i \rangle|^2$$

in Figure 7.2, where $(\theta_i, \mathfrak{z}_i)$ are the eigenpairs of U_m with normalized eigenvectors $\mathfrak{z}_i$. It is clear that the spectral density in a vicinity of 1 is small, which just reflects good (but not regular) convergence. It would be interesting to find rigorous statements for both the operator and matrix cases.

Acknowledgments. The author thanks Anne Greenbaum for useful discussions, supplying him with preliminary copies of a few papers, a book, and commentaries to them, and also for supplying a MATLAB program used in [9]; Zdeněk Strakoš for useful discussions and remarks; Jörg Liesen for useful discussions; Hakim Ikramov for supplying a few references; participants of Eugene Tyrtyshnikov's seminar at the Institute of Computational Mathematics for useful discussion; referee B for useful remarks.

REFERENCES

[1] G. S. Ammar, W. B. Gragg, and L. Reichel, *On the eigenproblem for orthogonal matrices*, in Proceedings of the 25th Conference on Decision and Control, Athens, Greece, IEEE, New York, 1986, pp. 1963–1966.

[2] N. I. Akhiezer and I. M. Glazman, *Theory of Linear Operators in Hilbert Space*, Dover, New York, 1993.

[3] M. Arioli, V. Pták, and Z. Strakoš, *Krylov sequences of maximal length and convergence of GMRES*, BIT, 38 (1998), pp. 636–643.

[4] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[5] J. Cullum and A. Greenbaum, *Relations between Galerkin and norm-minimizing iterative methods for solving linear systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 223–247.

[6] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.

[7] W. B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[8] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, PA, 1997.

[9] A. Greenbaum and Z. Strakoš, *Matrices that generate the same residual spaces*, in Recent Advances in Iterative Methods, G. Golub, A. Greenbaum, and M. Luskin, eds., IMA Vol. Math. Appl. 60, Springer-Verlag, 1994, pp. 95–118.

[10] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.

[11] L. Knizhnerman, *On adaptation of the Arnoldi method to the spectrum*, Schlumberger–Doll Research, Research Note #EMG-001-96-03, Ridgefield, CT, 1996.

[12] J. Liesen, *Computable convergence bounds for GMRES*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 882–903.

[13] O. Nevanlinna, *Convergence of Iterations for Linear Equations*, Lectures Math. ETH Zürich, Birkhäuser-Verlag, Basel, Switzerland, 1993.

[14] E. M. Nikishin and V. N. Sorokin, *Rational Approximations and Orthogonality*, Nauka, Moscow, 1988 (in Russian).

[15] M. Rozložník and Z. Strakoš, *Variants of the residual minimizing Krylov space methods*, in Proceedings of the Eleventh Summer School on Software and Algorithms of Numerical Mathematics, I. Marek, ed., University of West Bohemia, Plzeň, 1995, pp. 208–225.

[16] W. Rudin, *Functional Analysis*, McGraw-Hill, New York, 1973.

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

[18] G. Szegö, *Orthogonal Polynomials*, AMS, New York, 1959.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 213–229

ACCURACY OF TWO THREE-TERM AND THREE TWO-TERM RECURRENCES FOR KRYLOV SPACE SOLVERS*

MARTIN H. GUTKNECHT† AND ZDENĚK STRAKOŠ‡

Abstract. It has been widely observed that Krylov space solvers based on two three-term recurrences can give significantly less accurate residuals than mathematically equivalent solvers implemented with three two-term recurrences. In this paper we attempt to clarify and justify this difference theoretically by analyzing the gaps between recursively and explicitly computed residuals.

It is shown that, in contrast with the two-term recurrences analyzed by Sleijpen, van der Vorst, and Fokkema [*Numer. Algorithms*, 7 (1994), pp. 75–109] and Greenbaum [*SIAM J. Matrix Anal. Appl.*, 18 (1997), pp. 535–551], in the two three-term recurrences the contributions of the local roundoff errors to the analyzed gaps may be dramatically amplified while propagating through the algorithm. This result explains, for example, the well-known behavior of three-term-based versions of the biconjugate gradient method, where large gaps between recursively and explicitly computed residuals are not uncommon. For the conjugate gradient method, however, such a devastating behavior—although possible—is not observed frequently in practical computations, and the difference between two-term and three-term implementations is usually moderate or small. This can also be explained by our results.

Key words. system of linear algebraic equations, iterative method, Krylov space method, conjugate gradient method, three-term recurrence, accuracy, roundoff

AMS subject classifications. 65F10, 65G05

PII. S0895479897331862

1. Introduction.

Among the Krylov space solvers for linear systems $\mathbf{A}\mathbf{x} = \mathbf{b}$ (with $\mathbf{A}$ an $(N \times N)$-matrix and $\mathbf{b}$ an N-vector) there are quite a few that are based on three-term recurrences for both the *residuals* $\mathbf{r}_n$ and the *iterates* $\mathbf{x}_n$. Given an initial approximation $\mathbf{x}_0$, we let $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{r}_{-1} = \mathbf{o}$, $\mathbf{x}_{-1} = \mathbf{o}$, $\beta_{-1} = 0$ and consider for $n \geq 0$, while $\gamma_n \neq 0$,

$$\begin{aligned} \mathbf{r}_{n+1} &= (\mathbf{A}\mathbf{r}_n - \mathbf{r}_n\alpha_n - \mathbf{r}_{n-1}\beta_{n-1})/\gamma_n\,, \\ \mathbf{x}_{n+1} &= -(\mathbf{r}_n + \mathbf{x}_n\alpha_n + \mathbf{x}_{n-1}\beta_{n-1})/\gamma_n\,. \end{aligned} \tag{1.1}$$

In order that the recurrences (1.1) be consistent with the residual definition $\mathbf{r}_n \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n$, the scaling coefficients γ_n need to be chosen according to

$$\gamma_n = -(\alpha_n + \beta_{n-1}), \tag{1.2}$$

which means that the tridiagonal matrix with coefficients β_{n-1}, α_n, and γ_n in its $(n+1)$st column has column sums zero; see, for example, section 4.3 of [14].

The list of algorithms based on (1.1) and (1.2) includes the Chebyshev iteration [24, 21, 19], the second-order Richardson iteration [21] (which is the stationary form

*Received by the editors December 23, 1997; accepted for publication (in revised form) by G. H. Golub on September 22, 1999; published electronically June 3, 2000.

http://www.siam.org/journals/simax/22-1/33186.html

†Seminar for Applied Mathematics, ETH Zürich, ETH-Zentrum, CH-8092 Zürich, Switzerland (mhg@sam.math.ethz.ch).

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (on leave from Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague) (strakos@mathcs.emory.edu). This author's work was supported by ASCR grant A2030706 and by GA CR grant 205/96/0921. Part of the work was performed while he visited the Swiss Center for Scientific Computing (CSCS/SCSC) in 1997.

of the Chebyshev iteration), the three-term versions (ORes) of the conjugate gradient (CG) and the conjugate residual (CR) methods [24, 15], and the three-term version (BiORes) of the unsymmetric or two-sided Lanczos method [18, 14] (which is a variation of the biconjugate gradient (BiCG) method); see also [2, 15]. On the other hand, for example, neither the version of CG suggested by Rutishauser [21] (based on recurrences for the increments in $\mathbf{x}$ and $\mathbf{r}$) nor the MINRES algorithm of Paige and Saunders [20], which implements the CR method for symmetric indefinite matrices, nor their SYMMLQ algorithm is covered by our assumptions. An interesting contribution to the rounding error analysis of MINRES and SYMMLQ can be found in [23].

The CG, CR, and BiCG methods have better known versions (OMin and BiOMin) that are instead based on three two-term recurrences involving, in addition to the iterates and their residuals, direction vectors $\mathbf{p}_n$: for $n \geq 0$,

$$\begin{aligned} \mathbf{p}_n &= \mathbf{r}_n + \mathbf{p}_{n-1}\psi_{n-1}\,, \\ \mathbf{r}_{n+1} &= \mathbf{r}_n - \mathbf{A}\mathbf{p}_n\omega_n\,, \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{p}_n\omega_n \end{aligned} \tag{1.3}$$

with $\mathbf{p}_0 = \mathbf{r}_0$. Other methods like OrthoMin [28] use the last two of these recurrences, but have a more complex update formula for the direction vectors. In principle, the version (1.3) can be obtained from the three-term version (1.1)–(1.2) by an LU decomposition of the tridiagonal matrix of recurrence coefficients; see [1, 5, 14, 20]. The folklore is that implementations based on the two-term recurrences (1.3) are less affected by roundoff than those based on the three-term recurrences (1.1)–(1.2). It should be pointed out that the meaning of the phrase *less affected by roundoff* should be carefully specified, otherwise the previous statement is imprecise and can be misleading.

Recent work of Greenbaum [10, 11] shows that under the sole assumption that the last two recurrences (1.3) hold, there is a limitation on the accuracy of the iterates computed in finite precision arithmetic; the corresponding residuals $\mathbf{b}-\mathbf{A}\mathbf{x}_n$ cannot be expected to decrease below a certain level. (A similar but somewhat weaker result was given by Sleijpen, van der Vorst, and Fokkema [22].) This level depends primarily on the largest norm of an approximate solution $\mathbf{x}_n$ generated during the iteration, but it does not explicitly depend on how the coefficients ω_n and ψ_n are determined. Since, for example, the BiCG method may produce very large intermediate iterates and residuals, this result is of great importance in practice. In contrast, related work on GMRES showed that the size of intermediate iterates does not play a role [4, 12].

In this paper we investigate and answer the question when and why algorithms based on two three-term recurrences of the form (1.1)–(1.2) usually do not produce as small residuals as mathematically equivalent algorithms based on three two-term recurrences (1.3). Similarly to [10, 11, 22, 4], we investigate the gap $\mathbf{f}_n \equiv (\mathbf{b} - \mathbf{A}\mathbf{x}_n) - \mathbf{r}_n$ between the explicitly computed residuals $\mathbf{b} - \mathbf{A}\mathbf{x}_n$ and the recursively updated residuals $\mathbf{r}_n$. We will refer to the former as *true* residuals and to the latter as *updated* residuals. We show that for computations based on (1.1)–(1.2), the gap $\mathbf{f}_n$ satisfies a nonhomogeneous second-order difference equation. By writing n steps of this difference equation as the superposition of $n+1$ homogeneous difference equations (in a different context, this idea has been used by Grcar [8]), we receive an explicit formula for $\mathbf{f}_n$ in terms of the local roundoff errors. The resulting formula contains, in addition to the sum of local errors (which is the analog of the sum that represents the gap $\mathbf{f}_n$ in the case of two-term recurrences analyzed by Greenbaum), each local error

multiplied by a set of potentially large multipliers. Moreover, the local errors may become for the two three-term recurrences much larger than for two-term recurrences.

Assume that—in any application for which they are suitable—the methods based on the recurrences (1.1)–(1.2) or (1.3) will eventually produce small updated residuals (whose norm will decrease to the level of roundoff occurring in the finite precision computation of the residual $\mathbf{b} - \mathbf{A}\mathbf{x}$ for the exact solution $\mathbf{x}$). Then the size of the gap $\mathbf{f}_n$ determines the ultimate attainable accuracy measured by the size of the true residual; a large gap will eventually mean a poor residual $\mathbf{b} - \mathbf{A}\mathbf{x}_n$. The methods based on (1.1)–(1.2) are proven to be *in this sense* potentially much less accurate than those based on (1.3). In this sense, the folklore statement mentioned above is correct.

Our theoretical conclusions are well supported by numerical experiments.

It should be mentioned that the question of the ultimate attainable accuracy of iterative methods was studied by several other authors in addition to those mentioned above; see, for example, [3, 17, 25, 26, 27]. For a more detailed discussion we refer to [11]. However, to our knowledge, the problem of numerical differences between the recurrences (1.1)–(1.2) and (1.3) was not analyzed in these papers.

2. Local roundoff and the basic recurrence for the gap. In finite precision arithmetic, recurrences (1.1) have to be replaced by

$$\begin{aligned} \mathbf{r}_{n+1} &= (\mathbf{A}\mathbf{r}_n - \mathbf{r}_n\alpha_n - \mathbf{r}_{n-1}\beta_{n-1} + \mathbf{g}_n)/\gamma_n\,, \\ \mathbf{x}_{n+1} &= -(\mathbf{r}_n + \mathbf{x}_n\alpha_n + \mathbf{x}_{n-1}\beta_{n-1} - \mathbf{h}_n)/\gamma_n\,, \end{aligned} \tag{2.1}$$

where $\mathbf{g}_n$ and $\mathbf{h}_n$ contain all the local rounding errors produced at the step $n+1$, and $\mathbf{r}_n$, $\mathbf{x}_n$, etc., denote the actually computed quantities.

The first step of the analysis consists of estimating these local errors. We make the usual assumption that the floating-point arithmetic with roundoff unit ϵ satisfies

$$\mathrm{fl}(a \pm b) = a(1+\epsilon_1) \pm b(1+\epsilon_2), \qquad |\epsilon_1|, |\epsilon_2| \le \epsilon, \tag{2.2}$$

$$\mathrm{fl}(a \ \mathrm{op}\ b) = (a \ \mathrm{op}\ b)(1+\epsilon_3), \qquad |\epsilon_3| \le \epsilon, \qquad \mathrm{op} = *, /. \tag{2.3}$$

Then the roundoff in the matrix-vector multiplication (computed in a standard way) is bounded according to

$$|\mathrm{fl}(\mathbf{A}\mathbf{p}) - \mathbf{A}\mathbf{p}| \le m\,\epsilon\,|\mathbf{A}|\,|\mathbf{p}| + \mathcal{O}(\epsilon^2), \tag{2.4}$$

where $|\mathbf{A}|$ and $|\mathbf{p}|$ denote the elementwise absolute values of $\mathbf{A}$ and $\mathbf{p}$, and m is the maximal number of nonzeros in any row of $\mathbf{A}$. Assuming that the first and the third terms in (1.1) are summed up first, by applying these rules we get

$$|\mathbf{g}_n| \le ((m+3)\,|\mathbf{A}|\,|\mathbf{r}_n| + 3\,|\mathbf{r}_n\alpha_n| + 4\,|\mathbf{r}_{n-1}\beta_{n-1}|)\,\epsilon + \mathcal{O}(\epsilon^2), \tag{2.5}$$

$$|\mathbf{h}_n| \le (3\,|\mathbf{r}_n| + 3\,|\mathbf{x}_n\alpha_n| + 4\,|\mathbf{x}_{n-1}\beta_{n-1}|)\,\epsilon + \mathcal{O}(\epsilon^2). \tag{2.6}$$

Both $\mathbf{g}_n$ and $\mathbf{h}_n$ are bounded by a quantity proportional to ϵ, but the behavior of their bounds close to convergence is different. While the updated residual will become eventually small in reasonable computations, and the bound for $|\mathbf{g}_n|$ will decrease correspondingly, the bound for $|\mathbf{h}_n|$ will not. Note that we could consider a norm of $\mathbf{g}_n$ and $\mathbf{h}_n$ here, but there is no real need for this.

In the following estimates we assume that the computed coefficients α_n, β_{n-1}, and γ_n satisfy, in analogy to (1.2),

$$\gamma_0 = -\alpha_0\,, \qquad \gamma_n = -(\alpha_n + \beta_{n-1}) + \varepsilon_n \quad (n > 0) \tag{2.7}$$

with error terms ε_n (note that this symbol is distinct from ϵ) that are bounded by

$$|\varepsilon_n| \le (|\alpha_n| + |\beta_{n-1}|)\,\nu\,\epsilon \quad (n > 0)\,, \tag{2.8}$$

where ν is a suitable small constant. Note that $\nu = 1$ when γ_n is computed using (1.2). For later convenience we set $\varepsilon_0 = 0$.

We want to estimate the norm of the difference (or gap) between updated and true residuals, hence, of

$$\mathbf{f}_n \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n - \mathbf{r}_n\,.$$

For $n = 0$, the gap $\mathbf{f}_0$ is the roundoff in computing $\mathbf{r}_0$ from $\mathbf{A}$, $\mathbf{x}_0$, and $\mathbf{b}$; that is, $\mathbf{f}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 - \mathrm{fl}(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$, and this is bounded by

$$|\mathbf{f}_0| \le ((m+1)\,|\mathbf{A}|\,|\mathbf{x}_0| + |\mathbf{b}|)\ \epsilon + \mathcal{O}(\epsilon^2). \tag{2.9}$$

Inserting the recursions (2.1) and the equality (2.7) we have

$$\begin{aligned}
\mathbf{f}_{n+1} &= \mathbf{b} + (\mathbf{A}\mathbf{r}_n + \mathbf{A}\mathbf{x}_n\alpha_n + \mathbf{A}\mathbf{x}_{n-1}\beta_{n-1} - \mathbf{A}\mathbf{h}_n)\frac{1}{\gamma_n} \\
&\quad - (\mathbf{A}\mathbf{r}_n - \mathbf{r}_n\alpha_n - \mathbf{r}_{n-1}\beta_{n-1} + \mathbf{g}_n)\frac{1}{\gamma_n} \\
&= -\left[(\mathbf{b} - \mathbf{A}\mathbf{x}_n - \mathbf{r}_n)\,\alpha_n + (\mathbf{b} - \mathbf{A}\mathbf{x}_{n-1} - \mathbf{r}_{n-1})\,\beta_{n-1} - \mathbf{b}\,\varepsilon_n + \mathbf{A}\mathbf{h}_n + \mathbf{g}_n\right]\frac{1}{\gamma_n} \\
&= -\left[\mathbf{f}_n\,\alpha_n + \mathbf{f}_{n-1}\,\beta_{n-1} - \mathbf{b}\,\varepsilon_n + \mathbf{A}\mathbf{h}_n + \mathbf{g}_n\right]\frac{1}{\gamma_n}.
\end{aligned} \tag{2.10}$$

Let us gather the last three terms, the local errors, in

$$\mathbf{l}_n \equiv (-\mathbf{b}\,\varepsilon_n + \mathbf{A}\mathbf{h}_n + \mathbf{g}_n)\,\frac{1}{\gamma_n}\,.$$

By inserting the estimates (2.5), (2.6), and (2.8) we see that

$$\begin{aligned}
|\mathbf{l}_n| \le [\,&|\mathbf{b}|\,(|\alpha_n| + |\beta_{n-1}|)\,\nu + (m+6)\,|\mathbf{A}|\,|\mathbf{r}_n| + 3\,(|\mathbf{A}|\,|\mathbf{x}_n| + |\mathbf{r}_n|)\,|\alpha_n| \\
&+ 4\,(|\mathbf{A}|\,|\mathbf{x}_{n-1}| + |\mathbf{r}_{n-1}|)\,|\beta_{n-1}|\,]\,\frac{\epsilon}{|\gamma_n|} + \mathcal{O}(\epsilon^2).
\end{aligned}$$

For $n = 0$, we have $\gamma_0 = -\alpha_0$, $\varepsilon_0 = 0$, and thus

$$\mathbf{l}_0 = (\mathbf{A}\mathbf{h}_0 + \mathbf{g}_0)\frac{1}{\gamma_0}, \qquad \mathbf{f}_1 = \mathbf{f}_0 - \mathbf{l}_0.$$

In summary, (2.10) yields for the gaps $\mathbf{f}_n$ the linear second-order difference equation

$$\mathbf{f}_1 = \mathbf{f}_0 - \mathbf{l}_0\,, \qquad \mathbf{f}_{n+1} = -\left(\mathbf{f}_n\frac{\alpha_n}{\gamma_n} + \mathbf{f}_{n-1}\frac{\beta_{n-1}}{\gamma_n} + \mathbf{l}_n\right) \qquad (n \ge 1), \tag{2.11}$$

or, equivalently, the pair of first-order difference equations

$$\begin{bmatrix} \mathbf{f}_n \\ \mathbf{f}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -\frac{\beta_{n-1}}{\gamma_n}\mathbf{I} & -\frac{\alpha_n}{\gamma_n}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{n-1} \\ \mathbf{f}_n \end{bmatrix} - \begin{bmatrix} \mathbf{o} \\ \mathbf{l}_n \end{bmatrix} \qquad (n \ge 1) \tag{2.12}$$

with $\mathbf{f}_1 = \mathbf{f}_0 - \mathbf{l}_0$. These recurrences describe the propagation of the local rounding errors $\mathbf{l}_k$, $k = 0, \ldots, n$. We see that the gap $\mathbf{f}_n$ between the updated and the true residuals after n steps is determined by a nonhomogeneous second-order difference equation. This is in sharp contrast to the error behavior of the coupled two-term recurrences, where the gap after n steps is just a simple sum of local errors; see [11]. Consequently, as we will see in the next section, the two three-term recurrences may suffer from a strong amplification of the local errors.

3. Formula for the gap between true and updated residuals. For the moment, assume that the term ε_n in (2.7) vanishes, that is,

$$-\frac{\alpha_n}{\gamma_n} - \frac{\beta_{n-1}}{\gamma_n} = 1 \tag{3.1}$$

holds even in finite precision arithmetic. Denote by $\mathbf{z}_{n+1} = \mathcal{D}(\mathbf{z}_{n-m+1}, \mathbf{z}_{n-m}; m)$ the result of m steps of the recurrence

$$\mathbf{z}_{k+1} = -\mathbf{z}_k \frac{\alpha_k}{\gamma_k} - \mathbf{z}_{k-1} \frac{\beta_{k-1}}{\gamma_k}\,, \qquad k = n-m+1, \ldots, n, \tag{3.2}$$

started at the step $n-m$. Note that due to (3.1), $\mathbf{z}_{n-m+k+1} = \mathcal{D}(\mathbf{z}_{n-m+1}, \mathbf{z}_{n-m}; k) = \mathbf{z}_{n-m}$ for all k whenever $\mathbf{z}_{n-m+1} = \mathbf{z}_{n-m}$. Our discussion will rely heavily on this fact.

First, we derive how the gap $\mathbf{f}_{n+1}$ is affected by $\mathbf{f}_0$. Clearly, the part of this gap that depends on $\mathbf{f}_0$ is given by

$$\mathcal{D}(\mathbf{f}_0, \mathbf{f}_0; n) = \mathbf{f}_0\,,$$

that is, $\mathbf{f}_0$ is not amplified in the process. Next we have to analyze the dependence of $\mathbf{f}_{n+1}$ on the elementary rounding errors $\mathbf{l}_0$ born in the first step of the algorithm. Clearly, considering (2.11) for $n = 1$, subtracting and adding $\mathbf{l}_0 \frac{\beta_0}{\gamma_1}$, the contribution of $\mathbf{l}_0$ to the gap $\mathbf{f}_{n+1}$ can be decomposed into two parts: the part which propagates through the recurrence without any change,

$$\mathcal{D}(-\mathbf{l}_0, -\mathbf{l}_0; n) = -\,\mathbf{l}_0\,,$$

and the part depending on the modified local error of the first step,

$$\widetilde{\mathbf{l}}_1 \equiv\ \mathbf{l}_0 \frac{\beta_0}{\gamma_1} + \mathbf{l}_1\,,$$

which has yet to be analyzed. Repeating the same idea for the steps 2 through n, we can conclude that the gap $\mathbf{f}_{n+1}$ can be written as the following superposition of effects of local errors:

$$\begin{aligned}
\mathbf{f}_{n+1} = \mathbf{f}_0 \quad & -\mathbf{l}_0 \\
& -\mathbf{l}_0 \frac{\beta_0}{\gamma_1} - \mathbf{l}_1 \\
& -\mathbf{l}_0 \frac{\beta_0 \beta_1}{\gamma_1 \gamma_2} - \mathbf{l}_1 \frac{\beta_1}{\gamma_2} - \mathbf{l}_2 \\
& \ \ \vdots \\
& -\mathbf{l}_0 \frac{\beta_0 \beta_1 \cdots \beta_{n-1}}{\gamma_1 \gamma_2 \cdots \gamma_n} - \ldots - \mathbf{l}_{n-1} \frac{\beta_{n-1}}{\gamma_n} - \mathbf{l}_n.
\end{aligned} \tag{3.3}$$

Let us give another derivation of this fundamental result. From (2.12) we see that, in view of $\mathbf{f}_1 = \mathbf{f}_0 - \mathbf{l}_0$,

$$\left[\begin{array}{c}\mathbf{f}_n\\ \mathbf{f}_{n+1}\end{array}\right] = \prod_{k=1}^{n}\left[\begin{array}{cc}\mathbf{O} & \mathbf{I}\\ -\frac{\beta_{k-1}}{\gamma_k}\mathbf{I} & -\frac{\alpha_k}{\gamma_k}\mathbf{I}\end{array}\right]\left[\begin{array}{c}\mathbf{f}_0\\ \mathbf{f}_0\end{array}\right] - \sum_{j=0}^{n}\prod_{k=j+1}^{n}\left[\begin{array}{cc}\mathbf{O} & \mathbf{I}\\ -\frac{\beta_{k-1}}{\gamma_k}\mathbf{I} & -\frac{\alpha_k}{\gamma_k}\mathbf{I}\end{array}\right]\left[\begin{array}{c}\mathbf{o}\\ \mathbf{l}_j\end{array}\right]. \tag{3.4}$$

Here, due to (3.1), the matrices in the first product leave $[\ \mathbf{f}_0^\top \quad \mathbf{f}_0^\top\ ]^\top$ invariant. In the product that appears after the sum, we split off the last matrix (the one where $k = j+1$) and apply it to $[\ \mathbf{o}^\top \quad \mathbf{l}_j^\top\ ]^\top$ to get

$$\left[\begin{array}{c}\mathbf{l}_j\\ -\mathbf{l}_j\,\frac{\alpha_{j+1}}{\gamma_{j+1}}\end{array}\right] = \left[\begin{array}{c}\mathbf{l}_j\\ \mathbf{l}_j\end{array}\right] + \left[\begin{array}{c}\mathbf{o}\\ \mathbf{l}_j\,\frac{\beta_j}{\gamma_{j+1}}\end{array}\right].$$

Now we have again a first term that is left invariant by the matrices it is multiplied with and a second term of the form $[\ \mathbf{o}^\top \quad \star\ ]^\top$ that can be treated in the same way that $[\ \mathbf{o}^\top \quad \mathbf{l}_j^\top\ ]^\top$ was treated before. Repeating this trick we finally obtain

$$\left[\begin{array}{c}\mathbf{f}_n\\ \mathbf{f}_{n+1}\end{array}\right] = \left[\begin{array}{c}\mathbf{f}_0\\ \mathbf{f}_0\end{array}\right] - \sum_{j=0}^{n-1}\left[\begin{array}{c}\mathbf{l}_j\\ \mathbf{l}_j\end{array}\right]\left(1 + \frac{\beta_j}{\gamma_{j+1}} + \cdots + \frac{\beta_j\cdots\beta_{n-2}}{\gamma_{j+1}\cdots\gamma_{n-1}}\right) - \sum_{j=0}^{n}\left[\begin{array}{c}\mathbf{o}\\ \mathbf{l}_j\end{array}\right]\frac{\beta_j\cdots\beta_{n-1}}{\gamma_{j+1}\cdots\gamma_n}, \tag{3.5}$$

which is the same as formula (3.3), written for both $\mathbf{f}_n$ and $\mathbf{f}_{n+1}$.

Now we describe how the picture changes when the coefficients α_n, β_{n-1}, and γ_n are computed imprecisely, that is, when (3.1) is replaced by (2.7). We can follow the analysis described above with the only difference being that we should add the effect of the quantity $\mathbf{f}_0\varepsilon_1/\gamma_1$ propagating through $n-1$ steps of the recurrence (3.2) with $\mathbf{z}_1 := \mathbf{o}$, the effect of $\widetilde{\mathbf{l}}_1\varepsilon_2/\gamma_2$ propagating through $n-2$ steps of (3.2) with $\mathbf{z}_2 := \mathbf{o}$, and so on. As long as the constant ν is small and ε_n is close to the machine precision ϵ, these modifications will only cause effects proportional to $\mathcal{O}(\epsilon^2)$. In (3.3) we should therefore add terms $\mathcal{O}(\epsilon^2)$ to individual terms of the sum. However, once the size of these terms is considered, the new $\mathcal{O}(\epsilon^2)$ contributions can be thought of as being incorporated in the $\mathcal{O}(\epsilon^2)$ terms already present in the bounds for $\mathbf{f}_0, \mathbf{l}_0, \ldots, \mathbf{l}_n$. Therefore, we can use (3.3) in the further analysis with no change and no limitation.

We summarize our main result in the following theorem.

THEOREM 3.1. *Up to a term $\mathcal{O}(\epsilon^2)$, the gap $\mathbf{f}_{n+1}$ between true and updated*

residuals is given by the formula

$$
\begin{aligned}
\mathbf{f}_{n+1} = \mathbf{f}_0 \quad & - \sum_{j=0}^{n} \mathbf{l}_j \\
& - \mathbf{l}_0 \left(\frac{\beta_0}{\gamma_1} + \frac{\beta_0\beta_1}{\gamma_1\gamma_2} + \cdots + \frac{\beta_0 \cdots \beta_{n-1}}{\gamma_1 \cdots \gamma_n} \right) \\
& - \mathbf{l}_1 \left(\frac{\beta_1}{\gamma_2} + \cdots + \frac{\beta_1 \cdots \beta_{n-1}}{\gamma_2 \cdots \gamma_n} \right) \\
& \vdots \\
& - \mathbf{l}_{n-1} \frac{\beta_{n-1}}{\gamma_n} \, .
\end{aligned} \tag{3.6}
$$

It is tempting to estimate $||\mathbf{f}_n||$ directly on the basis of (3.4), using an appropriate norm for the 2×2 block matrices. However, the resulting estimate is too generous, as it does not take into account the fundamental special properties of these block matrices.

4. Comparison with three coupled two-term recurrences. In our notation, Greenbaum's gap [11] for the coupled two-term recurrences (1.3) is

$$
\mathbf{f}^{\mathrm{G}}_{n+1} = \mathbf{f}_0 - \sum_{j=0}^{n} \mathbf{l}^{\mathrm{G}}_j \, , \qquad \text{where} \quad \mathbf{l}^{\mathrm{G}}_j \equiv \mathbf{A}\mathbf{h}^{\mathrm{G}}_j + \mathbf{g}^{\mathrm{G}}_j \, , \tag{4.1}
$$

with $\mathbf{g}^{\mathrm{G}}_n$ and $\mathbf{h}^{\mathrm{G}}_n$ denoting the local rounding errors in the computation of the first two recurrences of (1.3), analogously to $\mathbf{g}_n$ and $\mathbf{h}_n$ in (2.1). A comparison of (4.1) with (3.6) is instructive.

We point out that the size of the local rounding errors may be considerably larger in the two three-term recurrences than in the three two-term recurrences; the size of the local error $\mathbf{l}^{\mathrm{G}}_j$ in the step n is essentially bounded by $\mathcal{O}(\epsilon)||\mathbf{A}|| \ \max_{1 \le j \le n} ||\mathbf{x}_j||$ (see [11]), where $||\mathbf{A}||$ denotes the spectral norm of $\mathbf{A}$. In our case, a similar term in the bound for $||\mathbf{l}_n||$ would be multiplied by the factor $(3|\alpha_n|+4|\beta_{n-1}|)/|\gamma_n|$, which can be substantially larger than 1; see section 5 for the specific case of the CG method. Nevertheless, as documented by our numerical experiments in section 6, the difference between the implementations based on the two three-term recurrences (1.1)–(1.2) and those using the three two-term recurrences (1.3) cannot be explained by the size of the local rounding error terms only. The amplification of the local errors due to possibly large multipliers plays a substantial if not decisive role: the additional terms in (3.6) can be similar in size to or even dominate the sum of local rounding errors. If the multipliers become very large, then the two three-term recurrences (1.1)–(1.2) are likely to exhibit a dramatically wider gap than the two-term recurrences (1.3).

Assuming, as in [11], that the updated residuals become eventually negligible, the relations (3.6) and (4.1) determine the ultimate attainable accuracy of the methods based on (1.1)–(1.2) and (1.3), respectively, measured by the norm of the true residuals.

5. Example: CG method. For the following discussion of the size of the multiplicative factors

$$
\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} \qquad (1 \le i \le k)
$$

we restrict ourselves to symmetric positive definite matrices $\mathbf{A}$ and to the CG method. First, for the simplicity of our exposition, we assume exact arithmetic.

The coefficients in the two-term recurrences (1.3) are for CG given by [16]

$$\omega_n = \frac{\langle \mathbf{r}_n, \mathbf{r}_n \rangle}{\langle \mathbf{p}_n, \mathbf{A}\mathbf{p}_n \rangle}, \qquad \psi_n = \frac{\langle \mathbf{r}_{n+1}, \mathbf{r}_{n+1} \rangle}{\langle \mathbf{r}_n, \mathbf{r}_n \rangle}. \tag{5.1}$$

Both ω_n and ψ_n are positive. Without specific knowledge about $\mathbf{A}$ and $\mathbf{r}_0$ we cannot say anything more about their values. More precisely, given any two sequences of positive numbers, $\omega_0, \ldots, \omega_{N-1}$ and $\psi_0, \ldots, \psi_{N-2}$, there is a symmetric positive definite matrix $\mathbf{A}$ and a vector $\mathbf{r}_0$ such that the classical OMin form (the Hestenes–Stiefel (HS) implementation) of the CG method applied to $\mathbf{A}$ with the initial residual $\mathbf{r}_0$ generates the given coefficients; see Theorem 18:3 of Hestenes and Stiefel [16]. This result allows us to construct examples having any given set of multipliers, and thus to find some with very large gaps. On the other hand, if the matrix $\mathbf{A}$ is reasonably well conditioned and if the CG method converges well, then the bounds derived for the multipliers will show that no substantial amplification of the local rounding errors will occur.

It is well known [20, 5, 1] that by eliminating the direction vectors $\mathbf{p}_n$ in (1.3) we obtain the three-term (ORes) variant of the CG method with recurrences of the form (1.1)–(1.2). From the orthogonality of the residuals we receive

$$\alpha_n = \frac{\langle \mathbf{r}_n, \mathbf{A}\mathbf{r}_n \rangle}{\langle \mathbf{r}_n, \mathbf{r}_n \rangle}, \qquad \beta_{n-1} = \gamma_{n-1} \frac{\langle \mathbf{r}_n, \mathbf{r}_n \rangle}{\langle \mathbf{r}_{n-1}, \mathbf{r}_{n-1} \rangle}. \tag{5.2}$$

Using (5.1) and $\gamma_n = -(\alpha_n + \beta_{n-1})$, we see that the coefficients of the two implementations are related by

$$\gamma_n = -\frac{1}{\omega_n} < 0, \qquad \frac{\beta_{n-1}}{\gamma_n} = \frac{\psi_{n-1}\omega_n}{\omega_{n-1}} \geq 0, \qquad \frac{\alpha_n}{\gamma_n} = -1 - \frac{\psi_{n-1}\omega_n}{\omega_{n-1}} \leq -1, \tag{5.3}$$

where $\psi_{-1} = 0$, $\omega_{-1} = 1$. The equality is attained in the last two formulas only if $\mathbf{x}_n = \mathbf{x}$, that is, if we have reached the solution. We conclude that the multiplicative factors in (3.3) have the form

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} = \frac{\omega_k}{\omega_{i-1}} \prod_{j=i}^{k} \psi_{j-1}\,, \tag{5.4}$$

and therefore they may exhibit, in general, an arbitrary behavior.

For a given matrix $\mathbf{A}$ and an initial residual $\mathbf{r}_0$, it is possible to relate the size of the multipliers to the condition number of $\mathbf{A}$ and the convergence of the CG process measured by the norm of the residuals. First, according to Theorem 5:5 in [16],

$$\frac{\langle \mathbf{p}_n, \mathbf{A}\mathbf{p}_n \rangle}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle} < \frac{1}{\omega_n} = |\gamma_n| < \frac{\langle \mathbf{r}_n, \mathbf{A}\mathbf{r}_n \rangle}{\langle \mathbf{r}_n, \mathbf{r}_n \rangle},$$

which yields, with the spectral norm,

$$\frac{1}{\|\mathbf{A}^{-1}\|} = \frac{1}{\sigma_{\min}(\mathbf{A})} < \frac{1}{\omega_n} = |\gamma_n| < \|\mathbf{A}\|. \tag{5.5}$$

Rewriting the multipliers in the form

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} = \frac{\omega_k}{\omega_{i-1}} \frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2},$$

we receive the following bounds:

$$\frac{1}{\kappa(\mathbf{A})}\,\frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2} \le \prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le \kappa(\mathbf{A})\,\frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2}, \tag{5.6}$$

where $\kappa(\mathbf{A})$ is the spectral condition number of the matrix $\mathbf{A}$. Note that

$$\frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2} = \frac{||\mathbf{A}^{1/2}\,\mathbf{A}^{1/2}\,(\mathbf{x}-\mathbf{x}_k)||^2}{||\mathbf{A}^{1/2}\,\mathbf{A}^{1/2}\,(\mathbf{x}-\mathbf{x}_{i-1})||^2} \le \frac{||\mathbf{A}||}{\sigma_{\min}(\mathbf{A})}\,\frac{||\mathbf{x}-\mathbf{x}_k||_{\mathbf{A}}^2}{||\mathbf{x}-\mathbf{x}_{i-1}||_{\mathbf{A}}^2} \le \kappa(\mathbf{A})$$

due to the monotonicity of the $\mathbf{A}$-norm of the error. Consequently,

$$\prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le \kappa^2(\mathbf{A}).$$

As mentioned in section 2, the bound for the size of the local rounding errors $\mathbf{l}_n$ in the two three-term recurrences (1.1)–(1.2) contains the factors $|\alpha_n/\gamma_n|$ and $|\beta_{n-1}/\gamma_n|$. In view of (5.2) and (5.5) we have $0 \le \alpha_n \le \|\mathbf{A}\|$ and $|\gamma_n|^{-1} \le \|\mathbf{A}^{-1}\|$. Using (5.3), we obtain the estimate

$$0 \le \frac{\beta_{n-1}}{\gamma_n} \le \left|\frac{\alpha_n}{\gamma_n}\right| \le \kappa(\mathbf{A}). \tag{5.7}$$

Surprisingly, to establish that the developed bounds remain relevant in the case of finite precision computation we do not need any extra work: the results of [9] and [13] imply that in finite precision arithmetic the following slightly relaxed bounds hold:

$$(1-\vartheta)\,\frac{1}{\kappa(\mathbf{A})}\,\frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2} \le \prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le (1+\vartheta)\,\kappa(\mathbf{A})\,\frac{||\mathbf{r}_k||^2}{||\mathbf{r}_{i-1}||^2}\,, \tag{5.8}$$

$$\prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le (1+\vartheta)\,\kappa^2(\mathbf{A})\,, \tag{5.9}$$

$$\frac{\beta_{n-1}}{\gamma_n} \le \left|\frac{\alpha_n}{\gamma_n}\right| \le (1+\vartheta)\,\kappa(\mathbf{A})\,, \tag{5.10}$$

where $0 \le \vartheta \ll 1$. (Here, we make the usual assumption about the numerical non-singularity of the matrix $\mathbf{A}$; for details see the references mentioned above.) Note, however, that the conclusion we just made is far from trivial. The values of the actually computed recurrence coefficients and of the residual norms may be completely different from their theoretical counterparts. But still, essentially the same bounds hold!

The large size of the upper bounds for ill-conditioned $\mathbf{A}$ suggest that though the size of the local errors may contribute to a possibly large gap between true and updated residuals, the further amplification of the local errors due to large multipliers may have a much stronger effect.

6. Numerical experiments with the CG method. The construction of our numerical experiments follows ideas from [16].

Example 1. We consider $N = 48$ and aim at the following values of the coefficients (5.1) for the classical HS form of the CG method:

$$\begin{aligned} &\omega_0 = \omega_1 = \cdots = \omega_{47} = 1, \\ &\psi_0 = 10, \qquad \psi_1 = \psi_3 = \cdots = \psi_{43} = 0.01, \qquad \psi_2 = \cdots = \psi_{42} = 100, \\ &\psi_{44} = 10^{-2}, \qquad \psi_{45} = 10^{-3}, \qquad \psi_{46} = 10^{-4}. \end{aligned} \tag{6.1}$$

Using the well-known formulas [9]

$$\begin{aligned} \mathbf{T}_{0,0} &= \frac{1}{\omega_0}\,, \\ \mathbf{T}_{i,i} &= \frac{1}{\omega_i} + \frac{\psi_{i-1}}{\omega_{i-1}}\,, \\ \mathbf{T}_{i,i-1} = \mathbf{T}_{i-1,i} &= \frac{\sqrt{\psi_{i-1}}}{\omega_{i-1}}\,, \qquad i = 1, \ldots, N-1, \end{aligned} \tag{6.2}$$

we construct an $N \times N$ symmetric positive definite tridiagonal matrix $\mathbf{T}$ with spectral norm $\|\mathbf{T}\| = 102$ and condition number $\kappa(\mathbf{T}) \approx 2 \times 10^6$ (for $N = 48$). For any unitary $N \times N$ matrix $\mathbf{V}$, the CG method (1.3), (5.1) applied to the system $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} = \mathbf{VTV}^\star$ and $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0 = \mathbf{Ve}_1$ then generates in steps 1 to N the prescribed coefficients ω_j, ψ_j, $j = 0, \ldots, N-1$, and the residual norms

$$\begin{aligned} \|r_j\| &= 10^{1/2} \qquad \text{for } \ j = 1, 3, \ldots, 43, \\ \|r_j\| &= 10^{-1/2} \qquad \text{for } \ j = 2, 4, \ldots, 44, \end{aligned}$$

with $\|r_j\|$ sharply decreasing in the steps 45 through 48. For an initial residual different from $\mathbf{Ve}_1$ the behavior of the residual norms will be different, but we may still expect some oscillations and, consequently, some large multipliers.

We have used the construction described above, choosing $\mathbf{V}$ as the unitary matrix resulting from the QR decomposition of a randomly generated $N \times N$ matrix; in MATLAB notation $[\mathbf{V}, \mathbf{R}] = qr(\text{randn}(N, N))$. Furthermore, we have chosen $\mathbf{x} = (1, \ldots, 1)^\top$, $\mathbf{b} = \mathbf{Ax}$, $\mathbf{x}_0 = \mathbf{o}$, $\mathbf{r}_0 = \mathbf{b}$. Hence, $\mathbf{r}_0 \neq \mathbf{Ve}_1$. Experiments were performed on an Sun Ultra 10 workstation with $\epsilon \approx 1.11 \times 10^{-16}$ using MATLAB 5.0.

Three implementations of the CG method have been compared: except for Figure 9, solid lines always represent results of the classical OMIN or Hestenes–Stiefel (HS) version given by (1.3) and (5.1), dots those of the Rutishauser (R) variant described in [21], and dashed lines those of the ORES implementation of the form (1.1)–(1.2) presented, for example, in [15, p. 143], and denoted here as HY. In the R variant the recurrences are, for $n \geq 0$, of the form

$$\begin{aligned} \Delta\mathbf{r}_n &= (-\mathbf{Ar}_n + \Delta\mathbf{r}_{n-1}\,\eta_{n-1})\,\tau_n^{-1}\,, & \mathbf{r}_{n+1} &= \mathbf{r}_n + \Delta\mathbf{r}_n\,, \\ \Delta\mathbf{x}_n &= (\mathbf{r}_n + \Delta\mathbf{x}_{n-1}\,\eta_{n-1})\,\tau_n^{-1}\,, & \mathbf{x}_{n+1} &= \mathbf{x}_n + \Delta\mathbf{x}_n\,, \end{aligned} \tag{6.3}$$

and they are started with $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$, $\Delta\mathbf{r}_{-1} = \mathbf{o}$, $\Delta\mathbf{x}_{-1} = \mathbf{o}$, and $\eta_{-1} = 0$. The coefficients are computed according to

$$\tau_n = \frac{\langle \mathbf{r}_n, \mathbf{Ar}_n \rangle}{\langle \mathbf{r}_n, \mathbf{r}_n \rangle} - \eta_{n-1}\,, \qquad \eta_n = \tau_n \frac{\langle \mathbf{r}_{n+1}, \mathbf{r}_{n+1} \rangle}{\langle \mathbf{r}_n, \mathbf{r}_n \rangle}. \tag{6.4}$$

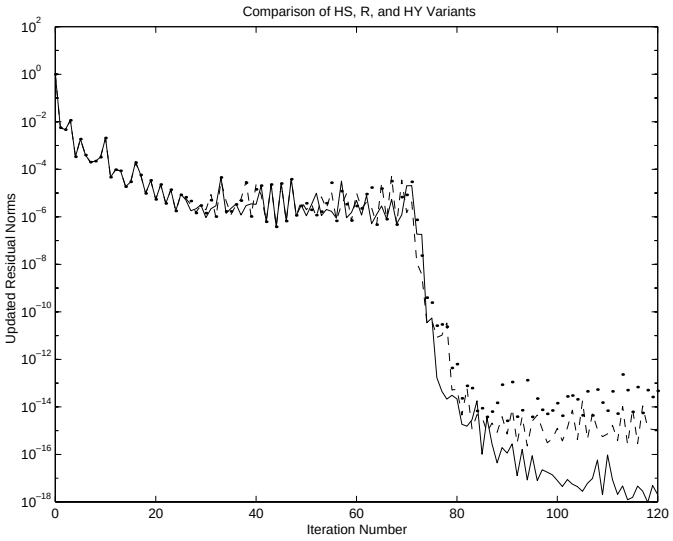


FIG. 1. *Example* 1: *Norms of the updated residuals for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

In the HY variant, the following recurrences are used for $n \geq 0$:

$$\begin{aligned} \mathbf{r}_{n+1} &= \theta_{n+1}(-\mu_{n+1}\mathbf{A}\mathbf{r}_n + \mathbf{r}_n) + (1-\theta_{n+1})\mathbf{r}_{n-1}\,, \\ \mathbf{x}_{n+1} &= \theta_{n+1}(\mu_{n+1}\mathbf{r}_n + \mathbf{x}_n) + (1-\theta_{n+1})\mathbf{x}_{n-1}\,. \end{aligned} \tag{6.5}$$

They are started with $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\theta_1 = 1$, $\mathbf{x}_{-1} = \mathbf{o}$, and $\mathbf{r}_{-1} = \mathbf{o}$, and the coefficients are computed according to

$$\mu_n = \frac{\langle \mathbf{r}_n, \mathbf{r}_n\rangle}{\langle \mathbf{r}_n, \mathbf{A}\mathbf{r}_n\rangle}\,, \qquad \theta_{n+1} = \left(1 - \frac{\mu_{n+1}}{\mu_n}\,\frac{\langle \mathbf{r}_n, \mathbf{r}_n\rangle}{\langle \mathbf{r}_{n-1}, \mathbf{r}_{n-1}\rangle}\,\frac{1}{\theta_n}\right)^{-1}. \tag{6.6}$$

Clearly, the finite precision equivalent of (6.5) can be written in the form (2.1). Consequently, Theorem 3.1 applies, although the bounds for the size of the local errors derived in section 2 have to be modified slightly.

Norms of the updated residuals are compared in Figure 1. We can see the oscillations followed by the fast convergence for n around 70. Of course, theoretically the method should converge in 48 steps, but, as can be explained by the analysis in [9, 13], the convergence is delayed due to roundoff effects. Norms of the true residuals $\|\mathbf{b} - \mathbf{A}\mathbf{x}_n\|$ are shown in Figure 2. Clearly, residual norms of the HY variant stagnate at a significantly worse level than those of the HS variant, as predicted by our analysis.

In Figure 3 the norms of the gaps $\mathbf{f}_n$ we investigated, that is, of the differences between true and updated residuals, are displayed. Note that for the HY variant the gap starts to grow soon, much earlier than one can detect from the two previous figures. Figure 4 shows the behavior of the error norms $\|\mathbf{x} - \mathbf{x}_n\|$. Surprisingly, the differences in the error norms are much less pronounced than those in the true residuals.

Example 2. The second example makes use of the same construction, but now, again for $N = 48$, we aim at

$$\begin{aligned} \omega_0 &= \omega_1 = \cdots = \omega_{47} = 1, \\ \psi_0 &= \psi_1 = \cdots = \psi_{39} = \sqrt{2}, \qquad \psi_{40} = \cdots = \psi_{46} = 2^{-7}, \end{aligned} \tag{6.7}$$

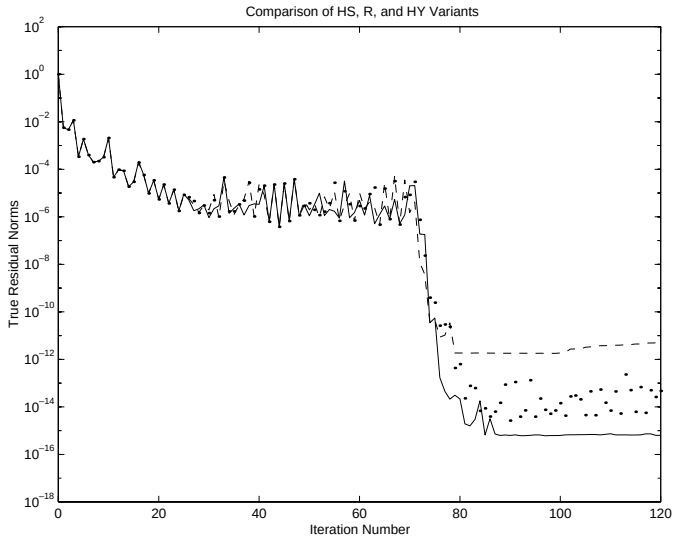


FIG. 2. *Example* 1: *Norms of the true residuals computed as* $\|\mathbf{b}-\mathbf{A}\mathbf{x}_n\|$ *for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

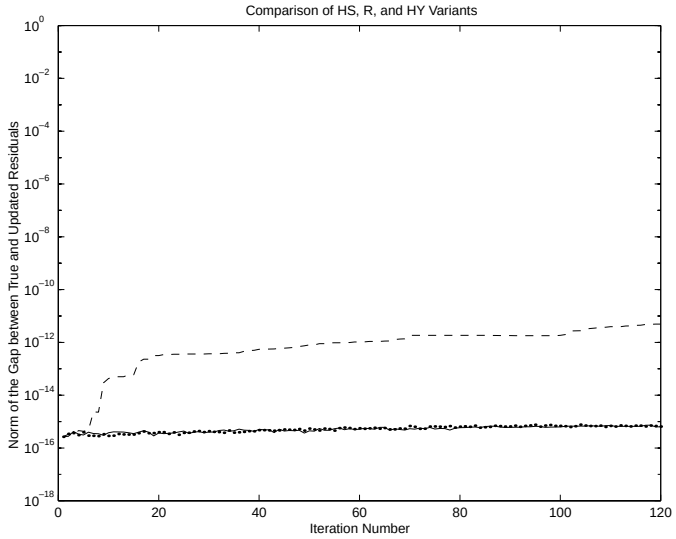


FIG. 3. *Example* 1: *Norms of the differences (gaps)* $\mathbf{f}_n$ *between the true and updated residuals for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

which gives $\|\mathbf{T}\| \approx 4.8$ and $\kappa(\mathbf{T}) \approx 6 \times 10^7$. Again, we consider the system $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} = \mathbf{V}\mathbf{T}\mathbf{V}^\star$, where $\mathbf{V}$ is determined as in Example 1, $\mathbf{x} = (1, \ldots, 1)^\top$, $\mathbf{b} = \mathbf{A}\mathbf{x}$. If we chose $\mathbf{x}_0$ so that $\mathbf{r}_0 = \mathbf{V}\mathbf{e}_1$, we would find residuals with

$$\|r_n\| = (\sqrt{2})^n \qquad \text{for} \qquad n = 1, 2, \ldots, 40$$

and a sharply decreasing norm in the subsequent steps. However, we have again chosen $\mathbf{x}_0$ differently, namely $\mathbf{x}_0 = \mathbf{o}$, so that $\mathbf{r}_0 = \mathbf{b}$. Then we do not find an initially

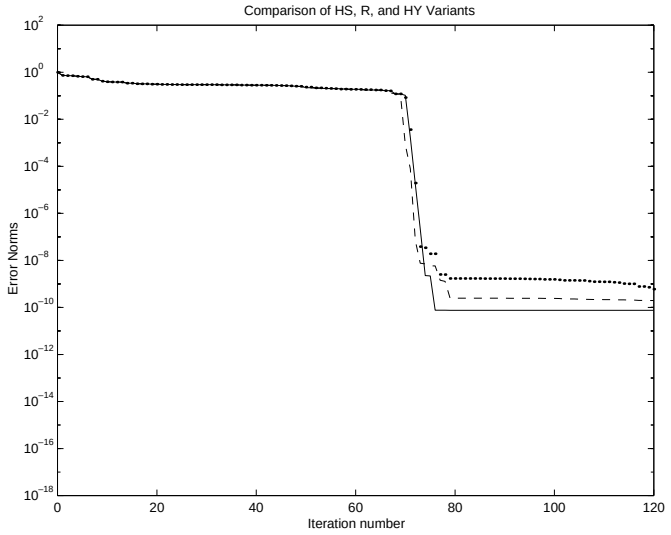


FIG. 4. *Example* 1: *Norms of the errors* $\|\mathbf{x}-\mathbf{x}_n\|$ *for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

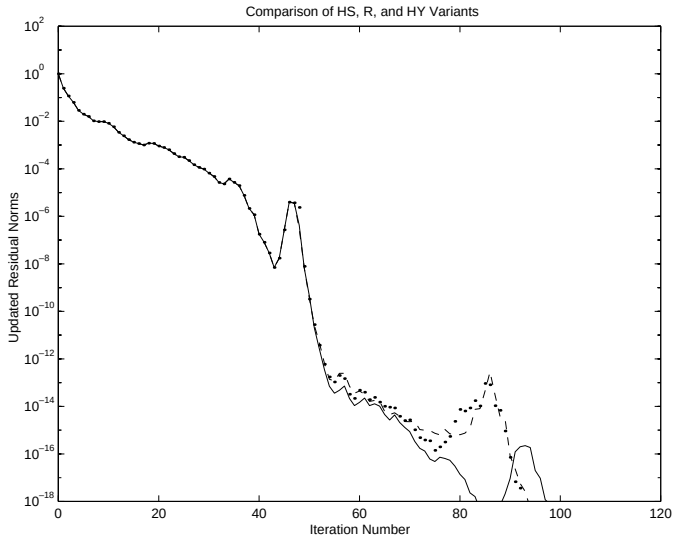


FIG. 5. *Example* 2: *Norms of the updated residuals for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

increasing but rather a quickly decreasing residual norm, both for the updated (see Figure 5) and the true residual (see Figure 6); note the significant oscillation around $n = 45$. The norm of the true residuals of the HY variant stagnates again at a significantly worse level than in the HS variant. Figure 7 shows the norm of the gaps $\mathbf{f}_n$. The differences in the norms of the errors, displayed in Figure 8, are again less pronounced.

To illustrate the contribution of the size of local rounding errors to the gap $\mathbf{f}_n$, we plotted in Figure 9 the size of the coefficients $|\alpha_n/\gamma_n|$, β_n/γ_n and $|1/\gamma_n|$. Clearly, while

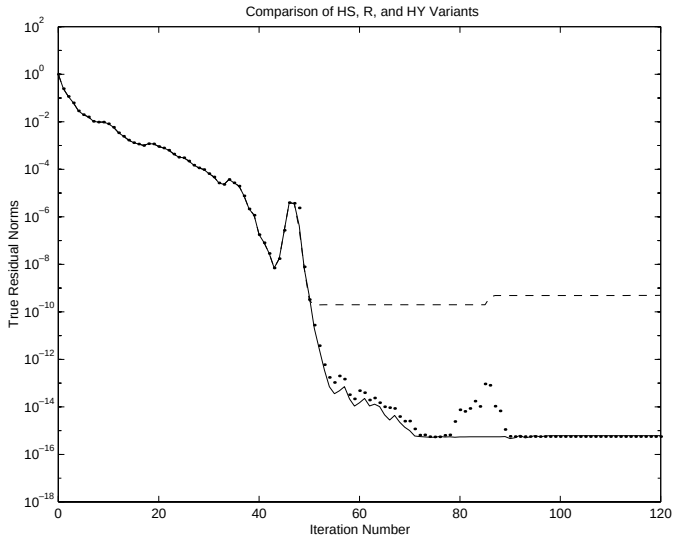


FIG. 6. *Example* 2: *Norms of the true residuals computed as* $\|\mathbf{b}-\mathbf{A}\mathbf{x}_n\|$ *for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

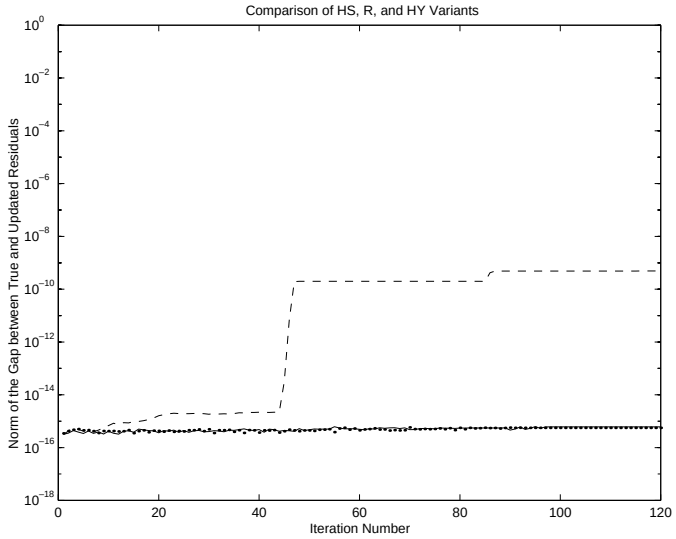


FIG. 7. *Example* 2: *Norms of the differences (gaps)* $\mathbf{f}_n$ *between the true and updated residuals for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

the gap exhibits a loss of accuracy of about six orders of magnitude, the anticipated contribution of the local errors to this gap is not greater than about two orders of magnitude. The disastrous difference between updated and true residuals must therefore be caused by an amplification of the local rounding errors due to large multipliers. In the analogous figure (not shown) for Example 1 the same behavior is slightly less pronounced.

A detailed explanation of the performance of the R variant and of the behavior of the error in all variants requires further work.

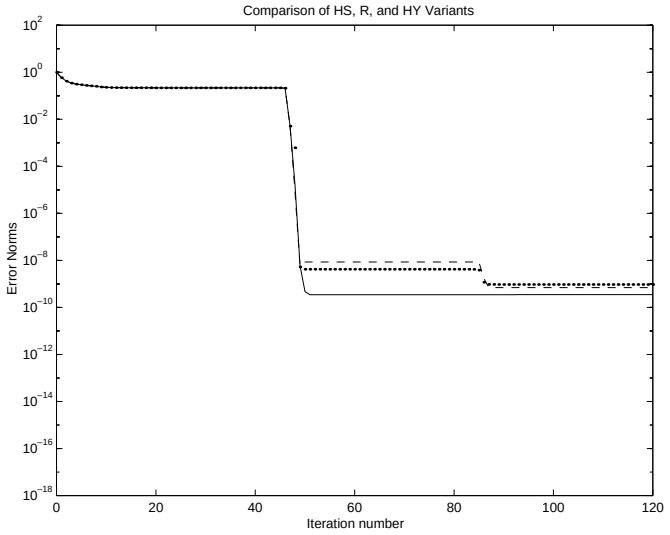


FIG. 8. *Example* 2: *Norms of the errors* $\|\mathbf{x}-\mathbf{x}_n\|$ *for the two-term (HS, solid line), three-term (HY, dashed line), and Rutishauser (R, dots) variants of the CG method.*

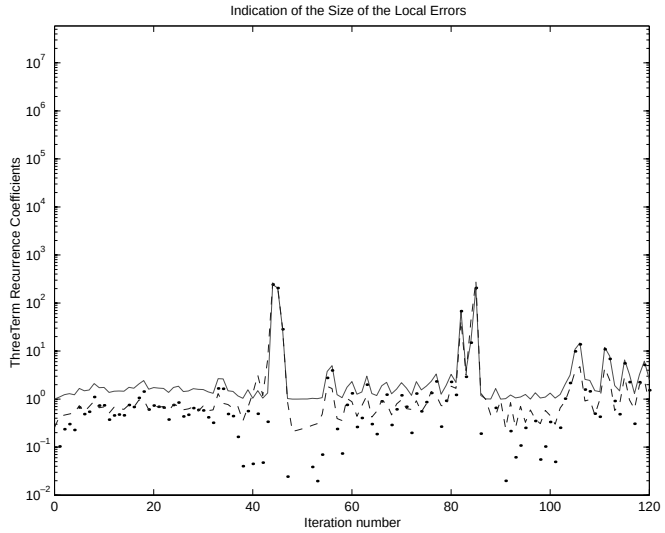


FIG. 9. *Example* 2: *Size of the three-term recurrence coefficients* $|\alpha_n/\gamma_n|$ *(solid line),* β_n/γ_n *(dots) and* $|1/\gamma_n|$ *(dashed line) for the HY variant of the CG method*

.

7. Conclusions. We have explained why the ultimate attainable accuracy measured by the norm of the true residual $\mathbf{b}-\mathbf{A}\mathbf{x}_n$ can be much worse for implementations of Krylov space methods based on the two three-term recurrences (1.1)–(1.2) than for the corresponding implementations based on two-term recurrences of the form (1.3). For example, in the three-term (ORES) version of the CG method, the gap between true and updated residuals is affected not only by the maximum size of the intermediate iterates $||\mathbf{x}_k||$ as in the coupled two-term (OMIN) version, but also by oscillations of the squared norms of the residuals, that is, the quantities $||\mathbf{r}_k||^2/||\mathbf{r}_{i-1}||^2,\ 1 \le i \le k$.

Many well-known algorithms like MINRES and SYMMLQ [20], or the three-term and the coupled two-term versions of the quasi-minimal residual (QMR) method [6, 7], as well as the Rutishauser variant of the CG method are not of the form (1.1)–(1.2) or (1.3). Hence, the results presented in this paper do not apply to them.

Chris Paige suggested another derivation of the results presented in this paper, based entirely on matrix formulations of the algorithms. His approach brings some additional insight into the problem and has potential for further generalization of the results. We hope to report about the results of the joint subsequent work in the near future.

Acknowledgments. The authors would like to thank Anne Greenbaum, Gerard Meurant, Chris Paige, Lisa Perrone, and Miro Rozložník for their helpful comments.

REFERENCES

[1] S. F. Ashby and M. H. Gutknecht, *A matrix analysis of conjugate gradient algorithms*, in Advances in Numerical Methods for Large Sparse Sets of Linear Systems, M. Natori and T. Nodera, eds., Parallel Processing for Scientific Computing 9, Keio University, Yokahama, Japan, 1993, pp. 32–47.

[2] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.

[3] J. A. M. Bollen, *Numerical stability of descent methods for solving linear equations*, Numer. Math., 43 (1984), pp. 361–377.

[4] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of the GMRES method*, BIT, 35 (1995), pp. 308–330.

[5] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, G. A. Watson, ed., Lecture Notes in Math. 506, Springer, Berlin, 1976, pp. 73–89.

[6] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[7] R. W. Freund and N. M. Nachtigal, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. Sci. Comput., 15 (1994), pp. 313–337.

[8] J. F. Grcar, *Analyses of the Lanczos Algorithm and of the Approximation Problem in Richardson's Method*, Ph.D. thesis, Report UIUCDCS-R-81-1074, University of Illinois at Urbana-Champaign, 1981.

[9] A. Greenbaum, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, Linear Algebra Appl., 113 (1989), pp. 7–63.

[10] A. Greenbaum, *Accuracy of computed solutions from conjugate-gradient-like methods*, in Advances in Numerical Methods for Large Sparse Sets of Linear Systems, M. Natori and T. Nodera, eds., Parallel Processing for Scientific Computing 10, Keio University, Yokahama, Japan, 1994, pp. 126–138.

[11] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[12] A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical behaviour of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.

[13] A. Greenbaum and Z. Strakoš, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.

[14] M. H. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numerica, 6 (1997), pp. 271–397.

[15] L. Hageman and D. Young, *Applied Iterative Methods*, Academic Press, Orlando, FL, 1981.

[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–435.

[17] N. J. Higham and P. A. Knight, *Componentwise error analysis for stationary iterative methods*, in Linear Algebra, Markov Chains, and Queueing Models, C. D. Meyer and R. J. Plemmons, eds., IMA Vol. Math. Appl. 48, Springer-Verlag, New York, 1993, pp. 29–46.

[18] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–281.

[19] T. A. Manteuffel, *The Tchebyshev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.

[20] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[21] H. Rutishauser, *Theory of gradient methods*, in Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems, Mitt. Inst. angew. Math. ETH Zürich, Birkhäuser-Verlag, Basel, Switzerland, 1959, pp. 24–49.
[22] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema, *BiCGstab(l) and other hybrid Bi-CG methods*, Numer. Algorithms, 7 (1994), pp. 75–109.
[23] G. L. G. Sleijpen, H. A. van der Vorst, and J. Modersitzki, *The main effects of rounding errors in Krylov solvers for symmetric linear systems*, SIAM J. Matrix Anal. Appl., submitted.
[24] E. Stiefel, *Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme*, Comm. Math. Helv., 29 (1955), pp. 157–179.
[25] H. Woźniakowski, *Numerical stability of the Chebyshev method for the solution of large linear systems*, Numer. Math., 28 (1977), pp. 191–209.
[26] H. Woźniakowski, *Round-off error analysis of iterations for large linear systems*, Numer. Math., 30 (1978), pp. 301–314.
[27] H. Woźniakowski, *Round-off error analysis of a new class of conjugate-gradient algorithms*, Linear Algebra Appl., 29 (1980), pp. 507–529.
[28] D. M. Young and K. C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 230–241

STABLE COMPUTATION WITH THE FUNDAMENTAL MATRIX OF A MARKOV CHAIN*

JESSE L. BARLOW†

Abstract. The short term behavior of a Markov chain can be inferred from its fundamental matrix F. One method of computing the parts of F that are needed is to compute $F\mathbf{y}$ for a given vector $\mathbf{y}$.

It is shown that all forward stable algorithms that solve a particular least squares problem lead to forward stable algorithms for computing $F\mathbf{y}$. This in turn leads to a class of algorithms that compute $F\mathbf{y}$ accurately whenever the underlying problem is well-conditioned. One algorithm from this class is based upon the Grassman–Taksar–Heyman variant of Gaussian elimination. Other such algorithms include one based upon orthogonal factorization and one based upon the conjugate gradient least squares algorithm.

Key words. fundamental matrix, pseudoinverse, group inverse, backward error, conditioning

AMS subject classifications. 65F05, 65F20

PII. S0895479898334538

1. Introduction.

Consider a matrix $A \in \Re^{n\times n}$ of the form

$$A = I - Q, \tag{1.1}$$

where

$$Q \geq 0, \quad Q\mathbf{c} = \mathbf{c}, \quad \mathbf{c} = (1, 1, \ldots, 1)^T \in \Re^n. \tag{1.2}$$

Thus

$$A\mathbf{c} = 0. \tag{1.3}$$

Matrices of the form (1.1) arise in Markov chains. The conditions (1.2) state that Q is row stochastic. We assume that A is irreducible; therefore $\operatorname{rank}(A) = n - 1$.

The stationary vector $\mathbf{p}$ of the Markov chain satisfies

$$\mathbf{p}^T A = 0, \tag{1.4}$$

$$\mathbf{c}^T\mathbf{p} = 1. \tag{1.5}$$

The vector $\mathbf{p}$ determines the long term behavior of the chain. Throughout the paper, we make use of the fact that $\|\mathbf{c}\|_\infty = \|\mathbf{p}\|_1 = 1$.

The short term behavior of the chain is studied by systems analysts. Considerable information about that behavior may be recovered from the fundamental matrix given by

$$F = (A - \mathbf{c}\mathbf{p}^T)^{-1}. \tag{1.6}$$

*Received by the editors February 26, 1998; accepted for publication (in revised form) by D. Calvetti November 22, 1999; published electronically June 3, 2000.
http://www.siam.org/journals/simax/22-1/33453.html

†Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802–6106 (barlow@cse.psu.edu). The research of this author was supported by the National Science Foundation under grants CCR–9424435 and CCR–9732081.

Since F is often large and dense when A is large and sparse, for a given vector $\mathbf{y}$, we compute the value

$$\mathbf{x} = F\mathbf{y} \tag{1.7}$$

rather than compute all of F explicitly.

First, for $k = 1, \ldots, n$, let P_k be a permutation that exchanges the kth and nth column of a matrix. Then define $B_k \in \Re^{n\times(n-1)}$ and $\mathbf{a}_k \in \Re^n$ by

$$AP_k = \begin{pmatrix} B_k & \mathbf{a}_k \end{pmatrix}. \tag{1.8}$$

The class of algorithms that are the main subject of this paper compute (1.7) as follows.

$$\hat{\mathbf{y}} \leftarrow \mathbf{y} - \mathbf{c}(\mathbf{p}^T\mathbf{y}), \tag{1.9}$$

$$\text{solve} \min_{\hat{\mathbf{x}}_k \in \Re^{n-1}} \|\hat{\mathbf{y}} - B_k\hat{\mathbf{x}}_k\|_2, \tag{1.10}$$

$$\hat{\mathbf{x}} = P_k \begin{pmatrix} \hat{\mathbf{x}}_k \\ 0 \end{pmatrix}, \tag{1.11}$$

$$\alpha \leftarrow \mathbf{p}^T(\mathbf{y} - \hat{\mathbf{x}}), \tag{1.12}$$

$$\mathbf{x} \leftarrow \hat{\mathbf{x}} + \alpha\mathbf{c}. \tag{1.13}$$

Step (1.9) assures that $\mathbf{p}^T\hat{\mathbf{y}} = 0$ and that $\hat{\mathbf{y}} \in \text{range}(A)$. Steps (1.12)–(1.13) are just the normalization $\mathbf{p}^T\mathbf{x} = \mathbf{p}^T\mathbf{y}$. Since A is irreducible, it is known that $\text{rank}(B_k) = n - 1$ [1]; thus (1.10) has a unique solution. Moreover, $B_k\hat{\mathbf{x}}_k = \hat{\mathbf{y}}$ is consistent; thus (1.10) always has a zero residual.

The group inverse can also be applied to a vector by making a slight change to the algorithm (1.9)–(1.13). The group inverse, called $A^{\#}$, is the unique matrix such that

$$(1)\ AA^{\#}A = A, \qquad (2)\ A^{\#}AA^{\#} = A^{\#}, \qquad (3)\ AA^{\#} = A^{\#}A. \tag{1.14}$$

To do the computation

$$\mathbf{x} = A^{\#}\mathbf{y}, \tag{1.15}$$

we only change (1.12) to

$$\alpha \leftarrow -\mathbf{p}^T\hat{\mathbf{x}}, \tag{1.16}$$

so that $\mathbf{p}^T\mathbf{x} = 0$.

In our analysis, we assume that $\mathbf{p}$ is the *exact* solution of (1.4)–(1.5). That is, all rounding errors come after $\mathbf{p}$ is computed.

Two terms used in the paper are *backward stable* and *forward stable*. Their definitions correspond to those given by Higham [7, pp. 8–10].

Three measures of the conditioning of A have been used frequently in the literature. In section 2.1, we show that they are all closely related. For convenience, we use the one associated with the Moore–Penrose inverse of A. In section 2.2, we show our main result (Theorem 2.6 and Corollary 2.7), that both forward stable methods and backward stable methods for solving (1.10) lead to forward stable methods for solving (1.7) and (1.15).

Heyman and O'Leary [6] applied a variant of the Grassman–Taksar–Heyman (GTH) algorithm to solve (1.7). In section 3, we show that this algorithm implicitly produces a backward stable solution of (1.10) (Proposition 3.2) and thus always obtains as accurate a solution of (1.7) as can be expected (based on Corollary 2.7). Several other algorithms that also obtain good solutions to (1.7) are simply those that use other forward stable methods to solve (1.10).

2. A stable class of algorithms for computing with the fundamental matrix.

2.1. The condition of the problem. At least three measures have been used for the condition of solving (1.4)–(1.5). These measures lead to reasonable approaches to understanding the condition of (1.7) or (1.15). The three measures are as follows:

(1) The group inverse, $A^{\#}$, given in (1.14). See Meyer [8].
(2) The sep function. See Meyer and Stewart [9] or Stewart and Sun [12, pp. 230–246].
(3) The Moore–Penrose inverse, $A^{\dagger}$. See, for instance, Barlow [1].

We will show that the first two measures can be bounded using $A^{\dagger}$, thereby justifying the use of the third measure.

Noting that A has rank $n-1$, it has the singular value decomposition (SVD)

$$A = \hat{U}\Sigma\hat{V}^T, \quad \hat{U}, \hat{V} \in \Re^{n\times(n-1)}, \tag{2.1}$$

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-1}) \in \Re^{(n-1)\times(n-1)}. \tag{2.2}$$

If $\hat{\mathbf{c}} = \mathbf{c}/\|\mathbf{c}\|_2$ and $\hat{\mathbf{p}} = \mathbf{p}/\|\mathbf{p}\|_2$, then

$$U = \begin{pmatrix} \hat{U} & \hat{\mathbf{p}} \end{pmatrix}, \quad V = \begin{pmatrix} \hat{V} & \hat{\mathbf{c}} \end{pmatrix} \tag{2.3}$$

are both orthogonal matrices.

The Moore–Penrose inverse, $A^{\dagger}$, is given by

$$A^{\dagger} = \hat{V}\Sigma^{-1}\hat{U}^T. \tag{2.4}$$

Thus

$$\|A^{\dagger}\|_2 = \|\Sigma^{-1}\|_2 = \sigma_{n-1}^{-1}.$$

Below we show an important relation between the Moore–Penrose inverse and other characterizations of the Markov chain, generalizing a result in [1].

For this problem, the important separation for the sep function is between the matrix

$$C = \hat{U}^T A \hat{U} \tag{2.5}$$

and the zero eigenvalue of A. Meyer and Stewart [9] show that

$$\operatorname{sep}(C, 0)^{-1} = \|C^{-1}\|_2 = \|A^{\#}\hat{U}\|_2 \leq \|A^{\#}\|_2. \tag{2.6}$$

For the class of matrices given in (1.1), $\|C^{-1}\|_2$ is bounded above and below in terms of $\|A^\dagger\|_2$.

PROPOSITION 2.1. *Let A be the matrix in* (1.1) *with the SVD given by* (2.1)–(2.2) *and let C be defined by* (2.5). *Then*

$$\|A^\dagger\|_2 \le \|C^{-1}\|_2 = \|A^\# \hat{U}\|_2 \le \sqrt{n}\|A^\dagger\|_2. \tag{2.7}$$

Proof. First we note that in terms of the SVD of A in (2.1)–(2.2), we have

$$C = \Sigma \hat{V}^T \hat{U}. \tag{2.8}$$

We now proceed to bound $\|C^{-1}\|_2$. Let U and V be defined by (2.3), and let $W = V^T U$. Note that W is an orthogonal matrix. It can be partitioned

$$W = \begin{matrix} \\ n-1 \\ 1 \end{matrix} \begin{matrix} \begin{matrix} n-1 & \quad 1 \end{matrix} \\ \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \end{matrix} = \begin{pmatrix} \hat{V}^T\hat{U} & \hat{V}^T\hat{\mathbf{p}} \\ \hat{\mathbf{c}}^T\hat{U} & \hat{\mathbf{c}}^T\hat{\mathbf{p}} \end{pmatrix}. \tag{2.9}$$

We have that

$$W_{22} = \hat{\mathbf{c}}^T\hat{\mathbf{p}} = \frac{\mathbf{c}^T\mathbf{p}}{\|\mathbf{c}\|_2\|\mathbf{p}\|_2}.$$

The vector $\mathbf{c}$ is just an n-vector of ones, so $\|\mathbf{c}\|_2 = \sqrt{n}$. Since $\|\mathbf{p}\|_1 = 1$, we have $1/\sqrt{n} \le \|\mathbf{p}\|_2 \le 1$. Using the condition (1.5) yields

$$\frac{1}{\sqrt{n}} \le W_{22} \le 1,$$

so

$$1 = \|W_{22}^{-1}\|_2 \le \sqrt{n}.$$

It is an immediate corollary of the C–S decomposition [5, 11] that W_{11} and W_{22} are singular or nonsingular together and that if they are nonsingular, then

$$\|W_{11}^{-1}\|_2 = \|W_{22}^{-1}\|_2.$$

Therefore, W_{11} is nonsingular and

$$\|W_{11}^{-1}\|_2 \le \sqrt{n}.$$

Since

$$C^{-1} = (V^T U)^{-1}\Sigma^{-1} = W_{11}^{-1}\Sigma^{-1},$$

we have

$$\|C^{-1}\|_2 \le \|W_{11}^{-1}\|_2\|\Sigma^{-1}\|_2 \le \sqrt{n}\|A^\dagger\|_2.$$

The lower bound in (2.7) comes from the observation that

$$\|A^\dagger\|_2 = \|\Sigma^{-1}\|_2 \le \|W_{11}\|_2\|C^{-1}\|_2 \le \|C^{-1}\|_2. \qquad \square$$

Thus, for the remainder of this paper, we consider the value

$$\kappa_2(A) = \|A\|_2\|A^\dagger\|_2 \tag{2.10}$$

as an appropriate condition number for the problem (1.7).

2.2. Characterizing the class of algorithms. The solution of (1.7), (1.15), and (1.10) are all solutions of the equation

$$A\mathbf{x} = \hat{\mathbf{y}}, \tag{2.11}$$

where (1.9) is used to make (2.11) consistent. The solution $\hat{\mathbf{x}}$ of (1.10) solves (2.11) subject to the condition

$$\mathbf{e}_k^T \hat{\mathbf{x}} = 0.$$

All solutions of (2.11) have the form

$$\mathbf{x} = \hat{\mathbf{x}} + \beta \mathbf{c}.$$

Steps (1.12)–(1.13) ensure that we are computing (1.7), and steps (1.16) with (1.13) ensure that we are computing (1.15).

We now show that the system (1.10) is bounded in terms of $\kappa_2(A)$ in (2.10). The proof technique here is similar to that used in [1].

PROPOSITION 2.2. *Let A be an irreducible matrix of the form* (1.1) *the condition* (2.10). *Let $B_k, k = 1, 2, \ldots, n$, be defined by* (1.8). *Then*

$$\|B_k^\dagger\|_2 \leq \sqrt{n}\|A^\dagger\|_2, \tag{2.12}$$

and

$$\kappa_2(B_k) \leq \sqrt{n}\kappa_2(A). \tag{2.13}$$

Proof. Without loss of generality we assume that $P_k = I$ since the 2-norm is invariant under row and column permutations. We then let $B = B_k$.

To bound $\|B^\dagger\|_2$ in terms of $\|A^\dagger\|_2$, consider the underdetermined system

$$B^T \mathbf{z} = \mathbf{r}_B. \tag{2.14}$$

The minimum length solution of (2.14) is

$$\mathbf{z} = B^{\dagger T} \mathbf{r}_B. \tag{2.15}$$

Since B is just the first $n-1$ rows of A, we have that

$$A^T \mathbf{z} = \mathbf{r}_A = \begin{matrix} n-1 \\ 1 \end{matrix} \begin{pmatrix} B^T \mathbf{z} \\ \mathbf{e}_n^T A^T \mathbf{z} \end{pmatrix} = \begin{matrix} n-1 \\ 1 \end{matrix} \begin{pmatrix} \mathbf{r}_B \\ \rho_A \end{pmatrix}.$$

Using (1.3) gives us

$$\mathbf{c}^T A^T \mathbf{z} = \mathbf{c}^T \mathbf{r}_A = 0.$$

Since $\mathbf{c} = (1, 1, \ldots, 1)^T$ we conclude that

$$|\rho_A| = |\mathbf{c}_{n-1}^T \mathbf{r}_B| \leq \|\mathbf{c}_{n-1}\|_2 \|\mathbf{r}_B\|_2, \tag{2.16}$$

where $\mathbf{c}_{n-1} = (1, 1, \ldots, 1)^T \in \Re^{n-1}$. Thus,

$$|\rho_A| \leq \sqrt{n-1}\|\mathbf{r}_B\|_2,$$

and

$$\|\mathbf{r}_A\|_2^2 = \|\mathbf{r}_B\|_2^2 + \rho_A^2 \le n\|\mathbf{r}_B\|_2^2. \tag{2.17}$$

Since A is irreducible,

$$\text{null}(A^T) = \text{null}(B^T) = \text{span}\{\mathbf{p}\}.$$

Therefore,

$$\mathbf{z} = A^{\dagger T}\mathbf{r}_A. \tag{2.18}$$

Combining (2.18), (2.15), and (2.17) leads to

$$\|\mathbf{z}\|_2 = \|B^{\dagger T}\mathbf{r}_B\|_2 = \|A^{\dagger T}\mathbf{r}_A\|_2 \le \sqrt{n}\|A^{\dagger T}\|_2\|\mathbf{r}_B\|_2. \tag{2.19}$$

Since (2.19) holds for any $\mathbf{r}_B$, we have that

$$\|B^{\dagger T}\|_2 \le \sqrt{n}\|A^{\dagger T}\|_2. \tag{2.20}$$

Taking transposes yields (2.12). Use of fact that $\|B\|_2 \le \|A\|_2$ yields (2.13). ∎

Remark 1. The inequality (2.16) can be replaced by

$$|\rho_A| \le \|\mathbf{c}_{n-1}\|_\infty \|\mathbf{r}_B\|_1 = \|\mathbf{r}_B\|_1$$

leading to the conclusion that the inequalities (2.12)–(2.13) can be replaced by

$$\|B_k^\dagger\|_\infty \le 2\|A^\dagger\|_\infty, \quad k = 1, 2, \ldots, n, \tag{2.21}$$

and

$$\kappa_\infty(B_k) \le 2\kappa_\infty(A). \tag{2.22}$$

Both of the inequalities (2.21) and (2.22) are independent of n.

To prove that forward stable algorithms for solving (1.10) lead to forward stable algorithms for solving (1.9)–(1.13), two simple lemmas on error analysis and a third with lower bounds on $\|A^\dagger\|_2$ and $\|A^\dagger\|_\infty$ are necessary.

LEMMA 2.3. *In floating point arithmetic with machine unit* ε_M*, step* (1.9) *satisfies*

$$\|\text{fl}(\hat{\mathbf{y}}) - \hat{\mathbf{y}}\|_\infty \le (n+4)\varepsilon_M\|\mathbf{y}\|_\infty + O(\varepsilon_M^2). \tag{2.23}$$

Proof. First, we note that

$$\|\hat{\mathbf{y}}\|_\infty \le \|\mathbf{y}\|_\infty + \|\mathbf{c}\|_\infty|\mathbf{p}^T\mathbf{y}| \le \|\mathbf{y}\|_\infty + \|\mathbf{c}\|_\infty\|\mathbf{p}\|_1\|\mathbf{y}\|_\infty.$$

Since $\|\mathbf{c}\|_\infty = \|\mathbf{p}\|_1 = 1$, we have

$$\|\hat{\mathbf{y}}\|_\infty \le 2\|\mathbf{y}\|_\infty. \tag{2.24}$$

Using standard error bounds on floating point operation yields

$$|\text{fl}(\mathbf{p}^T\mathbf{y}) - \mathbf{p}^T\mathbf{y}| \le n\varepsilon_M|\mathbf{p}|^T|\mathbf{y}| + O(\varepsilon_M^2).$$

Thus a componentwise bound is

$$|\text{fl}(\hat{\mathbf{y}}) - \hat{\mathbf{y}}| \le \varepsilon_M(2\max\{|\mathbf{y}|, |\hat{\mathbf{y}}|\} + n|\mathbf{p}|^T|\mathbf{y}||\mathbf{c}|) + O(\varepsilon_M^2).$$

Use of the above inequality, the fact that $\|\mathbf{p}\|_1 = \|\mathbf{c}\|_\infty = 1$, the inequality (2.24), and the infinity norm yields (2.23). ∎

LEMMA 2.4. *In floating point arithmetic with machine unit ε_M, the results of steps* (1.12)–(1.13) *satisfy*

$$\|\mathrm{fl}(\mathbf{x}) - \mathbf{x}\|_\infty \leq (4n+13)\varepsilon_M \max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty + O(\varepsilon_M^2). \tag{2.25}$$

Proof. Standard error bounds on floating point arithmetic yield

$$\begin{aligned}
|\mathrm{fl}(\alpha) - \alpha| &\leq (2n+4)(|\mathbf{p}|^T|\mathbf{y}| + |\mathbf{p}|^T|\hat{\mathbf{x}}_\mathbf{k}|)\varepsilon_M + O(\varepsilon_M^2) \\
&\leq (2n+4)\|\mathbf{p}\|_1(\|\mathbf{y}\|_\infty + \|\hat{\mathbf{x}}_k\|_\infty) \\
&\leq (2n+4)(\|\mathbf{y}\|_\infty + \|B_k^\dagger\|_\infty\|\mathbf{y}\|_\infty) \\
&\leq (4n+8)\max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty.
\end{aligned}$$

Thus,

$$|\mathrm{fl}(\mathbf{x}) - \mathbf{x}| \leq \varepsilon_M(n\max\{|\mathbf{x}|, |\hat{\mathbf{x}}_k|\} + (4n+8)\max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty|\mathbf{c}|) + O(\varepsilon_M^2),$$

which implies that

$$\begin{aligned}
\|\mathrm{fl}(\mathbf{x}) - \mathbf{x}\|_\infty \leq \varepsilon_M\ &(\max\{\|\mathbf{x}\|_\infty, \|\hat{\mathbf{x}}_k\|_\infty\} \\
&+(4n+8)\max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty) + O(\varepsilon_M^2).
\end{aligned} \tag{2.26}$$

Since

$$\begin{aligned}
\|\mathbf{x}\|_\infty &\leq \|\hat{\mathbf{x}}_k\|_\infty + \|\mathbf{c}\|_\infty|\alpha| \\
&\leq \|B_k^\dagger\|_\infty\|\hat{\mathbf{y}}\|_\infty + |\mathbf{p}^T\mathbf{y}| + |\mathbf{p}^T\hat{\mathbf{x}}_\mathbf{k}| \\
&\leq 2\|B_k^\dagger\|_\infty\|\hat{\mathbf{y}}\|_\infty + \|\mathbf{y}\|_\infty \leq 4\|B_k^\dagger\|_\infty\|\mathbf{y}\|_\infty + \|\mathbf{y}\|_\infty.
\end{aligned} \tag{2.27}$$

Combining (2.26) and (2.27) yields (2.25). ∎

To compute (1.15), we must do (1.16) instead of (1.12). This leads to the error bounds

$$|\mathrm{fl}(\alpha) - \alpha| \leq n|\mathbf{p}|^T|\mathbf{y}|\varepsilon_M + O(\varepsilon_M^2) \leq n\varepsilon_M\|\mathbf{y}\|_1 + O(\varepsilon_M^2).$$

Thus we may conclude by the same argument that

$$\begin{aligned}
\|\mathrm{fl}(\mathbf{x}) - \mathbf{x}\|_\infty &\leq \varepsilon_M(n\|\mathbf{y}\|_\infty + \max\{\|\mathbf{x}\|_\infty, \|\mathbf{x}_k\|_\infty\}) + O(\varepsilon_M^2) \\
&\leq (n+5)\max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty + O(\varepsilon_M^2).
\end{aligned}$$

We now give lower bounds for $\|A^\dagger\|_2$ and $\|A^\dagger\|_\infty$.

LEMMA 2.5. *Let A be a matrix of the form* (1.1). *Then*

$$\|A^\dagger\|_\infty \geq 0.5, \quad \|A^\dagger\|_2 \geq 1/(1+\sqrt{n}). \tag{2.28}$$

Proof. We note that

$$\|A\|_\infty = \|I - Q\|_\infty \le \|I\|_\infty + \|Q\|_\infty = 2.$$

Thus also, $\|A\|_2 \le 1 + \sqrt{n}$ by standard norm inequalities. Since

$$\|A\|\|A^\dagger\| \ge 1,$$

for any operator norm $\|\cdot\|$, we have (2.28). ∎

The main result of the paper is given in Theorem 2.6. Here we assume that we have an algorithm that solves (1.10) and produces a solution $\mathbf{z}_k$ such that

$$\|\mathbf{z}_k - \hat{\mathbf{x}}_k\|_2 \le \phi(n)\varepsilon_M \|B_k^\dagger\|_2(\|\mathbf{y}\|_2 + \|B_k\|_2\|\hat{\mathbf{x}}_k\|_2) + O(\varepsilon_M^2). \tag{2.29}$$

Several algorithms have been shown to satisfy the assumption (2.29). Since (1.10) has a zero residual, two such algorithms are the Q–R factorization of B_k [14, p. 236] and the corrected seminormal equations method [2]. Other examples are discussed in section 3.

THEOREM 2.6. *Let $A \in \Re^{n\times n}$ be an irreducible matrix of the form* (1.1) *and let $B_k \in \Re^{n\times(n-1)}$ be given by* (1.8). *Suppose that all computations are done in floating point arithmetic with machine unit ε_M. Suppose we use an algorithm to solve* (1.10) *that obtains a vector $\mathbf{z}_k$ satisfying* (2.29) *for some modestly growing function $\phi(n)$. Then using standard methods for computing steps* (1.9), (1.12), *and* (1.13) *the algorithm* (1.9)–(1.13) *produces a computed vector $\mathbf{z}$ for $F\mathbf{y}$ such that*

$$\|\mathbf{z} - \mathbf{x}\|_2 \le \sqrt{n}(\phi(n) + \psi(n))\varepsilon_M \|A^\dagger\|_2(\|\mathbf{y}\|_2 + \|A\|_2\|\hat{\mathbf{x}}_k\|_2) + O(\varepsilon_M^2), \tag{2.30}$$

where $\psi(n) = 10n + 26$. If we substitute (1.16) *for* (1.12) *to compute $A^\#\mathbf{y}$, then we obtain* (2.30) *with $\psi(n) = 4n + 8$.*

Proof. If we add the results of Lemmas 2.3 and 2.4, use the bound on $\|B_k\|_\infty$, and use standard inequalities relating the 2-norm and ∞-norm we obtain

$$\|\mathbf{z} - \mathbf{x}\|_2 \le \varepsilon_M\sqrt{n}\phi(n)\|A^\dagger\|_2(\|\mathbf{y}\|_2 + \|A\|_2\|\mathbf{x}_k\|_2)$$

$$+ (5n + 13)\max\{1, \|B_k^\dagger\|_\infty\}\|\mathbf{y}\|_\infty + O(\varepsilon_M^2).$$

Since $\|B_k^\dagger\|_\infty \le 2\|A^\dagger\|_\infty \le 2\sqrt{n}\|A^\dagger\|_2$ we have

$$\|\mathbf{z} - \mathbf{x}\|_2 \le \varepsilon_M\sqrt{n}\phi(n)\|A^\dagger\|_2(\|\mathbf{y}\|_2 + \|A\|_2\|\mathbf{x}_k\|_2)$$

$$+ (5n + 13)\max\{1, 2\sqrt{n}\|A^\dagger\|_2\}\|\mathbf{y}\|_2 + O(\varepsilon_M^2).$$

From (2.28), $2\sqrt{n}\|A^\dagger\|_2 \ge 1$; thus we may conclude (2.30). An analogous proof obtains (2.30) for computing $A^\#\mathbf{y}$. ∎

The following corollary makes the slightly stronger assumption that a backward stable method is used to solve (1.10). Clearly an analogous result holds for (1.15).

COROLLARY 2.7. *Assume that A and B_k satisfy the hypothesis of Proposition* 2.2. *Suppose we use an algorithm to solve* (1.10) *that obtains a solution $\mathbf{z}_k$ that satisfies*

$$\min_{\mathbf{z}_k \in \Re^{n-1}} \|\hat{\mathbf{y}} + \delta\hat{\mathbf{y}} - (B_k + \delta B_k)\mathbf{z}_k\|_2,$$

where $\eta = \|\delta B_k\|_2\|B_k^\dagger\|_2 \le 1 - \zeta < 1$ and

$$\|\delta B_k\|_2 \le \phi_0(n)\varepsilon_M\|B_k\|_2 + O(\varepsilon_M^2),$$

$$\|\delta\hat{\mathbf{y}}\|_2 \leq \phi_0(n)\varepsilon_M\|\mathbf{y}\|_2 + O(\varepsilon_M^2).$$

Then the algorithm (1.9)–(1.13) *computes a vector* $\mathbf{z}$ *that satisfies* (2.30) *with* $\phi(n) = \phi_0(n)/\zeta$.

Proof. This corollary is simply the result of a classic theorem on perturbation; see, for instance, Björck [3, p. 30, Theorem 1.4.6]. Using the fact that the residual is zero, we have that

$$\|\mathbf{z}_k - \hat{\mathbf{x}}_k\|_2 \leq \phi_0(n)\varepsilon_M \frac{\|B_k^\dagger\|_2}{1-\eta}(\|\mathbf{y}\|_2 + \|B_k^\dagger\|_2\|\hat{\mathbf{x}}_k\|_2) + O(\varepsilon_M^2).$$

The use of Lemma 2.2 and $\zeta = 1 - \eta$ yields the result. ∎

In the next section, we use the result of Corollary 2.7 to show that a new algorithm to compute $F\mathbf{y}$ obtains answers that are as good as can be expected.

3. Why the GTH algorithm and other algorithms yield accurate solutions. Heyman and O'Leary [6] suggest an algorithm that uses the GTH algorithm to solve (1.10) (although they do not state it as such). We now show that floating point implementation of that algorithm satisfies the hypothesis of Corollary 2.7. Thus it has a forward error bound of the form (2.30).

To review, the GTH algorithm is a variant of Gaussian elimination with two differences:

- The elimination proceeds from bottom to top (rather than top to bottom), thus producing the factors

$$A = RL, \tag{3.1}$$

  where R is upper triangular with diagonals equal to -1 and L is lower triangular with $\ell_{11} = 0$.
- Since $A\mathbf{c} = 0$, then $L\mathbf{c} = 0$ also. The diagonal elements of L are computed so as to satisfy this constraint. This modification leads to the componentwise accuracy in computing $\mathbf{p}$ in (1.4)–(1.5) [10].

Heyman and O'Leary [6] then use this factorization to solve (1.10) with the constraint

$$\mathbf{e}_1^T\mathbf{x} = 0. \tag{3.2}$$

The constraint arises naturally out of the factorization. The first row of L must be zero; thus if we solve

$$R\mathbf{v} = \mathbf{y}, \tag{3.3}$$

$$L\mathbf{x} = \mathbf{v}, \tag{3.4}$$

then $\mathbf{e}_1^T\mathbf{v} = 0$ because of consistency.

If we let

$$L = \begin{array}{c} \\ 1 \\ n-1 \end{array}\begin{array}{c} \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{pmatrix} 0 & 0 \\ \boldsymbol{\ell}_1 & \bar{L} \end{pmatrix} \end{array}, \quad R = \begin{array}{c} \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{pmatrix} \mathbf{r}_1 & \bar{R} \end{pmatrix} \end{array}$$

and we choose $k = 1$ in (1.8), then

$$B_1 = \bar{R}\bar{L}. \tag{3.5}$$

The GTH algorithm computes the factorization (3.5) without reference to the first column of A. This key fact means that the GTH algorithm may be used to solve (1.10) with $k = 1$ as given below.

ALGORITHM 3.1 (Heyman–O'Leary procedure for solving least squares (LS) problem (1.10)).

(1) *Factor* B_1 *as in* (3.5).

(2) *Using Gaussian elimination with partial pivoting, factor the upper Hessenberg matrix* $\bar{R}$ *into*

$$\bar{R} = P_R L_R U_R, \quad L_R \in \Re^{n\times n}, U_R \in \Re^{n\times n-1}, \tag{3.6}$$

where

$$L_R \in \Re^{n\times n}, \quad \textit{unit lower triangular,}$$

$$U_R \in \Re^{n\times(n-1)}, \quad \textit{upper trapeziodal (last row zero),}$$

$$P_R, \quad \textit{permutation matrix .}$$

(3) *Using* $\hat{\mathbf{y}}$ *from* (1.9)*, solve*

$$L_R \mathbf{f} = P_R^T \hat{\mathbf{y}}$$

by forward substitution, noting that $\mathbf{e}_n^T \mathbf{f} = 0$ *by consistency. Then solve*

$$U_R \bar{L} \hat{\mathbf{x}}_1 = \mathbf{f}$$

by back substitution followed by forward substitution.

Standard error analysis results about Gaussian elimination yield the following result. Its proof is in the appendix.

PROPOSITION 3.2. *Let Algorithm* 3.1 *be applied to* B_1 *as defined in* (1.8) *and* $\hat{\mathbf{y}}$ *as computed by* (1.9)*. Then, in floating point arithmetic with machine unit* ε_M*, the computed solution* $\hat{\mathbf{x}}_1$ *satisfies*

$$\min_{\hat{\mathbf{x}}_1 \in \Re^{n-1}} \|\hat{\mathbf{y}} + \delta\hat{\mathbf{y}} - (B_1 + \delta B_1)\hat{\mathbf{x}}_1\|_2,$$

where for modestly growing functions $\phi_i(n), i = 1, 2,$

$$\|\delta\hat{\mathbf{y}}\|_2 \le \phi_1(n)\varepsilon_M \|\hat{\mathbf{y}}\|_2 + O(\varepsilon_M^2), \tag{3.7}$$

$$\|\delta B_1\|_2 \le \phi_2(n)\varepsilon_M \|B_1\|_2 + O(\varepsilon_M^2).$$

Proposition 3.2 and Corollary 2.7 imply that Heyman and O'Leary's algorithm obtains a computed value $\mathbf{z}$ for $F\mathbf{y}$ that satisfies (2.30). This it obtains accurate results whenever A is well-conditioned.

In [6], the authors point out that R in (3.1) may be ill-conditioned and give an example where this ill-conditioning seems irrelevant. In fact, the above results show that it does not matter if R is ill-conditioned as long as A is well-conditioned.

From Theorem 2.6 we know that if $\kappa_2(A)$ is modest, any forward stable algorithm for solving (1.10) is a forward stable algorithm for solving (1.7) or (1.15). Other algorithms that could be used include the following:

- The conjugate gradient least squares (CGLS) algorithm applied to (1.10). An analysis by Björck, Elfving, and Strakoš [4] suggests that the conjugate-gradient-based CGLS algorithm could be used to solve (1.10) with forward stability.
- Orthogonal factorization applied to B_k. Well-known backward error bounds from [14, p. 236] on orthogonal factorization would be sufficient.
- Corrected seminormal equation applied to B_k with the upper triangular factor from orthogonal factorization [2].

The choice of algorithm for computing (1.7) depends upon the choice of algorithm to compute $\mathbf{p}$. If the factorization from the GTH algorithm has already been used to compute $\mathbf{p}$, the Heyman–O'Leary algorithm would be a good choice, since the additional work is just Gaussian elimination on a Hessenberg matrix, some back and forward solves, and some vector operations. Orthogonal-factorization-based approaches would be appropriate if the orthogonal factorization has already been done. If iterative methods have been used to compute $\mathbf{p}$, then the CGLS algorithm, with some appropriate preconditioner, would be a good choice to solve (1.10).

Appendix. Proof of Proposition 3.2. We will neglect the errors from the back and forward substitutions since these are strongly backward stable operations [7, Chapter 8].

We need two facts. First, that the computed $\bar{L}$ and $\bar{R}$ satisfy

$$\bar{R}\bar{L} = B_1 + \delta B_0,$$

where

$$\|\delta B_0\|_2 \leq \phi_0(n)\varepsilon_M \|B_1\|_2 + O(\varepsilon_M^2)$$

for a modestly growing function $\phi_0(n)$. This is a property of Gaussian elimination on a diagonally dominant matrix that the GTH version also satisfies. Second, a result of Wilkinson's [13] on Gaussian elimination applied to Hessenberg matrices states that the computed P_R, L_R, and U_R satisfy

$$P_R L_R U_R = R + \delta R, \quad \|\delta R\|_2 \leq \phi_R(n)\varepsilon_M \|R\|_2 + O(\varepsilon_M^2)$$

and

$$P_R L_R \mathbf{f} = \hat{\mathbf{y}} + \delta\hat{\mathbf{y}}, \quad \|\delta\hat{\mathbf{y}}\|_2 \leq \phi_y(n)\varepsilon_M \|\hat{\mathbf{y}}\|_2 + O(\varepsilon_M^2).$$

Thus the two factorizations together yield

$$P_R L_R U_R \bar{L} = B_1 + \delta R\bar{L} + \delta B_1 = B_1 + \delta B_1,$$

where

$$\|\delta B\|_2 \leq \varepsilon_M(\phi_R(n)\|\bar{R}\|_2\|\bar{L}\|_2 + \phi_0(n)\|B_1\|_2).$$

Since $\bar{R}$ and $\bar{L}$ are factors from Gaussian elimination applied to a diagonally dominant matrix, we have that

$$\|\bar{R}\|_2 \leq \sqrt{n}\|R\|_1 \leq 2\sqrt{n},$$

$$\|\bar{L}\|_2 \leq \sqrt{n}\|\bar{L}\|_\infty \leq 2\sqrt{n}\|B_1\|_\infty \leq 2n\|B_1\|_1.$$

Thus,

$$\|\delta B\|_2 \leq \phi_2(n)\varepsilon_M\|B_1\|_2 + O(\varepsilon_M^2),$$

where $\phi_2(n) = \phi_0(n) + 4n\phi_R(n)$. A similar argument leads to the backward error bound

$$P_R L_R \mathbf{f} = \hat{\mathbf{y}} + \delta\hat{\mathbf{y}}_0 + \delta\hat{\mathbf{y}}_1, \tag{A.1}$$

where $\delta\hat{\mathbf{y}}_0$ is the backward error from the substitution and $\delta\hat{\mathbf{y}}_1$ is the error from the consistency of (1.10). Standard error bounds on Gaussian elimination yield

$$\|\delta\hat{\mathbf{y}}_0\|_2 \leq \phi_y(n)\varepsilon_M\|\hat{\mathbf{y}}\|_2 + O(\varepsilon_M^2) \leq (1+\sqrt{n})\phi_y(n)\|\mathbf{y}\|_2 + O(\varepsilon_M^2).$$

Neglecting errors in $\mathbf{p}$, any residual in (1.10) would result from rounding errors in (1.9). Thus, from Lemma 2.3, the equation

$$B_1\mathbf{x}_1 = \hat{\mathbf{y}} + \delta\hat{\mathbf{y}}_1$$

is consistent for some $\delta\hat{\mathbf{y}}_1$ such that

$$\|\delta\hat{\mathbf{y}}_1\|_2 \leq \varepsilon_M\psi_y(n)\|\mathbf{y}\|_2 + O(\varepsilon_M^2),$$

which yields the bound (3.7) with $\phi_1(n) = (1+\sqrt{n})\phi_y(n) + \psi_y(n)$.

Acknowledgments. The author thanks Dianne O'Leary for her suggestions and Daniela Calvetti for her patience.

REFERENCES

[1] J.L. Barlow, *Error bounds for the computation of null vectors with applications to Markov chains*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 797–812.
[2] A. Björck, *Stability analysis of the method of semi-normal equations for linear least squares problems*, Linear Algebra Appl., 88/89 (1987), pp. 31–48.
[3] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.
[4] Å. Björck, T. Elfving, and Z. Strakoš, *Stability of Conjugate Gradient-Type Methods for Solving Linear Least Squares Problems*, Technical Report LiTH-MAT-R-1995-26, Department of Mathematics, Linköping University, Linköping, Sweden, 1995.
[5] C. Davis and W.M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.
[6] D.P. Heyman and D.P. O'Leary, *Overcoming instability in computing the fundamental matrix for a Markov chain*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 534–540.
[7] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
[8] C.D. Meyer, Jr., *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.
[9] C.D. Meyer and G.W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.
[10] C. O'Cinneide, *Entrywise perturbation theory and error analysis for Markov chains*, Numer. Math., 65 (1993), pp. 109–120.
[11] G.W. Stewart, *On the perturbation of psuedo-inverses, projections and linear least squares problems*, SIAM Rev., 19 (1977), pp. 634–662.
[12] G. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[13] J. Wilkinson, *Error analysis of direct methods of matrix inversion*, J. ACM, 8 (1961), pp. 281–330.
[14] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 242–257

CHARACTERIZATION OF CONTINUOUS, FOUR-COEFFICIENT SCALING FUNCTIONS VIA MATRIX SPECTRAL RADIUS*

MARKUS BRÖKER† AND XINLONG ZHOU‡

Abstract. We characterize the existence of continuous solutions of a four-coefficient dilation equation in terms of the usual spectral radius of a matrix. The criteria for the existence of such a solution can be very quickly examined. As a result we give an affirmative answer to a conjecture raised by Colella and Heil in 1992. Moreover, using our criteria we find the smoothest compactly supported four-coefficient orthogonal scaling function and thus the smoothest compactly supported orthonormal wavelet generated by this scaling function.

Key words. dilation equation, joint spectral radius, scaling function, subdivision scheme, wavelet

AMS subject classifications. 26A15, 26A18, 39A10, 42A05, 39B12, 15A18, 65D17

PII. S0895479897323750

1. Introduction.

Let A_0 and A_1 be two $N \times N$ matrices with real entries. The joint spectral radius for two matrices is given by (see [20])

$$\rho(A_0, A_1) = \limsup_{k\to\infty} \left(\sup_{d_i=0,1} ||A_{d_1} \cdots A_{d_k}|| \right)^{\frac{1}{k}}.$$

Moreover, as shown in [20] the joint spectral radius does not depend on the choice of the norm. One of the important applications of this concept is the characterization of the cascade algorithm or the so-called subdivision algorithm for fast generation of curves. In fact, just in this application Daubechies and Lagarias (see [7]) rediscovered the joint spectral radius and recognized its importance. To be more precise, for a given subdivision scheme $\{a_j\}_{j=0}^N$, $a_j \in \mathbb{R}$, we denote the associated polynomial by $a(z) = \sum_{j=0}^N a_j z^j$ (see [2] and [9]). Beginning with one finite sequence of control points $\{x_i^0\}$, we set

$$x_i^k := \sum_\tau a_{i-2\tau} x_\tau^{k-1}, \qquad k = 1, 2, \ldots,$$

where the range of summation will always be clear from the context. For $\psi(x) = 1 - |x|$, $|x| \le 1$; $\psi(x) = 0$ otherwise, the polygon generated by x_i^n can be written as

$$f_n(x) = \sum_i x_i^0 \sum_j a_j^n \psi(2^n(x - i) - j),$$

where $a_j^n = \sum_\tau a_\tau^{n-1} a_{j-2\tau}$ are the coefficients of the polynomial $\prod_{l=0}^{n-1} a(z^{2^l})$. Thus, the question of whether for all given x_j^0 the polygon determined by $\{a_j\}_{j=0}^N$ uniformly converges to a continuous curve is equivalent to the uniform convergence of

*Received by the editors July 8, 1997; accepted for publication (in revised form) by A. Edelman August 30, 1999; published electronically June 3, 2000.
http://www.siam.org/journals/simax/22-1/32375.html
†Informatik-Kooperation, Nevinghoff 26, 48147 Münster, Germany (Markus_Broeker@informatik-kooperation.de).
‡Department of Mathematics, Gerhard-Mercator-University of Duisburg, D-47057 Duisburg, Germany (zhou@riemann.informatik.uni-duisburg.de).

$\sum_j a_j^n \psi(2^n x - j)$. Define S to be the operator given by $Sf(x) = \sum_{i=0}^{N} a_i f(2x - i)$. It is easy to see that $S^n\psi(x) = \sum_i a_i^n \psi(2^n x - i)$. Therefore, the uniform convergence of the polygon sequence is equivalent to the uniform convergence of the iterates $S^n\psi$. If this is the case, then the limit of $S^n\psi$ (say, φ) is a fixed point of the operator S. In other words, φ satisfies

$$\varphi(x) = \sum_{i=0}^{N} a_i \varphi(2x - i). \tag{1}$$

Assume $a(z) = (1+z)b(z)$, $b(z) = \sum_{i=0}^{N-1} b_i z^i$,

$$B_0 = \begin{pmatrix} b_0 & 0 & 0 & \cdots & 0 & 0 \\ b_2 & b_1 & b_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{N-1} & b_{N-2} \end{pmatrix}, B_1 = \begin{pmatrix} b_1 & b_0 & 0 & \cdots & 0 & 0 \\ b_3 & b_2 & b_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & b_{N-1} \end{pmatrix}.$$

The following result can be found in [2], [13], and [21].

THEOREM A. *Under the above notations the iterates $S^n\psi$ uniformly converge to a continuous, compactly supported function φ if and only if*

(i) *$a(1) = 2$ and $a(z) = (1+z)b(z)$;*

(ii) *the joint spectral radius $\rho(B_0, B_1)$ is less than 1.*

There is another connection between the concept of the joint spectral radius and (1), namely, the characterization of compactly supported wavelets in wavelets analysis. Using the dilation equation (1), this problem may be reduced to find compactly supported scaling function, i.e., the solution φ of (1) (see, e.g., [3], [4], [5], [6]). It is known (see, e.g., [2], [5], [6], and [15]) that the existence of such φ and its regularity can be characterized in terms of the joint spectral radius of two $N \times N$ matrices defined by the coefficients $\{a_0, \ldots, a_N\}$ of the dilation equation (1) restricted to an appropriate subspace.

Clearly, (i) and (ii) of the above theorem are also sufficient conditions for the existence of a compactly supported solution φ of (1). The question is now the following: how can we quickly calculate the joint spectral radius for given A_0 and A_1? One can easily see that the direct estimation of this quantity has an exponentially increasing cost. Therefore, it is usually impractical to calculate the value by the definition of the joint spectral radius (see [4], [5], [6], [7], [8], [9], [10], [11], [15], and [19] for a related discussion).

In this paper we study the joint spectral radius constructed by a four-coefficient dilation equation, i.e., $N = 3$. As a result we obtain a computable condition for the existence of a continuous, compactly supported solution of (1). For certain families of 2×2 matrices we can calculate this joint spectral radius exactly (see Lemma 4). Let us assume in the following that $a(1) = 2$. We notice that this is not an essential restriction. In fact, as shown in [6], if (1) has a compactly supported integrable solution, one must have $a(1) = 2^m$ for some natural number m, and then the equation defined by the coefficients $a(z)/2^{m-1}$ has also a compactly supported solution in L_1. Our first result is the following theorem.

THEOREM 1. *For $N = 3$ the iterates $S^n\psi$ uniformly converge to a continuous, compactly supported function φ if and only if $a(z) = (1+z)b(z)$ and the quantity $\sup_{i+j\geq 1,\, i,j\geq 0} \rho(B_0^i B_1^j)$ is less than 1, where $\rho(\cdot)$ is the usual spectral radius of a matrix.*

It is clear that the value in this theorem can be evaluated very quickly. The necessary and sufficient condition for (1) to have a continuous, compactly supported solution can be stated as the following theorem.

THEOREM 2. *For $N = 3$, (1) has a continuous, compactly supported solution if and only if $a(z) = (1 + z^2)(\alpha + (1 - \alpha)z)$ with $0 < \alpha < 1$ or $a(z) = (1 + z)b(z)$ with $\sup_{i+j\geq 1,\ i,j\geq 0} \rho(B_0^i B_1^j) < 1$.*

After proving both results in the next section, we will give some applications concerning the smoothest four-coefficient orthogonal scaling function in the sense of [4] in section 3. As a result we can determine this function and thus the smoothest orthonormal wavelet generated by this scaling function. Moreover, recall that a function f is Hölder continuous with Hölder exponent α if there exists a constant C for which $|f(x) - f(y)| \leq C|x - y|^\alpha$ for all x, y. The four-coefficient orthogonal scaling function φ constructed in [4] (see also [11]), i.e.,

$$\varphi(x) = \frac{1}{5}(3\varphi(2x) + 6\varphi(2x - 1) + 2\varphi(2x - 2) - \varphi(2x - 3)), \tag{2}$$

has a Hölder exponent of approximately 0.60. We can calculate this value exactly, and thus give an affirmative answer to a conjecture raised in [4], i.e., $\rho(B_0, B_1) = \rho(B_0^{12} B_1)^{1/13}$, with $b(z) = (3+3z-z^2)/5$ in our notation. It is interesting to see that Daubechies' scaling function D_4 (see, e.g., [4]) has a Hölder exponent of about 0.550 only, while the smoothest four-coefficient scaling function has an exponent of about 0.628.

Remark. After we finished our paper we learned that, using the optimum unit ball technique (see [17]), Maesumi obtained some similar results and settled the Colella–Heil conjecture [4] mentioned above. Moreover, using the algorithm introduced in [18] Maesumi also finds the smoothest four-coefficient orthogonal scaling function. Our approach is quite different and does not rely on the finiteness conjecture (see [8] and [15]). Furthermore, the results in section 2 lead to the exact expression of the joint spectral radius for all four-coefficient orthogonal scaling functions.

2. Proof of the main results. Let us first recall the following result of Berger and Wang [1] in the following lemma.

LEMMA 3. *For A_0 and A_1 as above there holds*

$$\rho(A_0, A_1) = \limsup_{k\to\infty} \sup_{d_i=0,1} \left(\rho(A_{d_1} A_{d_2} \cdots A_{d_k})\right)^{\frac{1}{k}}. \tag{3}$$

Combining this result with some fundamental calculations we can prove the following lemma.

LEMMA 4. *Suppose C_0 and C_1 are two 2×2 matrices. If $\det(C_0) \leq 0$ or $\det(C_1) \leq 0$, then*

$$\rho(C_0, C_1) = \sup_{i+j\geq 1,\ i,j\geq 0} \left(\rho(C_0^i C_1^j)\right)^{\frac{1}{i+j}}.$$

Proof. Without loss of generality we may assume $\det(C_0) \leq 0$. Furthermore, the continuity of the joint spectral radius (see [12]) tells us that we need only prove our assertion for $\det(C_0) < 0$. With this in mind we write r_1 and r_2 for the eigenvalues of C_0. Hence $\det(C_0) = r_1 \cdot r_2 < 0$. Denote the value on the right-hand side by ρ. The condition on C_0 means that this matrix is similar to a diagonal matrix. Recall that for any regular 2×2 matrix T there holds (see, e.g., [11]) $\rho(C_0, C_1) =$

$\rho(TC_0T^{-1}, TC_1T^{-1})$. We may therefore suppose that C_0 is a diagonal matrix (say, $C_0 = \text{diag}\,(r_1, r_2)$). On the other hand, for the trace of a 2×2 matrix B one has

$$|\ |\mathrm{Tr}(B)| - \rho(B)| \le |\det(B)|^{\frac{1}{2}}.$$

Writing $\gamma := \max(\rho(C_0), \rho(C_1))$, the above inequality implies

$$|\ |\mathrm{Tr}(C_{d_1}C_{d_2}\cdots C_{d_k})| - \rho(C_{d_1}C_{d_2}\cdots C_{d_k})| \le \gamma^k, \tag{4}$$

where $d_j = 0$ or 1. Moreover, since for two square matrices A and B one has $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$, we may write $\mathrm{Tr}(C_{d_1}C_{d_2}\cdots C_{d_k})$ as $\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})$ with some j_τ, such that $\sum_{\tau=1}^m j_\tau = k$. Denote for the moment

$$C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Then $\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m}) = ar_1^{j_m} + dr_2^{j_m}$. Obviously, if $a \cdot d < 0$, then

$$\begin{aligned}|\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})| &\le \max(|r_1|^{j_m-1}, |r_2|^{j_m-1}) \\ &\quad \times|\mathrm{Tr}(C_1^{j_{m-1}}C_0C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-3}}C_0^{j_{m-2}})|.\end{aligned}$$

For the case $a \cdot d \ge 0$ we have

$$\begin{aligned}|\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})| &\le \max(|r_1|^{j_m}, |r_2|^{j_m})\ |\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}})| \\ &= \max(|r_1|^{j_m}, |r_2|^{j_m})\ |\mathrm{Tr}(C_1^{j_1+j_{m-1}}C_0^{j_2}\cdots C_1^{j_{m-3}}C_0^{j_{m-2}})|.\end{aligned}$$

Repeating this process we get for some k' and τ_i, such that $k' + \sum_{i=1}^{\mu}(\tau_i + 1) = k$, the estimate

$$\begin{aligned}|\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})| &\le \max(|r_1|^{k'}, |r_2|^{k'}) \\ &\quad \times|\mathrm{Tr}(C_1^{\tau_1}C_0C_1^{\tau_2}C_0\cdots C_1^{\tau_\mu}C_0)|.\end{aligned} \tag{5}$$

We note that (5) does not include the trivial case

$$|\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})| \le \max(|r_1|^{k''}, |r_2|^{k''})\ |\mathrm{Tr}(C_1^{\tau})|$$

for some k'' and τ. However, as $|\mathrm{Tr}(C_1^{\tau})| \le 2\rho^{\tau}(C_1)$ we conclude from (4) that for this case $\rho(C_0, C_1) = \max(\rho(C_0), \rho(C_1))$. Because of this fact in the following discussion (see (6), (7), and (8)) we shall omit such trivial cases.

To deal with $\mathrm{Tr}(C_1^{\tau_1}C_0C_1^{\tau_2}C_0\cdots C_1^{\tau_\mu}C_0)$, we notice that when $\det(C_1) < 0$, then the matrix C_1 is similar to a diagonal matrix. Thus, applying the above approach to the second factor of the right-hand side of (5), we conclude for some k'' and k_2, such that $k'' + 2k_2 = \mu + \sum_{i=1}^{\mu}\tau_i$,

$$|\mathrm{Tr}(C_1^{\tau_1}C_0C_1^{\tau_2}C_0\cdots C_1^{\tau_\mu}C_0)| \le \gamma^{k''}|\mathrm{Tr}((C_0C_1)^{k_2})|.$$

This together with (4) and (5) implies for $k_1 = k - 2k_2$ the inequality

$$\begin{aligned}|\mathrm{Tr}(C_1^{j_1}C_0^{j_2}\cdots C_1^{j_{m-1}}C_0^{j_m})| &\le \gamma^{k_1}|\mathrm{Tr}((C_0C_1)^{k_2})| \\ &\le \gamma^{k_1}(\rho^{2k_2} + \gamma^{2k_2}).\end{aligned} \tag{6}$$

If $\det(C_1) > 0$, let $\tau := \min(\tau_i \ : \ i = 1, \ldots, \mu)$ and $\gamma_1 := \sup_{j \geq 0}(\rho(C_1^j C_0))^{\frac{1}{j+1}}$. Using the property of the trace we may write for some ν_i

$$\mathrm{Tr}(C_1^{\tau_1} C_0 C_1^{\tau_2} C_0 \cdots C_1^{\tau_\mu} C_0) = \mathrm{Tr}(C_1^{\nu_1}(C_1^\tau C_0) \cdots C_1^{\nu_p}(C_1^\tau C_0)(C_1^\tau C_0)).$$

Obviously, $\det(C_1^\tau C_0) < 0$, and thus the above process implies either

$$|\mathrm{Tr}(C_1^{\nu_1}(C_1^\tau C_0) \cdots C_1^{\nu_p}(C_1^\tau C_0)(C_1^\tau C_0))| \leq \gamma_1^{\tau+1} |\mathrm{Tr}(C_1^{\nu_1}(C_1^\tau C_0) \cdots C_1^{\nu_p}(C_1^\tau C_0))|,$$

or

$$|\mathrm{Tr}(C_1^{\nu_1}(C_1^\tau C_0) \cdots C_1^{\nu_p}(C_1^\tau C_0)(C_1^\tau C_0))| \leq \gamma_1^{2\tau+2} |\mathrm{Tr}(C_1^{\nu_1}(C_1^\tau C_0) \cdots C_1^{\nu_p})|.$$

Hence, for some $j \geq \tau + 1$ and some τ_i' there holds

$$|\mathrm{Tr}(C_1^{\tau_1} C_0 C_1^{\tau_2} C_0 \cdots C_1^{\tau_\mu} C_0)| \leq \gamma_1^j |\mathrm{Tr}(C_1^{\tau_1'} C_0 C_1^{\tau_2'} C_0 \cdots C_1^{\tau_{\mu'}'} C_0)|. \tag{7}$$

We observe that the number of matrix products on the right-hand side of inequality (7) is reduced by at least $\tau + 1$. We may therefore repeat this process for $\tau' := \min(\tau_j' \ : \ j = 1, \ldots, \mu')$ instead of τ to reduce the number of the products on the right-hand side of (7). In this way we obtain finally from (3) and (7) for some ν

$$|\mathrm{Tr}(C_1^{\tau_1} C_0 C_1^{\tau_2} C_0 \cdots C_1^{\tau_\mu} C_0)| \leq \gamma_1^{\mu-\nu-1+\sum_{i=1}^{\mu} \tau_i} (\gamma_1^{\nu+1} + \gamma^{\nu+1}). \tag{8}$$

It follows from (4)–(6) and (8) that if $\det(C_1) \neq 0$, then, as $\rho \geq \gamma_1, \gamma$, one has

$$\rho(C_{d_1} C_{d_2} \cdots C_{d_k}) \leq 4\rho^k.$$

Using Lemma 3 we conclude from the last inequality and the fact that $\rho \leq \rho(C_0, C_1)$ the desired identity in case $\det(C_1) \neq 0$.

It remains to deal with the case $\det(C_1) = 0$. Clearly there exists a sequence ϵ_n with $\lim_{n\to\infty} \epsilon_n = 0$ such that for $C_{1,\epsilon_n} := C_1 + \epsilon_n I$ the determinant of C_{1,ϵ_n} is different from zero, i.e., $\det(C_{1,\epsilon_n}) \neq 0$. The joint spectral radius is continuous, as shown in [12]. Thus, $\lim_{n\to\infty} \rho(C_0, C_{1,\epsilon_n}) = \rho(C_0, C_1)$. Now the above calculations imply for some i_n and j_n

$$\rho(C_0, C_1) = \lim_{n\to\infty} \left(\rho(C_0^{i_n} C_{1,\epsilon_n}^{j_n})\right)^{\frac{1}{i_n + j_n}}.$$

We may assume one of the i_n and $j_n \to \infty$ (or for a subsequence of n). Otherwise, if both i_n and j_n are bounded, there is nothing to do. Let $j_n \to \infty$. Then

$$\rho(C_0^{i_n} C_{1,\epsilon_n}^{j_n}) \leq ||C_0^{i_n}||\ ||C_{1,\epsilon_n}^{j_n}||. \tag{9}$$

Obviously for any $\delta > 0$ there exists p such that

$$||C_1^\tau|| \leq (\rho(C_1) + \delta)^\tau \qquad \forall\ \tau \geq p.$$

Hence, for some $D > 0$ we conclude

$$\begin{aligned} ||C_{1,\epsilon_n}^{j_n}|| &= \left|\left| \sum_{\tau=0}^{j_n} \binom{j_n}{\tau} \epsilon_n^{j_n - \tau} C_1^\tau \right|\right| \\ &\leq \sum_{\tau=0}^{j_n} \binom{j_n}{\tau} \epsilon_n^{j_n-\tau} (\rho(C_1) + \delta)^\tau + D\epsilon_n^{j_n - p} j_n^p \\ &\leq (\rho(C_1) + \delta + \epsilon_n)^{j_n} + D\epsilon_n^{j_n-p} j_n^p. \end{aligned}$$

It follows from this estimate that $\limsup_{n\to\infty} ||C_{1,\epsilon_n}^{j_n}||^{\frac{1}{j_n}} \le \rho(C_1)+\delta$. This inequality is valid for arbitrary $\delta > 0$. Thus,

$$\limsup_{n\to\infty} ||C_{1,\epsilon_n}^{j_n}||^{\frac{1}{j_n}} \le \rho(C_1).$$

Combining this estimate with (9) we get

$$\lim_{n\to\infty} \left(\rho(C_0^{i_n} C_{1,\epsilon_n}^{j_n})\right)^{\frac{1}{i_n+j_n}} \le \max\{\rho(C_0), \rho(C_1)\},$$

from which the equality of our lemma follows. □

From the proof of the above lemma we obtain the following corollary.

COROLLARY 5. *Suppose C_0 and C_1 are two 2×2 matrices.*

(i) *If* $\det(C_0) \le 0$ *and* $\det(C_1) \le 0$, *then*

$$\rho(C_0, C_1) = \max\{(\rho(C_0C_1))^{\frac{1}{2}},\ \rho(C_0),\ \rho(C_1)\}.$$

(ii) *If* $\det(C_0) \le 0$ *and* $\det(C_1) \ge 0$, *then*

$$\rho(C_0, C_1) = \sup_{j\ge 0} \left(\rho(C_1^j C_0)\right)^{\frac{1}{j+1}}.$$

It is clear that (ii) of Corollary 5 implies that for some j' one has

$$\rho(C_0, C_1) = \max\left\{\max_{0\le j\le j'} \left(\rho(C_1^j C_0)\right)^{\frac{1}{j+1}}, \rho(C_1)\right\}. \tag{10}$$

Otherwise we would have $\rho(C_0, C_1) > \rho(C_1)$ and for any $j' > 1$

$$\begin{aligned}\rho(C_0, C_1) &= \sup_{j\ge j'} \left(\rho(C_1^j C_0)\right)^{\frac{1}{j+1}} \\ &\le \sup_{j\ge j'} \left(||C_1^j||^{\frac{1}{j+1}} ||C_0||^{\frac{1}{j+1}}\right),\end{aligned}$$

which, however, implies $\rho(C_0, C_1) \le \rho(C_1)$. The so-called finiteness conjecture (see [6], [7], and [15]) for two matrices asserts that there exists some k such that

$$\rho(A_0, A_1) = \sup_{d_i=0,1} (\rho(A_{d_1} A_{d_1} \cdots A_{d_k}))^{\frac{1}{k}},$$

where $\rho(\cdot)$ on the right-hand side is the usual spectral radius of a matrix. The affirmative answer of this conjecture will imply the existence of a terminating procedure for the estimation of $\rho(A_0, A_1)$, which may lead to find some effectively computable algorithm. This conjecture is still not settled. On the other hand, if A_0, A_1 can be simultaneously upper-triangularized or simultaneously Hermitianized, then $\rho(A_0, A_1) = \max\{\rho(A_0), \rho(A_1)\}$ (see [11]). That is, for this case the finiteness conjecture is true. Lemma 4 gives a partial answer to this conjecture.

Proof of Theorem 1. Using Theorem A we need only show that

$$\sup_{i+j\ge 1,\ i,j\ge 0} \rho(B_0^i B_1^j) < 1 \tag{11}$$

implies $\rho(B_0, B_1) < 1$. To see this we write

$$B_0 = \begin{pmatrix} b_0 & 0 \\ b_2 & b_1 \end{pmatrix} \quad \text{and} \quad B_1 = \begin{pmatrix} b_1 & b_0 \\ 0 & b_2 \end{pmatrix}.$$

Hence, if $b_0 \cdot b_1 \leq 0$ or $b_1 \cdot b_2 \leq 0$, then by Lemma 4

$$\rho(B_0, B_1) = \sup_{i+j \geq 1,\ i,j \geq 0} \left(\rho(B_0^i B_1^j)\right)^{\frac{1}{i+j}}.$$

From (11) the estimate $\rho(B_0, B_1) < 1$ follows. If $b_0 \cdot b_1 > 0$ and $b_1 \cdot b_2 > 0$, then, as $a(1) = 2$, we must have $b_0 + b_1 + b_2 = 1$, which in turn implies that all b_0, b_1, and b_2 are positive. Thus, there exists a norm such that $\max\{||B_0||, ||B_1||\} < 1$. As one always has $\rho(B_0, B_1) \leq \max\{||B_0||, ||B_1||\}$, we conclude $\rho(B_0, B_1) < 1$. □

Proof of Theorem 2. We verify first that if $a(z) = (1 + z^2)(\alpha + (1 - \alpha)z)$ with $0 < \alpha < 1$ or $a(z) = (1 + z)b(z)$ with $\sup_{i+j \geq 1,\ i,j \geq 0} \rho(B_0^i B_1^j) < 1$, then (1) has a continuous, compactly supported solution φ. Clearly, by Theorem 1 we need only show that if $a(z) = (1 + z^2)(\alpha + (1 - \alpha)z)$ with $0 < \alpha < 1$, then (1) has such a solution. To see this we apply Theorem A to the subdivision scheme generated by the coefficients of $(1 + z)(\alpha + (1 - \alpha)z)$ to obtain a $\tilde{\varphi}$. Then the function φ defined by

$$\varphi(x) := \tilde{\varphi}(x) + \tilde{\varphi}(x - 1)$$

satisfies (1).

To verify the opposite we distinguish whether φ is stable or not (see, e.g., [13] for the definition of stability). When φ is stable, then as shown in [13] the iterates $S^n\psi$ converge uniformly to φ. Hence, Theorem 1 tells us that $a(z) = (1 + z)b(z)$ and $\sup_{i+j \geq 1,\ i,j \geq 0} \rho(B_0^i B_1^j) < 1$.

Suppose now φ is not stable. Since $N = 3$ and a_i is real, we conclude from [14] (see Theorem 1 of [14]) that

$$a(z) = (1 + z^2)(\alpha + (1 - \alpha)z).$$

On the other hand, we notice that for any scheme $\{a_j\}_{j=0}^N$ with $a(1) = 2$ there is up to a factor a unique compactly supported distribution satisfying the corresponding equation (1) (see [2] and [6]). Using this we obtain a unique compactly supported distributional solution $\tilde{\varphi}$ of the dilation equation (1) generated by the coefficients of $(1 + z)(\alpha + (1 - \alpha)z)$. As proved in [13] there holds $\varphi(x) = \tilde{\varphi}(x) + \tilde{\varphi}(x - 1)$. Thus, $\tilde{\varphi}$ is continuous. Moreover, by [14], $\tilde{\varphi}$ is stable. Therefore, the iterates $S^n\psi$ defined by the coefficients of $(1 + z)(\alpha + (1 - \alpha)z)$ converge uniformly to $\tilde{\varphi}$, which, however, implies $\max\{|\alpha|, |1 - \alpha|\} < 1$, as Theorem 1 shows. □

3. Applications. A wavelet basis for $L^2(\mathbb{R})$ is an orthonormal basis $\{2^{k/2}\psi(2^k \cdot - j)\}_{k,j \in \mathbb{Z}}$ generated from a single function ψ, the wavelet. The variety of applications of this concept demands to have wavelet bases with specific properties available. It is, therefore, important to have means available by which wavelets with desired properties can be constructed. On the other hand, it is useful to have a ψ with compact support. One method to obtain such ψ is to use the compactly supported solution φ of the dilation equation (1). Using this φ, one can realize the wavelet ψ by

$$\psi(x) = \sum_i (-1)^i a_{1-i} \varphi(2x - i),$$

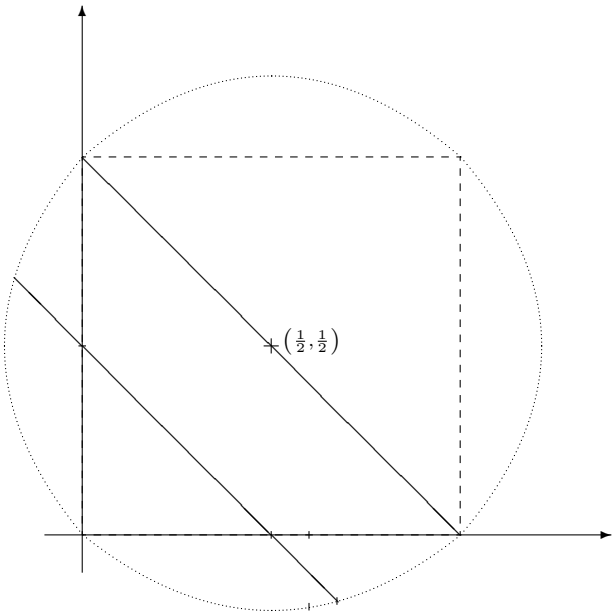


FIG. 1. *The circle* $(b_0 - 1/2)^2 + (b_2 - 1/2)^2 = 1/2$ *with* b_0 *on the horizontal axis. The point which produces the smoothest* φ *is between two points of the circle and its* b_0 *is about* 0.643198.

whenever the scaling function φ defines a multiresolution analysis. Cohen [3] and Lawton [16] have characterized the coefficients $\{a_i\}_{i=0}^N$ that give a multiresolution analysis. For $N = 3$ these are identified as $a_0 + a_2 = 1$, $a_1 + a_3 = 1$, and $(a_0 - 1/2)^2 + (a_3 - 1/2)^2 = 1/2$, except $(a_0, a_3) = (1, 1)$. That means the associated polynomial satisfies $a(z) = (1+z)(b_0+b_1z+b_2z^2)$ and (b_0, b_2) is on the circle (see Figure 1), except $b_0 = b_2 = 1$ (see [11]). As in [4] we shall call such φ an orthogonal scaling function. It is clear that the coefficients of the Daubechies scaling function D_4 and those given in (2) satisfy this criterion. Moreover, the supremum of the Hölder exponent of D_4 is $-\log_2((1 + \sqrt{3})/4)$ (see [6]). The exponent of φ of (2) is approximately 0.60 (see [11]). Furthermore, Colella and Heil conjecture in [11] that this φ has the largest Hölder exponent occurring for the points on the circle of Figure 1. To continue our discussion, let us consider the points (b_0, b_2) on this circle with $b_0 \geq 1$ or $b_2 \geq 1$. Evidently for these points $\rho(B_0, B_1) \geq 1$ holds; hence there is no continuous scaling function for (1) with such coefficients. We shall therefore focus on those points of this circle for which $0 \leq b_0 \leq 1$ and $b_2 \leq 0$. The case $0 \leq b_2 \leq 1$ and $b_0 \leq 0$ can be treated in the same way.

The matrices generated by these coefficients are

$$B_0 = \begin{pmatrix} b_0 & 0 \\ b_2 & b_1 \end{pmatrix} \quad \text{and} \quad B_1 = \begin{pmatrix} b_1 & b_0 \\ 0 & b_2 \end{pmatrix}.$$

Obviously, $b_1 = 1 - b_0 - b_2 \geq 0$. Thus by (10) one obtains for some j'

$$\rho(B_0, B_1) = \max\left\{ \max_{0 \leq j \leq j'} \left(\rho(B_0^j B_1)\right)^{\frac{1}{j+1}}, \rho(B_0) \right\}.$$

On the other hand, it is known (see, e.g., [4]) that in these cases the supremum of the Hölder exponent of the scaling function defined by these coefficients is $-\log_2 \rho(B_0, B_1)$. In what follows we will present an algorithm with which we can calculate the value $\max_{0\le j\le j'}(\rho(B_0^j B_1))^{\frac{1}{j+1}}$ very quickly. Let k be the number satisfying

$$\max_{0\le j\le j'} \left(\rho(B_0^j B_1)\right)^{\frac{1}{j+1}} = \left(\rho(B_0^k B_1)\right)^{\frac{1}{k+1}}.$$

Calculation (see Table 1) shows that in order to find this value one needs at most $k + C$ steps. We will use this algorithm to find the smoothest orthogonal scaling function. We will then compare our result with the smoothness conjecture of Colella and Heil (see [5]).

To begin with we notice that if $b_0 \neq b_1$ (i.e., $b_0 \neq 0.6$), there exists a 2×2 matrix T, such that $TB_0T^{-1} = \operatorname{diag}(b_0, b_1)$. Hence, some calculations yield

$$\begin{aligned}\rho(B_0^j B_1) &= \rho((TB_0T^{-1})^j (TB_1T^{-1})) \\ &= b_0^{j+1}\left(|\alpha+\beta u^{j+1}| + \sqrt{(\alpha+\beta u^{j+1})^2 + \gamma u^{j+1}}\right) \\ &=: b_0^{j+1} F_1(u^{j+1}),\end{aligned} \tag{12}$$

where $2\alpha = b_1/b_0 + b_2/(b_0 - b_1)$, $2\beta = -b_2/(b_0 - b_1)$, $\gamma = -b_2/b_0$, and $u = b_1/b_0$. In case $b_0 = 0.6$, we use the continuity of the spectral radius to obtain from above

$$\begin{aligned}\rho(B_0^j B_1) &= \frac{0.6^{j+1}}{6}\left(|3-(j+1)| + \sqrt{(3-(j+1))^2+12}\right) \\ &=: 0.6^{j+1} G(j+1).\end{aligned} \tag{13}$$

Let us determine next j_0 such that

$$\sup_{j\ge 0}\left(\rho(B_0^j B_1)\right)^{\frac{1}{j+1}} = \max_{0\le j\le j_0}\left(\rho(B_0^j B_1)\right)^{\frac{1}{j+1}}. \tag{14}$$

Case 1. $b_0 = 0.6$. For this case the corresponding scaling function is given by (2). Using (13) one can easily verify that for $y \ge 14$ there holds

$$\frac{d\ln(G(y))^{\frac{1}{y}}}{dy} = \frac{1}{y^2 G(y)}\{-G(y)\ln G(y) + yG'(y)\} < 0.$$

That means

$$\rho(B_0, B_1) = \max\left\{\max_{0\le j\le 14}\left(\rho(B_0^j B_1)\right)^{\frac{1}{j+1}}, \rho(B_0)\right\}.$$

Calculations give

$$\rho(B_0, B_1) = \left(\rho(B_0^{12} B_1)\right)^{\frac{1}{13}} = 0.65967890896$$

and $-\log_2 \rho(B_0, B_1) = 0.6001641146$. This gives an affirmative answer to a conjecture raised by Colella and Heil (see section 1).

Case 2. $0.6 < b_0 \le (1+\sqrt{3})/4$. In this case we have $u < 1$, $\alpha < 0$, and $\beta > 0$. Putting $k_0 := \min\{j \ : \ -\alpha - \beta u^j > 0\}$, we get $F_1(u^{j+1}) = -\alpha - \beta u^{j+1} +$

$\sqrt{(\alpha+\beta u^{j+1})^2+\gamma u^{j+1}}$ $\forall\ j \geq k_0$. We need to calculate $\max_{j\geq k_0}(F_1(u^j))^{1/j}$. Let us consider the set of $j \geq k_0$ satisfying

$$-F_1(u^j)\ln F_1(u^j) + F_1'(u^j)u^j \ln u^j < 0. \tag{15}$$

In what follows we shall prove that if $F_1(0) > 1$, then there exists a number k_2 such that (15) holds $\forall\ j \geq k_2$. Evidently, in this case one has $(F_1(u^j))^{1/j} < \max_{j\geq k_0}(F_1(u^j))^{1/j}\ \ \forall\ j \geq k_2+1$. On the other hand, it is easy to see that for $0 \leq y \leq u^{k_0}$ the function $F_1(y)$ is monotonically decreasing, $F_1(0) = -2\alpha$, and $F_1'(y) = (-\beta F_1(y)+\gamma/2)/\sqrt{(\alpha+\beta y)^2+\gamma y}$. If $F_1(0) \leq 1$, we conclude from the above that $\sup_{j\geq k_0}(F_1(u^j))^{1/j} = 1$. Hence,

$$\rho(B_0,B_1) = \max\left\{\max_{0\leq j\leq k_0}\left(\rho(B_0^jB_1)\right)^{\frac{1}{j+1}}, \rho(B_0)\right\}. \tag{16}$$

Assume now $F_1(0) > 1$. Then for $j \geq k_0$ we have

$$-F_1(u^j)\ln F_1(u^j) + F_1'(u^j)u^j\ln u^j \leq \frac{F_1(u^j)}{\sqrt{(\alpha+\beta u^j)^2+\gamma u^j}} \times\left(-\sqrt{(\alpha+\beta u^j)^2+\gamma u^j}\ln F_1(u^j) - \beta u^j \ln u^j\right).$$

Denote $k_1 := \min\{j\ :\ j \geq k_0 \text{ and } 2|\alpha+\beta u^j| > 1\}$. Since $F_1(u^j) > 2|\alpha+\beta u^j|$ provided $j \geq k_0$, we have $\forall\ j \geq k_1$

$$-\sqrt{(\alpha+\beta u^j)^2+\gamma u^j}\ln F_1(u^j) - \beta j u^j \ln u < -|\alpha+\beta u^j|\ln(2|\alpha+\beta u^j|) - \beta u^j \ln u^j.$$

The term on the right side is a decreasing function of j. Let $k_2 \geq k_1$ be a number such that

$$-|\alpha+\beta u^{k_2}|\ln(2|\alpha+\beta u^{k_2}|) - \beta k_2 u^{k_2}\ln u \leq 0.$$

Then (15) holds $\forall\ j \geq k_2$. In other words, we have

$$\sup_{j\geq k_0}(F_1(u^j))^{1/j} = \max_{k_0\leq j\leq k_2+1}(F_1(u^j))^{1/j},$$

from which relation (14) follows with $j_0 := k_2+1$. Hence,

$$\rho(B_0,B_1) = \max\left\{\max_{0\leq j\leq j_0}\left(\rho(B_0^jB_1)\right)^{\frac{1}{j+1}}, \rho(B_0)\right\}. \tag{17}$$

The above consideration suggests the following algorithm for the computation of $\rho(B_0,B_1)$.

ALGORITHM 1. If $0.6 \leq b_0 \leq (1+\sqrt{3})/4$,
(1) find k_0;
(2) if $F_1(0) \leq 1$, then calculate $\rho(B_0,B_1)$ by using (16);
(3) if $F_1(0) > 1$, then find k_2 and calculate $\rho(B_0,B_1)$ by using (17).

Later we will use this algorithm to find the smoothest φ.

Case 3. $0 < b_0 < 0.6$. Like (12) we have with the same α, β, and γ

$$\begin{aligned}\rho(B_0^jB_1) &= b_1^{j+1}\left(|\alpha u^{j+1}+\beta| + \sqrt{(\alpha u^{j+1}+\beta)^2+\gamma u^{j+1}}\right)\\ &=: b_1^{j+1}F_2(u^{j+1}),\end{aligned}$$

TABLE 1
The number k satisfies $\rho(B_0, B_1) = (\rho(B_0^k B_1))^{\frac{1}{k+1}}$, $--$ means that $\rho(B_0, B_1) = \rho(B_0)$ or $\rho(B_1)$, and $h = -\log_2 \rho(B_0, B_1)$.

b_0	k_2	k	$\rho(B_0, B_1)$	h
0.590	14	11	0.6637328532	0.5913254088
0.600	17	12	0.6596789090	0.6001641146
0.6431	25	22	0.6470557051	0.6280381756
0.6431982	25	22	0.6470546253	0.6280405832
0.6431983	25	23	0.6470546270	0.6280405792
0.6432	25	23	0.6470546638	0.6280404972
0.645	28	25	0.6471430784	0.6278433785
0.6471	36	34	0.6475244074	0.6269935210
0.650	$--$	$--$	0.6500000000	0.6214883767

where $u = b_0/b_1$. We notice that F_2 is obtained by exchanging the positions of α and β in the definition of F_1. Moreover, in case $0 < b_0 < 0.6$ one can easily obtain from (12) and the fact $b_0 + b_1 + b_2 = 1$ that $b_0 < b_1$, $\alpha > 0$, and $\beta < 0$. Hence, we can use the same process as in Case 2 to obtain first k_0, which is $\ln(|\beta|/\alpha)/\ln u$. The number k_1 is now $\min\{j \ : \ j \geq k_0 \text{ and } 2|\alpha u^j + \beta| > 1\}$. Finally, let the number $k_2 \geq k_1$ be such that

$$-|\alpha u^{k_2} + \beta| \ln(2|\alpha u^{k_2} + \beta|) - \alpha k_2 u^{k_2} \ln u \leq 0. \tag{18}$$

The algorithm to compute $\rho(B_0, B_1)$ for these b_0 is the following.

ALGORITHM 2. If $0 < b_0 < 0.6$,

(1) put $k_0 := \min\{j \ : \ -\alpha u^j - \beta > 0\}$;

(2) if $F_2(0) \leq 1$, then calculate $\rho(B_0, B_1)$ by using (16) with k_0 of step (1);

(3) if $F_2(0) > 1$, then find k_2 by (18) and calculate $\rho(B_0, B_1)$ by using (17) with this k_2.

The numerical results (see Table 1) were obtained by using Algorithms 1 and 2, respectively. Figure 2 illustrates the joint spectral radius $\rho(B_0, B_1)$ with (b_0, b_2) on the circle of Figure 1 and $0 \leq b_0 \leq 1$. We recall that D_4 is given by $b_0 = (\sqrt{3}+1)/4$, which is irrational. Figure 2 tells us that it is better in practice to replace this b_0 by some rational number $b' < b_0$. To find that (b_0, b_2) on the circle of Figure 1 which has the smallest joint spectral radius $\rho(B_0, B_1)$, let us denote $f(b_0, j) := F_1(u^j)$, where F_1 is defined in (12) and $u = b_1/b_0$. We notice that, by (12), for fixed b_0 the function $g_{b_0}(y) := (f(b_0, y))^{1/y}$ and its second derivate are continuous $\forall\ y \geq k_0$, where $k_0 := \min\{j \ : \ F_1(u^j) \geq 1\}$. Thus g_{b_0} is only defined for $2|\alpha| > 1$, where α is given by (12). This condition can be fulfilled if $0.6 < b_0 < 0.6471$.

The following result gives a characterization of the smallest joint spectral radius $\rho(B_0, B_1)$ and the corresponding point (b_0, b_2) on the circle of Figure 1.

THEOREM 6. *Let (b_0, b_2) be on the circle of Figure 1 with $b_0 \geq 0$. Then the smallest joint spectral radius $\rho(B_0, B_1)$ is given by b_0, which is the unique solution of*

$$(f(b_0, 23))^{24} = (f(b_0, 24))^{23}, \quad b_0 \in (0.6431, 0.6432). \tag{19}$$

Moreover, calculation shows that b_0 is approximately 0.64319821226 *and the joint spectral radius for this b_0 is about* 0.64705462514.

Proof. For simplicity we shall regard $\rho(B_0, B_1)$ as a function of b_0 (say, $\rho(b_0)$). The function $\rho(x)$ is continuous (see [12]). We prove our assertion by showing that the b_0

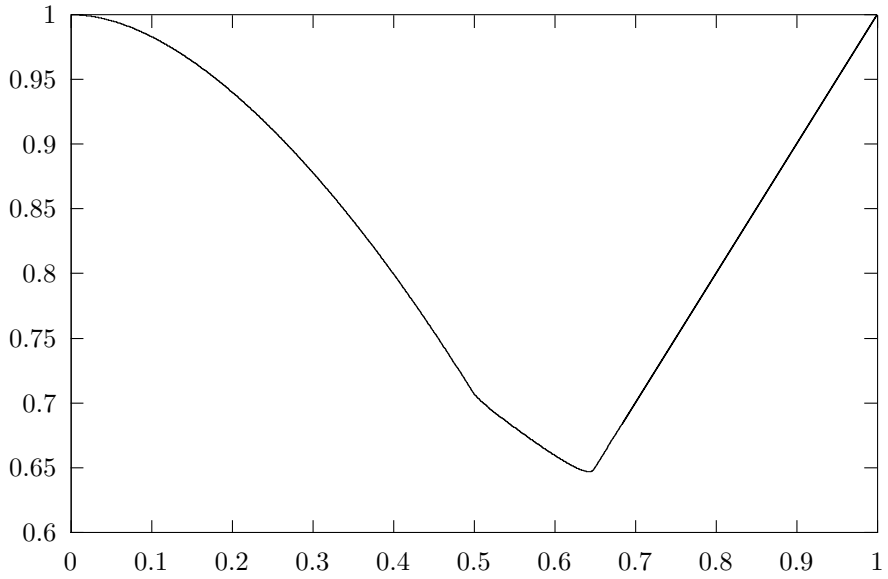


FIG. 2. *The curve of* $\rho(B_0, B_1)$ *with* $0 \le b_0 \le 1$ *and* (b_0, b_2) *on the circle of Figure* 1.

of the theorem is in the interval $(0.6431, 0.6432)$ and that both functions $xf^{1/23}(x, 23)$ and $xf^{1/24}(x, 24)$ are monotone in this interval. Using Algorithm 1 one gets that both end points of this interval do not give the smallest joint spectral radius; thus b_0 of this theorem must satisfy (19).

First, we will show that the b_0, giving the smallest joint spectral radius is not in the interval $[0, 0.6]$. In fact, using Table 1 we need only show that for b_0 in this interval the corresponding joint spectral radius is greater than 0.6471. With this in mind, we notice that, as $\rho(B_0, B_1) \ge b_1 = 1 - b_0 - b_2$, one has for $0 \le b_0 \le 1/2$ the estimate $\rho(B_0, B_1) \ge \sqrt{2}/2 > 0.6471$. For $b_0 \in [0.5, 0.6]$ we use the notation of Algorithm 2 to obtain, with $u = b_0/b_1$ and $j \ge 4$,

$$\begin{aligned}\rho(B_0, B_1) &\ge b_1(2|\alpha u^j + \beta|)^{\frac{1}{j}} \\ &\ge b_1 \left(\gamma u \sum_{i=0}^{j-4} u^i\right)^{\frac{1}{j}} \\ &=: b_1 r(b_0, j).\end{aligned}$$

Obviously, $\rho(B_0, B_1) \ge b_1 \max_{4 \le j \le 13} r(b_0, j)$. By the relation $b_1 = 1 - b_0 - b_2$ one may regard b_1 as a function of b_0 (say, $b_1(b_0)$), which is decreasing. Moreover, for these j the function $r(x, j)$ is increasing on $[0.5, 0.6]$. Thus, one may use the following algorithm to give a lower estimate of $\rho(B_0, B_1)$.

ALGORITHM 3.

(i) For a given positive integer M denote $x_k = 0.5 + 0.1k/M$, $k = 0, 1, \ldots, M + 1$;

(ii) for $k = 0, \ldots, M + 1$ calculate $b_1(x_k)$ and $\max_{4 \le j \le 13} r(x_k, j) =: r(x_k, j_k)$.

It is clear that for $x \in [x_k, x_{k+1}]$ the joint spectral radius $\rho(B_0, B_1)$ with $b_0 = x$ satisfies

$$\rho(B_0, B_1) \ge b_1(x_{k+1}) r(x_k, j_k).$$

Computation shows that with $M = 100$ the right-hand side of the above inequality is greater than 0.6471 for $k = 0, 1, \ldots, M$.

Now Table 1 tells us that if $0.6471 \le x \le 1$, then $\rho(x) \ge x \ge 0.6471 > \rho(0.6431982)$. Thus, the b_0 of this theorem is in the interval $(0.6, 0.6471)$.

Next we investigate the case $0.6 < b < 0.6471$. In what follows we shall prove $g_b''(y) \le 0$ for $y > k_0$. To this end we use the notations of (12) and put $z = u^y$. Using the relationship between g_b and F_1 and straightforward calculation we need only to prove for $0 < z < u^{k_0} < 1$ that

$$I := F_1'(z)(-\ln F_1(z) + \ln z) + F_1''(z) z \ln z \ge 0.$$

As $F_1'(z) = (-\beta F_1(z) + \gamma/2)/\sqrt{(\alpha+\beta z)^2 + \gamma z} \le 0$ and

$$\begin{aligned} F_1''(z) &= \frac{-\beta F_1'(z)}{((\alpha+\beta z)^2+\gamma z)^{\frac{1}{2}}} - \frac{(-\beta F_1(z) + \gamma/2)(\beta(\alpha+\beta z) + \frac{\gamma}{2})}{((\alpha+\beta z)^2+\gamma z)^{\frac{3}{2}}} \\ &= \frac{F_1'(z)}{(\alpha+\beta z)^2+\gamma z}\left\{\left(\beta F_1(z) - \frac{\gamma}{2}\right) - 2\beta(\alpha+\beta z)\right\}, \end{aligned}$$

we get

$$\begin{aligned} I = -F_1'(z)\ln F_1(z) + \frac{F_1'(z)\ln z}{(\alpha+\beta z)^2+\gamma z}\Big\{&((\alpha+\beta z)^2+\gamma z) \\ &+ z\left(\beta F_1(z) - \frac{\gamma}{2}\right) - 2\beta z(\alpha+\beta z)\Big\}. \end{aligned}$$

Noticing that $-\beta F_1(z) + \gamma/2 \le 0$, $-(\alpha+\beta z) \ge 0$, and $F_1(z) \ge 1$ we obtain $I \ge 0$, which verifies our assertion.

The convexity of g_b implies that if for a given b there exist two different j and j' such that

$$(f(b,j))^{\frac{1}{j}} = (f(b,j'))^{\frac{1}{j'}}$$

and

$$\rho(b) = b(f(b,k))^{\frac{1}{k}}, \qquad k = j, j', \tag{20}$$

then $|j - j'| = 1$. Furthermore, for each b_0 there exist at most two j satisfying (20). Let $j(b)$ be the smaller j in (20). We claim that $j(b)$ is a nondecreasing function of b. To see this, we notice that if $j(b) > j(b+\delta_i)$ for $\delta_i > 0$ and $\lim_{i\to\infty}\delta_i = 0$, then, since $j(b)$ is an integer, one must have $\liminf_{i\to\infty} j(b+\delta_i) \le j(b) - 1$ and for $j' := \liminf_{i\to\infty} j(b+\delta_i)$ there holds $\rho(b) = b(f(b,j'))^{1/j'}$, which, however, leads to a contradiction, as $j(b)$ is the smaller j of (20). Thus for small $\delta > 0$ one must have $j(b) \le j(b+\delta)$. Table 1 implies that the j in (20) lies between 13 and 35.

To reach our goal we need also the following assertion: let $j = 13, \ldots, 35$ and $x \in (0.6, 0.6471)$. If $f(x', j) \ge 1$, then $f(x, j+1)$ is a decreasing function of $x \le x'$. Indeed, let y be such that (x, y) is on the circle of Figure 1, $u = (1-x-y)/x$, and $\gamma = -y/x$. Then by the definition of f we have

$$\begin{aligned} f(x,j) &= \frac{1}{2}\left\{-u + \gamma\frac{1-u^j}{1-u}\right\} + \sqrt{\left(\frac{1}{2}\left\{-u+\gamma\frac{1-u^j}{1-u}\right\}\right)^2 + \gamma u^j} \\ &=: A_j(x) + \sqrt{A_j^2(x) + B_j(x)}. \end{aligned} \tag{21}$$

From this we get

$$\frac{\partial}{\partial x} f(x,j) = \frac{2f(x,j)A_j'(x) + B_j'(x)}{2f(x,j) - 2A_j(x)}. \tag{22}$$

Furthermore,

$$2A_j'(x) = \left(-u' + \frac{\gamma'}{\gamma}(1+u)\right) + \frac{\gamma'}{\gamma}(2A_j(x)-1) + \gamma\left(\sum_{i=0}^{j-1} u^i\right)'.$$

Calculation shows that the first term on the right-hand side is $-b_2'/\gamma b_0^2$, which is nonpositive. As u and γ are decreasing with respect to x, we obtain from above that $A_j(x)$ is decreasing provided $2A_j(x) \geq 1$. On the other hand, $f(x,j) \geq 1$ is equivalent to $2A_j(x) + B_j(x) \geq 1$. Thus, the assertion follows from the above calculation and the fact that $A_{j+1}(x) = A_j(x) + B_j(x)/2$. This assertion suggests that we should use the following algorithm.

ALGORITHM 4.

(i) For $\delta > 0$ and M define $x_{k+1} := x_k + \delta$ and $k < M$;

(ii) determine j_k such that

$$\max_{13 \leq j \leq 35} f^{\frac{1}{j}}(x_k, j) = f^{\frac{1}{j_k}}(x_k, j_k) =: d_k;$$

(iii) with this j_k and x_k calculate

$$r_k := x_{k-1} f^{\frac{1}{j'}}(x_k, j'),$$

where $j' = j_k + 1$ when $A_{j_k}(x_k) < 1/2$ and $j' = j_k$ otherwise.

As $\rho(x_k) \geq x_k$, it is clear that $d_k \geq 1$. Thus the above assertion implies that $f(x,j')$ is decreasing for $x \leq x_k$. Now for $x \in [x_{k-1}, x_k]$ we obtain

$$\rho(x) \geq x f^{\frac{1}{j'}}(x, j') \geq x_{k-1} f^{\frac{1}{j'}}(x_k, j').$$

With $\delta = 0.0000001$ we obtain that for $x_0 = 0.6$ and $x_M = 0.6431$ the right-hand side of the last inequality is greater than 0.64705463. This means that the joint spectral radius for $b_0 \in [0.6, 0.6431]$ is greater than the spectral radius with $b_0 = 0.6431982$ (see Table 1). Similarly, with $\delta = 0.00000001$ and $x_0 = 0.6432$ we conclude that the joint spectral radius for $b_0 \in [0.6432, 0.6471]$ is greater than 0.64705463.

Therefore, the smallest joint spectral radius is given by a point in $[0.6431, 0.6432]$. Using Algorithm 1 we can show that for $x \in [0.6431, 0.6432]$ the j of (20) is 23 or 24 (see Table 1). Thus, it remains to prove that, with $j = 23$ or 24, the function $xf^{1/j}(x,j)$ is monotone in this interval, that is, to prove that

$$\frac{\partial}{\partial x}(x f^{\frac{1}{j}}(x,j)) \neq 0, \quad x \in [0.6431, 0.6432]. \tag{23}$$

To see this, assume that there exist $x \in [0.6431, 0.6432]$ and $j = 23$ or 24 such that (23) is not true. Thus

$$j f(x,j) + x \frac{\partial}{\partial x} f(x,j) = 0.$$

Using (21) and (22) we obtain from this that

$$2j(f(x,j) - A_j(x))f(x,j) = -xB_j'(x) - 2xA_j'(x)f(x,j). \tag{24}$$

We recall that $A_j(z)$ is decreasing for $z \le x'$ if $2A_j(x') \ge 1$. For $x' = 0.6432$ one can show that $1.14 \le 2A_j(x') \le 1.16$. Thus, A_j is decreasing in $[0.6431, 0.6432]$. In what follows we shall prove that $\forall\, x \in [0.6431, 0.6432]$ both sides of (24) are different. Thus (23) is valid. To this end, denote $x_0 = 0.6431$, $\delta = 0.000001$, and $x_i = x_0 + \delta i$. It is clear that $x_{100} = 0.6432$. Let $R(x)$ be the right-hand side of (24) and $L(x)$ be the left one. We show $R(x) \ne L(x)$ for $x \in [x_i, x_{i+1}]$, $i = 0, \dots, 99$. $L(x)$ is decreasing; hence $L(x)$ is between $L(x_{i+1})$ and $L(x_i)$ for $x \in [x_i, x_{i+1}]$. To estimate the lower and the upper bound of $R(x)$, we notice that (x, y) is on the circle of Figure 1 and $u = (1 - x - y)/x$, $\gamma = -y/x$. One can easily verify that all these functions are decreasing on $[0.6431, 0.6432]$. Having this in mind, we observe that

$$\begin{aligned} 2A_j'(x) &= -u' + \gamma' \frac{1-u^j}{1-u} + \frac{\gamma u'}{1-u}\frac{1-u^j}{1-u} - \frac{(j)\gamma u^{j-1}u'}{1-u} \\ &= 2A_j(x)\left(\frac{\gamma'}{\gamma} + \frac{u'}{1-u}\right) + \left(\frac{\gamma' u}{\gamma} + \frac{(2u - 1 - j\gamma u^{j-1})u'}{1-u}\right) \\ &=: 2A_j(x)I(x) + I(x,j). \end{aligned}$$

The functions $I(x)$ and $I(x, j)$ are nonpositive for x in $[0.6431, 0.6432]$, as γ', u' are nonpositive and $(2u - 1 - j\gamma u^{j-1}) \ge 0$. Moreover, one has

$$|xI(x)| = (1-u)^{-1}\left(2 + \frac{(2x-1)^2}{|y|(1-2y)}\right)$$

and

$$|xI(x,j)| = ux|I(x)| - x|u'| - x|u'|\frac{j\gamma u^{j-1}}{1-u}.$$

From this we get the lower and the upper bound (say, l_i and u_i) for $R(x)$ with $x \in [x_i, x_{i+1}]$. Computation shows that for $j = 23$ there holds $L(x_i) < l_i$ and $L(x_{i+1}) > u_i$, provided $j = 24$ $(i = 0, \dots, 99)$. □

Acknowledgment. The authors are indebted to the referees for various helpful comments on this paper and for the information concerning the results of Maesumi.

REFERENCES

[1] M. A. Berger and Y. Wang, *Bounded semigroups of matrices*, Linear Algebra Appl., 166 (1992), pp. 21–27.

[2] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli, *Stationary Subdivision*, Mem. Amer. Math. Soc. 453, AMS, Providence, RI, 1991.

[3] A. Cohen, *Ondelettes, analyses multirésolutions et filtres miroirs en quadrature*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 7 (1990), pp. 57–61.

[4] D. Colella and C. Heil, *The characterization of continuous, four-coefficient scaling functions and wavelets*, IEEE Trans. Inform. Theory, 38 (1992), pp. 876–881.

[5] D. Colella and C. Heil, *Characterizations of scaling functions: Continuous solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 496–518.

[6] I. Daubechies and J. C. Lagarias, *Two-scale difference equations* I. *Existence and global regularity of solutions*, SIAM J. Math. Anal., 22 (1991), pp. 1388–1410.

[7] I. Daubechies and J. C. Lagarias, *Two-scale difference equations* II. *Local regularity, infinite products of matrices and fractals*, SIAM J. Math. Anal., 23 (1992), pp. 1031–1079.

[8] I. Daubechies and J. C. Lagarias, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 162 (1992), pp. 227–263.
[9] N. Dyn, J. A. Gregory, and D. Levin, *A 4-point interpolatory subdivision scheme for curve design*, Comput. Aided Geom. Design, 4 (1987), pp. 257–268.
[10] G. Gripenberg, *Computing the joint spectral radius*, Linear Algebra Appl., 234 (1996), pp. 43–60.
[11] C. Heil and D. Colella, *Dilation equation and the smoothness of compactly supported wavelets*, in Wavelets: Mathematics and Applications, J. J. Benedetto and M. W. Frazier, eds., Stud. Adv. Math., CRC Press, Boca Raton, FL, 1994, pp. 163–201.
[12] C. Heil and G. Strang, *Continuity of the joint spectral radius: Application to wavelets*, in Linear Algebra for Signal Processing, A. Bojanczyk and G. Cybenko, eds., IMA Vol. Math. Appl. 69, Springer-Verlag, New York, 1995, pp. 51–61.
[13] R. Q. Jia, *Subdivision schemes in L_p spaces*, Adv. Comput. Math., 3 (1995), pp. 309–341.
[14] R. Q. Jia and J. Z. Wang, *Stability and linear independence associated with wavelet decompositions*, Proc. Amer. Math. Soc., 117 (1993), pp. 1115–1124.
[15] J. C. Lagarias and Y. Wang, *The finiteness conjecture for the generalized spectral radius of a set of matrices*, Linear Algebra Appl., 214 (1995), pp. 17–42.
[16] W. Lawton, *Necessary and sufficient conditions for constructing orthonormal wavelet bases*, J. Math. Phys., 32 (1991), pp. 57–61.
[17] M. Maesumi, *Optimum unit ball for joint spectral radius: An example from four-coefficient MRA*, in Approximation Theory VIII, L. L. Schumaker, ed., Ser. Approx. Decompos. 6, World Scientific, River Edge, NJ, 1995, pp. 267–274.
[18] M. Maesumi, *Joint spectral radius and Hölder regularity of wavelets*, Comput. Math. Appl., 40 (2000), pp. 145–155.
[19] C. A. Micchelli and H. Prautzsch, *Uniform refinement of curves*, Linear Algebra Appl., 114/115 (1989), pp. 841–870.
[20] G. C. Rota and G. Strang, *A note on the joint spectral radius*, Indag. Math., 22 (1960), pp. 379–381.
[21] Y. Wang, *Two-scale dilation equations and the cascade algorithm*, Random Comput. Dynam, 3 (1995), pp. 289–307.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 258–275

A GEOMETRIC APPROACH TO THE CARLSON PROBLEM*

ALBERT COMPTA† AND JOSEP FERRER†

Abstract. The possible observability indices of an observable pair of matrices, when supplementary subpairs are prescribed, are characterized when the "quotient" one is nilpotent. The geometric techniques used are also valid in the classical Carlson problem for square matrices.

Key words. Carlson problem, supplementary pairs of matrices, (C, A)-invariant subspaces, block similarity, Littlewood–Richardson sequences, Brunovsky–Kronecker reduced form

AMS subject classifications. 15A04, 15A21, 93B10, 93B27

PII. S0895479898349148

1. Introduction. This work is a contribution to the problem analogous to the Carlson problem, but involves pairs of matrices instead of single square matrices. In addition, it should be emphasized that the geometric techniques used can also be applied to construct explicit solutions (see section 7) and to study the classical Carlson problem (section 2).

Because of our geometric approach, it is convenient to deal with vertical pairs of matrices, corresponding to linear maps defined on a subspace (see [5]). The dual case of horizontal pairs of matrices, corresponding to maps defined modulo a subspace, is more appropriate to matricial techniques (as in [3]).

So pairs of matrices $P = \binom{A}{C}$, where $A : \mathbb{C}^n \longrightarrow \mathbb{C}^n$, $C : \mathbb{C}^n \longrightarrow \mathbb{C}^m$ $(m \leq n)$, are considered with the following equivalence relation (named "block-similarity" in [8] or "equivalence" in [11]), which generalizes the usual similarity between square matrices: P and P' are block-similar if

$$P' \equiv \begin{pmatrix} A' \\ C' \end{pmatrix} = \begin{pmatrix} Q & S \\ 0 & T \end{pmatrix} \begin{pmatrix} A \\ C \end{pmatrix} Q^{-1}$$

or, equivalently,

$$A' = Q(A + FC)Q^{-1}, \quad C' = TCQ^{-1},$$

where Q and T are nonsingular, and $F = Q^{-1}S$. Throughout the paper, the letters BK (from Brunovsky–Kronecker) will denote the invariants, reduced canonical form, etc., relative to this equivalence relation (see, for example, [8, pp. 96–209] or [5, p. 52]).

With this notation, the general Carlson problem for pairs of matrices can be formulated as follows: characterization of the possible BK-invariants of the pair

$$P = \begin{pmatrix} A \\ C \end{pmatrix} = \left(\begin{array}{cc} A_1 & A_3 \\ 0 & A_2 \\ \hline C_1 & C_3 \\ 0 & C_2 \end{array}\right)$$

*Received by the editors December 11, 1998; accepted for publication (in revised form) by D. Boley January 20, 2000; published electronically June 20, 2000. This work was partially supported by DGICYT grant PB94-1365-C03-03. This work was partially presented in the ILAS Conference, Madison, WI, 1998 and in the SSC'98 Conference, Nantes, France, 1998.

http://www.siam.org/journals/simax/22-1/34914.html

†Departament de Matemàtica Aplicada I, Universitat Politècnica de Catalunya, Diagonal 647, 08028 Barcelona, Spain (compta@ma1.upc.es, ferrer@ma1.upc.es).

when A_3, C_3 vary, if the pairs (or equivalently its BK-invariants) $P_1 = \binom{A_1}{C_1}$ and $P_2 = \binom{A_2}{C_2}$ are fixed.

In system theory, this problem arises in a natural way, for example, when two systems are composed in a "simple cascade" (see [8], [1]).

Baragaña and Zaballa [2] characterize these possible BK-invariants for the particular case when P_2 is observable. Here, the "supplementary" (see Remark 4.7) particular case is considered, when P_2 is an endomorphism (i.e., $C_2 = 0$). When it has a single eigenvalue, Theorem 3.1 gives implicit and explicit characterizations of the possible BK-indices of P, the former being in some sense analogous to the existence of Littlewood–Richardson sequences for the classical Carlson problem. The proof of these implicit characterizations is constructive, so that some examples of explicit solutions P are included in the last section.

Section 2 contains a geometric approach to the classical Carlson problem, which is taken as a motivation of the techniques used in this paper. Section 3 contains the precise definitions and statement of the main theorem (Theorem 3.1), whose proof is delayed until section 5 (necessity) and section 6 (sufficiency), after a geometric reformulation of the problem (Corollary 4.5) in section 4. Some examples are presented in section 7.

In this paper, X will be a finite-dimensional vector space over the complex numbers $\mathbb{C}$, and Y, W, ... will denote vector subspaces of X. If $B \subset X$ is a subset, $[B]$ will be the subspace spanned by the vectors in B. A basis B of X will be called *adapted* to the subspaces Y, W, ... if $B \cap Y$, $B \cap W$, ... are respective bases.

$\mathbb{C}^{p\times q}$ means the set of complex matrices having p rows and q columns. $\mathbb{C}^{p\times q} \times \mathbb{C}^{p'\times q'}$ means the set of vertical pairs of matrices, the one at the top being of $\mathbb{C}^{p\times q}$ and the one at the bottom of $\mathbb{C}^{p'\times q'}$.

In the paper, a *partition*

$$a = (a_1, a_2, \dots, a_{\ell(a)}, 0, \dots, 0)$$

will be a finite nonincreasing sequence of nonnegative integers

$$a_1 \geq a_2 \geq \cdots \geq a_{\ell(a)} > 0,$$

where $\ell(a)$ is called its *length*. We note $|a| = a_1 + a_2 + \cdots + a_{\ell(a)}$ (named its *weight*).

Its *conjugate* partition (see [7, p. 54]) $a^* = (a_1^*, a_2^*, \dots)$ is defined by means of

$$a_j^* = \#\{1 \leq i \leq \ell(a) : a_i \geq j\},$$

where the symbol # means "cardinal." Notice that $a_1^* = \ell(a)$, $\ell(a^*) = a_1$, $|a^*| = |a|$, $(a^*)^* = a$.

Given two partitions a and b, symbol $a \prec b$ means $|a| = |b|$ and

$$a_1 + \cdots + a_i \leq b_1 + \cdots + b_i \quad (i \geq 1).$$

The Segre characteristic relative to any square matrix eigenvalue is the partition of the sizes of his Jordan blocks.

2. A geometric approach to the classical Carlson problem. Let us see how the geometric tools used in this paper arise in a natural way in the classical Carlson problem concerning square matrices. We recall the key theorem is due to Klein [9], relating the decomposition of p-modules with the existence of so-called LR-sequences. On the other hand, [6] proves the equivalence between the Carlson problem

and the one of invariant factors of the product of polynomial matrices, which in turn is related by [10] with the decomposition of p-modules. To summarize, we have the following well known result which reduces the Carlson problem to the existence of LR-sequences.

THEOREM 2.1. *Let there be three partitions*

$$\begin{aligned} \omega &= (\omega_1, \omega_2, \dots), \quad |\omega| = n, \\ w &= (w_1, w_2, \dots), \quad |w| = d, \\ b &= (b_1, b_2, \dots), \quad |b| = n - d. \end{aligned}$$

The following conditions are equivalent:

(I) *For any nilpotent matrices $A_1 \in \mathbb{C}^{d\times d}$ and $A_2 \in \mathbb{C}^{(n-d)\times(n-d)}$ having Segre characteristic w^* and b^*, respectively, there is a matrix $Z \in \mathbb{C}^{d\times(n-d)}$ such that the matrix*

$$A = \begin{pmatrix} A_1 & Z \\ 0 & A_2 \end{pmatrix} \in \mathbb{C}^{n\times n} \tag{2.1}$$

has Segre characteristic ω^*.

(II) *There is a finite sequence of partitions (named after Littlewood–Richardson) $w^0, w^1, \dots, w^s$ ($s = \ell(b)$) such that $w^0 = w$, $w^s = \omega$, and, for all $i, j \geq 1$,*

(a) $|w^j| - |w^{j-1}| = b_j$,

(b) $w_1^{j+1} = w_1^j$; $w_i^j \geq w_i^{j-1} \geq w_{i+1}^j$,

(c) $\sum_{\ell\leq i+1}(w_\ell^{j+1} - w_\ell^j) \leq \sum_{\ell\leq i}(w_\ell^j - w_\ell^{j-1})$.

From a geometric point of view, let us consider an endomorphism $f : X \longrightarrow X$, and $W \subset X$ an invariant subspace (i.e., $f(W) \subset W$). Then, in any basis B of X adapted to W, the matrix of f has the form (2.1) above, where A_1 and A_2 are the matrices of the natural endomorphisms

$$\begin{aligned} \widehat{f} &: W \longrightarrow W, \\ \widetilde{f} &: \frac{X}{W} \longrightarrow \frac{X}{W}, \end{aligned}$$

respectively, in the bases induced by B in a natural way.

If condition (I) holds, let us consider the subspaces

$$W_i^j = \operatorname{Ker} f^i \cap f^{-j}(W) \quad (i, j \geq 0),$$

which can be organized in the following diagram:

$$\begin{array}{ccccccccc}
W & \subset & f^{-1}(W) & \subset & f^{-2}(W) & \subset\cdots\subset & f^{-s}(W) & = & X \\
\| & & \| & & \| & & \| & & \| \\
\operatorname{Ker}\widehat{f}^n & \subset & W_n^1 & \subset & W_n^2 & \subset\cdots\subset & W_n^s & = & \operatorname{Ker} f^n \\
\cup & & \cup & & \cup & & \cup & & \cup \\
\vdots & & \vdots & & \vdots & & \vdots & & \\
\cup & & \cup & & \cup & & \cup & & \cup \\
\operatorname{Ker}\widehat{f}^i & \subset & W_i^1 & \subset & W_i^2 & \subset\cdots\subset & W_n^s & = & \operatorname{Ker} f^i \\
\cup & & \cup & & \cup & & \cup & & \cup \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\cup & & \cup & & \cup & & \cup & & \cup \\
\operatorname{Ker}\widehat{f} & \subset & W_1^1 & = & W_1^2 & =\cdots= & W_1^s & = & \operatorname{Ker} f
\end{array},$$

where $s = \ell(b)$.

Notice that $W_0^j = 0$ for all $j \geq 0$ and

$$\begin{aligned} \omega_i &= \dim \operatorname{Ker} f^i - \dim \operatorname{Ker} f^{i-1}, \\ w_i &= \dim \operatorname{Ker} \widehat{f}^i - \dim \operatorname{Ker} \widehat{f}^{i-1}, \\ b_j &= \dim f^{-j}(W) - \dim f^{-j+1}(W). \end{aligned}$$

Then, it can be proved that condition (II) holds by taking

$$w_i^j = \dim W_i^j - \dim W_{i-1}^j.$$

In fact, condition (II)(a) is trivial, and the other ones are equivalent to the injectivity of the maps

$$\frac{W_{i+1}^j}{W_i^j} \longrightarrow \frac{W_i^{j-1}}{W_{i-1}^{j-1}}, \qquad \frac{W_i^{j+1}}{W_i^j} \longrightarrow \frac{W_{i-1}^j}{W_{i-1}^{j-1}}$$

induced by f.

3. Precise definitions and statement of the main theorem. As a natural generalization of the Carlson problem, let us consider pairs of matrices of the form

$$P = \begin{pmatrix} A \\ C \end{pmatrix} = \left(\begin{array}{cc} A_1 & Z \\ 0 & A_2 \\ \hline C_1 & C_3 \\ 0 & C_2 \end{array}\right),$$

$$P_1 \equiv \begin{pmatrix} A_1 \\ C_1 \end{pmatrix}, \qquad P_2 \equiv \begin{pmatrix} A_2 \\ C_2 \end{pmatrix},$$

where A_1 and A_2 are square matrices. One wonders about the existence of, the way to obtain, etc., matrices Z when P_1, P_2, and C_3 as well as the block-similarity class of P are prescribed. Obviously, in the classical Carlson problem one assumes that $C_1 = 0$, $C_2 = 0$, and $C_3 = 0$, that is to say, P, P_1, and P_2 are endomorphisms.

In this paper, we consider the case when P is observable (and therefore so is P_1) with prescribed observability indices, and P_2 is an endomorphism (i.e., $C_2 = 0$; see Remark 4.7) having only an eigenvalue λ. We recall that the observability indices of P form the dual partition of

$$r_i = \operatorname{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^i \end{pmatrix} - \operatorname{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{pmatrix}$$

and P is observable if

$$\operatorname{rang} \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n.$$

In fact, one can assume that P_1 is a BK-matrix, $C_3 = 0$, and A_2 is a nilpotent Jordan matrix. This remark and the main results in this paper are summarized in the following theorem.

THEOREM 3.1. *Let there be three partitions:*

$$\begin{aligned} R &= (R_1, R_2, \dots), \quad |R| = n, \\ r &= (r_1, r_2, \dots), \quad |r| = d, \\ b &= (b_1, b_2, \dots), \quad |b| = n - d. \end{aligned}$$

The following conditions are equivalent:

(I) *For any observable pair* $P_1 \in \mathbb{C}^{d\times d} \times \mathbb{C}^{r_1\times d}$ *having observability indices* r^*, *any square matrix* $A_2 \in \mathbb{C}^{(n-d)\times(n-d)}$ *having only one eigenvalue* λ, *with Segre characteristic* b^*, *and any matrix* $C_3 \in \mathbb{C}^{r_1\times(n-d)}$ *there is a matrix* $Z \in \mathbb{C}^{d\times(n-d)}$ *such that the pair*

$$P = \left(\begin{array}{cc} A_1 & Z \\ 0 & A_2 \\ \hline C_1 & C_3 \end{array}\right)$$

is observable having observability indices R^*.

(I′) *Condition* (I) *holds in the particular case when* P_1 *is a BK-matrix,* A_2 *is a nilpotent Jordan matrix, and* $C_3 = 0$.

(II) *There is a finite sequence of partitions* $r^0, r^1, \dots, r^s$ $(s = \ell(b))$ *such that* $r^0 = r$, $r^s = R$, *and for all* $i, j \geq 1$

(a) $|r^j| - |r^{j-1}| = b_j$,

(b) $r_1^j = r_1^{j-1}$, $r_i^{j-1} \geq r_{i+1}^j \geq r_{i+1}^{j-1}$,

(c) $\sum_{\ell\geq i+1}(r_\ell^{j+1} - r_\ell^j) \leq \sum_{\ell\geq i}(r_\ell^j - r_\ell^{j-1})$.

(II′) *There is a finite sequence of partitions* $c^0, c^1, \dots, c^s$ $(s = \ell(b))$ *such that* $c^0 = r^*$, $c^s = R^*$, *and for all* $\nu, j, i \geq 1$

(a) $|c^j| - |c^{j-1}| = b_j$,

(b) $\ell(c^j) = r_1$, $c_\nu^{j-1} \leq c_\nu^j \leq c_\nu^{j-1} + 1$,

(c) $\sum_{\eta\in I(i+1,j+1)}(c_\eta^{j+1} - c_\eta^j) \leq \sum_{\eta\in I(i,j)}(c_\eta^j - c_\eta^{j-1})$, *where* $I(i,j) = \{\eta : c_\eta^j \geq i\}$.

(III) (see [3]) $b_1 \leq r_1 = R_1$, $(R^*)_\nu \geq (r^*)_\nu$ $(\nu = 1, 2, \dots)$, *and* $R^* - r^* \prec b^*$, *where* $R^* - r^*$ *is assumed to be reordered to become nonincreasing.*

Remark 3.2. Notice that conditions (II)(a)–(II)(c) are similar to the Littlewood–Richardson ones which appear in the classical Carlson problem (see Theorem 2.1). In fact, (II)(a) and (II)(b) are almost the same, whereas (II)(c) is in some sense "opposite."

Remark 3.3. Condition (III) has been suggested by [3] by means of fully different methods. In fact, it is an immediate consequence of (II′)(a) and (II′)(b). Likewise (see the preceding remark) condition (III), except where $b_1 \leq r_1 = R_1$ (which holds only if $\operatorname{Ker} f \subset W$), is a necessary condition in the classical Carlson problem, but in that case it is not sufficient.

Remark 3.4. When (III) holds, explicit solutions Z can be computed by means of (II), thus being nonequivalent for different sequences $r^0, r^1, \dots, r^s$ (see section 7).

Remark 3.5. Conditions (II)–(II′) can be sketched by means of the usual diagrams representing partitions in a way similar to the Littlewood–Richardson sequences.

In order to do that, let us take $c^j = (r^j)^*$ and represent them by a diagram D^j formed by r_1^j $(= r_1)$ towers having heights $c_1^j, c_2^j, \dots$, or, equivalently, each floor being $r_1^j, r_2^j, \dots$ large. Then, D^j should be obtained by adding b_j blocks to D^{j-1} (condition (II)(a) or (II′)(a)), in such a way that the rules (II)(b) and (II)(c), or equivalently, (II′)(b) and (II′)(c), are respected.

Condition (II)(b) says that the $(i+1)$-flat can increase up to the length of the i-one in D^{j-1}. That is to say, each tower can increase one block maximum (condition (II′)(b)). As for the rule (II)(c), let us represent partition b by an analogous diagram and label the blocks on the jth floor b_j blocks. Recall that the rule (II)(b), or equivalently, (II′)(b), means that to obtain D^j, the blocks labeled "j" should be assigned to different towers of D^{j-1}. Then, condition (II)(c), or (II′)(c), means that the number of $(j+1)$-blocks installed at levels greater than $(i+1)$ are at most the number of j-blocks at levels greater than i (for all i).

For instance, if $b=(3,2)$ and $r=(4,3,2,1)$, then the sequences

$$\begin{array}{cccc} & & & 2 \\ 1 & 2 & & 1 \\ * & 1 & & * \\ * & * & 1 & * \\ * & * & * & 2 \\ * & * & * & * \end{array} \qquad \begin{array}{cccc} & & & 2 \\ & & & 1 \\ & 1 & & * \\ 1 & * & & * \\ * & * & 2 & * \\ * & * & * & * \end{array}$$

are allowed, whereas

$$\begin{array}{cccc} 2 & 2 & & \\ * & 1 & & \\ * & * & 1 & \\ * & * & * & 1 \\ * & * & * & * \end{array}$$

is not.

Proof of the equivalence (I) *and* (I′)*, and* (II) *and* (II′). To see that (I′) implies (I), notice that

$$\left(\begin{array}{cc|c} Q_1 & Q_{12} & S_1 \\ 0 & Q_2 & 0 \\ \hline 0 & 0 & T_1 \end{array}\right)\left(\begin{array}{cc} A_1 & Z \\ 0 & A_2 \\ \hline C_1 & C_3 \end{array}\right)\begin{pmatrix} Q_1 & Q_{12} \\ 0 & Q_2 \end{pmatrix}^{-1}$$

$$= \left(\begin{array}{cc} Q_1(A_1+Q_1^{-1}S_1C_1)Q_1^{-1} & \dots \\ 0 & Q_2A_2Q_2^{-1} \\ \hline T_1C_1Q_1^{-1} & T_1(-C_1Q_1^{-1}Q_{12}+C_3)Q_2^{-1} \end{array}\right).$$

Hence, P_1 can be reduced to a BK-matrix, A_2 to a Jordan matrix, and C_3 to 0 (since C_1 has the maximal rank, after eliminating, if necessary, its null rows and the corresponding ones in C_3). Furthermore, one can assume $\lambda=0$ because

$$P-\lambda\begin{pmatrix} I_n \\ 0 \end{pmatrix}, \qquad P_1-\lambda\begin{pmatrix} I_d \\ 0 \end{pmatrix}$$

are block-similar to P and P_1, respectively.

The equivalence (II) $\Leftrightarrow$ (II′) is a straightforward computation, taking $c^j=(r^j)^*$ (see Remark 3.5).

The next three sections are devoted to prove that conditions (I)–(I′) are equivalent to the (II)–(II′) ones. In fact, in section 4 we introduce a geometric version (I″) of (I)–(I′), and in sections 5 and 6 we prove that (II) is a necessary and sufficient condition for (I″), respectively.

Proof of the equivalence (II′) *and* (III). As it has been already remarked, (III) follows immediately from (II′)(a)–(II′)(b) (use Remark 3.5). Conversely, if (III) holds, the following strategy allows us to construct recurrently a sequence $c^0, c^1, \dots, c^s$ which verifies condition (II′): for each $j = 1, 2, \dots$, let c^j be a maximal (with regard to the partial ordering $\prec$) partition such that

(a) $|c^j| - |c^{j-1}| = b_j$,
(b) $\ell(c^j) = r_1$, $c^{j-1}_\nu \le c^j_\nu \le c^{j-1}_\nu + 1$,
(c′) $c^s - c^j \prec (b^j)^*$,

where $b^j = (b_{j+1}, b_{j+2}, \dots, b_s)$.

Notice that the set of partitions verifying (a)–(c′) is not empty because, in general, if $(\alpha_1, \alpha_2, \dots) \prec (\delta_1, \delta_2, \dots, \delta_\ell)$, then $(\alpha_1 - 1, \alpha_2 - 1, \dots, \alpha_\ell - 1, \alpha_{\ell+1}, \dots) \prec (\delta_1 - 1, \delta_2 - 1, \dots, \delta_\ell - 1)$, where the left member is assumed reordered to become nonincreasing. The proposed strategy takes c^j as a maximal element in this nonempty set.

By construction, conditions (II′)(a) and (II′)(b) are verified. Finally, let us see that if (II′)(c) does not hold, then c^{j-1} is not in fact maximal among the partitions verifying (a)–(c′) in the previous step. Broadly speaking, the following lemma shows that if the sequence $c^0, \dots, c^{j-1}, c^j, c^{j+1}, \dots$ verifies (a)–(c′) and

$$\begin{aligned} c^j_\mu &= c^{j-1}_\mu, & c^j_\eta &= c^{j-1}_\eta + 1, \\ c^{j+1}_\mu &= c^j_\mu + 1, & c^{j+1}_\eta &= c^j_\eta, \end{aligned}$$

then conditions (a)–(c′) are verified too if the partition c^j is replaced by $\bar c^j$, which differs from c^j only in

$$\bar c^j_\mu = c^{j-1}_\mu + 1, \qquad \bar c^j_\eta = c^{j-1}_\eta$$

(that is to say, we have permuted the increasing order of the "towers" μ, η (see Remark 3.5)). In particular, if $c^{j+1}_\mu > c^{j+1}_\eta$, then $c^j \prec \bar c^j$, so that c^j was not in fact maximal among the partitions verifying (a)–(c′). □

LEMMA 3.6. *Let α, δ be partitions such that*

$$\alpha \equiv (\alpha_1, \alpha_2, \dots) \prec \delta \equiv (\delta_1, \dots, \delta_\ell).$$

Assume that there is β such that

$$\begin{aligned} (\beta_1, \beta_2, \dots) &\prec (\delta_1 - 1, \dots, \delta_\ell - 1), \\ |\beta| &= |\alpha| - \ell, \\ \beta_\mu &= \alpha_\mu, \qquad \beta_\eta = \alpha_\eta - 1, \\ \alpha_\nu - 1 \le \beta_\nu &\le \alpha_\nu \quad \textit{for all } \nu. \end{aligned}$$

Then partition $\bar\alpha$ defined by

$$\begin{aligned} \bar\alpha_\nu &= \alpha_\nu \quad \textit{if } \nu \ne \mu, \eta, \\ \bar\alpha_\mu &= \alpha_\mu + 1, \qquad \bar\alpha_\eta = \alpha_\eta - 1 \end{aligned}$$

verifies $\bar\alpha \prec \delta$.

Proof. If $\alpha_\eta = \alpha_\mu + 1$, then $\bar\alpha = \alpha$ and there is nothing to prove.

If $\alpha_\eta > \alpha_\mu + 1$, then $\bar\alpha \prec \alpha \prec \delta$.

If $\alpha_\eta \le \alpha_\mu$, we can assume (reordering, if it is necessary) $\alpha_{\mu-1} > \alpha_\mu \ge \alpha_\eta > \alpha_{\eta+1}$ (or $\mu = 1$ and $\alpha_1 \ge \alpha_\eta > \alpha_{\eta+1}$). Then the order in α and $\bar\alpha$ is the same and we have

$$\sum_{\nu\le\mu_0} \bar\alpha_\nu = \sum_{\nu\le\mu_0} \alpha_\nu \quad \text{if } \mu_0 < \mu \text{ or } \mu_0 \ge \eta,$$

and

$$\sum_{\nu\le\mu_0} \bar\alpha_\nu = 1 + \sum_{\nu\le\mu_0} \alpha_\nu \le 1 + \min(\mu_0 - 1, \ell - 1) + \sum_{\nu\le\mu_0} \beta_\nu$$
$$\le 1 + \min(\mu_0 - 1, \ell - 1) + \sum_{\eta\le\min(\mu_0,\ell)} (\delta_\nu - 1) = \sum_{\nu\le\mu_0} \delta_\nu \quad \text{if } \mu \le \mu_0 < \eta. \qquad \square$$

4. Geometric formulation. Let us consider a geometric approach, analogous to the one in section 2 for square matrices. The study of pairs of matrices (see [5])

$$P = \begin{pmatrix} A \\ C \end{pmatrix} \in \mathbb{C}^{(n+m)\times n}$$

under the block-similarity is equivalent to the one of linear maps defined on a subspace $f : Y \longrightarrow X$, $Y \subset X$ ($\dim Y = n$, $\dim X = n + m$) under the following natural equivalence relation: $f \sim f'$ if and only if there is an automorphism φ of X, such that $\varphi(Y) = Y$ and $\varphi \circ f = f' \circ \widehat{\varphi}$ (where $\widehat{\varphi}$ means the restriction of φ to the subspace Y).

In fact, it is sufficient to consider P to be the matrix of f in any basis of X adapted to Y. In particular, the condition of C having the maximal rank is equivalent to $X = Y + f(Y)$. (This equality will hold throughout the paper.)

In these conditions, the observability indices of P (or f) can be computed as the conjugate partition of $\dim Y_0 - \dim Y_1, \dim Y_1 - \dim Y_2, \dots$, where

$$Y_i = f^{-i}(Y), \quad Y = Y_0 \supset Y_1 \supset \cdots \supset Y_k = Y_{k+1} \equiv Y_\infty.$$

In particular, P (or f) is observable if and only if $Y_\infty = \{0\}$. Notice that then f is injective.

A subspace $W \subset Y$ is f-invariant (or (C, A)-invariant) if and only if $f(W) \cap Y \subset W$ (see [1], [4]). Let us see that the special form of P in section 3 appears in a natural way when invariant subspaces are considered.

DEFINITION 4.1. *Let $f : Y \longrightarrow X$ be a linear map defined on a subspace $Y \subset X$, and $W \subset Y$ an f-invariant subspace. Then*

$$\widehat{f} : W \longrightarrow W + f(W),$$
$$\widetilde{f} : \frac{Y}{W} \longrightarrow \frac{X}{W + f(W)}$$

will be the maps induced in a natural way by f.

Remark 4.2. It is clear that $\widehat{f}$ is a linear map defined on a subspace. Moreover, if W is f-invariant, $\widetilde{f}$ can also be considered to be of this kind by means of the following identification:

$$\frac{Y}{W} = \frac{Y}{W + (f(W) \cap Y)} \cong \frac{Y + f(W)}{W + f(W)}$$
$$\subset \frac{Y + f(Y)}{W + f(W)} = \frac{X}{W + f(W)}.$$

PROPOSITION 4.3 (see [1], [4]). *Let $f : Y \longrightarrow X$ be as above, and $W \subset Y$ an f-invariant subspace. If $W_i = \widehat{f}^{-i}(W)$, then $W_i = Y_i \cap W$.*

In particular, if f is observable, then $\widehat{f}$ is observable too. Moreover, if their observabiblity indices are $(R_1, R_2, \dots)^*$ and $(r_1, r_2, \dots)^*$, respectively, then $r_i \leq R_i$ for all $i = 1, 2, \dots$.

Let us characterize geometrically the special form of the matrices involved in this problem.

PROPOSITION 4.4. *Let $f : Y \longrightarrow X$ as above, and a subspace $W \subset Y$.*

(1) *W is f-invariant if and only if the matrix of f in any basis adapted to $W \subset Y \subset Y + f(W) \subset X$ has the form*

$$\begin{pmatrix} A \\ C \end{pmatrix} = \left(\begin{array}{cc} A_1 & A_3 \\ 0 & A_2 \\ \hline C_1 & C_3 \\ 0 & C_2 \end{array}\right),$$

where $\binom{A_1}{C_1}$ is the matrix of $\widehat{f}$ in the same basis.

(2) *In the conditions of* (1)*, the pair $\binom{A_2}{C_2}$ is the matrix of $\widetilde{f}$ in the basis induced in a natural way by the one considered in X.*

(3) *In the above conditions, if f is observable, then $\widetilde{f}$ is an endomorphism if and only if there is a basis of X adapted to $W \subset Y \subset X$ such that the matrix of f has the form*

$$\left(\begin{array}{cc} A_1 & A_3 \\ 0 & A_2 \\ \hline C_1 & 0 \end{array}\right).$$

Proof.

(1) It is a direct consequence of the inclusion $f(W) \cap Y \subset W$ which characterizes the f-invariant subspaces.

(2) It is straightforward.

(3) Because of Remark 4.2, $\widetilde{f}$ is an endomorphism if and only if $f(Y) \subset Y + f(W)$. Obviously, this relation is verified for the matrices of the form considered. Conversely, taking anti-images in this inclusion, $Y = Y_1 + W$. Hence, there is a subspace V such that

$$Y = V \oplus W, \quad Y_1 = V \oplus W_1.$$

The latter implies $f(V) \subset Y$, so that in any basis adapted to $W, V \subset Y \subset X$ the matrix of f has the desired form. □

Therefore, conditions (I)–(I′) in Theorem 3.1 can be translated in the following geometric way, which will be used in the proof of the main theorem.

COROLLARY 4.5. *Within the context of Theorem* 3.1*, conditions* (I)–(I′) *are equivalent to the following condition:*

(I″) *There is a linear map defined on a subspace $f : Y \longrightarrow X$, $Y \subset X$, and a f-invariant subspace $W \subset Y$ such that*

(1) *f is observable, having observability indices R^*;*

(2) *$\widehat{f}$ is observable, having observability indices r^*;*

(3) *$\widetilde{f}$ is a nilpotent endomorphism, having Segre characteristic b^*.*

Remark 4.6. From the proof of (3) in Proposition 4.4, it follows that if f is observable, then $\widetilde{f}$ is an endomorphism if and only if

$$\dim Y - \dim W = \dim Y_1 - \dim W_1.$$

Then a necessary condition for (I″) above is $R_1 = r_1$.

Remark 4.7. The assumption that $\widetilde{f}$ is an endomorphism is not a significant restriction. In general, if one considers the decreasing stationary chain of subspaces

$$\begin{aligned} &Y = Y^0 \supset Y^1 \supset \cdots \supset Y^h = Y^{h+1} \equiv Y^\infty \ (\supset W), \\ &Y^j = f^{-1}(Y^{j-1}) + W, \end{aligned}$$

then for the restriction $f^\infty : Y^\infty \longrightarrow X$ one has that $\widetilde{f^\infty}$ is an endomorphism, whereas the map induced by f in Y/Y^∞ is observable.

5. Proof of the necessity. Now let $f : Y \longrightarrow X$ and $W \subset X$ be as in (I″) of Corollary 4.5. Following the pattern in section 2 in order to prove that condition (II) in Theorem 3.1 is verified, a double family of subspaces will be introduced: $W_i^j = Y_i \cap W^j$, where W^j is defined in such a way that $\operatorname{Ker} \widetilde{f}^j = W^j/W$, as will be seen below.

Notice that, as it has been recalled in section 4 (see [5]),

$$\begin{aligned} R_i &= \dim Y_{i-1} - \dim Y_i, \\ r_i &= \dim W_{i-1} - \dim W_i, \end{aligned}$$

where $Y_i = f^{-i}(Y)$, $W_i = \widehat{f}^{-i}(W) = f^{-i}(W) \cap W = Y_i \cap W$.

Definition 5.1. *Let $f : Y \longrightarrow X$ and $W \subset Y$ be as in* (I″) *of Corollary* 4.5. *Then*

$$\begin{aligned} &W^0 = W, \\ &W^j = f^{-1}(W^{j-1} + f(W)) = f^{-1}(W^{j-1}) + W, \quad j \geq 1. \end{aligned}$$

Proposition 5.2. *With the notation in the above definition,*

(1) $\operatorname{Ker} \widetilde{f}^j = \frac{W^j}{W}$ *for all* $j \geq 1$*;*
(2) $W = W^0 \subset W^1 \subset \cdots \subset W^{\ell(b)} = W^{\ell(b)+1} = \cdots = Y$, $b_j = \dim W^j - \dim W^{j-1}$ *for all* $j \geq 1$*;*
(3) *the subspaces W^j are f-invariant. In fact, they verify $f(W^j) \cap Y \subset W^{j-1}$ for all $j \geq 1$.*

Proof. (1) We proceed by induction, using the identification in Remark 4.2. It is obvious for $j = 0$. Assume that

$$\operatorname{Ker} \widetilde{f}^j = \frac{W^j}{W} \cong \frac{W^j + f(W)}{W + f(W)} \subset \frac{X}{W + f(W)}.$$

Then

$$\begin{aligned} \operatorname{Ker} \widetilde{f}^{j+1} &= \widetilde{f}^{-1}(\operatorname{Ker} \widetilde{f}^j) \\ &= \frac{f^{-1}(W^j + f(W)) + W}{W} = \frac{W^{j+1}}{W}. \end{aligned}$$

(2) It follows immediately from (1).

(3) The proof also follows by induction. For $j = 1$, we have

$$\begin{aligned} f(W^1) \cap Y &= f(f^{-1}(W) + W) \cap Y \\ &\subset (W + f(W)) \cap Y \subset W. \end{aligned}$$

If the property is verified by W^j, then

$$\begin{aligned} f(W^{j+1}) \cap Y &= f(f^{-1}(W^j) + W) \cap Y \\ &\subset (W^j + f(W)) \cap Y \\ &\subset (W^j + f(W^j)) \cap Y \subset W^j + W^{j-1} = W^j. \qquad \square \end{aligned}$$

For each f-invariant subspace W^j, we consider the natural finite chain (W_i^j).

DEFINITION 5.3. *In the conditions of Definition* 5.1*, we define for all* $j \geq 0$

$$\begin{aligned} W_i^j &= f^{-i}(W^j) \cap W^j = Y_i \cap W^j, \quad i \geq 0, \\ r_i^j &= \dim W_{i-1}^j - \dim W_i^j, \quad i \geq 1. \end{aligned}$$

Remark 5.4.

(1) It will be useful to bear in mind the following finite diagram:

$$\begin{array}{ccccccccccccc}
\cdots \subset f(W_1) \subset & W & \subset & W^1 & \subset \cdots \subset & W^{j-1} & \subset & W^j & \subset \cdots \subset & W^{\ell(b)} & = & Y \\
& \cup & & \cup & & \cup & & \cup & & \cup & & \cup \\
\cdots \subset f(W_2) \subset & W_1 & \subset & W_1^1 & \subset \cdots \subset & W_1^{j-1} & \subset & W_1^j & \subset \cdots \subset & W_1^{\ell(b)} & = & Y_1 \\
& \cup & & \cup & & \cup & & \cup & & \cup & & \cup \\
& \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
& & & \cup & & \cup & & \cup & & \cup & & \cup \\
\cdots & \subset W_{i-1} & \subset & W_{i-1}^1 & \subset \cdots \subset & W_{i-1}^{j-1} & \subset & W_{i-1}^j & \subset \cdots \subset & W_{i-1}^{\ell(b)} & = & Y_{i-1} \\
& & & \cup & & \cup & & \cup & & \cup & & \cup \\
\cdots & \subset W_i & \subset & W_i^1 & \subset \cdots \subset & W_i^{j-1} & \subset & W_i^j & \subset \cdots \subset & W_i^{\ell(b)} & = & Y_i \\
& & & \cup & & \cup & & \cup & & \cup & & \cup \\
& & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots
\end{array}.$$

(2) Notice that

$$\frac{Y}{W} \cong \bigoplus_{\substack{i \geq 0 \\ 1 \leq j \leq \ell(b)}} \frac{W_i^j}{W_i^{j-1} + W_{i+1}^j},$$

$$\dim \frac{W_i^j}{W_i^{j-1} + W_{i+1}^j} = \dim W_i^j - \dim W_i^{j-1} - \dim W_{i+1}^j + \dim W_{i+1}^j = r_{i+1}^j - r_{i+1}^{j-1}.$$

These facts will guide the construction in the next section.

From the definitions and (3) of Proposition 5.2, it follows that $f(W_i^j) \subset W_{i-1}^{j-1}$ for all $i, j \geq 1$. Some basic properties of these maps are summarized in the following proposition.

PROPOSITION 5.5. *In the conditions of the above definition,*

(1) $f^{-1}(W_{i-1}^{j-1}) = W_i^j$;

(2) *the induced maps*

(i) $\dfrac{W_i^j}{W_{i+1}^j} \longrightarrow \dfrac{W_{i-1}^{j-1}}{W_i^{j-1}}$,

(ii) $\dfrac{W_i^{j+1}}{W_i^j} \longrightarrow \dfrac{W_{i-1}^j}{W_{i-1}^{j-1}}$

are injective for all $i, j \geq 1$.

Proof.

(1) $f^{-1}(W_{i-1}^{j-1}) = f^{-1}(Y_{i-1} \cap W^{j-1}) = Y_i \cap f^{-1}(W^{j-1}) = Y_i \cap [f^{-1}(W^{j-1}) + W] = Y_i \cap W^j = W_i^j$.

(2) It is a direct consequence of (1). □

Finally, let us use the above construction to prove that condition (I″) of Corollary 4.5 implies (II) of Theorem 3.1. Let there be $(r^j)^*$, $0 \leq j \leq \ell(b)$, the observability indices of the restriction of f to each subspace W^j, that is to say

$$r^j = (r_1^j, r_2^j, \dots)$$

(see Definition 5.3). Notice that this restriction is observable (see Proposition 4.3 and (3) of Proposition 5.2); hence $|r^j| = \dim W^j$.

Let us see that, in fact, these partitions verify the properties in (II) of Theorem 3.1:

(a) Obviously $r^0 = r$, $r^{\ell(b)} = R$. Moreover,

$$\begin{aligned} |r^j| - |r^{j-1}| &= \dim W^j - \dim W^{j-1} \\ &= \dim \operatorname{Ker} \widetilde{f}^j - \dim \operatorname{Ker} \widetilde{f}^{j-1} = b_j \end{aligned}$$

for all $1 \leq j \leq \ell(b)$.

(b) From Proposition 4.3, $r_i^{j-1} \leq r_i^j$ for all $i, j \geq 1$. The equality holds for $i = 1$, according to Remark 4.6.
The inequality $r_i^j \leq r_{i-1}^{j-1}$ follows from the injectivity of (2)(i) in Proposition 5.5.

(c) Finally, condition (c) is a consequence of the injectivity of (2)(ii) in Proposition 5.5.

6. Proof of the sufficiency. Let partitions R, r, b and $r^1, r^2, \dots, r^s$ be given, which verify (a)–(c) in (II) of Theorem 3.1. $f : Y \longrightarrow X$, $W \subset Y$ will be constructed in such a way that conditions (I″)(1), (I″)(2), and (I″)(3) in Corollary 4.5 are verified.

Let $W \subset Y \subset X$ be vector spaces having dimension $|r|$, $|R|$, and $|R| + R_1$, respectively. Let $\widehat{f} : W \longrightarrow X$ be an observable linear map having observability indices r^*, so that condition (I″)(2) in Corollary 4.5 is verified. Also, a BK-basis of W is formed by $r_1 - r_2$ BK-chains having length 1, $r_2 - r_3$ BK-chains having length 2, etc. Let

$$\begin{aligned} B^0 &= \bigcup_{1 \leq i \leq \ell(r)} B_i^0; \\ B_i^0 &= \{e_{i,k}^0;\ 1 \leq k \leq r_i - r_{i+1}\} \end{aligned}$$

be a set of generators of these BK-chains. Hence

$$W_i = [B_{i+1}^0;\ B_{i+2}^0, f(B_{i+2}^0);\ B_{i+3}^0, f(B_{i+3}^0), f^2(B_{i+3}^0); \dots]$$

for all $1 \leq i \leq \ell(r)$.

Now, $\widehat{f}$ must be extended to $f : Y \longrightarrow X$ verifying (I″)(1) and (I″)(3) in Corollary 4.5. In order to achieve that, we consider any supplementary subspace $\overline{W}$ of W in Y,

and any basis B of it. Taking into account Remark 5.4, $\overline{W}$ should be split into direct summands $\overline{V}_i^j$ having a dimension

$$d_{i+1}^j = r_{i+1}^j - r_{i+1}^{j-1} \quad (i \geq 0,\, 1 \leq j \leq \ell(b)),$$

respectively, and then f will be defined on each of these subspaces $\overline{V}_i^j$. First, B is distributed (in any way) into subsets having cardinal d_{i+1}^j:

$$B = \bigcup_{\substack{i\geq 0\\ 1\leq j\leq \ell(b)}} B_{i+1}^j,$$
$$B_{i+1}^j = \{e_{i+1,k}^j;\ 1 \leq k \leq d_{i+1}^j\}.$$

(Notice that $B_{i+1}^j = \emptyset$ if $d_{i+1}^j = 0$; in particular, $B_1^j = \emptyset$, and $B_{i+1}^j = \emptyset$ if $i \geq \ell(r^j)$.) Second, $\overline{V}_i^j = [B_{i+1}^j]$ $\quad (i \geq 0,\, 1 \leq j \leq \ell(b))$, so that

$$Y = W \oplus \overline{W} = W \oplus \left(\bigoplus_{\substack{i\geq 0\\ 1\leq j\leq \ell(b)}} \overline{V}_i^j \right),$$
$$\dim \overline{V}_i^j = d_{i+1}^j = r_{i+1}^j - r_{i+1}^{j-1}.$$

(Notice that $\overline{V}_0^j = \{0\}$, and $\overline{V}_i^j = \{0\}$ if $i \geq \ell(r^j)$.)

Considering the diagram

$$\begin{array}{ccccc} & & \cdots & & \cdots \\ & & \oplus & & \oplus \\ \cdots\oplus & \overline{V}_i^{j-1} & \oplus & \overline{V}_i^j & \oplus\cdots \\ & \oplus & & \oplus & \\ \cdots\oplus & \overline{V}_{i+1}^{j-1} & \oplus & \overline{V}_{i+1}^j & \oplus\cdots \\ & \oplus & & \oplus & \\ & \cdots & & \cdots & \end{array}$$

and defining, for $i \geq 0$, $0 \leq j \leq \ell(b)$,

$$V_i^j = W_i \oplus \left(\bigoplus_{\substack{\ell\geq i\\ 1\leq h\leq j}} \overline{V}_\ell^h \right),$$

the following diagram is obtained:

$$\begin{array}{ccccccccc} \cdots\subset & W & \subset & V^1 & \subset\cdots\subset & V^{\ell(b)} & = & Y \\ & \cup & & \cup & & \cup & & \cup \\ \cdots\subset & W_1 & \subset & V_1^1 & \subset\cdots\subset & V_1^{\ell(b)} & \equiv & V_1 \\ & \cup & & \cup & & \cup & & \cup \\ & \cdots & & \cdots & & \cdots & & \cdots \end{array}$$

(where $V^j \equiv V_0^j$ and $V_i \equiv V_i^{\ell(b)}$), analogous to the one in Remark 5.4. Now, f will be defined on each $\overline{V}_i^j$ in such a way that the corresponding subspaces W_i^j (according to Definition 5.3) are just V_i^j. Then, as desired, the observability indices of f will be R^* and the Segre characteristic of $\widetilde{f}$ will be b^*, $b_j = |r^j| - |r^{j-1}|$, so that the proof of the sufficiency will be finished.

To define f, in fact, two extensions, f_*, $f^* : Y \longrightarrow X$ of $\widehat{f}$, will be defined and then $f = \frac{1}{2}(f_* + f^*)$.

(1) For each $i \geq 1$, f_* on $\overline{V}_i^j$ will be defined by increasing recurrence over $1 \leq j \leq \ell(b)$.
For $j = 1$,

$$f_*(e^1_{i+1,k}) = e^0_{i,k} \in B_i^0 \subset W_{i-1}.$$

It is possible because the hypothesis (II)(b) implies

$$\dim \overline{V}_i^1 = r_{i+1}^1 - r_{i+1} \leq r_i - r_{i+1} = \#B_i^0.$$

For $j \geq 2$,

$$f_*(e^j_{i+1,k}) = e^{j-1}_{i,k} \in B_i^{j-1} \subset \overline{V}_{i-1}^{j-1}$$

if $1 \leq k \leq \min\{b_{i+1}^j, b_i^{j-1}\}$, and taking images

$$\begin{aligned} f_*(e^j_{i+1,k}) &\in B_i^{j-2} \cup B_i^{j-3} \cup \cdots \cup B_i^0 \\ &\subset \overline{V}_{i-1}^{j-2} \oplus \cdots \oplus \overline{V}_{i-1}^1 \oplus W_{i-1} \end{aligned}$$

in such a way that f_* is injective if $d_i^{j-1} < k \leq d_{i+1}^j$.
It is possible because, as above,

$$\begin{aligned} \dim(\overline{V}_i^1 \oplus \cdots \oplus \overline{V}_i^j) &= (r_{i+1}^1 - r_{i+1}) + (r_{i+1}^2 - r_{i+1}^1) + \cdots + (r_{i+1}^j - r_{i+1}^{j-1}) \\ &= -r_{i+1} + r_{i+1}^j \leq -r_{i+1} + r_i^{j-1} \\ &= (r_i - r_{i+1}) + (r_i^1 - r_i) + \cdots + (r_i^{j-1} - r_i^{j-2}) \\ &= \#B_i^0 + \#B_i^1 + \cdots + \#B_i^{j-2} + \#B_i^{j-1}. \end{aligned}$$

(2) Now

$$f^*(e^1_{i+1,k}) = f_*(e^1_{i+1,k}) = e^0_{i,k}.$$

For each $j \geq 2$, f^* is defined on $\overline{V}_i^j$ by decreasing recurrence over $1 \leq i < \ell(r^j)$.
For $1 \leq k \leq \min\{d_{i+1}^j, d_i^{j-1}\}$,

$$f^*(e^j_{i+1}, k) = f_*(e^j_{i+1,k}) = e^{j-1}_{i,k}$$

and for $d_i^{j-1} < k \leq d_{i+1}$, taking images

$$\begin{aligned} f^*(e^j_{i+1,k}) &\in B_{i+1}^{j-1} \cup B_{i+2}^{j-1} \cup \cdots \cup B_{\ell(r^{j-1})}^{j-1} \\ &\subset \overline{V}_i^{j-1} \oplus \overline{V}_{i+1}^{j-1} \oplus \cdots \oplus \overline{V}_{\ell(r^{j-1})-1}^{j-1} \end{aligned}$$

in such a way that f^* is injective.
This is possible because of hypothesis (II)(c):

$$\dim\left(\bigoplus_{\ell \geq i} \overline{V}_\ell^j\right) = \sum_{\ell > i}(r_\ell^j - r_\ell^{j-1}) \leq \sum_{\ell > i-1}(r_\ell^{j-1} - r_\ell^{j-2}) = \dim\left(\bigoplus_{\ell \geq i-1} \overline{V}_\ell^{j-1}\right).$$

(3) Finally, $f = \frac{1}{2}(f_* + f^*)$. Obviously, it is an extension of $\widehat{f}$.

The proof of Theorem 3.1 will be finished if $V_i^j = W_i^j$, or, equivalently, $Y_i = V_i$, $W^j = V^j$.

Obviously, $V_0 = Y$. Hence, it is sufficient to prove the following lemma.

LEMMA 6.1. *With the above notation,*

(1) $f^{-1}(V^{j-1}) + W = V^j$ *for all* $j \geq 1$*;*

(2) $f^{-1}(V_{i-1}) = V_i$ *for all* $i \geq 1$.

Proof. Previously notice that if a vector $e_{i,k}^{j-1} \in B$ belongs to $f_*(B)$ and also to $f^*(B)$, then either

$$e_{i,k}^{j-1} = f_*(e_{i+1,k}^j) = f^*(e_{i+1,k}^j) = f(e_{i+1,k}^j)$$

or there are some unique $h > 0$ and $\ell \geq 0$ such that

$$e_{i,k}^{j-1} \in f_*(B_{i+1}^{j+h}) \cap f^*(B_{i-\ell}^j).$$

(1) By construction

$$f(V^j) \subset V^{j-1} + f(W).$$

Hence

$$V^j \subset f^{-1}(V^{j-1}) + W.$$

For the opposite inclusion, assume $x \notin V^j$ and let J be the maximum index $J > j$ such that x has some nonzero component in $\overline{V}^J \equiv \oplus_i \overline{V}_i^J$. Then $f^*(x)$ should have some nonzero component in $\overline{V}^{J-1} \equiv \oplus_i \overline{V}_i^{J-1}$. According to the previous note, and bearing in mind the definition of J, this component cannot be canceled by any component of $f_*(x)$, so that $f(x)$ has in fact some nonzero component in $\overline{V}^{J-1}$. Therefore, $f(x) \notin V^{j-1} + f(W)$.

(2) By construction, $f(V_i) \subset V_{i-1}$. Hence, $V_i \subset f^{-1}(V_{i-1})$. For the opposite inclusion, we proceed by increasing recurrence over i, in an analogous way to (1). ∎

7. Construction of solutions. When condition (III) of Theorem 3.1 holds, explicit solutions Z verifying (I)–(I′) can be obtained by means of the construction in section 6, starting on any sequence of partitions verifying (II). Two of such solutions Z, Z' will be called *equivalent* if the associated matrices can be transformed one into the other by means of a change of basis preserving their block structure, that is to say, if

$$\left(\begin{array}{cc|c} Q_1 & Q_{12} & S_1 \\ 0 & Q_2 & 0 \\ \hline 0 & 0 & T_1 \end{array}\right)\left(\begin{array}{cc} A_1 & Z \\ 0 & A_2 \\ \hline C_1 & C_3 \end{array}\right)\begin{pmatrix} Q_1 & Q_{12} \\ 0 & Q_2 \end{pmatrix}^{-1} = \left(\begin{array}{cc} A_1 & Z' \\ 0 & A_2 \\ \hline C_1 & C_3 \end{array}\right),$$

where Q_1, Q_2, and T_1 are nonsingular. Clearly, different sequences of partitions as in (II) lead to nonequivalent solutions. Example 7.3 shows that nonequivalent solutions are possible even for the same sequence of partitions.

Example 7.1. Clearly, the partitions

$$R = (2, 2, 1), \qquad r = (2), \qquad b = (1, 1, 1)$$

verify condition (III) of Theorem 3.1. Two sequences of partitions verifying (II) are possible:

$$(2,1), \quad (2,2), \quad (2,2,1);$$
$$(2,1), \quad (2,1,1), \quad (2,2,1).$$

According to the construction in section 6, they lead, respectively, to the following nonequivalent solutions:

$$\left(\begin{array}{cc|ccc} 0&0&0&0&1\\ 0&0&0&1&0\\ \hline 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ \hline 1&0&0&0&0\\ 0&1&0&0&0 \end{array}\right), \qquad \left(\begin{array}{cc|ccc} 0&0&0&0&1\\ 0&0&1&0&0\\ \hline 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ \hline 1&0&0&0&0\\ 0&1&0&0&0 \end{array}\right).$$

Example 7.2. In general, condition (III) of Theorem 3.1 holds if $b = (1, 1, \dots, 1)$. It is not difficult to see that a sequence of partitions r^j, $0 \le j \le \ell(b)$, verifying (II) can be constructed by recurrence as follows:

$$r^j_{i(j)} = \quad r^{j-1}_{i(j)} + 1,$$
$$r^j_i = r^{j-1}_i \quad \text{if } i \neq i(j),$$

where

$$i(j) = \max\{i : \ r^{j-1}_i < R_i, \quad r^{j-1}_i < r^{j-1}_{i-1}\}.$$

Then, as in the previous example, explicit solutions can be obtained by means of the construction in section 6.

Example 7.3. Let us consider the partitions

$$R = (2,2,1,1), \quad r = (2), \quad b = (1,1,1,1).$$

It is a straightforward computation to see that the solutions

$$\left(\begin{array}{cc|cccc} 0&0&0&\lambda&0&1\\ 0&0&0&0&1&0\\ \hline 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ \hline 1&0&0&0&0&0\\ 0&1&0&0&0&0 \end{array}\right)$$

are nonequivalent for different values of $\lambda \in \mathbb{C}$, although all of them correspond to the sequence of partitions

$$(2,1), \quad (2,2), \quad (2,2,1), \quad (2,2,1,1).$$

In a similar way to the "condensation lemma" for the classical Carlson problem, let us see that many zero entries can be prescribed in the Z solutions.

LEMMA 7.4. *Let R, r, b be three partitions verifying the conditions in Theorem* 3.1, *and consider the particular case in* (I′). *Then*

(1) *for any Z solution, there is an equivalent Z' solution having nonzero entries only in the r_1 rows corresponding to the null ones in A_1;*
(2) *moreover, Z' can be chosen in such a way that its entries in the b_1 columns corresponding to the null ones in A_2 are also 0, except one of them in each column, which can be valued 1 and are placed in different rows.*

Proof. (1) It is immediate that each vector in the basis of Y, not in W, can be changed by adding a vector in W in such a way that its image by f would be a linear combination of the generators of the BK-chains of W. Explicitly, assume $P_1 = \binom{N}{E}$ is a BK-matrix and $A_2 = J$ is a nilpotent Jordan matrix. Notice that

$$\left(\begin{array}{cc|c} I & Q_{12} & 0 \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right)\left(\begin{array}{cc} N & Z \\ 0 & J \\ \hline E & 0 \end{array}\right)\begin{pmatrix} I & Q_{12} \\ 0 & I \end{pmatrix}^{-1} = \left(\begin{array}{cc} 0 & Z - NQ_{12} + Q_{12}J \\ 0 & J \\ \hline E & EQ_{12} \end{array}\right).$$

Let us choose Q_{12} in such a way that $EQ_{12} = 0$ and $Z' = Z - NQ_{12} + Q_{12}J$ verifies the desired property. For the first, it is sufficient to make null the rows in Q_{12} corresponding to the lowest one in each block of N. The remaining rows of Q_{12} can be computed easily by recurrence in order to cancel all the rows in Z except those corresponding to null ones in N. For example, let

$$N = \left(\begin{array}{cc|ccc} 0 & 0 & & & \\ 1 & 0 & & & \\ \hline & & 0 & 0 & 0 \\ & & 1 & 0 & 0 \\ & & 0 & 1 & 0 \end{array}\right), \quad J = \left(\begin{array}{c|ccc} 0 & & & \\ \hline & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & 0 & 1 & 0 \end{array}\right), \quad Q_{12} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ 0 & 0 & 0 & 0 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then $EQ_{12} = 0$,

$$-NQ_{12} + Q_{12}J = \begin{pmatrix} 0 & a_3 & a_4 & 0 \\ -a_1 & -a_2 & -a_3 & -a_4 \\ 0 & c_3 & c_4 & 0 \\ -c_1 & -c_1 + d_3 & -c_3 + d_4 & -c_4 \\ -d_1 & -d_2 & -d_3 & -d_4 \end{pmatrix}.$$

It is clear that Q_{12} can be chosen in such a way that $Z - NQ_{12} + Q_{12}J$ has zero entries in the second, fourth, and fifth rows.

(2) From Proposition 5.5, it follows immediately that, for all $i \geq 1$, the maps induced by f

$$\frac{W_i^1}{W_{i+1}^1 + W_i} \longrightarrow \frac{W_{i-1}}{W_i + f(W_i)}$$

are injective. Notice that the vectors in W^1/W are the eigenvectors of $\widetilde{f}$. Thus, because of the above injectivities, the images of a basis of $\widetilde{f}$-eigenvectors (in fact, of

a set of representative vectors in W^1) can be extended to a family of BK-generators of $\widehat{f}$. □

Solutions having a minimal number of nonzero entries arise when the subspace W is "marked" [8], [4], that is to say, when there is some BK-basis of $\widehat{f}$ extendible to a BK-basis of f.

COROLLARY 7.5. *Let R, r, b be three partitions verifying the conditions in Theorem* 3.1 *and Corollary* 4.5. *Then the following assertions are equivalent:*

(1) *In terms of condition* (I′)*, there is some solution Z whose only nonzero entries are those referred to in part* (2) *of Lemma* 7.4*, that is to say, b_1 1-valued entries placed in (different) columns corresponding to the null ones in J, and in (different) rows corresponding to the null ones in N.*

(2) *There is an f-marked subspace W verifying* (I″) *of Corollary* 4.5.

(3) *With the notation in* (III)*: $R^* - r^* = b^*$.*

REFERENCES

[1] I. BARAGAÑA AND I. ZABALLA, *Block similarity invariants of restrictions to (A, B)-invariants subspaces*, Linear Algebra Appl., 220 (1995), pp. 31–62.

[2] I. BARAGAÑA AND I. ZABALLA, *Feedback invariants of supplementary pairs of matrices*, Automatica J. IFAC, 33 (1997), pp. 2119–2130.

[3] I. BARAGAÑA AND I. ZABALLA, *Feedback Invariants of Restrictions and Quotients: Series and Parallel Connected Systems*, Preprint, 1998.

[4] A. COMPTA AND J. FERRER, *On $(A, B)^t$-invariant subspaces having extendible Brunovsky bases*, Linear Algebra Appl., 255 (1997), pp. 185–201.

[5] J. FERRER AND F. PUERTA, *Similarity of non-everywhere defined linear maps*, Linear Algebra Appl., 168 (1992), pp. 27–55.

[6] I. GOHBERG, AND M.A. KAASHOEK, *Unsolved problems in matrix and operator theory* II. *Partial multiplicities for products*, Integral Equations Operator Theory, 2 (1979), pp. 116–120.

[7] I. GOHBERG, M.A. KAASHOEK, AND F. VAN SCHAGEN, *Partially Specified Matrices and Operators: Classification, Completion, Applications*, Oper. Theory Adv. Appl. 79, Birkhäuser-Verlag, Basel, 1995.

[8] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Invariant Subspaces of Matrices with Applications*, John Wiley, New York, 1986.

[9] T. KLEIN, *The multiplication of Schur functions and extensions of p-modules*, J. London Math. Soc., 43 (1968), pp. 280–284.

[10] R.C. THOMPSON, *Smith invariants of a product of integral matrices*, Contemp. Math., 47 (1985), pp. 401–435.

[11] W.M. WONHAM, *Linear Multivariable Control: A Geometric Approach*, Springer, New York, 1979.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 276–281

THE USE AND PROPERTIES OF TIKHONOV FILTER MATRICES*

MÅRTEN GULLIKSSON† AND PER-ÅKE WEDIN†

Abstract. We consider the concept of Tikhonov filter matrices in connection with discrete ill-posed and rank-deficient linear problems. Important properties of the Tikhonov filter matrices are given together with their filtering and regularization effects.

We also present new perturbation identities for the Tikhonov regularized linear least squares problem using filter matrices generalizing well-known perturbation identities for the linear least squares problem and pseudoinverses.

Key words. Tikhonov regularization, rank-deficient, pseudoinverse, filter factors

AMS subject classification. 65K

PII. S0895479899355025

1. Introduction.

One way of getting a useful solution to a discrete ill-posed or rank-deficient linear problem is by regularization. The main idea of regularization is to find a new problem or method that damps the effect of noise in input data. This can be realized by using the filter matrices to be considered in detail in this paper.

To be more specific we consider the linear approximation problem

$$Jx \approx b, \tag{1.1}$$

where $J \in \Re^{m\times n}$ has rank $r \leq n$ and $b \in \Re^m$. We will assume that $m \geq n$ for simplicity. In fact, this is always the case when the problem is attained from an ill-posed infinite dimensional problem.

The regularization problem we will use is the Tikhonov problem

$$\min_x \ \|Jx - b\|^2 + \mu^2\|x\|^2, \tag{1.2}$$

where $\|\cdot\|$ is the 2-norm. The solution to the Tikhonov problem (1.2) may be written as

$$x_{\text{reg}} = J^{\#}b, \tag{1.3}$$

where

$$J^{\#} = (J^TJ + \mu^2 I)^{-1}J^T \tag{1.4}$$

is the Tikhonov regularized inverse and $\mu > 0$ is a regularization parameter.

We make the following definition of filter matrices also suggested by Hansen [3].

DEFINITION 1.1. *The Tikhonov filter matrices are defined as*

$$P_{\mathcal{R}} = JJ^{\#}, \quad P_{\mathcal{N}'} = I_m - JJ^{\#},$$
$$P_{\mathcal{R}'} = J^{\#}J, \quad P_{\mathcal{N}} = I_n - J^{\#}J.$$

*Received by the editors April 26, 1999; accepted for publication (in revised form) by G. Golub January 31, 2000; published electronically June 20, 2000.
http://www.siam.org/journals/simax/22-1/35502.html

†Department of Computer Science, Umeå University, S-901 87, Umeå, Sweden (marten@cs.umu.se, per-ake.wedin@cs.umu.se).

The filter matrices $P_{\mathcal{R}}$ and $P_{\mathcal{R}'}$ are sometimes called the influence matrix and the resolution matrix [3], respectively.

In the next section we will state some important properties of the filter matrices to be used in the following sections.

The filter matrices are closely related to the filtering of noise in b, and these aspects will be discussed briefly in section 3. A similar exposition can be found in [3].

In section 4 we derive perturbation identities for the Tikhonov inverse and the Tikhonov problem using filter matrices.

2. Properties of the Tikhonov inverse and the filter matrices. In this section we state some of the properties of $J^{\#}$ and the filter matrices connected to the pseudoinverse of J and projections. The main tool for expressing these properties is the SVD of J, i.e.,

$$J = U\Sigma V^T, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n).$$

The singular values, σ_i, are ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. We assume that J has rank r, giving $\sigma_r \neq 0$ and $\sigma_i = 0, i = r+1, \ldots, n$. The matrices $U \in \Re^{m\times m}$ and $V \in \Re^{n\times n}$ are orthogonal. For more details on the SVD we refer to [1].

If we partition U, V as

$$U = (\underset{r}{U_1}, \ \underset{m-r}{U_2}), \quad V = (\underset{r}{V_1}, \ \underset{n-r}{V_2})$$

and define

$$\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r), \tag{2.1}$$

then the SVD can be written as

$$J = U_1\Sigma_1 V_1^T.$$

We note that U_1 spans the range space of J, $\mathcal{R}(J)$, U_2 spans the null space of J^T, $\mathcal{N}(J^T)$, V_1 spans the range space of J^T, $\mathcal{R}(J^T)$, and V_2 spans the null space of J, $\mathcal{N}(J)$.

By defining

$$\Sigma_1^{\#} = \operatorname{diag}\left(\frac{\sigma_1}{\sigma_1^2+\mu^2}, \ldots, \frac{\sigma_r}{\sigma_r^2+\mu^2}\right) \tag{2.2}$$

we can write the Tikhonov inverse on the form

$$J^{\#} = V_1\Sigma_1^{\#}U_1^T. \tag{2.3}$$

Further, the pseudoinverse of J can be written as

$$J^{\dagger} = V_1\Sigma_1^{-1}U_1^T \tag{2.4}$$

and some algebra reveals that

$$J^{\#} = J^{\dagger} - \mu^2 V_1\Sigma_1^{\#}\Sigma_1^{-2}U_1^T,$$

giving the well-known fact

$$J^{\#} \to J^{\dagger}$$

as μ tends to zero.

Inserting the form on $J^{\#}$ in (2.3) into the definition of the filter matrices we get

$$P_{\mathcal{R}} = U_1\Sigma_1^{\#}\Sigma_1 U_1^T, \quad P_{\mathcal{R}'} = V_1\Sigma_1^{\#}\Sigma_1 V_1^T. \tag{2.5}$$

Further, if we use the fact that $I = UU^T = U_1U_1^T + U_2U_2^T$ we have that

$$P_{\mathcal{R}} = \mathcal{P}_{\mathcal{R}(J)} - \mu^2 U_1\Sigma_1^{\#}\Sigma_1^{-1}U_1^T, \quad P_{\mathcal{N}'} = \mathcal{P}_{\mathcal{N}(J^T)} + \mu^2 U_1\Sigma_1^{\#}\Sigma_1^{-1}U_1^T, \tag{2.6}$$

where $\mathcal{P}_{\mathcal{R}(J)} = U_1U_1^T$ and $\mathcal{P}_{\mathcal{N}(J^T)} = U_2U_2^T$ are the orthogonal projections on $\mathcal{R}(J)$ and $\mathcal{N}(J^T)$, respectively. Similarily, it is easily shown that

$$P_{\mathcal{R}'} = \mathcal{P}_{\mathcal{R}(J^T)} - \mu^2 V_1\Sigma_1^{\#}\Sigma_1^{-1}V_1^T, \quad P_{\mathcal{N}} = \mathcal{P}_{\mathcal{N}(J)} + \mu^2 V_1\Sigma_1^{\#}\Sigma_1^{-1}V_1^T, \tag{2.7}$$

where $\mathcal{P}_{\mathcal{R}(J^T)} = V_1V_1^T$ and $\mathcal{P}_{\mathcal{N}(J)} = V_2V_2^T$ are the orthogonal projections on $\mathcal{R}(J^T)$ and $\mathcal{N}(J)$, respectively. Thus, we can conclude that the filter matrices are close to the corresponding projections and that

$$P_{\mathcal{R}} \to \mathcal{P}_{\mathcal{R}(J)},\ P_{\mathcal{N}'} \to \mathcal{P}_{\mathcal{N}(J^T)},\ P_{\mathcal{R}'} \to \mathcal{P}_{\mathcal{R}(J^T)},\ P_{\mathcal{N}} \to \mathcal{P}_{\mathcal{N}(J)} \tag{2.8}$$

as μ tends to zero.

When working with the Tikhonov inverse and filter matrices there are several important relations that are useful.

LEMMA 2.1.

(1) *The filter matrices are symmetric with eigenvalues (equal to the singular values) in* $[0, 1]$.

(2) *The following relations hold:*

$$J^{\#} = J^{\dagger}P_{\mathcal{R}} = P_{\mathcal{R}'}J^{\dagger} \tag{2.9}$$

and

$$\mu^2(J^{\#})^T = JP_{\mathcal{N}} = P_{\mathcal{N}'}J. \tag{2.10}$$

(3) *The filter matrices satisfy the commutative rules*

$$P_{\mathcal{R}}P_{\mathcal{N}'} = P_{\mathcal{N}'}P_{\mathcal{R}}, \quad P_{\mathcal{R}'}P_{\mathcal{N}} = P_{\mathcal{N}}P_{\mathcal{R}'}. \tag{2.11}$$

Proof.

(1) The form of the filter matrices in (2.5) show that they are symmetric and that the eigenvalues of $P_{\mathcal{R}}$ and $P_{\mathcal{R}'}$ are $\sigma_i^2/(\sigma_i^2 + \mu^2) \leq 1$. Further, these eigenvalues are nonnegative and thus the eigenvalues of $P_{\mathcal{N}'} = I - P_{\mathcal{R}}$ and $P_{\mathcal{N}} = I - P_{\mathcal{R}'}$ will be nonnegative and less than or equal to one.

(2) From the form of $J^{\#} = V_1\Sigma_1^{\#}U_1^T$ in (2.3), $J^{\dagger} = V_1\Sigma_1^{-1}U_1^T$ in (2.4), and (2.5) we get

$$J^{\#} = V_1\Sigma_1^{\#}U_1^T = V_1\Sigma_1^{-1}U_1^T \cdot U_1\Sigma_1^{\#}\Sigma_1 U_1^T = J^{\dagger}P_{\mathcal{R}},$$

showing the first identity in (2.9). From (2.5) we have

$$P_{\mathcal{R}'}J^{\dagger} = V_1\Sigma_1^{\#}\Sigma_1 V_1^T \cdot V_1\Sigma_1^{-1}U_1^T = V_1\Sigma_1^{\#}U_1^T = J^{\#},$$

establishing the second identity in (2.9).

To show $\mu^2(J^{\#})^T = JP_{\mathcal{N}}$ in (2.10) we use (2.6) and (2.7) to get

$$\begin{aligned} JP_{\mathcal{N}} &= J(\mathcal{P}_{\mathcal{N}(J)} + \mu^2 V_1 \Sigma_1^{\#} \Sigma_1^{-1} V_1^T)\mu^2 J V_1 \Sigma_1^{\#} \Sigma_1^{-1} V_1^T \\ &= \mu^2 U_1 \Sigma_1 V_1^T \cdot V_1 \Sigma_1^{\#} \Sigma_1^{-1} V_1^T = \mu^2 U_1 \Sigma_1^{\#} V_1^T = \mu^2 (J^{\#})^T. \end{aligned}$$

Further,

$$P_{\mathcal{N}'} J = (\mathcal{P}_{\mathcal{N}(J^T)} + \mu^2 U_1 \Sigma_1^{\#} \Sigma_1^{-1} U_1^T) J = \mu^2 U_1 \Sigma_1^{\#} \Sigma_1^{-1} U_1^T J = \mu^2 (J^{\#})^T,$$

concluding the proof of (2.10).

(3) From (2.5), (2.6), and (2.7) we get

$$P_{\mathcal{R}} P_{\mathcal{N}'} = U_1 \Sigma_1^{\#} \Sigma_1 U_1^T \cdot (\mathcal{P}_{\mathcal{N}(J^T)} + \mu^2 U_1 \Sigma_1^{\#} \Sigma_1^{-1} U_1^T) = \mu^2 U_1 (\Sigma_1^{\#})^2 U_1^T$$

and

$$P_{\mathcal{N}'} P_{\mathcal{R}} = (\mathcal{P}_{\mathcal{N}(J^T)} + \mu^2 U_1 \Sigma_1^{\#} \Sigma_1^{-1} U_1^T) \cdot U_1 \Sigma_1^{\#} \Sigma_1 U_1^T = \mu^2 U_1 (\Sigma_1^{\#})^2 U_1^T,$$

attaining the first relation in (2.11). The second relation in (2.11) is derived in completely the same way or by just substituting J^T for J in the first relation. □

3. The filter matrices in a regularization context. Assume that $b = b_{\text{exact}} + \delta b$, where δb is the noise in b. Further, define the least norm solution

$$x^{\dagger} = J^{\dagger} b \tag{3.1}$$

as the nonfiltered solution and the "exact" solution as

$$x_{\text{exact}} = J^{\dagger} b_{\text{exact}}. \tag{3.2}$$

Generally, for a very ill-conditioned J the least norm solution $x^{\dagger}$ is not a good solution since it does not filter out the noise in b. On the other hand, from (2.9) we see that the Tikhonov regularized solution is

$$x_{\text{reg}} = J^{\#} b = J^{\dagger} P_{\mathcal{R}} b \tag{3.3}$$

giving

$$x_{\text{reg}} = J^{\dagger} P_{\mathcal{R}} b_{\text{exact}} + J^{\dagger} P_{\mathcal{R}} \delta b,$$

clearly showing how the regularization filters the noise. Further, from (2.9) we get

$$x_{\text{reg}} = P_{\mathcal{R}'} x_{\text{exact}} + J^{\dagger} P_{\mathcal{R}} \delta b$$

or, using the definition of the filter matrix $P_{\mathcal{R}'} = I - P_{\mathcal{N}'}$,

$$x_{\text{reg}} - x_{\text{exact}} = -P_{\mathcal{N}} x_{\text{exact}} + J^{\dagger} P_{\mathcal{R}} \delta b. \tag{3.4}$$

The first term in the right-hand side of (3.4) is the regularization error.

4. A new perturbation identity using filter matrices. In [4] the perturbation identity

$$\widetilde{J}^\dagger - J^\dagger = -\widetilde{J}^\dagger F J^\dagger + \widetilde{J}^\dagger (\widetilde{J}^\dagger)^T F^T \mathcal{P}_{\mathcal{N}(J^T)} + \mathcal{P}_{\mathcal{N}(\tilde{J})} F^T (J^\dagger)^T J^\dagger \tag{4.1}$$

with the perturbation $F = \widetilde{J} - J$ was derived. In this section we will generalize this identity to the Tikhonov problem using filter matrices instead of projections.

The following lemma is the first step toward the general perturbation identity.

LEMMA 4.1. *Define* $F = \widetilde{J} - J$. *Then*

$$\widetilde{J}^\# - J^\# = -\widetilde{J}^\# F J^\# + \frac{1}{\mu^2} \widetilde{P}_{\mathcal{N}} F^T P_{\mathcal{N}'}. \tag{4.2}$$

Proof. From the definition of F we have

$$\begin{aligned} \frac{1}{\mu^2} \widetilde{P}_{\mathcal{N}} F^T P_{\mathcal{N}'} &= \frac{1}{\mu^2} \widetilde{P}_{\mathcal{N}} \widetilde{J}^T P_{\mathcal{N}'} - \frac{1}{\mu^2} \widetilde{P}_{\mathcal{N}} J^T P_{\mathcal{N}'} \quad \text{(from (2.10))} \\ &= \widetilde{J}^\# (I - J J^\#) - (I - \widetilde{J}^\# \widetilde{J}) J^\# \quad \text{(from the definition of } F) \\ &= \widetilde{J}^\# - J^\# + \widetilde{J}^\# F J^\#, \end{aligned} \tag{4.3}$$

proving the lemma. ∎

We are now ready to state the main perturbation identity of this section.

THEOREM 4.2. *Let* $F = \widetilde{J} - J$. *Then*

$$\begin{aligned} \widetilde{J}^\# - J^\# = &-\widetilde{J}^\# F J^\# + \widetilde{J}^\# (\widetilde{J}^\#)^T F^T P^2_{\mathcal{N}'} + \widetilde{P}^2_{\mathcal{N}} F^T (J^\#)^T J^\# \\ &+ \mu^2 \widetilde{J}^\# (\widetilde{J}^\#)^T F^T (J^\#)^T J^\# + \frac{1}{\mu^2} \widetilde{P}^2_{\mathcal{N}} F^T P^2_{\mathcal{N}'}. \end{aligned} \tag{4.4}$$

Proof. Using $\widetilde{P}_{\mathcal{R}'} + \widetilde{P}_{\mathcal{N}} = I$ and $P_{\mathcal{R}} + P_{\mathcal{N}'} = I$ we get

$$\begin{aligned} \frac{1}{\mu^2} \widetilde{P}_{\mathcal{N}} F^T P_{\mathcal{N}'} &= \frac{1}{\mu^2} (\widetilde{P}_{\mathcal{R}'} + \widetilde{P}_{\mathcal{N}}) \widetilde{P}_{\mathcal{N}} F^T P_{\mathcal{N}'} (P_{\mathcal{R}} + P_{\mathcal{N}'}) \\ &= \frac{1}{\mu^2} \widetilde{P}_{\mathcal{R}'} \widetilde{P}_{\mathcal{N}} F^T P^2_{\mathcal{N}'} + \frac{1}{\mu^2} \widetilde{P}^2_{\mathcal{N}} F^T P_{\mathcal{N}'} P_{\mathcal{R}} \\ &\quad + \frac{1}{\mu^2} \widetilde{P}_{\mathcal{R}'} \widetilde{P}_{\mathcal{N}} F^T P_{\mathcal{N}'} P_{\mathcal{R}} + \frac{1}{\mu^2} \widetilde{P}^2_{\mathcal{N}} F^T P^2_{\mathcal{N}'}. \end{aligned} \tag{4.5}$$

We get (4.4) by inserting $P_{\mathcal{N}'} P_{\mathcal{R}} = \mu^2 (J^\#)^T J^\#$ and $\widetilde{P}_{\mathcal{R}'} \widetilde{P}_{\mathcal{N}} = \mu^2 \widetilde{J}^\# (\widetilde{J}^\#)^T$, attained from (2.10). ∎

By comparing the new result (4.4) with (4.1) we notice the similarity and the extra two terms

$$\mu^2 \widetilde{J}^\# (\widetilde{J}^\#)^T F^T (J^\#)^T J^\# + \frac{1}{\mu^2} \widetilde{P}^2_{\mathcal{N}} F^T P^2_{\mathcal{N}'}. \tag{4.6}$$

Moreover, it is easily seen that the first three terms in the right-hand side of (4.4) tend to the corresponding three terms in (4.1) and, consequently, that the two terms in (4.6) tend to zero as μ tends to zero.

We end this section with a perturbation identity for the Tikhonov problem that is a direct consequence of Theorem 4.2.

COROLLARY 4.3. *Let* $\tilde{x}_{\rm reg} = \widetilde{J}^{\#} b$ *and* $x_{\rm reg} = J^{\#} b$ *be the solutions of the perturbed and unperturbed Tikhonov problem* (1.2). *Then*

$$\begin{aligned}\tilde{x}_{\rm reg} - x_{\rm reg} &= -\widetilde{J}^{\#} F x_{\rm reg} + \frac{1}{\mu^2}\widetilde{P}_{\mathcal{N}} F^T (b - J x_{\rm reg}) \\ &= -\widetilde{J}^{\#} F x_{\rm reg} + \widetilde{J}^{\#} (\widetilde{J}^{\#})^T F^T P_{\mathcal{N}'} (b - J x_{\rm reg}) + \widetilde{P}^2_{\mathcal{N}} F^T (\widetilde{J}^{\#})^T x_{\rm reg} \\ &\quad + \mu^2 \widetilde{J}^{\#} (\widetilde{J}^{\#})^T F^T (J^{\#})^T x_{\rm reg} + \frac{1}{\mu^2}\widetilde{P}^2_{\mathcal{N}} F^T P_{\mathcal{N}'} (b - J x_{\rm reg}). \end{aligned} \tag{4.7}$$

Proof. The identity (4.7) follows by multiplying (4.4) with b and using $P_{\mathcal{N}'} b = b - J x_{\rm reg}$. ∎

5. Differentiating the filter matrices.

THEOREM 5.1. *Let* dJ *be the known differential of* J, *and let* $dP_{\mathcal{R}}$ *and* $dP_{\mathcal{R}'}$ *be the differentials of the filter matrices. Then*

$$dJ^{\#} = -J^{\#} dJ J^{\#} + (J^T J + \mu^2 I)^{-1} (dJ)^T P_{\mathcal{N}'}, \tag{5.1}$$

$$dP_{\mathcal{R}} = P_{\mathcal{N}'} dJ J^{\#} + (J^{\#})^T (dJ)^T P_{\mathcal{N}'}, \quad dP_{\mathcal{N}'} = -dP_{\mathcal{R}}, \tag{5.2}$$

$$dP_{\mathcal{R}'} = J^{\#} dJ P_{\mathcal{N}} + P_{\mathcal{N}} (dJ)^T (J^{\#})^T, \quad dP_{\mathcal{N}} = -dP_{\mathcal{R}'}. \tag{5.3}$$

Proof. The identity (5.1) follows directly from the perturbation identity (4.2) or by differentiating $J^{\#} = (J^T J + \mu^2 I)^{-1} J^T$.

Identity (5.2) is proved by using

$$dP_{\mathcal{R}} = (dJ) J^{\#} + J (dJ^{\#})$$

and taking the expression for $dJ^{\#}$ from identity (5.1).

To derive the identity (5.3) we begin by differentiating $dP_{\mathcal{R}'} = dJ^{\#} J + J^{\#} dJ$ and then we insert the expression for $dJ^{\#}$ in (5.1) to get

$$dP_{\mathcal{R}'} = J^{\#} dJ P_{\mathcal{N}} + (J^T J + \mu^2 I)^{-1} (dJ)^T P_{\mathcal{N}'} J. \tag{5.4}$$

We get (5.3) by using $\mu^2 (J^{\#})^T = P_{\mathcal{N}'} J$ from (2.10) and

$$(J^T J + \mu^2 I)^{-1} = \frac{1}{\mu^2}(J^T J + \mu^2 I)^{-1}(J^T J + \mu^2 I - J^T J) = \frac{1}{\mu^2}(I - J^{\#} J) = \frac{1}{\mu^2} P_{\mathcal{N}}$$

in (5.4). ∎

We note that by letting $\mu \to 0$ in identity (5.2) we get the well-known identity in [2],

$$d\mathcal{P}_{\mathcal{R}(J)} = \mathcal{P}_{\mathcal{N}(J^T)} dJ J^{\dagger} + (J^{\dagger})^T (dJ)^T \mathcal{P}_{\mathcal{N}(J^T)}. \tag{5.5}$$

Acknowledgments. First we wish to thank the anonymous referees for reading the manuscript with such care and expertise.

Second, we wish to thank Yimin Wei, Fudan University, China, for reading and commenting on the manuscript.

REFERENCES

[1] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[2] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[3] P.C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, PA, 1998.

[4] P.-Å. WEDIN, *Perturbation theory for pseudo-inverses*, BIT, 13 (1973), pp. 217–232.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 282–305

S^+: EFFICIENT 2D SPARSE LU FACTORIZATION ON PARALLEL MACHINES*

KAI SHEN†, TAO YANG†, AND XIANGMIN JIAO‡

Abstract. Static symbolic factorization coupled with supernode partitioning and asynchronous computation scheduling can achieve high gigaflop rates for parallel sparse LU factorization with partial pivoting. This paper studies properties of elimination forests and uses them to optimize supernode partitioning/amalgamation and execution scheduling. It also proposes supernodal matrix multiplication to speed up kernel computation by retaining the BLAS-3 level efficiency and avoiding unnecessary arithmetic operations. The experiments show that our new design with proper space optimization, called S^+, improves our previous solution substantially and can achieve up to 10 GFLOPS on 128 Cray T3E 450MHz nodes.

Key words. Gaussian elimination with partial pivoting, LU factorization, sparse matrices, elimination forests, supernode amalgamation and partitioning, asynchronous computation scheduling

AMS subject classifications. 65F50, 65F05

PII. S0895479898337385

1. Introduction.

The solution of sparse linear systems is a computational bottleneck in many scientific computing problems. When dynamic pivoting is required to maintain numerical stability in direct methods for solving nonsymmetric linear systems, it is challenging to develop high performance parallel code because pivoting causes severe caching miss and load imbalance on modern architectures with memory hierarchies. The previous work has addressed parallelization on shared memory platforms or with restricted pivoting [4, 13, 15, 19]. Most notably, the recent shared memory implementation of SuperLU has achieved up to 2.58 GFLOPS on 8 Cray C90 nodes [4, 5, 23]. For distributed memory machines, we proposed an approach that adopts a static symbolic factorization scheme to avoid data structure variation [10, 11]. Static symbolic factorization eliminates the runtime overhead of dynamic symbolic factorization with a price of overestimated fill-ins and, thereafter, extra computation [15]. However, the static data structure allowed us to identify data regularity, maximize the use of BLAS-3 operations, and utilize task graph scheduling techniques and efficient runtime support [12] to achieve high efficiency.

This paper addresses three issues to further improve the performance of parallel sparse LU factorization with partial pivoting on distributed memory machines. First, we study the use of elimination trees in optimizing matrix partitioning and task scheduling. Elimination trees or forests are used extensively in sparse Cholesky factorization [18, 26, 27] because they have a more compact representation of parallelism than task graphs. For sparse LU factorization, the traditional approach uses the elimination tree of A^TA, which can produce excessive false computational dependency. In this paper, we use the elimination trees (forest) of A to guide matrix

*Received by the editors April 15, 1998; accepted for publication (in revised form) by S. Vavasis February 14, 2000; published electronically June 20, 2000. This work was supported in part by NSF CCR-9702640 and by DARPA through UMD (ONR Contract number N6600197C8534).
http://www.siam.org/journals/simax/22-1/33738.html
†Department of Computer Science, University of California at Santa Barbara, CA 93106 (kshen@cs.ucsb.edu, tyang@cs.ucsb.edu).
‡Department of Computer Science, University of Illinois at Urbana–Champaign, IL 61801 (jiao@cs.uiuc.edu).

partitioning and parallelism control in LU factorization. We show that improved supernode partitioning and amalgamation effectively control extra fill-ins and produce optimized supernodal partitioning. We also use elimination forests to identify data dependence and potential concurrency among pivoting and updating tasks and thus maximize utilization of limited parallelism.

Second, we propose a fast and space-efficient kernel for supernode-based matrix multiplication to improve the performance of sparse LU factorization. This is based on the observation that nonzero submatrices generated by supernodal partitioning and amalgamation have special patterns. Namely, they contain either dense subrows or subcolumns. This new kernel avoids unnecessary arithmetic operations while it retains the BLAS-3 level efficiency.

Third, we evaluate the space requirement of static factorization and propose an optimization scheme that acquires memory on the fly only when it is necessary. This scheme can effectively control peak memory usage, especially when static symbolic factorization overestimates fill-ins excessively.

Our new design with these optimizations, called S^+, improves our previous code by more than 50% in execution time. In particular, S^+ without space optimization achieved up to 8.25 GFLOPS on 128 T3E 300MHz nodes and 10.85 GFLOPS[1] on 128 T3E 450MHz nodes. The space optimization technique slightly degrades overall time efficiency, but it reduces space requirement by up to 68% in some cases. S^+ with space optimization can still deliver up to 10.00 GFLOPS on 128 Cray 450MHz T3E nodes. Notice that we only count *true* operations, in the sense that no extra arithmetic operation introduced by static factorization or amalgamation is included in computing gigaflop rates of our algorithm.

The rest of this paper is organized as follows. Section 2 gives the background knowledge for sparse LU factorization. Section 3 presents a modified definition and properties of elimination trees for sparse LU factorization and their applications in supernode partitioning and amalgamation. Section 4 describes our strategies of exploiting 2D asynchronous parallelism. Section 5 discusses a fast matrix multiplication kernel suitable for submatrices derived from supernode partitioning. Section 6 presents experimental results on Cray T3E. Section 7 discusses space optimization for S^+. Section 8 concludes the paper. A summary of notations and the proof for each theorem are listed in the appendix.

2. Background. LU factorization with partial pivoting decomposes a nonsymmetric sparse matrix A into two matrices L and U, such that $PA = LU$, where L is a unit lower triangular matrix, U is an upper triangular matrix, and P is a permutation matrix containing pivoting information.

Static symbolic factorization. A static symbolic factorization approach is proposed in [14] to identify the worst case nonzero patterns for sparse LU factorization without knowing numerical values of elements. The basic idea is to statically consider all possible pivoting choices at each elimination step, and space is allocated for all possible nonzero entries. Static symbolic factorization annihilates data structure variation, and hence it improves predictability of resource requirements and enables static optimization strategies. On the other hand, dynamic factorization, which is used in SuperLU [4, 23], provides more accurate control of data structures on the fly.

[1] We reported a performance record of 11.04 GFLOPS in an earlier paper [29]. We later found that the operation count included extra computation due to amalgamation. In this paper, we disabled amalgamation in operation counting.

But it is challenging to parallelize dynamic factorization with low runtime overhead on distributed memory machines.

The static symbolic factorization for an $n \times n$ matrix is outlined as follows. At each step $k(1 \leq k < n)$, each row $i \geq k$ that has a nonzero element in column k is a *candidate pivot row* for row k. As the static symbolic factorization proceeds, at step k the nonzero structure of *each* candidate pivot row is replaced by the union of the structures of all these candidate pivot rows except the elements in the first $k-1$ columns. Using an efficient implementation [21] for the symbolic factorization algorithm proposed in [14], this preprocessing step can be very fast. For example, it costs less than one second for most of our test matrices, and at worst it costs two seconds on a single node of Cray T3E. The memory requirement is also fairly small. If LU factorization is used in an iterative numerical method, then the cost of symbolic factorization together with other preprocessing is amortized over multiple iterations.

In the previous work, we show that static factorization does not produce too many fill-ins for most of our test matrices, even for large matrices using a simple matrix ordering strategy (minimum degree ordering) [10, 11]. For a few matrices that we have tested, static factorization generates an excessive number of fill-ins. In section 7, we discuss space optimization for S^+ in addressing such a problem.

***L/U* supernode partitioning.** After the fill-in pattern of a matrix is predicted, the matrix is further partitioned using a supernodal approach to improve caching performance. In [23], a nonsymmetric supernode is defined as a group of consecutive columns, in which the corresponding L part has a dense lower triangular block on the diagonal and the same nonzero pattern below the diagonal. Based on this definition, in each column block the L part only contains dense subrows. We call this partitioning scheme L *supernode partitioning*. Here by "subrow" we mean the contiguous part of a row within a supernode.

After an L supernode partitioning has been performed on a sparse matrix A, the same partitioning is applied to the rows of A to further break each supernode into submatrices. This is also known as U *supernode partitioning*. Since coarse-grain partitioning can reduce available parallelism and produce large submatrices that do not fit into the cache, an upper bound on the supernode size is usually enforced in the L/U supernode partitioning. After the L/U supernode partitioning, each diagonal submatrix is dense, and each nonzero off-diagonal submatrix in the L part contains only dense subrows, and furthermore, each nonzero submatrix in the U part of A contains only dense subcolumns [11]. This is the key to maximize the use of BLAS-3 subroutines [7] in our algorithm. And on most current commodity processors with memory hierarchies, BLAS-3 subroutines usually outperform BLAS-2 subroutines substantially when implementing the same functionality [7]. Figure 1 illustrates an example of a partitioned sparse matrix, and the black areas depict dense submatrices, subrows, and subcolumns.

Data mapping. After symbolic factorization and matrix partitioning, a partitioned sparse matrix A has $N \times N$ submatrix blocks. For example, the matrix in Figure 1 has 8×8 submatrices. Let $A_{i,j}$ denote the submatrix in A with row block index i and column block index j. Let $L_{i,j}$ and $U_{i,j}$ denote a submatrix in the lower and upper triangular part of matrix A, respectively. For block-oriented matrix computation, 1D column block cyclic mapping and 2D block cyclic mapping are commonly used. In 1D column block cyclic mapping, a column block of A is assigned to one processor. In 2D mapping, processors are viewed as a 2D grid, and a column block is assigned to a column of processors. 2D sparse LU factorization is more scalable than

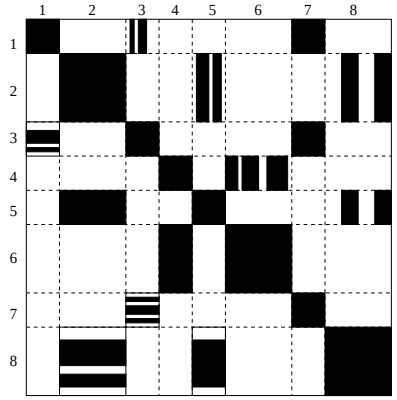


FIG. 1. *Example of a partitioned sparse matrix.*

```
for k = 1 to N
    Perform task Factor(k);
    for j = k + 1 to N with U_{k,j} ≠ 0
       Perform task Update(k, j);
    endfor
endfor
```

FIG. 2. *Partitioned sparse LU factorization with partial pivoting.*

1D data mapping [10]. However, 2D mapping introduces more overhead for pivoting and row swapping. Since asynchronous execution requires extensive use of buffers, in designing 2D codes we need to pay special attention to the usage of buffer space so that our 2D code is able to factorize larger matrices under memory constraints.

Program partitioning. Each column block k is associated with two types of tasks: $Factor(k)$ and $Update(k, j)$ for $1 \leq k < j \leq N$. Task $Factor(k)$ factorizes all the columns in the kth column block, and its function includes finding the pivoting sequence associated with those columns and updating the lower triangular portion of column block k. The pivoting sequence is held until the factorization of the kth column block is completed. Then the pivoting sequence is applied to the rest of the matrix. This is called "delayed pivoting" [3]. Task $Update(k, j)$ uses column block k $(L_{k,k}, L_{k+1,k}, \ldots, L_{N,k})$ to modify column block j. That includes "row swapping" that applies the pivoting derived by $Factor(k)$ to column block j, "scaling" that uses the factorized submatrix $L_{k,k}$ to scale $U_{k,j}$, and "updating" that uses submatrices $L_{i,k}$ and $U_{k,j}$ to modify $A_{i,j}$ for $k+1 \leq i \leq N$. Figure 2 outlines the partitioned LU factorization algorithm with partial pivoting.

3. Elimination forests and nonsymmetric supernode partitioning. In this section, we study properties of elimination forests [1, 15, 16, 25][2] and use them to design more robust strategies for supernode partitioning and parallelism detection. As a result, both sequential and parallel versions of our code can be improved.

We will use the following notations in our discussion. Let A be the given $n \times n$ sparse matrix. Notice that the nonzero structure of matrix A changes after symbolic

[2] An elimination forest has only one tree when the corresponding sparse matrix is irreducible. In that case, it is also called an elimination tree.

factorization and the algorithm design discussed in the rest of this paper addresses A after symbolic factorization. Let $a_{i,j}$ be the element in A with row index i and column index j, and let $a_{i:j,s:t}$ be the submatrix in A from row i to row j and from column s to t. Let l_k be column k in the lower triangular part, and let u_k be row k in the upper triangular part of A after symbolic factorization. Notice that both l_k and u_k include $a_{k,k}$. To emphasize nonzero patterns of A, we use the symbol "$\hat{}$" to express the nonzero structure after symbolic factorization. Expression $\hat{a}_{i,j} \neq 0$ means that $a_{i,j}$ is not zero after symbolic factorization. We assume that every diagonal element in the original sparse matrix is nonzero. Notice that for any nonsingular matrix that does not have a zero-free diagonal, it is always possible to permute the rows of A to obtain a matrix with zero-free diagonal [8]. Let $\hat{l}_k$ be the index set of nonzeros in l_k, i.e., $\{i \mid \hat{a}_{i,k} \neq 0 \wedge i \geq k\}$. Similarly, let $\hat{u}_k$ be the index set of nonzeros in u_k, i.e., $\{j \mid \hat{a}_{k,j} \neq 0 \wedge j \geq k\}$. Symbol $|\hat{l}_k|$ (or $|\hat{u}_k|$) denotes the cardinality of $\hat{l}_k$ (or $\hat{u}_k$).

3.1. The definition of elimination forests. We study the elimination forest of a matrix that may or may not be reducible. Previous research on elimination forests has shown that an elimination forest contains information about all potential dependency if the corresponding sparse matrix is irreducible [1, 15, 16, 25]. Although it is always possible to decompose a reducible matrix into several smaller irreducible matrices, the decomposition introduces extra burden on software design and implementation. Instead, we generalize the original definition of elimination tree to reducible matrices. Our definition, listed in Definition 3.1, differs from the original definition by imposing condition $|\hat{l}_k| > 1$. Imposing this condition not only avoids some false dependency but also allows us to derive the same properties for irreducible matrices and for reducible matrices, which are summarized in Theorems 3.2 and 3.4. Note that when A is irreducible, the condition $|\hat{l}_k| > 1$ holds for all $1 \leq k < n$ and the new definition generates the same elimination forest as the original definition. In practice, we find that some test matrices can have up to 50% of columns with zero lower-diagonal nonzeros after symbolic factorization.

DEFINITION 3.1. *An* LU elimination forest *for an $n \times n$ matrix A has n vertices numbered from 1 to n. For any two numbers k and j ($k < j$), there is an edge from vertex j to vertex k in the forest if $a_{k,j}$ is the first off-diagonal nonzero in $\hat{u}_k$ and $|\hat{l}_k| > 1$. Vertex j is called the* parent *of vertex k, and vertex k is called a* child *of vertex j.*

An elimination forest for a given matrix can be generated in a time complexity of $O(n)$ if computed as a byproduct of symbolic factorization. Figure 3 illustrates a sparse matrix after symbolic factorization and its elimination forest. We now discuss two properties of an elimination forest for a general sparse matrix.

THEOREM 3.2. *If vertex j is an ancestor of vertex k in the elimination forest, then $\{r \mid r \in \hat{l}_k \wedge j \leq r \leq n\} \subseteq \hat{l}_j$, and $\{c \mid c \in \hat{u}_k \wedge j \leq c \leq n\} \subseteq \hat{u}_j$.*

Theorem 3.2 (illustrated in Figure 4) captures the structural containment between two columns in L and between two rows in U. It indicates that the nonzero structure of l_j (or u_j) subsumes l_k (or u_k) if the corresponding vertices have an ancestor relationship. This information will be used for designing supernode partitioning with amalgamation in the next subsection.

DEFINITION 3.3. *Let $j > k$; l_k* directly updates *l_j if task Update(k, j) is performed in LU factorization, i.e., $\hat{a}_{k,j} \neq 0$ and $|\hat{l}_k| > 1$. l_k* indirectly updates *l_j if there is a sequence $s_1, s_2, \ldots, s_p$ such that $s_1 = k$, $s_p = j$, and l_{s_q} directly updates $l_{s_{q+1}}$ for each $1 \leq q \leq p - 1$.*

THEOREM 3.4. *Let $k < j$; l_k directly or indirectly updates l_j in LU factorization*

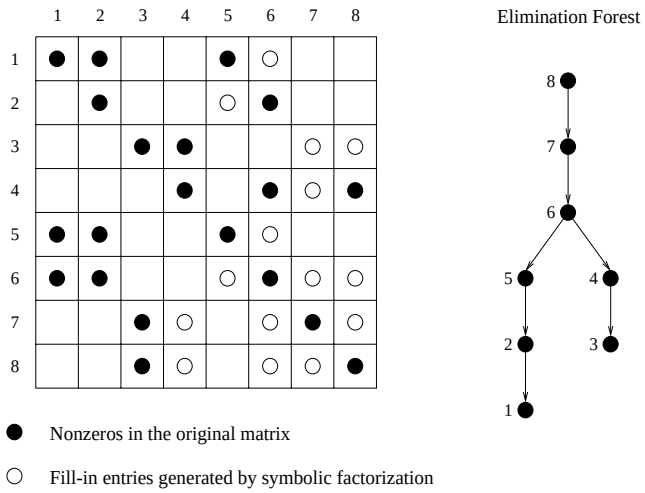


FIG. 3. *A sparse matrix and its elimination forest.*

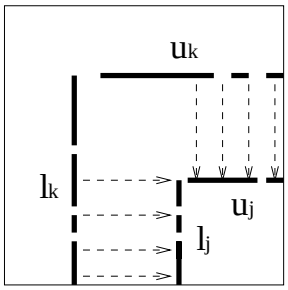


FIG. 4. *An illustration of Theorem* 3.2 *(vertex j is an ancestor of vertex k in the elimination forest).*

if and only if vertex j is an ancestor of vertex k in the elimination forest.

Theorem 3.4 indicates dependency information during numerical factorization, which can guide the scheduling of asynchronous parallelism.

3.2. 2D L/U supernode partitioning and amalgamation. Given a non-symmetric matrix A after symbolic factorization, in [11] we have described a two-stage L/U supernode partitioning method: At stage 1, a group of consecutive columns that have the same structure in the L part is considered as one supernode column block. Then the L part is sliced as a set of consecutive column blocks. After an L supernode partition has been obtained, at stage 2 the same partition is applied to rows of the matrix to break each supernode column block further into submatrices.

We examine how elimination forests can be used to guide and improve the 2D L/U supernode partitioning. The following corollary is a straightforward result of Theorem 3.2, and it shows that we can easily traverse an elimination forest to identify supernodes. Notice that each element in a dense structure can be a nonzero or a fill-in due to static symbolic factorization.

COROLLARY 3.5. *If for each $k \in \{s+1, s+2, \ldots, t\}$ vertex k is the parent of vertex $k-1$ and $|\hat{l}_k| = |\hat{l}_{k-1}| - 1$, then after symbolic factorization,* (1) *diagonal block $a_{s:t,\ s:t}$ is completely dense,* (2) *$a_{t+1:n,s:t}$ contains only dense subrows, and* (3)

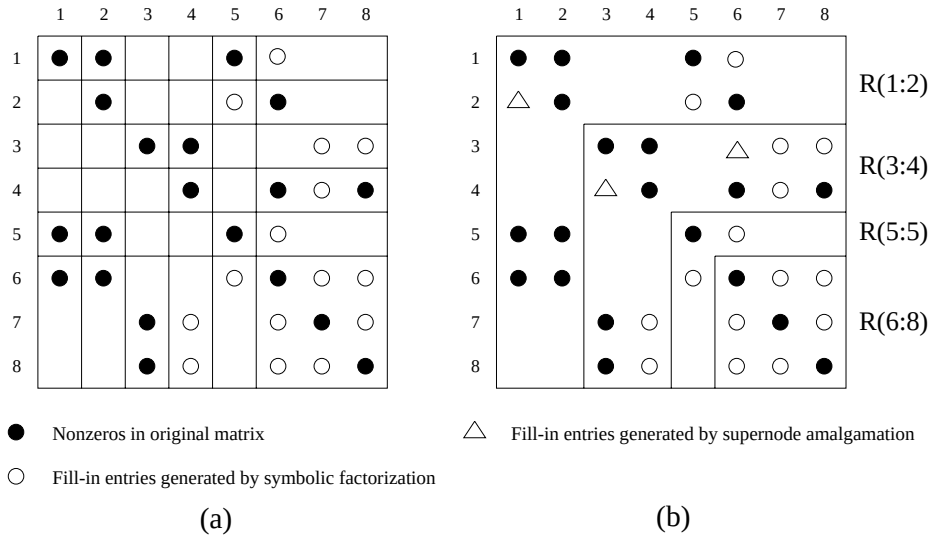


FIG. 5. (a) *Supernode partitioning for the matrix in Figure* 3*;* (b) *The result of supernode amalgamation with* 4 *related* L/U *supernodes.*

$a_{s:t,t+1:n}$ *contains only dense subcolumns.*

The partitioning algorithm using the above corollary is briefly summarized as follows. For each pair of two consecutively numbered vertices with the parent/child relationship in the elimination forest, we check the size difference between the two corresponding columns in the L part. If the difference is one, we assign these two columns into an L supernode. Since if a submatrix in a supernode is too large, it won't fit into the cache and also because large grain partitioning reduces available parallelism, we usually enforce an upper bound on the supernode size. Notice that U partitioning is applied after the L partitioning is completed. We need not check any constraint on U because as long as a child/parent pair $(i, i-1)$ satisfies $|\hat{l}_i| = |\hat{l}_{i-1}|-1$, it also satisfies $|\hat{u}_i| = |\hat{u}_{i-1}| - 1$ due to Theorem 1 in [10, 11]. Hence the structures of u_i and u_{i-1} are identical. Figure 5(a) illustrates supernode partitioning of the sparse matrix in Figure 3. There are 6 L/U supernodes in this figure. From the L partitioning point of view, columns from 1 to 5 are not grouped, but columns 6, 7, and 8 are clustered together.

For most of the test matrices in our experiments, the average supernode size after the above partitioning strategy is very small, about 1.5 to 2 columns. This leads to relatively fine grained computation. In practice, amalgamation is commonly adopted to increase the average supernode size by introducing some extra zero entries in the dense structures of supernodes. In this way, caching performance can be improved and interprocessor communication overhead may be reduced. For sparse Cholesky factorization (e.g., [26]), the basic idea of amalgamation is to relax the restriction that all the columns in a supernode must have exactly the same off-diagonal nonzero structure. In a Cholesky elimination tree, a parent could be merged with its children if merging does not introduce too many extra zero entries into a supernode. Row and column permutations are needed if the parent is not consecutive with its children. For sparse LU factorization, such a permutation may alter the result of symbolic factorization. In our previous approach [11], we simply compare consecutive columns of the L part and make a decision on merging if the total number of difference is

under a preset threshold. This approach is simple, resulting in a bounded number of extra zero entries included in the dense structure of an L supernode. However, the result of partitioning may lead to too many extra zero entries in the dense structure of a U supernode. Using Theorem 3.2, we can remedy this problem as follows by partitioning L and U parts simultaneously and controlling the number of fill-ins in both L and U.

We consider a supernode containing elements from both L and U parts and refer to a supernode after amalgamation as a *relaxed L/U supernode*. The definition is listed below.

DEFINITION 3.6. *A* relaxed L/U supernode *$R(s{:}t)$ contains three parts: the diagonal block $a_{s:t,s:t}$, the L supernode part $a_{t+1:n,s:t}$, and the U supernode part $a_{s:t,t+1:n}$. The* supernode size *of $R(s:t)$ is $t-s+1$.*

A partitioning example illustrated in Figure 5(b) has four relaxed L/U supernodes: $R(1:2)$, $R(3:4)$, $R(5:5)$, and $R(6:8)$. The following corollary, which is also a straightforward result of Theorem 3.2, can be used to bound the nonzero structure of a relaxed L/U supernode.

COROLLARY 3.7. *If for each k where $s+1 \le k \le t$, vertex k is the parent of vertex $k-1$ in an elimination forest, then $\{i \mid i \in \hat{l}_k \wedge t \le i \le n\} \subseteq \hat{l}_t$ and $\{j \mid j \in \hat{u}_k \wedge t \le j \le n\} \subseteq \hat{u}_t$.*

Using Corollary 3.7, in $R(s:t)$ the ratio of extra fill-ins introduced by amalgamation compared with the actual nonzeros can be computed as

$$z = \frac{(t-s+1)^2 + (t-s+1) \times (|\hat{l}_t| + |\hat{u}_t| - 2)}{nz(R(s:t))} - 1,$$

where $nz()$ gives the number of nonzero elements in the corresponding structure including fill-ins created by symbolic factorization. Also notice that both $\hat{l}_t$ and $\hat{u}_t$ include diagonal element $a_{t,t}$.

Thus our heuristic for 2D partitioning is to traverse the elimination forest and find relaxed supernodes $R(s:t)$ satisfying the following conditions:

(1) For each i where $s+1 \le i \le t$, vertex i is the parent of vertex $i-1$ in the elimination forest,
(2) the extra fill-in ratio, z, is less than the predefined threshold, and
(3) $t-s+1 \le$ the predefined upper bound for supernode sizes.

The complexity of such a partitioning algorithm with amalgamation is $O(n)$, which is very low and is made possible by Corollary 3.7. Our experiments show that the above strategy is very effective. The number of total extra fill-ins doesn't change much when the upper bound for z is in the range of $10-100\%$, and it seldom exceeds 2% of the total nonzeros in the whole matrix. In terms of upper bound for supernode size, 25 gives the best caching and parallel performance on the T3E. Thus all the experiments in section 6 are completed with $z \le 30\%$ and *supernode size* ≤ 25. Figure 5(b) is the result of supernode amalgamation for the sparse matrix in Figure 3 using condition $z \le 30\%$.

In the rest of this paper, we will simply refer to relaxed L/U supernodes as supernodes.

Compressed storage scheme for submatrices. In our implementation, every submatrix is stored in a compressed storage scheme with a bit map to indicate its nonzero structure. In addition to the storage saving, the compressed storage scheme can also eliminate certain unnecessary computations on zero elements, which will be discussed in details in section 5. For an L submatrix, its subrows are stored

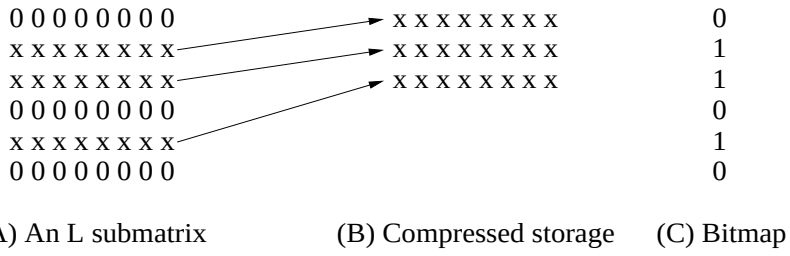


FIG. 6. *An illustration of a compressed storage scheme for an L submatrix.*

in a consecutive space even though their corresponding row numbers may not be consecutive. The bit map is used to identify dense subrows in L submatrices. A bit is set to 0 if the corresponding subrow is 0 and set to 1 otherwise. Figure 6 illustrates such a storage scheme for a 6×8 L submatrix. In this example, the second, third, and fifth subrows are dense and all other subrows are completely zero. The strategy for a U submatrix is the same except in a subcolumn-oriented fashion. Since level-1 cache is not large in practice and the supernode size is limited to fit the cache (limit is 25 on Cray T3E), we can use a 32-bit word to store the bit map of each submatrix and can determine efficiently if a subrow is dense using a single logical "and" operation.

Space overhead for a submatrix includes the bit map and global matrix index. Index information is piggybacked on a message when sending submatrices among processors. In terms of space for bit maps, if a submatrix is completely zero, its bit map vector is not needed. For a nonzero submatrix, the size of its bit map is just one word. Thus numerical values of the sparse matrix always dominate the overall storage requirement and space overhead for bit map vectors is insignificant. It should also be noted that in a future CPU architecture with a large level-1 cache, a 32-bit word may not be sufficient and that some minor changes in the implementation are needed to use two words or more. In this case, using more than one word for a bit vector should not cause space concern because amalgamation ensures that the average submatrix size is not too small. Also this compression scheme can be turned off for an extremely small submatrix (but we do not expect such a thing is needed in practice).

4. 2D asynchronous parallelism exploitation. In this section, we present scheduling strategies for exploiting asynchronous 2D parallelism so that different updating stages can be overlapped. After 2D L/U supernode partitioning and amalgamation, the $n \times n$ sparse matrix A is 2-dimensionally partitioned into $N \times N$ submatrices. Let symbol $A_{i,j}$ denote the submatrix in row block i and column block j, and let $A_{i:j,s:t}$ denote all submatrices from row block i to j and from column block s to t. Let $L_{i,j}$ and $U_{i,j}$ $(i \neq j)$ denote submatrices in the lower and upper triangular parts, respectively. Our 2D algorithm uses the standard cyclic mapping since it tends to distribute data evenly, which is important to solve large problems. In this scheme, p available processors are viewed as a 2D grid: $p = p_r \times p_c$. Then block $A_{i,j}$ is assigned to processor $P_{i \bmod p_r,\ j \bmod p_c}$.

In section 2, we have described two types of tasks involved in LU factorization. One is $Factor(k)$, which is to factorize all the columns in the kth column block, including finding the pivoting sequence associated with those columns. The other is $Update(k, j)$, which is to apply the pivoting sequence derived from $Factor(k)$ to the jth column block and modify the jth column block using the kth column block, where $k < j$ and $U_{k,j} \neq 0$. The 2D data mapping enables parallelization of a single $Factor(k)$

or $Update(k, j)$ task on p_r processors because each column block is distributed into p_r processors. The main challenge is the coordination of pivoting and data swapping across a subset of processors to exploit as much parallelism as possible with low buffer space demand.

For task $Factor(k)$, the computation is distributed among processors in column $k \bmod p_c$ of the processor grid, and global synchronization among this processor column is needed for correct pivoting. To simplify the parallelism control of task $Update(k, j)$ we split it into two subtasks: $ScaleSwap(k)$, which does scaling and delayed row interchange for submatrices $A_{k:N,\ k+1:N}$, and $Update2D(k, j)$, which modifies column block j using column block k. For $ScaleSwap(k)$, the synchronization among processors within the same column of the grid is needed. For $Update2D(k, j)$, no synchronization among processors is needed as long as the desired submatrices in column blocks k and j are made available to processor $P_{i \bmod p_r,\ j \bmod p_c}$, where $k + 1 \le i \le N$.

We discuss three scheduling strategies below. The first one as reported in [9] is a basic approach in which computation flow is controlled by pivoting tasks $Factor(k)$. The order of execution for $Factor(k)$, $k = 1, 2, \ldots, N$ is sequential, but $Update2D()$ tasks, where most of the computation comes from, can execute in parallel among all processors. Let symbol $Update2D(k, *)$ denote tasks $Update2D(k, t)$ for $k+1 \le t \le N$. The asynchronous parallelism comes from two levels. First, a single stage of tasks $Update2D(k, *)$ can be executed concurrently on all processors. In addition, different stages of $Update2D()$ tasks from $Update2D(k, *)$ and $Update2D(k', *)$, where $k \neq k'$, can also be overlapped.

The second approach is called factor-ahead, which improves the first approach by letting $Factor(k+1)$ start as soon as $Update2D(k, k+1)$ completes. This is based on an observation that in the basic approach, after all tasks $Update2D(k, *)$ are done, all processors must wait for the result of $Factor(k + 1)$ to start $Update2D(k + 1, *)$. It is not necessary that $Factor(k + 1)$ has to wait for the completion of all tasks $Update2D(k, *)$. This idea has been used in the dense LU algorithm [17], and we extend it for asynchronous execution and incorporate a buffer space control mechanism. The details are in [10].

The factor-ahead technique still imposes a constraint that $Factor(k+1)$ must be executed after the completion of $Factor(k)$. In order to exploit potential parallelism between $Factor()$ tasks, our third design is to utilize dependence information represented by elimination forests. Since we deal with a partitioned matrix, the elimination forest defined in Definition 3.1 needs to be clustered into a supernodewise elimination forest. We call the new forest a *supernodal elimination forest*. And we call the elementwise elimination forest a *nodal elimination forest*.

DEFINITION 4.1. *A supernodal elimination forest has N nodes. Each node corresponds to a relaxed L/U supernode. Supernode $R(i_1 : i_2)$ is the parent of supernode $R(j_1 : j_2)$ if there exists vertex $i \in \{i_1, i_1+1, \ldots, i_2\}$ and vertex $j \in \{j_1, j_1+1, \ldots, j_2\}$ such that i is j's parent in the corresponding nodal elimination forest.*

A supernodal elimination forest can be generated efficiently in $O(n)$ time using Theorem 4.2 below. Figure 7 illustrates the supernodal elimination forest for Figure 5(b). The corresponding matrix is partitioned into 4×4 submatrices.

THEOREM 4.2. *Supernode $R(i_1 : i_2)$ is the parent of supernode $R(j_1 : j_2)$ in the supernodal elimination forest if and only if there exists vertex $i \in \{i_1, i_1 + 1, \ldots, i_2\}$, which is the parent of vertex j_2 in the nodal elimination forest.*

Finally, the following theorem indicates computation dependence among supern-

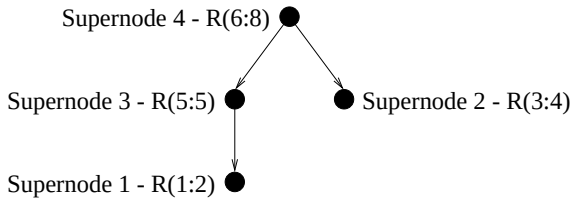


FIG. 7. *Supernodal elimination forest for the matrix in Figure* 5(b).

odes and exposes the possible parallelism that can be exploited.

THEOREM 4.3. *The L part of supernode* $R(j_1 : j_2)$ *directly or indirectly updates the L part of supernode* $R(i_1 : i_2)$ *if and only if* $R(i_1 : i_2)$ *is an ancestor of supernode* $R(j_1 : j_2)$.

Our design for LU factorization task scheduling using the above forest concept is different from the ones for Cholesky factorization [1, 26] because pivoting and row interchanges complicate the flow control in LU factorization. Using Theorem 4.3, we are able to exploit some parallelism among $Factor()$ tasks. After tasks $Factor(i)$ and $Update2D(i, k)$ have finished for every child i of supernode k, task $Factor(k)$ is ready for execution. Because of the space constraint on the buffer size, our current design does not fully exploit the parallelism, and this design is explained below.

Space complexity. We examine the degree of parallelism exploited in our algorithm by determining the maximum number of updating stages that can be overlapped. Using this information we can estimate the extra buffer space needed per processor for asynchronous execution. This buffer is used to accommodate nonzeros in $A_{k:N,k}$ and the pivoting sequence at each elimination step k. We define the *stage overlapping degree* for updating tasks as

$$\max\{|k - k'| \ \big| \ Update2D(k, *) \text{ and } Update2D(k', *) \text{ can execute concurrently.}\}$$

It is shown in [10] that for the factor-ahead approach, the reachable overlapping degree is p_c among all processors and the extra buffer space complexity is about $\frac{2.5 \cdot BSIZE}{n} \cdot S_1$, where S_1 is the sequential space size for storing the entire sparse matrix and $BSIZE$ is the maximum supernode size. This complexity is very small for a large matrix. Also because 2D cyclic mapping normally leads to a uniform data distribution, our factor-head approach is able to handle large matrices.

For the elimination forest guided approach, we enforce a constraint so that the above size of extra buffer space ($\frac{2.5 \cdot BSIZE}{n} \cdot S_1$) is also sufficient. This constraint is so that any processor that executes both $Factor(k)$ and $Factor(k')$, where $k < k'$, $Factor(k')$ cannot start until $Factor(k)$ completes. In other words, $Factor()$ tasks are executed sequentially on each single processor column, but they can be concurrent across all processor columns. As a result, our parallel algorithm is space-scalable for handling large matrices. Allocating more buffers can relax the above constraint and increase the degree of stage overlapping. However, our current experimental study does not show a substantial advantage from doing that, and we plan to investigate this issue further in the future. Figure 8 shows our elimination forest guided approach based on the above strategy.

Example. Figures 9(a) and (b) are the factor-ahead and elimination forest guided schedules for the partitioned matrix in Figure 5(b) on a 2×2 processor grid. Notice that some of the $Update2D()$ tasks, such as $U(1, 2)$, are not listed because they do not exist due to the matrix sparsity. To simplify our illustration, we assume that

```
(01) Let (my_rno, my_cno) be the 2D coordinates of this processor;
(02) Let m be the smallest column block number owned by this
       processor.
(03) if m doesn't have any child supernode then
(04)    Perform task Factor(m) for blocks this processor owns;
(05) endif
(06) for k = 1 to N − 1
(07)    Perform ScaleSwap(k) for blocks this processor owns;
(08)    Let m be the smallest column block number (m > k) this
          processor owns.
(09)    Perform Update2D(k, m) for blocks this processor owns;
(10)    if column block m is not factorized
          and all m's child supernodes have been factorized then
(11)       Perform Factor(m) for blocks this processor owns;
(12)    endif
(13)    for j = m + 1 to N
(14)       if my_cno = j mod p_c then
(15)          Perform Update2D(k, j) for blocks this processor owns;
(16)       endif
(17)    endfor
(18) endfor
```

FIG. 8. *Supernode elimination forest guided 2D approach.*

PC1	PC2
F(1)	Idle
S(1)	S(1)
U(1,3)	F(2)
F(3)	U(1,4)
S(2)	S(2)
S(3)	U(2,4)
Idle	S(3)
Idle	U(3,4)
Idle	F(4)

(a) Factor-ahead Approach

PC1	PC2
F(1)	F(2)
S(1)	S(1)
U(1,3)	U(1,4)
F(3)	S(2)
S(2)	U(2,4)
S(3)	S(3)
Idle	U(3,4)
Idle	F(4)

(b) Elimination Forest Guided Approach

FIG. 9. *Task schedules for matrix in Figure* 5(b). *F() stands for Factor(), S() stands for ScaleSwap(), U() stands for Update2D(), and PC stands for Processor Column.*

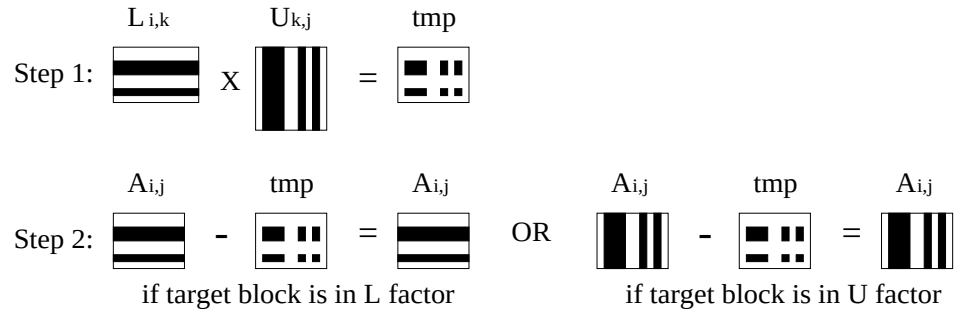


FIG. 10. *An illustration of Supernodal GEMM. Target block $A_{i,j}$ could be in the L part or U part.*

$Factor()$, $ScaleSwap()$, and $Update2D()$ each take one unit time and that communication cost is zero. In the factor-ahead schedule, $Factor(3)$ is executed immediately after $Update2D(1,3)$ on the processor column 1. The basic approach would schedule $Factor(3)$ after $ScaleSwap(2)$. Letting $Factor()$ tasks complete as early as possible is important since many updating tasks depend on $Factor()$ tasks. In the elimination forest based schedule, $Factor(2)$ is executed in parallel with $Factor(1)$ because there is no dependence between them, represented by the forest in Figure 7. As a result, the length of this schedule is one unit shorter than the factor-ahead schedule.

5. Fast supernodal GEMM kernel. We examine how the computation-dominating part of the LU algorithm can be efficiently implemented using the highest level of BLAS possible. Computations in task $Update2D()$ involve the following supernode block multiplication: $A_{i,j} = A_{i,j} - L_{i,k} * U_{k,j}$, where $k < i$ and $k < j$. As we mentioned in the end of section 3.2, submatrices like $A_{i,j}$, $L_{i,k}$, and $U_{k,j}$ are all stored in a compressed storage scheme with bit maps that identify their dense subcolumns or subrows. As a result, the BLAS-3 GEMM routine [7] may not be directly applicable to $A_{i,j} = A_{i,j} - L_{i,k} * U_{k,j}$ because subcolumns or subrows in those submatrices may not be consecutive and the target block $A_{i,j}$ may have a nonzero structure different from that of product $L_{i,k} * U_{k,j}$.

There could be several approaches to circumvent the above problem. One approach is to use a mixture of BLAS-1/2/3 routines. If $L_{i,k}$ and $A_{i,j}$ have the same row sparse structure, and $U_{k,j}$ and $A_{i,j}$ have the same column sparse structure, BLAS-3 GEMM can be directly used to modify $A_{i,j}$. If only one of the above two conditions holds, then the BLAS-2 routine GEMV can be employed. Otherwise only the BLAS-1 routine DOT can be used. In the worst case, the performance of this approach is close to the BLAS-1 performance. Another approach is to treat nonzero submatrices of A as dense during space allocation and submatrix computation, and hence BLAS-3 GEMM can be employed more often. But considering the average density of submatrices is only around 51% for our test matrices, this approach normally leads to an excessive amount of extra space and unnecessary arithmetic operations.

We propose the following approach called *Supernodal GEMM* to minimize unnecessary computation but retain high efficiency. The basic idea is described as follows. If the BLAS-3 GEMM is not directly applicable, we divide the operation into two steps. At the first step, we ignore the target nonzero structure of $A_{i,j}$ and directly use BLAS-3 GEMM to compute $L_{i,k} * U_{k,j}$. The result is stored in a temporary block. At the second step, we merge this temporary block with $A_{i,j}$ using subtraction. Figure 10 illustrates these two steps. Since the computation of the first step is

TABLE 1
Test matrices and their statistics.

Matrix	Order	$\|A\|$	Factor entries / $\|A\|$			Application domain
			Dynamic	Static	A^TA	
sherman5	3312	20793	12.03	15.70	20.42	Oil reservoir modeling
lnsp3937	3937	25407	17.87	27.33	36.76	Fluid flow modeling
lns3937	3937	25407	18.07	27.92	37.21	Fluid flow modeling
sherman3	5005	20033	22.13	31.20	39.24	Oil reservoir modeling
jpwh991	991	6027	23.55	34.02	42.57	Circuit physics
orsreg1	2205	14133	29.34	41.44	52.19	Oil reservoir simulation
saylr4	3564	22316	30.01	44.19	56.40	Oil reservoir modeling
goodwin	7320	324772	9.63	10.80	16.00	Fluid mechanics
e40r0100	17281	553562	14.76	17.32	26.48	Fluid dynamics
raefsky4	19779	1316789	20.36	28.06	35.68	Container modeling
inaccura	16146	1015156	9.79	12.21	16.47	Structure problem
af23560	23560	460598	30.39	44.39	57.40	Navier–Stokes solver
fidap011	16614	1091362	23.36	24.55	31.21	Finite element modeling
vavasis3	41092	1683902	29.21	32.03	38.75	PDE

more expensive than the second step, our code for multiplying supernodal submatrices can achieve performance comparable to BLAS-3 GEMM. A further optimization is to speed up the second step since the result merging starts to play some role for the total time after the GEMM routine reduces the cost of the first step. Our strategy is that if the result block and $A_{i,j}$ have the same row sparse structure or the same column sparse structure, the BLAS-1 AXPY routine should be used to avoid scalar operations. And to increase the probability of structure consistency between the temporary result block and $A_{i,j}$, we treat some of the L and U submatrices as dense during the space allocation stage if the percentage of nonzeros in such a submatrix compared to the entire block size exceeds a threshold. For Cray-T3E, our experiments show that threshold 85% is the best to reduce the result merging time with small space increase.

6. Experimental studies on Cray T3E. S^+ has been implemented on Cray T3E using its SHMEM communication library. Most of our experiments were conducted on a T3E machine at San Diego Supercomputing Center (SDSC). Each Cray-T3E processing element at SDSC has a clock rate of 300MHz, an 8KB internal cache, 96KB second level cache, and 128MB memory. The peak bandwidth between nodes is reported as 500MB/s, and the peak round trip communication latency is about 0.5-2μs [28]. We have observed that when the block size is 25, double-precision GEMM achieves 388 MFLOPS while double-precision GEMV reaches 255 MFLOPS. We have used a block size 25 in our experiments. We also obtained access to a Cray-T3E at the NERSC division of the Lawrence Berkeley Lab. Each node in this machine has a clock rate of 450MHz and 256MB memory. We have done one set of experiments to show the performance improvement on this faster machine.

In this section, we report the overall sequential and parallel performance of S^+ without incorporating space optimization techniques, and we measure the effectiveness of the optimization strategies proposed in sections 3 and 4. In the next section, we will study the memory requirement of S^+ with and without space optimization. Table 1 shows the statistics of the test matrices used in this section. Column 2 is the orders of the matrices, and column 3 is the number of nonzeros before symbolic factorization. In columns 4, 5, and 6 of this table, we have also listed the total number of nonzero entries divided by $|A|$ using three methods. Those nonzero entries including fill-ins are produced by dynamic factorization, static symbolic factorization,

TABLE 2
Sequential performance on a 300MHz Cray T3E node. The symbol"-" implies that the data is not available due to insufficient memory or is not meaningful due to paging.

Matrix	Sequential S^+		SuperLU		Sequential S^*		Time ratio	
	Time	MFLOPS	Time	MFLOPS	Time	MFLOPS	$\frac{S^+}{SuperLU}$	$\frac{S^+}{S^*}$
sherman5	0.65 (0.04)	38.6	0.78	32.2	0.94	26.7	0.83	0.69
lnsp3937	1.48 (0.08)	22.9	1.73	19.5	2.00	16.9	0.86	0.74
lns3937	1.58 (0.09)	24.2	1.84	20.8	2.19	17.5	0.86	0.72
sherman3	1.56 (0.03)	36.2	1.68	33.6	2.03	27.8	0.93	0.77
jpwh991	0.52 (0.03)	31.8	0.56	29.5	0.69	23.9	0.93	0.75
orsreg1	1.60 (0.04)	35.0	1.53	36.6	2.04	27.4	1.05	0.78
saylr4	2.67 (0.07)	37.2	2.69	36.9	3.53	28.1	0.99	0.76
goodwin	10.26 (0.35)	65.2	-	-	17.0	39.3	-	0.60

or Cholesky factorization of A^TA. The result shows that for these tested matrices, the total number of nonzeros predicted by static factorization is within 40% of what dynamic factorization produces. But the A^TA approach overestimates substantially more nonzeros, which indicates that the elimination tree of A^TA can introduce too many false dependency edges. All matrices are ordered using the minimum degree algorithm[3] on A^TA and the permutation algorithm for zero-free diagonal [8].

In calculating the MFLOPS achieved by our parallel algorithm, *we do not include extra floating point operations introduced by static fill-in overestimation and supernode amalgamation.* The achieved MFLOPS are computed by using the following formula:

$$\text{Achieved MFLOPS} = \frac{\text{True operation count}}{\text{Elapsed time of our algorithm on the T3E}}.$$

The true operation count is obtained by running SuperLU without amalgamation. Amalgamation can be turned off in SuperLU by setting the relaxation parameter for amalgamation to 1 [6, 24].

6.1. Overall code performance. Table 2 lists the sequential performance of S^+, our previous design S^*, and SuperLU.[4] The result shows S^+ can actually be faster than SuperLU because of the use of new supernode partitioning and matrix multiplication strategies. The test matrices are selected from Table 1 that can be executed on a single T3E node. The performance improvement ratios from S^* to S^+ vary from 22% to 40%. Notice that time measurement in Table 2 excludes symbolic preprocessing time. However, symbolic factorization in our algorithms is very fast and takes only about 4.2% of numerical factorization time for the matrices in Table 2. And this ratio tends to decrease as the matrix size increases. This preprocessing cost is insignificant, especially when LU factorization is used in an iterative algorithm. In Table 2, we list the time of symbolic factorization for each matrix inside the parentheses behind the time of S^+.

For parallel performance, we compare S^+ with a previous version [10] in Table 3, and the improvement ratio in terms of MFLOPS varies from 16% to 116%, in average more than 50%. Table 4 shows the absolute performance of S^+ on the Cray T3E machine with 450MHz CPU. The highest performance reached is 10.85 GFLOPS, while for the same matrix, 8.25 GFLOPS is reached on the T3E with 300MHz processors.

[3] A MATLAB program is used for minimum degree ordering.

[4] We did not compare with another well-optimized package UMFPACK [2] because SuperLU has been shown to be competitive to UMFPACK [4].

TABLE 3
MFLOPS performance of S^+ and S^* on the 300MHz Cray T3E.

Matrix	P=8		P=16		P=32		P=64		P=128	
	S^*	S^+	S^*	S^+	S^*	S^+	S^*	S^+	S^*	S^+
goodwin	215.2	403.5	344.6	603.4	496.3	736.0	599.2	797.3	715.2	826.8
e40r0100	205.1	443.2	342.9	727.8	515.8	992.8	748.0	1204.8	930.8	1272.8
raefsky4	391.2	568.2	718.9	1072.5	1290.7	1930.3	2233.3	3398.1	3592.9	5133.6
inaccura	272.2	495.5	462.0	803.8	726.0	1203.6	1172.7	1627.6	1524.5	1921.7
af23560	285.4	432.1	492.9	753.2	784.3	1161.3	1123.2	1518.9	1512.7	1844.7
fidap011	489.3	811.2	878.1	1522.8	1524.3	2625.0	2504.4	4247.6	3828.5	6248.4
vavasis3	795.5	937.3	1485.5	1823.7	2593.5	3230.8	4406.3	5516.2	6726.6	8256.0

TABLE 4
Experimental results of S^+ on the 450MHz Cray T3E. All times are in seconds.

Matrix	P=8		P=16		P=32		P=64		P=128	
	Time	MFLOPS	Time	MFLOPS	Time	MFLOPS	Time	MFLOPS	Time	MFLOPS
goodwin	1.21	552.6	0.82	815.4	0.69	969.0	0.68	983.2	0.67	997.9
e40r0100	4.06	609.4	2.50	989.7	1.87	1323.2	1.65	1499.6	1.59	1556.2
raefsky4	38.62	814.6	20.61	1526.3	11.54	2726.0	6.80	4626.2	4.55	6913.8
inaccura	6.56	697.2	4.12	1110.1	2.80	1633.4	2.23	2050.9	1.91	2394.6
af23560	10.57	602.1	6.17	1031.5	4.06	1567.5	3.47	1834.0	2.80	2272.9
fidap011	21.58	1149.5	11.71	2118.4	6.81	3642.7	4.42	5612.3	3.04	8159.9
vavasis3	62.68	1398.8	33.68	2603.2	19.26	4552.3	11.75	7461.9	8.08	10851.1

6.2. Effectiveness of the proposed optimization strategies. Elimination forest guided partitioning and amalgamation. Our new strategy for supernode partitioning with amalgamation simultaneously clusters columns and rows using structural containment information implied by an elimination forest. Our previous design S^* [10, 11] does not consider the bounding of nonzeros in the U part. We compare our new code S^+ with a modified version using the previous partitioning strategy. The performance improvement ratio by using the new strategy is listed in Figure 11, and an average of 20% improvement is obtained. The ratio for matrix "af23560" is not substantial because this matrix is very sparse and the partitioning/amalgamation strategy cannot produce large supernodes.

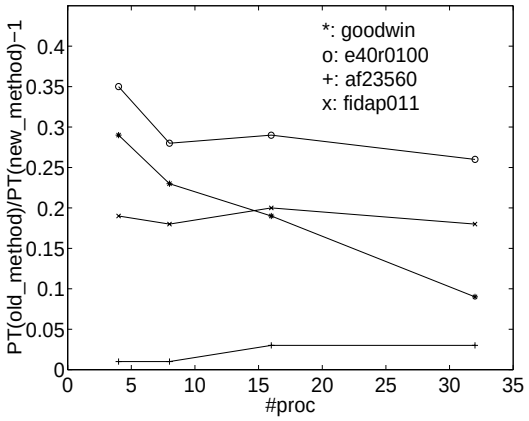


FIG. 11. *Performance improvement by using the new supernode partitioning/amalgamation strategy.*

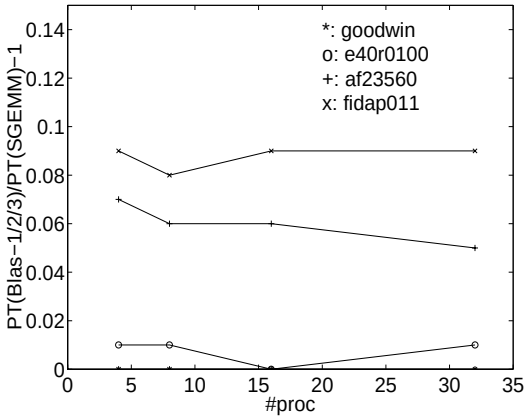


FIG. 12. *Performance improvement by using the supernodal GEMM.*

TABLE 5
Performance improvement by using the elimination forest guided approach.

Matrix	Improvement over Basic				Improvement over Factor-ahead			
	P=16	P=32	P=64	P=128	P=16	P=32	P=64	P=128
goodwin	41%	35%	19%	21%	8%	12%	10%	14%
e40r0100	38%	40%	30%	27%	15%	17%	12%	15%
raefsky4	21%	21%	34%	34%	7%	10%	11%	13%
inaccura	21%	28%	26%	27%	7%	13%	9%	13%
af23560	31%	37%	32%	30%	10%	15%	10%	13%
fidap011	24%	28%	36%	38%	8%	12%	11%	15%
vavasis3	17%	16%	31%	28%	3%	6%	8%	12%

Effectiveness of supernodal GEMM. We assess the gain due to the introduction of our supernodal GEMM operation. We compare S^+ with a modified version using an approach that mixes BLAS-1/2/3 as described in section 5. We do not compare our approach with the approach that treats all nonzero blocks as dense since it introduces too much extra space and computation. The performance improvement ratio of our supernodal approach over the mixed approach is listed in Figure 12. The improvement is not substantial for matrix "e40r0100" and is none for "goodwin". This is because they are relatively dense and the mixed approach has been employing BLAS-3 GEMM most of the time. For the other two matrices that are relatively sparse, the improvement ratio can be up to 10%.

A comparison of the control strategies for exploiting 2D parallelism. In Table 5 we assess the performance improvement by using the elimination forest guided approach against the factor-ahead and basic approaches described in section 4. Compared to the basic approach, the improvement ratios vary from 16% to 41% and the average is 28%. Compared to the factor-ahead approach, the average improvement ratio is 11% and the ratios tend to increase when the number of processors increases. This result is expected in the sense that the factor-ahead approach improves the degree of computation overlapping by scheduling factor tasks one step ahead, while using elimination forests can exploit more parallelism.

7. Space optimization. For all matrices tested above, static symbolic factorization provides fairly accurate prediction of nonzero patterns and only creates 10% to 50% more fill-ins compared to dynamic symbolic factorization used in SuperLU.

TABLE 6
Circuit and device simulation test matrices and their statistics.

Matrix	Order	$\|A\|$	Factor entries/$\|A\|$		
			Dynamic	Static	A^TA
TIa	3432	25220	24.45	42.49	307.1
TId	6136	53329	27.53	61.41	614.2
TIb	18510	145149	91.84	278.34	1270.7
memplus	17758	99147	71.26	168.77	215.19
wang3	26064	177168	197.30	298.12	372.71

However, for some matrices, especially those in circuit and device simulation, static symbolic factorization creates too many fill-ins. Table 6 shows characteristics of five matrices from circuit and device simulations. Static symbolic factorization does produce a large number of fill-ins for these matrices (up to 3 times higher than dynamic symbolic factorization using the same matrix ordering[5]). Our solution needs to provide a smooth adaptation in handling such cases.

For the above cases, we find that a significant number of predicted fill-ins remain zero throughout numerical factorization. This indicates that space allocated to those fill-ins is unnecessary. Thus our first space-saving strategy is to delay the space allocation decision and acquire memory only when a submatrix block becomes truly nonzero during numerical computation. Such a dynamic space allocation strategy can lead to a relatively small space requirement even if static symbolic factorization excessively over-predicts fill-ins. Another strategy is to examine if space recycling for some nonzero submatrices is possible since a nonzero submatrix may become zero during numerical factorization due to pivoting. This has a potential to save significantly more space since the early identification of zero blocks prevents their propagation in the update phase of numerical factorization.

Space requirements. We have conducted experiments [20] to study memory requirement by incorporating the above space optimization strategies into S^+ on a SUN Ultra-1 with 320MB memory. In the following study, we refer to the revised S^+ with space optimization as $SpaceS^+$. Table 7 lists the space requirement of S^+, SuperLU, and $SpaceS^+$ for the matrices from Tables 1 and 6. Matrix vavasis3 is not listed because its space requirement is too high for all three algorithms on this machine.

The result in Table 7 shows that our space optimization strategies are effective. $SpaceS^+$ uses slightly less space compared to S^+ for matrices in Table 1 and 37% less space on average for matrices in Table 6 (68% less space for matrix TIb). Compared to SuperLU, our algorithm actually uses 3.9% less space on average while static symbolic factorization predicts 38% more nonzeros. That is because the U structure in SuperLU is less regular than that in S^+ and the indexing scheme in S^+ is simpler. Notice that the space cost in our evaluation includes symbolic factorization. This part of the cost ranges from 1% to 7% of the total cost. We also list the ratio of $SpaceS^+$ processing time to S^+ and to SuperLU. Some entries are marked "-" instead of actual numbers because we observed paging on these matrices that may affect the accuracy of the result. In terms of average time cost, the new version is faster than SuperLU, which is consistent with the results in section 6.1. It is also slightly faster than S^+ because

[5]Using a different matrix ordering (MMD on A^T+A), SuperLU generates fewer fill-ins on certain matrices. This paper focuses algorithm design when ordering is given and studies performance using one ordering method. An interesting future research topic is to study ordering methods that optimize static factorization.

TABLE 7
Space requirement in MB on a SUN Ultra-1 machine. The symbol "-" implies that the data is not available due to insufficient memory or paging which affects the measurement.

Matrix	Space requirement			Space ratio		Time ratio	
	S^+	SuperLU	$SpaceS^+$	$\frac{SpaceS^+}{SuperLU}$	$\frac{SpaceS^+}{S^+}$	$\frac{SpaceS^+}{SuperLU}$	$\frac{SpaceS^+}{S^+}$
sherman5	3.061	3.305	2.964	0.90	0.97	0.853	0.959
sherman3	5.713	5.412	5.536	1.02	0.97	1.023	0.944
orsreg1	5.077	4.555	4.730	1.04	0.93	0.920	0.823
saylr4	8.509	7.386	8.014	1.09	0.94	0.964	0.870
goodwin	29.192	35.555	28.995	0.82	0.99	0.657	0.993
e40r0100	79.086	93.214	78.568	0.84	0.99	-	-
raefsky4	303.617	272.947	285.920	1.05	0.94	0.707	0.921
af23560	170.166	147.307	162.839	1.11	0.96	0.869	0.984
fidap011	221.074	271.423	219.208	0.81	0.99	-	-
TIa	8.541	6.265	7.321	1.17	0.86	0.675	0.629
TId	29.647	18.741	19.655	1.05	0.66	0.366	0.366
memplus	138.218	75.194	68.467	0.91	0.50	-	-
TIb	341.418	221.285	107.711	0.49	0.32	-	-
wang3	430.817	-	347.505	-	0.81	-	-

TABLE 8
MFLOPS performance of $SpaceS^+$ on 450MHz Cray T3E.

Matrix	vavasis3	TIa	TId	TIb	memplus	wang3
MFLOPS on 128 nodes	10004.0	739.9	1001.9	2515.7	6548.4	6261.0
MFLOPS on 8 nodes	1492.9	339.6	281.5	555.7	1439.4	757.8

TABLE 9
Performance difference of S^+ and $SpaceS^+$ on 300MHz Cray T3E. A positive number indicates an improvement of $SpaceS^+$ over the original S^+, while a negative number indicates a slowdown.

Matrix	P=8	P=16	P=32	P=64	P=128
goodwin	-7.28%	-8.29%	-8.10%	-1.17%	-4.69%
e40r0100	-6.81%	-8.81%	-11.34%	-8.84%	-7.13%
raefsky4	3.41%	2.52%	-0.42%	-1.82%	-5.02%
af23560	-3.17%	-3.98%	-9.72%	-4.56%	-13.76%
vavasis3	7.65%	-1.79%	5.02%	-2.16%	-6.13%
TIa	13.16%	10.42%	2.63%	-2.94%	-6.06%
TId	35.15%	23.81%	9.28%	-2.50%	-9.59%
TIb	352.20%	270.26%	209.27%	133.69%	78.10%
memplus	136.43%	115.38%	87.09%	61.84%	35.24%
wang3	10.51%	7.57%	3.52%	1.53%	-5.17%

the early elimination of zero blocks prevents their propagation and hence reduces unnecessary computation.

Parallel performance. Our experiments on Cray T3E show that the parallel time performance of $SpaceS^+$ is still competitive to S^+. Table 8 lists performance of $SpaceS^+$ on vavasis3 and circuit simulation matrices in 450MHz T3E nodes. $SpaceS^+$ can still achieve 10.00 GFLOPS on matrix vavasis3, which is not much less than the highest 10.85 GFLOPS achieved by S^+ on 128 450MHz T3E nodes. For circuit simulation matrices, $SpaceS^+$ delivers reasonable performance.

Table 9 is the time difference of S^+ with and without space optimization on 300MHz T3E nodes. For the matrices with high fill-in overestimation ratios, we observe that S^+ with dynamic space management is better than S^+. It is about 109% faster on 8 processors and 18% faster on 128 processors. As for other matrices, on 8 processors $SpaceS^+$ is about 1.24% slower than S^+, while on 128 processors, it is

7% slower than S^+. On average, $SpaceS^+$ tends to become slower when the number of processors becomes larger. This is because the lazy space allocation scheme introduces new overhead for dynamic memory management and for row and column broadcasts (blocks of the same L-column or U-row, now allocated in noncontiguous memory, can no longer be broadcasted as a unit). This new overhead affects critical paths, which dominate performance when parallelism is limited and the number of processors is large.

8. Concluding remarks. Our experiments show that the proper use of elimination forests allows for effective matrix partitioning and parallelism exploitation. Together with the supernodal matrix multiplication algorithm, our new design can improve the previous code substantially and set a new performance record. Our experiments also show that S^+ with dynamic space optimization can deliver high performance for large sparse matrices with reasonable memory cost. Static symbolic factorization may create too many fill-ins for some matrices, but our space optimization techniques can effectively reduce memory requirements. Our comparison with SuperLU indicates that the sequential version of S^+ is highly optimized and can be faster than SuperLU. Our evaluation has focused on using a simple, but popular, ordering strategy. Different matrix reordering methods can result in different numbers of fill-ins. More investigation is needed to address this issue in order to reduce overestimation ratios.

Performance of S^+ is sensitive to the underlying message-passing library performance. Our experiments use the SHMEM communication library on Cray T3E, and recently we have implemented S^+ using MPI 1.1. The MPI based S^+ version is more portable; however, the current version is about 30% slower than the SHMEM-based version. This is because SHMEM uses direct remote memory access, while MPI requires hand-shake between communication peers, which involves synchronization overhead. We expect that more careful optimization on this MPI version can lead to better performance, and use of one-side communication available in the future MPI-2 release may also help boosting performance. The source code of this MPI-based S^+ version is available at http://www.cs.ucsb.edu/research/S+, and the HPC group in SUN Microsystems plans to include it in their next release of the S3L library used for SUN SMPs and clusters [22].

Acknowledgments. We would like to thank Bin Jiang and Steven Richman for their contribution in implementing S^+, Horst Simon for providing access to a Cray T3E at the National Energy Research Scientific Computing Center, Stefan Boeriu for supporting access to a Cray T3E at San Diego Supercomputing Center, Andrew Sherman and Vinod Gupta for providing circuit simulation matrices, Tim Davis, Apostolos Gerasoulis, Xiaoye Li, Esmond Ng, and Horst Simon for their help during our research, and the anonymous referees for their valuable comments.

Appendix A. Notations.

A	The sparse matrix to be factorized. Notice that elements of A change during factorization. In this paper proposed optimizations are applied to A after symbolic factorization.
$a_{i,j}$	The element in A with row index i and column index j.
$a_{i:j,s:t}$	The submatrix in A from row i to row j and from column s to t.
l_k	Column k in the low triangular part of A.
u_k	Row k in the upper triangular part of A.
$\hat{a}_{i,j}$	$\hat{a}_{i,j} \neq 0$ if and only if $a_{i,j}$ is nonzero after symbolic factorization.

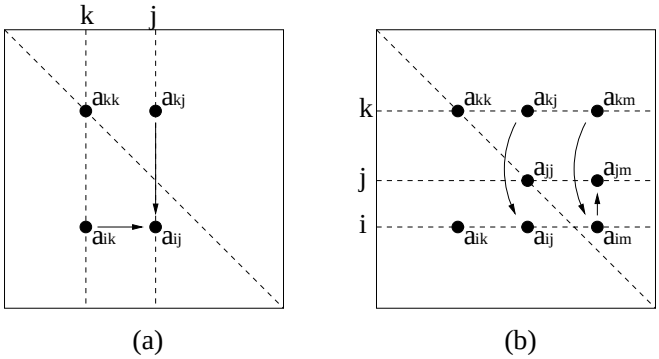


FIG. 13. *An illustration for the proof of Theorem* 3.2.

$\hat{l}_k$	The index set of nonzero elements in l_k after symbolic factorization.
$\hat{u}_k$	The index set of nonzero elements in u_k after symbolic factorization.
$\lvert\hat{l}_k\rvert$	The cardinality of $\hat{l}_k$.
$\lvert\hat{u}_k\rvert$	The cardinality of $\hat{u}_k$.
$A_{i,j}$	The submatrix in the partitioned A with row block index i and column block index j.
$A_{i:j,s:t}$	The submatrices in the partitioned A from row block i to j and from column block s to t.
$L_{i,j}$	The submatrix with block index i and j in the lower triangular part.
$U_{i,j}$	The submatrix with block index i and j in the upper triangular part.
$R(i:j)$	Relaxed L/U supernode, which contains a diagonal block, an L supernode and a U supernode.

Appendix B. Proof of theorems.

B.1. Theorem 1. *Proof.* To prove the theorem holds when vertex j is an *ancestor* of vertex k, we need only to show that it holds if vertex j is the *parent* of vertex k, because of the transitivity of "$\subseteq$".

If vertex j is the parent of vertex k in this elimination forest, $\hat{a}_{k,j} \neq 0$. Let $a^t_{i,j}$ denote the symbolic value of $a_{i,j}$ after step t of symbolic factorization. Since $a_{k,j}$ is not changed after step k of symbolic factorization, $a^k_{k,j} \neq 0$.

We first examine the L part as illustrated in Figure 13(a). For any $i > k$ and $i \in \hat{l}_k$, i.e., $\hat{a}_{i,k} \neq 0$, we have $a^k_{i,k} \neq 0$. Because $a^k_{i,k}$ and $a^k_{k,j}$ are used to update $a^k_{i,j}$, it holds that $i \in \hat{l}_j$. Therefore, $\{r \mid r{\in}\hat{l}_k \wedge j{\leq}r{\leq}n\} \subseteq \hat{l}_j$.

Next we examine the U part as illustrated in Figure 13(b). Since l_k must contain at least one nonzero off-diagonal element before step k of symbolic factorization, we assume it is $a^{k-1}_{i,k}$. Because $a_{k,j}$ is the first off-diagonal nonzero in $\hat{u}_k$, and $\hat{a}_{k,i} \neq 0$, we know $i \geq j$. For any $m > j$ and $m \in \hat{u}_k$, we prove $m \in \hat{u}_j$ as follows. Since $\hat{a}_{k,m} \neq 0$ and $a^k_{i,k} \neq 0$, it follows that $a^k_{i,j} \neq 0$ and $a^k_{i,m} \neq 0$. Therefore, $a^j_{i,j} \neq 0$. As a result, $a^k_{i,m} \neq 0$ and $a^j_{j,m} \neq 0$. And we conclude that $\{c \mid c{\in}\hat{u}_k \wedge j{\leq}c{\leq}n\} \subseteq \hat{u}_j$. ∎

B.2. Theorem 2. *Proof.* If l_k directly updates l_j in LU factorization, vertex k must have a parent in the forest. Let

$$T = \{t \mid t \leq j \text{ and } t \text{ is an ancestor of } k \text{ in the elimination forest}\}.$$

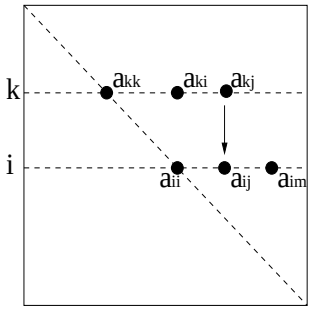


FIG. 14. *An illustration for the proof of Theorem* 3.4.

Since k's parent $\leq j$, set T is not empty. Let i be the largest element in T. We show $i = j$ by contradiction as illustrated in Figure 14. Assume $i < j$. Following Theorem 3.2, $\{c \mid c \in \hat{u}_k \wedge i \leq c \leq n\} \subseteq \hat{u}_i$. Since $\hat{a}_{k,j} \neq 0$, we know $\hat{a}_{i,j} \neq 0$. Let m be i's parent. Since i is the largest element in T and $m > i$, we know $m \notin T$. Thus, it holds that $m > j$. However, $a_{i,m}$ should be the first off-diagonal nonzero in $\hat{u}_i$. This is a contradiction since $\hat{a}_{i,j} \neq 0$. Thus vertex j is an ancestor of vertex k in the elimination forest.

If l_k indirectly updates l_j, there must be a sequence $s_1, s_2, \ldots, s_p$ such that $s_1 = k$, $s_p = j$, and l_{s_q} directly updates $l_{s_{q+1}}$ for each $1 \leq q \leq p-1$. That is, vertex s_{q+1} is an ancestor of vertex s_q for each $1 \leq q \leq p-1$. Thus, we conclude that vertex j is an ancestor of vertex k.

Conversely, if vertex j is an ancestor of vertex k in the elimination forest, there must be a sequence $s_1, s_2, \ldots, s_p$ such that $s_1 = k$, $s_p = j$, and vertex s_{q+1} is the parent of vertex s_q for each q, where $1 \leq q \leq p-1$. Then for each $1 \leq q \leq p-1$, l_{s_q} directly updates $l_{s_{q+1}}$ since $|\hat{l}_{s_q}| \neq 1$ and $\hat{a}_{s_q, s_{q+1}} \neq 0$. Thus, we conclude that l_k directly or indirectly updates l_j during numerical factorization. ∎

B.3. Theorem 3. *Proof.* The "if" part is an immediate result of Definition 4.1. Now we prove the "only if" part. If $R(i_1 : i_2)$ is the parent of $R(j_1 : j_2)$ in the supernodal elimination forest, there exists vertex $i \in \{i_1, i_1+1, \ldots, i_2\}$ and vertex $j \in \{j_1, j_1+1, \ldots, j_2\}$ such that i is j's parent in the corresponding nodal elimination forest. Below we prove by contradiction that such a vertex j is unique and it must be j_2.

Suppose j is not j_2, i.e., $j_1 \leq j < j_2$. Since the diagonal block of $R(j_1 : j_2)$ is considered to be dense (including symbolic fill-ins after amalgamation), for every $u \in \{j_1, j_1+1, \ldots, j_2-1\}$, u's parent is $u+1$ in the nodal elimination forest. Thus j's parent should be one in $\{j_1+1, \ldots, j_2\}$; however, we also know that j's parent is i in the nodal elimination forest and $j_2 < i$. That is a contradiction. ∎

B.4. Theorem 4. *Proof.* If the L part of supernode $R(j_1 : j_2)$ directly or indirectly updates L supernode $R(i_1 : i_2)$, there exists an l_j ($j \in \{j_1, j_1+1, \ldots, j_2\}$) that directly or indirectly updates column l_i ($i \in \{i_1, i_1+1, \ldots, i_2\}$). Because of Theorem 3.4, i is an ancestor of j. According to Definition 4.1, $R(i_1 : i_2)$ is an ancestor of supernode $R(j_1 : j_2)$.

On the other hand, if $R(i_1 : i_2)$ is an ancestor of supernode $R(j_1 : j_2)$, for each child/parent pair in the path from $R(j_1 : j_2)$ to $R(i_1 : i_2)$, we can apply both Theorem 4.2 and Theorem 3.4. Then, it is easy to show that the L part of each child supernode in this path directly or indirectly updates the L part of its parent

supernode. Thus L part of supernode $R(j_1 : j_2)$ directly or indirectly updates L part of supernode $R(i_1 : i_2)$. □

REFERENCES

[1] C. Ashcraft, R. Grimes, J. Lewis, B. Peyton, and H. Simon, *Progress in sparse matrix methods for large sparse linear systems on vector supercomputers*, Int. J. Supercomput. Appl., 1 (1987), pp. 10–30.

[2] T. A. Davis and I. S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM Matrix Anal. Appl., 18 (1997), pp. 140–158.

[3] J. Demmel, *Numerical linear algebra on parallel processors*, in Lecture Notes for NSF-CBMS Regional Conference in the Mathematical Sciences, San Francisco, CA, 1995.

[4] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[5] J. Demmel, J. Gilbert, and X. S. Li, *An Asynchronous Parallel Supernodal Algorithm for Sparse Gaussian Elimination*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 915–952.

[6] J. W. Demmel, J. R. Gilbert, and X. S. Li, *SuperLU Users' Guide*, 1997.

[7] J. Dongarra, J. D. Croz, S. Hammarling, and R. Hanson, *An extended set of basic linear algebra subroutines*, ACM Trans. Math. Software, 14 (1988), pp. 18–32.

[8] I. S. Duff, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Software, 7 (1981), pp. 315–330.

[9] C. Fu, X. Jiao, and T. Yang, *A comparison of* 1*-D and* 2*-D data mapping for sparse LU factorization on distributed memory machines*, in Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, 1997, CD-ROM, SIAM, Philadelphia, PA, 1997.

[10] C. Fu, X. Jiao, and T. Yang, *Efficient sparse LU factorization with partial pivoting on distributed memory architectures*, IEEE Trans. Parallel Distrib. Systems, 9 (1998), pp. 109–125.

[11] C. Fu and T. Yang, *Sparse LU factorization with partial pivoting on distributed memory machines*, in Proceedings of the ACM/IEEE Supercomputing, Pittsburgh, PA, 1996.

[12] C. Fu and T. Yang, *Space and time efficient execution of parallel irregular computations*, in Proceedings of the ACM Symposium on Principles & Practice of Parallel Programming, Las Vegas, NV, 1997, pp. 57–68.

[13] K. Gallivan, B. Marsolf, and H. Wijshoff, *The parallel solution of nonsymmetric sparse linear systems using H* reordering and an associated factorization*, in Proceedings of the ACM International Conference on Supercomputing, Manchester, NH, 1994, pp. 419–430.

[14] A. George and E. Ng, *Symbolic factorization for sparse Gaussian elimination with partial pivoting*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 877–898.

[15] A. George and E. Ng, *Parallel sparse Gaussian elimination with partial pivoting*, Ann. Oper. Res., 22 (1990), pp. 219–240.

[16] J. R. Gilbert and E. Ng, *Predicting structure in nonsymmetric sparse matrix factorizations*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, 1993, pp. 107–139.

[17] G. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing Compilers*, Academic Press, Boston, MA, 1993.

[18] A. Gupta, G. Karypis, and V. Kumar, *Highly scalable parallel algorithms for sparse matrix factorization*, IEEE Trans. Parallel Distrib. Systems, 8 (1995).

[19] S. Hadfield and T. Davis, *A Parallel Unsymmetric-pattern Multifrontal Method*, Tech. Report TR-94-028, Computer and Information Sciences Department, University of Florida, Gainesville, FL, 1994.

[20] B. Jiang, S. Richman, K. Shen, and T. Yang, *Efficient sparse LU factorization with lazy space allocation*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, Texas, 1999, CD-ROM, SIAM, Philadelphia, PA, 1997.

[21] X. Jiao, *Parallel Sparse Gaussian Elimination with Partial Pivoting and* 2*-D Data Mapping*, Master's thesis, Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA, 1997.

[22] G. Kechriotis, *private communication*, 1999.

[23] X. S. Li, *Sparse Gaussian Elimination on High Performance Computers*, Ph.D. thesis, Computer Science Division, EECS, University of California at Berkeley, Berkeley, CA, 1996.

[24] X. S. Li, *private communication*, 1998.

[25] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[26] E. Rothberg, *Exploiting the Memory Hierarchy in Sequential and Parallel Sparse Cholesky Factorization*, Ph.D. thesis, Department of Computer Science, Stanford University, CA, 1992.

[27] E. Rothberg and R. Schreiber, *Improved load distribution in parallel sparse Cholesky factorization*, in Proceedings of Supercomputing '94, Washington, D.C., 1994, pp. 783–792.

[28] S. L. Scott and G. M. Thorson, *The Cray T3E network: Adaptive routing in a high performance 3D Torus*, in Proceedings of HOT Interconnects IV, Stanford University, Stanford, CA, 1996.

[29] K. Shen, X. Jiao, and T. Yang, *Elimination forest guided 2D sparse LU factorization*, in Proceedings of the 10th ACM Symposium on Parallel Algorithms and Architectures, Puerto Vallarta, Mexico, 1998, pp. 5–15; also available online from http://www.cs.ucsb.edu/research/S+/.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 1, pp. 306–321

WHICH EIGENVALUES ARE FOUND BY THE LANCZOS METHOD?*

A. B. J. KUIJLAARS†

Abstract. When discussing the convergence properties of the Lanczos iteration method for the real symmetric eigenvalue problem, Trefethen and Bau noted that the Lanczos method tends to find eigenvalues in regions that have too little charge when compared to an equilibrium distribution. In this paper a quantitative version of this rule of thumb is presented. We describe, in an asymptotic sense, the region containing those eigenvalues that are well approximated by the Ritz values. The region depends on the distribution of eigenvalues and on the ratio between the size of the matrix and the number of iterations, and it is characterized by an extremal problem in potential theory which was first considered by Rakhmanov. We give examples showing the connection with the equilibrium distribution.

Key words. Ritz values, equilibrium distribution, potential theory

AMS subject classifications. 65F15, 31A15

PII. S089547989935527X

1. Introduction.

The Lanczos iteration is a popular method to compute eigenvalues of large real symmetric matrices. For a given real symmetric matrix A of size $N \times N$, the Lanczos method starts from a nonzero vector $b \in \mathbb{R}^N$ and generates two sequences of numbers (α_k) and (β_k) as follows. Put $\beta_0 = 0$, $v_0 = 0$, $v_1 = b/\|b\|_2$, and for $k = 1, 2, \ldots,$

$$\alpha_k = \langle v_k, Av_k \rangle, \qquad \beta_k v_{k+1} = Av_k - \alpha_k v_k - \beta_{k-1} v_{k-1},$$

where β_k is taken such that $\|v_{k+1}\|_2 = 1$. The vectors $v_1, v_2, \ldots, v_n$ are an orthonormal basis of the nth Krylov subspace spanned by $b, Ab, \ldots, A^{n-1}b$. The coefficients α_k and β_k are collected in the tridiagonal matrices

$$T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

for $n \leq N$. The eigenvalues of T_n are called Ritz values, and they are easier to compute because of the tridiagonal nature of T_n and because n is smaller than N. Some of the Ritz values turn out to be accurate approximations of some of the eigenvalues of A, also when n is much smaller than N. The Lanczos method is discussed in many books, e.g., [6, 8, 11, 17, 21, 25].

It is of basic importance for an appreciation of the Lanczos method to understand which eigenvalues of A are approximated by the Ritz values. Outliers in the spectrum

*Received by the editors April 27, 1999; accepted for publication (in revised form) by M. Hanke February 2, 2000; published electronically June 20, 2000. This work was supported in part by FWO research project G.0278.97 and a research grant from the Fund for Scientific Research, Flanders.
http://www.siam.org/journals/simax/22-1/35527.html
†Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200 B, 3001 Leuven, Belgium (arno@wis.kuleuven.ac.be).

are approximated very well, while eigenvalues in the bulk of the spectrum are typically harder to approximate. Trefethen and Bau [25] observe a relationship with electric charge distributions, and they state the following rule of thumb:

(1.1) The Lanczos iteration tends to converge to eigenvalues in regions of "too little charge" for an equilibrium distribution;

see [25, page 279]. This may be understood as follows. Assume that the eigenvalues of A are located on the interval $[-1,1]$, except perhaps for a few outliers. Then one has to compare the distribution of eigenvalues with the equilibrium distribution of $[-1,1]$, which is the measure with density $1/(\pi\sqrt{1-\lambda^2})$. The density of the equilibrium distribution is infinite at the endpoints ± 1. Thus if the eigenvalues of A are spread out more evenly over the interval $[-1,1]$, then the Lanczos method tends to find the extreme eigenvalues. On the other hand, if the eigenvalues of A are distributed like the equilibrium distribution, then the Lanczos iteration is very much useless if $n < N$, and does not find any eigenvalue until $n = N$. See [10] for more on the connection between potential theory and matrix iteration methods.

It is the goal of this paper to provide a quantitative version of the rule of thumb (1.1). We relate the rule of thumb to recent insights, initiated by Rakhmanov [18], on the zero distribution of polynomials satisfying a discrete orthogonality. This relation is used to describe the region of "too little charge" in an asymptotic regime where both N and n tend to infinity. The region depends on the asymptotic distribution of eigenvalues and on the ratio $t = n/N$. Specifically, we associate with a distribution σ and ratio $t \in (0,1)$ an open subset $\Lambda(t;\sigma)$ of $\mathbb{R}$. Under reasonable assumptions discussed in section 2, it is shown that eigenvalues in $\Lambda(t;\sigma)$ are approximated exponentially fast, and we give an estimate for the exponential convergence rate in Theorem 3.1. See [3, 4], where similar ideas are used in connection with other methods from numerical linear algebra.

The sets $\Lambda(t;\sigma)$ are determined explicitly for the two cases

$$d\sigma = \frac{1}{\pi\sqrt{1-\lambda^2}}d\lambda, \qquad \lambda \in [-1,1], \tag{1.2}$$

and

$$d\sigma = \frac{1}{2}d\lambda, \qquad \lambda \in [-1,1] \tag{1.3}$$

in section 4. The first case corresponds to eigenvalues distributed according to the equilibrium measure of $[-1,1]$, and it turns out that $\Lambda(t;\sigma) = \mathbb{R} \setminus [-1,1]$ for all $t \in (0,1)$. Thus only eigenvalues (if any) outside $[-1,1]$ are approximated. The case (1.3) corresponds to equally spaced eigenvalues in $[-1,1]$. We use results of Rakhmanov [18] to prove that in this case

$$\Lambda(t;\sigma) = \mathbb{R} \setminus [-r(t), r(t)] \tag{1.4}$$

with $r(t) = \sqrt{1-t^2}$. Hence, in addition to eigenvalues outside $[-1,1]$, also some extreme eigenvalues in the interval $[-1,1]$ are well approximated. We compare this asymptotic result with the behavior of Ritz values computed for a diagonal matrix with 201 equally spaced eigenvalues; see Figure 4.1 below.

A sufficient condition on the eigenvalue distribution σ is given in section 5, which ensures the behavior (1.4) of the sets $\Lambda(t;\sigma)$. In these cases extreme eigenvalues are

approximated. To illustrate the possibility that interior eigenvalues are found by the Lanczos iteration rather than the extreme eigenvalues, a condition on σ is given in section 6 which implies that for all $t \in (0,1)$,

$$\Lambda(t;\sigma) = (\mathbb{R} \setminus [-1,1]) \cup (-r(t), r(t)) \tag{1.5}$$

for some $r(t) \in [0,1)$. We apply these results to ultraspherical distributions

$$d\sigma(\lambda) = C_\alpha (1-\lambda)^\alpha d\lambda, \qquad \lambda \in [-1,1], \tag{1.6}$$

in section 7. In (1.6) the constant C_α is such that σ is a probability measure. It follows that for $\alpha > -1/2$, the sets $\Lambda(t;\sigma)$ have the form (1.4), while for $\alpha \in (-1,-1/2)$ we have (1.5). This behavior illustrates the rule of thumb (1.1). Indeed, the equilibrium distribution corresponds to $\alpha = -1/2$, and for $\alpha > -1/2$, there is less charge toward the end of the interval $[-1,1]$, while for $\alpha < -1/2$ there is more charge near the endpoints and less charge near 0.

The nature of our results is different from the existing convergence results for the Lanczos method, such as the error bounds of Kaniel [13] and Saad [20]; see also [17, 21]. These are a priori bounds valid for fixed n and N, while our estimates are valid in an asymptotic regime when both n and N tend to infinity. The Kaniel–Saad bounds may greatly overestimate the actual error, since they do not take into account the fine structure of the spectrum. Our results require an a priori knowledge of the asymptotic distribution of eigenvalues. Having this information, our asymptotic error bounds are more precise. Other papers discussing convergence rates of the Lanczos method include [12, 23, 24].

We emphasize that our results are of a theoretical nature and always assume exact arithmetic.

2. Discrete orthogonal polynomials. The orthogonal polynomials come in as follows. The Lanczos iteration is equivalent to a polynomial minimization problem. Let $p_n(\lambda) = \det(\lambda I - T_n)$ be the characteristic polynomial of T_n. Then p_n is a monic polynomial of degree n that minimizes $\|p_n(A)b\|_2$ among all monic polynomials of degree n. The zeros of p_n are of course equal to the Ritz values. The norm is equal to

$$\|p_n(A)b\|_2 = \left(\sum_{k=1}^N \langle b, e_k \rangle^2 p_n(\lambda_k)^2 \right)^{1/2}, \tag{2.1}$$

where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of A and $e_1, \ldots, e_N$ are the corresponding orthonormal eigenvectors. Thus p_n is orthogonal with respect to the discrete measure

$$\sum_{k=1}^N \langle b, e_k \rangle^2 \delta_{\lambda_k},$$

which has mass $\langle b, e_k \rangle^2$ at the eigenvalue λ_k. Here δ_{λ_k} is the Dirac measure concentrated at λ_k.

We are going to consider the situation where both N and n tend to infinity. We assume that we have a sequence of matrices (A_N) with A_N a real symmetric matrix of size $N \times N$. The eigenvalues of A_N are denoted by

$$\lambda_{1,N} < \lambda_{2,N} < \cdots < \lambda_{N,N},$$

and they are assumed to be distinct. This is not an essential restriction, since all of our results would remain valid if we would assume instead that A_N is of size $N' \times N'$ with $N' \geq N$, and A_N has exactly N distinct eigenvalues $\lambda_{1,N} < \cdots < \lambda_{N,N}$. Only for ease of exposition we assume that all eigenvalues are distinct. We also assume that the eigenvalues $\lambda_{k,N}$ are all contained in a fixed bounded interval and that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{\lambda_{k,N}} = \sigma, \tag{2.2}$$

with σ a Borel probability measure on $\mathbb{R}$ with compact support. The convergence is in the sense of weak convergence of measure. Thus for every continuous function $f : \mathbb{R} \to \mathbb{R}$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} f(\lambda_{k,N}) = \int f \, d\sigma.$$

The relation (2.2) expresses that σ is the asymptotic distribution of the eigenvalues. In many practical situations, matrices A_N appear as discretizations of a continuous operator. The size N is related to the mesh size of the discretization. A relation like (2.2) may then very well hold, where the measure σ is determined by the spectral properties of the continuous operator; see, e.g., [1, 3, 12]. Note that (2.2) forces "most" eigenvalues $\lambda_{k,N}$ to be in—or close to—the support of the measure σ. However, it does not exclude outliers lying anywhere on the real line, as long as their number is $o(N)$ as $N \to \infty$.

We also have for each N a starting vector $b_N \in \mathbb{R}^N$ which we assume to be normalized so that $\|b_N\|_2 = 1$. Thus

$$\sum_{k=1}^{N} \langle b_N, e_{k,N} \rangle^2 = 1,$$

where $(e_{k,N})_{k=1}^{N}$ is an orthonormal basis of eigenvectors of A_N. We assume that the vectors b_N are chosen sufficiently random, so that none of its Fourier coefficients in the basis $(e_{k,N})$ is exponentially small as $N \to \infty$. That is, we assume

$$\lim_{N\to\infty} \left(\min_{1\leq k\leq N} |\langle b_N, e_{k,N} \rangle| \right)^{1/N} = 1. \tag{2.3}$$

We need a further technical condition on the spacings of the eigenvalues, which prevents them from being too close. A possible condition is to assume that there exists $c > 0$ such that

$$|\lambda_{k+1,N} - \lambda_{k,N}| \geq \frac{c}{N} \tag{2.4}$$

for all N and all $k = 1, \ldots, N-1$. This condition was used by Rakhmanov [18] to prove Theorem 2.1 stated below. A more general condition was introduced by Dragnev and Saff [9], which we will also use here. We assume that whenever, for each N, an index $k = k_N \in \{1, \ldots, N\}$ is chosen such that

$$\lim_{N\to\infty} \lambda_{k,N} = \lambda \in \mathbb{R},$$

then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1, j\neq k}^{N} \log|\lambda_{k,N} - \lambda_{j,N}| = \int \log|\lambda - \lambda'|\, d\sigma(\lambda'). \tag{2.5}$$

As pointed out in [9], under the assumption (2.2) the condition (2.5) is strictly weaker than (2.4). For example, the Chebyshev points $\lambda_{k,N} = \cos((k-1/2)\pi/N)$ satisfy (2.5) with $d\sigma = 1/(\pi\sqrt{1-\lambda^2})\, d\lambda$, but they do not satisfy (2.4); see [9, Lemma 3.2], where also zeros of more general orthogonal polynomials are discussed. The condition (2.5) also prevents eigenvalues from getting too close, but not as strictly as (2.4) does. It is possible that (2.5) holds and to have a pair of eigenvalues at a distance $1/N^p$ for some $p > 0$. On the other hand, two exponentially close eigenvalues, i.e., $|\lambda_{k+1,N} - \lambda_{k,N}| \leq e^{-cN}$ for some $c > 0$, are not possible if (2.5) holds.

In what follows, we use U^μ to denote the logarithmic potential of a measure μ, i.e.,

$$U^\mu(\lambda) = \int \log \frac{1}{|\lambda - \lambda'|}\, d\mu(\lambda').$$

Thus the right-hand side of (2.5) is equal to $-U^\sigma(\lambda)$. It can be shown from (2.5) that $U^\sigma(\lambda)$ is a continuous function of $\lambda \in \mathbb{C}$. In particular, σ has no mass points.

For $0 \leq n \leq N$, we denote by $p_{n,N}$ the nth degree monic Lanczos polynomial associated with A_N. The zeros of $p_{n,N}$ are real and simple and we denote them by

$$\theta_{1,n,N} < \theta_{2,n,N} < \cdots < \theta_{n,n,N}.$$

The following is Rakhmanov's result in the more general situation given by Dragnev and Saff.

THEOREM 2.1. *Assume* (2.2), (2.3), and (2.5). *Let $n, N \to \infty$ in such a way that $n/N \to t \in (0,1)$. Then there is a Borel probability measure μ_t, depending only on t and σ, such that*

$$\lim_{N\to\infty} \frac{1}{n} \sum_{j=1}^{n} \delta_{\theta_{j,n,N}} = \mu_t \tag{2.6}$$

and a real constant F_t such that

$$\lim_{N\to\infty} \|p_{n,N}(A_N) b_N\|_2^{1/n} = \exp(-F_t). \tag{2.7}$$

The measure μ_t satisfies

$$0 \leq t\mu_t \leq \sigma, \qquad \int d\mu_t = 1 \tag{2.8}$$

and minimizes the logarithmic energy

$$\int\int \log \frac{1}{|\lambda - \lambda'|}\, d\mu(\lambda)\, d\mu(\lambda')$$

among all measures μ satisfying $0 \leq t\mu \leq \sigma$ and $\int d\mu = 1$. The logarithmic potential U^{μ_t} of μ_t is a continuous function on $\mathbb{C}$, and the constant F_t is such that

$$U^{\mu_t}(\lambda) = F_t \qquad \text{for } \lambda \in \operatorname{supp}(\sigma - t\mu_t), \tag{2.9}$$

$$U^{\mu_t}(\lambda) \leq F_t \qquad \text{for } \lambda \in \mathbb{C}. \tag{2.10}$$

The relations (2.8)–(2.10) *characterize the pair* (μ_t, F_t).

Proof. See Theorem 3.3 of [9]. In this paper it is assumed that $\operatorname{supp}(\sigma)$ is connected. However, this is not essential. See also [15] or [2, Theorem 1.3]. ∎

We note that in [9] a more general situation is considered which also involves an external field. Results similar to Theorem 2.1, under conditions different from (2.5), were given in [16] and [2].

Using (2.2), (2.6), and (2.8) one can easily show that for an interval (a, b), one has

$$\lim_{N\to\infty} \frac{\#\{j : \theta_{j,n,N} \in (a,b)\} - \#\{j : \lambda_{j,N} \in (a,b)\}}{N} = 0$$

if and only if

$$(a,b) \cap \operatorname{supp}(\sigma - t\mu_t) = \emptyset.$$

Thus one can expect convergence of Ritz values only outside the support of $\sigma - t\mu_t$. The set $\mathbb{R} \setminus \operatorname{supp}(\sigma - t\mu_t)$ may still be too big. Instead we consider $\Lambda(t;\sigma)$ defined in terms of μ_t and F_t as

$$\Lambda(t;\sigma) := \{\lambda \in \mathbb{R} : U^{\mu_t}(\lambda) < F_t\}. \tag{2.11}$$

From (2.9) it is clear that

$$\Lambda(t;\sigma) \subset \mathbb{R} \setminus \operatorname{supp}(\sigma - t\mu_t),$$

but equality need not hold in general. As indicated in the introduction, the sets (2.11) will be the regions of too little charge compared to the equilibrium distribution, as will become clear in the rest of the paper.

3. Main result. We assume we are in a situation as described in the previous section. That is, we have a sequence (A_N) of real symmetric matrices with eigenvalues $\lambda_{1,N}, \ldots, \lambda_{N,N}$. For $1 \le n \le N$, we have the Ritz values $\theta_{1,n,N}, \ldots, \theta_{n,n,N}$ generated by the Lanczos iteration with starting vector b_N. Our main result is the following.

THEOREM 3.1. *Assume* (2.2), (2.3), *and* (2.5). *Let* $k = k_N$ *be such that*

$$\lim_{N\to\infty} \lambda_{k,N} = \lambda.$$

Let $0 < t < 1$ *and assume* $n = n_N$ *is such that* $n/N \to t$ *as* $N \to \infty$. *Then*

$$\limsup_{N\to\infty} \left(\min_{1\le j\le n} |\lambda_{k,N} - \theta_{j,n,N}| \right)^{1/n} \le \exp\left(-(F_t - U^{\mu_t}(\lambda))/2\right), \tag{3.1}$$

where μ_t *and* F_t *are as in Theorem* 2.1.

Proof. We are going to estimate $p_{n,N}(\lambda_{k,N})$ in two ways. First, we have

$$\begin{aligned} \limsup_{N\to\infty} |p_{n,N}(\lambda_{k,N})|^{1/n} &= \limsup_{N\to\infty} |\langle b_N, e_{k,N}\rangle p_{n,N}(\lambda_{k,N})|^{1/n} \\ &\le \limsup_{N\to\infty} \|p_{n,N}(A_N) b_N\|_2^{1/n} \\ &= \exp(-F_t), \end{aligned} \tag{3.2}$$

where we used (2.3), (2.1), and (2.7), respectively.

Next, we note that

$$|p_{n,N}(\lambda_{k,N})| = \prod_{j=1}^{n} |\lambda_{k,N} - \theta_{j,n,N}|,$$

and to estimate the product, we are going to divide the Ritz values into three groups. First we let $j_0 \in \{1, \ldots, n\}$ be such that

$$\theta_{j_0,n,N} \leq \lambda_{k,N} < \theta_{j_0+1,n,N}. \tag{3.3}$$

(In case $\lambda_{k,N} \geq \theta_{n,n,N}$ or $\lambda_{k,N} < \theta_{1,n,N}$, the proof simplifies, and we will not consider this case explicitly.) Thus $\theta_{j_0,n,N}$ and $\theta_{j_0+1,n,N}$ are the Ritz values closest to $\lambda_{k,N}$. We put

$$J_0 := \{j_0, j_0 + 1\}.$$

For a given $r > 0$, we further introduce the sets

$$\begin{aligned} J_1 &:= \{j \in \{1, \ldots, n\} : |\theta_{j,n,N} - \lambda| < r,\ j \neq j_0,\ j \neq j_0 + 1\}, \\ J_2 &:= \{j \in \{1, \ldots, n\} : |\theta_{j,n,N} - \lambda| \geq r\}. \end{aligned}$$

For N large enough, the sets J_0, J_1, and J_2 form a partition of $\{1, \ldots, n\}$. The set J_2 contains the Ritz values that are "far" from λ. The weak convergence (2.6), together with the fact that σ and thus μ_t have no mass points, implies that

$$\lim_{N\to\infty} \frac{1}{n} \sum_{j\in J_2} \delta_{\theta_{j,n,N}} = \mu_t - (\mu_t)|_r,$$

where we use $(\mu_t)|_r$ to denote the restriction of μ_t to $[\lambda - r, \lambda + r]$. Since $\lambda_{k,N} \to \lambda$, it then follows that

$$\lim_{N\to\infty} \frac{1}{n} \sum_{j\in J_2} \log |\lambda_{k,N} - \theta_{j,n,N}| = -U^{\mu_t - (\mu_t)|_r}(\lambda). \tag{3.4}$$

Since the eigenvalues of A_N separate the Ritz values, it follows from (3.3) that

$$\theta_{j,n,N} < \lambda_{k+j-j_0,N} \qquad \text{if } j < j_0$$

and

$$\theta_{j,n,N} > \lambda_{k+j-j_0-1,N} \qquad \text{if } j > j_0 + 1.$$

Then

$$|\lambda_{k,N} - \theta_{j,n,N}| \geq |\lambda_{k,N} - \lambda_{k+j-j_0,N}| \qquad \text{if } j < j_0$$

and

$$|\lambda_{k,N} - \theta_{j,n,N}| \geq |\lambda_{k,N} - \lambda_{k+j-j_0-,N}| \qquad \text{if } j > j_0 + 1.$$

This implies that

$$\sum_{j\in J_1} \log |\lambda_{k,N} - \theta_{j,n,N}| \geq \sum_{i\neq k,\ |\lambda_{i,N}-\lambda|<r} \log |\lambda_{i,N} - \lambda_{k,N}|. \tag{3.5}$$

From (2.2) it follows that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{|\lambda_{i,N}-\lambda|\geq r}\log|\lambda_{k,N}-\lambda_{i,N}| = -U^{\sigma-\sigma|_r}(\lambda), \tag{3.6}$$

where $\sigma|_r$ is the restriction of σ to $[\lambda-r,\lambda+r]$, and from (2.5), we get

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i\neq k}\log|\lambda_{k,N}-\lambda_{i,N}| = -U^{\sigma}(\lambda). \tag{3.7}$$

Combining (3.6) and (3.7) we see that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i\neq k,\ |\lambda_{i,N}-\lambda|<r}\log|\lambda_{k,N}-\lambda_{i,N}| = -U^{\sigma|_r}(\lambda). \tag{3.8}$$

Then by (3.5) and (3.8) we get

$$\liminf_{N\to\infty}\frac{1}{N}\sum_{j\in J_1}\log|\lambda_{k,N}-\theta_{j,n,N}| \geq -U^{\sigma|_r}(\lambda). \tag{3.9}$$

Then by (3.4) and (3.9),

$$\liminf_{N\to\infty}\frac{1}{n}\sum_{j\in J_1\cup J_2}\log|\lambda_{k,N}-\theta_{j,n,N}| \geq -U^{\mu_t}(\lambda)+U^{(\mu_t)|_r}(\lambda)-\frac{1}{t}U^{\sigma|_r}(\lambda).$$

This holds for every $r>0$. Note that the left-hand side does not depend on r. Letting $r\to 0$, we get from Lebesgue's dominated convergence theorem that the potentials of $(\mu_t)|_r$ and $\sigma|_r$ tend to 0. Therefore

$$\liminf_{N\to\infty}\frac{1}{n}\sum_{j\neq j_0,j_0+1}\log|\lambda_{k,N}-\theta_{j,n,N}| \geq -U^{\mu_t}(\lambda),$$

and this is

$$\liminf_{N\to\infty}\left|\frac{p_{n,N}(\lambda_{k,N})}{(\lambda_{k,N}-\theta_{j_0,n,N})(\lambda_{k,N}-\theta_{j_0+1,n,N})}\right|^{1/n} \geq \exp\left(-U^{\mu_t}(\lambda)\right), \tag{3.10}$$

which is the other estimate we need on $|p_{n,N}(\lambda_{k,N})|$.

Combining (3.2) and (3.10) we obtain

$$\limsup_{N\to\infty}|(\lambda_{k,N}-\theta_{j_0,n,N})(\lambda_{k,N}-\theta_{j_0+1,n,N})|^{1/n} \leq \exp\left(-(F_t-U^{\mu_t}(\lambda))\right). \tag{3.11}$$

Then (3.1) follows, and the proof of the theorem is complete. □

Remark 3.2. By (2.9)–(2.10) we always have $F_t - U^{\mu_t}(\lambda)\geq 0$. The theorem does not give any information if $U^{\mu_t}(\lambda)=F_t$. Then the right-hand side of (3.1) is 1 and we cannot expect to find a Ritz value close to $\lambda_{k,N}$. However, if $U^{\mu_t}(\lambda)<F_t$, the right-hand side of (3.1) is less than 1. Then for N large enough, every eigenvalue of A_N close to λ is approximated by some Ritz value at an exponential rate. Recalling the definition of $\Lambda(t;\sigma)$ given in (2.11), we see indeed that eigenvalues in $\Lambda(t;\sigma)$ are well approximated by the Lanczos iteration as $n,N\to\infty$ and $n/N\to t$. The set

(2.11) is the region of too little charge for an eigenvalue distribution referred to in the rule of thumb (1.1). This will also become clear from the examples.

Remark 3.3. The factor $1/2$ in $(F_t - U^{\mu_t}(\lambda))/2$ in (3.1) does not seem natural at first sight. However, analyzing the proof of Theorem 3.1—especially (3.11)—it becomes clear that the factor $1/2$ appears if two Ritz values are exponentially close to the same eigenvalue. This could happen, for example, in a situation which is perfectly symmetric around 0 and where $F_t - U^{\mu_t}(0) > 0$. If N is odd, then 0 is an eigenvalue. If, in addition, n is even, then the Ritz values come in pairs, and there will be two Ritz values close to 0. In such a case the exponential convergence rate $(F_t - U^{\mu_t}(\lambda))/2$ arises. Such cases, however, are exceptional. In most cases, one expects only one Ritz value to approximate a particular eigenvalue. Then we can conclude from (3.11) that the exponential convergence rate in (3.1) improves to $F_t - U^{\mu_t}(\lambda)$.

However, it seems likely that the error estimate (3.1) is not best possible and can be improved in all cases. More delicate estimates may lead to

$$\limsup_{N\to\infty} \left(\min_{1\le j\le n} |\lambda_{k,N} - \theta_{j,n,N}| \right)^{1/n} \le \exp(-(F_t - U^{\mu_t}(\lambda)))$$

in general and to

$$\limsup_{N\to\infty} \left(\min_{1\le j\le n} |\lambda_{k,N} - \theta_{j,n,N}| \right)^{1/n} \le \exp(-2(F_t - U^{\mu_t}(\lambda)))$$

in all but the exceptional cases. I am very grateful to one of the referees for pointing out that (3.1) may be improved.

4. First examples. We have seen that, for a given eigenvalue distribution σ and a ratio $t = n/N$, eigenvalues in the set

$$\Lambda(t;\sigma) = \{\lambda \in \mathbb{R} : F_t - U^{\mu_t}(\lambda) > 0\}$$

are well approximated by the Lanczos method if N is large. We will determine $\Lambda(t;\sigma)$ in a number of cases. In general, it is an open set, since U^{μ_t} is a continuous function, and by (2.9) it is disjoint from the support of $\sigma - t\mu_t$. In many cases, $\Lambda(t;\sigma)$ is equal to $\mathbb{R} \setminus \operatorname{supp}(\sigma - t\mu_t)$.

4.1. Eigenvalues distributed as the equilibrium distribution. Suppose the eigenvalues of the matrices A_N are distributed like the equilibrium distribution of $[-1,1]$ as $N \to \infty$. This is, for example, the case if the eigenvalues are the Chebyshev points

$$\lambda_{k,N} = \cos\left(\frac{(k-1/2)\pi}{N}\right), \qquad 1 \le k \le N.$$

Then the measure σ from (2.2) is

$$d\sigma(\lambda) := \frac{1}{\pi\sqrt{1-\lambda^2}} d\lambda, \qquad \lambda \in [-1,1]. \tag{4.1}$$

The equilibrium measure σ satisfies $U^\sigma(\lambda) = \log 2$ if $\lambda \in [-1,1]$, and $U^\sigma(\lambda) < \log 2$ if $\lambda \in \mathbb{C} \setminus [-1,1]$; see, e.g., [19, 22]. It then follows from the relations (2.9)–(2.10) that characterize the measure μ_t and the constant F_t that $\mu_t = \sigma$ and $F_t = \log 2$ for every $t \in (0,1)$. Thus

$$\Lambda(t;\sigma) = \mathbb{R} \setminus [-1,1] \tag{4.2}$$

for every $t \in (0,1)$. We see that eigenvalues outside $[-1,1]$ (if any) are found by the Lanczos iteration, but no eigenvalues in $[-1,1]$. This confirms the idea from the rule of thumb that a distribution of eigenvalues according to the equilibrium measure is the worst possible case for the Lanczos iteration with $n < N$.

4.2. Equally spaced eigenvalues. Suppose the eigenvalues are (more or less) equally spaced on $[-1,1]$, so that

$$d\sigma(\lambda) = \frac{1}{2} d\lambda, \qquad \lambda \in [-1,1]. \tag{4.3}$$

The measures μ_t and the constants F_t for this case were determined by Rakhmanov [18]; see also [15]. Let $t \in (0,1)$ and

$$r = r(t) = \sqrt{1-t^2}. \tag{4.4}$$

Then

$$\frac{d\mu_t}{d\lambda} = \begin{cases} \dfrac{1}{2t} & \text{for } \lambda \in [-1,-r] \cup [r,1], \\[2ex] \dfrac{1}{\pi t} \arctan \dfrac{t}{\sqrt{r^2-\lambda^2}} & \text{for } \lambda \in [-r,r], \end{cases}$$

and

$$F_t = 1 + \log \frac{2}{\sqrt{1-t^2}} - \frac{1}{2t} \log \frac{1+t}{1-t}.$$

Hence we have

$$\mathbb{R} \setminus \operatorname{supp}(\sigma - t\mu_t) = \mathbb{R} \setminus [-r,r] = \mathbb{R} \setminus [-\sqrt{1-t^2}, \sqrt{1-t^2}].$$

In this case, one can show that $\Lambda(t;\sigma) = \mathbb{R} \setminus \operatorname{supp}(\sigma - t\mu_t)$; see also (5.6) below. Thus

$$\Lambda(t;\sigma) = \mathbb{R} \setminus [-\sqrt{1-t^2}, \sqrt{1-t^2}]. \tag{4.5}$$

Thus the eigenvalues outside the interval $[-\sqrt{1-t^2}, \sqrt{1-t^2}]$ are found by the Lanczos iteration if $n/N \to t$. These are the extreme eigenvalues on both sides, and their number is approximately

$$(1-r)N = (1-\sqrt{1-t^2})N = \frac{t^2}{1+\sqrt{1-t^2}} N.$$

The measure (4.3) has less charge than the equilibrium measure (4.1) near the endpoints ± 1 and more charge towards the middle of the interval $[-1,1]$, especially near 0. The sets (4.5) illustrate nicely the rule of thumb (1.1).

To see how Theorem 3.1, which is an asymptotic result, compares to a particular situation with finite N, we performed experiments with diagonal matrices A_N with N equally spaced eigenvalues

$$\lambda_{k,N} = -1 + 2\frac{k-1}{N-1}, \qquad k = 1, \ldots, N,$$

in the interval $[-1,1]$. The starting vector b is the all-one vector.

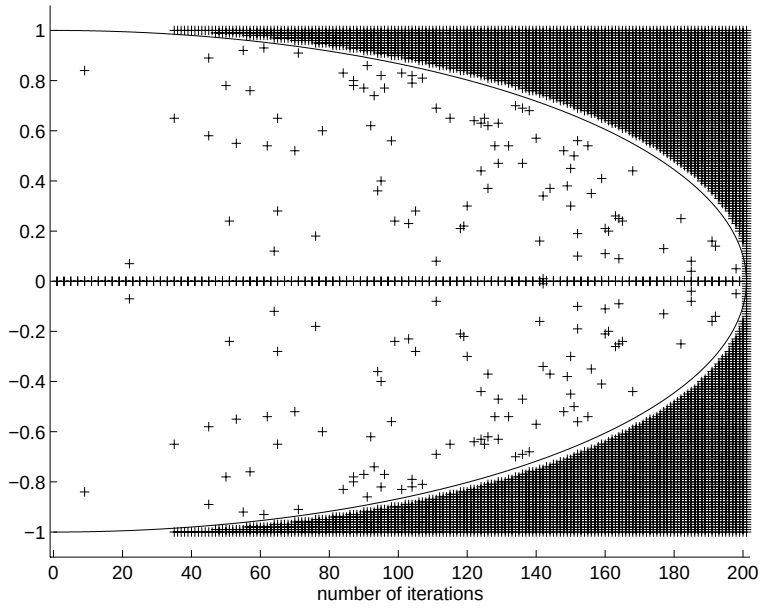


FIG. 4.1. *The figure shows, for every number n of iterations, the Ritz values that are closer than* 10^{-4} *to an eigenvalue. Calculations were done for a diagonal matrix with* $N = 201$ *equally spaced eigenvalues in* $[-1, 1]$. *The figure also shows the curves* $\pm r(t)$ *from* (4.4), *with* $t = n/N$. *These curves determine the sets* $\Lambda(t; \sigma)$ *of eigenvalues that are well approximated in the asymptotic sense; see* (4.5).

The results for $N = 201$ are shown in Figure 4.1. For every iteration, the Ritz values are calculated and compared with the eigenvalues. A "+" indicates a Ritz value that is closer than 10^{-4} to one of the eigenvalues. Also shown are the curves $r(t)$ and $-r(t)$ with t equal to the number of iterations n divided by N. According to (4.5) the eigenvalues outside $[-r(t), r(t)]$ are found if $n/N \to t$. The figure is in good agreement with the predicted asymptotic behavior. All Ritz values slightly bigger than $r(t)$ or slightly smaller than $-r(t)$ are very close to an eigenvalue. Some Ritz values inside the parabolic region bounded by $\pm r(t)$ are also close to an eigenvalue, but not in any systematic way (apart from 0, which by symmetry is a Ritz value for every odd-numbered iteration). These Ritz values are only "by accident" close to an eigenvalue.

The computations were done with MATLAB, based on a code written by J.W. Demmel for Lanczos iteration with full reorthogonalization. This code is part of a collection of MATLAB codes accompanying the book [8].

5. A sufficient condition to find extreme eigenvalues. In this section we give a sufficient condition on the eigenvalue distribution, which guarantees that the Lanczos method finds the extreme eigenvalues. Our result is the following.

THEOREM 5.1. *Let* σ *be supported on* $[-1, 1]$ *with a density* $w(\lambda)$ *which is even on* $[-1, 1]$, *i.e.,* $w(-\lambda) = w(\lambda)$. *Assume* $\sqrt{1-\lambda^2}w(\lambda)$ *decreases for* $\lambda \in [0, 1]$. *Then for every* $t \in (0, 1)$, *there exists* $r(t) \in (0, 1]$ *such that*

$$\Lambda(t; \sigma) = \mathbb{R} \setminus [-r(t), r(t)]. \tag{5.1}$$

The proof of the theorem is based on a number of lemmas related to extremal measures in an external field. The connection of the measures μ_t with problems with

external fields was pointed out by Rakhmanov [18], Kuijlaars and Rakhmanov [15], and Dragnev and Saff [9]. We review here the necessary notions.

For a continuous function Q on a compact interval Σ, the problem is to minimize

$$\int\int \log \frac{1}{|\lambda - \lambda'|}\, d\nu(\lambda)\, d\nu(\lambda') + 2\int Q(\lambda)\, d\nu(\lambda)$$

among all Borel probability measures ν supported on Σ. There is a unique minimizing measure, denoted here by ν_Q, with a support $S_Q = \mathrm{supp}(\nu_Q)$. We call ν_Q the extremal measure in the external field Q. The minimizer satisfies

$$U^{\nu_Q}(\lambda) + Q(\lambda) = C \qquad \text{if } \lambda \in S_Q, \tag{5.2}$$

$$U^{\nu_Q}(\lambda) + Q(\lambda) \geq C \qquad \text{if } \lambda \in \Sigma, \tag{5.3}$$

where C is a real constant. The relations (5.2)–(5.3) characterize the measure ν_Q and the constant C.

Now assume we have a measure σ with compact support as before. Assume its potential U^σ is continuous. Associated with σ we have the measures μ_t and the constants F_t for $t \in (0,1)$ as described in Theorem 2.1. Then for $t \in (0,1)$

$$\nu_t := \frac{\sigma - t\mu_t}{1-t} \tag{5.4}$$

is a probability measure on Σ, and rewriting the relations (2.9)–(2.10) we can easily show that ν_t is the extremal measure in the external field

$$Q_t := -\frac{1}{1-t} U^\sigma, \tag{5.5}$$

with constants $C_t := -(t/(1-t))F_t$. A comprehensive account about extremal measures in external fields can be found in [22]. See also [7].

As far as the t-dependence is concerned, it is important for us to know that the support of ν_t decreases as the parameter t increases; see [22, Theorem IV.1.6]. Thus in view of (5.4) the support of $\sigma - t\mu_t$ is decreasing. Also it is known that

$$\{\lambda : U^{\nu_t}(\lambda) + Q_t(\lambda) = C_t\} = \bigcap_{\epsilon > 0} \mathrm{supp}(\nu_{t-\epsilon});$$

see [22, Theorem IV.1.6] or [5]. Then it is easy to see that

$$\Lambda(t;\sigma) = \mathbb{R} \setminus \{\lambda : U^{\nu_t}(\lambda) + Q_t(\lambda) = C_t\} = \mathbb{R} \setminus \bigcap_{\epsilon > 0} \mathrm{supp}(\nu_{t-\epsilon}). \tag{5.6}$$

For the proof of Theorem 5.1, we need the following recent result.

LEMMA 5.2. *Let Q be a continuous external field on $[0,1]$ and let v be an integrable function on $[0,1]$ such that*

$$\int_0^1 \log|\lambda - \lambda'| v(\lambda')\, d\lambda' = Q(\lambda) \qquad \text{if } \lambda \in [0,1].$$

Then the following hold.

(a) *If $\sqrt{\lambda(1-\lambda)}v(\lambda)$ is decreasing on $[0,1]$, then $S_Q = [0,r]$ for some $r \in (0,1]$.*

(b) *If* $\sqrt{\lambda(1-\lambda)}v(\lambda)$ *is increasing on* $[0,1]$*, then* $S_Q = [r,1]$ *for some* $r \in [0,1)$.

Proof. See Theorem 2 of [14], where this result was proved under the assumption that Q is differentiable with a Hölder continuous derivative. The same proof works in the present situation. □

By a quadratic transformation we can use the lemma for even external fields on $[-1,1]$.

LEMMA 5.3. *Let* Q *be an even continuous external field on* $[-1,1]$ *and let* v *be an even integrable function on* $[-1,1]$ *such that*

$$\int_{-1}^{1} \log|\lambda-\lambda'|v(\lambda')\,d\lambda' = Q(\lambda) \qquad \text{if } \lambda \in [-1,1].$$

Then the following hold.

(a) *If* $\sqrt{1-\lambda^2}v(\lambda)$ *is decreasing on* $[0,1]$*, then* $S_Q = [-r,r]$ *for some* $r \in (0,1]$.

(b) *If* $\sqrt{1-\lambda^2}v(\lambda)$ *is increasing on* $[0,1]$*, then* $S_Q = [-1,r] \cup [r,1]$ *for some* $r \in [0,1)$.

Proof. Let $\tilde{Q}(\lambda) := Q(\sqrt{\lambda})$ for $\lambda \in [0,1]$. Then it is easy to see that

$$\int_0^1 \log|\lambda-\lambda'|\tilde{v}(\lambda')\,d\lambda' = \tilde{Q}(\lambda) \qquad \text{if } \lambda \in [0,1],$$

where

$$\tilde{v}(\lambda) = \frac{v(\sqrt{\lambda})}{2\sqrt{\lambda}} \qquad \text{for } \lambda \in [0,1].$$

Then

$$\sqrt{\lambda(1-\lambda)}\tilde{v}(\lambda) = \frac{1}{2}\sqrt{1-\lambda}v(\sqrt{\lambda}) \qquad \text{for } \lambda \in [0,1].$$

If we are in part (a), then we see that $\sqrt{\lambda(1-\lambda)}\tilde{v}(\lambda)$ decreases on $[0,1]$. From part (a) of Lemma 5.2 it follows that $S_{\tilde{Q}} = [0,r^2]$ for some $r \in (0,1]$. Then it readily follows that $S_Q = [-r,r]$.

Similarly, if we are in part (b), then $\sqrt{\lambda(1-\lambda)}\tilde{v}(\lambda)$ increases on $[0,1]$, and from part (b) of Lemma 5.2 it then follows that $S_{\tilde{Q}} = [r^2,1]$ for some $r \in [0,1)$. Then $S_Q = [-1,-r] \cup [r,1]$. □

Now we are ready for the proof of Theorem 5.1.

Proof. Let

$$Q_t(\lambda) = -\frac{1}{1-t}U^{\sigma}(\lambda) = \frac{1}{1-t}\int_{-1}^{1} \log|\lambda-\lambda'|w(\lambda')\,d\lambda'$$

be the external field as in (5.5). Then we are clearly in the situation of Lemma 5.3 with $v(\lambda) = (1/(1-t))w(\lambda)$. The assumption on w gives that we are in part (a) of Lemma 5.3, and it follows that $\operatorname{supp}(\nu_t) = S_{Q_t} = [-r_0(t), r_0(t)]$ for some $r_0(t) \in (0,1]$. Then $r_0(t)$ is a decreasing function of t. In view of (5.6) we then get

$$\Lambda(t;\sigma) = \mathbb{R} \setminus [-r(t), r(t)]$$

with

$$r(t) := r_0(t-) = \lim_{\epsilon \to 0+} r_0(t-\epsilon). \tag{5.7}$$

This completes the proof of Theorem 5.1. □

6. A sufficient condition for interior eigenvalues. Using ideas similar to those in the previous section, we can give a condition that guarantees that the Lanczos method does not find the extreme eigenvalues, but rather the eigenvalues in the interior of the spectrum.

THEOREM 6.1. *Let σ be supported on $[-1,1]$ with an even density $w(\lambda)$ on $[-1,1]$. Assume $\sqrt{1-\lambda^2}w(\lambda)$ increases for $\lambda \in [0,1]$. Then for every $t \in (0,1)$, we have either*

$$\Lambda(t;\sigma) = (-\infty,-1) \cup (1,\infty) \tag{6.1}$$

or

$$\Lambda(t;\sigma) = (-\infty,-1) \cup (-r(t),r(t)) \cup (1,\infty) \tag{6.2}$$

for some $r(t) \in (0,1)$.

Proof. The proof is the same as the proof of Theorem 5.1. The only difference is that we use part (b) of Lemma 5.3 instead of part (a). □

If we are in case (6.1), then no eigenvalues in $[-1,1]$ are well approximated by the Lanczos method. If we are in case (6.2), then we see that eigenvalues in an interval around 0 are found, but not the extreme eigenvalues in $[-1,1]$ close to ± 1.

The function $r(t)$ in Theorem 6.1 increases with t.

7. More examples: Ultraspherical distributions. We illustrate Theorems 5.1 and 6.1 using ultraspherical distributions

$$\frac{d\sigma}{d\lambda} = C_\alpha (1-\lambda^2)^\alpha \qquad \text{if } \lambda \in [-1,1], \tag{7.1}$$

with $\alpha > -1$, and $C_\alpha := \Gamma(\alpha+3/2)/(\sqrt{\pi}\Gamma(\alpha+1))$ is such that σ is a probability measure on $[-1,1]$. The distributions considered in section 4 belong to this class.

We have

$$\sqrt{1-\lambda^2}\frac{d\sigma}{d\lambda} = C_\alpha (1-\lambda^2)^{\alpha+1/2}$$

and this decreases on $[0,1]$ if $\alpha > -1/2$, and increases on $[0,1]$ if $-1 < \alpha < -1/2$. Thus if $\alpha > -1/2$, we have

$$\Lambda(t;\sigma) = \mathbb{R} \setminus [-r(t),r(t)] \tag{7.2}$$

with $r(t) \in (0,1]$ by Theorem 5.1. Hence in this case the extreme eigenvalues are computed by the Lanczos method. On the other hand, if $-1 < \alpha < -1/2$, then Theorem 6.1 applies, and it follows that either no eigenvalues in $[-1,1]$ or only eigenvalues in an interval around 0 are found.

It is possible to compute the numbers $r(t)$ from (7.2) explicitly. We assume $\alpha > -1/2$. Recall the connection with the extremal measure ν_t in the presence of the external field

$$Q_t(\lambda) = -\frac{1}{1-t}U^\sigma(\lambda)$$

as discussed in section 5. By Lemma 5.3, the support of ν_t is an interval $[-r_0(t),r_0(t)]$ and (5.7) gives the relation of $r_0(t)$ with $r(t)$. By Theorem IV.1.5 of [22], the number $r = r_0(t)$ maximizes the Mhaskar–Saff functional

$$F(r) := \log(r/2) - \int_{-r}^{r} Q_t(\lambda)\frac{d\lambda}{\pi\sqrt{r^2-\lambda^2}}.$$

A little calculation shows that

$$F(r) = \log(r/2) - \frac{1}{1-t}\int_{-r}^{r}\int_{-1}^{1} \log|\lambda-\lambda'|\, d\sigma(\lambda')\frac{d\lambda}{\pi\sqrt{r^2-\lambda^2}}$$
$$= \log(r/2) - \frac{2}{1-t}\int_0^1 \left(\int_{-r}^{r} \log|\lambda-\lambda'|\frac{d\lambda}{\pi\sqrt{r^2-\lambda^2}}\right) d\sigma(\lambda').$$

The inner integral is minus the potential of the equilibrium measure of $[-r,r]$ and its value is equal to $\log(r/2)$ for $\lambda' \in [0,r]$ and to

$$\log(r/2) + \log\left|\frac{\lambda'}{r} + \sqrt{\left(\frac{\lambda'}{r}\right)^2 - 1}\right|$$

for $\lambda' \in [r,1]$. Thus

$$F(r) = \left(1 - \frac{1}{1-t}\right)\log(r/2) - \frac{2}{1-t}\int_r^1 \log\left|\frac{\lambda'}{r} + \sqrt{\left(\frac{\lambda'}{r}\right)^2 - 1}\right| d\sigma(\lambda').$$

Taking the derivative with respect to r and equating this to 0, we find that $r = r_0(t)$ and t satisfy (we write λ instead of λ')

$$t = 2\int_r^1 \frac{\lambda}{\sqrt{\lambda^2 - r^2}}\, d\sigma(\lambda). \tag{7.3}$$

After substituting the formula (7.1) for σ, and introducing a change of variables $y = (\lambda^2 - r^2)/(1-r^2)$, we find

$$t = C_\alpha (1-r^2)^{\alpha+1/2}\int_0^1 \frac{(1-y)^\alpha}{\sqrt{y}}\, dy = (1-r^2)^{\alpha+1/2},$$

since the value of the beta integral is exactly C_α^{-1}. Rewriting this, we see

$$r = r_0(t) = \sqrt{1 - t^{2/(2\alpha+1)}}.$$

Since $r_0(t)$ is continuous in t, we finally find using (5.7)

$$r(t) = \lim_{\epsilon\to 0+} r_0(t-\epsilon) = \sqrt{1 - t^{2/(2\alpha+1)}}. \tag{7.4}$$

Note that (7.4) is in complete agreement with the formula (4.4) we found for the case of equally spaced eigenvalues, i.e., $\alpha = 0$.

Remark 7.1. The formula (7.3) is generally valid for measures σ that satisfy the conditions of Theorem 5.1. To find $r(t)$ one has to invert (7.3), which in general will not be possible explicitly. For the special case of an eigenvalue distribution arising from discretization of the Poisson equation in two dimensions, an explicit formula was obtained recently in [3].

Acknowledgments. I thank Walter Van Assche, Marc Van Barel, and Bernhard Beckermann for reading an earlier version of the manuscript and for making valuable comments. I am also grateful to the anonymous referees for their remarks, which helped to improve the manuscript.

REFERENCES

[1] W. Arveson, *C^*-algebras and numerical linear algebra*, J. Funct. Anal., 122 (1994), pp. 333–360.

[2] B. Beckermann, *On a conjecture of E.A. Rakhmanov*, Constr. Approx., 16 (2000), pp. 427–448.

[3] B. Beckermann and A. B. J. Kuijlaars, *Superlinear Convergence of Conjugate Gradients*, manuscript, 1999.

[4] B. Beckermann and E. B. Saff, *The sensitivity of least squares polynomial approximation*, Internat. Ser. Numer. Math. 131, Birkhäuser, Basel, 1999, pp. 1–19.

[5] V. S. Buyarov and E. A. Rakhkmanov, *Families of equilibrium measures with external field on the real axis*, Sb. Math., 190 (1999), pp. 791–802.

[6] J. K. Cullum and R. A. Willoughby, *Lanczos algorithms for large symmetric eigenvalue computations*, Birkhäuser, Boston, 1985.

[7] P. Deift, *Orthogonal Polynomials and Random Matrices: A Riemann-Hilbert Approach*, Courant Lect. Notes Math. 3, Courant Institute, New York, 1999.

[8] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[9] P. D. Dragnev and E. B. Saff, *Constrained energy problems with applications to orthogonal polynomials of a discrete variable*, J. Anal. Math., 72 (1997), pp. 223–259.

[10] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.

[11] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[12] M. Hanke, *Superlinear convergence rates for the Lanczos method applied to elliptic operators*, Numer. Math., 77 (1997), pp. 487–499.

[13] S. Kaniel, *Estimates for some computational techniques in linear algebra*, Math. Comp., 20 (1966), pp. 369–378.

[14] A. B. J. Kuijlaars and P. D. Dragnev, *Equilibrium problems associated with fast decreasing polynomials*, Proc. Amer. Math. Soc., 127 (1999), pp. 1065–1074.

[15] A. B. J. Kuijlaars and E. A. Rakhmanov, *Zero distributions for discrete orthogonal polynomials*, J. Comput. Appl. Math., 99 (1998), pp. 255–274.

[16] A. B. J. Kuijlaars and W. Van Assche, *Extremal polynomials on discrete sets*, Proc. London Math. Soc. (3), 79 (1999), pp. 191–221.

[17] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[18] E. A. Rakhmanov, *Equilibrium measure and the distribution of zeros of the extremal polynomials of a discrete variable*, Mat. Sb., 187 (1996), pp. 109–124 (in Russian); Sb. Math., 187 (1996), pp. 1213–1228 (in English).

[19] T. Ransford, *Potential theory in the complex plane*, Cambridge University Press, Cambridge, UK, 1995.

[20] Y. Saad, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[21] Y. Saad, *Numerical methods for large eigenvalue problems*, Manchester University Press, Manchester, UK, 1992.

[22] E. B. Saff and V. Totik, *Logarithmic Potentials with External Fields*, Springer-Verlag, Berlin, 1997.

[23] G. L. G. Sleijpen and A. van der Sluis, *Further results on the convergence behavior of conjugate-gradients and Ritz values*, Linear Algebra Appl., 246 (1996), pp. 233–278.

[24] A. van der Sluis and H. A. van der Vorst, *The convergence behavior of Ritz values in the presence of close eigenvalues*, Linear Algebra Appl., 88/89 (1987), pp. 651–694.

[25] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 323–341

CONDITION NUMBER AND BACKWARD ERROR FOR THE GENERALIZED SINGULAR VALUE DECOMPOSITION*

JI-GUANG SUN†

Abstract. Let A, B be two matrices having the same number of columns, and let $(A^T, B^T)^T$ have full column rank. Certain normwise condition numbers for a finite generalized singular value (GSV) of the matrix pair $\{A, B\}$ are defined, and explicit expressions of the condition numbers for a simple, nonzero GSV are derived. Moreover, a normwise backward error of $\{A, B\}$ with respect to an approximate GSV and an associated approximate generalized singular vector group is also defined, and a computable formula of the backward error is obtained. The results are illustrated by numerical examples.

Key words. generalized singular value and singular vector, condition number, backward error

AMS subject classifications. 65F15, 65F99

PII. S0895479898348854

1. Preliminaries.

The generalized singular value decomposition (GSVD) of two matrices having the same number of columns is a very useful tool in many matrix computation problems. Applications, numerical methods, perturbation analysis, and variational formulations of the GSVD have been developed during the last twenty years [1, 2, 4, 7, 9, 12, 13, 14, 15, 17, 19, 20, 21, 23, 24, 25]. The GSVD was first proposed by Van Loan [24].

This paper, as a continuation of [19, 20, 21], discusses normwise condition numbers for a finite generalized singular value (GSV), and a normwise backward error for an approximate GSV and an associated approximate generalized singular vector group. It is known that the study of condition numbers and backward errors is an important part of the subject of perturbation theory for matrix computation problems [11, sections 1.5 and 1.6], [22, sections 1.8 and 1.9].

Throughout this paper we use the following notation. $\mathcal{C}^{m\times n}$ denotes the set of $m\times n$ complex matrices, and $\mathcal{C}^m = \mathcal{C}^{m\times 1}$. A^H and $A^\dagger$ denote the conjugate transpose and the Moore–Penrose inverse of a matrix A, respectively. I_n stands for the identity matrix of order n, and $O_{m\times n}$ for the $m\times n$ null matrix. $P_{j,k}^{(n)}$ denotes the permutation matrix which is obtained by permuting the jth and the kth rows and columns of I_n. $\|\cdot\|_2$ denotes the Euclidean vector norm and the spectral norm. The norm $\|\cdot\|_\infty$ denotes the ∞-norm for vectors.

We begin with some definitions and basic results relating the GSVD.

1.1. GSV and GSVD.

Let $A, B \in \mathcal{C}^{n\times n}$. (A, B) is called a regular matrix pair if $\det(A - zB) \not\equiv 0$ for $z \in \mathcal{C}$. A complex number pair $(\mu, \nu) \neq (0, 0)$ is called a generalized eigenvalue of the regular matrix pair (A, B) if $\det(\nu A - \mu B) = 0$. If $\nu \neq 0$, then $\lambda = \mu/\nu$ is a finite generalized eigenvalue; otherwise, (A, B) has the generalized eigenvalue $\lambda = \infty$ [18]. The set of the generalized eigenvalues of a regular matrix pair (A, B) is denoted by $\lambda(A, B)$.

*Received by the editors December 10, 1998; accepted for publication (in revised form) by I. Ipsen on October 12, 1999; published electronically July 11, 2000. This work was supported by the Swedish Natural Science Research Council under Contract M-AA/MA 06952-307 and the Department of Computing Science, Umeå University.

http://www.siam.org/journals/simax/22-2/34885.html

†Department of Computing Science, Umeå University, S-901 87 Umeå, Sweden (jisun@cs.umu.se).

Let $A \in \mathcal{C}^{m\times n}$ and $B \in \mathcal{C}^{p\times n}$. The matrix pair $\{A, B\}$ is an (m, p, n)-Grassmann matrix pair (GMP) if $\operatorname{rank}(A^T, B^T) = n$ [13, 19, 20, 21].

Let $\{A, B\}$ be an (m, p, n)-GMP. A nonnegative number pair $(\alpha, \beta) \neq (0, 0)$ is a generalized singular value (GSV) of the GMP $\{A, B\}$ if

$$(\alpha, \beta) = (\sqrt{\mu}, \sqrt{\nu}), \quad \text{where} \quad (\mu, \nu) \in \lambda(A^H A, B^H B). \tag{1.1}$$

If $\beta \neq 0$, then $\sigma = \alpha/\beta$ is a finite GSV of $\{A, B\}$; otherwise, $\{A, B\}$ has the GSV $\sigma = \infty$ [19, 20, 24]. The set of the GSVs of an (m, p, n)-GMP $\{A, B\}$ is denoted by $\sigma\{A, B\}$.

Let $(\alpha, \beta) \in \sigma\{A, B\}$. Then by (1.1) there is a nonzero $x \in \mathcal{C}^n$ such that

$$\beta^2 A^H A x = \alpha^2 B^H B x. \tag{1.2}$$

The vector x of 1.2 is called a right generalized singular vector of $\{A, B\}$ associated with the GSV (α, β). By (1.2) we may express (α, β) by

$$(\alpha, \beta) = \tau \left(\|Ax\|_2, \ \|Bx\|_2 \right), \tag{1.3}$$

where τ is any positive scalar.

The following form of the GSVD is one of several formulations.

THEOREM 1.1 ((GSVD)[15]). *Let $\{A, B\}$ be an (m, p, n)-GMP. Then there exist unitary matrices $Z \in \mathcal{C}^{m\times m}, W \in \mathcal{C}^{p\times p}$, and a nonsingular matrix $X \in \mathcal{C}^{n\times n}$ such that*

$$Z^H A X = \Sigma_A, \quad W^H B X = \Sigma_B, \tag{1.4}$$

$$\Sigma_A = \begin{pmatrix} D_A & 0 \\ 0 & O_{(m-r-s)\times(n-r-s)} \end{pmatrix}, \quad \Sigma_B = \begin{pmatrix} O_{(p+r-n)\times r} & 0 \\ 0 & D_B \end{pmatrix}, \tag{1.5}$$

where

$$D_A = \operatorname{diag}(\alpha_1, \ldots, \alpha_{r+s}), \quad D_B = \operatorname{diag}(\beta_{r+1}, \ldots, \beta_n) \tag{1.6}$$

with

$$\begin{aligned} &1 = \alpha_1 = \cdots = \alpha_r > \alpha_{r+1} \geq \cdots \geq \alpha_{r+s} > \alpha_{r+s+1} = \cdots = \alpha_n = 0, \\ &0 = \beta_1 = \cdots = \beta_r < \beta_{r+1} \leq \cdots \leq \beta_{r+s} < \beta_{r+s+1} = \cdots = \beta_n = 1, \end{aligned} \tag{1.7}$$

and

$$\alpha_j^2 + \beta_j^2 = 1 \quad \forall j. \tag{1.8}$$

Remark 1.1. Theorem 1.1 implies that the GSVD of the GMP $\{A, B\}$ can also be expressed by

$$Z^H A Q = \Sigma_A R, \quad W^H B Q = \Sigma_B R, \tag{1.9}$$

where the matrices Z, W, Σ_A, Σ_B are as in Theorem 1.1, $Q \in \mathcal{C}^{n\times n}$ is unitary, and $R \in \mathcal{C}^{n\times n}$ is upper triangular and nonsingular.

By the definition of the GSV and its associated right generalized singular vector, we have $\sigma\{A, B\} = \{(\alpha_j, \beta_j)\}_{j=1}^n$ for the (m, p, n)-GMP $\{A, B\}$ in Theorem 1.1, and

every column x_j of the matrix X of (1.4) is a right generalized singular vector of $\{A, B\}$ associated with (α_j, β_j).

Observe that a GSV of $\{A, B\}$ can be regarded as a point on the real projective straight line, that is, for any $(\alpha, \beta) \in \sigma\{A, B\}$ and any $\tau > 0$, the pairs $(\tau\alpha, \tau\beta)$ and (α, β) express the same GSV of $\{A, B\}$. Hence, the distance between two GSVs (α, β) and $(\tilde{\alpha}, \tilde{\beta})$ can be measured in the chordal metric (see, e.g., [6, 18, 19, 20, 21, 22, 23]). But in practice, if we are interested only in finite GSVs, then to use the Euclidean distance $|\tilde{\alpha}/\tilde{\beta} - \alpha/\beta|$ is, probably, more natural and appropriate [8, 10]. In this note we use the Euclidean distance for finite GSVs.

1.2. Generalized singular vector groups. We first recall the singular value decomposition (SVD) of a matrix $A \in \mathcal{C}^{m\times n}$:

$$A = U\Sigma V^H, \tag{1.10}$$

where $U = (u_1, \ldots, u_m) \in \mathcal{C}^{m\times m}$ and $V = (v_1, \ldots, v_n) \in \mathcal{C}^{n\times n}$ are unitary, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots)$. For any integer $k \in [1,\ \min\{m, n\}]$, let

$$\sigma = \sigma_k, \quad v = v_k, \quad u = u_k.$$

Then the SVD (1.10) of A implies that

$$Av = \sigma u, \quad A^H u = \sigma v, \quad \|u\|_2 = \|v\|_2 = 1. \tag{1.11}$$

The vector pair $\{v, u\}$ of (1.11) can be called a singular vector pair of A associated with the singular value σ.

We now extend the singular vector pair concept to the case of the GSVD.

Let $\{A, B\}$ be an (m, p, n)-GMP with the GSVD expressed by (1.4)–(1.8). Write

$$Z = (z_1, \ldots, z_m), \quad W = (w_1, \ldots, w_p), \quad X = (x_1, \ldots, x_n),$$

and

$$Y = X^{-H} = (y_1, \ldots, y_n).$$

For any integer $k \in [\max\{1, n - p + 1\},\ \min\{m, n\}]$, let

$$(\alpha, \beta) = (\alpha_k, \beta_k), \quad x = x_k, \quad y = y_k, \quad z = z_k, \quad w = w_{p-n+k}.$$

Then the GSVD 1.4 of $\{A, B\}$ implies that

$$\begin{aligned} &Ax = \alpha z, \quad Bx = \beta w, \quad A^H z = \alpha y, \quad B^H w = \beta y, \\ &y^H x = 1, \quad \|z\|_2 = \|w\|_2 = 1. \end{aligned} \tag{1.12}$$

The vector group $\{x, y, z, w\}$ of (1.12) is called a generalized singular vector group of $\{A, B\}$ associated with the GSV (α, β).

The paper is organized as follows. In section 2 we define certain normwise condition numbers for a finite GSV and derive explicit expressions of the condition numbers for a simple, nonzero GSV. In section 3 we define a normwise backward error for an (m, p, n)-GMP with respect to an approximate GSV and an associated approximate generalized singular vector group, and obtain a computable formula of the backward error. The results of sections 2 and 3 are illustrated by numerical examples in section 4.

Note that certain connections between the results of this paper and the existing results in the special case of the ordinary SVD have been made by Remarks 2.3 and 3.2, respectively.

2. Condition number. Let $\{A, B\}$ be an (m, p, n)-GMP, and (α, β) be a finite GSV of $\{A, B\}$, that is, $(\alpha, \beta) \in \sigma\{A, B\}$ with $\beta \neq 0$. Suppose that the GMP $\{A, B\}$ is slightly perturbed to $\{\tilde{A}, \tilde{B}\}$ with $\tilde{A} = A + E$ and $\tilde{B} = B + F$, and (α, β) is correspondingly perturbed to $(\tilde{\alpha}, \tilde{\beta})$. Let

$$\sigma = \alpha/\beta, \qquad \tilde{\sigma} = \tilde{\alpha}/\tilde{\beta}.$$

Then referring to [8, 10] and [16] we may define the condition number $c(\sigma)$ for σ by

$$c(\sigma) = \lim_{\delta \to 0} \sup_{\left\| \left(\frac{\|E\|_2}{\gamma_{\rm A}}, \frac{\|F\|_2}{\gamma_{\rm B}} \right)^T \right\|_\infty \leq \delta} \frac{|\tilde{\sigma} - \sigma|}{\xi \delta}, \tag{2.1}$$

where $\gamma_{\rm A}$, $\gamma_{\rm B}$, and ξ are positive parameters. If one is interested in the sensitivity of σ to small perturbations in each individual member of A and B, then by [22, section 4.2.1] we may define the partial condition numbers $c_A(\sigma)$ and $c_B(\sigma)$ for σ as

$$c_A(\sigma) = \lim_{\delta \to 0} \sup_{\frac{\|E\|_2}{\gamma_{\rm A}} \leq \delta,\ F=0} \frac{|\tilde{\sigma} - \sigma|}{\xi \delta}, \qquad c_B(\sigma) = \lim_{\delta \to 0} \sup_{E=0,\ \frac{\|F\|_2}{\gamma_{\rm B}} \leq \delta} \frac{|\tilde{\sigma} - \sigma|}{\xi \delta}. \tag{2.2}$$

Taking $\gamma_{\rm A} = \gamma_{\rm B} = \xi = 1$ in (2.1) and (2.2), we get the absolute condition numbers $c_{\rm abs}(\sigma)$, $c_A^{(\rm abs)}(\sigma)$, $c_B^{(\rm abs)}(\sigma)$; and taking $\gamma_{\rm A} = \|A\|_2$, $\gamma_{\rm B} = \|B\|_2$, and $\xi = \sigma$ (if $\sigma > 0$), we get the relative condition numbers $c_{\rm rel}(\sigma)$, $c_A^{(\rm rel)}(\sigma)$, $c_B^{(\rm rel)}(\sigma)$, respectively.

From the definitions (2.1) and (2.2) we see that $c(\sigma)$ is a measure of the sensitivity of σ to small perturbations in $\{A, B\}$, and $c_A(\sigma)$ and $c_B(\sigma)$ are measures of the sensitivity of σ to small perturbations in A and B, separately. Moreover, in first order approximation the inequalities

$$\frac{|\tilde{\sigma} - \sigma|}{\xi} \leq c(\sigma) \left\| \left(\frac{\|E\|_2}{\gamma_{\rm A}}, \frac{\|F\|_2}{\gamma_{\rm B}} \right)^T \right\|_\infty \tag{2.3}$$

and

$$\frac{|\tilde{\sigma} - \sigma|}{\xi} \leq c_A(\sigma) \frac{\|E\|_2}{\gamma_{\rm A}}, \qquad \frac{|\tilde{\sigma} - \sigma|}{\xi} \leq c_B(\sigma) \frac{\|F\|_2}{\gamma_{\rm B}}$$

hold.

The following result gives explicit expressions of the condition numbers $c(\sigma)$, $c_A(\sigma)$, and $c_B(\sigma)$ for any simple, finite, nonzero GSV (α, β).

THEOREM 2.1. *Let (α, β) be a simple, finite, nonzero GSV of an (m, p, n)-GMP $\{A, B\}$, and let $x \in \mathcal{C}^n$ be an associated right generalized singular vector. Let $\sigma = \alpha/\beta$. Then the condition numbers $c(\sigma)$, $c_A(\sigma)$, and $c_B(\sigma)$ can be expressed by*

$$c(\sigma) = \frac{\|x\|_2 \left(\gamma_{_B} \|Ax\|_2 + \gamma_{_A} \|Bx\|_2 \right)}{\xi \|Bx\|_2^2} \tag{2.4}$$

and

$$c_A(\sigma) = \frac{\gamma_{_A} \|x\|_2}{\xi \|Bx\|_2}, \qquad c_B(\sigma) = \frac{\gamma_{_B} \|Ax\|_2 \|x\|_2}{\xi \|Bx\|_2^2}. \tag{2.5}$$

We first make two remarks before we give a proof of Theorem 2.1.

1. Observe that if $\{A, B\}$ is an (m, p, n)-GMP with $p < n$, then $\{A, B'\}$ with

$$B' = \begin{pmatrix} O_{(n-p)\times n} \\ B \end{pmatrix} \tag{2.6}$$

is an (m, n, n)-GMP, and

$$\sigma\{A, B'\} = \sigma\{A, B\}, \qquad \|B'x\|_2 = \|Bx\|_2.$$

Hence, without loss of generality we may assume $p \geq n$ in Theorem 2.1. It is worth pointing out that by the definitions (2.1) and (2.2) we consider only the perturbations $\{E, F'\}$ with

$$F' = \begin{pmatrix} O_{(n-p)\times n} \\ F \end{pmatrix} \tag{2.7}$$

in $\{A, B'\}$ when $\{A, B\}$ is an (m, p, n)-GMP with $p < n$, and we use the (m, n, n)-GMP $\{A, B'\}$ to replace $\{A, B\}$, where B' is the matrix of (2.6), and $\{E, F\}$ denotes any perturbation in $\{A, B\}$.

2. Let $\{A, B\}$ be an (m, p, n)-GMP with $p \geq n$, and let (α, β) be a simple, finite, nonzero GSV of $\{A, B\}$. The GSVD of $\{A, B\}$ expressed by (1.4)–(1.8) implies that $(\alpha, \beta) = (\alpha_k, \beta_k)$ for a certain integer $k \in [1, n]$. If the matrices Z, W, and X of (1.4) are replaced by $ZP^{(m)}_{1,k}, WP^{(p)}_{p-n+1,k}$, and $XP^{(n)}_{1,k}$, respectively, then the matrices Σ_A and Σ_B become

$$\begin{pmatrix} \alpha & 0 \\ 0 & A_2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} O_{(p-n)\times 1} & 0 \\ \beta & 0 \\ 0 & B_2 \end{pmatrix},$$

where A_2 and B_2 are $(m-1)\times(n-1)$ and $(n-1)\times(n-1)$ matrices, respectively. Consequently, without loss of generality we may assume that the GSVD of the GMP $\{A, B\}$ of Theorem 2.1 is in the following form:

$$Z^H AX = \begin{pmatrix} \alpha & 0 \\ 0 & A_2 \end{pmatrix}, \qquad W^H BX = \begin{pmatrix} O_{(p-n)\times 1} & 0 \\ \beta & 0 \\ 0 & B_2 \end{pmatrix}, \tag{2.8}$$

where $X = (x, X_2)$.

Proof of Theorem 2.1. Let $(\tilde{\alpha}, \tilde{\beta})$ be the corresponding perturbation of (α, β) when $\{A, B\}$ is slightly perturbed to $\{\tilde{A}, \tilde{B}\}$ with $\tilde{A} = A + E$ and $\tilde{B} = B + F$. Let $\tilde{\sigma} = \tilde{\alpha}/\tilde{\beta}$. We now prove the theorem by the following three steps.

1. Apply [21, Theorem 4.1] to prove the conclusion: There is a vector $s_* \in \mathcal{C}^{n-1}$ satisfying

$$\|s_*\|_2 = O\left(\left\|\begin{pmatrix} E \\ F \end{pmatrix}\right\|_2\right) \quad \text{as} \quad \left\|\begin{pmatrix} E \\ F \end{pmatrix}\right\|_2 \to 0 \tag{2.9}$$

such that the vector

$$\tilde{x} = X\begin{pmatrix} 1 \\ s_* \end{pmatrix} = x + X_2 s_* \tag{2.10}$$

is a right generalized singular vector of $\{\tilde{A}, \tilde{B}\}$ associated with $\tilde{\sigma}$. The proof is as follows. Since the GSV (α, β) is simple, the matrix T of [21, (1.30)] with

$$\Sigma_{a1} = \alpha \quad \text{and} \quad \Sigma_{b1} = \begin{pmatrix} O_{(p-n)\times 1} \\ \beta \end{pmatrix}$$

is nonsingular, and so we have the positive constants l_j $(j = 1, \ldots, 4)$ and l which are defined by using the spectral norms of the submatrices L_j $(j = 1, \ldots, 4)$ of T^{-1} (see [21, (1.34) and (4.2)]). By the proof of [21, Theorem 4.1], under the condition [21, (4.6)]

$$l(2\gamma + \epsilon) < 1, \tag{2.11}$$

there is a vector $s_* \in \mathcal{C}^{n-1}$ satisfying [21, (4.19)]

$$\|s_*\|_2 \le \rho_3 \tag{2.12}$$

such that the vector $\tilde{x}$ expressed by (2.10) is a right generalized singular vector of $\{\tilde{A}, \tilde{B}\}$ associated with $\tilde{\sigma}$, where γ, ϵ, and ρ_3 are positive scalars and satisfy

$$\begin{aligned} &\gamma \le \left\|\left(E^T, F^T\right)\right\|_F \qquad \text{(by [21, (4.3)])}, \\ &\epsilon \le \sqrt{2}\left\|\left(E^T, F^T\right)\right\|_F \qquad \text{(by [21, (4.4)])}, \end{aligned} \tag{2.13}$$

and

$$\rho_3 \le \frac{2l_3\gamma}{1 - l\epsilon} \qquad \text{(by [21, (4.5)])}. \tag{2.14}$$

From (2.13) we see that if $\left\|\left(E^T, F^T\right)\right\|_F$ is sufficiently small, then the condition (2.11) holds, and from (2.12)–(2.14) we get

$$\begin{aligned} \|s_*\|_2 \quad &\le \frac{2l_3\left\|\left(E^T, F^T\right)\right\|_F}{1 - \sqrt{2}l\left\|(E^T, F^T)\right\|_F} \\ &= O\left(\left\|\begin{pmatrix} E \\ F \end{pmatrix}\right\|_2\right) \quad \text{as} \quad \left\|\begin{pmatrix} E \\ F \end{pmatrix}\right\|_2 \to 0. \end{aligned}$$

Thus, our conclusion is proved.

It is easy to see that the relations (2.9) and (2.10) can be written as

$$\begin{aligned} \tilde{x} = x + h \quad \text{with} \qquad &\|h\|_2 = O\left(\left\|\left(\frac{\|E\|_2}{\gamma_{\text{A}}}, \frac{\|F\|_2}{\gamma_{\text{B}}}\right)^T\right\|_\infty\right) \\ &\text{as} \quad \left\|\left(\frac{\|E\|_2}{\gamma_{\text{A}}}, \frac{\|F\|_2}{\gamma_{\text{B}}}\right)^T\right\|_\infty \to 0. \end{aligned} \tag{2.15}$$

2. Prove (2.4). From (2.15) it follows that for any $\{\tilde{A}, \tilde{B}\} = \{A + E, B + F\}$ satisfying

$$\left\|\left(\frac{\|E\|_2}{\gamma_{\text{A}}}, \frac{\|F\|_2}{\gamma_{\text{B}}}\right)^T\right\|_\infty \le \delta,$$

we have

$$\begin{aligned}
\|\tilde{A}\tilde{x}\|_2 &= \|(A+E)(x+h)\|_2 \\
&= \left[(Ax+Ex+Ah+Eh)^H(Ax+Ex+Ah+Eh)\right]^{1/2} \\
&= \left[\|Ax\|_2^2 + x^HA^HEx + x^HA^HAh + x^HE^HAx + h^HA^HAx + O(\delta^2)\right]^{1/2} \\
&= \|Ax\|_2\left(1 + \frac{x^HA^HEx + x^HA^HAh + x^HE^HAx + h^HA^HAx}{\|Ax\|_2^2} + O(\delta^2)\right)^{1/2} \\
&= \|Ax\|_2 + \frac{x^HA^HEx + x^HA^HAh + x^HE^HAx + h^HA^HAx}{2\|Ax\|_2} \\
&\quad + O(\delta^2) \qquad \text{as } \delta \to 0,
\end{aligned}$$

and similarly,

$$\begin{aligned}
\|\tilde{B}\tilde{x}\|_2 &= \|Bx\|_2 + \frac{x^HB^HFx + x^HB^HBh + x^HF^HBx + h^HB^HBx}{2\|Bx\|_2} \\
&\quad + O(\delta^2) \qquad \text{as } \delta \to 0.
\end{aligned}$$

Substituting the above expressions of $\|\tilde{A}\tilde{x}\|_2$ and $\|\tilde{B}\tilde{x}\|_2$ into

$$|\tilde{\sigma} - \sigma| = \left|\frac{\|\tilde{A}\tilde{x}\|_2}{\|\tilde{B}\tilde{x}\|_2} - \frac{\|Ax\|_2}{\|Bx\|_2}\right|,$$

gives

$$\begin{aligned}
\frac{|\tilde{\sigma}-\sigma|}{\xi\delta} &= \frac{1}{2\xi\delta\|Bx\|_2^2}\left|\left(\gamma_{\text{A}}\frac{\|Bx\|_2}{\|Ax\|_2}x^HA^H,\ -\gamma_{\text{B}}\frac{\|Ax\|_2}{\|Bx\|_2}x^HB^H\right)\begin{pmatrix}\frac{E}{\gamma_{\text{A}}}\\ \frac{F}{\gamma_{\text{B}}}\end{pmatrix}x\right. \\
&\quad \left. + x^H\left(\frac{E^H}{\gamma_{\text{A}}},\ \frac{F^H}{\gamma_{\text{B}}}\right)\begin{pmatrix}\gamma_{\text{A}}\frac{\|Bx\|_2}{\|Ax\|_2}Ax\\ -\gamma_{\text{B}}\frac{\|Ax\|_2}{\|Bx\|_2}Bx\end{pmatrix}\right| + O(\delta) \qquad (2.16) \\
&\le \frac{\|x\|_2\left(\gamma_{\text{B}}\|Ax\|_2 + \gamma_{\text{A}}\|Bx\|_2\right)}{\xi\|Bx\|_2^2} + O(\delta),
\end{aligned}$$

which implies

$$c(\sigma) \le \frac{\|x\|_2\left(\gamma_{\text{B}}\|Ax\|_2 + \gamma_{\text{A}}\|Bx\|_2\right)}{\xi\|Bx\|_2^2}. \qquad (2.17)$$

On the other hand, take the special perturbation $\{\hat{E}, \hat{F}\}$ with

$$\hat{E} = \frac{\delta\gamma_{\text{A}}Axx^H}{\|Ax\|_2\|x\|_2}, \qquad \hat{F} = -\frac{\delta\gamma_{\text{B}}Bxx^H}{\|Bx\|_2\|x\|_2}, \qquad (2.18)$$

and let $(\hat{\alpha}, \hat{\beta})$ be the corresponding perturbation of (α, β). Then

$$\left\| \left(\frac{\|\hat{E}\|_2}{\gamma_{\text{A}}}, \frac{\|\hat{F}\|_2}{\gamma_{\text{B}}} \right)^T \right\|_\infty = \delta,$$

and for $\hat{\sigma} = \hat{\alpha}/\hat{\beta}$ we have

$$\frac{|\hat{\sigma} - \sigma|}{\xi\delta} = \frac{\|x\|_2 \left(\gamma_{\text{B}}\|Ax\|_2 + \gamma_{\text{A}}\|Bx\|_2\right)}{\xi\|Bx\|_2^2} + O(\delta).$$

Combining it with the relation (2.17) and the definition (2.1) shows (2.4).

3. Prove (2.5). By the first relation of (2.16) we have

$$\frac{|\tilde{\sigma} - \sigma|}{\xi\delta} \le \frac{\gamma_{\text{A}}\|x\|_2}{\xi\|Bx\|_2} + O(\delta) \quad \text{as} \quad \frac{\|E\|_2}{\gamma_{\text{A}}} \le \delta \to 0 \quad \text{and} \quad F = 0, \tag{2.19}$$

and the equalities in (2.19) are achieved for the specific perturbation $\{\hat{E}, \hat{F}\}$ with

$$\hat{E} = \frac{\delta\gamma_{\text{A}} A x x^H}{\|Ax\|_2\|x\|_2}, \qquad \hat{F} = 0.$$

Moreover, we have

$$\frac{|\tilde{\sigma} - \sigma|}{\xi\delta} \le \frac{\gamma_{\text{B}}\|Ax\|_2\|x\|_2}{\xi\|Bx\|_2^2} + O(\delta) \quad \text{as} \quad E = 0 \quad \text{and} \quad \frac{\|F\|_2}{\gamma_{\text{B}}} \le \delta \to 0, \tag{2.20}$$

and the equalities in (2.20) are achieved for the specific perturbation $\{\hat{E}, \hat{F}\}$ with

$$\hat{E} = 0, \qquad \hat{F} = -\frac{\delta\gamma_{\text{B}} B x x^H}{\|Bx\|_2\|x\|_2}. \tag{2.21}$$

Combining these facts with the definition (2.2) shows (2.5). □

Note that in the case of $p < n$, we use $\{A, B'\}$ with the matrix B' of (2.6) to replace $\{A, B\}$. In such a case, the matrix $\hat{F}$ of (2.18) and (2.21) will be replaced by

$$\hat{F}' = -\frac{\delta\gamma_{\text{B}} B' x x^H}{\|B'x\|_2\|x\|_2} = \begin{pmatrix} O_{(n-p)\times n} \\ \hat{F} \end{pmatrix},$$

which is just in the form of (2.7). Consequently, the above-mentioned proof of Theorem 2.1 is still valid.

Remark 2.1. From (1.3), (2.4), (2.5), and $\sigma = \alpha/\beta$ we get the following expressions:

$$\begin{aligned} c_{\text{abs}}(\sigma) &= \frac{\|x\|_2 \left(\|Ax\|_2 + \|Bx\|_2\right)}{\|Bx\|_2^2} = \frac{\|x\|_2(1+\sigma)}{\|Bx\|_2}, \\ c_A^{(\text{abs})}(\sigma) &= \frac{\|x\|_2}{\|Bx\|_2}, \qquad c_B^{(\text{abs})}(\sigma) = \frac{\sigma\|x\|_2}{\|Bx\|_2}, \end{aligned} \tag{2.22}$$

and

$$\begin{aligned} c_{\text{rel}}(\sigma) &= \frac{\|x\|_2 \left(\|B\|_2\|Ax\|_2 + \|A\|_2\|Bx\|_2\right)}{\sigma\|Bx\|_2^2} = \frac{\|x\|_2(\|A\|_2 + \sigma\|B\|_2)}{\sigma\|Bx\|_2}, \\ c_A^{(\text{rel})}(\sigma) &= \frac{\|A\|_2\|x\|_2}{\sigma\|Bx\|_2}, \qquad c_B^{(\text{rel})}(\sigma) = \frac{\|B\|_2\|x\|_2}{\|Bx\|_2}, \end{aligned} \tag{2.23}$$

respectively.

Remark 2.2. It is known [9, 18] that an $n \times n$ matrix pair (A, B) is said to be a definite pair if the matrices A and B are Hermitian, and

$$\min\left\{\left((x^HAx)^2 + (x^HBx)^2\right)^{1/2} : x \in \mathcal{C}^n, \|x\|_2 = 1\right\}$$

is positive. From the definition we see that if $\{A, B\}$ is a GMP, then (A^HA, B^HB) is a definite pair, and by the definition (1.1) we have

$$(\alpha, \beta) \in \sigma\{A, B\} \iff (\alpha^2, \beta^2) \in \lambda(A^HA, B^HB).$$

We are now going to make a comparison between the condition numbers expressed by (2.22) and (2.23) for a simple, finite, nonzero GSV σ and the condition numbers for a simple, finite, nonzero eigenvalue λ of a definite pair (A, B).

Let (A, B) be a definite pair, and λ be a finite eigenvalue of (A, B). Suppose that the definite pair (A, B) is slightly perturbed to a definite pair $(\tilde{A}, \tilde{B})$ with $\tilde{A} = A + E$ and $\tilde{B} = B + F$, and λ is correspondingly perturbed to $\tilde{\lambda}$. Then by [8, section 3], [10, section 2.2], and [22, section 4.2.1], we may define the condition number $K(\lambda)$ for λ by

$$K(\lambda) = \lim_{\delta\to 0} \sup_{\left\|\left(\frac{\|E\|_2}{\gamma_A}, \frac{\|F\|_2}{\gamma_B}\right)^T\right\|_\infty \le \delta} \frac{|\tilde{\lambda} - \lambda|}{\xi\delta},$$

where γ_A, γ_B, and ξ are positive parameters. Moreover, we may define the partial condition numbers $K_A(\lambda)$ and $K_B(\lambda)$ by

$$K_A(\lambda) = \lim_{\delta\to 0} \sup_{\frac{\|E\|_2}{\gamma_A} \le \delta,\ F=0} \frac{|\tilde{\lambda} - \lambda|}{\xi\delta}, \qquad K_B(\lambda) = \lim_{\delta\to 0} \sup_{E=0,\ \frac{\|F\|_2}{\gamma_B} \le \delta} \frac{|\tilde{\lambda} - \lambda|}{\xi\delta}.$$

Taking $\gamma_A = \gamma_B = \xi = 1$, we get the absolute condition numbers $K_{\mathrm{abs}}(\lambda)$, $K_A^{(\mathrm{abs})}(\lambda)$, $K_B^{(\mathrm{abs})}(\lambda)$; and taking $\gamma_A = \|A\|_2$, $\gamma_B = \|B\|_2$, and $\xi = |\lambda|$ (if $\lambda \neq 0$), we get the relative condition numbers $K_{\mathrm{rel}}(\lambda)$, $K_A^{(\mathrm{rel})}(\lambda)$, $K_B^{(\mathrm{rel})}(\lambda)$.

By [10, Theorem 2.5, Lemma 2.6, and their proofs] or using the technique described in the proof of Theorem 2.1, we can prove that if λ is a simple, nonzero eigenvalue, and x is an associated eigenvector, then we have the following expressions:

$$K_{\mathrm{abs}}(\lambda) = \frac{\|x\|_2^2(1 + |\lambda|)}{|x^HBx|}, \quad K_A^{(\mathrm{abs})}(\lambda) = \frac{\|x\|_2^2}{|x^HBx|}, \quad K_B^{(\mathrm{abs})}(\lambda) = \frac{|\lambda|\|x\|_2^2}{|x^HBx|},$$

and

$$K_{\mathrm{rel}}(\lambda) = \frac{\|x\|_2^2(\|A\|_2 + |\lambda|\|B\|_2)}{|\lambda||x^HBx|}, \quad K_A^{(\mathrm{rel})}(\lambda) = \frac{\|A\|_2\|x\|_2^2}{|\lambda||x^HBx|}, \quad K_B^{(\mathrm{rel})}(\lambda) = \frac{\|B\|_2\|x\|_2^2}{|x^HBx|}.$$

Obviously, these expressions appear similar to those of (2.22) and (2.23), respectively.

Remark 2.3. It may be asked, Does there exist some connection of our results (2.22) and (2.23) with the existing results of the singular value decomposition? The answer is positive. Let $\sigma > 0$ be a singular value of $A \in \mathcal{C}^{m\times n}$. Let $\tilde{A} = A + E$ be

a perturbation of A, and $\tilde{\sigma}$ be the corresponding perturbation of σ. Then we may define the condition number $\kappa(\sigma)$ for σ as

$$\kappa(\sigma) = \lim_{\delta \to 0} \sup_{\frac{\|E\|_2}{\gamma_{\rm A}} \le \delta} \frac{|\tilde{\sigma} - \sigma|}{\xi \delta},$$

where $\gamma_{\rm A}$ and ξ are positive parameters. Taking $\gamma_{\rm A} = \xi = 1$ gives the absolute condition number $\kappa_{\rm abs}(\sigma)$, and taking $\gamma_{\rm A} = \|A\|_2$ and $\xi = \sigma$ (if $\sigma > 0$) gives the relative condition number $\kappa_{\rm rel}(\sigma)$. It is well known that

$$\kappa_{\rm abs}(\sigma) = 1, \quad \kappa_{\rm rel}(\sigma) = \frac{\|A\|_2}{\sigma}. \tag{2.24}$$

We now regard the matrix A as an (m, n, n)-GMP $\{A, I_n\}$. If σ is a simple, nonzero singular value of A, then it is also a simple, finite, nonzero GSV of the GMP $\{A, I_n\}$. Observe that in this case only small perturbations in the matrix A of the GMP $\{A, I_n\}$ should be considered. Hence, it is natural to compare the condition numbers $\kappa_{\rm abs}(\sigma)$ and $\kappa_{\rm rel}(\sigma)$ of (2.24) with the partial condition numbers $c_A^{(\rm abs)}(\sigma)$ and $c_A^{(\rm rel)}(\sigma)$. By (2.22) and (2.23) we have

$$c_A^{(\rm abs)}(\sigma) = 1, \quad c_A^{(\rm rel)}(\sigma) = \frac{\|A\|_2}{\sigma},$$

which just coincide with $\kappa_{\rm abs}(\sigma)$ and $\kappa_{\rm rel}(\sigma)$, respectively.

Remark 2.4. Let (α, β) be a simple, finite, nonzero GSV of $\{A, B\}$, and $x \in \mathcal{C}^n$ be an associated right generalized singular vector. Let $\sigma = \alpha/\beta$. Then (1.3) implies that if

$$\alpha^2 + \beta^2 = 1 \quad \text{and} \quad \|Ax\|_2^2 + \|Bx\|_2^2 = 1, \tag{2.25}$$

then $(\alpha, \beta) = (\|Ax\|_2, \|Bx\|_2)$, and from (2.22) and (2.23) we see that the condition numbers can be expressed in the simpler forms:

$$\begin{aligned}
&c_{\rm abs}(\sigma) = \frac{\|x\|_2}{\beta}(1 + \sigma), \\
&c_A^{(\rm abs)}(\sigma) = \frac{\|x\|_2}{\beta}, \qquad c_B^{(\rm abs)}(\sigma) = \frac{\sigma \|x\|_2}{\beta}, \\
&c_{\rm rel}(\sigma) = \|x\|_2 \left(\frac{\|A\|_2}{\alpha} + \frac{\|B\|_2}{\beta} \right), \\
&c_A^{(\rm rel)}(\sigma) = \frac{\|A\|_2 \|x\|_2}{\alpha}, \qquad c_B^{(\rm rel)}(\sigma) = \frac{\|B\|_2 \|x\|_2}{\beta}.
\end{aligned} \tag{2.26}$$

Note that if $\{A, B\}$ has the GSVD expressed by (1.4)–(1.8), then every GSV (α_j, β_j) and the corresponding column x_j of X in the GSVD satisfy (2.25).

Remark 2.5. By a comment from the referees, we now take example by the formulas of $c_{\rm abs}(\sigma)$ and $c_{\rm rel}(\sigma)$ in (2.26) to give a short discussion and interpretation. Obviously, the absolute condition number $c_{\rm abs}(\sigma)$ may be large if β is small and/or σ is large or if $\|x\|_2$ is large. The former may happen when B is nearly singular. (See $c_{\rm abs}(\sigma_2)$ of Example 4.1 in section 4, where B is nearly singular.) The latter may

happen when the unit vector $x/\|x\|_2$ closes to a null-vector of both A and B; to see this, note that $\alpha = \|Ax\|_2$ and $\beta = \|Bx\|_2$ cannot both be small at the same time because $\alpha^2 + \beta^2 = 1$, so the fact that both $\|A\frac{x}{\|x\|_2}\|_2$ and $\|B\frac{x}{\|x\|_2}\|_2$ are small implies that $\|x\|_2$ is large. Rewrite the formula of $c_{\rm rel}(\sigma)$ as

$$c_{\rm rel}(\sigma) = \frac{\|A\|_2}{\left\|A\frac{x}{\|x\|_2}\right\|_2} + \frac{\|B\|_2}{\left\|B\frac{x}{\|x\|_2}\right\|_2},$$

which shows that the relative condition number may be large if the unit vector $x/\|x\|_2$ closes to a null-vector of A and/or B. (See $c_{\rm rel}(\sigma_2)$ of Example 4.1 in section 4, where $\|B\frac{x_2}{\|x_2\|_2}|_2 = (\frac{\|x_2\|_2}{\beta_2})^{-1} = 10^{-6}$; that is, the unit vector $x_2/\|x_2\|_2$ closes to a null-vector of B.)

The following fact is pointed out by a referee. With the definition of the GSVD in (1.4)–(1.8), the GSVs do not scale with $\begin{pmatrix} A \\ B \end{pmatrix}$, that is, the GSVs of $\{A, B\}$ and $\{\tau A, \tau B\}$ with any nonzero scalar τ are the same. The only place the scaling can go is in the matrix X, and therefore, by the formulas of $c_{\rm abs}(\sigma)$, $c_A^{(\rm abs)}(\sigma)$, and $c_B^{(\rm abs)}(\sigma)$ in (2.26), a scaling of A and B changes the absolute condition numbers; however, by the formulas of $c_{\rm rel}(\sigma)$, $c_A^{(\rm rel)}(\sigma)$, and $c_B^{(\rm rel)}(\sigma)$ in (2.26), the relative condition numbers are insensitive to the scaling, because X is scaled by τ^{-1} when $\begin{pmatrix} A \\ B \end{pmatrix}$ is scaled by τ.

Remark 2.6. Let (α, β) be a GSV of a GMP $\{A, B\}$. Since the definition of any normwise condition number for the GSV (α, β) is dependent on the metrics which are used to measure perturbations in $\{A, B\}$ and perturbations in (α, β), there are different condition numbers from different geometrical points of view: Euclidean, non-Euclidean, or mixed. In the paper [23], several condition numbers for (α, β) in different metrics are defined, computable formulas of the condition numbers are obtained, and comparisons between the different condition numbers are made. Note that the different condition numbers have different implications.

3. Backward error. Let $(\tilde\alpha, \tilde\beta) \neq (0, 0)$ with $\tilde\alpha, \tilde\beta \geq 0$ be an approximate GSV of an (m, p, n)-GMP $\{A, B\}$, and $\{\tilde x, \tilde y, \tilde z, \tilde w\}$ be an associated approximate generalized singular vector group, that is, the vectors $\tilde x, \tilde y \in \mathcal{C}^n$, $\tilde z \in \mathcal{C}^m$, and $\tilde w \in \mathcal{C}^p$ satisfy

$$\begin{aligned} &A\tilde x \approx \tilde\alpha\tilde z, \quad B\tilde x \approx \tilde\beta\tilde w, \quad A^H\tilde z \approx \tilde\alpha\tilde y, \quad B^H\tilde w \approx \tilde\beta\tilde y, \\ &\tilde y^H\tilde x = 1, \quad \|\tilde z\|_2 = \|\tilde w\|_2 = 1. \end{aligned} \tag{3.1}$$

Moreover, define the set $\mathcal{G}$ by

$$\mathcal{G} = \left\{ \begin{pmatrix} E \\ F \end{pmatrix} : E \in \mathcal{C}^{m\times n}, F \in \mathcal{C}^{p\times n}, \begin{array}{l} (A+E)\tilde x = \tilde\alpha\tilde z,\ (B+F)\tilde x = \tilde\beta\tilde w, \\ (A+E)^H\tilde z = \tilde\alpha\tilde y,\ (B+F)^H\tilde w = \tilde\beta\tilde y \end{array} \right\}, \tag{3.2}$$

and define the backward error $\eta((\tilde\alpha, \tilde\beta); \tilde x, \tilde y, \tilde z, \tilde w)$ of $\{A, B\}$ with respect to the approximate solution $\{(\tilde\alpha, \tilde\beta); \tilde x, \tilde y, \tilde z, \tilde w\}$ by

$$\eta((\tilde\alpha, \tilde\beta); \tilde x, \tilde y, \tilde z, \tilde w) = \min_{\begin{pmatrix} E \\ F \end{pmatrix} \in \mathcal{G}} \left\| \left(\frac{\|E\|_2}{\gamma_{\rm A}}, \frac{\|F\|_2}{\gamma_{\rm B}} \right)^T \right\|_\infty, \tag{3.3}$$

where $\gamma_{\rm A}$ and $\gamma_{\rm B}$ are positive parameters. Taking $\gamma_{\rm A} = \gamma_{\rm B} = 1$, we get the absolute backward error $\eta_{\rm abs}((\tilde\alpha, \tilde\beta); \tilde x, \tilde y, \tilde z, \tilde w)$; and taking $\gamma_{\rm A} = \|A\|_2$ and $\gamma_{\rm B} = \|B\|_2$, we get the relative backward error $\eta_{\rm rel}((\tilde\alpha, \tilde\beta); \tilde x, \tilde y, \tilde z, \tilde w)$.

The definition (3.3) is obviously consistent with the definition (2.1).

From the definition of $\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w})$ we see that a small $\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w})$ means that the approximate GSV $(\tilde{\alpha},\tilde{\beta})$ and associated approximate generalized singular vector group $\{\tilde{x},\tilde{y},\tilde{z},\tilde{w}\}$ are the exact GSV and associated generalized singular vector group of a slightly perturbed $\{\tilde{A},\tilde{B}\}$ of $\{A,B\}$. Consequently, a computable formula of $\eta(\cdot;\cdot)$ may be useful for assessing the numerical quality of a computed GSVD, and for testing the backward stability of algorithms for the computation of the GSVD.

The following result gives a computable formula of the backward error $\eta(\cdot;\cdot)$.

THEOREM 3.1. *Let*

$$\begin{aligned} r_1 &= \tilde{\alpha}\tilde{z} - A\tilde{x}, \qquad r_2 = \tilde{\beta}\tilde{w} - B\tilde{x}, \\ r_3 &= \tilde{\alpha}\tilde{y} - A^H\tilde{z}, \qquad r_4 = \tilde{\beta}\tilde{y} - B^H\tilde{w} \end{aligned} \tag{3.4}$$

be the residuals of $\{A,B\}$ with respect to $(\tilde{\alpha},\tilde{\beta})$ and $\{\tilde{x},\tilde{y},\tilde{z},\tilde{w}\}$, where $(\tilde{\alpha},\tilde{\beta}) \neq (0,0)$ with $\tilde{\alpha},\tilde{\beta} \geq 0$, and $\tilde{x},\tilde{y},\tilde{z},\tilde{w}$ satisfy (3.1). *Then the backward error $\eta(\cdot;\cdot)$ can be expressed by*

$$\begin{aligned} &\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w}) \\ &= \max\left\{ \frac{1}{\gamma_{_A}} \max\left\{ \frac{\|r_1\|_2}{\|\tilde{x}\|_2},\ \|r_3\|_2 \right\},\ \frac{1}{\gamma_{_B}} \max\left\{ \frac{\|r_2\|_2}{\|\tilde{x}\|_2},\ \|r_4\|_2 \right\} \right\}. \end{aligned} \tag{3.5}$$

The proof of Theorem 3.1 requires some preliminary theorems (Theorems 3.2 and 3.3).

THEOREM 3.2. *Let $A \in \mathcal{C}^{k\times m}, B \in \mathcal{C}^{n\times l}$, and $C \in \mathcal{C}^{k\times l}$ be given. Define the set $\mathcal{E}$ by*

$$\mathcal{E} = \{E \in \mathcal{C}^{m\times n} \ :\ AEB = C\}.$$

Then $\mathcal{E} \neq \emptyset$ (the empty set) if and only if A, B, and C satisfy

$$P_A C P_{B^H} = C, \tag{3.6}$$

and in the case of $\mathcal{E} \neq \emptyset$, any $E \in \mathcal{E}$ can be expressed by

$$E = A^\dagger C B^\dagger + Z - P_{A^H} Z P_B, \quad Z \in \mathcal{C}^{m\times n},$$

where $P_A = AA^\dagger$ is the orthogonal projection onto the column subspace of A.

Proof. The relation (3.6) is obviously a necessary condition for $\mathcal{E} \neq \emptyset$. We now define the set $\mathcal{F}$ by

$$\mathcal{F} = \{A^\dagger C B^\dagger + Z - P_{A^H} Z P_B \ :\ Z \in \mathcal{C}^{m\times n}\},$$

and prove that $\mathcal{E} = \mathcal{F}$ under the condition (3.6).

Assume $E \in \mathcal{E}$. Then we may represent the matrix E as

$$E = A^\dagger C B^\dagger + E - P_{A^H} E P_B.$$

This means that there exists a matrix $Z\ (= E) \in \mathcal{C}^{m\times n}$ such that the matrix $E \in \mathcal{E}$ can be expressed by

$$E = A^\dagger C B^\dagger + Z - P_{A^H} Z P_B \in \mathcal{F}. \tag{3.7}$$

Thus, $\mathcal{E} \subset \mathcal{F}$.

Conversely, assume $E \in \mathcal{F}$, and let E be expressed by (3.7) with some $Z \in \mathcal{C}^{m\times n}$. Then the expression (3.7) and the condition (3.6) imply $AEB = C$, that is, $E \in \mathcal{E}$. Thus, $\mathcal{F} \subset \mathcal{E}$. Consequently, we have $\mathcal{E} = \mathcal{F}$. □

Theorem 3.2 and its proof are cited from [22, section 1.5].

THEOREM 3.3 (Davis, Kahan, and Weinberger [5]). *Let*

$$f(X) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & X \end{pmatrix}$$

with $A_{11} \in \mathcal{C}^{k\times k}$ *and* $A_{21}, A_{12}^T \in \mathcal{C}^{l\times k}$. *Then*

$$\min_{X\in\mathcal{C}^{l\times l}} \|f(X)\|_2 = \max\left\{ \left\| \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \right\|_2, \ \|(A_{11}, A_{12})\|_2 \right\}.$$

Proof of Theorem 3.1. From (3.2) and (3.4) it follows that $\begin{pmatrix} E \\ F \end{pmatrix} \in \mathcal{G}$ if and only if $\begin{pmatrix} E \\ F \end{pmatrix}$ is a solution to the equations

$$E\tilde{x} = r_1, \quad F\tilde{x} = r_2, \quad E^H\tilde{z} = r_3, \quad F^H\tilde{w} = r_4. \tag{3.8}$$

Applying Theorem 3.2 to the first equation of (3.8) we see that the equation is solvable, and any solution of the equation can be expressed by

$$E = r_1\tilde{x}^\dagger + K(I - \tilde{x}\tilde{x}^\dagger), \quad K \in \mathcal{C}^{m\times n}. \tag{3.9}$$

Let

$$\tilde{u} = \tilde{x}/\|\tilde{x}\|_2, \tag{3.10}$$

and choose $\tilde{U}_2 \in \mathcal{C}^{n\times(n-1)}$ so that the matrix $\tilde{U} = (\tilde{u}, \tilde{U}_2)$ is unitary. Then (3.9) can be written

$$E = \frac{r_1\tilde{u}^H}{\|\tilde{x}\|_2} + K\tilde{U}_2\tilde{U}_2^H. \tag{3.11}$$

Combining it with the third equation of (3.8) shows that the matrix K of (3.11) satisfies

$$\frac{\tilde{u}r_1^H\tilde{z}}{\|\tilde{x}\|_2} + \tilde{U}_2\tilde{U}_2^H K^H\tilde{z} = r_3.$$

Multiplying the last equation by $\tilde{U}_2^H$ from the left yields

$$\tilde{z}^H K\tilde{U}_2 = r_3^H\tilde{U}_2. \tag{3.12}$$

By Theorem 3.2, (3.12) is solvable, and any solution K can be expressed by

$$K = \tilde{z}r_3^H\tilde{U}_2\tilde{U}_2^H + L - \tilde{z}\tilde{z}^H L\tilde{U}_2\tilde{U}_2^H, \quad L \in \mathcal{C}^{m\times n}. \tag{3.13}$$

Choose $\tilde{Z}_2 \in \mathcal{C}^{m\times(m-1)}$ so that the matrix $\tilde{Z} = (\tilde{z}, \tilde{Z}_2)$ is unitary. Then from (3.13)

$$K\tilde{U}_2 = \tilde{z}r_3^H\tilde{U}_2 + \tilde{Z}_2\tilde{Z}_2^H L\tilde{U}_2.$$

Substituting it into (3.11) gives

$$\begin{aligned} E &= \frac{r_1\tilde{u}^H}{\|\tilde{x}\|_2} + \tilde{z}r_3^H\tilde{U}_2\tilde{U}_2^H + \tilde{Z}_2\tilde{Z}_2^H L\tilde{U}_2\tilde{U}_2^H \\ &= \tilde{Z}\begin{pmatrix} \tilde{z}^H r_1/\|\tilde{x}\|_2 & r_3^H\tilde{U}_2 \\ \tilde{Z}_2^H r_1/\|\tilde{x}\|_2 & \tilde{Z}_2^H L\tilde{U}_2 \end{pmatrix}\tilde{U}^H \equiv E(L), \qquad L \in \mathcal{C}^{m\times n}. \end{aligned} \tag{3.14}$$

Similarly, choose $\tilde{W}_2 \in \mathcal{C}^{p\times(p-1)}$ so that the matrix $\tilde{W} = (\tilde{w}, \tilde{W}_2)$ is unitary. Then any solution F of the second and fourth equations of (3.8) can be expressed by

$$F = \tilde{W}\begin{pmatrix} \tilde{w}^H r_2/\|\tilde{x}\|_2 & r_4^H\tilde{U}_2 \\ \tilde{W}_2^H r_2/\|\tilde{x}\|_2 & \tilde{W}_2^H N\tilde{U}_2 \end{pmatrix}\tilde{U}^H \equiv F(N), \qquad N \in \mathcal{C}^{p\times n}. \tag{3.15}$$

Consequently, from (3.3), (3.14), and (3.15) we get

$$\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w}) = \min_{L\in\mathcal{C}^{m\times n},\, N\in\mathcal{C}^{p\times n}} \left\| \left(\frac{\|E(L)\|_2}{\gamma_{\mathrm{A}}}, \frac{\|F(N)\|_2}{\gamma_{\mathrm{B}}} \right)^T \right\|_\infty . \tag{3.16}$$

Observe the following facts: (i) Applying Theorem 3.3 to (3.14) and (3.15) gives

$$\min_{L\in\mathcal{C}^{m\times n}} \|E(L)\|_2 = \max\left\{ \left\| \begin{pmatrix} \tilde{z}^H r_1/\|\tilde{x}\|_2 \\ \tilde{Z}_2^H r_1/\|\tilde{x}\|_2 \end{pmatrix} \right\|_2, \ \left\| \left(\tilde{z}^H r_1/\|\tilde{x}\|_2, \ r_3^H\tilde{U}_2 \right) \right\|_2 \right\}$$

and

$$\min_{N\in\mathcal{C}^{p\times n}} \|F(N)\|_2 = \max\left\{ \left\| \begin{pmatrix} \tilde{w}^H r_2/\|\tilde{x}\|_2 \\ \tilde{W}_2^H r_2/\|\tilde{x}\|_2 \end{pmatrix} \right\|_2, \ \left\| \left(\tilde{w}^H r_2/\|\tilde{x}\|_2, \ r_4^H\tilde{U}_2 \right) \right\|_2 \right\}.$$

(ii) The relations (3.4), (3.10), and the last two relations of (3.1) imply

$$\frac{\tilde{z}^H r_1}{\|\tilde{x}\|_2} = \frac{\tilde{\alpha} - \tilde{z}^H A\tilde{x}}{\|\tilde{x}\|_2} = r_3^H\tilde{u}, \qquad \frac{\tilde{w}^H r_2}{\|\tilde{x}\|_2} = \frac{\tilde{\beta} - \tilde{w}^H B\tilde{x}}{\|\tilde{x}\|_2} = r_4^H\tilde{u}.$$

(iii) The matrices $(\tilde{z}, \tilde{Z}_2)$, $(\tilde{w}, \tilde{W}_2)$, and $(\tilde{u}, \tilde{U}_2)$ are unitary. Hence, we have

$$\begin{aligned} \min_{L\in\mathcal{C}^{m\times n}} \|E(L)\|_2 &= \max\left\{ \frac{\|r_1\|_2}{\|\tilde{x}\|_2}, \ \|r_3\|_2 \right\}, \\ \min_{N\in\mathcal{C}^{p\times n}} \|F(N)\|_2 &= \max\left\{ \frac{\|r_2\|_2}{\|\tilde{x}\|_2}, \ \|r_4\|_2 \right\}. \end{aligned} \tag{3.17}$$

Combining (3.16) with (3.17) shows the formula (3.5). □

Remark 3.1. Let (α,β) be a simple, finite, nonzero GSV of an (m,p,n)-GMP $\{A,B\}$, and let $x \in \mathcal{C}^n$ be an associated right generalized singular vector. Suppose that $(\tilde{\alpha},\tilde{\beta}) \neq (0,0)$ with $\tilde{\alpha},\tilde{\beta} \geq 0$ is an approximation of (α,β), and $\{\tilde{x},\tilde{y},\tilde{z},\tilde{w}\}$ is an associated approximate generalized singular vector group. Let $\sigma = \alpha/\beta$ and $\tilde{\sigma} = \tilde{\alpha}/\tilde{\beta}$. Then by the relation (2.3) and the definition (3.3) we have

$$|\tilde{\sigma} - \sigma| \lesssim c(\sigma)\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w}), \tag{3.18}$$

where $c(\sigma)$ and $\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w})$ have the formulas (2.4) and (3.5), respectively. One way to interpret the relation (3.18) is to say that the approximation $\tilde{\sigma}$ of σ may

not be close to σ if the condition number $c(\sigma)$ is very large, even if the approximate solution $\{(\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w}\}$ has a small backward error $\eta((\tilde{\alpha},\tilde{\beta});\tilde{x},\tilde{y},\tilde{z},\tilde{w})$. (See [11, section 1.6] for a clarification of a more general rule of thumb similar to (3.18).)

Remark 3.2. Let $\tilde{\sigma}$ be an approximate singular value of $A \in \mathcal{C}^{m\times n}$, and let $\{\tilde{v},\tilde{u}\}$ be an associated approximate singular vector pair. Then we may define the backward error $\eta(\tilde{\sigma};\tilde{v},\tilde{u})$ of A with respect to the approximate solution $\{\tilde{\sigma};\tilde{v},\tilde{u}\}$ by

$$\eta(\tilde{\sigma};\tilde{v},\tilde{u}) = \min_{E\in\mathcal{E}} \frac{\|E\|_2}{\gamma_{\scriptscriptstyle A}}, \tag{3.19}$$

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

$$\mathcal{E} = \left\{E \in \mathcal{C}^{m\times n} : (A+E)\tilde{v} = \tilde{\sigma}\tilde{u},\ (A+E)^H\tilde{u} = \tilde{\sigma}\tilde{v}\right\},$$

and $\gamma_{\scriptscriptstyle A}$ is a positive parameter. Taking $\gamma_{\scriptscriptstyle A} = 1$ gives the absolute backward error $\eta_{\rm abs}(\tilde{\sigma};\tilde{v},\tilde{u})$, and taking $\gamma_{\scriptscriptstyle A} = \|A\|_2$ gives the relative backward error $\eta_{\rm rel}(\tilde{\sigma};\tilde{v},\tilde{u})$. It is known (see, e.g., [22, Remark 3.4.3]) that we have the formula

$$\eta(\tilde{\sigma};\tilde{v},\tilde{u}) = \frac{1}{\gamma_{\scriptscriptstyle A}} \max\{\|r\|_2,\ \|s\|_2\}, \tag{3.20}$$

where

$$r = \tilde{\sigma}\tilde{u} - A\tilde{v}, \qquad s = \tilde{\sigma}\tilde{v} - A^H\tilde{u}.$$

The definition (3.3) is obviously a natural generalization of (3.19). It is worth pointing out that the formula (3.20) can be deduced from (3.5). In fact, if we consider the matrix A as the (m,n,n)-GMP $\{A, I_n\}$, then the relations of (3.1) are reduced to

$$A\tilde{v} \approx \tilde{\sigma}\tilde{u}, \quad A^H\tilde{u} \approx \tilde{\sigma}\tilde{v}, \quad \|\tilde{u}\|_2 = \|\tilde{v}\|_2 = 1,$$

because in such a case we have

$$\tilde{\alpha} = \tilde{\sigma}, \quad \tilde{\beta} = 1, \quad \tilde{x} = \tilde{y} = \tilde{w} = \tilde{v}, \quad \tilde{z} = \tilde{u}.$$

Consequently, the vectors r_1, r_2, r_3, r_4 of (3.4) are reduced to

$$r_1 = r, \quad r_3 = s, \quad r_2 = r_4 = 0,$$

and the formula (3.5) is reduced to (3.20).

4. Numerical examples. In this section we use two simple examples to illustrate the results of previous two sections. All computations were performed using MATLAB, 4.2c. The relative machine precision is 2.22×10^{-16}.

Example 4.1. Consider the $(2,2,2)$-GMP $\{A, B\}$ with

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{1+10^{-12}}} & \frac{1}{\sqrt{1+10^{-12}}} \end{pmatrix}, \qquad B = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ -\frac{10^{-6}}{\sqrt{1+10^{-12}}} & \frac{10^{-6}}{\sqrt{1+10^{-12}}} \end{pmatrix}.$$

The GSVs of the GMP are

$$(\alpha_1,\beta_1) = \frac{1}{\sqrt{2}}(1,1), \quad (\alpha_2,\beta_2) = \frac{1}{\sqrt{1+10^{-12}}}(1,10^{-6}), \qquad \text{i.e.,} \quad \sigma_1 = 1, \quad \sigma_2 = 10^6,$$

and the associated right generalized singular vectors are

$$x_1 = (1,1)^T, \qquad x_2 = (0,1)^T,$$

where every (α_j, β_j) and x_j satisfy (2.25). Computation gives

$$\frac{\|A\|_2}{\alpha_1} = 2.1358, \quad \frac{\|B\|_2}{\beta_1} = 1.0000, \quad \frac{\|A\|_2}{\alpha_2} = 1.5102, \quad \frac{\|B\|_2}{\beta_2} = 7.0711 \times 10^5,$$

$$\|x_1\|_2 = 1.4142, \quad \|x_2\|_2 = 1.0000, \quad \frac{\|x_1\|_2}{\beta_1} = 2.0000, \quad \frac{\|x_2\|_2}{\beta_2} = 1.0000 \times 10^6.$$

By using the formulas of (2.26) we get

$$\begin{aligned} &c_{\mathrm{abs}}(\sigma_1) = 4.0000, \qquad c_{\mathrm{abs}}(\sigma_2) = 1.0000 \times 10^{12}, \\ &c_{\mathrm{rel}}(\sigma_1) = 4.4347, \qquad c_{\mathrm{rel}}(\sigma_2) = 7.0711 \times 10^5, \end{aligned} \tag{4.1}$$

and

$$c_A^{(\mathrm{abs})}(\sigma_1) = c_B^{(\mathrm{abs})}(\sigma_1) = 2.0000, \qquad c_A^{(\mathrm{abs})}(\sigma_2) = c_B^{(\mathrm{abs})}(\sigma_2) = 1.0000 \times 10^6,$$

$$c_A^{(\mathrm{rel})}(\sigma_1) = 3.0204, \qquad c_B^{(\mathrm{rel})}(\sigma_1) = 1.4142,$$

$$c_A^{(\mathrm{rel})}(\sigma_2) = 1.5102, \qquad c_B^{(\mathrm{rel})}(\sigma_2) = 7.0711 \times 10^5.$$

The results of (4.1) show that the GSV σ_1 is well conditioned but the GSV σ_2 is ill conditioned, in both the absolute sense and relative sense.

Moreover, by using Algorithm GSVD22 presented by Bai and Demmel [1] (for computing the GSVD (1.9)), we get the computed GSVD [3]

$$\begin{aligned} \tilde{Z}^T A \tilde{Q} &= \begin{pmatrix} 4.999999999999999 \times 10^{-1} & -5.000000000000003 \times 10^{-1} \\ 5.551115123125783 \times 10^{-16} & 1.414213562372388 \times 10^{0} \end{pmatrix} = \tilde{\Sigma}_A \tilde{R}, \\ \tilde{W}^T A \tilde{Q} &= \begin{pmatrix} 4.999999999999999 \times 10^{-1} & -5.000000000000003 \times 10^{-1} \\ 5.293955920339377 \times 10^{-22} & 1.414213562372389 \times 10^{-6} \end{pmatrix} = \tilde{\Sigma}_B \tilde{R}, \end{aligned} \tag{4.2}$$

where

$$\begin{aligned} &\tilde{Z} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (\tilde{z}_1, \tilde{z}_2), \qquad \tilde{W} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (\tilde{w}_1, \tilde{w}_2), \\ &\tilde{Q} = \begin{pmatrix} 7.071067811865475 \times 10^{-1} & -7.071067811865480 \times 10^{-1} \\ 7.071067811865480 \times 10^{-1} & 7.071067811865475 \times 10^{-1} \end{pmatrix}. \end{aligned} \tag{4.3}$$

From (4.2) and (4.3) we get

$$(\tilde{\alpha}_1, \tilde{\beta}_1) = (7.071067811865476 \times 10^{-1},\ 7.071067811865476 \times 10^{-1}),$$

$$(\tilde{\alpha}_2, \tilde{\beta}_2) = (9.999999999995000 \times 10^{-1},\ 9.999999999995000 \times 10^{-7}),$$

and

$$\tilde{X} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = (\tilde{x}_1, \tilde{x}_2), \qquad \tilde{Y} = \tilde{X}^{-T} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} = (\tilde{y}_1, \tilde{y}_2).$$

For the computed $\{(\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1\}$, we have

$$\begin{aligned} &\|r_1\|_2 = 6.6613 \times 10^{-16}, \qquad \|r_2\|_2 = 6.3527 \times 10^{-22}, \\ &\|r_3\|_2 = 4.4409 \times 10^{-16}, \qquad \|r_4\|_2 = 4.4409 \times 10^{-16}, \end{aligned} \tag{4.4}$$

where r_1, r_2, r_3, r_4 are the residuals defined by (3.4). Substituting (4.4) and

$$\|A\|_2 = 1.5102, \qquad \|B\|_2 = 7.0711 \tag{4.5}$$

into the formula (3.5) gives

$$\begin{aligned} &\eta_{\mathrm{abs}}((\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1) = 4.7103 \times 10^{-16}, \\ &\eta_{\mathrm{rel}}((\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1) = 6.2804 \times 10^{-16}. \end{aligned} \tag{4.6}$$

Similarly, for the computed $\{(\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2\}$, we have

$$\begin{aligned} &\|r_1\|_2 = 2.2204 \times 10^{-16}, \qquad \|r_2\|_2 = 2.1176 \times 10^{-22}, \\ &\|r_3\|_2 = 1.1322 \times 10^{-15}, \qquad \|r_4\|_2 = 1.0798 \times 10^{-21}. \end{aligned} \tag{4.7}$$

Substituting (4.5) and (4.7) into the formula (3.5) gives

$$\begin{aligned} &\eta_{\mathrm{abs}}((\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2) = 1.1322 \times 10^{-15}, \\ &\eta_{\mathrm{rel}}((\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2) = 7.4970 \times 10^{-16}. \end{aligned} \tag{4.8}$$

The results (4.6) and (4.8) show that the computation of the GSVD (4.2) by using Algorithm GSVD22 has proceeded stably.

Example 4.2 (see [1, section 5.2]). Consider the $(2, 2, 2)$-GMP $\{A, B\}$ with

$$A = \begin{pmatrix} 2 & 0 \\ 1 & 10^{-8} \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}.$$

By using Algorithm GSVD22, we get the computed GSVD [3]

$$\begin{aligned} \tilde{Z}^T A \tilde{Q} &= \begin{pmatrix} 7.071067773681713 \times 10^{-1} & 2.121320344832436 \\ -2.504923900926630 \times 10^{-16} & 2.828427064871980 \times 10^{-8} \end{pmatrix} = \tilde{\Sigma}_A \tilde{R}, \\ \tilde{W}^T B \tilde{Q} &= \begin{pmatrix} 3.162277662065746 \times 10^{-1} & 9.486833043118236 \times 10^{-1} \\ -1.110223024625157 \times 10^{-16} & 3.162277658271013 \end{pmatrix} = \tilde{\Sigma}_B \tilde{R}, \end{aligned} \tag{4.9}$$

where

$$\begin{aligned} \tilde{Z} &= \begin{pmatrix} 8.944271963664790 \times 10^{-1} & -4.472135847668320 \times 10^{-1} \\ 4.472135847668320 \times 10^{-1} & 8.944271963664790 \times 10^{-1} \end{pmatrix} = (\tilde{z}_1, \tilde{z}_2), \\ \tilde{W} &= \begin{pmatrix} 1.000000000000000 & -1.999999963999999 \times 10^{-9} \\ 1.999999963999999 \times 10^{-9} & 1.000000000000000 \end{pmatrix} = (\tilde{w}_1, \tilde{w}_2), \end{aligned} \tag{4.10}$$

and

$$\tilde{Q} = \begin{pmatrix} 3.162277662065746 \times 10^{-1} & 9.486832979872684 \times 10^{-1} \\ -9.486832979872684 \times 10^{-1} & 3.162277662065746 \times 10^{-1} \end{pmatrix}. \tag{4.11}$$

From (4.9)–(4.11) we get

$$(\tilde{\alpha}_1, \tilde{\beta}_1) = (9.128709282624060 \times 10^{-1},\ 4.082482925051043 \times 10^{-1}),$$

$$(\tilde{\alpha}_2, \tilde{\beta}_2) = (8.944271726026845 \times 10^{-9},\ 1.000000000000000),$$

and

$$\tilde{X} = \begin{pmatrix} 4.082482925051043 \times 10^{-1} & -1.999999998947288 \times 10^{-9} \\ -1.224744876698816 & 1.000000006000000 \end{pmatrix} = (\tilde{x}_1, \tilde{x}_2)\,,$$

$$\tilde{Y} = \tilde{X}^{-T} = \begin{pmatrix} 2.449489745232669 & 2.999999998000000 \\ 4.898979444334373 \times 10^{-9} & 1.000000000000000 \end{pmatrix} = (\tilde{y}_1, \tilde{y}_2)\,.$$

For the computed $\{(\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1\}$, we have

$$\begin{aligned} &\|r_1\|_2 = 7.5503 \times 10^{-16}, \qquad \|r_2\|_2 = 1.4429 \times 10^{-16}, \\ &\|r_3\|_2 = 8.9057 \times 10^{-16}, \qquad \|r_4\|_2 = 2.2395 \times 10^{-16}. \end{aligned} \tag{4.12}$$

Substituting (4.12) and

$$\|A\|_2 = 2.2361, \qquad \|B\|_2 = 3.3028 \tag{4.13}$$

into the formula (3.5) gives

$$\begin{aligned} &\eta_{\mathrm{abs}}((\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1) = 8.9057 \times 10^{-16}, \\ &\eta_{\mathrm{rel}}((\tilde{\alpha}_1, \tilde{\beta}_1); \tilde{x}_1, \tilde{y}_1, \tilde{z}_1, \tilde{w}_1) = 3.9827 \times 10^{-16}. \end{aligned} \tag{4.14}$$

Similarly, for the computed $\{(\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2\}$, we have

$$\begin{aligned} &\|r_1\|_2 = 2.5016 \times 10^{-16}, \qquad \|r_2\|_2 = 2.2478 \times 10^{-16}, \\ &\|r_3\|_2 = 2.5049 \times 10^{-16}, \qquad \|r_4\|_2 = 4.4409 \times 10^{-16}. \end{aligned} \tag{4.15}$$

Substituting (4.13) and (4.15) into the formula (3.5) gives

$$\begin{aligned} &\eta_{\mathrm{abs}}((\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2) = 4.4409 \times 10^{-16}, \\ &\eta_{\mathrm{rel}}((\tilde{\alpha}_2, \tilde{\beta}_2); \tilde{x}_2, \tilde{y}_2, \tilde{z}_2, \tilde{w}_2) = 1.3446 \times 10^{-16}. \end{aligned} \tag{4.16}$$

The results (4.14) and (4.16) show that the computation of the GSVD (4.9) by using Algorithm GSVD22 has proceeded stably.

5. Acknowledgments. I would like to thank Ilse Ipsen and the referees for their helpful comments and valuable suggestions. I also thank Zhaojun Bai who gave me the computed GSVDs of the $(2, 2, 2)$-GMPs of Examples 4.1 and 4.2 by using Algorithm GSVD22 presented by [1].

REFERENCES

[1] Z. Bai and J. W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.
[2] Z. Bai and H. Zha, *A new preprocessing algorithm for the computation of the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1007–1012.
[3] Z. Bai, *private communication*, May 1999.
[4] M. T. Chu, R. E. Funderlic, and G. H. Golub, *On a variational formulation of the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1082–1092.
[5] C. Davis, W. M. Kahan, and H. F. Weinberger, *Norm-preserving dilations and their applications to optimal error bounds*, SIAM J. Numer. Anal., 19 (1982), pp. 445–469.
[6] J. P. Dedieu, *Condition operators, condition numbers and condition number theorem for the generalized eigenvalue problem*, Linear Algebra Appl., 263 (1997), pp. 1–24.
[7] L. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–502.
[8] V. Frayssé and V. Toumazou, *A note on the normwise perturbation theory for the regular generalized eigenproblem* $Ax = \lambda Bx$, Numer. Linear Algebra Appl., 5 (1998), pp. 1–10.
[9] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[10] D. J. Higham and N. J. Higham, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 493–512.
[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[12] B. Kågström, *The generalized singular value decomposition and the general* $(A - \lambda B)$*-problem*, BIT, 24 (1984), pp. 568–583.
[13] R.-C. Li, *Bounds on perturbations of generalized singular values and of associated subspaces*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 195–234.
[14] C. C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.
[15] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
[16] J. R. Rice, *A theory of condition*, SIAM J. Numer. Anal., 3 (1966), pp. 287–310.
[17] G. W. Stewart, *Computing the CS-decomposition of a partitioned orthonormal matrix*, Numer. Math., 40 (1982), pp. 297–306.
[18] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[19] J.-G. Sun, *On the perturbation of generalized singular values*, Math. Numer. Sinica, 4 (1982), pp. 229–233 (in Chinese).
[20] J.-G. Sun, *Perturbation analysis for the generalized singular value problem*, SIAM J. Numer. Anal., 20 (1983), pp. 611–625.
[21] J.-G. Sun, *Perturbation analysis of generalized singular subspaces*, Numer. Math., 79 (1998), pp. 615–641.
[22] J.-G. Sun, *Stability and Accuracy: Perturbation Analysis of Algebraic Eigenproblems*, Report UMINF 98.07, ISSN-0348-0542, Department of Computing Science, Umeå University, Umeå, Sweden, 1998.
[23] J.-G. Sun, *On Condition Numbers for Generalized Singular Values*, Report UMINF 99.14, ISSN-0348-0542, Department of Computing Science, Umeå University, 1999.
[24] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.
[25] C. F. Van Loan, *Computing the CS and the generalized singular value decomposition*, Numer. Math., 46 (1985), pp. 479–491.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 342–357

MULTILEVEL SOLUTIONS FOR STRUCTURED MARKOV CHAINS*

PETER BUCHHOLZ†

Abstract. In this paper, a new analysis approach for continuous time Markov chains (CTMCs) with a multidimensional structure is introduced. The presented solution technique employs the multidimensional structure to define aggregated CTMCs that can be analyzed more efficiently. Generator matrices of aggregated CTMCs are described in a compact form by exploiting the Kronecker structure of the generator matrix of a structured CTMC. The solution of aggregated systems is used to improve the solution of the original system. This idea of a multilevel solution is motivated by multigrid methods, which are efficient solvers for partial differential equations. The technique can be combined with different iterative solution techniques. It usually improves the convergence of these techniques significantly. Numerical results are given to illustrate that the new solution technique allows the fast and accurate analysis of very large CTMCs.

Key words. continuous time Markov chains, stationary analysis, iterative solution techniques, aggregation-disaggregation

AMS subject classifications. 60J20, 65C20, 65F10

PII. S0895479898342419

1. Introduction.

Markovian modeling is used in many areas in evaluating the performance or reliability of existing or planned systems. Often the stationary distribution of a continuous time Markov chain (CTMC) is computed to determine the long-run behavior of the system. Stationary analysis requires the solution of

$$\mathbf{pQ} = \mathbf{0} \text{ subject to } \|\mathbf{p}\|_1 = 1, \tag{1}$$

where $\mathbf{Q}$ is the generator matrix of the CTMC and $\mathbf{p}$ is the unknown stationary probability vector. From the stationary vector $\mathbf{p}$, system-related measures like throughputs, response times, and availability can be computed. The dimensions of $\mathbf{Q}$ and $\mathbf{p}$ equal the number of states of the CTMC, which often grows exponentially with the size of the model measured in the number of model components, such as queues and customers in queueing networks (QNs) or places and tokens in stochastic Petri nets (SPNs). We consider here finite CTMCs, which, nevertheless, may have a huge state space including several millions of states. In summary, determination of $\mathbf{p}$ requires the solution of a system of equations with a large, sparse and, in almost all cases, nonsymmetric coefficient matrix. This is a cumbersome task, even with contemporary computer equipment.

For the solution of (1) a large number of solution techniques exist (see [25] for an excellent overview), but even sophisticated iterative solution techniques combined with sparse storage schemes, as commonly used, reach their limits for most realistic examples. Problems may arise due to huge memory requirements or due to long solution times caused by a slow convergence of the iterative method. One way to overcome at least the problem of exceeding available memory is to represent matrix $\mathbf{Q}$ in a compact form as the Kronecker product of small component matrices. The compact representation of $\mathbf{Q}$ has been derived for specific finite QNs with overflow in [19, 10].

*Received by the editors July 27, 1998; accepted for publication (in revised form) by D. O'Leary February 4, 2000; published electronically July 11, 2000.
http://www.siam.org/journals/simax/22-2/34241.html
†Department of Computer Science, Dresden University of Technology, D-01062 Dresden, Germany (p.buchholz@inf.tu-dresden.de).

A more general class of models with a Kronecker structure of the generator matrix has been presented by Plateau [22]. Models are described as stochastic automata networks (SANs), which specify a CTMC under Markovian timing and transitions. SANs are a very general paradigm such that the underlying concepts can also be used to describe SPNs [15] or QNs in a structured way.

The compact representation of $\mathbf{Q}$ can be directly used in vector-matrix multiplications involving the generator matrix. In this way iterative techniques are realized without generating $\mathbf{Q}$ as a whole. Possible iterative techniques include the Power method and projection methods such as GMRES or Arnoldi [25, 26]. Additionally, Jacobi overrelaxation (JOR) and, with some additional effort, successive overrelaxation (SOR) [8, 27] can be implemented in conjunction with the compact representation of $\mathbf{Q}$. The compact representation of $\mathbf{Q}$ solves to some extent the problem of limited available space, but it generally does not solve the problem of long solution times. The major problem is that large CTMCs can be analyzed with the available memory, but due to slow convergence or even divergence, sufficiently exact solutions cannot be obtained. Thus there is a need for more efficient solution techniques exploiting the compact matrix representation.

For elliptic differential equations with a symmetric coefficient matrix, which sometimes also allow a compact matrix representation via Kronecker products of small matrices, efficient preconditioning techniques exist to accelerate the convergence of iterative solution methods [1]. Similar preconditioners have been used for overflow QNs [10, 11, 12]. However, the approach cannot be extended to more general models because it relies on the specific matrix structure of this kind of models. New analysis techniques for SANs have been proposed recently. In [25, 26, 6] preconditioning is used to accelerate convergence of iterative methods, and [4, 5] propose aggregation-disaggregation to speed up the solution.

In this paper, we present a new approach for the analysis of CTMCs with a multidimensional state space that extends the idea of aggregation-disaggregation steps to accelerate convergence of iterative techniques. The multilevel solution technique that is introduced here exploits the multidimensional structure of the CTMC by defining aggregated CTMCs. An aggregated CTMC results from the original CTMC by keeping the distribution in some dimensions constant. Since the state space of aggregated CTMCs is smaller than the state space of the original CTMC, iterations of iterative solution techniques are performed more efficiently for aggregated CTMCs and the resulting aggregated solution vector is used to improve the solution of the complete CTMC. This approach can be combined with any iterative solution technique exploiting the compact representation of $\mathbf{Q}$. The basic step follows ideas from algebraic multigrid techniques [2] and the multilevel approach for CTMCs [18]. The multidimensional structure of the CTMC and the Kronecker structure of the generator matrix allow a very convenient definition of probability vectors and generator matrices for reduced CTMCs.

In what follows, boldface capital and lowercase letters are used for matrices and vectors, respectively. All vectors are row vectors. Column vectors are represented as transposed row vectors, i.e., $\mathbf{p}^T$ is the column vector resulting from transposing row vector $\mathbf{p}$. $\mathbf{I}_n$ is used for the identity matrix of dimension n and $\mathbf{e}_n$ for a row vector of length n, where all elements are 1. If the dimension follows from the context, the subscript is suppressed. For an n-dimensional column vector $\mathbf{p}^T$, $\mathrm{diag}(\mathbf{p}^T)$ is an $n \times n$ diagonal matrix with $\mathbf{p}(s)$ in position (s, s). We use $\|.\|$ for vector norms and matrix norms. Sets are represented by calligraphic letters. We assume that all sets

are ordered such that the order of multiplication or summation over the elements of a set is well defined. $|.|$ stands for the number of elements in a set. Indices s and t are used for states, index e describes a synchronization event, and indices i and j denote components of a structured model. Subscripts and superscripts identify quantities belonging to components and synchronization events. $\mathbf{Q}_e^{(i)}$ is the matrix describing synchronized transition e in component i; s_i is a state belonging to component i.

The structure of the paper is as follows. In the next section we present structured CTMCs resulting from SANs or related modeling formalisms. Afterward, in section 3, aggregation-disaggregation with respect to components (i.e., dimensions) or sets of components is introduced. In section 4 the multilevel solution algorithm is presented. Then examples are used to show the benefits of the proposed algorithm compared to other solution techniques. The paper ends with a summary of the presented results and an outline of open research topics.

2. Structured CTMCs. Structured models consist of a set $\mathcal{J} = \{1, \ldots, J\}$ of components. Each component j has a finite state space $\mathcal{S}^{(j)}$ including n_j states. We assume that states are numbered consecutively from 0 through $n_j - 1$. The potential state space of the complete model composed of the components equals

$$\mathcal{S} = \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(J)}$$

and includes $n = \prod_{j \in \mathcal{J}} n_j$ states. A state of the complete model can be described by a vector $(s_1, \ldots, s_J)$ $(s_j \in \mathcal{S}^{(j)})$ or equivalently by an integer

$$s = \sum_{j \in \mathcal{J}} s_j \prod_{i \in \mathcal{J},\ i>j} n_i \tag{2}$$

resulting from a mixed radix number representation [13]. In what follows we use the vector representation and integer representation interchangeably.

Transitions in the model result from local transitions in one of the components and synchronized transitions among subsets of components. We assume that the future behavior of the model depends only on the state of the components and that no simultaneous transitions occur. Consequently, the model describes a CTMC. Let $\mathbf{Q}_l^{(j)}$ be an $n_j \times n_j$ matrix including in position (s,t) the transition rate of local transitions starting in state $s \in \mathcal{S}^{(j)}$ and ending in state $t \in \mathcal{S}^{(j)}$. Local transitions occur independently in the components. Apart from local transitions, synchronized transitions involving sets of components exist. Let $\mathcal{E}$ be the set of synchronized transitions and $\mathbf{Q}_e^{(j)}$ be an $n_j \times n_j$ matrix that describes the effect of synchronized transition $e \in \mathcal{E}$ on component $j \in \mathcal{J}$. Usually one component j includes the rate of transition e and the others include probabilities describing the choice of successor states when e occurs. Observe that if one component disables transition e (i.e., the corresponding row in matrix $\mathbf{Q}_e^{(j)}$ is zero), then e cannot occur. We define $\mathbf{Q}_e^{(j)} = \mathbf{I}_{n_j}$ if component j does not participate in the synchronization due to synchronized transition e. It is well known that the generator matrix of the structured model can be represented as [22, 26, 25]

$$\begin{aligned} \mathbf{Q} \;=\; & \sum_{j \in \mathcal{J}} \mathbf{I}_{l_j} \bigotimes \mathbf{Q}_l^{(j)} \bigotimes \mathbf{I}_{u_j} + \sum_{e \in \mathcal{E}} \bigotimes_{j \in \mathcal{J}} \mathbf{Q}_e^{(j)} \\ & - \sum_{j \in \mathcal{J}} \mathbf{I}_{l_j} \bigotimes \mathbf{D}_l^{(j)} \bigotimes \mathbf{I}_{u_j} - \sum_{e \in \mathcal{E}} \bigotimes_{j \in \mathcal{J}} \mathbf{D}_e^{(j)} \;, \end{aligned} \tag{3}$$

where $l_j = \prod_{i\in\mathcal{J},\ i<j} n_i$, $u_j = \prod_{i\in\mathcal{J},\ i>j} n_i$, $\mathbf{D}_l^{(j)} = \mathrm{diag}(\mathbf{Q}_l^{(j)}\mathbf{e}^T)$, $\mathbf{D}_e^{(j)} = \mathrm{diag}(\mathbf{Q}_e^{(j)}\mathbf{e}^T)$, and $\bigotimes$ is the Kronecker (tensor) product of matrices [13, 25]. Observe that (3) is a representation using only matrices of size $n_j \times n_j$ rather than $n \times n$.

In what follows we assume that $\mathcal{S}$ includes a single irreducible subset of states such that the stationary solution vector $\mathbf{p}$ exists uniquely. If this is not the case, one can still solve the system with an iterative technique by assigning nonzero probabilities to states in only one ergodic subset of states [8, 26]. The matrix representation (3) can be exploited by all iterative solution techniques that are based on the computation of vector-matrix products using the generator matrix or the generator matrix without diagonal elements. These techniques include the Power method, JOR, GMRES, Arnoldi, and several others. The basic step is the computation of the product of a vector with a tensor product of matrices; efficient algorithms for this purpose can be found in [8, 9, 16]. For details about numerical analysis techniques exploiting the matrix structure we refer to the literature [8, 4, 5, 26, 25, 27]. The advantage of using the representation (3), instead of the matrix $\mathbf{Q}$ stored in sparse format, is usually a drastic reduction of storage requirements such that the state space of theoretically solvable CTMCs on a given hardware can be increased by about one order of magnitude. However, the structured approach usually does not reduce the solution time significantly [6, 27]. Thus, the solution of large structured CTMCs is often a very time-consuming task and there is a need for more efficient solution techniques.

Example. As an example we consider a simple overflow QN with three queues. Customers arrive to queue i according to a Poisson process with rate λ_i, and they have exponentially distributed service requirements with rate μ_i. We assume that queue i has capacity k_i. We will consider two versions of this model.

In the first version ($V1$) customers arriving when queue 1 is full try to enter queue 2. If queue 2 is also full, they try to enter queue 3. If queue 3 is also full, the arriving customer gets lost. Customers arriving to queue 2, which is full, try to enter queue 3 and get lost if queue 3 is also full. Customers arriving to queue 3 immediately get lost if the queue is full.

This model can be described using three components. Each component models one queue and the corresponding arrival process. Three synchronized events are required, one involving all three components and two involving two components. We denote the events by 12, 13, and 23 to describe the arrival of a customer from the queue given by the first number to the queue given by the second number. The following matrices describe the components:

$$\mathbf{Q}_l^{(j)} = \begin{pmatrix} 0 & \lambda_j & & & 0 \\ \mu_j & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda_j \\ 0 & & & \mu_j & 0 \end{pmatrix},$$

$$\mathbf{Q}_{12}^{(1)} = \mathbf{Q}_{13}^{(1)} = \begin{pmatrix} 0 & 0 & & & 0 \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 0 \\ 0 & & & 0 & \lambda_1 \end{pmatrix}, \quad \mathbf{Q}_{23}^{(2)} = \begin{pmatrix} 0 & 0 & & & 0 \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 0 \\ 0 & & & 0 & \lambda_2 \end{pmatrix},$$

$$\mathbf{Q}_{12}^{(2)} = \mathbf{Q}_{13}^{(3)} = \mathbf{Q}_{23}^{(3)} = \begin{pmatrix} 0 & 1 & 0 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & \ddots & 1 \\ 0 & & & & 0 \end{pmatrix}, \quad \mathbf{Q}_{13}^{(2)} = \begin{pmatrix} 0 & 0 & & & 0 \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 0 & 0 \\ 0 & & & 0 & 1 \end{pmatrix}.$$

Observe that even for this simple model, the advantage of the structured approach is obvious. The size of the component state space equals $k_i + 1$, whereas the complete CTMC has $(k_1+1)(k_2+1)(k_3+1)$ states. Similarly, the number of nonzero elements in the component matrices including diagonal elements equals $5k_1 + 5k_2 + 5k_3 + 8$, whereas matrix $\mathbf{Q}$ contains $k_1k_2k_3 + k_1k_2 + k_1k_3 + k_3$ nonzeros.

The second version ($V2$) of the model allows the overflow of customers from each queue to each other queue. Thus, six additional synchronized transitions have to be added. The corresponding matrices are similar to the matrices shown for $V1$.

3. Aggregation and disaggregation. The idea of the multilevel solution approach proposed in this paper is to exploit the multidimensional structure to make iterative solution techniques more efficient. We define CTMCs of a lower dimension by keeping the distribution in some dimensions constant and building an aggregated CTMC with respect to this constant distribution. In what follows, some basic operations are introduced first.

Let $\mathcal{D} \subseteq \mathcal{J}$ be a subset of components that are considered in detail. The remaining components $\mathcal{J} \setminus \mathcal{D}$ are aggregated. We use superscript $\mathcal{D}$ for quantities belonging to the aggregated model with aggregated components $\mathcal{J} \setminus \mathcal{D}$ and detailed components $\mathcal{D}$. Usually the superscript is suppressed if $\mathcal{D} = \mathcal{J}$. The state space of the aggregated system is defined as $\mathcal{S}^{\mathcal{D}} = \times_{j\in\mathcal{D}}\mathcal{S}^{(j)}$ and includes $n^{\mathcal{D}} = \prod_{j\in\mathcal{D}} n_j$ states. States in $\mathcal{S}^{\mathcal{D}}$ are represented by a $|\mathcal{D}|$-dimensional description including the states of components in $\mathcal{D}$. $s_j^{\mathcal{D}} \in \mathcal{S}^{(j)}$ is the state of component $j \in \mathcal{D}$ belonging to state $s^{\mathcal{D}} \in \mathcal{S}^{\mathcal{D}}$. Using (2) we obtain an integer representation

$$s^{\mathcal{D}} = \sum_{j\in\mathcal{D}} s_j^{\mathcal{D}} \prod_{i\in\mathcal{D}, i>j} n_i \ .$$

Now assume that $\mathbf{x}^{\mathcal{D}}$ is a vector on state space $\mathcal{S}^{\mathcal{D}}$ and let $\mathcal{C} \subset \mathcal{D}$ be a subset of the components in $\mathcal{D}$. Let $s^{\mathcal{D}} \in \mathcal{S}^{\mathcal{D}}$ be a state of the aggregated CTMC described by components $\mathcal{D}$. Each state $s^{\mathcal{D}}$ belongs to a state $s^{\mathcal{C}} \in \mathcal{S}^{\mathcal{C}}$, where the state of components $j \in \mathcal{C}$ is given by $s_j^{\mathcal{D}}$ and the state of the remaining components $j \in \mathcal{D}\setminus\mathcal{C}$ is neglected. The index of state $s^{\mathcal{C}}$ is computed as

$$s^{\mathcal{C}} = \sum_{j\in\mathcal{C}} s_j^{\mathcal{D}} \prod_{i\in\mathcal{C}, i>j} n_i \ . \tag{4}$$

Let $proj(s^{\mathcal{D}}, \mathcal{D}, \mathcal{C})$ be a function computing $s^{\mathcal{C}}$ from $s^{\mathcal{D}}$ via (4). A vector $\mathbf{x}^{\mathcal{C}}$, which results from mapping $\mathbf{x}^{\mathcal{D}}$ onto $\mathcal{S}^{\mathcal{C}}$, can be computed elementwise as

$$\mathbf{x}^{\mathcal{C}}(s^{\mathcal{C}}) = \sum_{s^{\mathcal{D}}\in\mathcal{S}^{\mathcal{D}}, proj(s^{\mathcal{D}},\mathcal{D},\mathcal{C})=s^{\mathcal{C}}} \mathbf{x}^{\mathcal{D}}(s^{\mathcal{D}}) \tag{5}$$

for all $s^{\mathcal{C}} \in \mathcal{S}^{\mathcal{C}}$. The mapping will be used for iteration and residual vectors. Let $\mathbf{x}$ be an approximation for the stationary solution that has been computed by performing

some iteration steps with an iterative solution technique, and let $\mathbf{r} = \mathbf{xQ}$ be the corresponding residual vector. Via (5) residuals are mapped on the state space of a single component, i.e., $\mathbf{r}^{(i)}$ is computed. By comparing $\|\mathbf{r}^{(i)}\|$ it is possible to find dimensions in the state space with large residuals and dimensions with small residuals.

The opposite operation of a mapping from a more detailed to a less detailed state space is the redirection of a more detailed solution using a less detailed one. Let $\mathbf{x}^{\mathcal{D}}$ be a solution vector on $\mathcal{S}^{\mathcal{D}}$, and let $\mathbf{y}^{\mathcal{C}}$ be an improved solution on $\mathcal{S}^{\mathcal{C}}$. An improved solution vector $\mathbf{y}^{\mathcal{D}}$ on $\mathcal{S}^{\mathcal{D}}$ is computed using the following equation, where $\mathbf{x}^{\mathcal{C}}$ is the projection of $\mathbf{x}^{\mathcal{D}}$ on $\mathcal{S}^{\mathcal{C}}$ via (5):

$$\mathbf{y}^{\mathcal{D}}(s^{\mathcal{D}}) = \mathbf{x}^{\mathcal{D}}(s^{\mathcal{D}}) \frac{\mathbf{y}^{\mathcal{C}}(proj(s^{\mathcal{D}}, \mathcal{D}, \mathcal{C}))}{\mathbf{x}^{\mathcal{C}}(proj(s^{\mathcal{D}}, \mathcal{D}, \mathcal{C}))}. \tag{6}$$

This is the usual interpolation operation that is used in multigrid or aggregation-disaggregation techniques [2, 18, 25]. Observe that the mapping of $\mathbf{y}^{\mathcal{D}}$ onto $\mathcal{S}^{\mathcal{C}}$ equals $\mathbf{y}^{\mathcal{C}}$.

The next step introduces the computation of an aggregated generator matrix $\mathbf{Q}^{\mathcal{C}}$ from a generator matrix $\mathbf{Q}^{\mathcal{D}}$ and a vector $\mathbf{x}^{\mathcal{D}}$ for $\mathcal{C} \subset \mathcal{D}$. The idea of generating $\mathbf{Q}^{\mathcal{C}}$ from $\mathbf{Q}^{\mathcal{D}}$ is to aggregate components $j \in \mathcal{D} \setminus \mathcal{C}$. This aggregation is done with respect to vector $\mathbf{x}^{\mathcal{D}}$. Matrices $\mathbf{Q}_l^{(j)}$ for $j \in \mathcal{D} \setminus \mathcal{C}$ are not required to represent $\mathbf{Q}^{\mathcal{C}}$ because these matrices describe only transitions in dimensions which are aggregated. For synchronized events, the contribution of components $j \in \mathcal{D} \setminus \mathcal{C}$ has to be represented in an aggregated form and depends on the vector $\mathbf{x}^{\mathcal{D}}$. Define an $n^{\mathcal{C}} \times n^{\mathcal{C}}$ diagonal matrix as follows:

$$\begin{aligned} &\mathbf{A}_e^{\mathcal{C}}(s^{\mathcal{C}}, s^{\mathcal{C}}) \\ &= \left(\sum_{s^{\mathcal{D}} \in \mathcal{S}^{\mathcal{D}}, proj(s^{\mathcal{D}}, \mathcal{D}, \mathcal{C}) = s^{\mathcal{C}}} \mathbf{x}^{\mathcal{D}}(s^{\mathcal{D}}) \mathbf{A}_e^{\mathcal{D}}(s^{\mathcal{D}}, s^{\mathcal{D}}) \prod_{j \in \mathcal{D} \setminus \mathcal{C}} \mathbf{D}_e^{(j)}(s_j^{\mathcal{D}}, s_j^{\mathcal{D}}) \right) \; / \; \mathbf{x}^{\mathcal{C}}(s^{\mathcal{C}}), \end{aligned} \tag{7}$$

where $\mathbf{x}^{\mathcal{C}}$ is the mapping of $\mathbf{x}^{\mathcal{D}}$ onto $\mathcal{S}^{\mathcal{C}}$ via (5) and $\mathbf{A}_e^{\mathcal{J}} = \mathbf{I}_n$. Values of the aggregated matrix are computed according to the conditional probability distribution over states $s_j^{\mathcal{D}}$, $j \in \mathcal{D} \setminus \mathcal{C}$, with respect to a fixed state $s^{\mathcal{C}}$. This corresponds to the common computation of aggregated transition rates in CTMC aggregation algorithms [18, 25]. The aggregated generator matrix is given by

$$\begin{aligned} \mathbf{Q}^{\mathcal{C}} \;=\; & \sum_{j \in \mathcal{C}} \mathbf{I}_{l_j^{\mathcal{C}}} \bigotimes \mathbf{Q}_l^{(j)} \bigotimes \mathbf{I}_{u_j^{\mathcal{C}}} + \sum_{e \in \mathcal{E}} \mathbf{A}_e^{\mathcal{C}} \bigotimes_{j \in \mathcal{C}} \mathbf{Q}_e^{(j)} \\ & - \sum_{j \in \mathcal{C}} \mathbf{I}_{l_j^{\mathcal{C}}} \bigotimes \mathbf{D}_l^{(j)} \bigotimes \mathbf{I}_{u_j^{\mathcal{C}}} - \sum_{e \in \mathcal{E}} \mathbf{A}_e^{\mathcal{C}} \bigotimes_{j \in \mathcal{C}} \mathbf{D}_e^{(j)} \; , \end{aligned} \tag{8}$$

where $l_j^{\mathcal{C}} = \prod_{i \in \mathcal{C},\ i<j} n_i$ and $u_j^{\mathcal{C}} = \prod_{i \in \mathcal{C},\ i>j} n_i$. The first sum describes local transitions and the second sum synchronized transitions; the remaining parts determine the diagonal elements. In the second and fourth sums only those synchronized events from $\mathcal{E}$ have to be considered, for which at least one matrix $\mathbf{Q}_e^{(j)} \neq \mathbf{I}_{n_j}$ for some $j \in \mathcal{C}$. $\mathbf{Q}^{\mathcal{C}}$ is an $n^{\mathcal{C}} \times n^{\mathcal{C}}$ matrix that can be represented by the component matrices plus up to $|\mathcal{E}|$ diagonal matrices of order $n^{\mathcal{C}}$. This representation is compact as long as the number of synchronized events is moderate. Since $n = n^{\mathcal{C}} \cdot n^{\mathcal{J} \setminus \mathcal{C}}$, the number of states for the aggregated system $n^{\mathcal{C}}$ is usually much smaller than n.

THEOREM 3.1. *If $\bar{\mathcal{S}} \subseteq \mathcal{S}$ is an irreducible subset of states for matrix $\mathbf{Q}$ and $\mathbf{x}(s) > 0$ for $s \in \bar{\mathcal{S}}$, and $\mathbf{x}(s) = 0$ for $s \notin \bar{\mathcal{S}}$, then for each $\mathcal{D} \subseteq \mathcal{J}$, $\bar{\mathcal{S}}^{\mathcal{D}} \subseteq \mathcal{S}^{\mathcal{D}}$, defined as the mapping of states from $\bar{\mathcal{S}}$ onto $\mathcal{S}^{\mathcal{D}}$, is an irreducible subset of states for matrix $\mathbf{Q}^{\mathcal{D}}$.*

Proof. Let $s^{\mathcal{D}} = proj(s, \mathcal{J}, \mathcal{D})$ and $t^{\mathcal{D}} = proj(t, \mathcal{J}, \mathcal{D})$. We first show that $\mathbf{Q}(s,t) > 0$ implies $\mathbf{Q}(s^{\mathcal{D}}, t^{\mathcal{D}}) > 0$ for $s^{\mathcal{D}} \neq t^{\mathcal{D}}$. If $\mathbf{Q}(s,t)$ results from a local transition, then $\mathbf{Q}_l^{(j)}(s_j, t_j)$ for some $j \in \mathcal{D}$ implies that the same local transition is possible in the aggregated system and $\mathbf{Q}^{\mathcal{D}}(s^{\mathcal{D}}, t^{\mathcal{D}}) > 0$. If $\mathbf{Q}(s,t)$ results from a synchronized transition, then $\mathbf{Q}_e^{(j)}(s_j, t_j) > 0$ for all $j \in \mathcal{J}$. Since additionally $\mathbf{x}(s) > 0$, $\mathbf{A}^{\mathcal{D}}(s^{\mathcal{D}}, s^{\mathcal{D}}) > 0$, which implies $\mathbf{Q}^{\mathcal{D}}(s^{\mathcal{D}}, t^{\mathcal{D}}) > 0$. Thus, the existence of a path from s to t in the original CTMC implies the existence of a path from $s^{\mathcal{D}}$ $(= proj(s, \mathcal{J}, \mathcal{D}))$ to $t^{\mathcal{D}}$ $(= proj(t, \mathcal{J}, \mathcal{D}))$ in the aggregated CTMC.

It remains to show that a path from $s^{\mathcal{D}}$ to $t^{\mathcal{D}}$ in the aggregated CTMC implies the existence of a path from some s (with $s^{\mathcal{D}} = proj(s, \mathcal{J}, \mathcal{D})$) to some t (with $t^{\mathcal{D}} = proj(t, \mathcal{J}, \mathcal{D})$) such that $s, t \in \bar{\mathcal{S}}$. This can be done by showing that $\mathbf{Q}^{\mathcal{D}}(s^{\mathcal{D}}, t^{D}) > 0$ $(s^{\mathcal{D}} \neq t^{\mathcal{D}})$ implies $\mathbf{Q}(s,t) > 0$ for $(s, t \in \bar{\mathcal{S}})$. If $\mathbf{Q}^{\mathcal{D}}(s^{\mathcal{D}}, t^{D})$ results from a local transition, then this transition occurs in a component $j \in \mathcal{D} \subseteq \mathcal{J}$ and can also occur in the original CTMC. If the transition results from a synchronized transition, then $\mathbf{Q}_e^{(j)}(s_j^{\mathcal{D}}, t_j^{\mathcal{D}}) > 0$ for $j \in \mathcal{D}$. $\mathbf{A}_e^{\mathcal{D}}(s^{\mathcal{D}}, s^{\mathcal{D}}) > 0$ implies the existence of a state $s \in \bar{\mathcal{S}}$ with $s^{\mathcal{D}} = proj(s, \mathcal{J}, \mathcal{D})$ since $\mathbf{x}(s) > 0$ has to hold to assure $\mathbf{Q}^{\mathcal{D}}(s^{\mathcal{D}}, t^{\mathcal{D}}) > 0$. Additionally, $\mathbf{Q}_e^{(j)}(s_j, t_j) > 0$ for $j \in \mathcal{D}$, which implies $\mathbf{Q}(s,t) > 0$ for some $t \in \bar{\mathcal{S}}$ with $t^{\mathcal{D}} = proj(t, \mathcal{J}, \mathcal{D})$. □

The theorem indicates that matrix $\mathbf{Q}^{\mathcal{D}}$ can be used to compute the stationary solution of the aggregated CTMC. The following corollary shows the correspondence between the stationary solution at different levels.

COROLLARY 3.2. *If $\mathbf{Q}^{\mathcal{D}}$ has been generated using a probability vector $\mathbf{p}$ with $\mathbf{pQ} = \mathbf{0}$, then $\mathbf{p}^{\mathcal{D}}\mathbf{Q}^{\mathcal{D}} = \mathbf{0}$.*

Example (continued). Aggregation of one component in the example reduces the state space by a factor $k_j + 1$. If the first component is aggregated according to probability vector $\mathbf{x}$, then the elements of the matrices $\mathbf{A}_{12}^{(2,3)}$ and $\mathbf{A}_{13}^{(2,3)}$ are defined as

$$\mathbf{A}_{12}^{(2,3)}((s_2,s_3),(s_2,s_3)) = \mathbf{A}_{13}^{(2,3)}((s_2,s_3),(s_2,s_3)) = \lambda_1 \frac{\mathbf{x}(k_1,s_2,s_3)}{\sum_{s_1=0}^{k_1} \mathbf{x}(s_1,s_2,s_3)} ,$$

where $\mathbf{x}(k_1, s_2, s_3)$ is the conditional probability that the first queue contains k_1 customers when the queues 2 and 3 contain s_2 and s_3 customers, respectively.

4. A multilevel solution algorithm. With the steps presented in the previous section, it is straightforward to formulate a multilevel solution algorithm following the lines given by other multilevel solution techniques [2, 3, 18, 24, 25]. However, in contrast to other approaches applicable for CTMC analysis, aggregation along the multidimensional structure and exploitation of the Kronecker structure for building aggregated generator matrices introduce some convenient features, especially when applied to CTMCs with a large state space. Aggregated generator matrices are built by removing some component matrices from the Kronecker products and adding diagonal matrices $\mathbf{A}_e^{\mathcal{D}}$ to describe the effect of synchronized transitions with respect to aggregated dimensions in the state space.

The multilevel solution combines iterative analysis with aggregation-disaggregation steps to build aggregated CTMCs. Denote by $iter(\mathbf{Q}, \mathbf{x}, \nu)$ the computation of ν iteration steps, using the compact representation of matrix $\mathbf{Q}$ as given in (3) or (8), starting with vector $\mathbf{x}$. The result of this operation is a vector $\mathbf{y}$. We do not fix the iteration technique; in principle, all iterative techniques can be applied that exploit the Kronecker structure of $\mathbf{Q}$. Some remarks on choosing an appropriate technique are given below. We use $solve(\mathbf{Q}, \mathbf{x})$ for the computation of the solution $\mathbf{xQ} = \mathbf{0}$ subject to $\|\mathbf{x}\|_1 = 1$. Usually this solution is computed with a direct method, such as the GTH algorithm [25], which implies that $\mathbf{Q}$ is generated as a whole. If $\mathbf{Q}$ contains several irreducible subsets of states, we assume that the solution is computed with respect to one irreducible subset, denoted as $\bar{\mathcal{S}}$ and $\bar{\mathcal{S}}^{\mathcal{D}}$, respectively.

The following function performs the basic steps of the multilevel solution approach which will be denoted as a multilevel cycle.

function $multi\_level\_algo(\mathbf{Q}^{\mathcal{D}}, \mathbf{x}^{\mathcal{D}}, \mathcal{D})$
  if $|\mathcal{D}| = 1$ then
    $\mathbf{y}^{\mathcal{D}} = solve(\mathbf{Q}^{\mathcal{D}}, \mathbf{x}^{\mathcal{D}})$; (step 1)
  else
    $\mathbf{x}^{\mathcal{D}} = iter(\mathbf{Q}^{\mathcal{D}}, \mathbf{x}^{\mathcal{D}}, \nu_1)$; (step 2)
    find $i = \min_{i \in \mathcal{D}}(\|\mathbf{r}^{(i)}\|$,
      where $\mathbf{r}^{(i)}$ results from $\mathbf{r}^{\mathcal{D}} = \mathbf{x}^{\mathcal{D}}\mathbf{Q}^{\mathcal{D}}$ via (5)) ; (step 3)
    $\mathcal{C} = \mathcal{D} \setminus (i)$;
    compute $\mathbf{x}^{\mathcal{C}}$ from $\mathbf{x}^{\mathcal{D}}$ using (5) ; (step 4)
    compute $\mathbf{Q}^{\mathcal{C}}$ from $\mathbf{Q}^{\mathcal{D}}$ and $\mathbf{x}^{\mathcal{D}}$ using (7) and (8) ; (step 5)
    $\mathbf{y}^{\mathcal{C}} = multi\_level\_algo(\mathbf{Q}^{\mathcal{C}}, \mathbf{x}^{\mathcal{C}}, \mathcal{C})$; (step 6)
    compute $\mathbf{y}^{\mathcal{D}}$ from $\mathbf{x}^{\mathcal{D}}$, $\mathbf{x}^{\mathcal{C}}$ and $\mathbf{y}^{\mathcal{C}}$ via (6) ; (step 7)
    $\mathbf{y}^{\mathcal{D}} = iter(\mathbf{Q}^{\mathcal{D}}, \mathbf{y}^{\mathcal{D}}, \nu_2)$; (step 8)
  fi
  return($\mathbf{y}^{\mathcal{D}}$) ;
end

A complete solution requires the following steps:

initialize $\mathbf{x}$ such that $\mathbf{x}(s) > 0$ for $s \in \bar{\mathcal{S}}$ and $\mathbf{x}(s) = 0$ otherwise ;
repeat
  $\mathbf{x} = multi\_level\_algo(\mathbf{Q}, \mathbf{x}, \mathcal{J})$;
until $\|\mathbf{xQ}\| \leq \epsilon$.

The solution stops if the norm of the residual vector is smaller than some predefined constant ϵ. Other stopping criteria can be applied as well [25].

In step 1 the aggregated system for a single component is solved; the size of this system is at most $\max_{j \in \mathcal{J}} n_j$. Usually, this system is small enough to be solved with a direct solver. If this is not the case, then an iterative technique can be used. In step 2 and step 8 iteration steps are performed to improve the approximation of the solution. We denote this as preiteration and postiteration. Every iteration method that can be used in conjunction with the compact matrix representation is applicable in these steps. Methods with a relatively smooth convergence should be chosen because ν_1 and ν_2 are usually relatively small. It is, of course, possible to use different iterative techniques at different levels. Values for ν_1 and ν_2 can be fixed or can be chosen adaptively during the iteration depending on the observed convergence behavior. It is not necessary to solve aggregated CTMCs with a high accuracy, unless the residuals $\mathbf{r}^{\mathcal{J}}$ are relatively small. In step 3 the component or dimension to be aggregated is chosen. In the proposed realization, this choice is adaptive by aggregating the

component with the smallest local residuals. The underlying assumption is that small residuals in one dimension indicate a good approximation in this dimension. It is also possible to aggregate more than one component in a step, e.g., by aggregating all components where the local residual norm is less than some predefined threshold. Step 6 describes the recursive call of the procedure to compute the approximation for the aggregated system. As presented here, the multilevel solution approach realizes a *V*-cycle. In multigrid techniques *W*-cycles are also used by making two consecutive recursive calls [3].

The method is related to aggregation-disaggregation methods for the solution of CTMCs, and it is also related to block Jacobi and block Gauss–Seidel, which have in [14] shown to be very efficient solvers for large CTMCs. Global convergence of aggregation-disaggregation techniques applied to CTMCs has been recently proved [20]. By adopting a simple modification proposed in [24], convergence of the multilevel method can also be ensured whenever the iterative method used for the complete CTMC converges. To realize the modification, the number of iterations ν_1 is chosen adaptively for $\mathcal{D} = \mathcal{J}$ such that the residuals of the result vector are smaller than the residuals of the initial vectors. Then the following steps have to be introduced:

if $\mathcal{D} = \mathcal{J}$ then $\mathbf{z}^{\mathcal{J}} = \mathbf{x}^{\mathcal{J}}$; (step 2.1)
if $\mathcal{D} = \mathcal{J}$ and $\|\mathbf{y}^{\mathcal{J}}\mathbf{Q}^{\mathcal{J}}\| > \|\mathbf{z}^{\mathcal{J}}\mathbf{Q}^{\mathcal{J}}\|$ then (step 8.1)
$\mathbf{y}^{\mathcal{J}} = \mathbf{z}^{\mathcal{J}}$. (step 8.2)

The result of the multilevel step is used only if the residual norm is not increased by the multilevel step. Thus, the residual norm of the iteration vector of the complete CTMC is decreased by the iteration steps and it is not increased by a multilevel step, which implies convergence of the method whenever the iteration method converges. If, for example, the Power method is used for the complete system and the modification is introduced, then the complete method will converge.

5. Some examples. We begin with relatively small versions of the overflow QN examples with three queues and 20 queueing places in each of the queues. State spaces of the components contain 21 states; the state spaces of the complete CTMCs include 9,261 states, and generator matrices have 62,181 nonzero entries. To represent these matrices in compact form, 18 matrices with 304 nonzero entries overall are necessary for version $V1$. For version $V2$, the compact representation requires 36 matrices with 418 nonzero entries. The complete generator matrix contains 63,501 nonzero entries for this example. In both cases, the Kronecker representation needs significantly less space. The examples are small enough to be analyzed with different analysis techniques, including LU factorization, although the direct solution with LU factorization requires much more time than the solution with an iterative technique. For the first series of experiments we choose $\mu_i = 1.0$, $\lambda_1 = 0.5$, $\lambda_2 = 0.7$, and $\lambda_3 = 0.9$.

All experiments have been performed on a PC with a 450 MHz CPU and 128 MB main memory. Programs are written in C, and CPU times are measured with the available C library functions. We compare three different classes of solution techniques. For details about the basic analysis methods we refer to [25], which describes all used methods except BiCGSTAB and TFQMR, which can be found in [17, 28]. The first class includes conventional techniques that are based on the sparse matrix representation of $\mathbf{Q}$. In this class we use the well-known LU factorization (*LU*), the Power method (*Power*), successive overrelaxation (*SOR*), and the generalized minimal residual method with a restart after 20 iterations (*GMRES*), as well as *BiCGSTAB* and *TFQMR*. For the latter three methods ILU0 preconditioning is used to acceler-

TABLE 1
Results for example $V1$.

Method	Iter.	$\|xQ\|_1$	$\|xQ\|_\infty$	CPU-time in sec.
LU	–	$3.296e-17$	$1.617e-15$	550
Power	3163	$9.999e-11$	$5.756e-9$	43
SOR($\omega = 1.8$)	133	$9.653e-11$	$8.841e-10$	1
BiCGStab + ILU0	58	$2.040e-11$	$1.407e-9$	4
TFQMR +ILU0	60	$4.996e-11$	$5.034e-9$	4
GMRES+ILU0	60	$3.824e-11$	$1.730e-8$	5
StPower	3163	$9.999e-11$	$5.756e-9$	52
StJOR ($\omega = 1.0$)	2507	$9.943e-11$	$9.027e-9$	40
StGS	1309	$9.926e-11$	$5.425e-9$	34
StSOR ($\omega = 1.8$)	133	$9.653e-11$	$8.841e-10$	3
StBiCGStab	172	$5.140e-11$	$5.277e-9$	4
StTFQMR	464	$1.635e-11$	$1.926e-9$	12
GMRES ($m = 20$)	300	$3.383e-11$	$1.115e-8$	12
MlPower	81	$8.842e-11$	$7.028e-9$	3
MlJOR ($\omega = 0.95$)	189	$9.989e-11$	$1.730e-8$	6
MlGS	46	$9.630e-11$	$1.754e-8$	2
MlSOR ($\omega = 1.65$)	40	$9.291e-11$	$8.602e-9$	2

ate convergence [25]. In the second class we consider iterative methods exploiting the compact matrix representation. We use the Power method, JOR, Gauss–Seidel, SOR, GMRES, BiCGStab, and TFQMR, denoted as *StPower, StJOR, StGS, StSOR, StGMRES, StBiCGStab*, and *StTFQMR*, respectively. The third class contains the multilevel methods based on the Power methods, JOR, Gauss–Seidel, and SOR. These methods are denoted as *MlPower, MlJOR, MlGS*, and *MlSOR*. For all multilevel techniques at the outermost level, i.e., for the complete system, one postiteration step and no preiteration steps are performed. For aggregated systems 10 preiteration and 10 postiteration steps are performed. Experiments indicate that a small number of outer iterations usually improves solution times. For JOR and SOR, optimal relaxation parameters are determined by a search algorithm such that the results for these methods are best cases that can hardly be achieved in practice.

Results for example $V1$ are shown in Table 1. Iterations are stopped when the maximum absolute value of the residual vector becomes smaller than 10^{-10}. The first column of the table contains the solution method including the value of the relaxation parameter if necessary followed by the number of vector matrix products in the second column. Vector matrix products using aggregated matrices are not counted since they are relatively cheap, which can be seen by comparing the time per iteration for the structured methods and the multilevel methods. A single vector matrix product for the structured Power method requires about 0.016 seconds, whereas a single iteration of the multilevel Power method requires 0.033 seconds, which is twice the time. However, the latter iteration time contains the time to perform 20 iterations at the first aggregated level with 2 components and the time for the direct solution of the aggregated system with one component. Columns 3 and 4 include the norms for the residual vectors after termination of the solution method. The last column includes CPU-times in seconds required for the solution. These values describe the complete solution times including times for preconditioning or aggregation-disaggregation. Comparison of CPU-times is sometimes inconclusive since CPU-times are implementation dependent. However, one should be aware that the implementations for the methods from class 1 and also partially from class 2 are optimized realizations that are used for a long time. Implementation of the multilevel

TABLE 2
Results for example V2.

Method	Iter.	$\|xQ\|_1$	$\|xQ\|_\infty$	CPU-time in sec.
LU	–	$6.614e-18$	$1.103e-15$	547
Power	4373	$9.982e-11$	$5.973e-9$	61
SOR ($\omega = 1.85$)	197	$9.714e-11$	$5.209e-8$	2
BiCGStab + ILU0	68	$1.641e-11$	$1.788e-8$	4
TFQMR + ILU0	80	$2.054e-11$	$1.946e-9$	4
GMRES + ILU0	80	$3.859e-12$	$9.904e-10$	6
StPower	4373	$9.982e-11$	$5.973e-9$	114
StJOR ($\omega = 1.0$)	4203	$9.977e-11$	$9.409e-9$	113
StGS	2048	$9.978e-11$	$1.846e-8$	85
StSOR ($\omega = 1.85$)	197	$9.714e-11$	$5.209e-8$	8
StBiCGStab	226	$8.365e-12$	$5.696e-9$	7
StTFQMR	256	$5.713e-11$	$5.519e-8$	9
GMRES ($m = 20$)	298	$9.697e-11$	$9.847e-9$	14
MlPower	854	$9.885e-11$	$2.898e-7$	45
MlJOR ($\omega = 0.95$)	1239	$9.864e-11$	$8.964e-8$	71
MlGS	310	$9.588e-11$	$1.235e-7$	26
MlSOR ($\omega = 1.7$)	82	$9.175e-11$	$1.960e-8$	7

techniques exploits for the iteration steps the same iteration procedures as the structured methods. The aggregation-disaggregation parts are prototype implementations that probably can be optimized. Consequently, CPU-times present in some sense the worst case results for the multilevel solutions.

By comparing CPU-times of *Power* with *StPower* and of *SOR* with *StSOR*, the overhead introduced by exploiting the compact matrix representation can be seen. The overhead is moderate for the Power method and for all other iterative methods based on simple computations of the product of the original matrix $\mathbf{Q}$ with the iteration vector. The overhead is much larger for SOR based on the iteration procedure presented in [27]. Convergence of the Power method, JOR, and GS is slow. SOR behaves much better. The three projection methods with ILU0 preconditioning behave even better than SOR, at least in counting the number of iterations. Without ILU0 preconditioning, which cannot be applied without destroying the compact matrix representation [6, 7], projection methods require additional effort but are still more efficient than all of the other methods except SOR.

The multilevel methods behave very well for this example. The number of iterations and also the solution times are reduced significantly in all cases by introducing multilevel steps. With the exception of MlJOR, multilevel methods are faster than the other solution methods, excluding SOR. With multilevel steps, even the Power method and GS become efficient solvers for this model.

Table 2 contains the results for model $V2$ with arrival rates 1.0, 0.95, and 0.9, respectively. This example contains nine instead of three synchronized events. Consequently, one can expect a larger overhead for the structured methods because the handling of synchronized transitions with sparse matrices belonging to the events is costly. This effect can be observed by comparing the time per iteration for the structured and the conventional methods. For this example, a structured iteration using matrix $\mathbf{Q}$ requires roughly twice the time of a conventional iteration. For the SOR iteration step this difference is increased to a factor of nearly 5. Due to increased arrival rates, the components are more tightly coupled than in the first version. Thus, we can also expect less gain from the use of multilevel steps. In fact, this behavior can be observed. Multilevel steps decrease the number of iterations, and they also

TABLE 3
Results for the large version of example V2.

Method	Iter.	$\|xQ\|_1$	$\|xQ\|_\infty$	CPU-time in sec.	Memory in MB
StPower	1000	$2.666e-7$	$2.377e-4$	19054	39
StJOR ($\omega = 1.0$)	1000	$2.008e-8$	$3.180e-4$	18260	39
StGS	586	$9.917e-9$	$8.733e-6$	25321	39
StSOR ($\omega = 1.75$)	90	$9.965e-9$	$6.245e-6$	3901	39
StBiCGStab	128	$4.204e-9$	$1.080e-5$	2484	76
StTFQMR	492	$4.608e-9$	$1.389e-5$	9665	84
MlPower	15	$7.674e-9$	$6.330e-5$	890	63
MlJOR ($\omega = 0.8$)	11	$5.883e-9$	$8.669e-5$	663	63
MlGS	9	$9.652e-9$	$1.271e-4$	726	63
MlSOR ($\omega = 1.25$)	9	$7.761e-9$	$9.973e-5$	724	63

decrease the solution times of the same method without multilevel steps; however they still require more time than conventional methods and, with the exception of *MlSOR*, which is the fastest solver using the compact matrix representation, they require more iterations than the projection methods.

The major advantage of the compact representation of $\mathbf{Q}$ is that it allows us to solve larger models. We present now two examples that are too large to be analyzed with methods using a sparse matrix representation. The first large example is an extension of $V2$. We consider a configuration with 6 queues and 10 queueing places per queue for the queues 1 through 5. The last queue has 5 buffer places. The resulting CTMC contains 966,306 states and matrix $\mathbf{Q}$ has 10,995,385 nonzero entries. With 8 bytes per double precision value and 4 bytes per integer, the sparse representation of the matrix requires about 132 MB memory, too much for the available PC. The compact representation requires 252 matrices with 960 nonzero entries. Although this representation is compact, it is relatively complex because a large number of matrices are necessary due to 30 synchronized transitions in the model. Complexity can be reduced by collecting all diagonal elements in a vector of length 966,306. In this way storage requirements are slightly increased, but only 126 matrices with 510 nonzero entries are now necessary to describe the nondiagonal elements of $\mathbf{Q}$. The mean service time in all queues equals 1.0. The arrival rate of the queues 1 through 5 equals 0.5, 0.6, 0.7, 0.8, and 0.9. The last queue is overloaded with an arrival rate of 1.5.

Table 3 contains the results for this example. The last column of the table shows the memory required by the different solution methods when diagonal elements are stored in a vector. Memory requirements include the whole memory the solution program occupies. For this example the iterations stops when the largest value of the residual vector becomes smaller than 10^{-8} or when 1,000 iterations have been performed. The Power method and JOR do not reach the required accuracy within 1,000 iterations. The other methods reach the required accuracy, but with significant differences in the number of iterations. SOR and BiCGStab are the best structured methods in term of iterations and CPU-time, respectively. Projection methods BiCGStab and TFQMR require additional vectors for the solution and therefore also additional memory that is roughly twice the memory needed by the stationary iterative methods. GMRES cannot be applied for this example because it needs too much memory, even with the compact matrix representation. The large state space and the large number of synchronization events causes very long solution times. BiCGStab is the only method without multilevel steps which solves the system in less than an hour. We did not analyze the example with block SOR, which probably is more ef-

ficient than SOR as indicated in [14, 27]. However, since the multilevel versions of GS and SOR are extremely efficient for this example, it is very unlikely that block SOR will outperform these methods for this example. The use of multilevel steps also increases memory requirements due to the storage of additional vectors, but the multilevel methods require less memory than the projection methods. Multilevel steps drastically reduce the number of iterations and also the solution time for this example. Thus, all multilevel techniques solve the system with at most 15 iterations needing less than a quarter of an hour for the solution. GS and SOR require only 9 outer iterations, which is an extremely good result for this large model.

The last example describes a simplified version of a manufacturing system with kanban control [21]. We consider a model where each component describes a cell including a single queue and an input and output buffer. The capacity of cell j equals k_j. The first cell always contains k_1 parts that are either processed in the queue or are waiting to enter the second cell. Parts which have been processed in cell j are allowed to enter cell $j+1$ if less than k_{j+1} parts are currently in cell $j+1$. Parts leave the system immediately after leaving the queue in the last cell; parts that leave the first cell are immediately substituted with new parts arriving to the queue in the first cell. Processing and traveling times between cells are exponentially distributed with rates μ and ω, respectively. In our example we consider a system with 4 queues, $k_j = 10$, $\mu = 1.0$, and $\omega = 10.0$. The first and last components have 11 states, the remaining two components 66 states. The complete CTMC has 527,076 states and 3,528,481 nonzero entries in $\mathbf{Q}$. The compact representation requires 20 matrices with 740 nonzero entries. By representing diagonal entries in a vector, the nondiagonal part of $\mathbf{Q}$ can be described with 10 matrices and 370 nonzero entries. Although this model has only 4 synchronized transitions, components are strongly connected because the model is a tandem system where parts start in the first component and travel consecutively through all components before leaving the system. Results for this example are shown in Table 4. For this example the iteration stops when the maximum value of the residual vector becomes smaller than 10^{-8} or 2,000 iterations have been performed. Without multilevel steps only the three projection methods and SOR reach the required accuracy. Iterations are performed much faster than in the previous example, which is caused by the smaller number of states and, in particular, by the much smaller number of synchronized transitions. As in the previous example, the introduction of multilevel steps reduces the number of iterations and also the solution time of all methods. Only the Power method with multilevel steps requires a similar amount of time for the solution as the projection methods BiCGStab and TFQMR need. The remaining multilevel methods, including GS, are significantly faster. As in the previous example we did not apply block SOR, which usually has a faster convergence than SOR. Additionally, preconditioning may be used to improve the convergence of projection methods. Unfortunately, generation of efficient preconditioners preserving the compact matrix representation is still an open research problem [25].

To summarize the results of the presented examples and also of some other examples not shown here, multilevel steps reduce the solution effort in terms of iterations and CPU-time of the corresponding iteration technique in all cases we tested so far. However, several aspects still require additional experiments. The number of iterations, which was chosen as a fixed value in the experiments presented here, might be set adaptively by observing the convergence behavior. Additionally, other types of aggregation-disaggregation cycles like W instead of V cycles should be tested. Since projection methods like BiCGStab and TFQMR are efficient solvers for many

TABLE 4
Results for the kanban system.

Method	Iter.	$\|xQ\|_1$	$\|xQ\|_\infty$	CPU-time in sec.	Memory in MB
StPower	2000	$1.854e-5$	$3.313e-3$	2752	22
StJOR ($\omega = 1.0$)	2000	$9.183e-8$	$7.787e-5$	2759	22
StGS	2000	$2.207e-6$	$1.665e-3$	8655	22
StSOR ($\omega = 0.95$)	1823	$9.086e-9$	$6.492e-6$	7886	22
StBiCGStab	1006	$5.131e-10$	$1.160e-6$	1681	42
StTFQMR	958	$7.097e-9$	$4.387e-6$	1795	46
StGMRES	1040	$1.157e-8$	$3.583e-6$	3203	100
MlPower	522	$8.931e-9$	$1.814e-6$	1668	24
MlJOR ($\omega = 0.75$)	81	$9.908e-9$	$2.031e-5$	232	24
MlGS	75	$8.316e-9$	$6.672e-6$	348	24
MlSOR ($\omega = 0.8$)	57	$9.577e-9$	$1.111e-5$	269	24

Markov models, multilevel steps may be combined with these techniques. However, our first experience with this combination, where after some iterations with BiCGStab or TFQMR a multilevel cycle was introduced, were not as encouraging as the combination of multilevel steps with stationary iterative methods. The reason seems to be that the convergence of projection methods is usually not as regular as the convergence of stationary iterative methods and a smooth convergence seems to be necessary for the efficient use of multilevel steps.

6. Conclusion. This paper presents a new solution approach for the analysis of continuous time Markov chains with a multidimensional structure. Many Markov chains can be described in a multidimensional form by considering the model as a set of interacting components. The solution exploits the multidimensional structure and defines aggregated chains by keeping the distribution in some dimensions constant and aggregating these dimensions. Solution vectors for aggregated chains are used to redirect the solution vector for the complete chain. In this way, it is possible to reduce the number of time-consuming iterations with the complete system and speed up iterative solution techniques. The idea of multilevel solutions can also be found in multigrid techniques for differential equations and a multilevel solution algorithm for Markov chains. However, the concrete realization differs since the proposed approach exploits the Kronecker structure of a structured Markov chain that provides a very compact and easy-to-generate representation of aggregated generator matrices. Furthermore the multidimensional structure, which usually corresponds to the structure of the modeled system, defines a natural way to build aggregated CTMCs.

Different examples show that the multilevel solution speeds up the convergence of iterative techniques without increasing the required memory too much. In particular, models with loosely coupled components can be analyzed much more efficiently because the aggregated solutions are good approximations of the mapping of the stationary solution on the aggregated state space. However, even for strongly interacting components, solution time is usually reduced by the multilevel solution.

The proposed approach can be extended in different directions. Other iterative techniques can be enhanced by multilevel steps. It is possible to adopt other aggregation strategies as in multigrid techniques. Iteration techniques can be combined with preconditioning, although efficient preconditioners respecting the Kronecker structure are still an open research problem (see [7, 25, 26]).

Additionally, the technique can be applied to slightly more general models. Discrete time Markov chains with a multidimensional structure are handled similarly. In

[23] stochastic automata networks in discrete time are introduced. The corresponding transition matrix has a Kronecker structure similar to the one proposed for generator matrices here. Thus, the multilevel solution approach can be applied to this model class as well. Another extension, which has been proposed in the context of stochastic automata, are state dependent transition rates. In this case, the rate of a transition in a component may depend on the state of other components. Since state dependent transitions rates in SANs can be transformed into synchronized transitions without state dependency as shown in [22], they can be handled with the proposed solution algorithms, although an efficient handling requires a more sophisticated realization without explicit transformation of state dependent rates into synchronized transitions. An efficient algorithm for the computation of vector matrix products in stochastic automata networks with state dependent transitions is introduced in [16] and can be used in iterative methods combined with multilevel steps.

Acknowledgments. I thank the anonymous referees for their very detailed comments on the previous version of the paper.

REFERENCES

[1] G. Birkhoff and R. E. Lynch, *Numerical Solution of Elliptic Problems*, SIAM Stud. Appl. Math. 6, SIAM, Philadelphia, 1984.

[2] A. Brandt, *Multi-level adaptive solutions to boundary value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] W. L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[4] P. Buchholz, *An aggregation/disaggregation algorithm for stochastic automata networks*, Probab. Engrg. Inform. Sci., 11 (1997), pp. 229–253.

[5] P. Buchholz, *An adaptive aggregation/disaggregation algorithm for hierarchical Markovian models*, European J. Oper. Res., 116 (1999), pp. 85–104.

[6] P. Buchholz, *Structured analysis approaches for large Markov chains*, Appl. Numer. Math., 31 (1999), pp. 375–404.

[7] P. Buchholz, *Projection methods for the analysis of stochastic automata networks*, in Numerical Solution of Markov Chains, B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, 1999, pp. 149–168.

[8] P. Buchholz, G. Ciardo, S. Donatelli, and P. Kemper, *Complexity of Kronecker operations and sparse matrices with applications to the solution of Markov models*, INFORMS J. Comput., to appear.

[9] P. E. Buis and W. R. Dyksen, *Efficient vector and parallel manipulation of tensor products*, ACM Trans. Math. Software, 22 (1996), pp. 18–23.

[10] R. Chan, *Iterative methods for overflow queueing networks* I, Numer. Math., 51 (1987), pp. 143–180.

[11] R. Chan, *Iterative methods for overflow queueing networks* II, Numer. Math., 54 (1988), pp. 57–78.

[12] R. Chan and W. Ching, *Circulant preconditioners for stochastic automata networks*, Numer. Math., to appear.

[13] M. Davio, *Kronecker products and shuffle algebra*, IEEE Trans. Comput., 30 (1981) pp. 116–125.

[14] T. Dayar and W. J. Stewart, *Comparison of partitioning techniques for two-level iterative solvers on large, sparse Markov chains*, SIAM J. Sci. Comput., 21 (2000), pp. 1691–1705.

[15] S. Donatelli, *Superposed generalized stochastic Petri nets: Definition and efficient solution*, in Application and Theory of Petri Nets, Zaragoza, 1994, R. Valette, ed., Lecture Notes in Comput. Sci. 815, Springer-Verlag, Berlin, 1994, pp. 258–277.

[16] P. Fernandes, B. Plateau, and W. J. Stewart, *Efficient descriptor-vector multiplication in stochastic automata networks*, J. ACM, 45 (1998), pp. 381–414.

[17] R. W. Freund and M. Hochbruck, *On the use of two QMR for solving singular systems and applications in Markov chain modelling*, Numer. Linear Algebra Appl., 1 (1994), pp. 403–420.

[18] G. Horton and S. Leutenegger, *A multi-level solution algorithm for steady state Markov-chains*, ACM Perform. Eval. Rev., 22 (1994), pp. 191–200.

[19] L. Kaufman, *Matrix methods for queuing problems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 525–552.

[20] I. Marek and P. Mayer, *Convergence analysis of an iterative aggregation/disaggregation method for computing the probability vector of stochastic matrices*, Numer. Linear Algebra Appl., 5 (1998), pp. 253–274.

[21] D. Mitra and I. Mitrani, *Analysis of a kanban discipline for cell coordination in production lines* II: *Stochastic demands*, Oper. Res., 39 (1991), pp. 807–823.

[22] B. Plateau, *On the stochastic structure of parallelism and synchronisation models for distributed algorithms*, ACM Perform. Eval. Rev., 13 (1985), pp. 142–154.

[23] B. Plateau and K. Atif, *Stochastic automata networks for modeling parallel systems*, IEEE Trans. Software Engrg., 17 (1991), pp. 1093–1108.

[24] P. Schweitzer and K. W. Kindle, *An iterative aggregation-disaggregation algorithm for solving linear equations*, Appl. Math. Comput., 18 (1986), pp. 313–353.

[25] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

[26] W.J. Stewart, K. Atif, and B. Plateau, *The numerical solution of stochastic automata networks*, European J. Oper. Res., 86 (1995), pp. 503–525.

[27] E. Uysal and T. Dayar, *Iterative methods based on splittings for stochastic automata networks*, European J. Oper. Res., 110 (1998), pp. 166–186.

[28] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 358–363

ON THE ALMOST STRONG STABILITY OF THE CIRCULAR DECONVOLUTION ALGORITHM*

PLAMEN Y. YALAMOV†

Abstract. The stability of the circular deconvolution algorithm for the solution of a circulant linear system is studied. This algorithm is known to be not strongly stable. The notion of almost strong stability is introduced, and it is shown that it leads to results similar to those for strongly stable algorithms. Then it is proved that the circular deconvolution algorithm based on fast Fourier transforms is almost strongly stable with respect to the 2-norm. A numerical example illustrates the theoretical conclusions.

Key words. circulant deconvolution, circulant matrix, stability, fast Fourier transform

AMS subject classifications. 65G05, 65T20

PII. S0895479899351052

1. Introduction.

We are interested in the solution of a linear system

$$H_c x = b, \tag{1.1}$$

where

$$H_c = \begin{pmatrix} c_0 & c_{n-1} & \dots & c_1 \\ c_1 & & \dots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \dots & c_0 \end{pmatrix}$$

is a circulant matrix. Such problems arise in various applications, e.g., geophysical inversion problems [7, 8], and solution of banded [5] and dense [9] Toeplitz systems.

It is well known (e.g., [10, section 4.2.2], [11]) that circulant matrices can be decomposed as

$$H_c = F_n^{-1} \Lambda F_n, \quad \Lambda = \operatorname{diag}(F_n c),$$

where F_n is the Fourier matrix of order n, i.e.,

$$F_n = \frac{1}{\sqrt{n}} \left\{ e^{-\frac{i2\pi jk}{n}} \right\}_{j,k=0}^{n-1}, \quad i = \sqrt{-1}.$$

Thus the solution of (1.1) is done in $O(n \log_2 n)$ steps by using the fast Fourier transform (FFT) according to the following algorithm:

Step 1. $d = Fc$,
Step 2. $e = Fb$,
Step 3. $f = e./d$ (i.e., $f_i = e_i/d_i, \quad i = 0, \dots, n-1$),
Step 4. $x = F^{-1} f$.

*Received by the editors February 4, 1999; accepted for publication (in revised form) by L. Reichel January 13, 2000; published electronically July 11, 2000. This research was supported by grants MM-707 and I-702 from the National Scientific Research Fund of the Bulgarian Ministry of Education and Science.

http://www.siam.org/journals/simax/22-2/35105.html

†Center of Applied Mathematics and Informatics, University of Rousse, 7017 Rousse, Bulgaria (yalamov@ami.ru.acad.bg).

We assume that the FFT is implemented by the Cooley–Tukey algorithm (see [10] for more details).

One of the important questions in applications is the numerical stability of the proposed algorithm with respect to roundoff errors. This problem is studied in [4, section 23.2] and [6]. In [6] it is shown that the algorithm is forward stable (see [4, section 1.5] for definitions of forward and backward stability) with respect to the 2-norm and a special condition number. This is the weakest type of stability for any numerical algorithm. Then in [4, section 23.2] a stronger result is proved. Namely, the algorithm is normwise backward stable. Matrix H_c has a special structure, and its entries are defined by n parameters. This is not taken into account in the roundoff error analysis of [4, section 23.2], and the equivalent perturbations from the backward analysis in matrix H are not structured. According to the classification proposed in [1] this algorithm is not strongly stable, i.e., determining small equivalent perturbations in vector $c = (c_0, c_1, \ldots, c_{n-1})^T$ (and perhaps in the right-hand side b) is not possible. There is a numerical example in [6] which shows this fact.

In the present paper we define a new type of stability, i.e., almost strong stability. We show that strong and almost strong stability are close in some sense. Then we prove that the algorithm presented in this paper is almost strongly stable. The theoretical result is illustrated by the numerical example taken from [6].

2. Types of stability. Let us review in brief the types of stability discussed in [1]. Throughout the paper we adopt the standard model of floating-point arithmetic with a guard digit:

$$f(x * y) = (x * y)(1 + \sigma), \quad |\sigma| \le \rho_0,$$

where ρ_0 is the machine roundoff unit. By a tilde we denote computed results in the following definition.

DEFINITION 2.1. *An algorithm is weakly stable for the class of circulant matrices Ω if for every well-conditioned $H_c \in \Omega$, the computed solution $\tilde{x}$ is close to x, i.e.,*

$$\frac{\|\widetilde{x} - x\|}{\|x\|} \le C\rho_0,$$

where $C \ll 1/\rho_0$.

DEFINITION 2.2. *An algorithm for solving circulant systems is strongly stable if the quantity*

$$\eta = \min\{\varepsilon : H_{c+\Delta c}\widetilde{x} = b + \Delta b, \quad \|\Delta c\| \le \varepsilon \|c\|, \quad \|\Delta b\| \le \varepsilon \|b\|\} \tag{2.1}$$

exists, and $\eta \ll 1$.

It is clear that if an algorithm is strongly stable on the class of circulant matrices, then it is also weakly stable on the class of circulant matrices. This can be shown after some standard manipulation.

There is no general formula for η, but it can be computed numerically as a solution to a constrained optimization problem. Let us consider the following numerical example (taken from [6]):

$$c = (1.5 + \mu \quad 0.5 - \mu \quad \mu - 0.5 \quad 0.5 - \mu)^T,$$

$$b = (2 \quad 1 \quad 2 \quad 1)^T, \tag{2.2}$$

where μ is chosen to be small. In [6] it is shown that the circular deconvolution algorithm is not strongly stable. The author shows that the structured backward error grows very fast when μ decreases. We also tried to solve the minimization problem (2.1) in MATLAB for small μ. The algorithm failed. The reason is that for μ small we are close to a problem having no solution.

Now let us introduce a new type of stability.

DEFINITION 2.3. *An algorithm is almost strongly stable if the quantity*

$$\omega = \min \{\varepsilon : H_{c+\Delta c}(\widetilde{x} + \Delta x) = b + \Delta b, \\ \|\Delta c\| \le \varepsilon \|c\|, \|\Delta b\| \le \varepsilon \|b\|, \quad \|\Delta x\| \le \varepsilon \|x\|\} \tag{2.3}$$

exists and $\omega \ll 1$.

This new definition allows more degrees of freedom in the optimization problem. So, there are larger chances that we find a solution to (2.3). As we will see in the next section the perturbations Δc, Δb, and Δx exist and can be bounded by not large constants for the algorithm considered in this paper.

Let us note that now we allow perturbations in the computed solution $\widetilde{x}$ as well. The next theorem shows that the bound on the forward error is not changed essentially in this case.

THEOREM 2.4. *If* $H_{c+\Delta c}(\widetilde{x} + \Delta x) = b + \Delta b$, *and* $\|\Delta c\|_2 \le \delta \|c\|_2, \|\Delta b\|_2 \le \delta \|b\|_2, \|\Delta x\|_2 \le \delta \|x\|_2$, *then*

$$\frac{\|\widetilde{x} - x\|_2}{\|x\|_2} \le \frac{2\kappa_2(H)\delta}{1 - \kappa_2(H)\delta} + \delta, \quad \kappa_2(H) = \|H^{-1}\|_2 \|H\|_2. \tag{2.4}$$

The proof is similar to the proofs in [2, section 2.7.4] and we omit it here. Let us note that the corresponding result without perturbation in $\widetilde{x}$ looks as follows:

$$\frac{\|\widetilde{x} - x\|_2}{\|x\|_2} \le \frac{2\kappa_2(H)\delta}{1 - \kappa_2(H)\delta}. \tag{2.5}$$

Thus allowing perturbation in $\widetilde{x}$ leads to a negligible change in the bound (2.5), which can be obtained for a strongly stable algorithm. Therefore, we have introduced the notion of almost strong stability that will lead to bounds of the type (2.4). In this sense strong and almost strong stability are close.

3. Stability analysis of the circular deconvolution algorithm. Now let us obtain a mixed stability analysis for the algorithm presented in the introduction. This means that we allow perturbations in the inputs and outputs. The inputs are taken with their structure, i.e., the vectors c and b are inputs.

The stability of the FFT depends on the stability of computing the so-called weights (or twiddle factors) (for more details see [10, section 1.3]). We assume that a twiddle factor w is computed as

$$\hat{w} = w + \tau, \quad |\tau| \le c_n \rho_0, \tag{3.1}$$

where τ is the absolute error from the computation. The constant c_n depends on the algorithm for the computation of the twiddle factors. Among the different methods we can choose for the compuation of the weights are those for which we can take $c_n = c, c_n = c \log_2 n, c_n = cn$, where c is a small constant not depending on n (see [10, section 1.4]).

We will need the following result which is proved in [12].

THEOREM 3.1. *For the FFT computed with roundoff errors we have*

$$\widetilde{y} = F_n\left(x + \Delta x\right),$$

where

$$\left\|\Delta x\right\|_2 / \left\|x\right\|_2 \leq g_n \rho_0 + O\left(\rho_0^2\right), \quad g_n = \sqrt{2}(2 + \sqrt{2} + c_n)\log_2 n$$

and the tilde denotes computed results.

Let us note that the same bound for the inverse FFT can be proved similarly, and we will use it here. Now we have

$$\widetilde{d} = F\left(c + \delta c\right), \quad \left\|\delta c\right\|_2 \leq g_n \left\|c\right\|_2 \rho_0 + O(\rho_0^2), \tag{3.2}$$

$$\widetilde{e} = F\left(b + \Delta b\right), \quad \left\|\Delta b\right\|_2 \leq g_n \left\|b\right\|_2 \rho_0 + O(\rho_0^2), \tag{3.3}$$

$$\widetilde{f}_i = \frac{\widetilde{e}_i}{\widetilde{d}_i}\left(1 + \sigma_i\right), \quad \left|\sigma_i\right| \leq \rho_0, \quad i = 0, \ldots, n-1, \tag{3.4}$$

$$\widetilde{x} = F^{-1}\left(\widetilde{f} + \Delta f\right), \quad \left\|\Delta f\right\|_2 \leq g_n \left\|f\right\|_2 \rho_0 + O(\rho_0^2). \tag{3.5}$$

Let us note that (3.4) can be presented in the following way:

$$\widetilde{f} = \widetilde{\Lambda}^{-1}\left(I + D\right)\widetilde{e}, \quad \widetilde{\Lambda} = \operatorname{diag}\left(\widetilde{d}\right), \quad D = \operatorname{diag}\left(\sigma\right). \tag{3.6}$$

Here and in the following, $\operatorname{diag}(s)$ denotes a diagonal matrix whose principal diagonal is stored in a vector s. Also, (3.5) can be transformed to

$$\widetilde{x} + \Delta x = F^{-1}\widetilde{f}, \quad \left\|\Delta x\right\|_2 \leq g_n \left\|x\right\|_2 \rho_0 + O(\rho_0^2) \tag{3.7}$$

because F^{-1} is orthogonal. Combining (3.2)–(3.3), (3.6), and (3.7) we get

$$\begin{aligned}\widetilde{x} + \Delta x &= F^{-1}\widetilde{f} = F^{-1}\widetilde{\Lambda}^{-1}\left(I + D\right)\widetilde{e} \\ &= F^{-1}\widetilde{\Lambda}^{-1}\left(I + D\right)F\left(b + \Delta b\right),\end{aligned}$$

from which we obtain

$$F^{-1}\left(I + D_1\right)\widetilde{\Lambda}F\left(\widetilde{x} + \Delta x\right) = b + \Delta b, \tag{3.8}$$

where

$$\left(I + D_1\right) = \left(I + D\right)^{-1} \quad \text{and} \quad \left|\left(D_1\right)_{ii}\right| \leq \rho_0 + O(\rho_0^2). \tag{3.9}$$

We also have

$$\left(I + D_1\right)\widetilde{\Lambda} = \operatorname{diag}\left(\left(I + D_1\right)\widetilde{d}\,\right) = \operatorname{diag}\left(\left(I + D_1\right)F\left(c + \delta c\right)\right). \tag{3.10}$$

Let us represent the vector inside (3.10) as follows:

$$(I + D_1) F (c + \delta c) = F (c + \Delta c) . \tag{3.11}$$

Then from (3.11) we obtain that

$$\Delta c = \delta c + F^{-1} D_1 F (c + \delta c) ,$$

from which it follows that

$$\|\Delta c\|_2 \leq g'_n \|c\|_2 \rho_0 + O(\rho_0^2), \quad g'_n = (1 + \rho_0) g_n + 1, \tag{3.12}$$

by using the orthogonality of F, (3.2), and (3.9).

Finally, from (3.8), (3.10), and (3.11) with (3.12) we obtain

$$F^{-1} (I + D_1) \widetilde{\Lambda} F = F^{-1} \mathrm{diag} (F (c + \Delta c)) F = H_{c+\Delta c},$$

where the perturbation in matrix H_c is now structured. Thus, we have proved the following theorem.

THEOREM 3.2. *The computed solution $\widetilde{x}$ of the circular deconvolution problem satisfies*

$$H_{c+\Delta c} (\widetilde{x} + \Delta x) = b + \Delta b,$$

where

$$\|\Delta c\|_2 / \|c\|_2 \leq g'_n \rho_0 + O(\rho_0^2), \quad \|\Delta b\|_2 / \|b\|_2 \leq g_n \rho_0 + O(\rho_0^2),$$

$$\|\Delta x\|_2 / \|x\|_2 \leq g_n \rho_0 + O(\rho_0^2),$$

and $g'_n = (1 + \rho_0) g_n + 1, g_n = \sqrt{2}(2 + \sqrt{2} + c_n) \log_2 n$.

This result shows that the algorithm is almost strongly stable according to our definition in the previous section. Then the forward error in x can be bounded by Theorem 1:

$$\frac{\|\widetilde{x} - x\|_2}{\|x\|_2} \leq \left[\frac{2\kappa_2 (H)}{1 - \kappa_2 (H) g'_n \rho_0} + 1 \right] g'_n \rho_0.$$

This bound is not essentially different from the corresponding bound for the case when the algorithm would be strongly stable.

At the end of this section we come back to example (2.2). We compute the quantity ω from (2.3) for different choices of μ in (2.2). Now it is possible, i.e., ω exists. The number μ is chosen in the range $10^{-30} \div 1$, i.e., we choose very small numbers less than the machine precision and relatively large numbers close to 1. For small values of μ, matrix H_c is almost singular. For all values of μ the quantity ω is approximately equal to 10^{-15} (in MATLAB, where $\rho_0 \approx 2.22 \times 10^{-16}$). Thus, this example illustrates our theoretical result that the algorithm under consideration is almost strongly stable.

Acknowledgment. The author thanks the referee for the useful suggestions.

REFERENCES

[1] J. R. Bunch, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88 (1987), pp. 49–66.
[2] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The John Hopkins University Press, Baltimore, MD, 1996.
[3] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 162–175.
[4] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[5] A. K. Jain, *Fast inversion of banded Toeplitz matrices by circular decomposition*, IEEE Trans. Acoust. Speech Signal Process., 26 (1978), pp. 121–126.
[6] E. Linzer, *On the stability of transform-based circular deconvolution*, SIAM J. Numer. Anal., 29 (1992), pp. 1482–1492.
[7] W. Menke, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, New York, 1984.
[8] M. T. Silva and E. A. Robinson, *Deconvolution of Geophysical Time Series in the Exploration for Oil and Natural Gas*, Elsevier, New York, 1979.
[9] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.
[10] C. F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, PA, 1992.
[11] V. V. Voevodin and E. E. Tyrtyshnikov, *Computational Processes with Toeplitz Matrices*, Nauka, Moscow, 1987 (in Russian).
[12] P. Y. Yalamov, *Normwise and componentwise stability of the fast Fourier transform*, Preprint N41, University of Rousse, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 364–375

PERFORMANCE OF THE QZ ALGORITHM IN THE PRESENCE OF INFINITE EIGENVALUES*

DAVID S. WATKINS†

Abstract. The implicitly shifted (bulge-chasing) QZ algorithm is the most popular method for solving the generalized eigenvalue problem $Av = \lambda Bv$. This paper explains why the QZ algorithm functions well even in the presence of infinite eigenvalues. The key to rapid convergence of QZ (and QR) algorithms is the effective transmission of shifts during the bulge chase. In this paper the mechanism of transmission of shifts is identified, and it is shown that this mechanism is not disrupted by the presence of infinite eigenvalues. Both the QZ algorithm and the preliminary reduction to Hessenberg-triangular form tend to push the infinite eigenvalues toward the top of the pencil. Thus they should be deflated at the top.

Key words. matrix pencil, eigenvalue computation, QZ algorithm

AMS subject classifications. 65F15, 15A18

PII. S0895479899360376

1. Introduction.

The standard matrix eigenvalue problem for the $n \times n$ matrix A has the form

$$(A - \lambda I)v = 0.$$

Many matrix eigenvalue problems are more naturally presented not in this form but as *generalized eigenvalue problems*

$$(A - \lambda B)v = 0.$$

Generalized eigenvalue problems have some interesting special features; for example, they can have infinite eigenvalues.

The most popular algorithm for solving generalized eigenvalue problems is the QZ algorithm of Moler and Stewart [2]. This is a generalization of the QR algorithm, which solves standard eigenvalue problems. An important feature of the QZ algorithm that was rightly emphasized by its inventors is that it functions perfectly well in the presence of infinite eigenvalues. However, when one looks at explanations of the QZ algorithm, e.g., [1], [2], [7], they always assume from the outset that B is nonsingular, which rules out infinite eigenvalues. As far as this author knows, all explanations of the QZ algorithm that have been published so far have shared this weakness. Thus there are good explanations of the QZ algorithm in the literature, but none of them holds for the case when infinite eigenvalues are present.

This paper discusses the processing of infinite eigenvalues by the QZ algorithm and also by the algorithm that carries out the preliminary reduction to Hessenberg-triangular form. The latter (if implemented in the usual way) tends to push the infinite eigenvalues to the top of the pencil. The QZ iterations also push the infinite eigenvalues steadily upward, so that they can be deflated from the top of the pencil after finitely many iterations.

*Received by the editors August 26, 1999; accepted for publication (in revised form) by J. Varah March 6, 2000; published electronically July 11, 2000.
http://www.siam.org/journals/simax/22-2/36037.html
†Department of Pure and Applied Mathematics, Washington State University, Pullman, WA 99164-3113 (watkins@wsu.edu).

Each QZ iteration makes use of shifts to introduce a bulge in the Hessenberg form at the top of the pencil. Then the bulge is chased to the bottom and off of the edge of the pencil, restoring the Hessenberg-triangular form. In the course of the iterations, eigenvalues are deflated one or more at a time. While infinite eigenvalues emerge at the top of the pencil, the finite eigenvalues are normally deflated at the bottom. The key to rapid convergence of the finite eigenvalues is the effective transmission of the shifts from top to bottom of the pencil during the bulge chase. (Of course the shifts must also be good approximations to eigenvalues, but this is not hard to arrange.) In this paper we identify the mechanism by which shifts are transmitted through the pencil during the bulge chase, and we demonstrate that the shift-transmission mechanism is not disrupted by the presence of infinite eigenvalues. The demonstration consists of two parts. First we present a theorem that shows that (ignoring roundoff errors) the shifts are transmitted effectively. Then we present numerical evidence that the results hold up in the presence of roundoff errors. Thus finite eigenvalues converge rapidly at the bottom of the pencil, regardless of whether or not infinite eigenvalues are present. In either case, every few iterations produces one or more new finite eigenvalues for deflation.

Deflation of infinite eigenvalues must not be neglected. If it is, the shift-transmission process breaks down, and progress toward convergence comes to a halt.

The implicitly shifted QZ algorithms that we study in this paper are members of the larger family of implicitly shifted GZ algorithms [7]. The ideas presented here are applicable to the larger family. We will restrict our attention to the QZ case in order to keep the presentation as simple as possible.

In the interest of brevity we have refrained from describing the algorithms in detail. More details and motivation are given in an earlier version of this paper [5], which is available electronically.

2. Basic facts and terminology. Given a pair of real or complex $n \times n$ matrices A and B, the matrix polynomial $A - \lambda B$ with indeterminate λ is called a *matrix pencil*. A finite complex number λ is called an *eigenvalue* of the pencil $A - \lambda B$ if there is a nonzero vector v (called an *eigenvector*) such that $(A - \lambda B)v = 0$. The problem of finding the eigenvalues of a matrix pencil is called the *generalized eigenvalue problem.* One easily sees that if the matrix B is nonsingular, the eigenvalues of the matrix pencil $A - \lambda B$ are exactly the eigenvalues of the matrix $B^{-1}A$. There are n of them, and they are (finite) complex numbers.

Regardless of whether or not B is singular, the (finite) eigenvalues of the matrix pencil are exactly the solutions of the *characteristic equation*

$$\det(A - \lambda B) = 0.$$

This is analogous to the standard eigenvalue problem. The difference is that if B is singular, the *characteristic polynomial* $\det(A - \lambda B)$ has degree less than n. In fact, it can even happen that $\det(A - \lambda B)$ is identically zero. For example, this happens when A and B have a common null vector. Then every λ is an eigenvalue. If $\det(A - \lambda B)$ is identically zero, we call $A - \lambda B$ a *singular* pencil. Otherwise it is a *regular* pencil. We will focus on regular pencils.

Two matrix pencils $A - \lambda B$ and $\tilde{A} - \lambda \tilde{B}$ are called *strictly unitarily equivalent* if there are unitary matrices U and V such that $\tilde{A} - \lambda \tilde{B} = U(A - \lambda B)V$. Obviously strictly unitarily equivalent pencils have the same eigenvalues. The generalized Schur theorem [1, Theorem 7.7.1] states that every pencil is strictly unitarily equivalent to a pencil $\tilde{A} - \lambda \tilde{B}$ for which $\tilde{A}$ and $\tilde{B}$ are upper triangular. Letting $\alpha_1, \ldots, \alpha_n$ and

$\beta_1, \ldots, \beta_n$ denote the main diagonal entries of $\tilde{A}$ and $\tilde{B}$, respectively, we see that the characteristic equation of $\tilde{A} - \lambda\tilde{B}$ is

$$\prod_{i=1}^{n}(\alpha_i - \lambda\beta_i) = 0.$$

If $\alpha_i = \beta_i = 0$ for some i, the pencil is singular. Otherwise it is regular, and each pair (α_i, β_i) for which $\beta_i \neq 0$ gives rise to an eigenvalue α_i/β_i. If the pencil is regular but the matrix B (and $\tilde{B}$) is singular, there will be at least one pair for which $\beta_i = 0$ (and $\alpha_i \neq 0$). It is reasonable to say that each of these gives rise to an *infinite eigenvalue*. (Each corresponds to a zero eigenvalue of the reciprocal pencil $\mu A - B$.) If we make this convention, then each regular pencil has exactly n eigenvalues, counting the infinite ones.

Since the generalized Schur form tells everything about the eigenvalues of a pencil, one would naturally like to have an algorithm that transforms a pencil to generalized Schur form by a sequence of unitary equivalence transformations. A big step in this direction is to transform the pencil to *Hessenberg-triangular form*. Every pencil is strictly unitarily equivalent to a pencil $\hat{A} - \lambda\hat{B}$ for which $\hat{A}$ is upper Hessenberg ($\hat{a}_{ij} = 0$ if $i > j + 1$) and $\hat{B}$ is upper triangular. The reduction can be carried out by a direct procedure in $O(n^3)$ flops.

3. Movement of zeros during the reduction to Hessenberg-triangular form. The standard algorithm for transforming a pencil to Hessenberg-triangular form [1, section 7.7.4] begins by using either a QR or an RQ decomposition to transform B to upper triangular form. Whatever tranformations we apply to B, we must also apply to A. Once B is upper triangular, if there are infinite eigenvalues, there must be zeros on the main diagonal of B. Let us study the fate of these zeros as the algorithm proceeds.

The rest of the algorithm consists of a sequence of pairs of Givens rotators, the first of which annihilates an entry of A and creates a nonzero entry in the lower triangle of B. The second rotator then restores B to upper triangular form. It is a general principle of the algorithm that whenever the triangular form of B is disturbed, we restore it immediately. For example, the first transformation is a Givens rotator (or Householder reflector or other unitary transformation) that is applied on the left, acts on rows n and $n-1$, and annihilates a_{n1}. This same transformation must also be applied to B, and when it is, it recombines rows n and $n-1$, creating a new nonzero entry in position $(n, n-1)$. The next rotator is then applied on the right to columns n and $n-1$ and returns $b_{n,n-1}$ to zero. When the same rotator is applied to columns n and $n-1$ of A, the zero in position $(n, 1)$ is not disturbed. The next pair of rotators acts on rows and columns $n-1$ and $n-2$ to transform $a_{n-1,1}$ to zero. The first rotator in the pair creates a new nonzero in position $b_{n-1,n-2}$, which is immediately eliminated by the second rotator. Continuing up the first column, we create zeros in A up through position $(3, 1)$. Thus an *upward wave* of rotators has cleared out the first column. Then a second upward wave is used to clear out the second column in the same way, up through position $(4, 2)$, and so on. After $n-2$ upward waves, consisting of $(n-1)(n-2)/2$ pairs of rotators in all, the pencil has reached Hessenberg-triangular form.

Now suppose B has zeros on its main diagonal. How are the zeros affected by these transformations? Suppose $b_{kk} = 0$, $k > 2$. Then it will remain zero until one of the rotators touches the kth row or column. The first one to do so acts on rows

k and $k+1$. This one leaves b_{kk} at zero, since it recombines the zeros in positions b_{kk} and $b_{k+1,k}$ to create new zeros in those positions. The next transformation is a rotator in columns k and $k+1$ that creates a zero in position $b_{k+1,k}$. Since that entry was already zero to begin with, the rotator is trivial (i.e., it has angle zero) and leaves $b_{kk} = 0$. The next rotator acts on rows $k-1$ and k, and this one normally does make b_{kk} nonzero. Let us focus on the 2×2 submatrix of B consisting of rows and columns $k-1$ and k. Before the rotator, it has the form

$$\begin{bmatrix} b_{k-1,k-1} & b_{k-1,k} \\ 0 & 0 \end{bmatrix}. \tag{3.1}$$

Its rank is obviously one. When the rotator is applied, it disturbs both of the zeros. The submatrix now looks like

$$\begin{bmatrix} \tilde{b}_{k-1,k-1} & \tilde{b}_{k-1,k} \\ \tilde{b}_{k,k-1} & \tilde{b}_{k,k} \end{bmatrix}, \tag{3.2}$$

but its rank is still one. The next step in the QZ iteration is to apply a rotator to columns $k-1$ and k to annihilate $\tilde{b}_{k,k-1}$. Application of this rotator transforms the submatrix to

$$\begin{bmatrix} 0 & \hat{b}_{k-1,k} \\ 0 & \hat{b}_{k,k} \end{bmatrix} \tag{3.3}$$

since the rank is still one. The zero has been moved from position b_{kk} to position $b_{k-1,k-1}$.

The next pair of rotators acts on columns $k-2$ and $k-1$ and pushes the zero up to position $(k-2, k-2)$ by the same process. The zero thus normally continues upward until it either arrives at position $(2,2)$ or bumps into another zero. The only way the upward drift can be stopped is if a trivial rotator is applied on the left at some point. This happens when and only when the entry of A that is to be annihilated is already zero. In this event the zero in B stops and remains where it is until the next upward wave of rotators (corresponding to elimination of the next column of A) passes through.

Collision of two zeros. Suppose B has two or more zeros on the main diagonal. Note that the number of zeros on the main diagonal of B is not necessarily equal to the number of infinite eigenvalues of the pencil. For example, bulge pencils $C_j - \lambda F_j$, which we will discuss below (see (5.1), (5.2)), have only one infinite eigenvalue, even though all of the entries on the main diagonal of F_j are zero. During the reduction to Hessenberg-triangular form, the number of zeros on the main diagonal of B need not remain constant. Consider what happens when a zero that is moving upward on the main diagonal of B runs into another zero. Then we have the configuration

$$\begin{bmatrix} 0 & b_{k-1,k} \\ 0 & 0 \end{bmatrix}. \tag{3.4}$$

The left transformation of rows $k-1$ and k then gives

$$\begin{bmatrix} 0 & \tilde{b}_{k-1,k} \\ 0 & \tilde{b}_{kk} \end{bmatrix}, \tag{3.5}$$

in which $\tilde{b}_{kk}$ is normally nonzero. Once it becomes nonzero, it stays nonzero. Thus the lower zero is destroyed. This can be prevented by two things: (i) $b_{k-1,k}$ could also

be zero, and (ii) the left rotator on rows $k-1$ and k could be trivial. The conclusion is that when two zeros collide, the upper one survives, but the lower one may (or may not) be destroyed.

The number of zeros on the main diagonal of B can decrease, but it cannot increase. If the matrix

$$\begin{bmatrix} b_{k-1,k-1} & b_{k-1,k} \\ 0 & b_{kk} \end{bmatrix}$$

has rank two before the tranformations on rows and columns $k-1$ and k, then it will still have rank two afterwards. Thus zeros cannot spontaneously appear. The same is true during the QZ iterations, as we shall see. It follows that the number of zeros on the main diagonal of B can never be less than the number of infinite eigenvalues, since the generalized Schur form at which we arrive in the end must have one zero on the main diagonal of B for each infinite eigenvalue.

An interesting conclusion. Now consider the overall reduction algorithm. We restrict our attention to the generic case, in which all of the left rotators are nontrivial. Once B is in triangular form, the entry b_{11} is never touched again by the reduction algorithm. If it is zero, it stays zero. If there are other zeros on the main diagonal, they will be moved upward during the first upward wave of rotations (corresponding to elimination of the first column of A). Some of them may be destroyed, but the uppermost one will end up in position $(2,2)$. If there are any other zeros left over, then they will be moved upward during the next upward wave of rotators (corresponding to elimination of the second column of A). The uppermost will arrive in position $(3,3)$ since the last rotator in this wave acts on columns 3 and 4. If there are any zeros left over after this, the upper one will be moved into position $(4,4)$ by the next wave of rotators, and so on.

Once the pencil is in Hessenberg-triangular form, one has the option of deflating the infinite eigenvalues using an algorithm such as the one described in [1, section 7.7.5] (and implemented in standard, public domain software). However, that algorithm is inefficient when used in conjunction with the standard reduction algorithm [1, section 7.7.4] because the latter moves the zeros to the top of B, then the former chases them downward for deflation of infinite eigenvalues at the bottom. It would be much more efficient to use the upward-chasing analogue of [1, section 7.7.5], which deflates infinite eigenvalues at the top. Better yet, one can let the QZ algorithm deal with the zeros, as we shall explain.

4. Movement of zeros during QZ iterations. We now consider iterations of the QZ algorithm on a pencil that is in Hessenberg-triangular form. If any one of the subdiagonal entries $a_{j+1,j}$ is zero, we can split the eigenvalue problem into two (or more) subproblems involving subpencils. Therefore, we can assume without loss of generality that our A is a *proper* upper Hessenberg matrix, i.e., $a_{j+1,j} \neq 0$ for all j.

If there are zeros on the main diagonal of B, we normally expect them to be near the top, due to the action of the reduction algorithm. However, it could be that the pencil was already in or near Hessenberg-triangular form to begin with and the reduction algorithm was either not applied or only partially applied. Then the zeros could lie anywhere on the main diagonal of B. Thus we will make no special assumption about where the zeros lie.

A QZ iteration begins with an equivalence transformation that disturbs the Hessenberg form of A by introducing a bulge near the upper left-hand corner. The rest

of the iteration consists of returning A to Hessenberg form by chasing the bulge from one end of the matrix to the other and, finally, off the edge. The bulge-chasing part of the algorithm is in fact an instance of the reduction to Hessenberg-triangular form that we have just discussed. However, it is a highly nongeneric instance. Due to the large number of zeros in the array, the vast majority of the rotators are trivial.

A QZ iteration of degree 1 consists of a sequence of $n-1$ pairs of rotators, acting on rows and columns 1–2, 2–3, $\ldots$, $(n-1)$–n, in that order. (The 1–2 pair of rotators creates the bulge.) All of the rotators are nontrivial.

With no more information than this, we can see what happens to the zeros on the main diagonal of B during a QZ iteration. If $b_{kk} = 0$, then it stays zero until the rotations that act on rows and columns $k-1$ and k are performed. This pair of rotators then moves the zero to position $(k-1, k-1)$, as shown in (3.1), (3.2), and (3.3). The subsequent pairs of rotators do not act on that entry, so we have $b_{k-1,k-1} = 0$ at the end of the iteration. In conclusion, each zero gets moved upward by one position during a QZ iteration of degree 1. Once a zero reaches the top of B, an infinite eigenvalue can be deflated from the top of the pencil. Zeros are neither created nor destroyed during the QZ iteration, but they can be destroyed during the deflation process.

The finite emergence of infinite eigenvalues at the top of the pencil is consistent with the convergence theory of the QZ algorithm [7]. The largest (in magnitude) eigenvalues should emerge at the top. Nothing is bigger than infinity. The analysis of [7] applies only to finite eigenvalues, but if one considers what happens in the limit as several of the eigenvalues are pushed out toward infinity, one expects the infinite eigenvalues to converge superlinearly. Finite emergence is certainly superlinear.

A QZ iteration of degree m is equivalent to m QZ iterations of degree 1. Thus each QZ iteration of degree m moves each zero upward by m positions.

5. Transmission of shifts in the QZ algorithm. We have seen that in the course of QZ iterations the infinite eigenvalues move steadily to the top of the pencil and can be deflated when they arrive there. Thus it appears that there is no need to deflate them beforehand. However, before we can be certain that this is so, we must demonstrate that the presence of zeros on the main diagonal of B does not interfere with the normal functioning of the QZ iterations. In order to do this, we must identify the mechanism by which the QZ algorithm functions. Here we offer a brief sketch; more details are given in [5].

Typical implementations of the QZ algorithm have degree $m = 1$ or $m = 2$. There is no theoretical reason why m cannot be higher, say 4 or 6. However, shift blurring [3] due to roundoff errors limits the effectiveness of the QZ algorithm when m is large (e.g., $m = 20$). Thus we shall think of m as a small number.

A QZ iteration of degree m begins with the choice of m shifts $\mu_1, \ldots, \mu_m$. A common strategy is to take them to be the eigenvalues of the lower right-hand $m \times m$ subpencil. That is, if

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

where $A_{22} - \lambda B_{22}$ is $m \times m$, then we take the shifts to be the eigenvalues of the small subpencil $A_{22} - \lambda B_{22}$. The hope is that these shifts will be good approximations to eigenvalues of the pencil. Notice that A_{21} has only one nonzero entry, $a_{n-m+1,n-m}$. If this entry is small, all of the shifts will be excellent approximations to eigenvalues of the pencil, except in ill-conditioned situations.

Once the shifts have been chosen, they are used in the computation of a unitary transformation that when applied on the left (with accompanying right transformations to maintain the triangular form of B) results in the formation of a bulge in A that protrudes m diagonals beyond the subdiagonal. For example, in the case $m = 2$, the transformed pencil looks like

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ & & & a_{54} & a_{55} & a_{56} \\ & & & & a_{65} & a_{66} \end{bmatrix} - \lambda \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ & & b_{33} & b_{34} & b_{35} & b_{36} \\ & & & b_{44} & b_{45} & b_{46} \\ & & & & b_{55} & b_{56} \\ & & & & & b_{66} \end{bmatrix}.$$

The rest of the iteration consists of returning the pencil to Hessenberg-triangular form by chasing the bulge from top to bottom. This is a rather long trip if the matrices are, say, 1000×1000.

The objective of the QZ iteration is to drive $a_{n-m+1,n-m}$ to zero. First, suppose B is nonsingular. Then, if the shifts are good approximations to eigenvalues, then $|a_{n-m+1,n-m}|$ will be decreased substantially from one iteration to the next. Asymptotically $a_{n-m+1,n-m} \to 0$ quadratically in most cases. This claim is based on a connection between the QZ algorithm and the power method that can be made when B is nonsingular [7]. Once $|a_{n-m+1,n-m}|$ is small enough, it can be set to zero, and m (finite) eigenvalues can be deflated from the bottom.

We would like to determine whether this good performance is maintained when B is singular. To answer this question we consider the role of the shifts. These are taken from the bottom of the pencil and used to determine a transformation that creates a bulge at the top. Once we have a bulge, we forget about the shifts and mechanically chase the bulge to the bottom of the pencil. We hope for a deflation at or around $a_{n-m+1,n-m}$, i.e., near the bottom of the pencil. Good shifts are crucial to rapid convergence. However, the information about the shifts is loaded into the top of the pencil, and convergence takes place at the bottom. Somehow the information about the shifts is transferred from the top to the bottom during the bulge chase. Our task is to identify the mechanism and determine whether or not it is disrupted by the presence of zeros on the main diagonal of B.

For the standard eigenvalue problem the shift-transmission mechanism was identified in [3], [4]. For the generalized problem, the mechanism turns out to be about the same. Consider a pencil that has a bulge somewhere in the middle. Say the initial bulge has been created, and it has been chased $j-1$ positions down and to the right. The current pencil $A_j - \lambda B_j$ has a bulge (in A_j) starting in column j. If the degree of the QZ iteration is m, the tip of the bulge is at $a_{j+m+1,j}$. We define the *bulge pencil* $C_j - \lambda F_j$ to be the $(m+1) \times (m+1)$, nonprincipal subpencil of $A_j - \lambda B_j$ consisting of rows $j+1$ through $j+m+1$ and columns j through $j+m$. Thus

$$C_j = \begin{bmatrix} a_{j+1,j} & a_{j+1,j+1} & a_{j+1,j+2} & \cdots & a_{j+1,j+m} \\ a_{j+2,j} & a_{j+2,j+1} & a_{j+2,j+2} & \cdots & a_{j+2,j+m} \\ \vdots & \vdots & & & \vdots \\ a_{j+m,j} & a_{j+m,j+1} & a_{j+m,j+2} & \cdots & a_{j+m,j+m} \\ a_{j+m+1,j} & a_{j+m+1,j+1} & a_{j+m+1,j+2} & \cdots & a_{j+m+1,j+m} \end{bmatrix} \tag{5.1}$$

and

$$F_j = \begin{bmatrix} 0 & b_{j+1,j+1} & b_{j+1,j+2} & \cdots & b_{j+1,j+m} \\ 0 & 0 & b_{j+2,j+2} & \cdots & b_{j+2,j+m} \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & b_{j+m,j+m} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{5.2}$$

The bulge pencil is centered on the subdiagonal of the big pencil, and it is just big enough to accommodate the bulge. One can show by induction on j that the entry $a_{j+m+1,j}$ cannot be zero. (The original A is properly upper Hessenberg.) F_j is strictly upper triangular. If all of the superdiagonal entries of F_j are nonzero, then the degree of the characteristic polynomial $\det(C_j - \lambda F_j)$ is exactly m. In this case the bulge pencil has m finite eigenvalues and one infinite eigenvalue.

The main theorem is that the m finite eigenvalues of the bulge pencil are the shifts $\mu_1, \ldots, \mu_m$. Thus the shifts are transmitted from top to bottom of the matrix as eigenvalues of the bulge pencil. In order to prove this, we need to introduce a "zeroth" bulge pencil.

The zeroth bulge. Given m shifts, we calculate a vector

$$x = \alpha(A - \mu_1 B)B^{-1} \cdots (A - \mu_m B)B^{-1} e_1, \tag{5.3}$$

where α is any convenient nonzero scale factor. Because A is upper Hessenberg and B is upper triangular, only the first $m+1$ components of x are nonzero. Although the symbol B^{-1} appears in (5.3), it is possible to compute x even if B is singular. Only the upper $m \times m$ block of B^{-1} is used in the computation, so x is well defined so long as $b_{kk} \neq 0$ for $k = 1, \ldots, m$.

Define the *zeroth bulge pencil* $C_0 - \lambda F_0$ by

$$C_0 = \begin{bmatrix} x_1 & a_{1,1} & \cdots & a_{1,m-1} & a_{1,m} \\ x_2 & a_{2,1} & \cdots & a_{2,m-1} & a_{2,m} \\ \vdots & \vdots & & \vdots & \vdots \\ x_m & 0 & \cdots & a_{m,m-1} & a_{m,m} \\ x_{m+1} & 0 & \cdots & 0 & a_{m+1,m} \end{bmatrix}$$

and

$$F_0 = \begin{bmatrix} 0 & b_{1,1} & \cdots & b_{1,m-1} & b_{1,m} \\ 0 & 0 & \cdots & b_{2,m-1} & b_{2,m} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & b_{m,m} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

The entries $x_1, x_2, \ldots, x_{m+1}$ are the nonzero entries of the vector x defined by (5.3), and the entries a_{ij} and b_{ij} are from the pencil $A - \lambda B$ before the beginning of the iteration. The "bulge" in this pencil is caused by x. We can view $C_0 - \lambda F_0$ as a subpencil of the augmented pencil obtained by adjoining a "zeroth" column $x - \lambda 0$ to the pencil $A - \lambda B$. If we take this view, then $C_0 - \lambda F_0$ is not so different from the other bulge pencils $C_j - \lambda F_j$.

Because A is Hessenberg and B is triangular, the computation of x (cf. (5.3)) uses only the upper left-hand corner entries of A and B. One easily checks that the entries that participate in the computation are exactly those that are contained in $C_0 - \lambda F_0$. This is the only part of the computation that uses the shifts. We therefore expect that it should be possible to recover the shifts from $C_0 - \lambda F_0$.

THEOREM 5.1. *Suppose $b_{kk} \neq 0$ for $k = 1, \ldots, m$. Then the eigenvalues of the zeroth bulge pencil $C_0 - \lambda F_0$ are ∞ and the shifts $\mu_1, \ldots, \mu_m$.*

Proof. Since A is properly upper Hessenberg, we deduce easily that $x_{m+1} \neq 0$. This condition and the conditions $b_{kk} \neq 0$, $k = 1, \ldots, m$, together imply that the characteristic polynomial $\det(C_0 - \lambda F_0)$ has degree exactly m. Thus $C_0 - \lambda F_0$ has one infinite eigenvalue and m finite eigenvalues. To see that each shift μ_i is an eigenvalue of $C_0 - \lambda F_0$, write $p(AB^{-1}) = (A - \mu_1 B)B^{-1} \cdots (A - \mu_m B)B^{-1}$ in partially factored form: $p(AB^{-1}) = (A - \mu_i B)B^{-1}q(AB^{-1})$, where q has degree $m-1$. Then $x = (A - \mu_i B)y$, where $y = B^{-1}q(AB^{-1})e_1$. Let $\hat{y}$ be the subvector of y consisting of the first m entries, and note that the rest of y is zero. Then the equation $x = (A - \mu_i B)y$ can be recast as

$$(C_0 - \mu_i F_0) \begin{bmatrix} 1 \\ -\hat{y} \end{bmatrix} = 0.$$

Thus μ_i is an eigenvalue of $C_0 - \lambda F_0$.

This argument holds even if B^{-1} does not exist; all that is needed is that the upper left-hand corner of B is invertible.

If $\mu_1, \ldots, \mu_m$ are distinct, then there can be no other finite eigenvalues. If $\mu_1, \ldots, \mu_m$ are not distinct, we draw the same conclusion by a continuity argument: Perturb the shifts slightly so that they are distinct. This implies a small perturbation of x. The m perturbed shifts are the m finite eigenvalues of the slightly perturbed bulge pencil. Now move the shifts continuously back to their original values and invoke continuity of eigenvalues of a pencil. ∎

Now we can present our main result.

THEOREM 5.2. *Suppose $b_{kk} \neq 0$ for $k = 1, \ldots, m$. Then all of the bulge pencils $C_j - \lambda F_j$, $j = 1, 2, \ldots$ have ∞ and $\mu_1, \ldots, \mu_m$ as their eigenvalues.*

Proof. The proof is by induction. We just need to show that $C_{j+1} - \lambda F_{j+1}$ has the same eigenvalues as $C_j - \lambda F_j$. Suppose we have pushed the bulge forward to the point where we have reached the pencil $A_j - \lambda B_j$. The bulge begins in column j. In preparation for pushing the bulge further, consider the $(m+2) \times (m+2)$ subpencil of $A_j - \lambda B_j$ that consists of $C_j - \lambda F_j$ plus one additional column on the right and one additional row on the bottom. This augmented bulge pencil, which we will call $\hat{C}_j - \lambda \hat{F}_j$ has the same eigenvalues as $C_j - \lambda F_j$, except for one additional infinite eigenvalue. The transformation that moves the bulge one row down and one column to the right transforms $\hat{C}_j - \lambda \hat{F}_j$ to a new pencil $\check{C}_j - \lambda \check{F}_j$, which has the same eigenvalues, because the transformation is a strict equivalence. If we now delete the first row and column from $\check{C}_j - \lambda \check{F}_j$, we obtain the new bulge pencil $C_{j+1} - \lambda F_{j+1}$. The effect of the deletion is just to remove an infinite eigenvalue. Thus $C_{j+1} - \lambda F_{j+1}$ has exactly the same eigenvalues as $C_j - \lambda F_j$.

This argument is applicable even in the case $j = 0$. The transformation that is used to set up the initial bulge is exactly the transformation one would use to chase the "bulge" x from $C_0 - \lambda F_0$. Thus $C_1 - \lambda F_1$ is produced from $C_0 - \lambda F_0$ in exactly the same way as each subsequent bulge pencil is produced from its predecessor. This completes the proof. ∎

The hypothesis $b_{kk} \neq 0$ for $k = 1, \ldots, m$ is crucial. Thus, before we start a QZ iteration, any zeros that are near the top of B need to be pushed to the top by the upward-chasing variant of [1, section 7.7.5] and deflated. The cost of this is negligible because m is small.

The important point is that Theorem 5.2 does not require that B be nonsingular; main diagonal zeros occurring below b_{mm} pose no problem. What happens to the bulge pencil when it meets up with such a zero? Suppose there is a zero immediately below the bulge pencil $C_j - \lambda F_j$. In the next step an upward wave of m pairs of rotators pushes the zero m positions upward while pushing the bulge one position downward. Now the zero lies immediately above the bulge pencil $C_{j+1} - \lambda F_{j+1}$. The zero has hopped over the bulge pencil.

Consider an iteration with shifts $\mu_1, \ldots, \mu_m$. Suppose we have pushed the bulge down to the point where it has passed through all of the zeros on the main diagonal of B. Say we have reached a point

$$\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} - \lambda \begin{bmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ 0 & \tilde{B}_{22} \end{bmatrix},$$

where the bulge is about to enter the subpencil $\tilde{A}_{22} - \lambda \tilde{B}_{22}$. By Theorem 5.2, the finite eigenvalues of the bulge pencil are $\mu_1, \ldots, \mu_m$, so the rest of the bulge chase is essentially a QZ iteration on $\tilde{A}_{22} - \lambda \tilde{B}_{22}$ with shifts $\mu_1, \ldots, \mu_m$. The shifts were determined by information in the lower right-hand corner of $A - \lambda B$, which is the same as the lower right-hand corner of $\tilde{A}_{22} - \lambda \tilde{B}_{22}$. If the shifts are good estimates of eigenvalues of $A - \lambda B$, they will also normally be good estimates of eigenvalues of $\tilde{A}_{22} - \lambda \tilde{B}_{22}$. Since $\tilde{B}_{22}$ is nonsingular, the standard convergence theory applies to the subpencil, and we expect good progress toward convergence, as measured by the reduction in $|a_{n-m+1,n-m}|$.

This heuristic argument proves nothing; it is only meant to be suggestive of success. If it is really true, then as the zeros gradually float upward through B, accurate shifts are transmitted in the bulge, through these zeros, to the bottom of the pencil, resulting in rapid convergence and deflation of finite eigenvalues at the bottom, just as if there were no zeros on the main diagonal of B.

6. Numerical tests. Since our argument is only suggestive of success, it is crucial to put it to some numerical tests. Another reason for caution is that Theorem 5.2 is true only in the absence of roundoff errors. In [3] it was shown that if m is large (e.g., $m = 20$), Theorem 5.2 often fails to hold in practice; roundoff errors prevent the effective transmission of shifts. This gives us yet another reason to perform some numerical experiments.

Here we are not concerned with large m. We want to ascertain whether the shift mechanism works well for small values of m in the presence of infinite eigenvalues. We conducted numerous experiments with pencils of various sizes and types and found that it does.

We will report on just a couple of examples. The experiments were performed using MATLAB and IEEE standard double precision arithmetic. First, consider a random, complex 20×20 Hessenberg-triangular pencil with no infinite eigenvalues, to which we apply the QZ algorithm with $m = 1$. Although we are discussing a single pencil, the results reported here are typical of many examples that we looked at. We find that at each point in the bulge chase, the single finite eigenvalue of the bulge pencil agrees with the intended shift to 15 or more decimal places. Thus the shift is

transmitted effectively. Checking the convergence pattern, we observe that the first four eigenvalues are deflated after 9, 12, 16, and 23 iterations, respectively. Each eigenvalue converges quadratically, as evidenced by the rate at which the bottom subdiagonal entry tends to zero.

Now suppose we alter the pencil by setting $b_{5,5}$ and $b_{15,15}$ to zero. Since the zeros are pushed up by one position per iteration, we expect to deflate infinite eigenvalues at the top after 4 and 13 iterations, and we do. At the same time we hope to have normal convergence behavior at the bottom of the pencil. Checking the single finite eigenvalue of the bulge pencil, we find that at each stage of the bulge chase it agrees with the shift to 15 or more decimal places. Thus the shift is transmitted effectively. We have no reason to believe that the altered pencil will have the same convergence pattern as the original pencil, but we hope that the trend will be comparable. Indeed it is; the first four eigenvalues are deflated after iterations 8, 11, 15, and 18, respectively. Quadratic convergence is observed. For example, Table 6.1 shows the values

TABLE 6.1
Quadratic convergence of an eigenvalue.

Iteration	$\lvert a_{20,19}\rvert$	Shift-transmission error
3	2.1×10^{-2}	1.2×10^{-15}
4	4.3×10^{-3}	1.1×10^{-15}
5	1.8×10^{-4}	3.3×10^{-16}
6	1.1×10^{-6}	1.0×10^{-15}
7	3.9×10^{-11}	4.4×10^{-16}
8	5.2×10^{-20}	7.0×10^{-16}

of $\lvert a_{20,19}\rvert$ in iterations 3 through 8. The approximate doubling of the exponent of $a_{20,19}$ from one iteration to the next indicates quadratic convergence to zero. Thus $a_{20,20}$ converges quadratically to a finite eigenvalue of the pencil. The presence of infinite eigenvalues does not in any way impede convergence. The *shift-transmission error* given in Table 6.1 is the difference between the intended shift and the finite eigenvalue of the bulge pencil when the bulge pencil has reached the bottom of the matrix. We see that these errors are a small multiple of the unit roundoff for IEEE double precision arithmetic.

It is interesting to see how the algorithm behaves if we neglect to deflate infinite eigenvalues as they emerge. Table 6.2 shows the same information as Table 6.1, except

TABLE 6.2
Breakdown caused by failure to deflate an infinite eigenvalue.

Iteration	$\lvert a_{20,19}\rvert$	Shift-transmission error
3	2.1×10^{-2}	1.2×10^{-15}
4	4.3×10^{-3}	1.1×10^{-15}
5	2.2×10^{-3}	1.7×10^{-1}
6	2.2×10^{-3}	7.1×10^{0}
7	2.2×10^{-3}	9.7×10^{0}

that in this run we do not deflate out the infinite eigenvalue that emerges after the fourth iteration. In principle the algorithm should crash because of a division by zero (b_{11}). In practice it does not, because roundoff errors prevent b_{11} from being exactly zero. Instead we have a breakdown of the shift-transmission process, as evidenced by

the large shift-transmission errors. As a consequence, the convergence process stalls.

Our second example is also a random Hessenberg-triangular pencil, but this one is 30×30 and has real entries. We apply the double-shift QZ algorithm ($m = 2$). The first four pairs of eigenvalues are deflated after 9, 12, 17, and 19 iterations, respectively. Quadratic convergence of $a_{n-1,n-2}$ to zero is observed.

Calculating the two finite eigenvalues of the bulge pencil near the end of the bulge chase, we observe that they never differ from the intended shifts by more than 2×10^{-14}. Thus the shift-transmission error is a bit larger than in the case $m = 1$ but is still tiny.

If we modify the pencil by setting $b_{13,13}$ and $b_{17,17}$ to zero, we get comparable results. Since the zeros float up by two positions per iteration, we deflate an infinite eigenvalue at the top after six iterations, and again after seven. While this is happening, finite eigenvalues are emerging at the bottom. The first four pairs of finite eigenvalues are deflated from the bottom after 7, 14, 20, and 26 iterations, respectively.[1] Quadratic convergence is observed.

Comparing the intended shifts with the eigenvalues of the bulge pencil near the end of the bulge chase, we find that the shift-transmission error never exceeds 1.3×10^{-13}. Thus the shifts are transmitted effectively.

These good results depend upon deflation of the infinite eigenvalues as they emerge. If we fail to do this, the shift-transmission mechanism breaks down and progress toward convergence comes to a halt.

7. Conclusions. We have shown that the reduction to Hessenberg-triangular form and the QZ algorithm both move the infinite eigenvalues toward the top of the pencil. Thus infinite eigenvalues should be deflated at the top. We have identified the mechanism by which shifts are transmitted through the pencil by a bulge chase and have shown that the presence of infinite eigenvalues does not interfere with this mechanism. Finite eigenvalues converge quadratically at the bottom of the pencil, regardless of whether or not infinite eigenvalues are present.

REFERENCES

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[2] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[3] D. S. Watkins, *The transmission of shifts and shift blurring in the QR algorithm*, Linear Algebra Appl., 241–243 (1996), pp. 877–896.

[4] D. S. Watkins, *Bulge exchanges in algorithms of QR type*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1074–1096.

[5] D. S. Watkins, *Infinite eigenvalues and the QZ algorithm*, Preprint SFB393/99-23, Technische Universität Chemnitz, Chemnitz, Germany, 1999; also available online from http://www.tu-chemnitz.de/sfb393/.

[6] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.

[7] D. S. Watkins and L. Elsner, *Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 943–967.

[1] Some of the eigenvalues are real, and they don't always emerge in pairs. For example, the "pair" of eigenvalues that emerged after 14 iterations was really two real eigenvalues that were deflated after 12 and 14 iterations, respectively.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 376–391

ON THE ITERATIVE SOLUTION OF A CLASS OF NONSYMMETRIC ALGEBRAIC RICCATI EQUATIONS*

CHUN-HUA GUO† AND ALAN J. LAUB‡

Abstract. We consider the iterative solution of a class of nonsymmetric algebraic Riccati equations, which includes a class of algebraic Riccati equations arising in transport theory. For any equation in this class, Newton's method and a class of basic fixed-point iterations can be used to find its minimal positive solution whenever it has a positive solution. The properties of these iterative methods are studied and some practical issues are addressed. An algorithm is then proposed to find the minimal positive solution efficiently. Numerical results are also given.

Key words. nonsymmetric algebraic Riccati equations, M-matrices, Newton's method, fixed-point iterations, minimal positive solution, convergence rate

AMS subject classifications. 15A24, 65F10, 82C70

PII. S089547989834980X

1. Introduction.

In transport theory, we encounter nonsymmetric algebraic Riccati equations of the form

$$XCX - XD - AX + B = 0 \tag{1.1}$$

(see [10]), where $A, B, C, D \in \mathbb{R}^{n\times n}$ have the following structures:

$$A = \operatorname{diag}(\delta_1, \delta_2, \ldots, \delta_n) - eq^T, \tag{1.2}$$

$$B = ee^T, \tag{1.3}$$

$$C = qq^T, \tag{1.4}$$

and

$$D = \operatorname{diag}(d_1, d_2, \ldots, d_n) - qe^T. \tag{1.5}$$

In the above,

$$\delta_i = \frac{1}{cw_i(1+\alpha)}, \quad d_i = \frac{1}{cw_i(1-\alpha)}, \tag{1.6}$$

and

$$e = (1, 1, \ldots, 1)^T, \quad q = (q_1, q_2, \ldots, q_n)^T \text{ with } q_i = \frac{c_i}{2w_i}, \tag{1.7}$$

*Received by the editors December 23, 1998; accepted for publication (in revised form) by V. Mehrmann February 21, 2000; published electronically July 11, 2000. This research was supported in part by National Science Foundation grant ECS-9633326.
http://www.siam.org/journals/simax/22-2/34980.html

†Department of Computer Science, University of California, Davis, One Shields Avenue, Davis, CA 95616-8562 (chguo@math.uregina.ca). Current address: Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada. This author was partially supported by an NSERC postdoctoral fellowship.

‡College of Engineering, University of California, Davis, One Shields Avenue, Davis, CA 95616-5294 (laub@ucdavis.edu).

where $0 < c \leq 1,\ 0 \leq \alpha < 1$, and

$$0 < w_n < \cdots < w_2 < w_1 < 1,$$
$$\sum_{i=1}^{n} c_i = 1,\ c_i > 0, \quad i = 1, 2, \ldots, n.$$

For descriptions on how these equations arise in transport theory, see [10] and references cited therein. Here we only note that the constants c and α have physical meanings and the constants c_i and w_i appear in a numerical quadrature formula of the form $\int_0^1 f(w)dw \approx \sum_{i=1}^{n} c_i f(w_i)$.

For any matrices $A, B \in \mathbb{R}^{m\times n}$, we write $A \geq B (A > B)$ if $a_{ij} \geq b_{ij} (a_{ij} > b_{ij})$ for all i, j. We can then define positive matrices, nonnegative matrices, etc. The existence of positive solutions of (1.1) has been shown in [9] and [10]. However, only the minimal positive solution is physically meaningful.

The minimal positive solution of (1.1) can be found by basic fixed-point iterations (see [9], for example). It is mentioned in [10] that the convergence of these fixed-point iterations can be very slow when $c \approx 1$ and $\alpha \approx 0$. In [10], the minimal positive solution of (1.1) is constructed explicitly. The solution formula needs all the zeros of a certain secular equation. To get a good approximation of the minimal positive solution, the secular equation must be solved very accurately. We note that Newton's method is not always valid as a correction method when $c \approx 1$ and $\alpha \approx 0$. This point will be made clear in later discussions.

General nonsymmetric algebraic Riccati equations of the form

$$\mathcal{R}(X) = XCX - XD - AX + B = 0, \tag{1.8}$$

where A, B, C, D are real matrices of sizes $m \times m, m \times n, n \times m, n \times n$, respectively, have also been studied in the literature. See [18], for example. All the solutions of (1.8) can be found, in theory, by finding all the Jordan chains of the matrix

$$H = \begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \tag{1.9}$$

(see Theorem 7.1.2 of [14]). Iterative methods have also been studied for the solution of (1.8). For example, a convergence result for Newton's method is given in [4] under a certain condition on the matrices A, B, C, and D.

Iterative methods with good convergence properties are not available for (1.8) in its full generality. However, for a certain class of these equations, a fairly complete theory can be established for Newton's method and a class of basic fixed-point iterations. Our paper is devoted to the study of these iterative methods.

We start with some definitions. A real square matrix A is called a Z-matrix if all its off-diagonal elements are nonpositive. It is clear that any Z-matrix A can be written as $sI - B$ with $B \geq 0$. A Z-matrix A is called an M-matrix if $s > \rho(B)$, where $\rho(\cdot)$ is the spectral radius. It is called a singular M-matrix if $s = \rho(B)$. Note that A is an M-matrix if and only if A^T is so. Note also that a singular M-matrix is indeed singular ($\rho(B)$ is an eigenvalue of B by the theory of nonnegative matrices; see [21], for example).

The following result is well known (see [2] and [5], for example).

THEOREM 1.1. *For a Z-matrix A, the following are equivalent:*

(1) *A is an M-matrix.*

(2) $A^{-1} \geq 0$.

(3) $Av > 0$ *for some vector* $v > 0$.

(4) *All eigenvalues of* A *have positive real parts.*

The next result is also standard (see [17], for example).

THEOREM 1.2. *Let* $A \in \mathbb{R}^{n\times n}$ *be an* M*-matrix. If the elements of* $B \in \mathbb{R}^{n\times n}$ *satisfy the relations*

$$b_{ii} \geq a_{ii}, \quad a_{ij} \leq b_{ij} \leq 0, \quad i \neq j,\ 1 \leq i, j \leq n,$$

then B *is also an* M*-matrix.*

In this paper we consider nonsymmetric algebraic Riccati equations (1.8) with the following conditions:

$$B > 0, \quad C > 0, \quad I \otimes A + D^T \otimes I \text{ is an } M\text{-matrix}, \tag{1.10}$$

where $\otimes$ is the Kronecker product (for basic properties of the Kronecker product, see [15], for example).

Remark 1.1. It is clear that $I \otimes A + D^T \otimes I$ is a Z-matrix if and only if both A and D are Z-matrices. Since any eigenvalue of $I \otimes A + D^T \otimes I$ is the sum of an eigenvalue of A and an eigenvalue of D (see [15], for example), it follows from the equivalence of (1) and (4) in Theorem 1.1 in this paper that $I \otimes A + D^T \otimes I$ is an M-matrix when A, D are both M-matrices. That the converse is not true is shown by $A = I$ and $D = 0$.

The matrices A and D in (1.1) are both M-matrices by Theorem 1.1 since $Aw > 0$ and $D^T w > 0$ for $w = (w_1, w_2, \ldots, w_n)^T$. Therefore, (1.1) with A, B, C, D defined by (1.2)–(1.7) is a special case of (1.8) with the conditions in (1.10).

From now on, when we speak of (1.8), we always assume that the conditions in (1.10) are satisfied.

2. Newton's method. We now consider the application of Newton's method to the Riccati equation (1.8). For any matrix norm $\mathbb{R}^{m\times n}$ is a Banach space, and the Riccati function $\mathcal{R}$ is a mapping from $\mathbb{R}^{m\times n}$ into itself. The first Fréchet derivative of $\mathcal{R}$ at a matrix X is a linear map $\mathcal{R}_X^{'} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ given by

$$\mathcal{R}'_X(Z) = -\big((A - XC)Z + Z(D - CX)\big). \tag{2.1}$$

Also, the second derivative at X, $\mathcal{R}_X^{''} : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, is given by

$$\mathcal{R}_X^{''}(Z_1, Z_2) = Z_1 C Z_2 + Z_2 C Z_1. \tag{2.2}$$

The Newton method for the solution of (1.8) is

$$X_{i+1} = X_i - (\mathcal{R}_{X_i}^{'})^{-1}\mathcal{R}(X_i), \quad i = 0, 1, \ldots, \tag{2.3}$$

given that the maps $\mathcal{R}_{X_i}^{'}$ are all invertible. In view of (2.1), the iteration (2.3) is equivalent to

$$(A - X_i C)X_{i+1} + X_{i+1}(D - CX_i) = B - X_i C X_i, \quad i = 0, 1, \ldots. \tag{2.4}$$

THEOREM 2.1. *If there is a positive matrix* X *such that* $\mathcal{R}(X) \leq 0$, *then* (1.8) *has a positive solution* S *such that* $S \leq X$ *for every positive matrix* X *for which* $\mathcal{R}(X) \leq 0$. *In particular,* S *is the minimal positive solution of* (1.8). *For the Newton*

iteration (2.3) *with* $X_0 = 0$, *the sequence* $\{X_i\}$ *is well defined,* $X_0 < X_1 < \cdots$, *and* $\lim X_i = S$. *Furthermore, the matrix* S *is such that*

$$M_S = I \otimes (A - SC) + (D - CS)^T \otimes I$$

is either an M*-matrix or a singular* M*-matrix.*

Proof. Let X be any positive matrix such that

$$XCX - XD - AX + B \leq 0. \tag{2.5}$$

For the Newton iteration (2.4) with $X_0 = 0$, we have

$$AX_1 + X_1 D = B.$$

This equation is equivalent to

$$(I \otimes A + D^T \otimes I)\mathrm{vec}X_1 = \mathrm{vec}B, \tag{2.6}$$

where the vec operator stacks the columns of a matrix into one long vector (see [14, p. 99], for example). Since $I \otimes A + D^T \otimes I$ is an M-matrix by assumption, we get from (2.6) that $\mathrm{vec}X_1 > 0$, i.e., $X_1 > 0$. Therefore, the statement

$$X_k < X_{k+1},\ X_k < X,\ I \otimes (A - X_k C) + (D - CX_k)^T \otimes I \text{ is an } M\text{-matrix} \tag{2.7}$$

is true for $k = 0$.

We now assume that (2.7) is true for $k = i \geq 0$. By (2.4) and (2.5) we have

$$\begin{aligned} &(A - X_i C)(X_{i+1} - X) + (X_{i+1} - X)(D - CX_i) \\ &= B - X_i CX_i - AX + X_i CX - XD + XCX_i \\ &\leq -(X - X_i)C(X - X_i). \end{aligned} \tag{2.8}$$

Since $X_i < X$ and $I \otimes (A - X_i C) + (D - CX_i)^T \otimes I$ is an M-matrix, it follows from (2.8) that $X_{i+1} < X$. By (2.4)

$$\begin{aligned} &(A - X_{i+1}C)X_{i+1} + X_{i+1}(D - CX_{i+1}) \\ &= \big(A - X_i C - (X_{i+1} - X_i)C\big)X_{i+1} + X_{i+1}\big(D - CX_i - C(X_{i+1} - X_i)\big) \\ &= B - (X_{i+1} - X_i)C(X_{i+1} - X_i) - X_{i+1}CX_{i+1}. \end{aligned} \tag{2.9}$$

It follows from (2.9) and (2.5) that

$$\begin{aligned} &(A - X_{i+1}C)(X_{i+1} - X) + (X_{i+1} - X)(D - CX_{i+1}) \\ &\leq -(X_{i+1} - X_i)C(X_{i+1} - X_i) - (X_{i+1} - X)C(X_{i+1} - X) < 0. \end{aligned}$$

Therefore,

$$\big(I \otimes (A - X_{i+1}C) + (D - CX_{i+1})^T \otimes I\big)\mathrm{vec}(X - X_{i+1}) > 0.$$

Thus $I \otimes (A - X_{i+1}C) + (D - CX_{i+1})^T \otimes I$ is an M-matrix by Theorem 1.1. By (2.9) and (2.4)

$$\begin{aligned} &(A - X_{i+1}C)(X_{i+1} - X_{i+2}) + (X_{i+1} - X_{i+2})(D - CX_{i+1}) \\ &= -(X_{i+1} - X_i)C(X_{i+1} - X_i) < 0. \end{aligned}$$

Therefore, $X_{i+1} < X_{i+2}$. We have thus proved that (2.7) is true for $k = i + 1$. Hence, by the principle of mathematical induction, (2.7) is true for all $k \geq 0$. The Newton sequence is now well defined, monotonically increasing, and bounded above. Let $\lim_{k\to\infty} X_k = S$. Then S is a solution of (1.8) by (2.4). Since $S \leq X$ for any X such that $\mathcal{R}(X) \leq 0$, S is the minimal positive solution of (1.8). For all $i \geq 0$, we can write $I \otimes (A - X_iC) + (D - CX_i)^T \otimes I = rI - T_i$ with $T_i \geq 0$ and $\rho(T_i) < r$. Now, $M_S = rI - T$ with $T = \lim_{i\to\infty} T_i$. Since $\rho(T) \leq r$, the matrix M_S is either an M-matrix or a singular M-matrix. □

Remark 2.1. The above result is similar in nature to Theorem 9.1.1 of [14]. The result is also somewhat related to a monotone convergence result on Newton's method for convex operators in partially ordered spaces, as described in Theorem 5.1 of [20]. In order to apply that theorem, we need to know that there is a positive matrix X such that $\mathcal{R}(X) \leq 0$ and $I \otimes (A - XC) + (D - CX)^T \otimes I$ is an M-matrix. When this is true, that theorem implies that the Newton sequence with $X_0 = 0$ is well defined, $X_0 \leq X_1 \leq \cdots$, and $\lim X_k = X^* \leq X$ is a solution of $\mathcal{R}(X) = 0$. With the hindsight from Theorem 2.1 in this paper, such a positive matrix X does not exist if $I \otimes (A - SC) + (D - CS)^T \otimes I$ is a singular M-matrix for the minimal positive solution S. In fact, the existence of such an X would imply $S \leq X$ by Theorem 2.1, which would in turn imply that $I \otimes (A - SC) + (D - CS)^T \otimes I$ is an M-matrix by Theorem 1.2.

Remark 2.2. Even if A and D are both M-matrices, it is not necessarily true that $A - SC$ and $D - CS$ are both M-matrices or singular M-matrices. This is shown by the scalar case with $B = C = 1$, $D = 1/2$, and $A = 3/2$. For this example, $S = 1$, $A - SC = 1/2$, and $D - CS = -1/2$. This example also shows that the matrix M_S in Theorem 2.1 can indeed be a singular M-matrix.

The following comparison result is an immediate consequence of Theorem 2.1.

COROLLARY 2.2. *Let S be the minimal solution of* (1.8). *If any element of B or C decreases but remains positive, or if any diagonal element of $I \otimes A + D^T \otimes I$ increases, or if any off-diagonal element of $I \otimes A + D^T \otimes I$ increases but remains nonpositive, then the equation so obtained also has a minimal positive solution $\tilde{S}$. Moreover, $\tilde{S} \leq S$.*

Proof. Let the new equation be

$$\tilde{\mathcal{R}}(X) = X\tilde{C}X - X\tilde{D} - \tilde{A}X + \tilde{B} = 0.$$

It is clear that $\tilde{\mathcal{R}}(S) \leq 0$. Since $I \otimes \tilde{A} + \tilde{D}^T \otimes I$ is still an M-matrix by Theorem 1.2, the conclusions follow from Theorem 2.1. □

Remark 2.3. As an easy consequence of the above corollary, we can conclude that the minimal positive solution of (1.1) increases in c. In [10], it is also concluded that the minimal solution decreases in α. This conclusion is not a consequence of the above corollary and is, in fact, not valid.

Example 2.1. Consider the Riccati equation (1.1) with $n = 2$ and

$$c_1 = c_2 = 1/2, \quad w_1 = 3/4, \quad w_2 = 1/4, \quad c = 1/2.$$

If $\alpha = 0.1$, then the minimal solution (to four digits without rounding) is

$$\begin{pmatrix} 0.2758 & 0.1196 \\ 0.1344 & 0.0766 \end{pmatrix}.$$

If $\alpha = 0.2$, then the minimal solution (to four digits without rounding) is

$$\begin{pmatrix} 0.2639 & 0.1087 \\ 0.1372 & 0.0746 \end{pmatrix}.$$

This example shows that the minimal solution does not necessarily decrease in α.

As to the convergence rate of Newton's method, the following result is immediate.

THEOREM 2.3. *If the matrix M_S in Theorem* 2.1 *is an M-matrix, then for $X_0 = 0$ the Newton sequence $\{X_k\}$ converges to S quadratically.*

Proof. If M_S is an M-matrix, then the Fréchet derivative $\mathcal{R}_S^{'}$ is an invertible map. Since $\mathcal{R}$ is a smooth function, the convergence of the Newton sequence must be quadratic (see [11] and [19], for example). ∎

If the matrix M_S is a singular M-matrix, the map $\mathcal{R}_S^{'}$ is not invertible and the convergence of Newton's method is more complicated. The convergence behavior of Newton's method in this case will be clarified by following the strategy used in [8] for symmetric algebraic Riccati equations and by using a theorem on Newton's method at singular points (see [3, Theorem 1.2] and [12, Theorem 1.1], for example).

LEMMA 2.4. *If M_S is a singular M-matrix, then* 0 *is a simple eigenvalue of M_S. Let $\mathcal{N} = \mathrm{Ker}(\mathcal{R}_S^{'})$ and $\mathcal{M} = \mathrm{Im}(\mathcal{R}_S^{'})$. Then $\mathcal{N}$ is one-dimensional, $\mathbb{R}^{m\times n} = \mathcal{N} \oplus \mathcal{M}$, and the map $\mathcal{B} : \mathcal{N} \to \mathcal{N}$ given by*

$$\mathcal{B}(N) = P_{\mathcal{N}}\mathcal{R}_S^{''}(N_0, N)$$

is invertible for nonzero $N_0 \in \mathcal{N}$, where $P_{\mathcal{N}}$ is the projection on the null space $\mathcal{N}$ parallel to the range $\mathcal{M}$.

Proof. We write $M_S = rI - T$ with $T \geq 0$ and $\rho(T) = r > 0$. Since T is clearly irreducible, we know by the Perron–Frobenius theorem (see [21]) that $\rho(T)$ is a simple eigenvalue of T with a positive eigenvector. Thus, we can find mn orthonormal vectors $u_1, u_2, \ldots, u_{mn}$ such that $u_1 > 0$ and

$$U^{-1}M_S U = \begin{pmatrix} 0 & 0 \\ 0 & M_{22} \end{pmatrix}, \tag{2.10}$$

where $U = (u_1\ u_2\ \cdots u_{mn})$ and M_{22} is an $(mn-1)\times(mn-1)$ nonsingular matrix. Now, $\mathcal{R}_S^{'}(N) = -(A - SC)N - N(D - CS) = 0$ if and only if $M_S\mathrm{vec}N = 0$. In view of (2.10), $M_S\mathrm{vec}N = 0$ if and only if $\mathrm{vec}N = U(a, 0, \ldots, 0)^T = a\,u_1$ for some $a \in \mathbb{R}$, in which case we write $N = a\,\mathrm{unvec}u_1$ (i.e., the unvec operator is the inverse of the vec operator). Thus $\mathcal{N} = \{a\,\mathrm{unvec}u_1 \mid a \in \mathbb{R}\}$. Similarly, $\mathcal{M} = \{b_2\mathrm{unvec}u_2 + \cdots + b_{mn}\mathrm{unvec}u_{mn} \mid b_2, \ldots, b_{mn} \in \mathbb{R}\}$. Therefore, $\mathcal{N}$ is one-dimensional and $\mathbb{R}^{m\times n} = \mathcal{N} \oplus \mathcal{M}$. To prove the map $\mathcal{B}$ is invertible, we only need to show $P_{\mathcal{N}}(\mathrm{unvec}u_1 C\mathrm{unvec}u_1) \neq 0$ (see (2.2)). Since $u_1 > 0$ and $\mathrm{vec}(\mathrm{unvec}u_1 C\mathrm{unvec}u_1) = k_1u_1 + k_2u_2 + \cdots + k_{mn}u_{mn}$ for some real numbers $k_1, k_2, \ldots, k_{mn}$, we have

$$k_1 = u_1^T\,\mathrm{vec}(\mathrm{unvec}u_1 C\mathrm{unvec}u_1) > 0.$$

Thus, $P_{\mathcal{N}}(\mathrm{unvec}u_1 C\mathrm{unvec}u_1) = k_1\mathrm{unvec}u_1 \neq 0$, as required. ∎

LEMMA 2.5. *For any fixed $\theta > 0$, let*

$$Q = \{i :\ \|P_{\mathcal{M}}(X_i - S)\| > \theta\|P_{\mathcal{N}}(X_i - S)\|\}.$$

Then there exist an integer i_0 and a constant $\eta > 0$ such that $\|X_{i+1} - S\| \leq \eta\|X_i - S\|^2$ for all i in Q for which $i \geq i_0$.

Proof. The proof is analogous to that of [8, Theorem 2.2], although the algebraic Riccati equations considered in that paper are different from the Riccati equations being considered here. ∎

COROLLARY 2.6. *Assume that, for given $\theta > 0$, $\|P_{\mathcal{M}}(X_i - S)\| > \theta\|P_{\mathcal{N}}(X_i - S)\|$ for all i large enough. Then $X_i \to S$ quadratically.*

We are now ready to clarify the convergence behavior of Newton's method when the matrix M_S is a singular M-matrix.

THEOREM 2.7. *If M_S is a singular M-matrix and the convergence of the Newton sequence $\{X_i\}$ in Theorem* 2.1 *is not quadratic, then $\|(\mathcal{R}'_{X_i})^{-1}\| \le \beta\|X_i - S\|^{-1}$ for all $i \ge 1$ and some constant $\beta > 0$. Moreover,*

$$\lim_{i\to\infty} \frac{\|X_{i+1} - S\|}{\|X_i - S\|} = \frac{1}{2}, \quad \lim_{i\to\infty} \frac{\|P_{\mathcal{M}}(X_i - S)\|}{\|P_{\mathcal{N}}(X_i - S)\|^2} = 0.$$

Proof. The result follows from Theorem 2.1, Lemma 2.4, Corollary 2.6, and [12, Theorem 1.1]. ∎

3. A class of fixed-point iterations. If we write

$$A = A_1 - A_2, \quad D = D_1 - D_2,$$

(1.8) becomes

$$A_1X + XD_1 = XCX + XD_2 + A_2X + B.$$

We use only those splittings of A and D such that $A_2, D_2 \ge 0$, and A_1 and D_1 are Z-matrices. In these situations, the matrix $I \otimes A_1 + D_1^T \otimes I$ is an M-matrix by Theorem 1.2. We then have a class of fixed-point iterations

$$X_{k+1} = \mathcal{L}^{-1}(X_kCX_k + X_kD_2 + A_2X_k + B), \tag{3.1}$$

where the linear operator $\mathcal{L}$ is given by

$$\mathcal{L}(X) = A_1X + XD_1.$$

Since $I \otimes A_1 + D_1^T \otimes I$ is an M-matrix, the operator $\mathcal{L}$ is invertible and $\mathcal{L}^{-1}(X) > 0$ for $X > 0$.

THEOREM 3.1. *If $\mathcal{R}(X) \le 0$ for some positive matrix X, then for the fixed-point iterations* (3.1) *and $X_0 = 0$, we have for any $k \ge 1$,*

$$X_0 < X_1 < \cdots < X_k < X. \tag{3.2}$$

Moreover, $\lim_{k\to\infty} X_k = S$.

Proof. The order relation (3.2) can easily be proved by induction. The limit X^* is then a solution of $\mathcal{R}(X) = 0$ and must be the minimal positive solution S, since $X^* \le X$ for any positive matrix X such that $\mathcal{R}(X) \le 0$. ∎

Remark 3.1. The comparison result on the minimal positive solution (Corollary 2.2) also follows from the above simple result.

The following result is concerned with the convergence rates of these fixed-point iterations.

THEOREM 3.2. *For the fixed-point iterations* (3.1) *with $X_0 = 0$, we have*

$$\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} = \rho\big((I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\big).$$

Proof. By a theorem on general fixed-point iterations (see [13, p. 21], for example), we have

$$\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} \le \rho(\mathcal{G}'_S), \tag{3.3}$$

where $\mathcal{G}_S^{'}$ is the Fréchet derivative at S of the map $\mathcal{G}$ given by

$$\mathcal{G}(X) = \mathcal{L}^{-1}(XCX + XD_2 + A_2X + B).$$

It is easily found that $\mathcal{G}_S^{'}$ is given by

$$\mathcal{G}_S^{'}(H) = \mathcal{L}^{-1}\big((A_2 + SC)H + H(D_2 + CS)\big).$$

We now show that, in fact, equality holds in (3.3). We may assume the norm in (3.3) is the Frobenius norm.

Let $E_k = S - X_k$. We have $E_{k+1} = P_k(E_k)$, where the operator P_k is given by

$$P_k(H) = \mathcal{L}^{-1}\big((A_2 + SC)H + H(D_2 + CX_k)\big). \tag{3.4}$$

Note that $\lim_{k\to\infty} P_k = \mathcal{G}_S^{'}$. Thus, for any $\epsilon > 0$, we can find an integer l such that

$$\rho(P_l) \geq \rho(\mathcal{G}_S^{'}) - \epsilon.$$

Now, since $0 = X_0 < X_1 < \cdots$, we have

$$\begin{aligned}\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} &= \limsup_{k\to\infty} \sqrt[k]{\|P_{k-1}\cdots P_l P_{l-1}\cdots P_0(S)\|} \\ &\geq \limsup_{k\to\infty} \sqrt[k]{\|(P_l)^{k-l}(P_0)^l(S)\|}.\end{aligned}$$

Since $(P_0)^l(S) > 0$, we have $(P_0)^l(S) > c_l E$, where $c_l > 0$ is a constant and E is the matrix with all its elements equal to one. Also, $\|(P_l)^{k-l}\| = \|(P_l)^{k-l}(S_{l,k})\|$, where $S_{l,k} \in \mathbb{R}^{m\times n}$ is such that $\|S_{l,k}\| = 1$ and $S_{l,k} \geq 0$. Now,

$$\begin{aligned}\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} &\geq \limsup_{k\to\infty} \sqrt[k]{\|c_l(P_l)^{k-l}(E)\|} \\ &\geq \limsup_{k\to\infty} \sqrt[k]{c_l\|(P_l)^{k-l}(S_{l,k})\|} \\ &= \limsup_{k\to\infty} \sqrt[k]{\|(P_l)^{k-l}\|} \\ &= \rho(P_l) \geq \rho(\mathcal{G}_S^{'}) - \epsilon.\end{aligned}$$

Since $\epsilon > 0$ is arbitrary, we have

$$\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} = \rho(\mathcal{G}_S^{'}).$$

A number λ is an eigenvalue of $\mathcal{G}_S^{'}$ if and only if for some $H \neq 0$,

$$\mathcal{L}^{-1}\big((A_2 + SC)H + H(D_2 + CS)\big) = \lambda H,$$

which is the same as

$$(A_2 + SC)H + H(D_2 + CS) = \lambda(A_1H + HD_1)$$

or

$$(I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\mathrm{vec}H = \lambda \mathrm{vec}H.$$

Thus,

$$\rho(\mathcal{G}_S^{'}) = \rho\big((I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\big).$$

This completes the proof. ∎

We can say something more about the spectral radius in Theorem 3.2.

THEOREM 3.3. *If M_S is a singular M-matrix, then*

$$\rho\big((I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\big) = 1.$$

If M_S is an M-matrix, and $A = \tilde{A}_1 - \tilde{A}_2$, $D = \tilde{D}_1 - \tilde{D}_2$ are such that $0 \le \tilde{A}_2 \le A_2$ and $0 \le \tilde{D}_2 \le D_2$, then

$$\begin{aligned}&\rho\big((I \otimes \tilde{A}_1 + \tilde{D}_1^T \otimes I)^{-1}(I \otimes (\tilde{A}_2 + SC) + (\tilde{D}_2 + CS)^T \otimes I)\big)\\ &\le \rho\big((I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\big) < 1.\end{aligned}$$

Proof. Since

$$M_S = (I \otimes A_1 + D_1^T \otimes I) - (I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)$$

and

$$M_S = (I \otimes \tilde{A}_1 + \tilde{D}_1^T \otimes I) - (I \otimes (\tilde{A}_2 + SC) + (\tilde{D}_2 + CS)^T \otimes I)$$

are regular splittings [21] of M_S, the second conclusion follows from the standard results in [21]. If M_S is a singular M-matrix, then $M_S v = 0$ for some $v \neq 0$. Thus,

$$(I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)v = v,$$

and the first conclusion follows. ∎

Therefore, the convergence of these iterations is linear if M_S is an M-matrix. When M_S is a singular M-matrix, the convergence is sublinear. Within this class of iterative methods, three iterations are worthy of special mention. The first one is obtained when we take A_1 and D_1 to be the diagonal part of A and D, respectively. This is the simplest iteration in the class and will be called FP1. The second one is obtained when we take A_1 to be the lower triangular part of A and take D_1 to be the upper triangular part of D. This iteration will be called FP2. The last one is obtained when we take $A_1 = A$ and $D_1 = D$. This is the fastest iteration in this class (see the second part of Theorem 3.3) and will be called FP3.

4. Some practical issues and an overall algorithm. If (1.8) has a positive solution, the minimal positive solution can thus be found by the Newton iteration or some basic fixed-point iterations. Starting with the zero matrix, each of these iterations produces a monotonically increasing sequence, the limit of which is the minimal positive solution S. The matrix M_S associated with S is either an M-matrix or a singular M-matrix. When M_S is an M-matrix, the convergence of Newton's method is quadratic and the convergence of the basic fixed-point iterations is linear. When M_S is a singular M-matrix, the convergence of Newton's method is at least linear and the convergence of the basic fixed-point iterations is sublinear. Therefore, Newton's method is always much faster than the other methods in terms of iteration counts. It must be noted, however, that the computational work involved in one step of Newton's method is much higher than that involved in one step of a basic fixed-point iteration. For the Newton iteration (2.3), the equation $-\mathcal{R}_{X_k}^{'}(H) = \mathcal{R}(X_k)$, i.e.,

$(A - X_kC)H + H(D - CX_k) = \mathcal{R}(X_k)$, can be solved by the algorithms described in [1] and [6]. If we use the Bartels–Stewart algorithm [1] to solve the Sylvester equation, the computational work for each Newton iteration is about $62n^3$ flops when $m = n$ (see [7] for the definition of a "flop"). By comparison, FP1 and FP2 need about $8n^3$ flops for each iteration. For FP3 we can use the Bartels–Stewart algorithm for the first iteration. It needs about $54n^3$ flops. For each subsequent iteration, it needs about $14n^3$ flops.

For the basic fixed-point iteration (3.1), the error reduction at the $(k+1)$th step is determined by the operator P_k in (3.4). Since $0 = X_0 < X_1 < \cdots$, we can see that the error reduction is more significant initially. For Newton's method, of course, the error reduction is much more significant at a late stage of iteration unless the matrix M_S is nearly singular. It is therefore advisable to start with some basic fixed-point iteration and switch to Newton's method after the residual error has been reduced to a certain level. From Theorem 2.1 we know that Newton's method, starting with the zero matrix, produces a monotonically increasing sequence. Now, with the initial guess produced by some basic fixed-point iteration, will the Newton sequence still be monotonic?

PROPOSITION 4.1. *Assume that $\mathcal{R}(X) \le 0$ for some positive matrix X. If $\{X_k\}_{k=1}^{k_0}$ is produced by basic fixed-point iteration* (3.1) *with $X_0 = 0$ and $\{X_k\}_{k=k_0+1}^{\infty}$ is produced by Newton's method with X_{k_0} as an initial guess, then*

$$0 < X_1 < X_2 < \cdots < X_{k_0} < X_{k_0+1} < \cdots,$$

and $\lim_{k\to\infty} X_k = S$, the minimal positive solution.

Proof. We already know from Theorem 3.1 that $0 < X_1 < X_2 < \cdots < X_{k_0} < S$. Now, for $1 \le k \le k_0$, we have

$$\begin{aligned}\mathcal{R}(X_k) &= X_kCX_k + X_kD_2 + A_2X_k + B - A_1X_k - X_kD_1\\ &= X_kCX_k - X_{k-1}CX_{k-1} + (X_k - X_{k-1})D_2 + A_2(X_k - X_{k-1}) > 0.\end{aligned}$$

Since X_{k_0+1} is obtained from X_{k_0} by Newton's method, $-\mathcal{R}'_{X_{k_0}}(X_{k_0+1} - X_{k_0}) = \mathcal{R}(X_{k_0})$. Thus, $(A - X_{k_0}C)(X_{k_0+1} - X_{k_0}) + (X_{k_0+1} - X_{k_0})(D - CX_{k_0}) > 0$. Since $X_{k_0} < S$ and $I \otimes (A - SC) + (D - CS)^T \otimes I$ is either an M-matrix or a singular M-matrix, it follows from the Perron–Frobenius theorem that $I \otimes (A - X_{k_0}C) + (D - CX_{k_0})^T \otimes I$ is an M-matrix. Therefore, $X_{k_0+1} > X_{k_0}$. Once this is proved, it follows as in the proof of Theorem 2.1 that $X_{k_0} < X_{k_0+1} < \cdots$, and $\lim_{k\to\infty} X_k = S$. □

Remark 4.1. We may apply the above strategy without knowing whether (1.8) has a positive solution. If we find that $X_k < X_{k+1}$ is not true for some $k \ge k_0$, then we can conclude that (1.8) does not have a positive solution. Note, however, that $X_k < X_{k+1}$ is true for all $0 \le k < k_0$, even if (1.8) has no positive solutions. This is another difference between Newton's method and the basic fixed-point iterations.

Remark 4.2. The results in section 2 are still valid when the Newton iteration is started with a matrix produced by a basic fixed-point iteration as in Proposition 4.1.

The convergence behavior of the iterative methods we have discussed depends on the matrix $M_S = I \otimes (A - SC) + (D - CS)^T \otimes I$, in which S is the minimal positive solution to be found. The matrix M_S is a singular M-matrix if and only if $\lambda_i + \mu_j = 0$ for some eigenvalue λ_i of $A - SC$ and some eigenvalue μ_j of $D - CS$. There is some connection between the eigenvalues of $A - SC$ (or $D - CS$) and the eigenvalues of the matrix H in (1.9). In fact, the following result is true.

PROPOSITION 4.2. *If X is any solution of* (1.8)*, then any eigenvalue of $D-CX$ is an eigenvalue of H and any eigenvalue of $A-XC$ is the negative of some eigenvalue of H.*

Proof. It is easy to verify that

$$\begin{pmatrix} I & 0 \\ X & I \end{pmatrix}^{-1} \begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} I & 0 \\ X & I \end{pmatrix} = \begin{pmatrix} D-CX & -C \\ 0 & -(A-XC) \end{pmatrix}.$$

The conclusions follow immediately. ∎

However, when we are going to use iterative methods to find the minimal positive solution, we would not bother to find all the eigenvalues of the matrix H. Even if we know all the eigenvalues of H, Proposition 4.2 is not adequate to determine all the eigenvalues of $A-SC$ and $D-CS$. For (1.1), we know from the results in [10] that M_S is a singular M-matrix if and only if $c=1$ and $\alpha=0$. This explains why Newton's method may not be valid as a correction method when $c\approx 1$ and $\alpha\approx 0$. For (1.8), whether the matrix M_S is a singular M-matrix (or nearly so) can be inferred from the speed of convergence of the iterative method we are using. For example, very slow convergence of a basic fixed-point iteration indicates that the matrix M_S is a singular M-matrix or nearly so. By Proposition 4.1 we can always use the Newton iteration when the convergence of the fixed-point iteration is unsatisfactory.

When the matrix M_S is singular and the convergence of Newton's method is not quadratic, we know from Theorem 2.7 that the convergence must be linear with rate $1/2$ and the error will rapidly be dominated by the null space component. As is the case for symmetric algebraic Riccati equations (see Theorems 3.1 and 3.2 of [8]), very accurate approximation for the minimal positive solution can be obtained by computing $Y_{k+1}=X_k-2(\mathcal{R}'_{X_k})^{-1}\mathcal{R}(X_k)$ when $\|P_{\mathcal{M}}(X_k-S)\|\le\epsilon\|P_{\mathcal{N}}(X_k-S)\|$ and ϵ is very small. Note that $\|X_k-S\|$ need not be very small when ϵ is very small. In this case, the Sylvester equation $-\mathcal{R}'_{X_k}(H)=\mathcal{R}(X_k)$ is not nearly singular and can be solved by the Bartels–Stewart algorithm very accurately. Without this double Newton step, Newton's method will take many more iterations. Even linear convergence with rate $1/2$ can fail to be realized due to a nearly singular Jacobian at a late stage. Therefore, when we apply Newton's method, we can *try* a double Newton step first. If the approximation obtained fails to satisfy a given stopping criterion, then we use the original Newton iteration instead and try a double Newton step with the new iterate, i.e., we have an algorithm similar to Algorithm 3.3 of [8] for symmetric algebraic Riccati equations. Although the added cost of trying the double Newton step is minor, the strategy can be used in a wiser way. That is, we can try the double Newton step only when there are indications that we are solving a problem with $\mathcal{R}'_S$ singular (or nearly singular) and that the error is already essentially in the null space (or approximate null space). The next result shows how we can get such indications.

PROPOSITION 4.3. *Assume that $\mathcal{R}'_S$ is singular and $\{X_k\}_{k=k_0}^{\infty}$ is the Newton sequence in Proposition* 4.1*. If $X_k-S\in\mathcal{N}(k\ge k_0)$, then*

$$X_{k+1}-S=\frac{1}{2}(X_k-S),\quad \mathcal{R}(X_{k+1})=\frac{1}{4}\mathcal{R}(X_k).$$

Furthermore,

$$\lim_{r_k\to 0}\frac{\|X_{k+1}-S\|}{\|X_k-S\|}=\frac{1}{2},\quad \lim_{r_k\to 0}\frac{\mathcal{R}(X_k)}{\|X_k-S\|^2}=C_0, \tag{4.1}$$

where

$$r_k = \frac{\|P_{\mathcal{M}}(X_k - S)\|}{\|P_{\mathcal{N}}(X_k - S)\|^2}$$

and C_0 is a constant positive matrix. In particular,

$$\lim_{r_k \to 0} \frac{\|\mathcal{R}(X_{k+1})\|}{\|\mathcal{R}(X_k)\|} = \frac{1}{4}.$$

Proof. As in Theorem 3.1 of [8], we have $X_{k+1} - S = \frac{1}{2}(X_k - S) \in \mathcal{N}$ when $X_k - S \in \mathcal{N}$. Thus, in view of (2.2),

$$\begin{aligned}\mathcal{R}(X_{k+1}) &= \mathcal{R}(S) + \mathcal{R}_S^{'}(X_{k+1} - S) + \frac{1}{2}\mathcal{R}_S^{''}(X_{k+1} - S, X_{k+1} - S) \\ &= (X_{k+1} - S)C(X_{k+1} - S) = \frac{1}{4}(X_k - S)C(X_k - S) = \frac{1}{4}\mathcal{R}(X_k).\end{aligned}$$

If r_k is sufficiently small, we have as in Theorem 3.2 of [8]

$$\|X_k - 2(\mathcal{R}_{X_k}^{'})^{-1}\mathcal{R}(X_k) - S\| \le \gamma \frac{\|P_{\mathcal{M}}(X_k - S)\|}{\|P_{\mathcal{N}}(X_k - S)\|} \tag{4.2}$$

for some constant γ. Since the left-hand side of (4.2) can be written as $\|2(X_{k+1} - S) - (X_k - S)\|$, the first limit in (4.1) follows easily. Now, let $\mathcal{N} = \text{span}\{N_0\}$ with $N_0 > 0$ and $\|N_0\| = 1$. Since

$$\begin{aligned}\mathcal{R}(X_k) &= \mathcal{R}(S) + \mathcal{R}_S^{'}(X_k - S) + \frac{1}{2}\mathcal{R}_S^{''}(X_k - S, X_k - S) \\ &= \mathcal{R}_S^{'}(P_{\mathcal{M}}(X_k - S)) + (X_k - S)C(X_k - S),\end{aligned}$$

we get easily that

$$\lim_{r_k \to 0} \frac{\mathcal{R}(X_k)}{\|X_k - S\|^2} = N_0 C N_0 > 0.$$

The proof is thus complete. □

When $\mathcal{R}_S^{'}$ is singular, we know from Theorem 2.7 that $\lim_{k\to\infty} r_k = 0$ unless the convergence of Newton's method is quadratic. The above proposition tells us that we may choose to try the double Newton step with a current Newton iterate X_k only when $\|\mathcal{R}(X_k)\|/\|\mathcal{R}(X_{k-1})\| \approx 1/4$.

We now propose the following algorithm for finding the minimal positive solution S of (1.8) whenever it has a positive solution. The algorithm may also detect that the equation actually does not have a positive solution. The choices of the splittings and parameters in step 1 of the algorithm can be made according to the guidelines provided immediately after the algorithm.

ALGORITHM 4.4.

(1) *Choose splittings $A = A_1 - A_2$ and $D = D_1 - D_2$;*
    choose parameters $k_0, \epsilon, \eta_1, \eta_2, \eta_3 > 0$.
(2) *Set $X_0 = 0$, $T(X_0) = B$, $r_0 = \|B\|_\infty$.*
(3) *For $k = 1, 2, \ldots$, do:*
    solve $A_1 X_k + X_k D_1 = T(X_{k-1})$;
    compute $\mathcal{R}(X_k)$, $r_k = \|\mathcal{R}(X_k)\|_\infty$;
    if $r_k/r_0 < \eta_1$ or $k \ge k_0$, goto 4;
    compute $T(X_k) = T(X_{k-1}) + \mathcal{R}(X_k)$.

(4) *For* $p = k, k+1, \ldots,$ *do:*
solve $-\mathcal{R}'_{X_k}(H) = \mathcal{R}(X_k)$ *for* $H = (h_{ij})$*;*
if $h_{ij} < -\eta_2 \|H\|_\infty$ *for some* (i,j)*, then stop (no solution);*
compute $X_{p+1} = X_p + H$, $\mathcal{R}(X_{p+1})$, $r_{p+1} = \|\mathcal{R}(X_{p+1})\|_\infty$;
if $r_{p+1}/r_0 < \epsilon$*, then stop and* $S \approx X_{p+1}$*;*
if $|\frac{r_{p+1}}{r_p} - \frac{1}{4}| < \eta_3$*, then*
compute $Z = X_p + 2H$, $r = \|\mathcal{R}(Z)\|_\infty$;
if $r/r_0 < \epsilon$*, then stop and* $S \approx Z$*.*

In the above algorithm, we can select a particular basic fixed-point iteration by choosing proper splittings of A and D. Normally we can use FP1 or FP2. Although FP2 is faster in general, FP1 may take advantage of the structures in a specific equation more easily. In the algorithm, ϵ is the required precision and is usually much smaller than η_1. The small number η_2 is introduced to numerically check if $H > 0$ has been violated. This number should be related to the unit roundoff. The small number η_3 is used to determine if the double Newton step should be tried. A smaller η_3 should be used for a smaller ϵ. If (1.8) does not have a positive solution, the criterion $r_k/r_0 < \eta_1$ in step 3 of the algorithm may never be satisfied. In the algorithm, we have let k_0 be the maximal number of fixed-point iterations allowed. The nonexistence of a positive solution can often be detected by Newton's method in step 4. When (1.8) has a positive solution, the algorithm will produce a finite sequence approaching the minimal positive solution. The sequence is obtained by a fixed-point iteration followed by ordinary Newton's method, with the exception that the last term in the sequence is possibly obtained by the double Newton step. It should be noted that the matrix Z produced by the double Newton step is not used in subsequent Newton iterations.

5. Numerical results. First we give a simple example to illustrate the performance of the iterative methods we have studied.

Example 5.1. Consider (1.8) with $m = n = 2$ and

$$A = \begin{pmatrix} \alpha & -2 \\ -1 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 3 & 4 \\ 2 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 5 & -1 \\ -1 & 4 \end{pmatrix}.$$

For $\alpha = 4.26$, we apply Newton's method with $X_0 = 0$ and find

$$X_6 = \begin{pmatrix} 0.3865 & 0.4048 \\ 0.3583 & 0.2943 \end{pmatrix}, \quad X_7 = \begin{pmatrix} 0.3713 & 0.3872 \\ 0.3490 & 0.2836 \end{pmatrix}.$$

Since $X_6 < X_7$ is not true, the equation has no positive solutions in this case. Experiments show that the equation has a positive solution for $\alpha = 4.267191$. Thus, it has positive solutions for all $\alpha \geq 4.267191$ by Corollary 2.2. In Tables 5.1–5.3, we have recorded, for three values of α, the number of iterations needed to have $\|\mathcal{R}(X_k)\|_\infty < \epsilon$ for Newton's method (NM) and the three basic fixed-point iterations. For all four methods, we use $X_0 = 0$. From the tables, we can see that the three basic fixed-point iterations have similar efficiency. For $\alpha = 6.0$, the basic fixed-point iterations are still adequate. For $\alpha = 4.27$ and $\alpha = 4.267191$, however, the advantage of Newton's method is very clear. In all three cases, the basic fixed-point iterations are useful for initial error reduction. We may consider using Newton's method after a certain number of fixed-point iterations. However, other features of Algorithm 4.4 have no role to play, since the existence of a positive solution is known for each α and quadratic convergence of Newton's method is visible even for $\alpha = 4.267191$.

TABLE 5.1
Iteration counts for Example 5.1, $\alpha = 6.0$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
NM	3	4	4	5	5	5
FP1	11	22	33	44	54	65
FP2	10	19	29	38	48	57
FP3	7	15	23	31	38	46

TABLE 5.2
Iteration counts for Example 5.1, $\alpha = 4.27$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
NM	5	7	8	9	9	10
FP1	40	245	533	822	1112	1402
FP2	36	222	480	739	998	1257
FP3	29	182	396	611	827	1042

TABLE 5.3
Iteration counts for Example 5.1, $\alpha = 4.267191$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
NM	5	8	11	14	15	15
FP1	40	450	4477	25328	54350	83603
FP2	37	414	4119	23000	49020	75239
FP3	29	335	3339	18899	40559	62395

Example 5.2. We now consider (1.1) for $n = 64$ and $n = 128$. The constants c_i and w_i are given by a numerical quadrature formula on the interval $[0, 1]$, which is obtained by dividing $[0, 1]$ into $n/4$ subintervals of equal length and applying Gauss–Legendre quadrature with four nodes to each subinterval.

We apply Algorithm 4.4 with the splittings of A and D being those corresponding to FP1, and we take $k_0 = 200$, $\epsilon = 10^{-12}$, $\eta_1 = 10^{-3}$, $\eta_2 = 10^{-6}$, and $\eta_3 = 10^{-6}$. For this example it is actually unnecessary to introduce the parameter η_2, since the existence of positive solutions has been guaranteed by the theoretical results in [9] and [10]. We carry out the computation for $n = 64$ and $n = 128$. The parameter pair (α, c) is taken to be $(0.5, 0.5), (10^{-8}, 0.999999), (10^{-14}, 1)$, and $(0, 1)$. The results are recorded in Tables 5.4–5.5. For example, when $n = 64$ and $(\alpha, c) = (0, 1)$, the residual is reduced to 0.9916D-03r_0 after 170 FP1 iterations (r_0 is the initial residual). The residual is then reduced to 0.4937D-05r_0 after four Newton iterations. The fifth Newton iteration fails to achieve the required accuracy, but the double Newton step (DN) works (it reduces the residual to 0.1763D-13r_0). The double Newton step is also tried with the fourth Newton iteration, but without success. For this example, $\mathcal{R}_S'$ is singular when $(\alpha, c) = (0, 1)$.

TABLE 5.4
Convergence history for Example 5.2, $n = 64$.

(0.5, 0.5)	(10^{-8}, 0.999999)	(10^{-14}, 1)	(0, 1)
5 FP1	170 FP1	170 FP1	170 FP1
0.6844D-03	0.9889D-03	0.9916D-03	0.9916D-03
2 NM	7 NM	4 NM	4 NM
0.5464D-15	0.5832D-14	0.4937D-05	0.4937D-05
no DN tries	no DN tries	DN (second try)	DN (second try)
		0.1671D-13	0.1763D-13

TABLE 5.5
Convergence history for Example 5.2, $n = 128$.

(0.5, 0.5)	$(10^{-8}, 0.999999)$	$(10^{-14}, 1)$	(0, 1)
5 FP1	170 FP1	170 FP1	170 FP1
0.6847D-03	0.9915D-03	0.9942D-03	0.9942D-03
2 NM	7 NM	4 NM	4 NM
0.1117D-14	0.5677D-14	0.4953D-05	0.4953D-05
no DN tries	no DN tries	DN (second try)	DN (second try)
		0.1606D-13	0.1650D-13

6. Conclusions. We have discussed the iterative solution of a class of nonsymmetric algebraic Riccati equations, which includes a class of algebraic Riccati equations arising in transport theory. The coefficient matrices of any equation in this larger class have a special sign structure. Using this structure and the theory of M-matrices, we have shown that Newton's method and a class of basic fixed-point iterations can be used to find its minimal positive solution whenever it has a positive solution. We have also proposed an overall algorithm for the solution of the nonsymmetric algebraic Riccati equation. The algorithm is basically a combination of Newton's method and a basic fixed-point iteration, but it has two additional features: (1) the algorithm can detect that an equation actually does not have a positive solution, and (2) it can also detect and solve a singular or nearly singular problem efficiently. There are still, however, some unsolved problems about the nonsymmetric algebraic Riccati equation. For example, it is of interest to know what reasonable conditions on the coefficient matrices of the equation will ensure the existence of a positive solution. It is also of interest to determine if quadratic convergence is really possible for Newton's method in the singular case. For symmetric algebraic Riccati equations, subspace methods are frequently used (see [16], for example). It would be worthwhile to consider whether the minimal positive solution of the equation can also be found efficiently by subspace methods.

Acknowledgments. Chun-Hua Guo would like to thank Peter Lancaster for introducing him to the study of algebraic Riccati equations several years ago. He also gratefully acknowledges the support of an NSERC postdoctoral fellowship. Both authors thank the referees for their very helpful comments.

REFERENCES

[1] R. H. BARTELS AND G. W. STEWART, *Solution of the matrix equation* $AX + XB = C$, Comm. ACM, 15 (1972), pp. 820–826.
[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
[3] D. W. DECKER AND C. T. KELLEY, *Newton's method at singular points.* I, SIAM J. Numer. Anal., 17 (1980), pp. 66–70.
[4] J. W. DEMMEL, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.
[5] M. FIEDLER AND V. PTAK, *On matrices with non-positive off-diagonal elements and positive principal minors*, Czechoslovak Math. J., 12 (1962), pp. 382–400.
[6] G. H. GOLUB, S. NASH, AND C. VAN LOAN, *A Hessenberg-Schur method for the problem* $AX + XB = C$, IEEE Trans. Automat. Control, 24 (1979), pp. 909–913.
[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[8] C.-H. GUO AND P. LANCASTER, *Analysis and modification of Newton's method for algebraic Riccati equations*, Math. Comp., 67 (1998), pp. 1089–1105.

[9] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.
[10] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 228–243.
[11] L. V. Kantorovich and G. P. Akilov, *Functional Analysis in Normed Spaces*, Pergamon, New York, 1964.
[12] C. T. Kelley, *A Shamanskii-like acceleration scheme for nonlinear equations at singular roots*, Math. Comp., 47 (1986), pp. 609–623.
[13] M. A. Krasnoselskii, G. M. Vainikko, P. P. Zabreiko, Ya. B. Rutitskii, and V. Ya. Stetsenko, *Approximate Solution of Operator Equations*, Wolters-Noordhoff Publishing, Groningen, The Netherlands, 1972.
[14] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, London, 1995.
[15] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, Orlando, FL, 1985.
[16] A. J. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.
[17] 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.
[18] H.-B. Meyer, *The matrix equation $AZ+B-ZCZ-ZD=0$*, SIAM J. Appl. Math., 30 (1976), pp. 136–142.
[19] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[20] J. S. Vandergraft, *Newton's method for convex operators in partially ordered spaces*, SIAM J. Numer. Anal., 4 (1967), pp. 406–432.
[21] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 392–412

THE RECURSIVE INVERSE EIGENVALUE PROBLEM*

MARINA ARAV†, DANIEL HERSHKOWITZ†, VOLKER MEHRMANN‡, AND HANS SCHNEIDER§

Abstract. The recursive inverse eigenvalue problem for matrices is studied, where for each leading principal submatrix an eigenvalue and associated left and right eigenvectors are assigned. Existence and uniqueness results as well as explicit formulas are proven, and applications to nonnegative matrices, Z-matrices, M-matrices, symmetric matrices, Stieltjes matrices, and inverse M-matrices are considered.

Key words. inverse eigenvalue problem, recursive solution, nonnegative matrices, Z-matrices, M-matrices, Hermitian matrices, Stieltjes matrices, inverse M-matrices

AMS subject classifications. 15A29, 15A18, 15A48, 15A57

PII. S0895479899354044

1. Introduction.

Inverse eigenvalue problems are a very important subclass of inverse problems that arise in the context of mathematical modeling and parameter identification. They have been studied extensively in the last 20 years; see, e.g., [3, 5, 6, 8, 12, 13, 14] and the references therein. In particular, the inverse eigenvalue problem for nonnegative matrices is still a topic of very active research, since a necessary and sufficient condition for the existence of a nonnegative matrix with a prescribed spectrum is still an open problem; see [4, 12].

In this paper we study inverse eigenvalue problems in a recursive matter, which allows us to extend existing solutions.

We investigate the following *recursive inverse eigenvalue problem of order n*:

Let F be a field, let $s_1, \ldots, s_n \in F$, and let

$$l_1 = \begin{bmatrix} l_{1,1} \end{bmatrix},\ l_2 = \begin{bmatrix} l_{2,1} \\ l_{2,2} \end{bmatrix},\ \ldots,\ l_n = \begin{bmatrix} l_{n,1} \\ \vdots \\ l_{n,n} \end{bmatrix},$$

$$r_1 = \begin{bmatrix} r_{1,1} \end{bmatrix},\ r_2 = \begin{bmatrix} r_{1,2} \\ r_{2,2} \end{bmatrix},\ \ldots,\ r_n = \begin{bmatrix} r_{1,n} \\ \vdots \\ r_{n,n} \end{bmatrix}$$

be vectors with elements in F. Construct a matrix $A \in F^{n,n}$ such that

$$\begin{cases} l_i^T A\langle i\rangle = s_i l_i^T, \\ \\ A\langle i\rangle r_i = s_i r_i, \end{cases} \qquad i = 1, \ldots, n,$$

*Received by the editors April 12, 1999; accepted for publication (in revised form) by M. Chu January 11, 2000; published electronically July 11, 2000. The work of the second, third, and fourth authors was partially supported by Sonderforschungsbereich 393 *Numerische Simulation auf massive parallelen Rechnern* at TU Chemnitz.

http://www.siam.org/journals/simax/22-2/35404.html

†Department of Mathematics, Technion, Haifa 32000, Israel (hershkow@tx.technion.ac.il).

‡Fakultät für Mathematik, TU Chemnitz, D-09107 Chemnitz, Germany (mehrmann@mathematik.tu-chemnitz.de).

§Mathematics Department, University of Wisconsin, Madison, WI 53706 (hans@math.wisc.edu).

where $A\langle i\rangle$ denotes the ith leading principal submatrix of A.

In what follows we shall use the notation **RIEP**(n) for "the recursive inverse eigenvalue problem of order n."

It should be noted that most of the results that we present below are recursive solutions (hence the name *recursive inverse eigenvalue problem*) in the sense that the existence and uniqueness conditions are of a recursive nature, i.e., once the existence and/or uniqueness of **RIEP**$(n-1)$ have been established, the presented conditions describe when the solution of **RIEP**(n) exists and is unique. Such results are very useful, in particular when a solution has been computed and new data later become available.

To consider a simple example, let us consider a closed Leontief model in economics which is typically described by the action of a nonnegative matrix T with spectral radius 1 on a vector; see, e.g., [2]. A nonnegative eigenvector associated with the eigenvalue 1 of the matrix T then is an equilibrium point of the model [2]; a model that has such a vector is called feasible.

Suppose now that a feasible model with $n-1$ inputs and $n-1$ outputs, i.e., an $(n-1)\times(n-1)$ nonnegative matrix that describes the model and has an equilibrium point, has been constructed.

An immediate question then is whether adding an input and output to the system can again lead to a feasible model with prescribed equilibrium point. This immediately leads to the recursive inverse eigenvalue problem. Our results give necessary and sufficient conditions for several classes of matrices including nonnegative matrices and M-matrices which are the classes of interest in Leontief models and the analysis of Markov chains; see [2].

Existence and uniqueness of nonnegative solutions is therefore one of the major topics of this paper.

In section 2 we study the existence and uniqueness of solutions for **RIEP**(n) in the general case. Our main result gives a recursive characterization of the solution for **RIEP**(n). We also obtain a nonrecursive necessary and sufficient condition for unique solvability as well as an explicit formula for the solution in case of uniqueness.

The results of section 2 are applied in the subsequent sections to special cases. In section 3 we discuss nonnegative solutions for **RIEP**(n) over the field $\mathbb{R}$ of real numbers. We also introduce a nonrecursive sufficient condition for the existence of a nonnegative solution for **RIEP**(n). Uniqueness of nonnegative solutions for **RIEP**(n) is discussed in section 4. In section 5 we study Z-matrix and M-matrix solutions for **RIEP**(n) over $\mathbb{R}$. In section 6 we consider real symmetric solutions for **RIEP**(n) over $\mathbb{R}$. In section 7 we consider positive semidefinite real symmetric solutions for **RIEP**(n) over $\mathbb{R}$. In section 8 we combine the results of the previous two sections to obtain analogous results for Stieltjes matrices. Finally, in section 9 we investigate inverse M-matrix solutions for **RIEP**(n). A summary is given in section 10.

2. Existence and uniqueness results. In this section we study the existence and uniqueness of solutions for **RIEP**(n) in the general case. For this purpose we introduce some further notation. For the vectors l_i, r_i we set

$$\tilde{l}_i = \begin{bmatrix} l_{i,1} \\ \vdots \\ l_{i,i-1} \end{bmatrix}, \quad \tilde{r}_i = \begin{bmatrix} r_{1,i} \\ \vdots \\ r_{i-1,i} \end{bmatrix}.$$

The case $n = 1$ is easy to verify.

PROPOSITION 1. *If $l_{1,1} = r_{1,1} = 0$, then every 1×1 matrix A solves* **RIEP**(1). *If either $l_{1,1} \neq 0$ or $r_{1,1} \neq 0$, then $A = [s_1]$ is the unique solution for* **RIEP**(1).

For $n \geq 2$ we have the following recursive characterization of the solution for **RIEP**(n).

THEOREM 2. *Let $n \geq 2$. There exists a solution for* **RIEP**(n) *if and only if there exists a solution B for* **RIEP**$(n\text{-}1)$ *such that*

$$l_{n,n} = 0 \quad \Longrightarrow \quad \tilde{l}_n^T B = s_n \tilde{l}_n^T \tag{1}$$

and

$$r_{n,n} = 0 \quad \Longrightarrow \quad B\tilde{r}_n = s_n \tilde{r}_n. \tag{2}$$

There exists a unique solution for **RIEP**(n) *if and only if there exists a unique solution for* **RIEP**$(n\text{-}1)$ *and $l_{n,n} r_{n,n} \neq 0$.*

Proof. Let A be an $n \times n$ matrix. Partition A as

$$A = \begin{bmatrix} B & y \\ x^T & z \end{bmatrix}, \tag{3}$$

where B is an (n-1) $\times$ (n-1) matrix. Clearly, A solves **RIEP**(n) if and only if B solves **RIEP**(n-1) and

$$(s_n I_{n-1} - B)\tilde{r}_n = r_{n,n} y, \tag{4}$$

$$\tilde{l}_n^T (s_n I_{n-1} - B) = l_{n,n} x^T, \tag{5}$$

$$x^T \tilde{r}_n + z r_{n,n} = s_n r_{n,n}, \tag{6}$$

$$\tilde{l}_n^T y + z l_{n,n} = s_n l_{n,n}. \tag{7}$$

It thus follows that there exists a solution for **RIEP**(n) if and only if there exists a solution B for **RIEP**(n-1) such that (4)–(7) (with unknown x, y, and z) are solvable. We now show that these equations are solvable if and only if (1) and (2) hold. Distinguish between four cases:

1. $r_{n,n} = 0,\ l_{n,n} \neq 0$. Here (4) is equivalent to (2), (5) is equivalent to
$$x^T = \frac{\tilde{l}_n^T (s_n I_{n-1} - B)}{l_{n,n}}, \tag{8}$$
and (6) then follows from (4). For every $y \in F^{n-1}$ we can find $z \in F$ such that (7) holds.
2. $l_{n,n} = 0,\ r_{n,n} \neq 0$. Here (5) is equivalent to (1), (4) is equivalent to
$$y = \frac{(s_n I_{n-1} - B)\tilde{r}_n}{r_{n,n}}, \tag{9}$$
and (7) then follows from (5). For every $x \in F^{n-1}$ we can find $z \in F$ such that (6) holds.
3. $l_{n,n} = r_{n,n} = 0$. Here (4) is equivalent to (2) and (5) is equivalent to (1). For any $x \in F^{n-1}$ with $x^T \tilde{r}_n = 0$ we have (6), and for any $y \in F^{n-1}$ with $\tilde{l}_n^T y = 0$ we have (7), where z can be chosen arbitrarily.
4. $l_{n,n} \neq 0,\ r_{n,n} \neq 0$. Here (4)–(7) have a unique solution, given by (8), (9), and
$$z = s_n - \frac{\tilde{l}_n^T (s_n I_{n-1} - B)\tilde{r}_n}{l_{n,n} r_{n,n}}. \tag{10}$$

It follows that (4)–(7) are solvable if and only if (1) and (2) hold.

To prove the uniqueness assertion, note that it follows from our proof that if either $l_{n,n} = 0$ or $r_{n,n} = 0$, then a solution is not unique, since at least one of the vectors x, y, and z can be chosen arbitrarily. If both $l_{n,n} \neq 0$ and $r_{n,n} \neq 0$, then every solution B for **RIEP**(n-1) defines a unique solution A for **RIEP**(n). The uniqueness claim follows. ∎

This result is recursive and allows us to derive a recursive algorithm to compute the solution, but we do not get explicit nonrecursive conditions that characterize the existence of solutions. In order to get a necessary and sufficient condition for unique solvability as well as an explicit formula for the solution in case of uniqueness, we define the $n \times n$ matrix R_n to be the matrix whose columns are $r_1, \ldots, r_n$ with zeros appended at the bottom to obtain n-vectors. Similarly, we define the $n \times n$ matrix L_n to be the matrix whose rows are $l_1, \ldots, l_n$ with zeros appended at the right to obtain n-vectors. That is, we have

$$L_n = \begin{bmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & & \ddots & \\ l_{n,1} & \cdots & l_{n,n-1} & l_{n,n} \end{bmatrix}, \quad R_n = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ & r_{2,2} & & \vdots \\ & & \ddots & r_{n-1,n} \\ & & & r_{n,n} \end{bmatrix}. \tag{11}$$

We denote

$$S_n = \begin{bmatrix} s_1 & s_2 & s_3 & \cdots & s_n \\ s_2 & s_2 & s_3 & \cdots & s_n \\ s_3 & s_3 & s_3 & \cdots & s_n \\ \vdots & & & & \vdots \\ s_n & s_n & \cdots & \cdots & s_n \end{bmatrix}. \tag{12}$$

Also, we denote by $\circ$ the Hadamard (or elementwise) product of matrices.

PROPOSITION 3. *A solution A for* **RIEP**(n) *satisfies*

$$L_n A R_n = S_n \circ (L_n R_n). \tag{13}$$

Proof. We prove our claim by induction on n. For $n = 1$ the claim follows easily. Assume that the assertion holds for $n < k$ and let $n = k$. Partition A as in (3). We have

$$\begin{aligned} L_n A R_n &= \begin{bmatrix} L_{n-1} & 0 \\ \tilde{l}_n^T & l_{n,n} \end{bmatrix} \begin{bmatrix} B & y \\ x^T & z \end{bmatrix} \begin{bmatrix} R_{n-1} & \tilde{r}_n \\ 0 & r_{n,n} \end{bmatrix} \\ &= \begin{bmatrix} L_{n-1} B R_{n-1} & L_{n-1}(B\tilde{r}_n + r_{n,n} y) \\ (\tilde{l}_n^T B + l_{n,n} x^T) R_{n-1} & (\tilde{l}_n^T B + l_{n,n} x^T)\tilde{r}_n + (\tilde{l}_n^T y + l_{n,n} z) r_{n,n} \end{bmatrix}. \end{aligned}$$

By the inductive assumption we have $L_{n-1} B R_{n-1} = S_{n-1} \circ (L_{n-1} R_{n-1})$. Also, by (4) we have $B\tilde{r}_n + r_{n,n} y = s_n \tilde{r}_n$, by (5) we have $\tilde{l}_n^T B + l_{n,n} x^T = s_n \tilde{l}_n^T$, and by (7) we have $\tilde{l}_n^T y + l_{n,n} z = s_n l_{n,n}$. It thus follows that

$$L_n A R_n = \begin{bmatrix} S_{n-1} \circ (L_{n-1} R_{n-1}) & s_n L_{n-1} \tilde{r}_n \\ s_n \tilde{l}_n^T R_{n-1} & s_n(\tilde{l}_n^T \tilde{r}_n + l_{n,n} r_{n,n}) \end{bmatrix} = S_n \circ (L_n R_n). \quad ∎$$

In general, the converse of Proposition 3 does not hold, that is, a matrix A satisfying (13) does not necessarily form a solution for **RIEP**(n), as is demonstrated by Example 5 below.

THEOREM 4. *There is a unique solution for* **RIEP**(n) *if and only if*

$$l_{1,1} \neq 0 \quad or \quad r_{1,1} \neq 0$$

and

$$l_{i,i} r_{i,i} \neq 0, \quad i = 1, \ldots, n.$$

Furthermore, the unique solution is given by

$$A = L_n^{-1}[S_n \circ (L_n R_n)]R_n^{-1}. \tag{14}$$

Proof. The uniqueness claim follows from Proposition 1 and Theorem 2. The fact that the unique solution for **RIEP**(n) is given by (14) follows immediately from Proposition 3. □

In the case that the solution is not unique, that is, whenever $l_{1,1} = r_{1,1} = 0$ or whenever $l_{i,i}$ or $r_{i,i}$ vanish for some $i > 1$, the matrices L_n and R_n defined in (11) are not invertible. Therefore, in this case (14) is invalid. We conclude this section with an example showing that, in general, a revised form of (14), with inverses replaced by generalized inverses, does not provide a solution for **RIEP**(n).

Example 5. Let

$$s_1 = 1, \quad s_2 = 2, \quad s_3 = 3,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

We have

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad R = L^T, \quad S = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}.$$

Let L^+ and R^+ be the Moore–Penrose inverses of L and R, respectively; see [1]. We have

$$A = L^+[S \circ (LR)]R^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.5 & 1.5 \\ 0 & 1.5 & 1.5 \end{bmatrix}.$$

Since $A\langle 2 \rangle$ does not have an eigenvalue 2, A is not a solution for **RIEP**(3). Note that we still have $L_n A R_n = S_n \circ (L_n R_n)$.

In this section we have characterized solvability of **RIEP**(n) over a general field F in terms of recursive conditions. We have also given a necessary and sufficient condition for unique solvability and an explicit formula for the unique solution. In the following sections we shall discuss the special cases of nonnegative matrices, Z-matrices, M-matrices, real symmetric matrices, positive semidefinite matrices, Stieltjes matrices, and inverse M-matrices.

3. Existence of nonnegative solutions. In this section we apply the results of section 2 to nonnegative solutions for **RIEP**(n) over the field $\mathbb{R}$ of real numbers. A matrix $A \in \mathbb{R}^{n,n}$ is said to be *nonnegative* [*positive*] if all elements of A are nonnegative [positive]. In this case we write $A \geq 0$ [$A > 0$]. For matrices $A, B \in \mathbb{R}^{n,n}$ we write $A \geq B$ if $A - B \geq 0$ and $A > B$ if $A - B > 0$.

In order to state our results we define a vector over $\mathbb{R}$ to be *unisign* if its nonzero components have the same sign.

THEOREM 6. *Let $n \geq 2$. There exists a nonnegative solution for* **RIEP**(n) *if and only if*

$$l_i \text{ or } r_i \text{ is a unisign nonzero vector} \implies s_i \geq 0, \quad i = 1, \ldots, n, \tag{15}$$

and there exists a nonnegative solution B for **RIEP**$(n\text{-}1)$ *satisfying*

$$\begin{cases} \frac{s_n \tilde{r}_n}{r_{n,n}} \geq \frac{B\tilde{r}_n}{r_{n,n}}, & r_{n,n} \neq 0, \\ \\ s_n \tilde{r}_n = B\tilde{r}_n, & r_{n,n} = 0, \end{cases} \tag{16}$$

$$\begin{cases} \frac{\tilde{l}_n^T s_n}{l_{n,n}} \geq \frac{\tilde{l}_n^T B}{l_{n,n}}, & l_{n,n} \neq 0, \\ \\ \tilde{l}_n^T s_n = \tilde{l}_n^T B, & l_{n,n} = 0, \end{cases} \tag{17}$$

and

$$l_{n,n} r_{n,n} \neq 0 \quad \implies \quad s_n \left(\frac{\tilde{l}_n^T \tilde{r}_n}{l_{n,n} r_{n,n}} - 1 \right) \leq \frac{\tilde{l}_n^T B \tilde{r}_n}{l_{n,n} r_{n,n}}. \tag{18}$$

There exists a positive solution for **RIEP**(n) *if and only if there exists a positive solution B for* **RIEP**$(n\text{-}1)$ *such that* (15)–(18) *hold with strict inequalities and every nonzero unisign vector l_i or r_i has no zero components.*

Proof. Let $A \in \mathbb{R}^{n,n}$. As in the proof of Theorem 2, partition A as in (3), and so A solves **RIEP**(n) if and only if B solves **RIEP**(n-1) and (4)–(7) hold. Therefore, if A is a nonnegative solution for **RIEP**(n), then we have (16)–(18). Also, it follows from the nonnegativity of A that (15) holds. Conversely, assume that (15) holds and that B forms a nonnegative solution for **RIEP**(n-1) satisfying (16)–(18). We show that in this case we can find nonnegative solutions x, y, and z for (4)–(7). Distinguish between four cases:

1. $r_{n,n} = 0$, $l_{n,n} \neq 0$. Here x is given by (8), y can be chosen arbitrarily, and z should be chosen such that (7) holds. It follows from (17) that x is nonnegative. If $s_n \geq 0$, then we choose $y = 0$, we have $z = s_n$, and so we have a nonnegative solution for (4)–(7). If $s_n < 0$, then, by (15), l_n is not unisign and hence $\frac{\tilde{l}_n^T}{l_{n,n}}$ has at least one negative component. It follows that we can find a positive vector y such that $\frac{\tilde{l}_n^T y}{l_{n,n}} < s_n$. Since by (7) we have $z = s_n - \frac{\tilde{l}_n^T y}{l_{n,n}}$, it follows that $z > 0$, and so again we have a nonnegative solution for (4)–(7).
2. $l_{n,n} = 0$, $r_{n,n} \neq 0$. Here y is given by (9), x can be chosen arbitrarily, and z should be chosen such that (6) holds. The proof follows as in the previous case.

3. $l_{n,n} = r_{n,n} = 0$. Here x and y should be chosen such that $x^T \tilde{r}_n = \tilde{l}_n^T y = 0$ and z can be chosen arbitrarily. In order to obtain a nonnegative solution we can choose x, y, and z to be zero.
4. $l_{n,n} \neq 0,\ r_{n,n} \neq 0$. Here x is given by (8), y is given by (9), and z is given by (10). It follows from (17), (16), and (18) that x, y, and z are nonnegative.

Assume now that A is a *positive* solution for **RIEP**(n). It is easy to verify that in this case (15)–(18) should hold with strict inequalities. Also, for every nonzero unisign vector l_i $[r_i]$, the vector $l_i^T A\langle i\rangle$ $[A\langle i\rangle r_i]$ has no zero components, implying that l_i, $[r_i]$ has no zero components. Conversely, assume that (15) holds with a strict inequality, that every nonzero unisign vector l_i or r_i has no zero components, and that B forms a positive solution for **RIEP**(n-1) satisfying (16)–(18) with strict inequalities. We show that in this case we can find positive solutions x, y, and z for (4)–(7). Note that in case 1 above, the vector x now becomes positive. Also, since the inequality in (15) is now strict, we have either $s_n > 0$, in which case we can choose *positive* y sufficiently small such that z is positive, or $s_n \leq 0$, in which case y can be chosen positive as before and the resulting z is positive. The same arguments hold for case 2. In case 4, it follows from the strict inequalities (16)–(18) that x, y, and z are positive. Finally, in case 3, since l_n and r_n both have at least one zero component, it follows that both vectors are not unisign. Hence, we can find positive x and y such that $x^T \tilde{r}_n = \tilde{l}_n^T y = 0$. We assign any positive number to z to find a positive solution A for **RIEP**(n). ∎

By the Perron–Frobenius theory (see, e.g., [9, 2]) the largest absolute value $\rho(A)$ of an eigenvalue of a nonnegative $n \times n$ matrix A is itself an eigenvalue of A, the so-called *Perron root* of A, and it has an associated nonnegative eigenvector. Furthermore, if A is *irreducible*, that is, if either $n = 1$ or $n \geq 2$ and there exists no permutation matrix P such that $P^T A P = \left[\begin{smallmatrix} B & C \\ 0 & D \end{smallmatrix}\right]$, where B and D are square, then $\rho(A)$ is a simple eigenvalue of A with an associated positive eigenvector. If A is not necessarily irreducible, then we have the following; see, e.g., [2].

THEOREM 7. *If B is a principal submatrix of a nonnegative square matrix A, then $\rho(B) \leq \rho(A)$. Furthermore, $\rho(A)$ is an eigenvalue of some proper principal submatrix of A if and only if A is reducible.*

Note that if we require that the s_i are the Perron roots of the principal submatrices $A\langle i\rangle$, $i = 1, \ldots, n$, then, by Theorem 7, we have

$$0 \leq s_1 \leq s_2 \leq \cdots \leq s_n. \tag{19}$$

If, furthermore, all the leading principal submatrices of A are required to be irreducible, then

$$0 \leq s_1 < s_2 < \cdots < s_n. \tag{20}$$

Condition (19) is not sufficient to guarantee that a nonnegative solution A for **RIEP**(n) necessarily has $s_1, \ldots, s_n$ as Perron roots of $A\langle i\rangle$, $i = 1, \ldots, n$, as is demonstrated by the following example.

Example 8. Let

$$s_1 = s_2 = 1, \quad s_3 = 2,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

The nonnegative matrix

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

solves **RIEP**(3). Note that $\rho(A) = 3 > s_3$.

In order to see cases in which $s_1, \ldots, s_n$ are the Perron roots of $A\langle i\rangle$, $i = 1, \ldots, n$, respectively, we prove the following.

PROPOSITION 9. *If the vector l_n or r_n is positive, then for a nonnegative solution A for* **RIEP**(n) *we have $\rho(A) = s_n$.*

Proof. The claim follows immediately from the known fact that a positive eigenvector of a nonnegative matrix corresponds to the spectral radius; see, e.g., Theorem 2.1.11 in [2, p. 28]. ∎

COROLLARY 10. *If for every $i \in \{1, \ldots, n\}$ we have either $l_i > 0$ or $r_i > 0$, then for every nonnegative solution A for* **RIEP**(n) *we have $\rho(A\langle i\rangle) = s_i$, $i = 1, \ldots, n$.*

LEMMA 11. *Assume that there exists a nonnegative solution A for* **RIEP**(n) *such that $\rho(A\langle n\text{-}1\rangle) < s_n$. If $r_n \neq 0$ or $l_n \neq 0$, then $\rho(A) = s_n$.*

Proof. Since $r_n \neq 0$ or $l_n \neq 0$ it follows that s_n is an eigenvalue of A. Assume that $s_n \neq \rho(A)$. It follows that the nonnegative matrix A has at least two eigenvalues larger than or equal to s_n. By [7, p. 473] (see also [11, Corollary 1]) it follows that $\rho(A\langle \text{n-}1\rangle) \geq s_n$, which is a contradiction. Therefore, we have $s_n = \rho(A)$. ∎

COROLLARY 12. *If for every $i \in \{1, \ldots, n\}$, we have either $r_i \neq 0$ or $l_i \neq 0$, and if* (20) *holds, then for every nonnegative solution A for* **RIEP**(n) *we have $\rho(A\langle i\rangle) = s_i$, $i = 1, \ldots, n$.*

Proof. Note that $A\langle 1\rangle = [s_1]$ and so $\rho(A\langle 1\rangle) = s_1$. Our result follows using Lemma 11 repeatedly. ∎

LEMMA 13. *Assume that $r_n \geq 0$ and $r_{n,n} \neq 0$ or that $l_n \geq 0$ and $l_{n,n} \neq 0$. Then for every nonnegative solution A for* **RIEP**(n) *we have $\rho(A) = \max\{\rho(A\langle n\text{-}1\rangle), s_n\}$.*

Proof. Without loss of generality, we consider the case where $r_n \geq 0$ and $r_{n,n} \neq 0$. If r_n is positive, then, by Proposition 9, we have $\rho(A) = s_n$ and, since by the Perron–Frobenius theory we have $\rho(A\langle \text{n-}1\rangle) \leq \rho(A)$, the result follows. Otherwise, r_n has some zero components. Let α be the set of indices i such that $r_{i,n} > 0$ and let α^c be the complement of α in $\{1, \ldots, n\}$. Note that since r_n is a nonnegative eigenvector of the nonnegative matrix A it follows that the submatrix $A[\alpha^c|\alpha]$ of A, with rows indexed by α^c and columns indexed by α, is a zero matrix. It follows that A is a reducible matrix and $\rho(A) = \max\{\rho(A[\alpha^c|\alpha^c]), \rho(A[\alpha|\alpha])\}$. Note that the subvector $r_n[\alpha]$ of r_n indexed by α is a positive eigenvector of $A[\alpha|\alpha]$ associated with the eigenvalue s_n. It thus follows that $\rho(A[\alpha|\alpha]) = s_n$. Since $n \in \alpha$ it follows that $A[\alpha^c|\alpha^c]$ is a submatrix of $A\langle \text{n-}1\rangle$. Thus, by the Perron–Frobenius theory we have $\rho(A[\alpha^c|\alpha^c]) \leq \rho(A\langle \text{n-}1\rangle) \leq \rho(A)$. Hence, it follows that $\rho(A) = \max\{s_{n-1}, s_n\}$. ∎

COROLLARY 14. *Assume that for every $i \in \{1, \ldots, n\}$ we have either $r_i \geq 0$ and $r_{i,i} \neq 0$ or $l_i \geq 0$ and $l_{i,i} \neq 0$. Then for every nonnegative solution A for* **RIEP**(n) *we have $\rho(A\langle i\rangle) = \max_{j=1,\ldots,i}\{s_j\}$.*

Proof. Note that $A\langle 1\rangle = [s_1]$ and so $\rho(A\langle 1\rangle) = s_1$. Our result follows using Lemma 13 repeatedly. ∎

COROLLARY 15. *Assume that for every $i \in \{1, \ldots, n\}$, we have either $r_i \geq 0$ and $r_{i,i} \neq 0$ or $l_i \geq 0$ and $l_{i,i} \neq 0$. If* (19) *holds, then for every nonnegative solution A we have $\rho(A\langle i\rangle) = s_i$, $i = 1, \ldots, n$.*

Another interesting consequence of Theorem 4 is the following relationship between the matrix elements and the eigenvectors associated with the Perron roots of the leading principal submatrices of a nonnegative matrix.

COROLLARY 16. *Let $n \geq 2$. Let $A \in \mathbb{R}^{n,n}$ be a nonnegative matrix, let s_i, l_i, and r_i be the Perron roots and associated left and right eigenvectors of $A\langle i\rangle$, $i = 1, \ldots, n$, respectively, and assume that* (20) *holds. Let S_n, L_n, R_n be defined as in* (11) *and* (12). *Then*

$$A\langle i\rangle = L_i^{-1}[S_i \circ (L_i R_i)]R_i^{-1}, \quad i = 1, \ldots, n. \tag{21}$$

Proof. Since (20) holds, it follows that s_i is not an eigenvalue of $A\langle$i-1$\rangle$, $i = 2, \ldots, n$. Therefore, it follows from (1) and (2) that $l_{i,i}r_{i,i} \neq 0$. Also, since l_1 and r_1 are eigenvectors of $A\langle 1\rangle$, we have $l_{1,1}r_{1,1} \neq 0$. It now follows from Theorem 4 that $A\langle i\rangle$ is the unique solution for **RIEP**(i), and is given by (21). ∎

While Theorem 6 provides a recursive characterization for nonnegative solvability of **RIEP**(n), in general nonrecursive necessary and sufficient conditions for the existence of nonnegative solution are not known. We now present a nonrecursive sufficient condition.

COROLLARY 17. *Assume that the vectors l_i, r_i, $i = 1, \ldots, n$, are all positive and that the numbers $s_1, \ldots, s_n$ are all positive. Let*

$$M_i^r = \max_{j=1,\ldots,i-1} \frac{r_{j,i}}{r_{j,i-1}}, \qquad m_i^r = \min_{j=1,\ldots,i-1} \frac{r_{j,i}}{r_{j,i-1}},$$

$$M_i^l = \max_{j=1,\ldots,i-1} \frac{l_{i,j}}{l_{i-1,j}}, \qquad m_i^l = \min_{j=1,\ldots,i-1} \frac{l_{i,j}}{l_{i-1,j}}.$$

If we have

$$s_i m_i^r \geq s_{i-1} M_i^r, \qquad i = 2, \ldots, n, \tag{22}$$

$$s_i m_i^l \geq s_{i-1} M_i^l, \qquad i = 2, \ldots, n, \tag{23}$$

and

$$s_i(\tilde{l}_i^T \tilde{r}_i - l_{i,i}r_{i,i}) \leq s_{i-1} \max\left\{m_i^r \tilde{l}_i^T r_{i-1}, m_i^l l_{i-1}^T \tilde{r}_i\right\}, \qquad i = 2, \ldots, n, \tag{24}$$

then there exists a (unique) nonnegative solution A for **RIEP**(n).

Furthermore, if all the inequalities (22)–(24) *hold with strict inequality, then there exists a (unique) positive solution A for* **RIEP**(n).

Proof. We prove our assertion by induction on n. The case $n = 1$ is trivial. By the inductive assumption we can find a nonnegative solution B for **RIEP**(n-1). Note that

$$M_n^r r_{n-1} \geq \tilde{r}_n \geq m_n^r r_{n-1}. \tag{25}$$

Therefore, it follows from (22) that

$$s_n \tilde{r}_n \geq s_n m_n^r r_{n-1} \geq s_{n-1} M_n^r r_{n-1} = M_n^r B r_{n-1} \geq B\tilde{r}_n,$$

and so (16) holds. Similarly, we prove that (17) holds. To prove that (18) holds note that by (25) we have $B\tilde{r}_n \geq B m_n^r r_{n-1} = s_{n-1} m_n^r r_{n-1}$. Similarly, we have

$\tilde{l}_n^T B \geq s_{n-1} m_n^l l_{n-1}^T$. Hence, it follows that $\tilde{l}_n^T B \tilde{r}_n \geq s_{n-1} \max\{m_n^r \tilde{l}_n^T r_{n-1}, m_n^l l_{n-1}^T \tilde{r}_n\}$. By applying (24) to $i = n$ we obtain (18). By Theorem 6, there exists a nonnegative solution for **RIEP**(n). The proof of the positive case is similar. □

The conditions in Corollary 17 are not necessary as is demonstrated by the following example.

Example 18. Let $s_1 = 1$, $s_2 = 2$, $s_3 = 3$ and let

$$r_1 = l_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad r_2 = l_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad r_3 = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}, \quad l_3 = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

We have $m_3^r = 3$, $M_3^r = 5$, $m_3^l = 1$, and $M_3^l = 2$. Note that both (22) and (23) do not hold for $i = 3$. Nevertheless, the unique solution for **RIEP**(3) is the nonnegative matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 7 \\ 1 & 0 & 0 \end{bmatrix}.$$

4. Uniqueness of nonnegative solutions. When considering uniqueness of nonnegative solutions for **RIEP**(n), observe that it is possible that **RIEP**(n) does not have a unique solution but does have a unique nonnegative solution, as is demonstrated by the following example.

Example 19. Let

$$s_1 = s_2 = 0,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

By Theorem 2, there is no unique solution for **RIEP**(2). Indeed, the solutions for **RIEP**(2) are all matrices of the form

$$\begin{bmatrix} 0 & 0 \\ a & -a \end{bmatrix}.$$

Clearly, the zero matrix is the only nonnegative solution for **RIEP**(2).

Observe that, unlike in Theorem 2, the existence of a unique nonnegative solution for **RIEP**(n) does not necessarily imply the existence of a unique nonnegative solution for **RIEP**(n-1), as is demonstrated by the following example.

Example 20. Let

$$s_1 = s_2 = 0, \quad s_3 = 2,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Observe that all matrices of the form

$$\begin{bmatrix} 0 & 0 \\ a & a \end{bmatrix}$$

solve **RIEP**(2), and hence there is no unique nonnegative solution for **RIEP**(2). However, the only nonnegative solution for **RIEP**(3) is the matrix

$$\begin{bmatrix} 0 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

We remark that one can easily produce a similar example with nonnegative vectors r_i and l_i, $i = 1, \ldots, n$.

In order to introduce necessary conditions and sufficient conditions for uniqueness of nonnegative solutions for **RIEP**(n), we prove the following.

LEMMA 21. *Let $n \geq 2$, and assume that B forms a nonnegative solution for* **RIEP**(n-1) *satisfying* (15)–(18). *Then there exist unique nonnegative vectors x, y, and z such that the matrix $\left[\begin{smallmatrix} B & y \\ x^T & z \end{smallmatrix}\right]$ solves* **RIEP**(n) *if and only if either $l_{n,n}r_{n,n} \neq 0$, or $s_n = 0$ and l_n is a unisign vector with no zero components, or $s_n = 0$ and r_n is a unisign vector with no zero components.*

Proof. We follow the proof of Theorem 6. Consider the four cases in that proof. In case 1, the vector x is uniquely determined and any nonnegative assignment for y is valid as long as $z = s_n - \frac{\tilde{l}_n^T y}{l_{n,n}} \geq 0$. If $s_n > 0$, then every nonnegative vector y sufficiently small will do. If $s_n < 0$, then, as is shown in the proof of Theorem 6, we can find a positive y such that $z > 0$, and by continuity arguments there exist infinitely many such vectors y. If $s_n = 0$, then a unique such y exists if and only if there exists a unique nonnegative vector y such that $\frac{\tilde{l}_n^T y}{l_{n,n}} \leq 0$. Clearly, if $\tilde{l}_n$ has a nonpositive component, then every vector y whose corresponding component is positive and all other components are zero solves the problem. On the other hand, if $\tilde{l}_n > 0$, which is equivalent to saying that l_n is a unisign vector with no zero components, then the only nonnegative vector y that solves the problem is $y = 0$. Similarly, we prove that, in case 2, a unique nonnegative solution exists if and only if $s_n = 0$ and r_n is a unisign vector with no zero components. We do not have uniqueness in case 3 since then z can be chosen arbitrarily. Finally, there is always uniqueness in case 4. ∎

Lemma 21 yields sufficient conditions and necessary conditions for uniqueness of nonnegative solutions for **RIEP**(n). First, observe that if $s_n = 0$ and l_n is a unisign vector with no zero components, or if $s_n = 0$ and r_n is a unisign vector with no zero components, then the zero matrix is the only nonnegative solution of the problem. A less trivial sufficient condition is the following.

COROLLARY 22. *Let $n \geq 2$, and let A be a nonnegative solution for* **RIEP**(n). *If $A\langle n\text{-}1\rangle$ forms a unique nonnegative solution for* **RIEP**(n-1) *and if $l_{n,n}r_{n,n} \neq 0$, then A is the unique nonnegative solution for* **RIEP**(n).

Necessary conditions are given by the following corollary.

COROLLARY 23. *Let $n \geq 2$. If there exists a unique nonnegative solution for* **RIEP**(n), *then either $l_{n,n}r_{n,n} \neq 0$, or $s_n = 0$ and l_n is a unisign vector with no zero components, or $s_n = 0$ and r_n is a unisign vector with no zero components.*

The condition $l_{n,n}r_{n,n} \neq 0$ is not sufficient for the uniqueness of a nonnegative solution for **RIEP**(n), as is shown in the following example.

Example 24. Let

$$s_1 = s_2 = s_3 = 0,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}.$$

Although we have $l_{n,n}r_{n,n} \neq 0$, all matrices of the form

$$\begin{bmatrix} 0 & 0 & 0 \\ a & a & 0 \\ a & a & 0 \end{bmatrix}$$

solve **RIEP**(3), and hence there is no unique nonnegative solution for **RIEP**(3).

5. The *Z*-matrix and *M*-matrix case. A real square matrix A is said to be a *Z-matrix* if it has nonpositive off-diagonal elements. Note that A can be written as $A = \alpha I - B$, where α is a real number and B is a nonnegative matrix. If we further have that $\alpha \geq \rho(B)$, then we say that A is an *M-matrix*.

In this section we discuss Z-matrix and M-matrix solutions for **RIEP**(n) over the field $\mathbb{R}$ of real numbers. The proofs of the results are very similar to the proofs of the corresponding results in sections 3 and 4 and, thus, are omitted in most cases.

THEOREM 25. *Let $n \geq 2$. There exists a Z-matrix solution for* **RIEP**(n) *if and only if there exists a Z-matrix solution B for* **RIEP**$(n\text{-}1)$ *satisfying*

$$\begin{cases} \frac{s_n \tilde{r}_n}{r_{n,n}} \leq \frac{B\tilde{r}_n}{r_{n,n}}, & r_{n,n} \neq 0, \\ \\ s_n \tilde{r}_n = B\tilde{r}_n, & r_{n,n} = 0, \end{cases}$$

and

$$\begin{cases} \frac{\tilde{l}_n^T s_n}{l_{n,n}} \leq \frac{\tilde{l}_n^T B}{l_{n,n}}, & l_{n,n} \neq 0, \\ \\ \tilde{l}_n^T s_n = \tilde{l}_n^T B, & l_{n,n} = 0. \end{cases}$$

Furthermore, if l_n or r_n is positive, then a Z-matrix solution for **RIEP**(n) *is an M-matrix if and only if $s_n \geq 0$.*

Proof. The proof of the first part of the theorem is similar to the proof of Theorem 6, observing that here the vectors x and y are required to be nonnegative and that the sign of z is immaterial. The proof of the second part of the theorem follows, similarly to Proposition 9, from the known fact that a positive eigenvector of a Z-matrix corresponds to the least real eigenvalue. ∎

THEOREM 26. *Let $n \geq 2$. Let $A \in \mathbb{R}^{n,n}$ be a Z-matrix, let s_i, l_i, and r_i be the least real eigenvalues and the corresponding left and right eigenvectors of $A\langle i \rangle$, $i = 1, \ldots, n$, respectively, and assume that*

$$s_1 > s_2 > \cdots > s_n.$$

Let S_n, L_n, R_n be defined as in (11) *and* (12). *Then*

$$A\langle i \rangle = L_i^{-1}[S_i \circ (L_i R_i)]R_i^{-1}, \quad i = 1, \ldots, n.$$

For the numbers M_i^r, m_i^r, M_i^l, and m_i^l, defined in Corollary 17, we have the following.

THEOREM 27. *Assume that the vectors l_i, r_i, $i = 1, \ldots, n$, are all positive and that the numbers $s_1, \ldots, s_n$ are all positive. If we have*

$$s_i M_i^r \leq s_{i-1} m_i^r, \qquad i = 2, \ldots, n,$$

and

$$s_i M_i^l \leq s_{i-1} m_i^l, \qquad i = 2, \ldots, n,$$

then there exists a (unique) M-matrix solution A for **RIEP**(n).

THEOREM 28. *Let $n \geq 2$, let A be a Z-matrix solution for* **RIEP**(n)*, and assume that $A\langle n\text{-}1\rangle$ forms a unique Z-matrix solution for* **RIEP**$(n\text{-}1)$*. Then A is the unique Z-matrix solution for* **RIEP**(n) *if and only if $l_{n,n} r_{n,n} \neq 0$.*

Here too, unlike in Theorem 2, the existence of a unique Z-matrix solution for **RIEP**(n) does not necessarily imply the existence of a unique Z-matrix solution for **RIEP**(n-1), as is demonstrated by the following example.

Example 29. Let $s_1 = s_2 = s_3 = 0$, and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Observe that all matrices of the form

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

solve **RIEP**(2), and hence there is no unique Z-matrix solution for **RIEP**(2). However, it is easy to verify that the zero matrix is the only Z-matrix solution for **RIEP**(3).

6. The real symmetric case. The inverse eigenvalue problem for real symmetric matrices is well studied; see, e.g., [3]. In this section we consider symmetric solutions for **RIEP**(n) over the field $\mathbb{R}$ of real numbers. We obtain the following consequence of Theorem 2, characterizing the real symmetric case.

THEOREM 30. *Let $n \geq 2$. There exists a symmetric solution for* **RIEP**(n) *if and only if there exists a symmetric solution B for* **RIEP**$(n\text{-}1)$ *such that the implications* (1) *and* (2) *hold, and*

$$l_{n,n} r_{n,n} \neq 0 \quad \Longrightarrow \quad (s_n I_{n-1} - B) \left(\frac{\tilde{l}_n}{l_{n,n}} - \frac{\tilde{r}_n}{r_{n,n}} \right) = 0. \tag{26}$$

Furthermore, if there exists a unique symmetric solution for **RIEP**(n)*, then $l_{n,n} \neq 0$ or $r_{n,n} \neq 0$.*

Proof. Let $A \in \mathbb{R}^{n,n}$. Partition A as in (3), and so A solves **RIEP**(n) if and only if B solves **RIEP**(n-1) and (4)–(7) hold. It was shown in the proof of Theorem 2 that (4)–(7) are solvable if and only if (1) and (2) hold. Therefore, all we have to show is that if B is symmetric, then we can find solutions x, y, and z for (4)–(7) such that $y = x$ if and only if (26) holds. We go along the four cases discussed in Theorem 2. In case 1, the vector x is uniquely determined and the vector y can be chosen arbitrarily. Therefore, in this case we set $y = x$, and z is then uniquely determined. In case 2, the vector y is uniquely determined and the vector x can be chosen arbitrarily. Thus, in this case we set $x = y$, and z is then uniquely determined. In case 3, we can choose

any x and y as long as $x^T\tilde{r}_n = 0$ and $\tilde{l}_n^T y = 0$. In particular, we can choose $x = y = 0$. Furthermore, z can be chosen arbitrarily. Finally, in case 4, we have $x = y$ if and only if (26) holds. Note that this is the only case in which, under the requirement that $y = x$, the vectors x, y, and z are uniquely determined. ∎

We remark that, unlike in Theorem 2, the existence of a unique symmetric solution for **RIEP**(n) does not necessarily imply the existence of a unique symmetric solution for **RIEP**(n-1), as is demonstrated by the following example.

Example 31. Let

$$s_1 = 1, \quad s_2 = 2, \quad s_3 = 0,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix},$$

$$l_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad r_4 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}.$$

It is easy to verify that all symmetric matrices of the form

$$\begin{bmatrix} 1 & 1 & a \\ 1 & 1 & a \\ a & a & b \end{bmatrix}, \quad a, b \in \mathbb{R},$$

solve **RIEP**(3), while the unique solution for **RIEP**(4) is

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

This example also shows that there may exist a unique solution for **RIEP**(n) even if $l_{i,i} = r_{i,i} = 0$ for some $i \in 1, \ldots, n$.

Naturally, although not necessarily, one may expect in the symmetric case to have the condition

$$r_i = l_i, \quad i = 1, \ldots, n. \tag{27}$$

Indeed, in this case we have the following corollary of Theorems 2 and 30.

COROLLARY 32. *Let $n \geq 2$ and assume that* (27) *holds. The following are equivalent:*

(i) *There exists a symmetric solution for* **RIEP**(n).

(ii) *There exists a solution for* **RIEP**(n).

(iii) *There exists a symmetric solution B for* **RIEP**$(n$-1$)$ *such that* (1) *holds.*

(iv) *There exists a solution B for* **RIEP**$(n$-1$)$ *such that* (1) *holds.*

Proof. Note that since (27) holds, we always have (26). We now prove the equivalence between the four statements of the theorem.

(i) $\Longrightarrow$ (ii) is trivial.

(ii) $\Longrightarrow$ (iv) by Theorem 2.

(iv) $\Longrightarrow$ (iii). Since (27) holds, it follows that $\frac{B+B^T}{2}$ also solves **RIEP**(n-1).

(iii) $\Longrightarrow$ (i). Since B is symmetric and since we have (27), the implications (1) and (2) are identical. Our claim now follows by Theorem 30. ∎

For uniqueness we have the following.

THEOREM 33. *Let $n \geq 2$ and assume that* (27) *holds. The following are equivalent:*

(i) *There exists a unique symmetric solution for* **RIEP**(n).

(ii) *There exists a unique solution for* **RIEP**(n).

(iii) *We have $l_{i,i} \neq 0,\ i = 1, \ldots, n$.*

Proof. In view of (27), the equivalence of (ii) and (iii) follows from Theorem 4. To see that (i) and (iii) are equivalent note that, by the construction in Theorem 30, for every symmetric solution B for **RIEP**(n-1) there exists a solution A for **RIEP**(n) such that $A\langle \text{n-1} \rangle = B$. Furthermore, A is uniquely determined if and only if $l_{n,n} \neq 0$. Therefore, it follows that there exists a unique symmetric solution for **RIEP**(n) if and only if there exists a unique symmetric solution for **RIEP**(n-1) and $l_{n,n} \neq 0$. Our assertion now follows by induction on n. ∎

We conclude this section remarking that a similar discussion can be carried over for complex Hermitian matrices.

7. The positive semidefinite case. In view of the discussion of the previous section, it would be interesting to find conditions for the existence of a *positive (semi)definite* real symmetric solution for **RIEP**(n). Clearly, a necessary condition is nonnegativity of the numbers s_i whenever $r_i \neq 0$ or $l_i \neq 0$, $i = 1, \ldots, n$. Nevertheless, this condition is not sufficient even if a real symmetric solution exists, as is demonstrated by the following example.

Example 34. Let

$$s_1 = 1, \quad s_2 = 3, \quad s_3 = 5,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The unique solution for **RIEP**(3) is the symmetric matrix

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix},$$

which is not positive semidefinite.

The following necessary and sufficient condition follows immediately from Theorem 4.

THEOREM 35. *Let $n \geq 2$ and assume that* (27) *holds. Assume, further, that $r_{i,i} \neq 0,\ i = 1, \ldots, n$. Then the unique solution for* **RIEP**(n) *is positive semidefinite [positive definite] if and only if $S_n \circ (R_n^T R_n)$ is positive semidefinite [positive definite].*

Remark 36. By Theorem 33, in the case that $r_{i,i} = 0$ for some i we do not have uniqueness of symmetric solutions for **RIEP**(n). Hence, if there exists a symmetric solution for **RIEP**(n), then there exist at least two different such solutions A and B.

Note that $A + c(B - A)$ also forms a symmetric solution for **RIEP**(n) for every real number c. It thus follows that in this case it is impossible to have *all* solutions for **RIEP**(n) positive semidefinite. Therefore, in this case we are looking for conditions for the existence of *some* positive semidefinite solution for **RIEP**(n).

The following necessary condition follows immediately from Proposition 3.

THEOREM 37. *Let $n \geq 2$ and assume that* (27) *holds. If there exists a positive semidefinite real symmetric solution for* **RIEP**(n)*, then $S_n \circ (R_n^T R_n)$ is positive semidefinite.*

In order to find sufficient conditions for the existence of a positive semidefinite solution for **RIEP**(n), we denote by $\sigma(A)$ the least eigenvalue of a real symmetric matrix A.

LEMMA 38. *Let $n \geq 2$ and assume that* (27) *holds. Assume that there exists a symmetric solution A for* **RIEP**(n) *such that $\sigma(A\langle n\text{-}1\rangle) > s_n$. If $r_n \neq 0$, then $\sigma(A) = s_n$.*

Proof. Since $r_n \neq 0$ it follows that s_n is an eigenvalue of A. Assume that $\sigma(A) \neq s_n$. It follows that A has at least two eigenvalues smaller than or equal to s_n. By the Cauchy interlacing theorem for Hermitian matrices (see, e.g., [9, Theorem 4.3.8, p. 185]), it follows that $\sigma(A\langle \text{n-}1\rangle) \leq s_n$, which is a contradiction. Therefore, we have $\sigma(A) = s_n$. ∎

COROLLARY 39. *Let $n \geq 2$ and assume that* (27) *holds. If $r_i \neq 0$ for all i, $i = 1, \ldots, n$, and if $s_1 > s_2 > \cdots > s_n \geq 0$, then every real symmetric solution A for* **RIEP**(n) *is positive semidefinite. If $s_n > 0$, then every real symmetric solution for* **RIEP**(n) *is positive definite.*

Proof. Note that $A\langle 1\rangle = [s_1]$ and so $\sigma(A\langle 1\rangle) = s_1$. Using Lemma 38 repeatedly we finally obtain $\sigma(A) = s_n$, implying our claim. ∎

Remark 40. In view of Remark 36, it follows from Corollary 39 that if $r_i \neq 0$ for all i and if $s_1 > s_2 > \cdots > s_n \geq 0$, then $r_{i,i} \neq 0$, $i = 1, \ldots, n$, and so **RIEP**(n) has a unique (positive semidefinite) solution.

The converse of Corollary 39 is, in general, not true. That is, even if every real symmetric solution for **RIEP**(n) is positive semidefinite, we do not necessarily have $s_1 > s_2 > \cdots > s_n \geq 0$, as is demonstrated by the following example.

Example 41. Let

$$s_1 = 2, \quad s_2 = 3,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The unique solution for **RIEP**(2) is the positive definite matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Nevertheless, we do not have $s_1 \geq s_2$.

We conclude this section with a conjecture motivated by Theorems 35 and 37. One direction of the conjecture is proven by Theorem 37.

CONJECTURE 42. *Let $n \geq 2$, let* (27) *hold, and assume that a solution for* **RIEP**(n) *exists. Then there exists a positive semidefinite [positive definite] real symmetric solution for* **RIEP**(n) *if and only if $S_n \circ (R_n^T R_n)$ is positive semidefinite [positive definite].*

In Conjecture 42, the requirement that a solution for **RIEP**(n) exists is necessary, as is demonstrated by the following example.

Example 43. Let

$$s_1 = 2, \quad s_2 = 1,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The unique solution for **RIEP**(1) is the matrix $B = \begin{bmatrix} 2 \end{bmatrix}$, and so by Theorem 2 there exists no solution for **RIEP**(2). Nevertheless, the matrix $S_2 \circ (R_2^T R_2)$ is the positive semidefinite matrix

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

8. The Stieltjes matrix case. In this section we combine the results of the previous two sections to obtain analogous results for *Stieltjes matrices*, that is, symmetric M-matrices.

The following theorem follows immediately from Theorems 30 and 25.

THEOREM 44. *Let $n \geq 2$. There exists a symmetric Z-matrix solution for* **RIEP**(n) *if and only if there exists a symmetric Z-matrix solution B for* **RIEP**$(n$-1) *satisfying*

$$\begin{cases} \frac{s_n \tilde{r}_n}{r_{n,n}} \leq \frac{B\tilde{r}_n}{r_{n,n}}, & r_{n,n} \neq 0, \\ s_n \tilde{r}_n = B\tilde{r}_n, & r_{n,n} = 0, \end{cases}$$

$$\begin{cases} \frac{\tilde{l}_n^T s_n}{l_{n,n}} \leq \frac{\tilde{l}_n^T B}{l_{n,n}}, & l_{n,n} \neq 0, \\ \tilde{l}_n^T s_n = \tilde{l}_n^T B, & l_{n,n} = 0, \end{cases}$$

and

$$l_{n,n} r_{n,n} \neq 0 \quad \Longrightarrow \quad (s_n I_{n-1} - B) \left(\frac{\tilde{l}_n}{l_{n,n}} - \frac{\tilde{r}_n}{r_{n,n}} \right) = 0.$$

Furthermore, if l_n or r_n is positive, then a symmetric Z-matrix solution for **RIEP**(n) *is a Stieltjes matrix if and only if $s_n \geq 0$.*

COROLLARY 45. *Let $n \geq 2$, and assume that the vectors l_i, $i = 1, \ldots, n$, are all positive and that* (27) *holds. There exists a symmetric Z-matrix solution A for* **RIEP**(n) *if and only if there exists a symmetric Z-matrix solution B for* **RIEP**$(n$-1) *satisfying $s_n \tilde{r}_n \leq B\tilde{r}_n$. The solution A is a Stieltjes matrix if and only if $s_n \geq 0$.*

The following nonrecursive sufficient condition follows from Theorem 27.

THEOREM 46. *Let $n \geq 2$, and assume that the vectors l_i, $i = 1, \ldots, n$, are all positive, that* (27) *holds, and that the numbers $s_1, \ldots, s_n$ are all positive. If we have*

$$s_i M_i^r \leq s_{i-1} m_i^r, \qquad i = 2, \ldots, n,$$

then there exists a (unique) Stieltjes matrix solution A for **RIEP**(n).

Proof. By Theorem 27 there exists a unique M-matrix solution A for **RIEP**(n). Since A^T also solves the problem, it follows that $A = A^T$ and the result follows. ∎

9. The inverse M-matrix case. It is well known that for a nonsingular M-matrix A we have $A^{-1} \geq 0$. Accordingly, a nonnegative matrix A is called an *inverse M-matrix* if it is invertible and A^{-1} is an M-matrix. An overview of characterizations of nonnegative matrices that are inverse M-matrices can be found in [10]. In this section we discuss, as a final special case, inverse M-matrix solutions for **RIEP**(n).

The following theorem follows immediately from two results of [10].

THEOREM 47. *Let $A \in \mathbb{R}^{n,n}$ be partitioned as in* (3). *Then A is an inverse M-matrix if and only if B is an inverse M-matrix and*

$$v = B^{-1}y \geq 0, \tag{28}$$

$$u^T = x^T B^{-1} \geq 0, \tag{29}$$

$$s = z - u^T B v > 0, \tag{30}$$

and

$$vu^T \leq -sB^{-1}, \quad \text{except for the diagonal entries.} \tag{31}$$

Proof. By Corollary 3 in [10], if A is an inverse M-matrix, then B is an inverse M-matrix. By Theorem 8 in [10], if B is an inverse M-matrix, then A is an inverse M-matrix if and only if (28)–(31) hold. Our claim follows. ∎

The next result gives necessary and sufficient recursive conditions for the existence of an inverse M-matrix solution for **RIEP**(n).

THEOREM 48. *Let $n \geq 2$. There exists an inverse M-matrix solution for* **RIEP**(n) *if and only if $s_n > 0$ and there exists an inverse M-matrix solution B for* **RIEP**$(n\text{-}1)$ *satisfying*

$$\begin{cases} \frac{N\tilde{r}_n}{r_{n,n}} \geq 0, & r_{n,n} \neq 0, \\ N\tilde{r}_n = 0, & r_{n,n} = 0, \end{cases} \tag{32}$$

$$\begin{cases} \frac{\tilde{l}_n^T N}{l_{n,n}} \geq 0, & l_{n,n} \neq 0, \\ \tilde{l}_n^T N = 0, & l_{n,n} = 0, \end{cases} \tag{33}$$

$$l_{n,n}r_{n,n} \neq 0 \quad \Longrightarrow \quad \frac{\tilde{l}_n^T N \tilde{r}_n}{l_{n,n}r_{n,n}} < 1, \tag{34}$$

and, except for the diagonal entries,

$$l_{n,n}r_{n,n} \neq 0 \quad \Longrightarrow \quad s_n \left(\frac{\tilde{l}_n^T N \tilde{r}_n}{l_{n,n}r_{n,n}} - 1 \right) B^{-1} \geq \frac{N\tilde{r}_n \tilde{l}_n^T N}{l_{n,n}r_{n,n}}, \tag{35}$$

where $N = s_n B^{-1} - I_{n-1}$.

Proof. As in the proof of Theorem 2, partition A as in (3). If A is an inverse M-matrix solution for **RIEP**(n), then, as is well known, its eigenvalues lie in the open right half plane, and so the real eigenvalue s_n must be positive. Furthermore, by Theorem 47, B is an inverse M-matrix and (28)–(31) hold. Finally, we have (4)–(7). Distinguish between four cases:

1. $r_{n,n} = 0$, $l_{n,n} \neq 0$. Here x is given by (8), and so it follows from (29) that $\frac{\tilde{l}_n^T N}{l_{n,n}} \geq 0$. By Theorem 2 we have $B\tilde{r}_n = s_n\tilde{r}_n$, implying that $N\tilde{r}_n = 0$.
2. $l_{n,n} = 0$, $r_{n,n} \neq 0$. Here y is given by (9), and so it follows from (28) that $\frac{N\tilde{r}_n}{r_{n,n}} \geq 0$. By Theorem 2 we have $\tilde{l}_n^T N = 0$.
3. $l_{n,n} = r_{n,n} = 0$. Similarly to the previous cases, prove that $N\tilde{r}_n = 0$ and $\tilde{l}_n^T N = 0$.
4. $l_{n,n} \neq 0$, $r_{n,n} \neq 0$. Here x is given by (8), y is given by (9), and z is given by (10). It follows from (28) that $\frac{N\tilde{r}_n}{r_{n,n}} \geq 0$, and from (29) that $\frac{\tilde{l}_n^T N}{l_{n,n}} \geq 0$. It follows from (30) that

$$s = z - u^T B v$$

$$= s_n - \frac{\tilde{l}_n^T(s_n I_{n-1} - B)\tilde{r}_n}{l_{n,n}r_{n,n}} - \frac{\tilde{l}_n^T(s_n I_{n-1} - B)}{l_{n,n}} B^{-1} B B^{-1} \frac{(s_n I_{n-1} - B)\tilde{r}_n}{r_{n,n}}$$

$$= s_n \left(1 - \frac{\tilde{l}_n^T N \tilde{r}_n}{l_{n,n}r_{n,n}}\right) > 0.$$

Since $s_n > 0$, it now follows that $\frac{\tilde{l}_n^T N\tilde{r}_n}{l_{n,n}r_{n,n}} < 1$. Finally, it follows from (31) that, except for the diagonal entries,

$$\frac{N\tilde{r}_n\tilde{l}_n^T N}{l_{n,n}r_{n,n}} = B^{-1}\frac{(s_n I_{n-1} - B)\tilde{r}_n}{r_{n,n}} \frac{\tilde{l}_n^T(s_n I_{n-1} - B)}{l_{n,n}} B^{-1} = vu^T$$
$$\leq -sB^{-1} = s_n\left(\frac{\tilde{l}_n^T N\tilde{r}_n}{l_{n,n}r_{n,n}} - 1\right) B^{-1}.$$

We have thus proven that if A is an inverse M-matrix solution for **RIEP**(n), then $s_n > 0$ and B is an inverse M-matrix solution B for **RIEP**(n-1) satisfying (32)–(35).

Conversely, assume that $s_n > 0$ and B is an inverse M-matrix solution B for **RIEP**(n-1) satisfying (32)–(35). We show that x, y, and z can be chosen such that (28)–(31) hold, and so by Theorem 47, A is an inverse M-matrix. Here too we distinguish between four cases:

1. $r_{n,n} = 0$, $l_{n,n} \neq 0$. Here x is given by (8), and by (33) we obtain (29). Note that y can be chosen arbitrarily, and z should be chosen such that (7) holds. If we choose $y = 0$, then we obtain (28) and $z = s_n$. It follows that $z - u^T B v = s_n > 0$, and so we also have (30). Finally, since $v = 0$, since $s > 0$, and since B^{-1} is an M-matrix, it follows that (31) holds (except for the diagonal entries).

2. $l_{n,n} = 0,\ r_{n,n} \neq 0$. Here y is given by (9), and by (32) we obtain (28). The vector x can be chosen arbitrarily, so we choose $x = 0$. The proof follows as in the previous case.
3. $l_{n,n} = r_{n,n} = 0$. Here x and y should be chosen such that $x^T \tilde{r}_n = \tilde{l}_n^T y = 0$ and z can be chosen arbitrarily. We choose $x = y = 0$ and the proof follows.
4. $l_{n,n} \neq 0,\ r_{n,n} \neq 0$. Here x is given by (8), y is given by (9), and z is given by (10). By (32) and (33) we obtain (28) and (29), respectively. Finally, similarly to the corresponding case in the proof of the other direction, (34) implies (30) and (35) implies (31). ∎

Note that Conditions (32)–(33) immediately imply Conditions (16)–(17) by multiplying the inequality by the nonnegative matrix B. This is not surprising, since an inverse M-matrix is a nonnegative matrix. The converse, however, does not hold in general. The following example shows that although (16)–(17) is satisfied, (32)–(33) do not hold.

Example 49. Let

$$s_1 = 2, \quad s_2 = 5.2361, \quad s_3 = 21.2552,$$

and let

$$l_1 = r_1 = \begin{bmatrix} 1 \end{bmatrix}, \quad l_2 = r_2 = \begin{bmatrix} 0.5257 \\ 0.8507 \end{bmatrix}, \quad l_3 = r_3 = \begin{bmatrix} 0.1349 \\ 0.3859 \\ 0.9126 \end{bmatrix}.$$

The unique solution for **RIEP**(3) is the nonnegative matrix

$$A = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 4 & 7 \\ 2 & 7 & 18 \end{bmatrix},$$

which is not an inverse M-matrix since

$$A^{-1} = \begin{bmatrix} 1.6429 & -1.5714 & 0.4286 \\ -1.5714 & 2.2857 & -0.7143 \\ 0.4286 & -0.7143 & 0.2857 \end{bmatrix}.$$

Indeed, the unique nonnegative solution $B = \left[\begin{smallmatrix} 2 & 2 \\ 2 & 4 \end{smallmatrix}\right]$ for **RIEP**(2) satisfies (16), as

$$s_3 \tilde{r}_3 = \begin{bmatrix} 2.8673 \\ 8.2024 \end{bmatrix} \geq \begin{bmatrix} 1.0416 \\ 1.8134 \end{bmatrix} = B\tilde{r}_3.$$

However, B does not satisfy (32), since the vector

$$N\tilde{r}_3 = (s_3 B^{-1} - I_2)\tilde{r}_3 = \begin{bmatrix} -1.3688 \\ 2.2816 \end{bmatrix}$$

is not nonnegative.

10. Summary. We have given a detailed analysis of the recursive inverse eigenvalue problem, providing recursive and nonrecursive existence and uniqueness results for general matrices as well as specific classes of matrices. We summarize the results in Table 1.

TABLE 1
Table of results.

Result	Class of matrices	Existence	Uniqueness	Recurs./nonrec.
Theorem 2	$F^{n,n}$	$\surd$	$\surd$	recursive
Theorem 4	$F^{n,n}$	$\surd$	$\surd$	nonrecursive
Theorem 6	nonneg. matrices	$\surd$		recursive
Theorem 6	pos. matrices	$\surd$		recursive
Corollary 17	nonneg. matrices	$\surd$	$\surd$	recursive
Corollary 17	pos. matrices	$\surd$	$\surd$	recursive
Corollary 22	nonneg. matrices		$\surd$	recursive
Corollary 23	nonneg. matrices		$\surd$	recursive
Theorem 25	Z-matrices	$\surd$		recursive
Theorem 25	M-matrices	$\surd$		recursive
Theorem 26	Z-matrices	$\surd$		nonrecursive
Theorem 27	M-matrices	$\surd$	$\surd$	recursive
Theorem 28	M-matrices		$\surd$	recursive
Theorem 30	real symm. matrices	$\surd$		recursive
Corollary 32	real symm. matrices	$\surd$		recursive and nonrecursive
Theorem 33	real symm. matrices	$\surd$	$\surd$	nonrecursive
Theorem 35	pos. semidef. matrices	$\surd$	$\surd$	nonrecursive
Corollary 39	pos. semidef. matrices	$\surd$		nonrecursive
Theorem 43	Stieltjes matrices	$\surd$	$\surd$	recursive
Corollary 44	Stieltjes matrices	$\surd$		recursive
Theorem 45	Stieltjes matrices	$\surd$	$\surd$	recursive
Theorem 47	inverse M-matrices	$\surd$		recursive

REFERENCES

[1] A. BEN-ISRAEL AND T.N.E. GREVILLE, *Generalized Matrix Inverses: Theory and Applications*, John Wiley, New York, 1974.

[2] A. BERMAN AND R.J. PLEMMONS, *Nonnegative Matrices in Mathematical Sciences*, Classics in Applied Mathematics 9, SIAM, Philadelphia, PA, 1994.

[3] D. BOLEY AND G.H. GOLUB, *A survey of matrix inverse eigenvalue problems*, Inverse Problems, 3 (1987), pp. 595–622.

[4] M. BOYLE AND D. HANDELMAN, *The spectra of non-negative matrices via symbolic dynamics*, Ann. of Math. (2), 133 (1991), pp. 249–316.

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

[6] S. FRIEDLAND, *On an inverse problem for nonnegative and eventually nonnegative matrices*, Israel J. Math., 29 (1978), pp. 43–60.

[7] G. FROBENIUS, *Über Matrizen aus positiven Elementen*, S.–B. Preuss. Akad. Wiss., 1909, pp. 471–476.

[8] H. HOCHSTADT, *On some inverse problems in matrix theory*, Arch. Math. (Basel), 18 (1967), pp. 201–207.

[9] R.A. HORN AND C.R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[10] C.R. JOHNSON, *Inverse M-matrices*, Linear Algebra Appl., 47 (1982), pp. 195–216.

[11] D.M. KOTELJANSKIĬ, *On some properties of matrices with positive elements*, Trans. Amer. Math. Soc. Ser. 2, 27 (1963), pp. 9–18.

[12] T.J. LAFFEY, *Inverse eigenvalue problems for matrices*, Proc. Roy. Irish Acad. Sect. A, 95 (1995), pp. 81–88.

[13] R. LOEWY AND D. LONDON, *A note on an inverse problem for nonnegative matrices*, Linear and Multilinear Algebra, 6 (1978), pp. 83–90.

[14] G.N. DE OLIVEIRA, *Note on an inverse characteristic value problem*, Numer. Math., 15 (1970), pp. 345–347.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 413–429

METHODS FOR LARGE SCALE TOTAL LEAST SQUARES PROBLEMS*

ÅKE BJÖRCK†, P. HEGGERNES‡, AND P. MATSTOMS†

Abstract. The solution of the total least squares (TLS) problems, $\min_{E,f} \|(E,f)\|_F$ subject to $(A+E)x = b+f$, can in the generic case be obtained from the right singular vector corresponding to the smallest singular value σ_{n+1} of $(A,\ b)$. When A is large and sparse (or structured) a method based on Rayleigh quotient iteration (RQI) has been suggested by Björck. In this method the problem is reduced to the solution of a sequence of symmetric, positive definite linear systems of the form $(A^TA - \bar{\sigma}^2 I)z = g$, where $\bar{\sigma}$ is an approximation to σ_{n+1}. These linear systems are then solved by a *preconditioned* conjugate gradient method (PCGTLS). For TLS problems where A is large and sparse a (possibly incomplete) Cholesky factor of A^TA can usually be computed, and this provides a very efficient preconditioner. The resulting method can be used to solve a much wider range of problems than it is possible to solve by using Lanczos-type algorithms directly for the singular value problem. In this paper the RQI-PCGTLS method is further developed, and the choice of initial approximation and termination criteria are discussed. Numerical results confirm that the given algorithm achieves rapid convergence and good accuracy.

Key words. total least squares, Rayleigh quotient iteration, conjugate gradient method, singular values

AMS subject classification. 65F20

PII. S0895479899355414

1. Introduction.

The estimation of parameters in linear models is a fundamental problem in many scientific and engineering applications. A statistical model that is often realistic is to assume that the parameters x to be determined satisfy a linear relation

$$(A+E)x = b+f, \tag{1.1}$$

where $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$ are known and $(E,\ f)$ is an error matrix with rows which are independently and identically distributed with zero mean and the same variance. (To satisfy this assumption the data $(A,\ b)$ may need to be premultiplied by appropriate scaling matrices; see Golub and Van Loan [10].) In statistics this model is known as the "errors-in-variables model."

The estimate of the true but unknown parameter vector x in the model (1.1) is obtained from the solution of the total least squares (TLS) problem

$$\min_{E,f} \|(E,\ f)\|_F \ \text{ subject to } \ (A+E)x = b+f, \tag{1.2}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. If a minimizing pair $(E,\ f)$ has been found for the problem (1.2), then any x satisfying $(A+E)x = b+f$ is said to solve the TLS problem.

*Received by the editors April 28, 1999; accepted for publication (in revised form) by E. Ng March 21, 2000; published electronically July 11, 2000.
http://www.siam.org/journals/simax/22-2/35541.html
†Department of Mathematics, Linköping University, S-581 83, Linköping, Sweden (akbjo@math.liu.se, pomat@math.liu.se). The work of P. Matstoms was supported by the Swedish Research Council for Engineering Sciences, TFR.
‡Department of Informatics, University of Bergen, NO-5020 Bergen, Norway (pinar@ii.uib.no).

Due to recent advances in data collection techniques least squares (LS) or TLS problems where A is large and sparse (or structured) frequently arise, e.g., in signal and image processing applications. For the solution of the LS problem both direct methods based on sparse matrix factorizations and iterative methods are well developed; see [3].

An excellent treatment of theoretical and computational aspects of the TLS problem is given in Van Huffel and Vandewalle [24]. Solving the TLS problem requires the computation of the smallest singular value and the corresponding right singular vector of $(A,\ b)$. When A is large and sparse this is a much more difficult problem than that of computing the LS solution. For example, it is usually not feasible to compute the SVD or any other two-sided orthogonal factorization of A since the factors typically are not sparse.

Iterative algorithms for computing the singular subspace of a matrix associated with its smallest singular values, with applications to TLS problems with slowly varying data, have previously been studied by Van Huffel [23]. Three iterative methods, namely inverse iteration, Chebyshev, and inverse Chebyshev iterations are analyzed and compared. In [26, 4] a new class of methods based on a Rayleigh quotient iteration (RQI) was developed for the efficient solution of large scale TLS problems. Related methods for Toeplitz systems were studied by Kamm and Nagy [13].

In this paper we further develop the methods first presented in [4] and give numerical results. Similar algorithms for solving large scale multidimensional TLS problems will be considered in a forthcoming paper [5].

In section 2 we recall how the solution to the TLS problem in the so-called generic case can be expressed in terms of the smallest singular value and corresponding right singular vector of the compound matrix $(A,\ b)$. We discuss the conditioning of the LS and TLS problems and illustrate how the TLS problem can rapidly become intractable. Section 3 first reviews a Newton iteration for solving a secular equation. For this method to converge to the TLS solution, strict conditions on the initial approximation have to be satisfied. We then derive the RQI method, which ultimately achieves cubic convergence. The choice of initial estimates and termination criteria are discussed. A preconditioned conjugate gradient method (PCGTLS) is developed in section 4 for the efficient solution of the resulting sequence of sparse symmetric linear systems. Finally, in section 5, numerical results are given which confirm the rapid convergence and numerical stability of this class of methods.

We remark that the methods discussed here all compute a perturbation E, which in general is dense, even when A is sparse. Sometimes it is desired to find a perturbation E that preserves the sparsity structure of A. A Newton method for this more difficult problem has been developed by Rosen, Park, and Glick [20]. However, the complexity of this algorithm limits its applications to fairly small-sized problems. Recently a method that has the potential to be applied to large sparse problems has been given by Yalamov and Yuan [25]. Although their algorithm only converges with linear rate, this may suffice to obtain a low accuracy solution.

2. Preliminaries.

2.1. The TLS problem. The TLS problem (1.2) is equivalent to finding a perturbation matrix $(E,\ f)$ having minimal Frobenius norm, which lowers the rank of the matrix (A, b). Hence it can be analyzed in terms of the singular value decomposition

$$(A,\ b) = U\Sigma V^T, \qquad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n+1}),$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+1} \geq 0$ are the singular values of $(A,\ b)$. Note that by the minmax characterization of singular values it follows that the singular values ${\sigma'}_i$ of A interlace those of $(A,\ b)$, i.e.,

$$\sigma_1 \geq {\sigma'}_1 \geq \sigma_2 > \cdots \geq \sigma_n \geq {\sigma'}_n \geq \sigma_{n+1}. \tag{2.1}$$

We assume in the following that A has full rank, that is, ${\sigma'}_n > 0$, and that $\sigma_n > \sigma_{n+1}$. Then the minimum is attained for the rank one perturbation

$$(E,\ f) = -(A,\ b)v_{n+1}v_{n+1}^T = -\sigma_{n+1}u_{n+1}v_{n+1}^T,$$

for which $\|(E,\ f)\|_F = \sigma_{n+1}$. A TLS solution is then obtained from the right singular vector

$$v_{n+1} = \begin{pmatrix} z \\ \zeta \end{pmatrix} = -\zeta \begin{pmatrix} x_{TLS} \\ -1 \end{pmatrix}, \tag{2.2}$$

provided that $\zeta \neq 0$. If $\zeta = 0$ the TLS problem is called *nongeneric*, and there is no solution. This case cannot occur if ${\sigma'}_n > \sigma_{n+1}$, and in the following we always assume that this condition holds.

From the characterization (2.2) it follows that $\lambda = \sigma_{n+1}^2$ and $x = x_{TLS}$ satisfy the system of nonlinear equations

$$\begin{pmatrix} A^TA & A^Tb \\ b^TA & b^Tb \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \lambda \begin{pmatrix} x \\ -1 \end{pmatrix}. \tag{2.3}$$

Putting $\lambda = \sigma_{n+1}^2$ the first block row of this system of equations can be written

$$(A^TA - \sigma_{n+1}^2 I)x = A^Tb, \tag{2.4}$$

which can be viewed as "the normal equations" for the TLS problem. Note that from our assumption that ${\sigma'}_n > \sigma_{n+1}$ it follows that $A^TA - \sigma_{n+1}^2 I$ is positive definite.

2.2. Conditioning of the TLS problem. For the evaluation of accuracy and stability of the algorithms to be presented we need to know the sensitivity of the TLS problem to perturbations in data. We first recall that if $x_{LS} \neq 0$ the condition number for the LS problem is (see [3, Sec. 1.4])

$$\kappa_{LS}(A,b) = \kappa(A)\left(1 + \frac{\|r_{LS}\|_2}{{\sigma'}_n\|x_{LS}\|_2}\right), \tag{2.5}$$

where $\kappa(A) = {\sigma'}_1/{\sigma'}_n$. Note that the condition number depends on both A and b and that for large residual problems the second term may dominate.

Equation (2.4) shows that the TLS problem is always worse conditioned than the LS problem. Golub and Van Loan [10] showed that an approximate condition number for the TLS problem is

$$\kappa_{TLS}(A,b) = \frac{{\sigma'}_1}{{\sigma'}_n - \sigma_{n+1}} = \kappa(A)\frac{{\sigma'}_n}{{\sigma'}_n - \sigma_{n+1}}. \tag{2.6}$$

When $1 - \sigma_{n+1}/{\sigma'}_n \ll 1$ the TLS condition number can be much greater than $\kappa(A)$. The relation between the two condition numbers (2.5) and (2.6) depends on the relation between the $\|r_{LS}\|_2$ and σ_{n+1}, which is quite intricate. (For a study of this relation in another context, see Paige and Strakoš [16].)

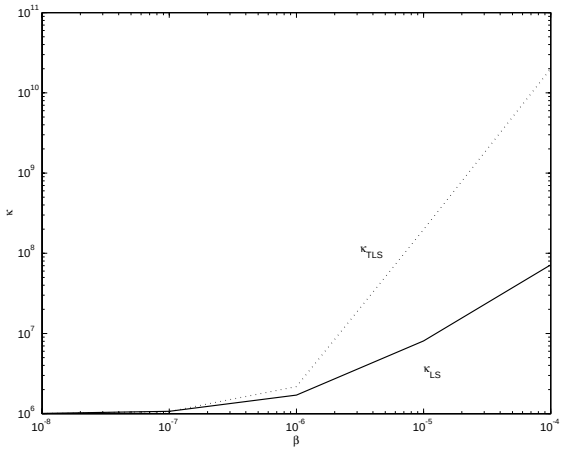


FIG. 2.1. *Condition numbers κ_{LS} and κ_{TLS} as a function of $\beta = \|r_{LS}\|_2$.*

From (2.3), multiplying from the left with $(x^T \ -1)$ we get

$$\|r_{TLS}\|_2^2 = \sigma_{n+1}^2(\|x_{TLS}\|_2^2 + 1), \qquad r_{TLS} = b - Ax_{TLS}.$$

Since $\|r_{LS}\|_2 \le \|r_{TLS}\|_2$ and $\sigma_{n+1} \le \sigma'_n$ it follows that

$$\|x_{TLS}\|_2^2 \ge (\|r_{LS}\|_2/\sigma'_n)^2 - 1. \tag{2.7}$$

This inequality is weak, but it shows that $\|x_{TLS}\|_2$ will be large when $\|r_{LS}\|_2 \gg \sigma'_n$.

As an illustration we consider the following small overdetermined system

$$\begin{pmatrix} \sigma'_1 & 0 \\ 0 & \sigma'_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \beta \end{pmatrix}. \tag{2.8}$$

Trivially, the LS solution is

$$x_{LS} = (c_1/\sigma'_1, c_2/\sigma'_2)^T, \qquad \|r_{LS}\|_2 = |\beta|.$$

If we take in (2.8) $\sigma'_1 = c_1 = 1$, $\sigma'_2 = c_2 = 10^{-6}$, then $x_{LS} = (1,\ 1)^T$ independent of β, and hence does not reflect the ill-conditioning of A. The TLS solution is of similar size as the LS solution as long as $|\beta| \le \sigma'_2$. However, when $|\beta| \gg \sigma'_2$ then from (2.7) it follows that $\|x_{TLS}\|_2$ is large.

In Figure 2.1 the two condition numbers are plotted as a function of $\beta \in [10^{-8}, 10^{-4}]$. We note that κ_{LS} increases proportionally to β because of the second term in (2.5). For $\beta > \sigma'_2$ the condition number κ_{TLS} grows proportionally to β^2. It can be verified that $\|x_{TLS}\|_2$ also grows proportionally to β^2.

3. Newton and Rayleigh quotient methods.

3.1. A Newton method. Equation (2.3) constitutes a system of $(n+1)$ nonlinear equations in x and λ. One way to proceed (see [13]) is to eliminate x to obtain the rational secular equation for $\lambda = \sigma_{n+1}^2$:

$$g(\lambda) = -b^T(b - Ax(\lambda)) + \lambda = 0, \tag{3.1}$$

where $x(\lambda) = (A^TA - \lambda I)^{-1}A^Tb$. Newton's method applied to (3.1) leads to the iteration

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{b^T(b - Ax^{(k)}) - \lambda^{(k)}}{1 + \|x^{(k)}\|_2^2}, \tag{3.2}$$

$$x^{(k)} = (A^TA - \lambda^{(k)}I)^{-1}A^Tb. \tag{3.3}$$

This iteration will converge monotonically at a rate that is asymptotically quadratic. The convergence can be improved by using a rational interpolation similar to that in [7] to solve the secular equation. However, in any case, λ will converge to σ_{n+1}^2 and $x^{(k)}$ to the TLS solution only if the initial approximation satisfies

$$\lambda^{(0)} \in (\sigma_{n+1}^2, \sigma_n'^2). \tag{3.4}$$

In general it is hard to verify this assumption. For the special case of a Toeplitz TLS problem Kamm and Nagy [13] use a bisection algorithm based on a fast algorithm for factorizing Toeplitz matrices to find an initial starting value satisfying (3.4).

3.2. The RQI method. The main drawback of the Newton method above is that unless (3.4) is satisfied it will converge to the wrong solution. A different Newton method is obtained as follows. Note that equation (2.3) can be rewritten as

$$\begin{pmatrix} A^T \\ b^T \end{pmatrix} \begin{pmatrix} A & b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = \begin{pmatrix} A^T \\ b^T \end{pmatrix} (-r) = \lambda \begin{pmatrix} x \\ -1 \end{pmatrix},$$

where $r = b - Ax$. Hence we can apply Newton's method to the nonlinear system in x and λ:

$$\begin{pmatrix} f(x,\lambda) \\ g(x,\lambda) \end{pmatrix} = \begin{pmatrix} -A^Tr - \lambda x \\ -b^Tr + \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.5}$$

As remarked in [19] this is closely related to inverse iteration, which is one of the most widely used methods for refining eigenvalues and eigenvectors. Rayleigh quotient iteration (RQI) is inverse iteration with a shift equal to the Rayleigh quotient. RQI has *cubic convergence* for the symmetric eigenvalue problem (see [17, Sections 4–7]) and is superior to the standard Newton method applied to (3.5).

For the eigenvalue problem (2.3) the Rayleigh quotient equals

$$\rho(x) = \frac{(x^TA^T - b^T)(Ax - b)}{x^Tx + 1} = \frac{r^Tr}{x^Tx + 1}. \tag{3.6}$$

Let $x^{(k)}$ be the current approximation and let ρ_k be the corresponding Rayleigh quotient. Then the next approximation $x^{(k+1)}$ in RQI and the scaling factor β_k are obtained from the symmetric linear system

$$\begin{pmatrix} J^{(k)} & A^Tb \\ b^TA & \eta_k \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ -1 \end{pmatrix} = \beta_k \begin{pmatrix} x^{(k)} \\ -1 \end{pmatrix}, \tag{3.7}$$

where

$$J^{(k)} = A^TA - \rho_kI, \qquad \eta_k = b^Tb - \rho_k.$$

If $J^{(k)}$ is positive definite the solution can be obtained by block Gaussian elimination,

$$\begin{pmatrix} J^{(k)} & A^Tb \\ 0 & \tau_k \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ -1 \end{pmatrix} = \beta_k \begin{pmatrix} x^{(k)} \\ -(z^{(k)})^Tx^{(k)} - 1 \end{pmatrix}, \tag{3.8}$$

where

$$J^{(k)}z^{(k)} = A^T b, \qquad \tau_k = b^T(b - Az^{(k)}) - \rho_k. \tag{3.9}$$

It follows that $x^{(k+1)} = z^{(k)} + u^{(k)}$, where

$$J^{(k)}u^{(k)} = \beta_k x^{(k)}, \qquad \beta_k = \tau_k/((z^{(k)})^T x^{(k)} + 1). \tag{3.10}$$

In [3] a reformulation was made to express the solution in terms of the residual vectors of (3.5):

$$\begin{pmatrix} f^{(k)} \\ g^{(k)} \end{pmatrix} = \begin{pmatrix} -A^T r^{(k)} - \rho_k x^{(k)} \\ -b^T r^{(k)} + \rho_k \end{pmatrix}, \tag{3.11}$$

where $r^{(k)} = b - Ax^{(k)}$. This uses the following formulas to compute τ_k:

$$J^{(k)}w^{(k)} = -f^{(k)}, \qquad z^{(k)} = x^{(k)} + w^{(k)}, \tag{3.12}$$

$$\tau_k = (z^{(k)})^T f^{(k)} - g^{(k)}. \tag{3.13}$$

The RQI is defined by equations (3.10)–(3.13).

3.3. Initial estimate and global convergence. Parlett and Kahan [18] have shown that for almost all initial vectors the RQI converges to some singular value and vector pair. However, in general we cannot say to *which* singular vector RQI will converge.

If the LS solution is known, a suitable starting approximation for λ may be

$$\rho(x_{LS}) = \frac{\|r_{LS}\|^2}{\|x_{LS}\|^2 + 1}. \tag{3.14}$$

Conditions to ensure that RQI will converge to the TLS solution from the starting approximation $(\rho(x_{LS}), x_{LS})$ are in general difficult to verify and often not satisfied in practice. However, in contrast to the simple Newton iteration in section 3.1, the method may converge to the TLS solution even when the initial approximation $\rho(x_{LS}) \notin (\sigma_{n+1}^2, \sigma'^2_n)$.

The Rayleigh quotient $\rho(x_{LS})$ will be a large overestimate of σ_{n+1}^2 when the residual norm $\|r_{LS}\|_2$ is large and $\|x_{LS}\|_2$ does not reflect the ill-conditioning of A. Note that it is typical for ill-conditioned LS problems that the right-hand side is such that $\|x_{LS}\|_2$ is not large! For example, LS problems arising from ill-posed problems usually satisfy a so-called Picard condition, which guarantees that the right-hand side has this property; see [11, Section 1.2.3].

Szyld [22] suggested that $p \geq 1$ steps of inverse iteration are applied initially before switching to RQI, in order to ensure convergence to the smallest eigenvalue. Inverse iteration for σ_{n+1}^2 corresponds to taking $\sigma^2 = 0$ in the RQI algorithm. Starting from $x = x_{LS}$ the first step of inverse iteration simplifies as follows. Using (3.9) and (3.10) with $\rho_k = 0$ and $x^{(k)} = x_{LS}$ we get

$$z^{(k)} = x_{LS}, \qquad \tau_k = \|r_{LS}\|_2^2,$$

and the new approximation becomes

$$x_{INV} = x_{LS} + \beta(A^T A)^{-1} x_{LS}, \qquad \beta = \rho(x_{LS}).$$

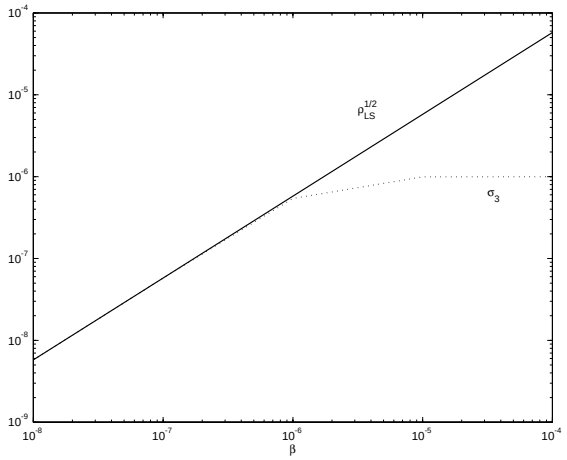


FIG. 3.1. *Rayleigh quotient approximation and σ_3 for $|\beta| = \|r_{LS}\|_2 = 10^{-k}$.*

Several steps of inverse iteration may be needed to ensure convergence of RQI to the smallest singular value. However, since inverse iteration only converges linearly, taking more than one step will often just hold up the rapid convergence of RQI. In general we therefore recommend as default taking just one step.

To illustrate the situation consider again the small 3×2 system (2.8) with $\sigma'_1 = c_1 = 1$, $\sigma'_2 = c_2 = 10^{-6}$. This has the LS solution $x_1 = x_2 = 1$, which does not reflect the ill-conditioning of A ($\kappa = 10^6$). With $\|r_{LS}\|_2 = \beta$ the initial Rayleigh quotient approximation equals

$$\rho(x_{LS}) = \beta^2/(1+2) = \beta^2/3.$$

By the interlacing property we have that $\sigma_3 \le \sigma'_2$. Since $|\beta| \gg \sigma'_2$ it is clear that the Rayleigh quotient fails to approximate σ_3^2. This is illustrated in Figure 3.1, where $\rho(x_{LS})^{1/2}$ and σ_3 are plotted as a function of $|\beta|$. It is easily verified, however, that after one step of inverse iteration $\rho(x_{INV})$ will be close to σ'^2_2.

3.4. Termination criteria for RQI. The RQI algorithm for the TLS problem is defined by (3.10)–(3.13). When should the RQI be terminated? We suggest two different criteria.

The first is based on the key fact in the proof of global convergence that the normalized residual norm

$$\gamma_k = \left(\frac{\|f_k\|_2^2 + g_k^2}{\|x^{(k)}\|_2^2 + 1} \right)^{1/2}, \qquad \begin{pmatrix} f_k \\ g_k \end{pmatrix} = \begin{pmatrix} -A^T r^{(k)} - \rho_k x^{(k)} \\ -b^T r^{(k)} + \rho_k \end{pmatrix} \tag{3.15}$$

always decreases, $\gamma_{k+1} \le \gamma_k$, for all k. Thus, if an increase in the norm occurs this must be caused by roundoff, and then it makes no sense to continue the iterations. This suggests that we terminate the iterations with x_{k+1} when

$$\gamma_{k+1} > \gamma_k. \tag{3.16}$$

A second criterion is based on the observation that since the condition number for computing σ_{n+1} equals 1, we can expect to obtain σ_{n+1} to full machine precision. Since convergence of RQI is cubic, a criterion could be to stop when the change in

the approximation to σ_{n+1} is of the order of $\sigma_1 u^{1/p}$, where $p = 3$. (A similar criterion with $p = 2$ is used by Kamm and Nagy [13] for terminating the Newton iteration.) However, as will be evident from the numerical results in section 5, full accuracy in x_{TLS} in general requires one more iteration after σ_{n+1} has converged. Therefore we recommend to stop when either (3.16) or

$$|\rho(x_{k+1}) - \rho(x_k)| \le Cu \tag{3.17}$$

is satisfied, where u is the machine unit and C a suitable constant.

We summarize below the RQI algorithm with one step of inverse iteration (cf. [4]):

ALGORITHM 3.1 (RQI).

$x = x_{LS}$;
$r = b - Ax$;
$\sigma^2 = r^T r/(1 + x^T x)$;
solve $A^T Au = x$;
$x = x + \sigma^2 u$;
for $k = 1, 2, \ldots$
    $r = b - Ax$;
    $\sigma^2 = r^T r/(1 + x^T x)$;
    $f = -A^T r - \sigma^2 x$;
    $g = -b^T r + \sigma^2$;
    solve $(A^T A - \sigma^2 I)w = -f$;
    $z = x + w$;
    $\beta = (z^T f - g)/(z^T x + 1)$;
    solve $(A^T A - \sigma^2 I)u = x$;
    $x = z + \beta u$;
end

3.5. Rounding errors and stability. If the RQI converges, then $f^{(k)}$, $g^{(k)}$, and β_k will tend to zero. Consider the rounding errors that occur in the evaluation of the residuals (3.11). Let $\tilde{u} = 1.06u$, where u is the unit roundoff; see [12, Chapter 3]. Then the computed residual vector satisfies $\bar{r} = r + \delta r$, where

$$\|\delta r\|_2 \le n\tilde{u}(\|b\|_2 + \|A\|_2\|x\|_2).$$

Obviously convergence will cease when the residuals (3.11) are dominated by roundoff. Assume that we perform one iteration from the exact solution, x_{TLS}, r_{TLS}, and $\lambda = \sigma_{n+1}^2$. Then the first correction to the current approximation is obtained by solving the linear system in (3.12), which now becomes

$$(A^T A - \sigma_{n+1}^2 I)w^{(k)} = -A^T \delta r^{(k)}. \tag{3.18}$$

For the correction this gives the estimate

$$\|w^{(k)}\|_2 = \frac{n\tilde{u}\sigma'_n}{\sigma'^2_n - \sigma^2_{n+1}}\big(\|b\|_2 + \|A\|_2\|x_{TLS}\|_2\big). \tag{3.19}$$

This estimate is consistent with the condition estimate for the TLS problem.

We note that the equations (3.18) are of similar form to those that appear in the corrected seminormal equations for the LS problem; see [2], [3, Section 6.6.5]. A detailed roundoff error analysis similar to that done for the LS problem would become very complex and is not attempted here. It seems reasonable to conjecture that if ${\sigma'}_n^2 - \sigma_{n+1}^2 < u^{1/2}$, it will suffice to solve the linear equations for the correction $w^{(k)}$ using the Cholesky factorization of $(A^TA - \sigma_{n+1}^2 I)$. Methods for the solution of the linear systems are considered in more detail in section 4.

4. Solving the linear systems. In the RQI method formulated in the previous section the main work consists of solving in each step two linear systems of the form

$$(A^TA - \sigma^2 I)w = f, \qquad \sigma \approx \sigma_{n+1}. \tag{4.1}$$

Here σ is an approximation to σ_{n+1} and varies from step to step. Provided that $\sigma < {\sigma'}_n$, the system (4.1) is symmetric and positive definite.

4.1. Direct linear solvers. If $\sigma < {\sigma'}_n$, then the system (4.1) can be solved by computing the (sparse) Cholesky factorization of the matrix $A^TA - \sigma^2 I$. Note that A^TA only has to be formed once and the symbolic phase of the factorization does not have to be repeated. However, it is a big disadvantage that a new numerical factorization has to be computed at each step of the RQI algorithm.

For greater accuracy and stability in solving LS problems it is often preferred to use a QR factorization instead of a Cholesky factorization. However, since in the TLS normal equations the term $\sigma^2 I$ is *subtracted* from A^TA, this is not straightforward. The Cholesky factor of the matrix $A^TA - \sigma^2 I$ can be obtained from the QR factorization of the matrix $\begin{pmatrix} A^T & i\sigma I \end{pmatrix}^T$, where i is the imaginary unit. This is a downdating problem for the QR factorization and can be performed using stabilized hyperbolic rotations (see [3, pp. 143–144]) or hyperbolic Householder transformations (see [21]). However, in the sparse case this is not an attractive alternative, since it would require nontrivial modifications of existing software (see, e.g., Matstoms [15] and Adlers [1]) for sparse QR factorization.

4.2. Iterated deregularization. One way to solve the TLS normal equations using only a single factorization of A^TA would be to adapt an iterated regularization scheme due to Riley and analyzed by Golub [9]. In this scheme, we solve the TLS normal equations by the iteration $x^{(0)} = 0$, and for $k = 0, 1, \ldots$

$$\begin{aligned} r^{(k)} &= b - Ax^{(k)}, \\ A^TA\delta^{(k)} &= A^Tr^{(k)} + \sigma^2 x^{(k)}, \\ x^{(k+1)} &= x^{(k)} + \delta^{(k)}. \end{aligned}$$

If $\lim_{k\to\infty} x^{(k)} = x$, then $(A^TA - \sigma^2 I)x = A^Tb$. This iteration will converge with the linear rate equal to $\rho = \sigma^2/{\sigma'}_n^2$ provided that $\rho < 1$, and it may be implemented very efficiently if the QR decomposition of A is available. We do not pursue this method further, since it has no advantage over the preconditioned conjugate gradient method developed in [4].

4.3. A preconditioned conjugate gradient algorithm (PCGTLS). Performing the change of variables $y = Sw$, where S is a given nonsingular matrix, and multiplying from the left with S^{-T} the system (4.1) becomes

$$(S^{-T}A^TAS^{-1} - \sigma^2 S^{-T}S^{-1})y = S^{-T}f. \tag{4.2}$$

This system is symmetric positive definite provided that $\sigma < \sigma'_n$, and hence the conjugate gradient (CG) method can be applied. We can use for S the same preconditioners as have been developed for the LS problem; for a survey, see [3, chapter 7].

In the following we consider a special choice of preconditioner, the *complete* Cholesky factor R of A^TA (or R from a QR decomposition of A). Unless A is huge this is often a feasible choice, since efficient software for sparse Cholesky and sparse QR factorization are readily available [3, chapter 7]. Using $AR^{-1} = Q_1$, where $Q_1^TQ_1 = I$, the preconditioned system (4.2) simplifies to

$$(I - \sigma^2 R^{-T}R^{-1})y = R^{-T}f, \qquad w = R^{-1}y. \tag{4.3}$$

(Note that although A and A^T have disappeared from this system of equations, matrix-vector multiplications with these matrices are used to compute the right-hand side f!) In the inverse iteration step used in the initialization, $\sigma = 0$, and the solution $w = R^{-1}R^{-T}f$ is obtained by two triangular solves.

The standard CG method applied to the system (4.2) can be formulated in terms of the original variables w. The resulting algorithm is a slightly simplified version of the algorithm PCGTLS given in [4] and can be written as follows.

ALGORITHM 4.1 (PCGTLS). *Preconditioned gradient method for solving* $(A^TA - \sigma^2 I)w = f$, *using the Cholesky factor* R *of* A^TA *as preconditioner.*

Initialize: $w^{(0)} = 0, \quad p^{(0)} = s^{(0)} = R^{-T}f, \quad \eta_0 = \|s^{(0)}\|_2^2$.

For $j = 0, 1, \ldots, l$, while $\delta_j \neq 0$ compute

$$\begin{aligned}
&q^{(j)} = R^{-1}p^{(j)}\\
&\delta_j = \|p^{(j)}\|_2^2 - \sigma^2\|q^{(j)}\|_2^2\\
&\alpha_j = \eta_j/\delta_j\\
&w^{(j+1)} = w^{(j)} + \alpha_j q^{(j)}\\
&q^{(j)} = R^{-T}q^{(j)}\\
&s^{(j+1)} = s^{(j)} - \alpha_j(p^{(j)} - \sigma^2 q^{(j)})\\
&\eta_{j+1} = \|s^{(j+1)}\|_2^2\\
&\beta_j = \eta_{j+1}/\eta_j\\
&p^{(j+1)} = s^{(j+1)} + \beta_j p^{(j)}
\end{aligned}$$

Denote the original and the preconditioned matrix by $C = A^TA - \sigma^2 I$ and $\widetilde{C} = I - \sigma^2 R^{-T}R^{-1}$, respectively. Then a simple calculation shows that for $\sigma = \sigma_{n+1}$ the condition number of the transformed system is reduced by a factor of $\kappa(A)$,

$$\kappa(\widetilde{C}) = \left(\frac{(\sigma'_1)^2 - \sigma_{n+1}^2}{(\sigma'_n)^2 - \sigma_{n+1}^2}\right)\left(\frac{(\sigma'_n)^2}{(\sigma'_1)^2}\right) = \frac{\kappa(C)}{\kappa^2(A)}.$$

The spectrum of $\tilde{C}$ will be clustered close to 1. In particular in the limit when $\sigma \to \sigma_{n+1}$, the eigenvalues of $\widetilde{C}$ will lie in the interval

$$\left[1 - \sigma_{n+1}^2/(\sigma'_n)^2, 1\right]. \tag{4.4}$$

(Note the relation to the condition number κ_{TLS}!) Hence, unless $\sigma'_n \approx \sigma_{n+1}$, we can expect this choice of preconditioner to work very well for solving the shifted system (4.1).

The matrix $R^TR - \sigma^2 I$ is positive definite if $\sigma < \sigma'_n$. In this case $\delta_k > 0$ in PCGTLS, and the division in computing α_k can always be carried out. If $\sigma \geq \sigma'_n$, then the system (4.2) is not positive definite and a division by zero can occur. This can be avoided by including a test to ensure that $\delta_k > 0$. If $\delta_k < 0$, or equivalently $\|p^{(k)}\|_2 < \sigma\|q^{(k)}\|_2$, the CG iterations are considered to have failed. The RQI step is then repeated with a new smaller value of σ^2_{n+1}, e.g.,

$$\sigma^2 = \frac{1}{2}\|p^{(k)}\|_2^2/\|q^{(k)}\|_2^2. \tag{4.5}$$

The accuracy of TLS solutions computed by RQI will basically depend on the accuracy residuals and the stability of the method used to solve the linear systems (4.1). We note that the CG method CGLS1 for the LS problem, which is related to PCGTLS, has been shown to have very good numerical stability properties; see [6].

4.4. Termination criteria in PCGTLS. The RQI, using PCGTLS as an inner iteration for solving the linear systems, is an inexact Newton method for solving a system of nonlinear equations. Such methods have been studied by Dembo, Eisenstat, and Steihaug [8], who consider the problem of how to terminate the iterative solver so that the rate of convergence of the outer Newton method is preserved.

Consider the iteration

$$F'(x_k)s_k = -F(x_k) + r_k, \quad k = 0, 1, \ldots,$$

where r_k is the residual error. In [8] it is shown that maintaining a convergence order of $1 + p$ requires that when $k \to \infty$, the residuals satisfy inequalities

$$\|r_k\| \leq \eta_k\|F(x_k)\|, \qquad \eta_k = O(\|F(x_k)\|^p), \tag{4.6}$$

where η_k is a forcing sequence.

In practice the above asymptotic result turns out to be of little practical use in our context. Once the asymptotic cubic convergence is realized, the ultimate accuracy possible in double precision already has been achieved. A more practical, ad hoc termination criterion for the PCGTLS iterations will be described together with the numerical results reported below.

Remark. In the second linear system to be solved in RQI, $(A^TA - \sigma^2 I)u = x$, the right-hand side converges to x_{TLS}. Hence it is tempting to use the value of u obtained from the last RQI to initialize PCGTLS in the next step. However, our experience is that this slows down the convergence compared to initializing u to zero.

5. Numerical results.

5.1. Accuracy and termination criteria. Numerical tests were performed in MATLAB on a SUN SPARC station 10 using double precision with unit roundoff $u = 2.2 \cdot 10^{-16}$. For the initial testing we used contrived test problems $[A, b] = P(m, n, \epsilon)$, similar to those in [6]. These test problems are neither large nor sparse, but they will be used to test the convergence properties of the algorithm. They are generated in the following way: Let

$$\widetilde{A} = Y \begin{pmatrix} D \\ 0 \end{pmatrix} Z^T \in \mathcal{R}^{m\times n},$$

where Y, Z are random orthogonal matrices and $D = \mathrm{diag}(1, 2^{-1}, \ldots, 2^{-n+1})$. Further, let

$$x = (1, 1/2, \ldots, 1/n), \qquad \widetilde{b} = \widetilde{A}x.$$

This ensures that the norm of the solution does not reflect the ill-conditioning of A. We then add random perturbations

$$A = \widetilde{A} + E, \qquad b = \widetilde{b} + r,$$
$$E = \epsilon * \text{rand}(m, n), \quad r = \epsilon * \text{rand}(m, 1).$$

Note that since $\sigma'_n = 2^{-n+1}$ there is a perturbation E to A with $\|E\|_2 = 2^{n-1}$, which makes A rank deficient. It is therefore not realistic to consider perturbations such that $m\epsilon \geq 2^{-n+1}$.

To test the termination criteria for the inner iterations we used problem $P(30, 15)$, $\sigma'_n = 2^{-14} = 6.1 \cdot 10^{-5}$, with error level $\epsilon = 10^{-6}$. The linear systems arising in RQI were solved using PCGTLS with the Cholesky factor of $A^T A$ as preconditioner. According to the criterion (4.6) the linear systems should be solved more and more accurately as the RQI method converges. The rate of convergence of PCGTLS depends on the ratio σ_{n+1}/σ'_n (see (4.4)) and is usually very rapid. We have used a very simple strategy where in the kth step of RQI $(k + \nu)$ PCGTLS iterations are performed. Here $\nu \geq 0$ is a parameter to be chosen. Note that since no refactorizations are performed the object should be to minimize the total number of PCGTLS iterations.

Figure 5.1 shows a plot of the errors $\|x^{(k)} - x\|_2$ and $|\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ (logarithmic scale) after k RQI iterations, for $\nu = 0, 1, 2$. The plots for $\nu = 1$ and $\nu = 2$ are almost indistinguishable, whereas $\nu = 0$ gives a slight delay in convergence. Indeed, for this problem taking $k + 1$ iterations in PCGTLS suffices to give the same result as using a direct solver for the linear systems. Note that the final accuracy in x (σ_{n+1}) is on the order of 10^{-11} and (10^{-17}). This confirms the excellent stability of the method.

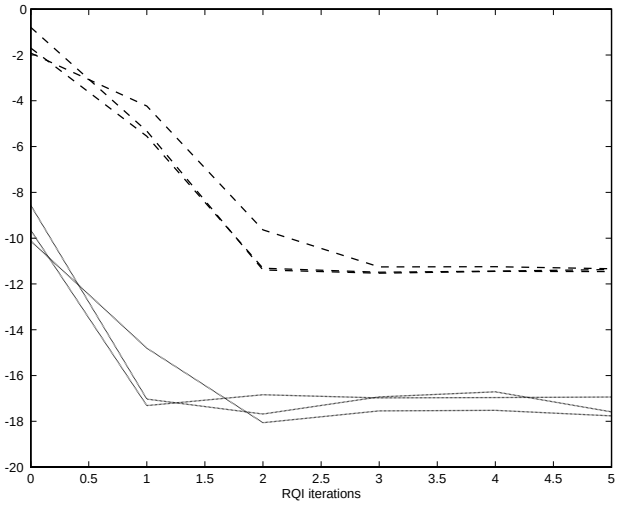


FIG. 5.1. *Errors* $\log \|x^{(k)} - x\|_2$ $(--)$ *and* $\log |\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ $(- \cdot -)$ *for problem PS*$(30, 15)$*, with* $\epsilon = 10^{-6}$ *(*$\hat{\sigma}_n = 2^{-14}$*). Linear systems solved by PCGTLS with* $k + \nu$ *iterations,* $\nu = 0, 1, 2$.

Based on these considerations and the test results we recommend taking $\nu = 1$, although $\nu = 0$ should work well for problems where the ratio σ_{n+1}/σ'_n is smaller. In all the following tests we have used $\nu = 1$. In Figure 5.2 we show results for problem P(30,15), and different error levels $\epsilon = 10^{-8}, 10^{-7}, 10^{-6}$. Here 1, 2, and 3–4

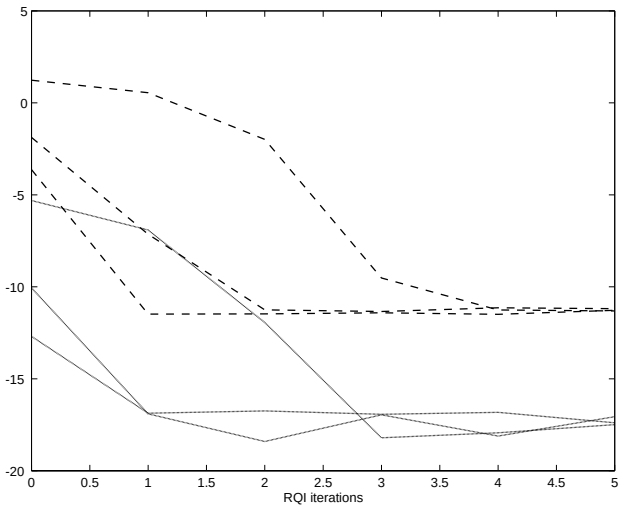


FIG. 5.2. *Errors* $\log \|x^{(k)} - x\|_2$ $(--)$ *and* $\log |\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ $(- \cdot -)$ *for problem PS*$(30, 15)$, $\epsilon = 10^{-8}, 10^{-7}, 10^{-6}$, $\hat{\sigma}_n = 2^{-14}$. *Linear systems solved by PCGTLS with* $k+1$ *iterations.*

RQIs, respectively, were needed to achieve an accuracy of about 10^{-11} in x_{TLS}. Since $\sigma'_n = 2^{-15} = 3.05 \cdot 10^{-5}$, this is equal to the best limiting accuracy that can be expected. Note also that the error in σ_{n+1} converges to machine precision, usually in one less iteration, which supports the use of the criterion (3.17) to terminate RQI.

5.2. Improvement from inverse iteration. We now show the improvement resulting from including an initial step of inverse iteration. In Figure 5.3 we show results for the same problems as considered in Figure 5.2. For the first two error levels only one RQI now suffices to obtain limiting accuracy. For the highest error level σ_{n+1} converges in two iterations and x_{TLS} in three.

We now consider the second test problem in [13], which is defined as

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \cdot & & & & & & & \\ \cdot & & & & & & & \\ \cdot & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ n-1 \end{bmatrix} + e = \tilde{g} + e,$$

where $A \in \mathbf{R}^{n \times n-1}$. Here e is a vector with entries generated randomly from a normal distribution with mean 0.0 and variance 1.0, and scaled so that $||e||_2 = \eta ||\tilde{g}||_2$. For $n = 100$ we have $\kappa(A) = 2.62 \cdot 10^3$ and for $\eta = 0.01$ the condition numbers in (2.5)–(2.6) are

$$\kappa_{LS} = 3.98 \cdot 10^5, \qquad \kappa_{TLS} = 1.25 \cdot 10^8,$$

respectively. This problem has features similar to those of the small ill-conditioned example discussed previously in section 2.2, although here the norm of the solution x_{LS} is large.

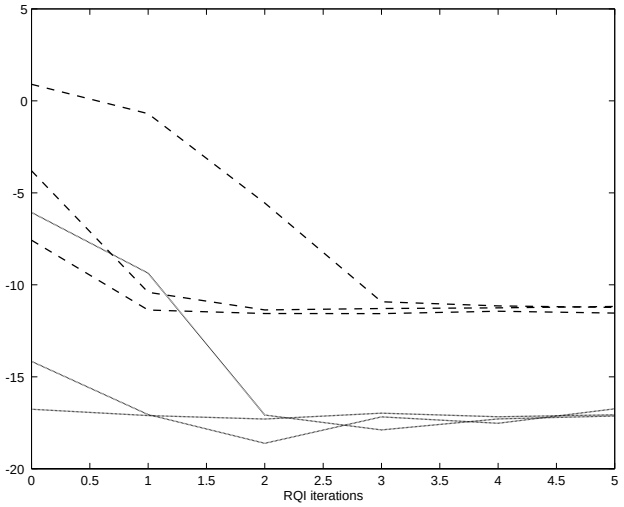


FIG. 5.3. *Errors* $\log \|x^{(k)} - x\|_2$ $(--)$ *and* $\log |\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ $(- \cdot -)$ *for problem PS*$(30, 15)$, $\epsilon = 10^{-8}, 10^{-7}, 10^{-6}$, $\hat{\sigma}_n = 2^{-14}$. *One step of inverse iteration and RQI. Linear systems solved by PCGTLS with* $k + 1$ *iteration.*

Applying the RQI algorithm we obtained the results shown in Figure 5.4. The initial approximation $\rho(x_{LS})$ is here far outside the interval $[\sigma_{n+1}, \sigma'_n)$. Thus the matrix $A^T A - \sigma^2 I$ is initially not positive definite and we cannot guarantee the existence of the Cholesky factor. However, the algorithm PCGTLS still does not break down, and, as shown in Figure 5.4, the limiting accuracy is obtained after five RQIs. This surprisingly good performance of RQI can be explained by the fact that even though x_{LS} does not approximate x_{TLS} well, the angle between them is small; the cosine

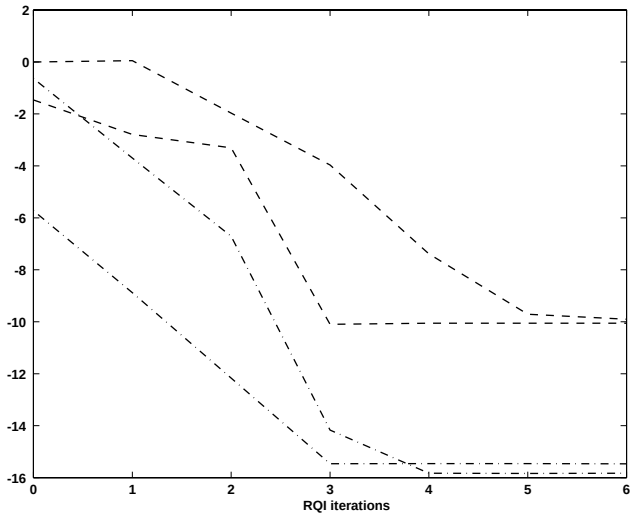


FIG. 5.4. *Errors* $\log \|x^{(k)} - x\|_2$ $(--)$ *and* $\log |\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ $(- \cdot -)$ *for second test problem with* $\eta = 0.001$. *Results for RQI without/with one step of inverse iteration.*

equals 0.98453. Performing one initial step of inverse iteration, the limiting accuracy is obtained after only three RQI steps.

Performing one step of inverse iteration before applying the RQI algorithm gives much improved convergence. The one initial step of inverse iteration here suffices to give an initial approximation in the interval $[\sigma_{n+1}, \sigma'_n)$. This can be compared with 12–23 steps of bisection needed to achieve such a starting approximation; see [13]! Three RQIs now give the solution x_{TLS} with an error close to the limiting accuracy; see Figure 5.4. In both cases we obtained σ_{n+1} to full machine precision. Also, the relative error norm of in the TLS solution was consistent with the condition number.

5.3. A problem in signal restoration. The Toeplitz matrix used in this example comes from an application in signal restoration; see [13, Example 3]. Specifically, an $n \times (n-2\omega)$ convolution matrix $\widetilde{T}$ is constructed to have entries in the first column given by

$$t_{i,1} = \frac{1}{\sqrt{2\pi\alpha^2}} \exp\left[\frac{-(\omega - i + 1)^2}{2\alpha^2}\right], \quad i = 1, 2, \ldots, 2\omega + 1,$$

and zero otherwise. Entries in the first row given by $t_{1,j} = t_{1,1}$ if $j = 1$, and zero otherwise, where $\alpha = 1.25$ and $\omega = 8$. A Toeplitz matrix T and right-hand side vector g is then constructed as $T = \widetilde{T} + E$ and $g = \tilde{g} + e$, where E is a random Toeplitz matrix with the same structure as T and e is a random vector. The entries in E and e are generated randomly from a normal distribution with mean 0.0 and variance 1.0, and they are scaled so that

$$\|e\|_2 = \eta\|\tilde{g}\|_2, \qquad \|E\|_2 = \eta\|\widetilde{T}\|_2.$$

In [13] difficulties with convergence were reported. However, these are due to the choice of right-hand side $\tilde{g}_1$, which was taken to be a vector of all ones. For the unperturbed problem ($\gamma = 0$) this vector is orthogonal to the space spanned by the left singular vector corresponding to the smallest singular value. Therefore the magnitude of the component in this direction of the initial vector x_{LS} will be very small, of the order γ. Also, although T is quite well conditioned the LS residual is large. The TLS problem is therefore close to a nongeneric problem and thus very ill-conditoned.

Because of the extreme ill-conditioning for this right-hand side, the behavior of any solution method becomes very sensitive to the particular random perturbation added. We have therefore instead chosen a right-hand side $\tilde{g}_2$ given by $\tilde{g}(i) = (m - 2i)/m$, $i = 1, \ldots, m$. For this the TLS problem is much better conditioned; see Table 5.1. Convergence is now obtained in just two iterations; see Figure 5.5.

TABLE 5.1
Condition numbers for test problem 3 *for right-hand sides* $\tilde{g}_i$, $n = 100$.

γ	i	$\kappa(A)$	κ_{LS}	κ_{TLS}
0	1	1.094484e+03	1.968723e+04	$>$ 1.0e+16
	2	1.094484e+03	2.101815e+04	3.069664e+07
0.001	1	1.220696e+03	2.538016e+04	1.692483e+10
	2	1.220696e+03	2.687055e+04	1.202459e+07

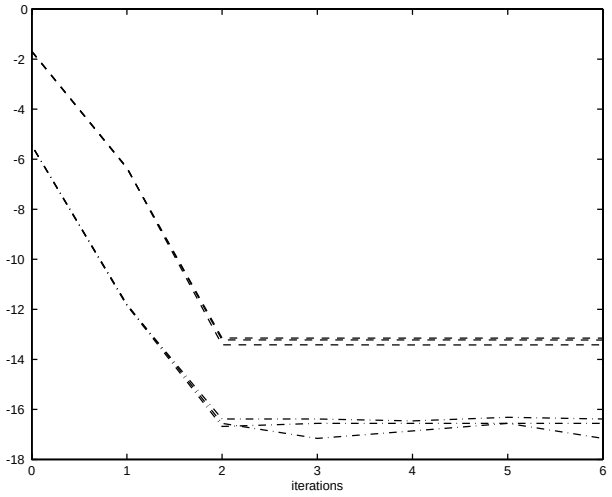


FIG. 5.5. *Errors* $\log \|x^{(k)} - x\|_2$ $(--)$ *and* $\log |\sigma_{n+1}^{(k)} - \sigma_{n+1}|$ $(- \cdot -)$ *for third test problem; RQI with one step of inverse iteration,* $n = 100$*,* $\eta = 0.0001, 0.001, 0.01$*.*

6. Conclusions. In this paper we have developed and analyzed an algorithm for solving large scale TLS problems based on a RQI for the smallest singular value σ_{n+1} and corresponding right singular vector of $(A,\ b)$. In this algorithm we need to solve a sequence of linear systems with symmetric, positive definite matrix $A^TA - \bar{\sigma}^2 I$, where $\bar{\sigma}$ is the current approximation to σ_{n+1}. This approach has the advantage that the linear systems can be solved by a *preconditioned* CG method. Further, for large and sparse TLS problems a (possibly incomplete) Cholesky factor of A^TA can usually be computed, which provides a very efficient preconditioner. Therefore our method can solve a much wider range of problems than it is possible to solve by using Lanczos-type algorithms directly for the singular value problem, which does not allow for the use of preconditioning.

Methods for solving the TLS problem are by necessity more complex than those for the (linear) LS problem. On the test problems we have tried so far our algorithm has only failed for almost singular problems. For such problems the TLS model is in any case not relevant and should not be used. Otherwise we conjecture that with the given ad hoc termination criteria for the inner (RQI) and outer (CG) iterations the algorithm computes the TLS solution with an accuracy compatible with a backward stable method. Although a detailed error analysis is not carried out, this conjecture is supported by numerical results.

As illustrated by the given examples, provided the Cholesky factor is available as preconditioner, rarely more than two RQI iterations will be needed. With the recommended strategy this requires $2(2 + 3) = 10$ steps of PCGTLS, each using one matrix-vector multiplication with A and A^T and one solve with R and R^T. (In addition the initial inverse iteration step requires one solve with R and R^T.) In many cases the total cost is dominated by the cost of computing the sparse Cholesky factor, and then the cost of computing x_{TLS} is of the same order as computing x_{LS}.

REFERENCES

[1] M. Adlers, *Computing Sparse QR Factorizations in MATLAB*, Tech. Report, LiTH-MAT-R-98-19, Linköping University, Sweden, 1999.

[2] Å. Björck, *Stability analysis of the method of semi-normal equations for least squares problems*, Linear Algebra Appl., 88/89 (1987), pp. 31–48.

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

[4] Å. Björck, *Newton and Rayleigh quotient methods for total least squares problems*, in Recent Advances in Total Least Squares Techniques and Errors–in–Variables Modeling: Proceedings of the Second International Workshop on Total Least Squares and Errors–in–Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 149–160.

[5] Å. Björck, *Solving Large Scale Multidimensional Total Least Squares Problems*, in preparation.

[6] Å. Björck, T. Elfving, and Z. Strakos, *Stability of conjugate gradient and Lanczos methods for linear least squares problems*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 720–736.

[7] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[9] G. H. Golub, *Numerical methods for solving least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[10] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[11] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Conversion*, SIAM, Philadelphia, 1998.

[12] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[13] J. Kamm and J. G. Nagy, *A total least squares method for Toeplitz systems of equations*, BIT, 38 (1998), pp. 560–582.

[14] W. Mackens and H. Voss, *The minimum eigenvalue of a symmetric positive-definite Toeplitz matrix and rational Hermitian interpolation*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 521–534.

[15] P. Matstoms, *Sparse QR factorization in MATLAB*, ACM Trans. Math. Software, 20 (1994), pp. 136–159.

[16] C. C. Paige and Z. Strakoš, *Weighted total least squares problems and bounds for the least squares distance*, Numer. Math., submitted.

[17] B. N. Parlett, *The Symmetric Eigenvalue Problem*, 2nd ed., SIAM, Philadelphia, 1998.

[18] B. N. Parlett and W. Kahan, *On the convergence of a practical QR algorithm*, in Information Processing 68, the Proceedings of the IFIP Congress, Edinburgh, Scotland, 1968, North-Holland, Amsterdam, The Netherlands, 1969, pp. 114–118.

[19] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method,* SIAM Rev., 21 (1979), pp. 339–360.

[20] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems,* SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.

[21] M. Stewart and G. W. Stewart, *On hyperbolic triangularization: Stability and pivoting*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 847–860.

[22] D. B. Szyld, *Criteria for combining inverse and Rayleigh quotient iteration,* SIAM J. Numer. Anal., 25 (1988), pp. 1369–1375.

[23] S. Van Huffel, *Iterative algorithms for computing the singular subspace of a matrix associated with its smallest singular values*, Linear Algebra Appl., 154/156 (1991), pp. 675–709.

[24] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.

[25] P. Y. Yalamov and J. Y. Yuan, *A Successive Least Squares Method for Structured Total Least Squares*, preprint.

[26] T. Yang, *Iterative Methods for Least Squares and Total Least Squares Problems*, Lic. thesis LiU-TEK-LIC-1996-25, Department of Mathematics, University of Linköping, Sweden, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 430–452

ON THE CONVERGENCE OF RESTARTED KRYLOV SUBSPACE METHODS*

V. SIMONCINI†

Abstract. We are interested in the convergence analysis of restarted Krylov subspace iterative methods for the solution of large nonsymmetric linear systems. Several contributions in the literature have associated the convergence to some spectral properties of the coefficient matrix, while little work has been devoted to investigating how the singular values of A may influence the convergence. In this paper we present new relations that can be used to monitor the behavior of the restarted methods, especially GMRES, when the coefficient matrix has small (but not tiny) singular values and the right-hand side has a dominant component onto the corresponding left singular space.

We also present some simple but insightful relations that highlight the dependence of the restarted schemes on new matrices; moreover, closed forms of the restarted solutions are used to relate the approximations of the unrestarted and restarted approaches.

Key words. linear systems, restarting, iterative methods, GMRES, full orthogonalization method

AMS subject classification. 65F10

PII. S0895479898348507

1. Introduction.

We are interested in the convergence analysis of restarted Krylov subspace iterative solvers for the solution of the large linear system of equations

$$Ax = b, \quad \text{with} \quad A \in \mathbb{R}^{n\times n},\ b \in \mathbb{R}^n. \tag{1.1}$$

Classical results associate the asymptotic convergence of the unrestarted schemes to some spectral properties of the coefficient matrix; see, e.g., [23]. More recently, pseudospectrum has been shown to be useful in characterizing asymptotic convergence [27], while harmonic Ritz values [21] have been employed to improve the convergence of the algorithm in a restarted context [18].

Little work has been devoted to investigating if the singular values of A play a role in the convergence of Krylov subspace methods. The main reason is that Krylov subspaces do not contain significant information on the singular triplets of A, and therefore asymptotic results cannot be clearly stated in terms of singular triplets. Nevertheless, certain singular value distributions provide an interesting setting for analyzing *restarted* Krylov subspace methods.

Restarted methods terminate the process after a fixed number of iterations and then repeat the procedure using the residual of the current approximate solution as new initial vector. In practice, however, we will see that a restarted Krylov subspace solver behaves as if the method were not interrupted but continued with a different approximation criterion. Therefore, as long as the generated global subspace retains maximum dimension, the behavior of the restarted scheme will not substantially differ from that of the full (unrestarted) method. The aim of this paper is to analyze the computation of the restarted quantities in order (i) to identify the quantities that influence the performance degradation of the restarted methods, and (ii) to compare

*Received by the editors December 4, 1998; accepted for publication (in revised form) by R. Freund March 17, 2000; published electronically July 11, 2000. This paper is a revised version of IAN-CNR Technical Report N. 1093, Istituto di Analisi Numerica, Pavia, Italy, November 1998.
http://www.siam.org/journals/simax/22-2/34850.html
†Istituto di Analisi Numerica - CNR, Via Ferrata, 1 - 27100 Pavia, Italy (val@ian.pv.cnr.it).

the performance of different restarted methods in the given setting. To this end, we first evaluate the dependence of the new starting vector from the last built Krylov subspace: crucial information is the distance between the starting vectors of two subsequent restarts. We show that the presence of possibly small (but not tiny) singular values of A together with a dominant component of b onto the corresponding left singular space may strongly influence the selection of the new starting vector at restart time, possibly leading to lack of convergence of the restarted process.

We will theoretically and experimentally show that restarted GMRES [25] is prone to be influenced by these components and we will see that in certain cases other methods such as restarted full orthogonalization method (FOM) [22] and restarted minimum perturbation method (Minpert) [14] are less sensitive.

Then we show how the restarting vector influences the procedures after restarting. We present some simple but insightful relations that highlight the dependence of the restarted schemes on the condition number of the generated nonorthogonal basis. Moreover, new closed forms are given for the restarted solutions of GMRES and FOM which allow us to explicitly relate the approximations of the restarted and unrestarted schemes.

The main conclusions from our theoretical and experimental results are summarized below, although a thorough computationally oriented analysis still needs to be done:

- In order to prevent stagnation in the restarting phase, the quantity that is minimized by the method (in GMRES, the residual norm) should be appreciably lower than its starting value;
- A new restarting provides improvement in the approximation when the generated basis is linear independent with respect to the previous basis, unless ill-conditioning of the generated matrices deteriorates the performance;
- For GMRES and FOM, restarting effects can be monitored at run time so that dynamic restarting could be implemented.

Krylov subspace methods built on singular value information have been proposed in the past [16, 17]. However, in these articles, as well as in the analysis of full GMRES in [4], the coefficient matrix was taken to be almost singular. Under this assumption, different considerations such as the existence of the approximate solution need be taken into account. In our analysis crucial singular values have a gap of just a few orders of magnitude from the rest of the singular value set and the matrix is far from singular. This setting was found to be particularly interesting in our earlier experience [14, 26]; however, the results of this paper remain valid in general and without further assumptions.

The paper is organized as follows. In section 2 we introduce the notion of Krylov subspace method and we set the notation that will be used throughout the paper. In section 3 we describe the quantities we use to monitor the behavior of the methods that will be presented in the subsequent sections: GMRES, FOM, and Minpert. As a particular case, explicit bounds are given in the appendix for the case in which the right-hand side b is the left singular vector corresponding to the smallest singular value of the coefficient matrix. In section 6 and its subsections, results concerning the restarted methods are given, and considerations on the behavior of the restarted schemes follow from the results of the previous sections. Numerical experiments validating the theoretical results are summarized in section 7, while further comments and conclusions are given in section 8.

We would like to mention that whereas the asymptotic behavior of Krylov subspace methods has been deeply investigated, the analysis of their restarted counter-

part has only been recently addressed; see, for instance, [13, 12, 7, 23]. The fact that restarted approaches may lose significant information is a well-known fact [23]; much recent effort has been devoted to the analysis of ways to restore the lost information [7, 18, 20, 5]. In section 8 we shall indicate how our analysis fits in this framework. It is also important to remark that in our work we assume that the convergence difficulties are primarily due to the effect of restarting and that the full method would converge in a reasonable (much smaller than n) number of iterations. This assumption is crucial, since it is well known [11] that full GMRES, for instance, may show extremely slow convergence irrespectively of the eigenvalue distribution of the coefficient matrix.

2. Preliminaries.

2.1. Restarted Krylov subspace methods. Krylov subspace solvers for linear systems are iterative techniques based on the construction of the Krylov subspace $K_m(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$, where $r_0 = b - Ax_0$ is the residual associated with a starting approximate solution x_0. An approximate solution $x_m \in x_0 + K_m(A, r_0)$ is determined by imposing additional conditions on the residual $r_m = b - Ax_m$. From now on and without loss of generality we assume $\|b\| = 1$ and $x_0 = 0$ so that $r_0 = b$. The method stops when a convergence criterion is satisfied, usually an inequality involving the residual norm.

In this paper we shall focus on methods that construct an orthogonal basis V_m of $K_m(A, r_0)$. Using the Arnoldi process [1], V_m satisfies

$$AV_m = V_{m+1}H_m, \qquad V_m = [v_1, \ldots, v_m] \in \mathbb{R}^{n\times m}, \quad V_m^\top V_m = I, \tag{2.1}$$

and $H_m = (h_{i,j})_{i=1,m+1;j=1,m}$ upper Hessenberg;'$\top$' indicates real transposition. We also use the square matrix

$$\overline{H}_m = [I, 0]\, H_m \in \mathbb{R}^{m\times m}$$

so that we can write $AV_m = V_m\overline{H}_m + h_{m+1,m}v_{m+1}e_m^\top$. Another important matrix in our analysis is the matrix $\widehat{H}_m = [0, I]H_m$ equivalently defined by

$$H_m = \begin{bmatrix} h^\top \\ \widehat{H}_m \end{bmatrix}, \quad h \in \mathbb{R}^m, \tag{2.2}$$

where $h^\top$ is the first row of H_m. We will omit the dimension subscript on h since it is usually clear from the context. We shall always assume that the triangular matrix $\widehat{H}_m$ is nonsingular.

Using the computed basis, the approximate solution is written as $x_m = V_m y_m$, where y_m is determined by the solution of a smaller problem of size m; more details on the specific methods will be given in the next sections. Due to the high cost required for the orthogonalization process, the scheme might not be continued for an m large enough that would meet the requested stopping criterion. Therefore, the method is usually stopped for a reasonably small m and then restarted by constructing the subspace $K_m(A, r_m)$, $r_m = b - Ax_m$, while the current approximation x_m becomes the starting approximation for the next phase. As a side effect, however, the restarted process is not always ensured to converge. We are interested in analyzing the dependence of the restarted approach on the restarting vector and on the conditioning of matrices that are built during the process.

2.2. Notation. MATLAB notation for matrices and vectors will be used; $(u)_i$ denotes the ith component of the vector u, while u_m indicates the mth term in a vector sequence.

We shall denote by (w_i, σ_i, u_i) the singular triplets of A, so that $Aw_i = \sigma_i u_i$ and $A^\top u_i = \sigma_i w_i$, for $i = 1, \ldots, n$ [9]. Throughout the paper we assume that the coefficient matrix is scaled so that $\sigma_1 = \mathcal{O}(1)$; therefore, we shall only refer to the size of the smallest singular values. We shall use the 2-norm for vectors and the induced norm for matrices; $(w_i(X), \sigma_i(X), u_i(X))$ will indicate the ith singular triplet of the matrix X. The condition number of a full rank rectangular matrix $V \in \mathbb{R}^{n\times m}$ with $m \le n$ is defined as $\kappa(V) = \sigma_1(V)/\sigma_m(V)$. $X^\dagger$ will indicate the pseudoinverse of a tall rectangular matrix X, $X^\dagger = (X^\top X)^{-1}X^\top$. The vector e_i will denote the ith column of the identity matrix I, whose dimension will be clear from the context. Moreover, $\cos\theta(x, y)$ will denote the cosine of the angle between the two vectors x and y; the dependence on x, y will be omitted when clear from the context. Exact arithmetic will be assumed throughout the paper. In the following lemma we recall some known facts.

LEMMA 2.1 (see [9]). *Let $L = \hat{L} + E \in \mathbb{R}^{m\times m}$. Let (w_i, σ_i, u_i) and $(\hat{w}_i, \hat{\sigma}_i, \hat{u}_i)$ be the singular triplets of L and $\hat{L}$, respectively. Then*

(1) *For each $k = 1, \ldots, m$, $|\sigma_k - \hat{\sigma}_k| \le \|E\|$;*

(2) *Let $\gamma_i = \min_{k\neq i} |\sigma_i - \sigma_k|$. The singular vectors satisfy* [2]

$$\max\left(\sin\theta(u_i, \hat{u}_i), \sin\theta(w_i, \hat{w}_i)\right) \le \frac{\|E\|}{\gamma_i - \|E\|}.$$

We shall always assume that γ_i is such that the upper bound only depends on $\|E\|$.

3. The composition of the residual. The residual in the Krylov subspace of dimension m can be written as

$$r_m = -V_{m+1}[-e_1, H_m]\begin{bmatrix} 1 \\ y_m \end{bmatrix} = v_1(1 - h^\top y_m) - [v_2, \ldots, v_{m+1}]\widehat{H}_m y_m, \tag{3.1}$$

where y_m is the solution obtained by the method of choice in the Krylov subspace. If $r_m = v_1$, then the subspace $K_m(A, r_m) \equiv K_m(A, v_1)$ is generated after restart, and the restarting phase is locked. It is nearly locked when $r_m \approx v_1$, that corresponds to $|1 - h^\top y_m| \gg \|\widehat{H}_m y_m\|$. It might well happen that $K_m(A, r_m) \approx K_m(A, v_1)$ even though r_m is very different from v_1; however, we shall focus on the case in which the closeness of the two spaces is highly influenced by the choice of the restarting vector.

In the next sections, for each method analyzed we investigate the composition of the associated residual in terms of the vector basis, and in particular we give expressions for $v_1^\top r_m$. We will show that $\|h\|$ and $\|\widehat{H}_m^{-1}\|$ are key quantities in such analysis. More precisely, for the GMRES residual we will show that

$$\cos\theta(v_1, r_m) = \frac{1}{\sqrt{1 + \|\widehat{H}_m^{-\top} h\|^2}} \quad \forall m > 0,$$

while for the Minpert residual

$$|v_1^\top r_m| \le \sigma_{min}(\widehat{H}_m)^2 \quad \forall m > 0.$$

Considering that $\|\widehat{H}_m^{-1} h\| \le \|\widehat{H}_m^{-1}\|\,\|h\|$, the projection of the GMRES residual onto v_1 will not decrease for m such that the product $\|\widehat{H}_m^{-1}\|\,\|h\|$ is small. Our interest is

in the case in which a small value of $\|h\|$ is given (from the properties of the linear system, for instance; see below). In such a case, the projection will depend on the magnitude of $\|\widehat{H}_m^{-1}\|$ as m increases. Since $\|\widehat{H}_k^{-1}\| \le \|\widehat{H}_{k+1}^{-1}\|$ for any $k > 0$, a value of m large enough such that $\|\widehat{H}_m^{-1}\|\,\|h\|$ is sufficiently large can be determined.

The magnitude of $h^\top := v_1^\top A V_m$ may depend on different factors; moreover, a small $\|h\|$ may arise at later restarts. We shall focus mostly on the case where $\|h\|$ is small due to the presence of small singular values in the coefficient matrix and due to a large component of b onto the corresponding left singular vectors. When $b = u_n$, then $v_1 = b$ implies $h^\top = \sigma_n w_n^\top V_m$ and $\|h\| \le \sigma_n$ for any $m > 0$. A more likely situation is the one in which $b = \sum_i \beta_i u_i$ with $|\beta_n| \gg |\beta_i|$, for $i < n$, and we again assume that $\|b\| = 1$ so that $\sum_i \beta_i^2 = 1$. In such a case the projection of b onto the smallest left singular vector is less than one but dominant, with respect to the projection onto the rest of the space. Setting $v_1 = b$, we obtain $h^\top = \sigma_n \beta_n w_n^\top V_m + \sum_{i \ne n} \sigma_i \beta_i w_i^\top V_m$ so that

$$\|h\| \le \sigma_n |\beta_n| + \sigma_1 \sqrt{1 - \beta_n^2} \quad \forall m > 0.$$

Hence, when σ_n is small and $|\beta_n|$ is much larger than the other β_i's, so as to make the first addend dominant in the right-hand side of the inequality above, we will still have $\|h\| \ll 1$. Note that the assumption that $\sigma_n \ll \sigma_i$ is crucial for assessing the magnitude of $\|h\|$. Moreover, even if β_n is not dominant when starting the process, a large component of the first basis vector v_1 onto u_n may appear at later restarts (cf. Example 1 in section 7).

The discussion above easily generalizes to the case of a cluster of small singular values (cf. Example 2 in section 7).

4. The GMRES approximate solution. In the GMRES algorithm, the solution vector y_m is determined so as to solve the least squares problem [25]

$$\min_{y \in \mathbb{R}^m} \|\beta_0 e_1 - H_m y\|, \tag{4.1}$$

where in our case $\beta_0 = \|r_0\| = 1$. In what follows we shall omit β_0 when it corresponds to a unit starting residual norm. The GMRES residual satisfies the Petrov–Galerkin condition

$$r_m^G \perp A K_m(A, r_0). \tag{4.2}$$

We next show that the projection of the residual onto the first basis vector is equal to the residual norm.

PROPOSITION 4.1. *For any $m > 0$,*

$$\cos^2 \theta \left(r_m^G, v_1\right) = \|r_m^G\|^2 = \frac{1}{1 + \|\widehat{H}_m^{-\top} h\|^2}.$$

Proof. Using (3.1), we first show that

$$v_1^\top r_m^G = \frac{1}{1 + \|\widehat{H}_m^{-\top} h\|^2}, \qquad [v_2, \cdots, v_{m+1}]^\top r_m^G = -\frac{1}{1 + \|\widehat{H}_m^{-\top} h\|^2} \widehat{H}_m^{-\top} h. \tag{4.3}$$

The GMRES solution is $y_m^G = (H_m^\top H_m)^{-1} H_m^\top e_1$. Writing $H_m^\top H_m = \widehat{H}_m^\top \widehat{H}_m + h h^\top$ and using the Sherman–Morrison formula we get $y_m^G = (1 + \|\widehat{H}_m^{-\top} h\|^2)^{-1} (\widehat{H}_m^\top \widehat{H}_m)^{-1} h$. The formulas follow from substituting y_m^G in $h^\top y_m^G$ and in $\widehat{H}_m y_m^G$ in (3.1).

Let $\alpha = \|\widehat{H}_m^{-\top} h\|^2$. Then using (4.3)

$$\|r_m^G\|^2 = (1 - h^\top y_m^G)^2 + \|\widehat{H}_m y_m^G\|^2 = \frac{1}{(1+\alpha)^2} + \frac{\alpha}{(1+\alpha)^2} = \frac{1}{1+\alpha}.$$

The relation for $\cos\theta\left(r_m^G, v_1\right)$ follows from combining the result above with (4.3). ∎

The relation for $\cos\theta\left(r_m^G, v_1\right)$ is somehow unexpected. Restarting is usually carried out when the residual norm does not decrease sufficiently fast. Proposition 4.1 shows that if the residual norm has not decreased, then the projection of the residual vector onto the first basis vector v_1 is large, and restarting will produce little new information.

From a practical point of view, this implies that monitoring the residual norm decrease is also fundamental for restarting purposes. If the residual norm has not decreased sufficiently before restarting with the new GMRES direction vector, a different restarting vector should be selected; this alternative is explored in Example 2. Similar strategies were studied in [26].

The following proposition gives a sufficient condition, in terms of the singular value decomposition of $\widehat{H}_m^{-\top}$, for ensuring that restarting with r_m^G will not lead to stagnation.

PROPOSITION 4.2. *Let $(\hat{w}_m, \hat{\sigma}_m, \hat{u}_m)$ be the smallest singular triplet of $\widehat{H}_m = [0, I]H_m$ and let $\xi_m = \cos(\hat{w}_m, h)$. Given $\varepsilon > 0$, if for m large enough $|\xi_m|\hat{\sigma}_m^{-1}\|h\| \geq \frac{1}{\sqrt{\varepsilon}}$, then the GMRES residual r_m^G satisfies*

$$v_1^\top r_m^G \leq \frac{\varepsilon}{1+\varepsilon}. \tag{4.4}$$

Proof. Due to (3.1) and (4.3), the inequality (4.4) corresponds to writing

$$\frac{1}{1 + \|\widehat{H}_m^{-\top} h\|^2} \leq \frac{\varepsilon}{1+\varepsilon} \quad \text{or, equivalently,} \quad \frac{1}{\varepsilon} \leq \|\widehat{H}_m^{-\top} h\|^2.$$

Let $(\hat{w}_i, \hat{\sigma}_i, \hat{u}_i)$ be the singular triplets of $\widehat{H}_m$. Then

$$\|\widehat{H}_m^{-\top} h\|^2 = \sum_{i=1}^{m} \frac{(\hat{w}_i^\top h)^2}{\hat{\sigma}_i^2} \geq \frac{(\hat{w}_m^\top h)^2}{\hat{\sigma}_m^2} = \xi_m^2 \frac{\|h\|^2}{\hat{\sigma}_m^2}.$$

Therefore, if $\hat{\sigma}_m^2 \leq \varepsilon \xi_m^2 \|h\|^2$, it follows that $\|\widehat{H}_m^{-\top} h\|^2 \geq \xi_m^2 \hat{\sigma}_m^{-2} \|h\|^2 \geq \frac{1}{\varepsilon}$. ∎

Roughly speaking, Proposition 4.2 shows how much the product $\|\widehat{H}_m^{-1}\|\,\|h\|$ influences the projection of the residual onto v_1. For m such that the product $\hat{\sigma}_m^{-1}\|h\|$ is less than one, the bound in (4.4) may be close to one. Such a situation is reported in Figure 4.1, for the first example in section 7 with right-hand side $b = u_n(A)$ so that $\|h\| \leq \sigma_n$. In the figure, we show the dependence of the residual on $\hat{\sigma}_m^{-1} \equiv \|\widehat{H}_m^{-1}\|$: the GMRES residual starts decreasing when m is large enough so that $\hat{\sigma}_m^{-1} \geq \sigma_n^{-1}$.

We next provide an estimate of the distance between the approximate solution at step m and the (zero) starting approximate solution. In order to do so, we define the triangular matrix $L_m := [-e_1, H_m]$. The matrix L_m is singular if and only if $\widehat{H}_m$ is singular and satisfies [14]

$$[A, -b]\left[e_{n+1}, \begin{bmatrix} V_m \\ 0 \end{bmatrix}\right] = V_{m+1} L_m \tag{4.5}$$

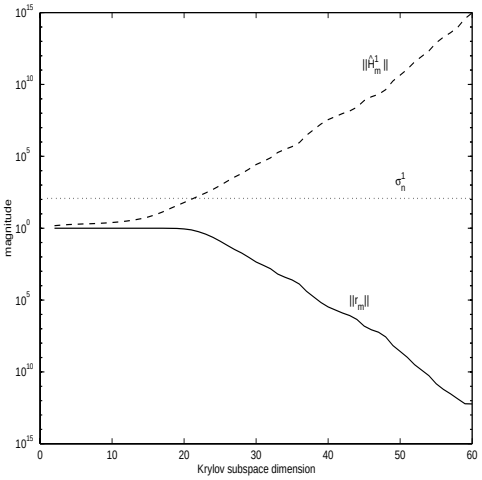


FIG. 4.1. *Example* 1. *Convergence history of GMRES (solid line) as the Krylov subspace dimension increases. Also plotted are* $\sigma_{min}(A)^{-1}$ *and* $\|\widehat{H}_m^{-1}\|$ *as* m *increases.*

so that $\sigma_{min}(L_m) \to 0$ as $m \to n$. Of interest is the fact that L_m can be written as perturbation of a triangular matrix with one known singular triplet, that is

$$L_m = \begin{bmatrix} -1 & 0 \\ 0 & \widehat{H}_m \end{bmatrix} + \begin{bmatrix} 0 & h^\top \\ 0 & 0 \end{bmatrix} =: \hat{L} + E, \tag{4.6}$$

where $\|E\| = \|h\|$ and observe that $(e_1, 1, -e_1)$ is a singular triplet of $\hat{L}$. From Lemma 2.1 there exists k such that $|\sigma_k(L_m) - 1| \le \|E\|$. Using the next proposition, we can infer that the case $k = m + 1$ corresponds to almost stagnation of GMRES, when $\|h\|$ is small.

PROPOSITION 4.3. *Let* x_m^G *be the GMRES approximate solution in* $K_m(A, r_0)$. *Then*

$$\cos\theta\left([x_m^G; 1], e_{n+1}\right) \ge \frac{\sigma_{min}(L_m)}{\|r_m^G\|}.$$

Proof. Since $\|[x_m^G; 1]\| = \|[1; y_m^G]\|$, $\cos\theta = (1+\|y_m^G\|^2)^{-1/2}$. The GMRES problem can be formulated as the augmented system

$$\begin{bmatrix} L_m^\top L_m & e_1 \\ e_1^\top & 0 \end{bmatrix} \begin{bmatrix} t \\ \eta \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

from which it follows that

$$\begin{bmatrix} 1 \\ y_m^G \end{bmatrix} = \frac{1}{\|L_m^{-\top} e_1\|^2} L_m^{-1} L_m^{-\top} e_1 \quad \text{so that} \quad \sqrt{1 + \|y_m^G\|^2} \le \frac{\|L_m^{-1}\|}{\|L_m^{-\top} e_1\|}.$$

The result follows from noticing that $\|L_m^{-\top} e_1\|^2 = 1 + \|\widehat{H}_m^{-\top} h\|^2$ and from using Proposition 4.1. ∎

Although $\sigma_{min}(L_m)$ is always not greater than $\|r_m^G\|$ [14], the bound in Proposition 4.3 shows that there must be a gap between $\sigma_{min}(L_m)$ and $\|r_m^G\|$ in order to improve the approximate solution x_m^G.

5. Two other approaches.

5.1. The full orthogonalization method (FOM) approximate solution. In the full orthogonalization method (FOM) [23] the approximate solution $y_m \in \mathbb{R}^m$ solves the linear system $\overline{H}_m y = \beta_0 e_1$. This is equivalent to imposing the following Galerkin condition on the residual [23]:

$$r_m^F \perp K_m(A, r_0). \tag{5.1}$$

Such a condition should be compared to the GMRES Petrov–Galerkin condition in (4.2). The vector y_m is well defined for $\overline{H}_m$ nonsingular; the singularity of $\overline{H}_m$ corresponds to exact stagnation of GMRES. A detailed matrix analysis of (full) FOM and GMRES has been done in [3, 6]; see also [28, 10].

In a restarted context, the two methods may differ substantially, because the Galerkin property ensures that $r_m^F \perp v_1$ so that at each restart a completely new direction vector is taken. However, lack of stagnation in the restarting phase of FOM does not necessarily imply convergence of the restarted process; as a result, the residual norm of restarted FOM may oscillate endlessly. In section 6.2 we analyze the quantities that may influence the degradation of the restarting process.

5.2. The minimum perturbation approximate solution. In [14] a new Krylov subspace method called Minpert was developed. Borrowing the idea from the total least squares theory, the approximate solution $x_m^M \in x_0 + K_m(A, r_0)$ is computed so as to solve the following problem:

$$\min_{x_m \in x_0 + K_m(A,r_0)} \|[\Delta_A, \Delta_b]\|_F, \quad \text{subject to} \quad (A - \Delta_A)x_m = b + \Delta_b. \tag{5.2}$$

Here $\|\cdot\|_F$ indicates the Frobenius norm. Therefore, x_m^M is the closest approximation in $x_0 + K_m(A, r_0)$ in a backward error sense. We shall again assume $x_0 = 0$. The problem can be formulated as follows [14, 26]:

$$\min_{\substack{z \in \mathcal{G}_m \\ \|z\|=1}} \|[A, -b]z\|, \quad \mathcal{G}_m = \text{span}\left\{ e_{n+1}, \begin{bmatrix} V_m \\ 0 \end{bmatrix} \right\}.$$

The approximate solution is then computed as $x_m^M = (z)_{1:n}/(z)_{n+1}$. Using (4.5), the problem transforms into

$$\min_{\substack{w \in \mathbb{R}^{m+1} \\ \|w\|=1}} \|L_m w\|$$

so that

$$y_m^M = \frac{1}{(w_{m+1})_1}(w_{m+1})_{2:m+1}, \quad x_m^M = V_m y_m^M, \quad r_m^M = \frac{\sigma_{m+1}}{(w_{m+1})_1} V_{m+1} u_{m+1}, \tag{5.3}$$

where $(w_{m+1}, \sigma_{m+1}, u_{m+1})$ is the smallest singular triplet of L_m; the achieved constrained minimum is $\|[\Delta_A, \Delta_b]\|_F = \sigma_{min}(L_m)$. The procedure is then restarted by generating the Krylov subspace $K_m(A, r_m^M)$. It was shown in [14, 26] that in some problems this approach may be more effective than restarted GMRES.

A necessary condition for the approximate solution to exist is that $(w_{m+1})_1 \neq 0$. On the other hand, if $w_{m+1} \equiv e_1$, then the solution exists but the method stagnates, since $y_m^M = 0$. If $h = 0$ the first row of L_m is $e_1^\top$; therefore, the Minpert solution

may be either zero or undefined, depending on whether $w_{m+1} = e_1$ or $w_{m+1} \neq e_1$, respectively [26].

The case $0 < \|h\| \ll 1$ provides a less pessimistic picture. We start by showing that exactly as in GMRES (cf. Proposition 4.1), the quantity $v_1^\top r_m^M$ depends on the value which is actually minimized by the method.

PROPOSITION 5.1. *For any $m > 0$, $v_1^\top r_m^M = \sigma_{min}(L_m)^2$. Moreover, given $\varepsilon > 0$, if $\sigma_m(\widehat{H}_m) \leq \sqrt{\varepsilon}$, then $v_1^\top r_m^M \leq \varepsilon$.*

Proof. Let (w, σ, u) be the smallest singular triplet of L_m. From (3.1) we have $v_1^\top r_m^M = 1 - h^\top y_m^M$. The solution y_m^M is written as in (5.3); therefore, we have

$$[-e_1, H_m] \begin{bmatrix} 1 \\ y_m^M \end{bmatrix} = \frac{\sigma}{(w)_1} u$$

so that $H_m y_m^M = e_1 + \frac{\sigma}{(w)_1} u$ and $h^\top y_m^M = 1 + \frac{\sigma}{(w)_1}(u)_1$. From $L_m^\top u = \sigma w$ we also have $-(u)_1 = \sigma(w)_1$ from which $h^\top y_m^M = 1 - \sigma^2$. Moreover, we have $\sigma_{m+1}(L_m)^2 \leq \sigma_m(\widehat{H}_m)^2 \leq \varepsilon$ from which the second bound follows. ∎

Referring to Figure 4.1 and comparing to Proposition 4.2 for GMRES, the Minpert method may compute a better restarting vector than GMRES for $m < 20$, since in such a case, $\sigma_n \leq \sigma_m(\widehat{H}_m) < 1$.

In the next proposition we evaluate the angle between the Minpert residual and the first basis vector v_1. We show that if $\sigma_{min}(L_m)$ is sufficiently smaller than one, then restarting with r_m^M will not lead to stagnation.

PROPOSITION 5.2. *Let $L_m = \hat{L}_m + E$ and k be such that $|\sigma_k(L_m) - 1| \leq \|E\| \equiv \|h\|$. Let r_m^M be the Minpert residual. If $k < m + 1$, then*

$$\left|\cos\theta\left(v_1, r_m^M\right)\right| \leq \frac{\delta}{\sqrt{1-\delta^2}} \qquad \textit{with} \quad \delta = \mathcal{O}(\|h\|).$$

Proof. Let (w, σ, u) be the smallest singular triplet of L_m and note that $|\cos\theta| = |(u)_1|$. Using (4.6), Lemma 2.1 ensures that there exists k such that the left singular vector u_k of L_m satisfies

$$\cos^2\theta(u_k, e_1) \equiv |(u_k)_1|^2 \geq 1 - \frac{\|E\|^2}{(\gamma_k - \|E\|)^2}, \qquad \|E\| = \|h\|. \tag{5.4}$$

Therefore, we can write $u_k = \sqrt{1-\delta^2} e_1 + \delta q$, $\|q\| = 1, e_1^\top q = 0$. For $k < m + 1$, $0 = u_k^\top u = \sqrt{1-\delta^2} e_1^\top u + \delta q^\top u$ so that $|(u)_1| = \delta(1-\delta^2)^{-1/2}|q^\top u| \leq \delta(1-\delta^2)^{-1/2}$. ∎

In [14] a detailed analysis of Minpert was provided and comparison results with GMRES were proved. Here we show that the Minpert approximate solution can also be related to the FOM approximate solution.

PROPOSITION 5.3. *Let $y_m^M = (H_m^\top H_m - \sigma^2 I)^{-1} H_m^\top e_1$ and let $y_m^F = \overline{H}_m^{-1} e_1$ be the Minpert and FOM solutions, respectively, with $\sigma = \sigma_{m+1}(L_m)$. Then*

$$\frac{\|y_m^M - y_m^F\|}{\|y_m^M\|} \leq \frac{\max(\sigma^2, |h_{m+1,m}^2 - \sigma^2|)}{\sigma_{min}(\overline{H}_m)^2},$$

where $h_{m+1,m} = e_{m+1}^\top H_m e_m$.

Proof. We note that $H_m^\top (e_1)_{1:m+1} = \overline{H}_m^\top (e_1)_{1:m}$. Therefore,

$$(H_m^\top H_m - \sigma^2 I) y_m^M = \overline{H}_m^\top \overline{H}_m y_m^F,$$

and $\overline{H}_m^\top \overline{H}_m (y_m^M - y_m^F) = (\sigma^2 I - h_{m+1,m}^2 e_m e_m^\top) y_m^M$, from which the result follows. ∎

The bound of Proposition 5.3 suggests that the two approximate solutions will be very different when $\sigma_{min}(\overline{H}_m)$ is much less than 1, that is, when the FOM solution is ill-conditioned.

6. Restarting the process. In this section we show how the selection of the new direction influences the performance of the restarted process. To simplify the presentation we shall only work on the first restart, although analogous considerations can be derived for later restarts. All variables that change with restart will be equipped with a superscript: $V_m^{(k)}$ indicates the Krylov orthogonal basis at the kth restart; $k = 0$ corresponds to the very first Arnoldi process (before the first restart). To avoid indexing overwhelming, we omit the superscript that qualifies the method when restarting is also indicated.

All methods we have introduced determine the residual after m iterations as $r_m = V_{m+1}g$ for some $g \in \mathbb{R}^{m+1}$ and then generate a new Krylov subspace $K_m(A, r_m)$. The subspace generated after one restart is $\text{span}\{v_1, \ldots, v_m, r_m, Ar_m, \ldots, A^{m-1}r_m\}$ and we have

$$\text{span}\{v_1, \ldots, v_m, r_m, Ar_m, \ldots, A^{m-1}r_m\} \subseteq \text{span}\{v_1, \ldots, v_m, \ldots, v_{2m}\}, \tag{6.1}$$

where $\text{span}\{v_1, \ldots, v_m, \ldots, v_{2m}\}$ is the subspace of dimension $2m$ generated by the method with the same starting vector without restarting. Note that the first m orthogonal vectors are the same, and that the two subspaces in (6.1) coincide if $v_{m+1}^\top r_m \neq 0$, while strict inclusion appears otherwise. In exact arithmetic, restarted FOM generates the entire subspace $\text{span}\{v_1, \ldots, v_{2m}\}$, since r_m and v_{m+1} are collinear.

The effectiveness of the restarted approach will first depend on how well the entire subspace $K_{2m}(A, r_m)$ is approximated by the subspace generated with restarting. Second, it may depend on the generation of new quantities after restart, that may lead to more ill-conditioned computation than in the unrestarted process.

Here and in what follows we use the term *global subspace* to indicate the subspace generated as a sum of Krylov subspaces in the restarting process. A *complete subspace* will instead denote the subspace generated without restarting. Moreover, while we can assume that the dimension of a complete subspace coincides with the number of processed Arnoldi vectors $\{v_1, v_2, \ldots, v_k\}$, the dimension of the global subspace will in general be less than or equal to the total number of constructed Arnoldi vectors.

6.1. Restarted GMRES. In this section we show that the updated GMRES solution after one restart can be written in closed form as the solution of a least squares problem in $\mathbb{R}^{2m}$ with a nonorthogonal basis.

We start by showing that an Arnoldi–type relation in $\mathbb{R}^{2m\times 2m}$ holds for the restarted process. The relation can be easily verified by comparing the two sides of the equality.

LEMMA 6.1. *Let $V_m^{(0)} = [v_1^{(0)}, \ldots, v_m^{(0)}]$ be the orthogonal basis for $K_m(A, b)$ and $r_m^G = V_{m+1}^{(0)}g$ be the GMRES residual in $K_m(A, b)$ with $g = e_1 - H_m^{(0)}(H_m^{(0)})^\dagger e_1$. Also let $V_m^{(1)} = [v_1^{(1)}, \ldots, v_m^{(1)}]$ be the orthogonal basis for $K_m(A, r_m^G)$. Then*

$$A[V_m^{(0)}, V_m^{(1)}] = [V_{m+1}^{(0)}, v_2^{(1)}, \ldots, v_{m+1}^{(1)}]N_{2m}, \qquad N_{2m} = \begin{bmatrix} H_m^{(0)} & \frac{g}{\|g\|}(h^{(1)})^\top \\ 0 & \widehat{H}_m^{(1)} \end{bmatrix}$$

with $(h^{(1)})^\top = e_1^\top H_m^{(1)}$ and $\widehat{H}_m^{(1)} = [0, I]H_m^{(1)}$.

Other constructions would be possible, although this definition of N_{2m} made simple the derivation of an explicit form for the restarted solution.

PROPOSITION 6.2. *Let $V_m^{(0)}$ and $V_m^{(1)}$ be as above. Then the GMRES restarted solution can be written as $x_m^{(1)} = V_m^{(0)} y_m^{(0)} + V_m^{(1)} y_m^{(1)} = [V_m^{(0)}, V_m^{(1)}]z$ with $z = N_{2m}^{\dagger} e_1$. That is, z solves the least squares problem*

$$\min_{z \in \mathbb{R}^{2m}} \|e_1 - N_{2m} z\|.$$

Proof. We have $y_m^{(0)} = (H_m^{(0)})^{\dagger} e_1$ and $y_m^{(1)} = (H_m^{(1)})^{\dagger}(\|r_m^G\| e_1)$. By direct calculation, we find

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = (N_{2m}^{\top} N_{2m})^{-1} N_{2m}^{\top} e_1 = \begin{bmatrix} ((H_m^{(0)})^{\top} H_m^{(0)})^{-1} (H_m^{(0)})^{\top} & 0 \\ ((\widehat{H}_m^{(1)})^{\top} \widehat{H}_m^{(1)} + h^{(1)}(h^{(1)})^{\top})^{-1} h^{(1)} \frac{g^{\top}}{\|g\|} & * \end{bmatrix} e_1.$$

It readily follows that $z_1 = y^{(0)}$. Noticing that $\frac{g^{\top}}{\|g\|} e_1 = (1 - (h^{(0)})^{\top} y^{(0)})/\|r_m^G\| = \|r_m^G\|$, it also follows that $z_2 = y_m^{(1)}$. ∎

COROLLARY 6.3. *Let $V_m^{(0)}$ and $V_m^{(1)}$ be as above and let $\mathcal{V}_{2m+1} = [V_{m+1}^{(0)}, v_2^{(1)}, \ldots, v_{m+1}^{(1)}]$, $S_{2m+1} = V_{2m+1}^{\top} \mathcal{V}_{2m+1}$, and $R_{2m} = V_{2m}^{\top} [V_m^{(0)}, V_m^{(1)}]$. If $\mathcal{V}_{2m+1}$ is full rank, then $N_{2m} = S_{2m+1}^{-1} H_{2m} R_{2m}$ and*

$$\|r_m^{(1)}\| \le \kappa(\mathcal{V}_{2m+1}) \, \|r_{2m}^G\|, \tag{6.2}$$

where r_{2m}^G is the GMRES residual obtained in $K_{2m}(A, b)$. Moreover,

$$\cos\theta(r_{2m}^G, r_m^{(1)}) = \frac{\|r_{2m}^G\|}{\|r_m^{(1)}\|}. \tag{6.3}$$

Proof. The restarted solution is computed so as to minimize $\|e_1 - N_{2m} z\|$. On the other hand, $\|r_m^{(1)}\| = \|\mathcal{V}_{2m+1}(e_1 - N_{2m} z)\| \le \|\mathcal{V}_{2m+1}\| \, \|e_1 - N_{2m} z\|$, and we have

$$\min_z \|e_1 - N_{2m} z\| \le \|S_{2m+1}^{-1}\| \min_z \|e_1 - H_{2m} R_{2m} z\| = \|S_{2m+1}^{-1}\| \min_y \|e_1 - H_{2m} y\|.$$

Noticing that $\|S_{2m+1}^{-1}\| \, \|\mathcal{V}_{2m+1}\| = \kappa(\mathcal{V}_{2m+1})$, the bound (6.2) follows. We have

$$\begin{aligned} (r_{2m}^G)^{\top} r_m^{(1)} &= (e_1 - H_{2m} y)^{\top} V_{2m+1}^{\top} \mathcal{V}_{2m+1} (e_1 - N_{2m} z) \\ &= (e_1 - H_{2m} y)^{\top} S_{2m+1} (e_1 - N_{2m} z) \\ &= e_1^{\top} S_{2m+1} e_1 - e_1^{\top} S_{2m+1} N_{2m} z - y^{\top} H_{2m}^{\top} S_{2m+1} e_1 + y^{\top} H_{2m}^{\top} S_{2m+1} N_{2m} z \\ &= 1 - e_1^{\top} H_{2m} R_{2m} z - y^{\top} H_{2m}^{\top} S_{2m+1} e_1 + e_1^{\top} H_{2m} R_{2m} z \\ &= 1 - y^{\top} h = \|r_{2m}^G\|^2, \end{aligned}$$

where we have used $S_{2m+1} e_1 = e_1$. The relation for $\cos\theta$ immediately follows. ∎

Note that the two matrices S_{2m} and R_{2m} only differ in the $m+1$st column.

The bound (6.2) shows that the norm of the residual after one restart may be much larger than the optimal residual norm generated in the complete subspace by full GMRES, if the basis $\mathcal{V}_{2m+1}$ is ill-conditioned. On the other hand, (6.3) says that if the global basis is well conditioned, the two residuals are very close, also in terms of direction. This shows that in this case iterating the restarted procedure with that value of m does not deteriorate the performance of the process, with respect, for instance, to restarting after $2m$ iterations. Intuitively, this result shows that if

m large enough is determined so that a good restarting vector can be constructed, then a larger value of m is unnecessary. Clearly, the problem of selecting a good m is encountered at each restarting phase, suggesting that a dynamic selection of the subspace dimension could be a good strategy to optimize the computational cost. In other words, if a maximum of m_{max} basis vectors can be stored, then $m = m_{max}$ may be necessary in just a few cases, whereas most restarting phases may need $m < m_{max}$ basis vectors in order to effectively improve the approximation while providing a good new restarting vector; this strategy would allow us to lower the computational cost of the overall process.

Bounds similar to (6.2) have been determined for other nonorthogonal quasi-minimization procedures [8, 10], such as a truncated version of GMRES [23]. A discussion on the subspace $\mathcal{V}_{2m}$, the restarting effects and relations between $r_m^{(1)}$ and r_{2m}^G, can be found in [7].

The nonorthogonal basis $[V_m^{(0)}, V_m^{(1)}]$ becomes ill-conditioned if the projection of the new starting vector $v_1^{(1)}$ onto the old space $V_m^{(0)}$ is too large. We have shown in the previous section that this may be the case when $\|h\| \ll 1$. The basis $[V_m^{(0)}, V_m^{(1)}]$ will certainly lose rank if $(v_1^{(1)})^\top v_{m+1}^{(0)} = 0$; a lower bound for $(v_1^{(1)})^\top v_{m+1}^{(0)}$ is given next.

PROPOSITION 6.4. *With the notation above, we have*

$$\cos^2\theta(v_1^{(1)}, v_{m+1}^{(0)}) \geq \frac{\sigma_{min}^2(\overline{H}_m^{(0)})}{(h_{m+1,m}^{(0)})^2 + \sigma_{min}^2(\overline{H}_m^{(0)})}.$$

Proof. Recall that $v_1^{(1)} = |r_m|/\|r_m\|$. Since all terms refer to the first Arnoldi process, we shall drop the restart superscript. Let $\bar{u}$ be the unit null vector of $H_m^\top$. Then $r_m = V_{m+1}(e_1 - H_m H_m^\dagger e_1) = (\bar{u})_1 V_{m+1}\bar{u}$ so that $|r_m|/\|r_m\| = |V_{m+1}\bar{u}|$ and $\cos^2\theta = (\bar{u})_{m+1}^2$. Moreover, $0 = H_m^\top \bar{u} = \overline{H}_m^\top(\bar{u})_{1:m} + h_{m+1,m}e_m(\bar{u})_{m+1}$. Therefore, if $(\bar{u})_{1:m} = 0$, then $\cos\theta = 1$, otherwise

$$\sigma_{min}(\overline{H}_m) = \min_{x\in\mathbb{R}^m} \frac{\|\overline{H}_m^\top x\|}{\|x\|} \leq \frac{\|\overline{H}_m^\top(\bar{u})_{1:m}\|}{\|(\bar{u})_{1:m}\|} = \frac{|(\bar{u})_{m+1}|\,|h_{m+1,m}|}{\sqrt{1-|(\bar{u})_{m+1}|^2}}.$$

Collecting all terms and comparing with $\cos\theta$, the result follows. ∎

The bound shows that ill-conditioning of the global basis may result from the ill-conditioning of $\overline{H}_m^{(0)}$, unless convergence is approached (small $|h_{m+1,m}^{(0)}|$).

6.2. Restarted FOM. In this section we show that an Arnoldi-type relation can be exploited to relate the restarted version of FOM to the complete approach.

LEMMA 6.5. *Let $V_m^{(0)} = [v_1^{(0)}, \ldots, v_m^{(0)}]$ be the Arnoldi basis for $K_m(A, b)$ and let r_m^F be the FOM residual in $K_m(A, b)$. Let also $V_m^{(1)} = [v_1^{(1)}, \ldots, v_m^{(1)}]$ be the Arnoldi basis for $K_m(A, r_m^F)$. Then span$\{[V_m^{(0)}, V_m^{(1)}]\} = K_{2m}(A, b)$, and*

$$A[V_m^{(0)}, -V_m^{(1)}] = [V_m^{(0)}, -V_{m+1}^{(1)}]M_{2m}, \tag{6.4}$$

where $(M_{2m})_{1:m+1,1:m} := H_m^{(0)}$ and $(M_{2m})_{m:2m+1,m+1:2m} := H_m^{(1)}$.

Proof. We have $AV_m^{(0)} = V_{m+1}^{(0)}H_m^{(0)}$ and $AV_m^{(1)} = V_{m+1}^{(1)}H_m^{(1)}$. Using the fact that $v_1^{(1)} = r_m^F/\|r_m^F\| = -v_{m+1}^{(0)}$, the result follows from explicitly writing $A[V_m^{(0)}, -V_m^{(1)}]$. ∎

The representation matrix M_{2m} is upper Hessenberg and has the following form:

$$M_{2m} = \begin{bmatrix} x & x & x & & & & \\ x & x & x & & & & \\ & x & x & & & & \\ & & x & x & x & x & \\ & & & x & x & x & \\ & & & & x & x & \\ & & & & & x & \end{bmatrix}.$$

This pattern highlights the fact that the restarted procedure is simply a particular way of truncating the full orthogonalization scheme. A common truncation strategy is the incomplete orthogonalization method (IOM(m)), where the corresponding matrix has banded Hessenberg form [23].

PROPOSITION 6.6. *The FOM approximate solution after one restart can be written as*

$$x_m^{(1)} = V_m^{(0)} y_m^{(0)} + V_m^{(1)} y_m^{(1)} = [V_m^{(0)}, -V_m^{(1)}] z \qquad \text{with} \qquad z = \overline{M}_{2m}^{-1} e_1,$$

where $\overline{M}_{2m} = [I, 0] M_{2m}$.

Proof. We have $y_m^{(0)} = (\overline{H}_m^{(0)})^{-1} e_1$ and $y_m^{(1)} = (\overline{H}_m^{(1)})^{-1} (\|r_m^F\| e_1)$. Note that $\|r_m^F\| = |h_{m+1,m}^{(0)} e_m^\top y_m^{(0)}|$. The system $\overline{M}_{2m} z = e_1$ can be written as

$$\begin{bmatrix} \overline{H}_m^{(0)} & 0 \\ h_{m+1,m}^{(0)} e_1 e_m^\top & \overline{H}_m^{(1)} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = e_1.$$

The first block row gives $z_1 = (\overline{H}_m^{(0)})^{-1} e_1 = y_m^{(0)}$. The second block row gives $\overline{H}_m^{(1)} z_2 = -h_{m+1,m}^{(0)} e_1 e_m^\top z_1$, so that $x_m^{(1)} = V_m^{(0)} y_m^{(0)} + V_m^{(1)} y_m^{(1)} = [V_m^{(0)}, -V_m^{(1)}] z$. ∎

The distance between the two residuals $r_m^{(1)}$ and r_{2m}^F is simply measured by the angle between the last orthogonal vectors of the bases $V_{m+1}^{(1)}$ and V_{2m+1}, that is,

$$\cos\theta \left(r_{2m}^F, r_m^{(1)} \right) = v_{2m+1}^\top v_{m+1}^{(1)},$$

measuring how far the last generated basis vector is from the latest vector of the fully orthogonal basis.

The relation with the FOM solution $x_{2m}^F = V_{2m} \overline{H}_{2m}^{-1} e_1$ can be made more explicit. We first need the following lemma, which relates the representation matrices of the restarted and unrestarted processes.

LEMMA 6.7. *Let* $V_m^{(0)} = [v_1^{(0)}, \ldots, v_m^{(0)}]$ *and* $V_m^{(1)} = [v_1^{(1)}, \ldots, v_m^{(1)}]$ *be as in Proposition* 6.5 *and let* $S_{2m} = V_{2m}^\top [V_m^{(0)}, -V_m^{(1)}]$ *be nonsingular. Then* $H_{2m} = S_{2m+1} M_{2m} S_{2m}^{-1}$ *and*

$$\overline{H}_{2m} = S_{2m} \overline{M}_{2m} S_{2m}^{-1} + h_{m+1,m}^{(1)} (S_{2m})_{2m,2m}^{-1} s e_{2m}^\top,$$

where $s = -V_{2m}^\top v_{m+1}^{(1)}$.

Proof. The relation for H_{2m} follows from comparing (2.1) and (6.4). The relation for $\overline{H}_{2m}$ follows from explicitly writing

$$\overline{H}_{2m} = [S_{2m}, s] \begin{bmatrix} \overline{M}_{2m} \\ h_{m,m+1}^{(1)} e_{2m}^\top \end{bmatrix} S_{2m}^{-1}.$$ ∎

Note that the matrices $\overline{H}_{2m}$ and $S_{2m}\overline{M}_{2m}S_{2m}^{-1}$ only differ in the last column.

PROPOSITION 6.8. *With the notation of Lemma* 6.7, *let* $q = [V_m^{(0)}, -V_m^{(1)}]\cdot \overline{M}_{2m}^{-1}S_{2m}^{-1}s$. *Let also* $\alpha_1 = h_{m+1,m}^{(1)}(z)_{2m}$ *with* $|\alpha_1| = \|r_m^{(1)}\|$, *where* $z = \overline{M}_{2m}^{-1}e_1$ *is the restarted FOM solution, and let* $\alpha_2 = 1 + h_{m+1,m}^{(1)}e_{2m}^\top \overline{M}_{2m}^{-1}S_{2m}^{-1}s$. *Then*

$$x_m^{(1)} = x_{2m}^F + \frac{\alpha_1}{\alpha_2}q, \tag{6.5}$$

$$\|x_m^{(1)} - x_{2m}^F\| \le \frac{\|r_m^{(1)}\|}{|\alpha_2|}\,\kappa([V_m^{(0)}, -V_m^{(1)}])\|\overline{M}_{2m}^{-1}\|\,\|s\|. \tag{6.6}$$

Proof. Let $\tau = h_{m+1,m}^{(1)}/(S_{2m})_{2m,2m}$ and observe that $S_{2m}e_1 = e_1$. Let us drop the subscripts and apply the Sherman–Morrison formula to $\overline{H}$ so as to obtain

$$\begin{aligned}
\overline{H}^{-1}e_1 &= S\overline{M}^{-1}S^{-1}e_1 - \tau S\overline{M}^{-1}S^{-1}s\left(1 + \tau e_{2m}^\top S\overline{M}^{-1}S^{-1}s\right)^{-1} e_{2m}^\top S\overline{M}^{-1}S^{-1}e_1 \\
&= S\overline{M}^{-1}e_1 - h_{m+1,m}^{(1)}S\overline{M}^{-1}S^{-1}s\left(1 + h_{m+1,m}^{(1)}e_{2m}^\top \overline{M}^{-1}S^{-1}s\right)^{-1} e_{2m}^\top \overline{M}^{-1}e_1 \\
&= S\overline{M}^{-1}e_1 - \frac{\alpha_1}{\alpha_2}S\overline{M}^{-1}S^{-1}s.
\end{aligned}$$

Therefore,

$$\begin{aligned}
x_{2m}^F &= V_{2m}\overline{H}^{-1}e_1 \\
&= [V_m^{(0)}, -V_m^{(1)}]\overline{M}^{-1}e_1 - \frac{\alpha_1}{\alpha_2}[V_m^{(0)}, -V_m^{(1)}]\overline{M}^{-1}S^{-1}s = x_m^{(1)} - \frac{\alpha_1}{\alpha_2}q. \qquad \square
\end{aligned}$$

The solution x_{2m}^F can be recovered from (6.5) for $S_{2m+1} = I$; in such a case indeed $s = 0$, and therefore $q = 0$. In general, $\|s\| \le 1$. A relative bound for the difference between the two residuals can be easily obtained from (6.6) as

$$\frac{\|r_m^{(1)} - r_{2m}^F\|}{\|r_m^{(1)}\|} \le \frac{\|A\|}{|\alpha_2|}\,\kappa([V_m^{(0)}, -V_m^{(1)}])\|\overline{M}_{2m}^{-1}\|\,\|s\|. \tag{6.7}$$

As in restarted GMRES, the condition number of the basis enters in the bound, whereas $\|A\|$ is around one, by assumption. While it is difficult to evaluate the role of α_2, it is clear that the estimates in (6.6) and (6.7) are influenced by the magnitude of $\|\overline{M}_{2m}^{-1}\|$, and our experiments confirmed that its magnitude strongly affects the restarted approach (cf. Example 3, section 7). Using the definition of $\overline{M}_{2m}$ we can write

$$\overline{M}_{2m}^{-1} = \begin{bmatrix} (\overline{H}_m^{(0)})^{-1} & 0 \\ -h_{m+1,m}(\overline{H}_m^{(1)})^{-1}e_1e_m^\top(\overline{H}_m^{(0)})^{-1} & (\overline{H}_m^{(1)})^{-1} \end{bmatrix}$$

so that

$$\|\overline{M}_{2m}^{-1}\| \ge \max\left\{\|(\overline{H}_m^{(0)})^{-1}\|, \|(\overline{H}_m^{(1)})^{-1}\|, |h_{m+1,m}|\,\|(\overline{H}_m^{(1)})^{-1}e_1\|\,\|e_m^\top(\overline{H}_m^{(0)})^{-1}\|\right\}.$$

Therefore, the conditioning of the restarted representation matrix $\overline{M}_{2m}$ is as large as that of each restarting $m \times m$ representation matrix; in fact, it could be as large as their product.

6.3. Restarted Minpert. As for the previous methods, an Arnoldi-like relation can be deduced for restarted Minpert. Unfortunately, due to the derivation of the solution which involves a nonlinear problem, the approximate solution cannot be written explicitly. For the sake of completeness, we report below the matrix relation similar to that proved for restarted GMRES and restarted FOM.

PROPOSITION 6.9. *Let $V_m^{(0)} = [v_1^{(0)}, \dots, v_m^{(0)}]$ be the Arnoldi basis for $K_m(A, b)$ and let r_m^M be the Minpert residual in $K_m(A, b)$. Also let $V_m^{(1)} = [v_1^{(1)}, \dots, v_m^{(1)}]$ be the Arnoldi basis for $K_m(A, r_m^M)$. Then span$\{[V_m^{(0)}, V_m^{(1)}]\} \subseteq K_{2m}(A, b)$, and*

$$A[V_m^{(0)}, V_m^{(1)}] = [V_{m+1}^{(0)}, v_2^{(1)}, \dots, v_{m+1}^{(1)}]T_{2m}, \quad T_{2m} = \begin{bmatrix} H_m^{(0)} & \varsigma u\,(h^{(1)})^\top \\ 0 & \widehat{H}_m^{(1)}, \end{bmatrix}, \tag{6.8}$$

where $u = u_{m+1}(L_m^{(0)})$ and $\varsigma = sgn((u)_1)$.

Proof. Let (w, σ, u) be the smallest singular triplet of $L_m^{(0)}$. We have $r_m^M = -\sigma/(w)_1 V_{m+1}^{(0)} u$, that is, r_m^M is a linear combination of the basis elements, so that the first assertion follows. Since $-\sigma/(w)_1 = \sigma^2/(u)_1$, using the fact that $v_1^{(1)} = r_m^M/\|r_m^M\|$, we have $v_1^{(1)} = \varsigma V_{m+1}^{(0)} u$. The remaining result follows from forming $A[V_m^{(0)}, -V_m^{(1)}]$. ∎

COROLLARY 6.10. *Let $V_m^{(0)}$ and $V_m^{(1)}$ be as above and let $\mathcal{V}_{2m+1} = [V_{m+1}^{(0)}, v_2^{(1)}, \dots, v_{m+1}^{(1)}]$, $S_{2m+1} = V_{2m+1}^\top \mathcal{V}_{2m+1}$, and $R_{2m} = V_{2m}^\top [V_m^{(0)}, V_m^{(1)}]$. If $\mathcal{V}_{2m+1}$ is full rank, then $T_{2m} = S_{2m+1}^{-1} H_{2m} R_{2m}$.*

Note that the triangular matrices S_{2m+1} and R_{2m} are not in general the same as those of restarted GMRES in Corollary 6.3.

The matrix T_{2m} in (6.8) can only be formed once the singular value problem with $L_m^{(0)}$ has been solved; therefore, a closed form for the restarted solution cannot be derived.

7. Numerical experiments. In this section we provide additional numerical evidence of the theoretical results presented in the paper. In the description below, GMRES(m) stands for restarted GMRES with maximum Krylov subspace dimension equal to m; analogous notation is used for the other methods. In all tests we set $x_0 = 0$. All tests were done with MATLAB 5.2 [15] on a Sun workstation.

Indefiniteness of the coefficient matrix is a possible additional bottleneck for Krylov subspace methods, as convergence may slow down. For this reason, we will only show experiments with positive definite matrices.

Example 1. The matrix in this test comes from the discretization via centered finite differences of the problem [14]

$$(-e^{-xy}u_x)_x + (-e^{xy}u_y)_y + 10(u_x + u_y) - 60u = f$$

with Dirichlet zero boundary conditions on the unit square. The resulting non-symmetric matrix has size $n = 100$ and smallest singular values $\sigma_{99} = 1.1982e-01, \sigma_{100} = 8.4646e-03$. The plots shown in previous sections described the performance of the methods with right-hand side $b = u_n$.

Here we report on the convergence history of the methods for different right-hand side selections:

- *Random entries.* We consider b having random entries uniformly distributed in $[0, 1]$. In Figure 7.1(left) the composition of b in terms of left singular

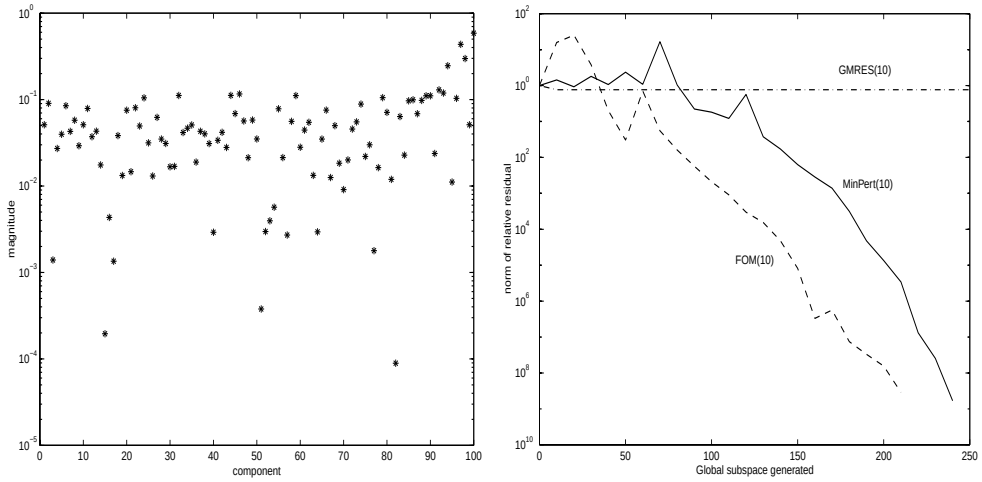


FIG. 7.1. *Example* 1. *Left: composition of the right-hand side in terms of singular vectors. Right: convergence history of the methods.*

vectors of the coefficient matrix is depicted; note that this selection provides a very large component onto the critical singular vector, that is, the one corresponding to the smallest singular value. Figure 7.1(right) reports on the performance of the methods for $m = 10$ versus the number of vectors generated in the global subspace span$\{[V_m^{(0)}, V_m^{(1)}, V_m^{(2)}, \dots]\}$ =span$\{\mathcal{V}_{km}\}$. In Figure 7.2 we report on the convergence history of the restarted approaches for different values of m; on the right of each plot, another plot depicts the dimension growth of the global subspace span$\{[V_m^{(0)}, V_m^{(1)}, V_m^{(2)}, \cdots]\}$ generated during the restarted procedure; this is measured using the MATLAB function `rank`$(\mathcal{V}_{km}, 10^{-12})$ [15]. The connection between loss of rank of the generated basis and the stagnation of the restarted process is evident in GMRES. Note also that the dimension of the global subspace in restarted FOM and restarted Minpert is much closer to the full dimension mk after $k - 1 \geq 0$ restarts, especially for $m = 10$.

- $b = u_{80}$. In this test we consider $b = u_{80}$. The convergence history of the methods for $m = 10$ is shown in Figure 7.3(left). After five restarts, both GMRES and Minpert show stagnation. Then, Minpert is able to recover while GMRES is not. The residual norm of FOM rapidly and monotonically decreases. Even if not present at the beginning, the projection onto u_{100} appears at later restarts: the projection of the restarting vector onto u_{100} is shown in Figure 7.3(right) for all methods. After five restarts, the convergence history behaves as if the right-hand side had a dominant component onto u_{100}, as in the previous case; therefore, similar considerations apply.

Concerning the first selection of a right-hand side, we remark that had one chosen a random right-hand side with entries normally distributed (MATLAB function `randn`, for instance), the projection onto u_n would not have been so large. This phenomenon should be related to the fact that the coefficient matrix is a discretization of a differential operator. The selection of the right-hand side in testing iterative solvers is an important problem that deserves a separate analysis.

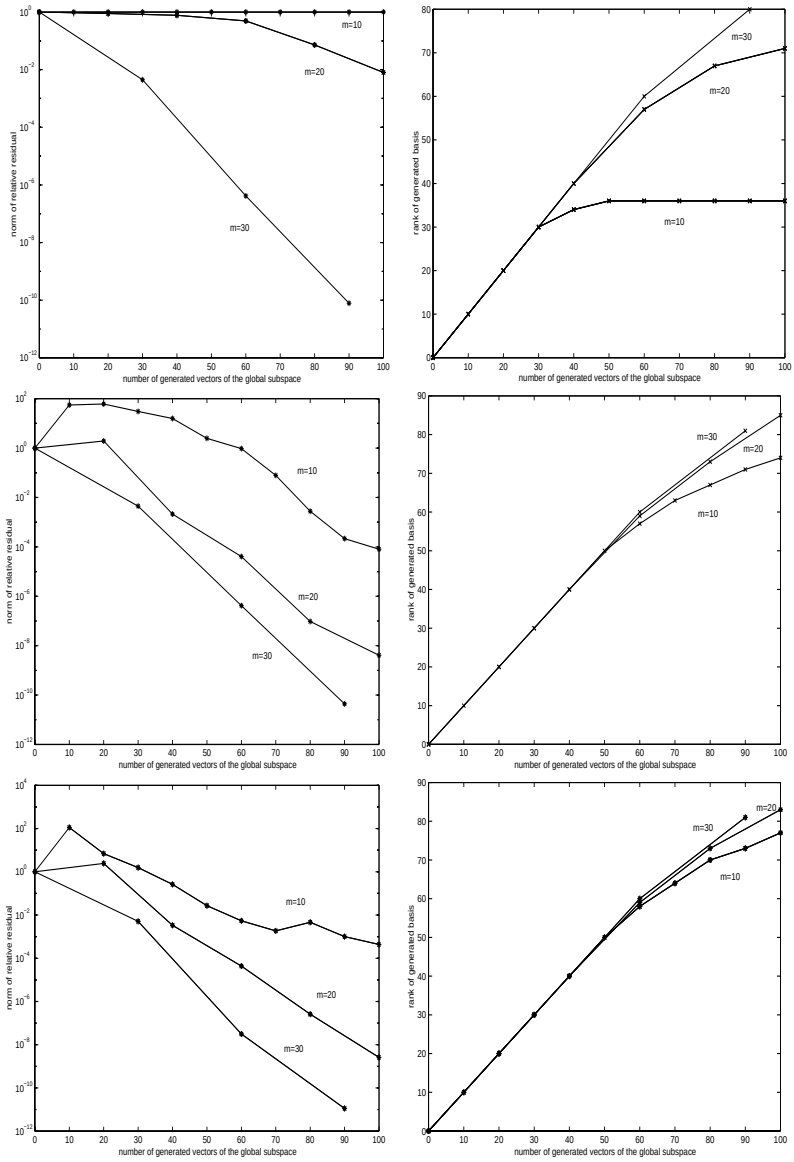


FIG. 7.2. *Example* 1. *Relation between convergence and global basis rank for restarted GMRES, Minpert, and FOM (from top to bottom, respectively). Left: Convergence history of each method for* $m = 10, 20, 30$. *Right: dimension of global subspace for* $m = 10, 20, 30$.

Example 2. We analyze the behavior of the restarted methods when A has two small singular values and the right-hand side has a large component onto the corresponding left singular space. Using the matrix A from Example 1, we define $A_1 := A - u_{99}.11w_{99}^\top$ with smallest singular values $\sigma_{99} = 9.815e-03, \sigma_{100} = 8.4646e-03$. We set $b = \tilde{b}/\|\tilde{b}\|$, where $\tilde{b} = u_{99} - u_{100}$. In Figure 7.4(left) the convergence history of the restarted approaches is reported for $m = 15$. The convergence history of GMRES(20) is also plotted. Convergence for $m = 20$ is expected: indeed, Figure 7.4(right) reports the residual norm and $\|\widehat{H}_m^{-1}\|$ as done in Figure 4.1. The plot shows that after 20

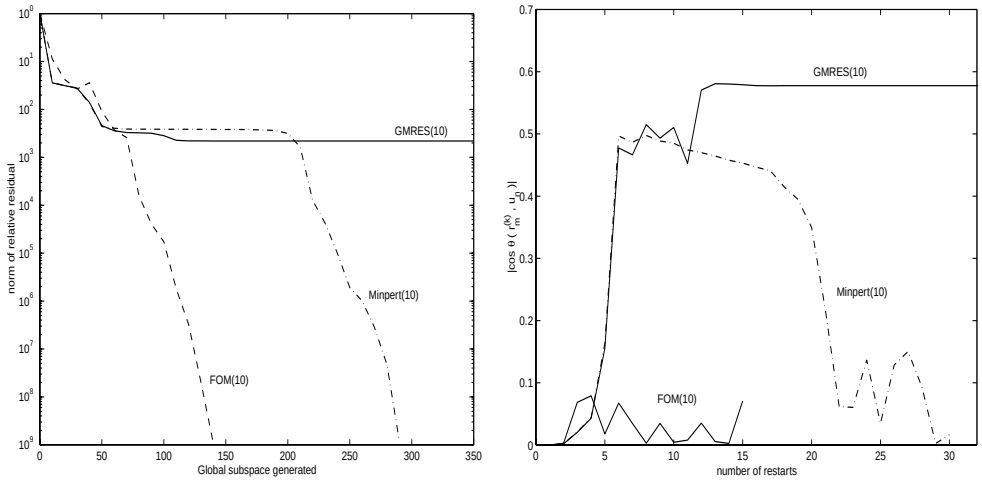


FIG. 7.3. *Example* 1. *Left: convergence history of the restarted methods for* $m = 10$ *and* $b = u_{80}$. *Right: values of* $|\cos\theta(r_m^{(k)}, u_n)|$ *at restart* k *for each method.*

iterations $\|\widehat{H}_m^{-1}\| \geq \sigma_{n-1}^{-1}$ so that, according to Proposition 4.2, the projection onto the first basis vector becomes less dominant and restarting will not lead to stagnation, at least during the first restart.

In Figure 7.5 we also show the convergence history of GMRES(15) when at the first restart the GMRES residual is replaced by the FOM residual; the next subspace is thus built with that vector as starting direction. Note that after that, GMRES(15) converges very satisfactorily. Similar considerations were done in [26]. This example shows that missing the good very first restarting direction may have disastrous effects.

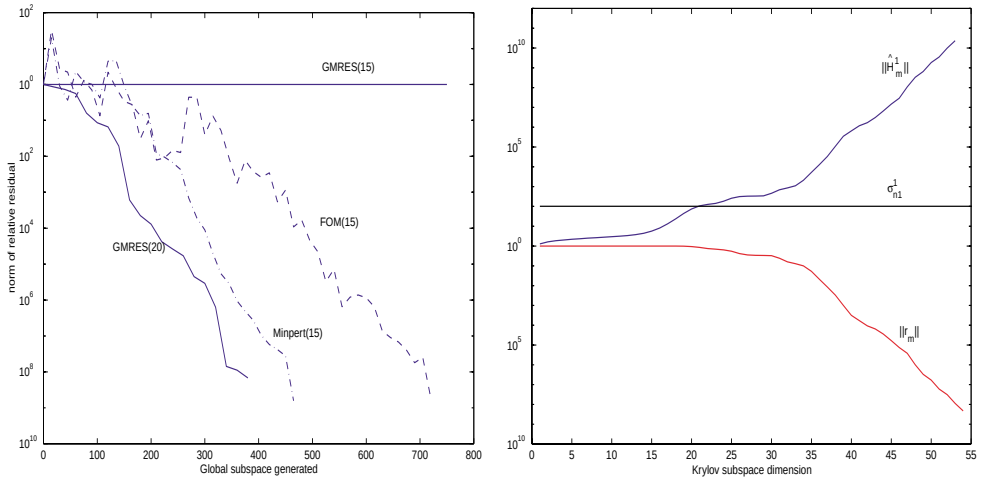


FIG. 7.4. *Example* 2. *Left: Convergence history of restarted GMRES, Minpert, and FOM. Right: Relation between GMRES residual norm and* $\|\widehat{H}_m^{-1}\|$.

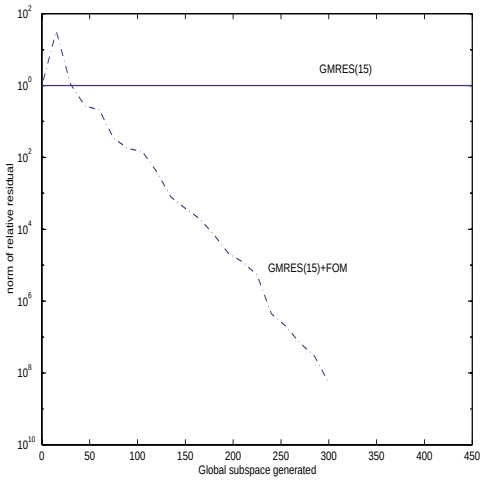


FIG. 7.5. *Example* 2. *Convergence of restarted GMRES. GMRES*(15) *stagnates whereas simply switching to FOM at the very first restart makes the algorithm converge.*

Example 3. We consider the elliptic problem associated with the operator

$$\mathcal{L}(u) = -\Delta u + 2e^{2(x^2+y^2)}u_x - 10u$$

and Dirichlet boundary conditions in $[0,1] \times [0,1]$. We consider the centered finite differences discretization that leads to the matrix $A \in \mathbb{R}^{100\times 100}$; tests were run with the matrix $\tilde{A} = A - .1I$ with smallest singular value $\sigma_{min}(\tilde{A}) = 1.208 \cdot 10^{-2}$. Figure 7.6 shows the convergence history of the restarted methods with $b = u_{100}(\tilde{A})$. The convergence curve of full FOM is also reported. The condition number $\kappa([V_m^{(0)}, V_m^{(1)}, V_m^{(2)}, \dots])$ of the global basis generated by FOM(m) and Minpert(m) grows very quickly: for restarted FOM the values of $\kappa \approx \kappa([V_m^{(0)}, V_m^{(1)}, V_m^{(2)}, \dots])$ as the number j of restarts grows are reported below.

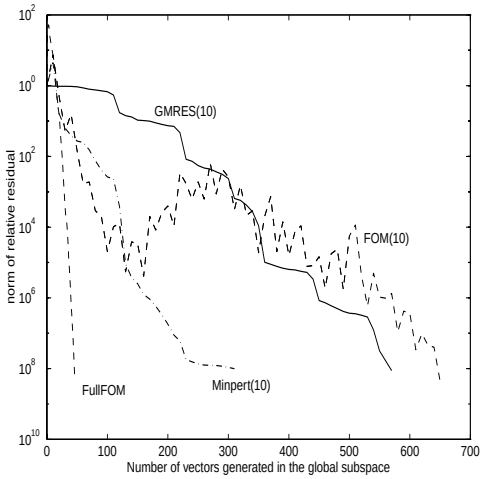


FIG. 7.6. *Example* 3. *Convergence of restarted methods for* $m = 10$ *and of full FOM.*

j	0	1	2	3	4	5
κ	1	$1 \cdot 10^1$	$3 \cdot 10^2$	$5 \cdot 10^4$	$2 \cdot 10^8$	$1 \cdot 10^{13}$

The bound in (6.7) shows that the condition number of the basis influences the distance between the unrestarted and restarted FOM processes, and the plot confirms that, on this problem, restarting has a harmful effect; indeed, full FOM converges to the required tolerance in only 46 iterations. Similar considerations hold for Minpert. We also observe that the restarted FOM residual norm oscillates for a very large number of restarts; this behavior can be explained by monitoring the magnitude of $\|\overline{M}_{km}^{-1}\|$ or, more cheaply, of $\|(\overline{H}_m^{(k)})^{-1}\|$, and recalling the discussion after Proposition 6.8. In addition to κ, also the value of $\|\overline{M}_{km}^{-1}\|$ influences the difference between the unrestarted and restarted FOM procedures. However, in order to correctly apply the result of Proposition 6.8 to this stage of the process we have to assume that the global basis has full rank. To this end, we have stopped FOM(m) after 11 restarts and then started both full FOM and restarted FOM with the obtained approximate solution $x_{10}^{(11)}$ as initial approximation. We can thus compare restarted FOM with full FOM with the same starting vector. The convergence curve of FOM(m) obviously continued the one in Figure 7.6, while unrestarted FOM with initial approximation $x_{10}^{(11)}$ converged in about 50 iterations. The basis of the global subspace generated by restarted FOM quickly became ill-conditioned, but more importantly, $\|\overline{M}_{km}^{-1}\| \approx 10^3$ after three restarts; we recall that $\|\overline{M}_{km}^{-1}\|$ will be always at least as large as the value $\|(\overline{H}_m^{(k)})^{-1}\|$ of each restarting phase. Therefore, already by simply monitoring the magnitude of $\|(\overline{H}_m^{(k)})^{-1}\|$ at each restart we may be able to predict the behavior of the restarted approach.

8. Discussion and conclusions. In this paper we have only occasionally mentioned the eigenvalues of the coefficient matrix A. In practice, however, the spectrum of A and, more precisely, the distribution of the eigenvalues, is important in the performance of restarted Krylov subspace methods. Recent papers have focused on improving the information generated by GMRES and FOM at restart time by including spectral information, in the form of eigenvalue or eigenvector approximations [18, 20, 5, 24]. Such information can indeed considerably improve the convergence, making in some cases the performance of the restarted versions comparable to that of the unrestarted ones, in terms of iterations [24, 20]. On the other hand, the improvement usually depends on the value of the restarting parameter m and on the amount of information that is kept at later restarts. Some tuning is therefore necessary in order to attain the expected improvement.

Often problems are due to the presence of eigenvalues that are clustered and close to the origin. Unfortunately, unless a priori information is available, these eigenvalues may be difficult to detect. However, such clusters are sometimes associated with small singular values of the matrix, for which our theory applies. As an example, consider the following matrix, which is a variant of a problem in [18]:

$$A = \text{bidiag}(\underline{d}, 1), \qquad A \in \mathbb{R}^{100\times 100}$$

with diagonal $d = [.01, .02, .03, .04, 10, 11, \ldots, 105]$. The diagonal elements coincide with the eigenvalues of A. The smallest singular values of the matrix are $\sigma_{99} \approx 0.98$ and $\sigma_{100} \approx 2.3 \cdot 10^{-7}$. If one restart of all methods is carried out for $b = u_{100}$ and $m = 10$, we get the quantities listed in the following table.

	$v_1^\top r_m^{(0)}$	$\|r_m^{(0)}\|$	$\|r_m^{(1)}\|$
GMRES(10)	0.494	0.70	0.4940
Minpert(10)	5e-14	0.98	1.3e-07
FOM(10)	1e-16	0.98	3.0e-08

The behavior of the methods could be explained in terms of eigenvalues: both FOM (through Ritz values) and Minpert (through the roots of its residual polynomial [26]) are able to accurately detect the smallest eigenvalues for $m = 10$ before the first restart, whereas GMRES is not. However, the failure of the first GMRES restart is well predicted (with no a priori information on the eigenvalue distribution) by inspecting $\|(\widehat{H}_m^{(0)})^{-1}\| \, \|h^{(0)})\|$ before the first restart. Also inspired by this numerical evidence, we are currently further investigating the connection between eigenvalues and singular values.

In this paper we have highlighted the quantities that play an important role at restart time, although we believe our analysis is still far from being exhaustive. We have derived new tools for monitoring the goodness of the current GMRES and Minpert residuals as new starting vectors in a restarting phase when A has small (but not tiny) singular values. Moreover, we have given closed forms for the approximate solutions of restarted GMRES and FOM, making explicit their dependence on the sensitivity of new matrices. Numerical experiments show that restarted FOM may be superior to the other two approaches, unless ill-conditioning of the matrices generated during the restarted process becomes too harmful.

It seems that restarted GMRES is penalized on the problems analyzed in this paper, which assume a particular distribution of singular values and a corresponding selection of right-hand side. We have shown, however, that simple variants (such as restarting with a different direction vector) can naturally adjust its performance. We have mainly focused on the selection of the restarting vector; although we have not treated the topic, it would also be important to explicitly quantify the convergence delay of the restarted process, say in terms of number of iterations, with respect to the full process.

Appendix. The case $b = u_n$.

When the right-hand side is $b = u_n$ we can explicitly estimate the distance of the approximate solution given by the chosen method from the exact solution $x = \sigma_n^{-1} w_n$, in terms of $\|h\|$, $\|\widehat{H}_m^{-1}\|$. We first need the following lemma.

LEMMA A.1. *Let (w_n, σ_n, u_n) be the smallest singular triplet of A. Suppose m steps of the Arnoldi recurrence have been taken, with $AV_m = V_m \overline{H}_m + h_{m+1,m} v_{m+1} e_m^\top$ and $v_1 = u_n$. Let also $(\bar{w}_m, \bar{\sigma}_m, \bar{u}_m)$ be the smallest singular triplet of $\overline{H}_m$. Then*

$$|\cos\theta\,(w_n, V_m \bar{w}_m)\,| = \frac{\bar{\sigma}_m}{\sigma_n}\sqrt{1-\delta^2}, \qquad \delta = \mathcal{O}(\|h\|).$$

Proof. Let $\tau = \pm\sqrt{1-\delta^2}$. Writing $w_n^\top = \sigma_n^{-1} u_n^\top A$ and $\bar{u}_m = \tau e_1 + \delta q$, with $\|q\| = 1$, $e_1^\top q = 0$, we have

$$\begin{aligned} w_n^\top V_m \bar{w}_m &= \sigma_n^{-1} u_n^\top V_m \overline{H}_m \bar{w}_m + h_{m+1,m}\sigma_n^{-1} u_n^\top v_{m+1} e_m^\top \bar{w}_m \\ &= \frac{\bar{\sigma}_m}{\sigma_n} u_n^\top V_m(\tau e_1 + \delta q) + h_{m+1,m}\frac{1}{\sigma_n} u_n^\top v_{m+1} e_m^\top \bar{w}_m = \frac{\bar{\sigma}_m}{\sigma_n}\tau. \end{aligned}$$

The last equality follows from $u_n = v_1$. ∎

THEOREM A.2. *Let x be the exact solution to (1.1) for $b = u_n$ and let x_m^G, x_m^F and x_m^M be the GMRES, FOM, and Minpert approximate solutions, respectively. Let σ_n be the smallest singular value of A and $\sigma = \sigma_{min}(L_m)$, $\delta = \mathcal{O}(\|h\|)$. Then*

$$\cos\theta\left(x, x_m^G\right) = \frac{1}{\sigma_n}\frac{\|\widehat{H}_m^{-\top}h\|^2}{\|(\widehat{H}_m^\top\widehat{H}_m)^{-1}h\|},$$
$$\left|\cos\theta\left(x, x_m^F\right)\right| \geq \left|\frac{\bar{\sigma}_m}{\sigma_n} - \frac{\delta}{\sqrt{1-\delta^2}}\right|,$$
$$\left|\cos\theta\left(x, x_m^M\right)\right| \geq \frac{1}{\sigma_n}\frac{(1-\sigma^2)(\sigma_m(H_m)^2 - \sigma^2)}{\|H_m\|}.$$

Proof. For $b = u_n$, we have $w_n = x/\|x\|$, where $w_n = \sigma_n^{-1}A^\top u_n$. Moreover, $x_m^G/\|x_m^G\| = V_m y_m^G/\|y_m^G\|$. Note that

$$w_n^\top V_m y_m^G = \sigma_n^{-1}e_1^\top H_m y_m^G = \sigma_n^{-1}h^\top y_m^G$$

so that, using (4.3) and $y_m^G = (1 + \|\widehat{H}_m^{-\top}h\|^2)^{-1}(\widehat{H}_m^\top\widehat{H}_m)^{-1}h$,

$$\cos\theta\left(x, x_m^G\right) = \frac{1}{\|y_m^G\|}\frac{1}{\sigma_n}h^\top y_m^G = \frac{\|\widehat{H}_m^{-\top}h\|^2}{\sigma_n\|(\widehat{H}_m^\top\widehat{H}_m)^{-1}h\|}.$$

For the FOM approximation, let $|\tau| = \sqrt{1-\delta^2}$ and $\bar{u}_m$ be as in the proof of Lemma A.1. Then, $y_m^F = \overline{H}_m^{-1}e_1 = \tau^{-1}\overline{H}_m^{-1}(\bar{u}_m - \delta q)$, and $\|x_m^F\| = \|y_m^F\| \leq \bar{\sigma}_m^{-1}$. Using Lemma A.1,

$$|w_n^\top x_m^F| = \frac{1}{|\tau|}\left|\frac{1}{\bar{\sigma}_m}w_n^\top V_m\bar{w}_m - \delta w_n^\top V_m\overline{H}_m^{-1}q\right| = \frac{1}{|\tau|}\left|\frac{|\tau|}{\sigma_n} - \delta w_n^\top V_m\overline{H}_m^{-1}q\right|$$
$$\geq \frac{1}{|\tau|}\left|\frac{|\tau|}{\sigma_n} - \delta|w_n^\top V_m\overline{H}_m^{-1}q|\right| \geq \frac{1}{|\tau|}\left|\frac{|\tau|}{\sigma_n} - \delta\frac{1}{\bar{\sigma}_m}\right|.$$

It follows that

$$|\cos\theta\left(x, x_m^F\right)| = \frac{1}{\|x_m^F\|}|w_n^\top x_m^F| \geq \left|\frac{\bar{\sigma}_m}{\sigma_n} - \frac{\delta}{|\tau|}\right|.$$

For the Minpert approximation, we have $\cos\theta\left(x, x_m^M\right) = \sigma_n^{-1}u_n^\top AV_m y_m^M/\|y_m^M\|$ $= \sigma_n^{-1}e_1^\top H_m y_m^M/\|y_m^M\|$. Moreover, $e_1^\top H_m y_m^M = 1 - \sigma^2$ and using the closed form of the solution, $\|y_m^M\| \leq \|H_m\|/(\sigma_m(H_m)^2 - \sigma^2)$, from which the third result follows. ∎

Acknowledgments. We thank E. De Sturler for explanations on [7] and Z. Strakoš for helpful comments on an earlier version of the paper. We also thank the two anonymous referees for encouraging the revision of the presentation.

REFERENCES

[1] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
[2] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.
[3] P. N. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 58–78.

[4] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.
[5] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, J. Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.
[6] J. Cullum and A. Greenbaum, *Relations between Galerkin and norm-minimizing iterative methods for solving linear systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 223–247.
[7] E. De Sturler, *Truncation Strategies for Optimal Krylov Subspace Methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.
[8] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.
[9] G. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[10] A. Greenbaum, *Iterative methods for solving linear systems*, SIAM, Philadelphia, PA, 1997.
[11] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.
[12] Y. Huang and H. van der Vorst, *Some observations on the convergence behavior of GMRES*, Tech. Rep. 89-09, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, 1989.
[13] W. D. Joubert, *On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems*, Numer. Linear Algebra Appl., 1 (1994), pp. 427–448.
[14] E. M. Kasenally and V. Simoncini, *Analysis of a minimum perturbation algorithm for nonsymmetric linear systems*, SIAM J. Numer. Anal., 34 (1997), pp. 48–66.
[15] *MATLAB User's Guide*, The MathWorks, Inc., Natick, MA, 1998.
[16] J. Meza and W. Symes, *Deflated Krylov methods for nearly singular linear systems*, J. Optim. Theory Appl., 72 (1992), pp. 441–458.
[17] J. C. Meza, *A modification to the GMRES method for ill-conditioned linear systems*, Tech. Rep. SAND95-8220, Scientific Computing Dept., Sandia National Labs., Livermore, CA, 1995.
[18] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[19] R. B. Morgan, *On restarting the Arnoldi method for large scale eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.
[20] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, Tech. Rep., Baylor University, Waco, TX, 1997.
[21] C. Paige, B. Parlett, and H. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–134.
[22] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–125.
[23] Y. Saad, *Iterative methods for sparse linear systems*, The PWS Publishing Company, Boston, MA, 1996.
[24] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[26] V. Simoncini, *A new variant of restarted GMRES*, Numer. Linear Algebra Appl., 6 (1999), pp. 61–77.
[27] L. N. Trefethen, *Approximation theory and numerical linear algebra*, in Algorithms for Approximation II, J. C. Mason and M. G. Cox, eds., Chapman and Hall, London, 1990, pp. 336–360.
[28] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 453–471

AN ALGEBRAIC APPROACH TO THE CONSTRUCTION OF POLYHEDRAL INVARIANT CONES*

MARIA ELENA VALCHER† AND LORENZO FARINA‡

Abstract. In this paper, based on algebraic arguments, a new proof of the spectral characterization of those real matrices that leave a proper polyhedral cone invariant [*Trans. Amer. Math. Soc.*, 343 (1994), pp. 479–524] is given. The proof is a constructive one, as it allows us to explicitly obtain for every matrix A, which satisfies the aforementioned spectral requirements, an A-invariant proper polyhedral cone $\mathcal{K}$.

Some new results are also presented, concerning the way A acts on the cone $\mathcal{K}$. In particular, $\mathcal{K}$-irreducibility, $\mathcal{K}$-primitivity, and $\mathcal{K}$-positivity are fully characterized.

Key words. invariant cones, polyhedral cones, $\mathcal{K}$-irreducibility/primitivity/positivity, spectral radius, maximal modulus eigenvalues

AMS subject classifications. 15A18, 15A48, 51M20, 52B99, 93C05

PII. S0895479898335465

1. Introduction and motivation.

The last decades have witnessed a long stream of research aiming at generalizing the results of the Perron–Frobenius theory for nonnegative matrices (see [6, 9, 15, 21] for a complete survey) to a larger class of linear transformations. As a result, an extensive literature on the subject is now available. In fact, the infinite-dimensional case, first developed by Krein and Rutman in [14], is fully discussed in [17, 18], while the finite-dimensional aspects of this theory can be found in [6].

In the finite-dimensional context, research efforts led to the introduction of the notion of a matrix that leaves a proper cone invariant [2, 3, 7, 24], and to the determination of necessary and sufficient conditions for a real square matrix A to exhibit this property. It turns out that the existence of a proper A-invariant cone depends only on the spectral structure of A and, in particular, on its maximal modulus eigenvalues [2, 3, 6, 24].

More recently, some authors [4, 11] have investigated the above problem under the additional requirement that the proper A-invariant cone is polyhedral, namely, has a finite set of extremal vectors. It is worthwhile noticing that this problem arises in quite a few applications, such as in the nonnegative realization problem [1], in the relative stability of the Leontieff models in economics [16], in the description of dynamic systems by means of behavioral inequalities [23], in the synthesis of feedback control laws under state and/or input constraints [5], and in the analysis of positively invariant sets for continuous/discrete time systems [10, 22]. See [5] for further applications. For all these applications, in fact, the existence of a polyhedral cone, left invariant by a given square matrix A, represents the main ingredient of the problem itself.

In a recent paper [21], Tam and Schneider analyzed, by means of geometric tools, the properties of the core of a cone-preserving map, thus obtaining, as a significant

*Received by the editors March 16, 1998; accepted for publication (in revised form) by S. Van Huffel February 18, 2000; published electronically July 11, 2000.
http://www.siam.org/journals/simax/22-2/33546.html

†Dipartimento di Ingegneria dell'Innovazione, Università di Lecce, strada per Monteroni, 73100 Lecce, Italy (elena.valcher@unile.it).

‡Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", via Eudossiana 18, 00184 Roma, Italy (farina@dis.uniroma1.it).

by-product of this more general analysis, a spectral characterization of those real matrices that leave a proper polyhedral cone invariant (see Theorem 7.8 in [21]).

In this paper, based on algebraic arguments, we provide an alternative proof of the above characterization. Significantly enough, this proof is a constructive one, as it yields, when the assigned matrix A fulfills the aforementioned spectral conditions, a proper polyhedral A-invariant cone. Furthermore, by exploiting this algebraic approach, we devise what seems to be new conditions on the spectrum of A, which are equivalent to the existence of a proper polyhedral cone $\mathcal{K}$ such that A is $\mathcal{K}$-irreducible, $\mathcal{K}$-primitive, or $\mathcal{K}$-positive.

We will not address explicitly here the interesting problem of characterizing all possible A-invariant cones, but we will deal with the preliminary step of constructing at least one of them. This represents an important starting point, especially for the aforementioned control problem under state constraints. On the first hand, a fundamental issue in this context is that of *determining* a suitable invariant region in which to constrain the state evolution. This region can be a polyhedral cone, for instance, but even when our interest is in a different kind of invariant region (typically a polytope), by preliminarily constructing a proper polyhedral A-invariant cone one can obtain, as a by-product, a positively invariant polytope for the state dynamics. Indeed, in order to construct an invariant polytope for the $n \times n$ system matrix A, it is sufficient to construct a polyhedral cone, left invariant by the extended matrix (of size $n+1$) $\left[\begin{smallmatrix} r & 0 \\ 0 & A \end{smallmatrix}\right]$, where r is any positive real number strictly larger than the spectral radius of A. The projection of the cone over its last n components provides the desired invariant polytope.

On the other hand, if we assume a somehow opposite point of view and suppose that the physical constraints on the system naturally define a set S in which we want our state evolution to be confined, a major issue is that of constructing a suitable invariant region (as large as possible) that is strictly included in S [8, 10]. By assuming this point of view, a possible choice is that of choosing as invariant region a polyhedral cone or a polytope (again, obtained by first constructing a suitable polyhedral cone, left invariant by the extended matrix, and by later considering its projection). In the concluding section, once the details of our algorithm have been clarified, we will show, by means of an example, how our constructive procedure allows us to tackle this problem.

Finally, it is worthwhile to remark that in the special case of economical models, for which positively invariant regions correspond to "conservative" economical situations [16], the explicit construction of positively invariant regions allows us to determine areas of operating conditions where it can be convenient to lead the system, by means of suitable economic policies.

The paper is organized as follows. In section 2 we introduce some basic definitions of the theory of cones and some technical lemmas necessary for the subsequent analysis. Section 3 presents the main results of the paper.

2. Definitions and preliminary results. Throughout the paper we let $\mathbb{R}^n_+$ denote the nonnegative orthant, namely the set of nonnegative vectors in the n-dimensional Euclidean space $\mathbb{R}^n$. A set $\mathcal{K} \subset \mathbb{R}^n$ is said to be a *cone* if $\alpha\mathcal{K} \subset \mathcal{K}$ for all $\alpha \geq 0$; a cone is *convex* if it contains, with any two points, the line segment between them. A convex cone $\mathcal{K}$ is *solid* if it contains an open ball of $\mathbb{R}^n$, or, equivalently, if the interior of $\mathcal{K}$, $\text{int}(\mathcal{K})$, is nonempty, and it is *pointed* if $\mathcal{K} \cap \{-\mathcal{K}\} = \{0\}$. A closed, pointed, solid convex cone is called a *proper cone*. A cone $\mathcal{K}$ is said to be

polyhedral if it can be expressed as the set of nonnegative linear combinations of a finite set of *generating vectors*. This amounts to saying that a positive integer k and an $n \times k$ matrix C can be found such that $\mathcal{K}$ coincides with the set of nonnegative combinations of the columns of C. In this case, we adopt the notation $\mathcal{K} := \mathrm{Cone}(C)$.

A convex cone $\mathcal{F} \subset \mathcal{K}$ is a *face of* $\mathcal{K}$ if for every $\mathbf{v} \in \mathcal{F}$, condition $\mathbf{v} = \mathbf{u}_1 + \mathbf{u}_2$, for some $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{K}$, implies $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{F}$.

We call a bounded polyhedral set, namely, a bounded intersection of a finite family of closed halfspaces, a *polytope*. Every polytope can be expressed as the set of convex linear combinations of a finite family of (extremal) points.

If A is an $n \times n$ real matrix, we denote by $\sigma(A)$ its *spectrum* and by $\rho(A)$ its *spectral radius*, i.e., $\rho(A) := \max\{|\lambda| : \lambda \in \sigma(A)\}$. For every $\lambda \in \sigma(A)$, the *degree* of λ in A, $\deg \lambda$, is the size of the largest diagonal block in the Jordan canonical form of A that contains λ (i.e., the multiplicity of λ as a zero of the minimal polynomial of A).

Given $A \in \mathbb{R}^{n\times n}$ and a cone $\mathcal{K} \subseteq \mathbb{R}^n$, we say that A *leaves* $\mathcal{K}$ *invariant* ($\mathcal{K}$ is A-invariant) if $A\mathcal{K} \subseteq \mathcal{K}$. In this case, A is

- $\mathcal{K}$-*irreducible* if the only faces of $\mathcal{K}$ that A leaves invariant are $\{0\}$ or $\mathcal{K}$ itself;
- $\mathcal{K}$-*primitive* if the only nonempty subset of the boundary of $\mathcal{K}$ that A leaves invariant is $\{0\}$;
- $\mathcal{K}$-*positive* if $A(\mathcal{K} \setminus \{0\}) \subseteq \mathrm{int}(\mathcal{K})$, namely, A maps each point of $\mathcal{K}$, distinct from the origin, into $\mathrm{int}(\mathcal{K})$.

As is well known [6], for every proper cone $\mathcal{K}$ we have

$$A \text{ is } \mathcal{K}\text{-positive} \Rightarrow A \text{ is } \mathcal{K}\text{-primitive} \Rightarrow A \text{ is } \mathcal{K}\text{-irreducible}.$$

If $A = [a_{ij}]$ is a matrix (in particular, a vector), we write

- $A \geq 0$ (A *nonnegative*), if $a_{ij} \geq 0$ for all i, j;
- $A > 0$ (A *nonzero nonnegative*), if $a_{ij} \geq 0$ for all i, j, and $a_{hk} > 0$ for at least one pair (h, k);
- $A \gg 0$ (A *positive*), if $a_{ij} > 0$ for all i, j.

As a first step, we present some technical lemmas.

LEMMA 2.1 (see [11], Theorem 3.1). *Let C be any $n \times k$ real matrix, devoid of zero columns. Cone(C) is a proper (polyhedral) cone if and only if C has rank n and $\ker C$ includes no nonzero nonnegative vector.*

LEMMA 2.2 (see [19]). *Let A be an $n\times n$ real matrix that leaves a proper polyhedral cone $\mathcal{K}$ invariant. Then, for every nonsingular matrix $T \in \mathbb{R}^n$, $T^{-1}AT$ leaves a proper polyhedral cone invariant.*

LEMMA 2.3. *Let $\mathcal{P}$ be a convex polytope of $\mathbb{R}^2$, which includes the origin in its interior, and let $C \in \mathbb{R}^{2\times k}$ be a matrix having as column vectors the extremal points of $\mathcal{P}$. Every point $\mathbf{x}$ lying in the interior of $\mathcal{P}$ can be expressed as*

$$\mathbf{x} = C\mathbf{a},$$

for some vector $\mathbf{a} \gg 0$ with $\sum_{i=1}^{k} a_i < 1$.

Proof. The result follows immediately from elementary geometric arguments. ∎

3. Main results. In [24] Vandergraft proved that the spectrum of every matrix A that leaves a proper cone invariant satisfies two important requirements, i.e., $\rho(A)$ is in $\sigma(A)$ and for every maximal modulus eigenvalue λ of A we have $\deg \lambda \leq \deg \rho(A)$. The above pair of constraints constitutes the well-known *Perron–Schaefer condition* [21]. Conversely, when $\sigma(A)$ fulfills this condition, a proper cone $\mathcal{K}$ can be found such that $A\mathcal{K} \subseteq \mathcal{K}$.

The following theorem shows that, by simply introducing further "regularity" constraints on the maximal modulus eigenvalues, it is possible to obtain a similar statement for matrices that leave proper *polyhedral* cones invariant. More precisely, polyhedrality depends on the fact that the argument of each maximal modulus eigenvalue $\lambda \in \sigma(A)$ is a rational multiple of 2π or, equivalently, that $\lambda/\rho(A)$ is a root of unity. Notice that this is not unexpected, as nonnegative matrices always leave the nonnegative orthant invariant and, indeed, their spectral structure satisfies all these requirements.

THEOREM 3.1 (see [21]). *An $n \times n$ real matrix A leaves a proper polyhedral cone $\mathcal{K} \subset \mathbb{R}^n$ invariant if and only if*

(i) $\rho(A) \in \sigma(A)$*;*

and, when $\rho(A) \neq 0$*,*

(ii) $\lambda \in \sigma(A)$ *with* $|\lambda| = \rho(A)$ *implies* $\deg \lambda \leq \deg \rho(A) =: m$*;*

(iii) $\lambda \in \sigma(A)$ *with* $|\lambda| = \rho(A)$ *implies* $\lambda/\rho(A)$ *is a root of unity.*

Proof. [Necessity.] As $\mathcal{K}$ is, in particular, a proper cone, (i) and (ii) follow from the well-known theorem credited to Birkhoff and Vandergraft (see also [6], pp. 6–7). The necessity of (iii) was proved in [4]. The same result was later proved also in [20] (see Theorem 7.6) by means of a rather straightforward proof.

[Sufficiency.] By Lemma 2.2, it entails no loss of generality assuming that A is in real Jordan form. In fact, we can always reduce to this case by means of a suitable change of basis in $\mathbb{R}^n$.

If A is nilpotent, it is nonnegative and hence leaves the nonnegative orthant $\mathbb{R}^n_+$ invariant. If $\rho(A)$ is positive, we can suppose $\rho(A) = 1$, because A leaves a proper polyhedral cone invariant if and only if $A/\rho(A)$ does. Possibly after suitable row-column permutations, A takes the following block-diagonal structure:

$$A = \left[\begin{array}{c|c|c|c} J_1 & & & \\ & \begin{array}{ccc} J_2 & & \\ & \ddots & \\ & & J_t \end{array} & & \\ \hline & & \begin{array}{ccc} J_{t+1} & & \\ & \ddots & \\ & & J_r \end{array} & \\ \hline & & & \begin{array}{ccc} F_1 & & \\ & \ddots & \\ & & F_s \end{array} \end{array}\right], \tag{1}$$

where the J_i's, $i = 1, 2, \ldots, r$, are Jordan blocks corresponding to real eigenvalues, i.e.,

$$J_i = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \lambda_i & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{bmatrix} \in \mathbb{R}^{n_i \times n_i}, \tag{2}$$

and satisfying the following three conditions:

(a) $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_t \geq 0 > \lambda_{t+1} \geq \cdots \geq \lambda_r \geq -1$,

(b) $\lambda_i = \lambda_{i+1}$ implies $n_i \geq n_{i+1}$, (consequently, J_1 is the Jordan block of size $m = \deg 1$ corresponding to the eigenvalue 1), and

(c) $\lambda_i = -1$ implies $n_i \le m$,

whereas the F_i's, $i = 1, 2, \ldots, s$, are Jordan blocks associated with pairs of conjugate complex eigenvalues $\sigma_i \pm j\omega_i$, $\omega_i \neq 0$, i.e.,

$$F_i = \left[\begin{array}{cc|cc|cc|c|cc} \sigma_i & \omega_i & 1 & 0 & & & & & \\ -\omega_i & \sigma_i & 0 & 1 & & & & & \\ \hline & & \sigma_i & \omega_i & 1 & 0 & & & \\ & & -\omega_i & \sigma_i & 0 & 1 & & & \\ \hline & & & & \ddots & & \ddots & & \\ \hline & & & & & & & 1 & 0 \\ & & & & & & & 0 & 1 \\ \hline & & & & & & & \sigma_i & \omega_i \\ & & & & & & & -\omega_i & \sigma_i \end{array}\right] \in \mathbb{C}^{(2\bar{n}_i)\times(2\bar{n}_i)}, \tag{3}$$

and satisfying

(d) $1 \ge (\sigma_1^2 + \omega_1^2) \ge \cdots \ge (\sigma_s^2 + \omega_s^2) > 0$,

(e) $\sigma_i + j\omega_i = \sigma_{i+1} + j\omega_{i+1}$ implies $\bar{n}_i \ge \bar{n}_{i+1}$, and

(f) $\sigma_i^2 + \omega_i^2 = 1$ implies $\bar{n}_i \le m$.

We aim at constructing a block-triangular matrix C such that $\mathrm{Cone}(C)$ is an A-invariant (polyhedral) proper cone. To this end, we separately analyze each block appearing in A.

• **Blocks corresponding to nonnegative real eigenvalues:** every $n_i \times n_i$ block J_i, corresponding to an eigenvalue $\lambda_i \ge 0$, is a nonzero nonnegative matrix and hence leaves the positive orthant $\mathbb{R}^{n_i}_+ = \mathrm{Cone}(I_{n_i})$ invariant. Therefore,

$$J_i I_{n_i} = I_{n_i} J_i, \qquad i = 1, 2, \ldots, t. \tag{4}$$

• **Blocks corresponding to negative real eigenvalues:** each $n_i \times n_i$ block J_i, corresponding to an eigenvalue $\lambda_i < 0$, leaves the vector space $\mathbb{R}^{n_i}$, which is a solid (of course, not pointed) polyhedral cone, invariant. Indeed, for $i = t+1, \ldots, r$,

$$J_i \,[I_{n_i} | -I_{n_i}] = [I_{n_i} | -I_{n_i}] \left[\begin{array}{c|c} N_i & -\lambda_i I_{n_i} \\ \hline -\lambda_i I_{n_i} & N_i \end{array}\right], \tag{5}$$

where

$$N_i := \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{n_i \times n_i}_+. \tag{6}$$

As a consequence, for $n_i \le m$ we get

$$\left[\begin{array}{c|c} J_1 & 0 \\ \hline 0 & J_i \end{array}\right] \left[\begin{array}{c|c|c} I_m & \begin{array}{c} I_{n_i} \\ 0 \end{array} & \begin{array}{c} I_{n_i} \\ 0 \end{array} \\ \hline 0 & I_{n_i} & -I_{n_i} \end{array}\right] = \left[\begin{array}{c|c|c} I_m & \begin{array}{c} I_{n_i} \\ 0 \end{array} & \begin{array}{c} I_{n_i} \\ 0 \end{array} \\ \hline 0 & I_{n_i} & -I_{n_i} \end{array}\right] \left[\begin{array}{c|c|c} J_1 & \begin{array}{c} (1+\lambda_i) I_{n_i} \\ 0 \end{array} & \begin{array}{c} (1+\lambda_i) I_{n_i} \\ 0 \end{array} \\ \hline 0 & N_i & -\lambda_i I_{n_i} \\ \hline 0 & -\lambda_i I_{n_i} & N_i \end{array}\right], \tag{7}$$

while for $n_i > m$, a case that possibly occurs for negative real eigenvalues λ_i different from -1, we have

$$\left[\begin{array}{c|c} J_1 & 0 \\ \hline 0 & J_i \end{array}\right] \left[\begin{array}{c|c|c} I_m & I_m \quad \mathbf{e}_m \mathbf{w}(\lambda_i, n_i - m) & I_m \quad \mathbf{e}_m \mathbf{w}(\lambda_i, n_i - m) \\ \hline 0 & I_{n_i} & -I_{n_i} \end{array}\right]$$

$$= \left[\begin{array}{c|cc|cc} I_m & I_m & \mathbf{e}_m\mathbf{w}(\lambda_i, n_i - m) & I_m & \mathbf{e}_m\mathbf{w}(\lambda_i, n_i - m) \\ \hline 0 & \multicolumn{2}{c|}{I_{n_i}} & \multicolumn{2}{c}{-I_{n_i}} \end{array}\right],$$

$$\left[\begin{array}{c|cc|cc} J_1 & (1+\lambda_i)I_m & \mathbf{e}_{m-1}\mathbf{w}(\lambda_i, n_i - m) & (1+\lambda_i)I_m & \mathbf{e}_{m-1}\mathbf{w}(\lambda_i, n_i - m) \\ \hline 0 & \multicolumn{2}{c|}{N_i} & \multicolumn{2}{c}{-\lambda_i I_{n_i}} \\ \hline 0 & \multicolumn{2}{c|}{-\lambda_i I_{n_i}} & \multicolumn{2}{c}{N_i} \end{array}\right], \tag{8}$$

where $\mathbf{e}_0$ is by definition the zero vector, while, when i is positive, $\mathbf{e}_i$ is the ith canonical (column) vector in $\mathbb{R}^m$ (having all zero entries, except for the ith, which is 1), and

$$\mathbf{w}(\lambda, k) := \left[\begin{array}{cccc} \frac{1}{1+\lambda} & \frac{1}{(1+\lambda)^2} & \cdots & \frac{1}{(1+\lambda)^k} \end{array}\right].$$

• **Blocks corresponding to pairs of complex conjugate eigenvalues $\boldsymbol{\sigma_i \pm j\omega_i}$:** let $\mathbf{v}_1^{(i)}$ be a nonzero real vector (for instance, $[1\ 0]^T$), and consider the vector sequence $\mathbf{v}_\ell^{(i)} := \left[\begin{array}{cc} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{array}\right]\mathbf{v}_{\ell-1}^{(i)}, \ell > 1$. Since $\sigma_i^2 + \omega_i^2 \leq 1$, and when $\sigma_i^2 + \omega_i^2 = 1$ then $\sigma_i \pm j\omega_i$ are roots of unity, there exists some positive integer k_i such that

$$\left[\begin{array}{cc} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{array}\right]\mathbf{v}_{k_i}^{(i)} = \sum_{\ell=0}^{k_i-1} a_\ell^{(i)}\mathbf{v}_{\ell+1}^{(i)}$$

holds true for suitable $a_\ell^{(i)} \geq 0$, $\ell = 0, 1, \ldots, k_i - 1$, with $\sum_\ell a_\ell^{(i)} \leq 1$. More precisely, when $\sigma_i \pm j\omega_i$ are roots of unity, then it is possible to choose k_i as the smallest positive integer such that $(\sigma_i + j\omega_i)^{k_i+1} = 1$ and set $a_0^{(i)} = 1$ and $a_\ell^{(i)} = 0$ for $\ell = 1, 2, \ldots, k_i - 1$. Meanwhile, for $\sigma_i^2 + \omega_i^2 < 1$, the integer k_i is the smallest positive integer such that $(k_i + 1)|\arg(\sigma_i + j\omega_i)|$ is at least 2π radians, and the positive coefficients $a_\ell^{(i)}$ sum up to a quantity that is strictly smaller than 1 (see Lemma 2.3). Therefore, in both cases, the real matrix

$$C_i := \left[\begin{array}{c|c|c|c} \mathbf{v}_1^{(i)} & \mathbf{v}_2^{(i)} & \cdots & \mathbf{v}_{k_i}^{(i)} \end{array}\right] \in \mathbb{R}^{2\times k_i}$$

is of full row rank and satisfies

$$\left[\begin{array}{cc} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{array}\right] C_i = C_i\, \Gamma_i, \tag{9}$$

where

$$\Gamma_i := \begin{bmatrix} 0 & & & & & a_0^{(i)} \\ 1 & 0 & & & & a_1^{(i)} \\ & 1 & 0 & & & a_2^{(i)} \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & 0 & a_{k_i-2}^{(i)} \\ & & & & 1 & a_{k_i-1}^{(i)} \end{bmatrix} \in \mathbb{R}_+^{k_i\times k_i}. \tag{10}$$

Thus we get

$$F_i \begin{bmatrix} C_i & & \\ & \ddots & \\ & & C_i \end{bmatrix} = \begin{bmatrix} C_i & & \\ & \ddots & \\ & & C_i \end{bmatrix} \begin{bmatrix} \Gamma_i & I_{k_i} & & & \\ & \Gamma_i & I_{k_i} & & \\ & & \Gamma_i & \ddots & \\ & & & \ddots & I_{k_i} \\ & & & & \Gamma_i \end{bmatrix}, \tag{11}$$

where, of course, the block diagonal matrix in (11), having all C_i's as diagonal blocks, has size $(2\bar{n}_i) \times (\bar{n}_i k_i)$.

Let $\mathbf{1}_k$ be the k-dimensional (row) vector with all entries equal to 1. For $\bar{n}_i \le m$ we get

$$\begin{aligned} &\left[\begin{array}{c|c} J_1 & 0 \\ \hline 0 & F_i \end{array}\right] \left[\begin{array}{c|ccc} I_m & \mathbf{e}_1\mathbf{1}_{k_i} & \dots & \mathbf{e}_{\bar{n}_i}\mathbf{1}_{k_i} \\ \hline & C_i & & \\ 0 & & \ddots & \\ & & & C_i \end{array}\right] \\ &= \left[\begin{array}{c|ccc} I_m & \mathbf{e}_1\mathbf{1}_{k_i} & \dots & \mathbf{e}_{\bar{n}_i}\mathbf{1}_{k_i} \\ \hline & C_i & & \\ 0 & & \ddots & \\ & & & C_i \end{array}\right] \left[\begin{array}{c|ccccc} J_1 & \mathbf{e}_1\mathbf{c}_i & \mathbf{e}_2\mathbf{c}_i & \cdots & & \mathbf{e}_{\bar{n}_i}\mathbf{c}_i \\ \hline & \Gamma_i & I_{k_i} & & & \\ & & \Gamma_i & I_{k_i} & & \\ 0 & & & \Gamma_i & \ddots & \\ & & & & \ddots & I_{k_i} \\ & & & & & \Gamma_i \end{array}\right] \end{aligned} \tag{12}$$

with $\mathbf{c}_i := \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 - \sum_\ell a_\ell^{(i)} \end{bmatrix} \ge 0$. On the other hand, if $\bar{n}_i > m$, a situation that may occur only for $\sigma_i^2 + \omega_i^2 < 1$ and hence for $\sum_\ell a_\ell^{(i)} < 1$, we have that the spectral radius of Γ_i is strictly smaller than 1, and the identity

$$\left[\begin{array}{c|c} J_1 & 0 \\ \hline 0 & F_i \end{array}\right] \left[\begin{array}{c|ccccc} I_m & \mathbf{e}_1\mathbf{1}_{k_i} & \dots & \mathbf{e}_m\mathbf{1}_{k_i} & & X_i \\ \hline & C_i & & & & \\ & & \ddots & & & \\ 0 & & & C_i & & \\ & & & & C_i & \\ & & & & & \ddots \\ & & & & & C_i \end{array}\right]$$

(13)

$$= \left[\begin{array}{c|ccccccc} I_m & \mathbf{e}_1\mathbf{1}_{k_i} & \mathbf{e}_2\mathbf{1}_{k_i} & \dots & \mathbf{e}_m\mathbf{1}_{k_i} & & X_i \\ \hline & C_i & & & & & \\ & & C_i & & & & \\ & & & \ddots & & & \\ 0 & & & & C_i & & \\ & & & & & C_i & \\ & & & & & & \ddots \\ & & & & & & C_i \end{array}\right] \left[\begin{array}{c|ccccccc} J_1 & \mathbf{e}_1\mathbf{c}_i & \mathbf{e}_2\mathbf{c}_i & \dots & \mathbf{e}_m\mathbf{c}_i & 0 & \cdots & 0 \\ \hline & \Gamma_i & I_{k_i} & & & & & \\ & & \Gamma_i & \ddots & & & & \\ & & & \ddots & & & & \\ 0 & & & & \Gamma_i & I_{k_i} & & \\ & & & & & \Gamma_i & \ddots & \\ & & & & & & \ddots & I_{k_i} \\ & & & & & & & \Gamma_i \end{array}\right]$$

holds true for some nonnegative matrix X_i, of size $m \times [(\bar{n}_i - m)k_i]$, devoid of zero columns. In fact, as the spectral radius of Γ_i is strictly smaller than 1, the matrix equation in the unknown matrix X

$$J_1 X = [\, \mathbf{e}_m \mathbf{1}_{k_i} \quad 0 \,] + X \begin{bmatrix} \Gamma_i & I_{k_i} & & & \\ & \Gamma_i & I_{k_i} & & \\ & & \Gamma_i & \ddots & \\ & & & \ddots & I_{k_i} \\ & & & & \Gamma_i \end{bmatrix} \tag{14}$$

is solvable [13] and admits as its (unique) solution

$$X_i := \int_0^{+\infty} \exp(-J_1 t)\, [\, \mathbf{e}_m \mathbf{1}_{k_i} \quad 0 \,] \exp\left(\begin{bmatrix} \Gamma_i & I_{k_i} & & & \\ & \Gamma_i & I_{k_i} & & \\ & & \Gamma_i & \ddots & \\ & & & \ddots & I_{k_i} \\ & & & & \Gamma_i \end{bmatrix} t \right) dt.$$

(This can be proved by simply replacing the above expression in (14).) By the structure and the nonnegativity property of the matrices involved, X_i is nonnegative and has no zero columns.

So, if we now consider the block triangular matrix

$$C = \left[\begin{array}{c|c|c|c|c|c|c|c|c} I_m & 0 & \cdots & 0 & \bar{X}_{t+1} & \cdots & \bar{X}_r & \bar{X}_1 & \cdots \quad \bar{X}_s \\ \hline & I_{n_2} & & & & & & & \\ & & \ddots & & & & & & \\ & & & I_{n_t} & & & & & \\ \hline & & & & I_{n_{t+1}} - I_{n_{t+1}} & & & & \\ & & & & & \ddots & & & \\ \hline & & & & & & I_{n_r} - I_{n_r} & & \\ & & & & & & & C_1 & \\ & & & & & & & & \ddots \\ & & & & & & & & C_1 \\ & & & & & & & & \\ & & & & & & & & \ddots \\ & & & & & & & & C_s \\ & & & & & & & & \\ & & & & & & & & \ddots \\ & & & & & & & & C_s \end{array}\right],$$

where the nonnegative matrices $\tilde{X}_i$ and $\bar{X}_i$ (devoid of zero columns) are easily derived by the previous equations (4), (7), (8), (12), and (14), it is easy to check that C is of two full row rank and its kernel does not include nonnegative vectors, except for the zero one. Consequently, the A-invariant polyhedral cone Cone(C) is, by Lemma 2.1, proper. □

Remarks. The above proof provides, for a matrix A that fulfills (i)–(iii) of Theorem 3.1, an explicit procedure for constructing a proper polyhedral cone left invariant by A. As a matter of fact, this procedure does not lead to the construction of a unique cone: different choices of the vectors $\mathbf{v}_1^{(i)}$ lead to different A-invariant polyhedral cones. Also, small variations could be introduced in the design procedure that keep in with the spirit of the above constructive algorithm but better enlighten the existence of several choices and hence of different polyhedral cones. This aspect, however, falls outside the goals we aimed to achieve in this paper.

The basic steps of the constructive algorithm can be briefly summarized as follows:

- construct the Jordan form of the matrix A (by assuming the same ordering adopted in the proof);
- construct, for every pair of complex conjugate eigenvalues of A, the matrices C_i and, correspondingly, the matrices Γ_i and X_i;
- now that we have obtained a proper polyhedral cone that is left invariant by the Jordan form of A, apply the appropriate change of coordinates, and obtain a proper polyhedral A-invariant cone.

It is worthwhile noticing that the procedure for obtaining a proper A-invariant cone given by Vandergraft in [6] and [24] does not generally lead to a polyhedral cone, not even when the eigenvalues of A satisfy all the abovementioned conditions. This fact is clearly pointed out in the following example.

Example 3.1. Consider the real matrix

$$A = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & -1 & 0 \end{array}\right].$$

Its spectrum is $\sigma(A) = (1, j, -j)$ and hence fulfills all the assumptions of Theorem 3.1. A (complex) Jordan basis is

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{v}_2 = \begin{bmatrix} 0 \\ j/2 \\ 1/2 \end{bmatrix}, \qquad \bar{\mathbf{v}}_2 = \begin{bmatrix} 0 \\ -j/2 \\ 1/2 \end{bmatrix},$$

and hence the Vandergraft cone is given by

$$\mathcal{K} = \left\{ \mathbf{v} \in \mathbb{R}^n : \mathbf{v} = \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ b \end{bmatrix}, a^2 + b^2 \leq \alpha^2, \alpha \geq 0 \right\}$$

and is clearly not polyhedral. By applying the procedure described in the previous proof, we construct a proper polyhedral cone left invariant by A, namely, the one generated by the columns of the matrix

$$C = \left[\begin{array}{c|cccc} 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{array}\right].$$

Theorem 3.1 clarifies under what conditions a matrix A leaves a proper polyhedral cone, say $\mathcal{K}$, invariant. When trying to strengthen this result, by requiring that A is also $\mathcal{K}$-irreducible, we have to restrict our attention to a smaller class of matrices, namely, those that satisfy the conditions of Theorem 3.1 and whose maximal modulus eigenvalues have degree 1.

The proof of the following theorem resorts to a well-known characterization of a $\mathcal{K}$-irreducible matrix [6].

THEOREM 3.2. *Let A be an $n \times n$ real matrix. The matrix A leaves invariant a proper polyhedral cone $\mathcal{K} \subset \mathbb{R}^n$, for which it is $\mathcal{K}$-irreducible if and only if*

(i) $\rho(A) \in \sigma(A)$;

(ii) $\rho(A)$ *is simple and for every* $\lambda \in \sigma(A)$ *with* $|\lambda| = \rho(A)$, $\deg \lambda = 1$,

and, if $\rho(A) \neq 0$,

(iii) *for every* $\lambda \in \sigma(A)$ *with* $|\lambda| = \rho(A)$, $\lambda/\rho(A)$ *is a root of unity.*

Proof. [Necessity.] As A leaves a proper polyhedral cone invariant, (i) and (iii) follow from Theorem 3.1. Moreover, since $\mathcal{K}$ is a proper A-invariant cone, (ii) follows from the characterization of $\mathcal{K}$-irreducibility due to Vandergraft [24] and Elsner [12].

[Sufficiency.] As in the proof of Theorem 3.1, we can assume that A is in real Jordan form. If A is nilpotent, then, by assumption (ii), A has to be the 1×1 zero matrix, which leaves $\mathcal{K} = \mathbb{R}_+$ invariant and is $\mathcal{K}$-irreducible.

Now suppose $\rho(A) = 1$ and assume

$$A = \left[\begin{array}{c|ccc|ccc} 1 & & & & & & \\ \hline & J_2 & & & & & \\ & & \ddots & & & & \\ & & & J_r & & & \\ \hline & & & & A_1 & & \\ & & & & & \ddots & \\ & & & & & & A_s \end{array}\right], \tag{15}$$

where J_i, $i = 2, \dots, r$, is the $n_i \times n_i$ Jordan block corresponding to the nonnegative real eigenvalue λ_i, and $1 > \lambda_2 \geq \cdots \geq \lambda_r \geq 0$, whereas A_i, $i = 1, 2, \dots, s$, represents the $n_i \times n_i$ Jordan block associated either with negative eigenvalues or with pairs of complex conjugate eigenvalues. All Jordan blocks corresponding to the eigenvalue -1 have size 1, while those associated with any pair of complex conjugate eigenvalues of modulus 1 have dimension 2.

We aim at explicitly constructing an A-invariant proper polyhedral cone $\mathcal{K} = \text{Cone}(C)$, which includes only one (up to scalar multiples) eigenvector of A, lying in $\text{int}(\mathcal{K})$. This guarantees [7, Theorem 3.16, p. 11] that A is $\mathcal{K}$-irreducible.

As in the previous proof, we can find full row rank matrices $\bar{C}_i$, nonnegative matrices $\tilde{P}_i$ and P_i, and positive row vectors X_i (notice that $m = 1$) such that

$$\begin{bmatrix} 1 & 0 \\ 0 & A_i \end{bmatrix} \begin{bmatrix} 1 & X_i \\ 0 & \bar{C}_i \end{bmatrix} = \begin{bmatrix} 1 & X_i \\ 0 & \bar{C}_i \end{bmatrix} \begin{bmatrix} 1 & \tilde{P}_i \\ 0 & P_i \end{bmatrix}, \qquad i = 1, 2, \dots, s.$$

Now consider the Jordan blocks J_i's. Once we introduce the $n_i \times (2n_i)$ (full row rank) matrices

$$D(\lambda_i, n_i) := \left[\begin{array}{ccc|ccc} & & \frac{1}{(1-\lambda_i)^{n_i-1}} & & & -\frac{1}{(1-\lambda_i)^{n_i-1}} \\ & \iddots & & & \iddots & \\ \frac{1}{(1-\lambda_i)} & & & -\frac{1}{(1-\lambda_i)} & & \\ 1 & & & -1 & & \end{array}\right]$$

for $i = 2, 3, \dots, r$, we have

$$J_i D(\lambda_i, n_i) = D(\lambda_i, n_i) P_i,$$

where

$$P_i := \left[\begin{array}{ccccc|ccccc} \lambda_i & & & & & & & & & \\ 1-\lambda_i & \lambda_i & & & & & & & & \\ & 1-\lambda_i & \lambda_i & & & & & 0 & & \\ & & \ddots & \ddots & & & & & & \\ & & & 1-\lambda_i & \lambda_i & & & & & \\ \hline & & & & & \lambda_i & & & & \\ & & & & & 1-\lambda_i & \lambda_i & & & \\ & & 0 & & & & 1-\lambda_i & \lambda_i & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & 1-\lambda_i & \lambda_i \end{array}\right]$$

is a $(2n_i) \times (2n_i)$ nonnegative matrix.

So, it is easy to check that the following matrix

$$C = \left[\begin{array}{c|ccc|ccc} 1 & \mathbf{1}_{2n_2} & \cdots & \mathbf{1}_{2n_r} & X_1 & \dots & X_s \\ \hline & D(\lambda_2, n_2) & & & & & \\ & & \ddots & & & & \\ & & & D(\lambda_r, n_r) & & & \\ \hline & & & & \bar{C}_1 & & \\ & & & & & \ddots & \\ & & & & & & \bar{C}_s \end{array}\right]$$

generates an A-invariant proper polyhedral cone $\mathcal{K} = \mathrm{Cone}(C)$. It is easily seen (due to the fact that A is in real Jordan form, and hence its eigenvectors have a very simple structure) that the only eigenvector of A lying in $\mathcal{K}$ is $\mathbf{e}_1$ and it corresponds to the eigenvalue $\rho(A) = 1$. Moreover, by exploiting the property that the columns of each matrix $\bar{C}_i \in \mathbb{R}^{2\times\cdot}$ generate a convex polytope, having the origin as an internal point, and by recalling that the same holds true for each matrix $D(\lambda_i, n_i)$, we can apply Lemma 2.3, and finally express $\mathbf{e}_1$ as a strictly positive combination of the columns of C. Consequently, $\mathbf{e}_1$ belongs to $\mathrm{int}(\mathcal{K})$ (see [11]), and A is $\mathcal{K}$-irreducible. □

We provide, now, a characterization of a matrix A for which a proper polyhedral cone $\mathcal{K}$ can be found such that A is $\mathcal{K}$ -primitive. As we will see, the spectral conditions allowing for $\mathcal{K}$-primitivity, with $\mathcal{K}$ proper and polyhedral, are the same allowing for $\mathcal{K}'$-positivity, with respect to some proper polyhedral cone $\mathcal{K}'$, in general distinct from $\mathcal{K}$.

THEOREM 3.3. *Let A be any $n \times n$ real matrix. The following conditions are equivalent:*

(a) *A leaves invariant a proper polyhedral cone $\mathcal{K}$ in $\mathbb{R}^n$ for which it is $\mathcal{K}$-positive;*
(b) *A leaves invariant a proper polyhedral cone $\mathcal{K}'$ in $\mathbb{R}^n$ for which it is $\mathcal{K}'$-primitive;*
(c) *the spectrum of A satisfies the following constraints:*
  (i) *$\rho(A)$ is a simple positive eigenvalue in $\sigma(A)$;*
  (ii) *for every $\lambda \in \sigma(A)$ with $\lambda \neq \rho(A)$, $|\lambda| < \rho(A)$.*

Proof. (a) ⇒ (b). If A leaves $\mathcal{K}$ invariant and is $\mathcal{K}$-positive, it is also $\mathcal{K}$-primitive, and (b) holds for $\mathcal{K}' := \mathcal{K}$.

(b) ⇒ (c). If A leaves invariant a proper polyhedral cone $\mathcal{K}'$ in $\mathbb{R}^n$ for which it is $\mathcal{K}'$-primitive, then, in particular, it leaves a proper cone invariant for which it is primitive. So, by a well-known result (see Theorem 4.10 in [6]), conditions (i) and (ii) are satisfied.

(c) ⇒ (a). As in the previous proofs, we assume that A is in real Jordan form. As $\rho(A)$ is positive, we can suppose, without loss of generality, that $\rho(A)$ is one and A has the following form

$$A = \left[\begin{array}{c|ccc|ccc} 1 & & & & & & \\ \hline & J_2 & & & & & \\ & & \ddots & & & & \\ & & & J_r & & & \\ \hline & & & & F_1 & & \\ & & & & & \ddots & \\ & & & & & & F_s \end{array}\right], \tag{16}$$

where J_i, $i = 2, \dots, r$, is the $n_i \times n_i$ Jordan block corresponding to the real eigenvalue λ_i of A, with $1 > \lambda_i > -1$, whereas F_i, $i = 1, 2, \dots, s$, is the $2\bar{n}_i \times 2\bar{n}_i$ Jordan block associated with the pair of complex conjugate eigenvalues $\sigma_i \pm j\omega_i$, with $\sigma_i^2 + \omega_i^2 < 1$.

• We consider, first, any Jordan block J_i. As $1 - \lambda_i > 0$, we can select ε_i, $0 < \varepsilon_i < 1 - \lambda_i$, and introduce the $n_i \times (2n_i)$ full row rank matrix (see the proof of Theorem 3.2)

$$D(\lambda_i+\varepsilon_i, n_i) := \left[\begin{array}{ccc|ccc} & & \frac{1}{(1-\lambda_i-\varepsilon_i)^{n_i-1}} & & & -\frac{1}{(1-\lambda_i-\varepsilon_i)^{n_i-1}} \\ & \iddots & & & \iddots & \\ \frac{1}{(1-\lambda_i-\varepsilon_i)} & & & -\frac{1}{(1-\lambda_i-\varepsilon_i)} & & \\ 1 & & & -1 & & \end{array}\right].$$

It is easy to see that

$$J_i D(\lambda_i + \varepsilon_i, n_i) = D(\lambda_i + \varepsilon_i, n_i) P_i,$$

where

$$P_i := \left[\begin{array}{ccccc|ccccc} \lambda_i & & & & & & & & & \\ 1-\lambda_i-\varepsilon_i & \lambda_i & & & & & & & & \\ & 1-\lambda_i-\varepsilon_i & \lambda_i & & & & & 0 & & \\ & & \ddots & \ddots & & & & & & \\ & & & 1-\lambda_i-\varepsilon_i & \lambda_i & & & & & \\ \hline & & & & & \lambda_i & & & & \\ & & & & & 1-\lambda_i-\varepsilon_i & \lambda_i & & & \\ & & 0 & & & & 1-\lambda_i-\varepsilon_i & \lambda_i & & \\ & & & & & & & \ddots & \ddots & \\ & & & & & & & & 1-\lambda_i-\varepsilon_i & \lambda_i \end{array}\right]$$

is a $(2n_i) \times (2n_i)$ nonnegative matrix.

• Consider now the $(2\bar{n}_i) \times (2\bar{n}_i)$ Jordan block F_i corresponding to the pair of conjugate complex eigenvalues $\sigma_i \pm j\omega_i$. As $\sigma_i^2 + \omega_i^2 < 1$, there exists $\delta_i > 0$ such that $(1 + \delta_i)(\sigma_i \pm j\omega_i)$ still constitutes a pair of conjugate complex eigenvalues, with modulus smaller than 1. The Hurwitz stability of

$$\begin{bmatrix} (1+\delta_i)\sigma_i & (1+\delta_i)\omega_i \\ -(1+\delta_i)\omega_i & (1+\delta_i)\sigma_i \end{bmatrix}$$

guarantees, as in Theorem 3.1, that for any $\mathbf{v}_1^{(i)} \neq 0$ (for instance, $\mathbf{v}_1^{(i)} := \mathbf{e}_1$), the vector sequence $\mathbf{v}_1^{(i)}$, $\mathbf{v}_\ell^{(i)} := \left[\begin{smallmatrix} (1+\delta_i)\sigma_i & (1+\delta_i)\omega_i \\ -(1+\delta_i)\omega_i & (1+\delta_i)\sigma_i \end{smallmatrix}\right] \mathbf{v}_{\ell-1}^{(i)}$, $\ell > 1$ satisfies the following condition: there exists some positive integer k_i such that

$$\begin{bmatrix} (1+\delta_i)\sigma_i & (1+\delta_i)\omega_i \\ -(1+\delta_i)\omega_i & (1+\delta_i)\sigma_i \end{bmatrix} \mathbf{v}_{k_i}^{(i)} = \sum_{\ell=0}^{k_i-1} a_\ell^{(i)} \mathbf{v}_{\ell+1}^{(i)},$$

for suitable $a_\ell^{(i)} > 0$, $\ell = 0, 1, \dots, k_i - 1$, with $\sum_\ell a_\ell^{(i)} < 1$. Therefore,

$$C_i := \left[\begin{array}{c|c|c|c} \mathbf{v}_1^{(i)} & \mathbf{v}_2^{(i)} & \cdots & \mathbf{v}_{k_i}^{(i)} \end{array}\right] \in \mathbb{R}^{2 \times k_i}$$

is a full row rank matrix satisfying

$$\begin{bmatrix} (1+\delta_i)\sigma_i & (1+\delta_i)\omega_i \\ -(1+\delta_i)\omega_i & (1+\delta_i)\sigma_i \end{bmatrix} C_i = C_i \begin{bmatrix} 0 & & & & a_0^{(i)} \\ 1 & 0 & & & a_1^{(i)} \\ & 1 & 0 & & a_2^{(i)} \\ & & 1 & \ddots & \vdots \\ & & & \ddots & 0 \quad a_{k_i-2}^{(i)} \\ & & & & 1 \quad a_{k_i-1}^{(i)} \end{bmatrix}, \tag{17}$$

and, consequently,

$$\begin{bmatrix} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{bmatrix} C_i = C_i \begin{bmatrix} 0 & & & & b_0^{(i)} \\ \frac{1}{1+\delta_i} & 0 & & & b_1^{(i)} \\ & \frac{1}{1+\delta_i} & \ddots & & b_2^{(i)} \\ & & \ddots & 0 & \vdots \\ & & & \frac{1}{1+\delta_i} & b_{k_i-1}^{(i)} \end{bmatrix}, \tag{18}$$

with $b_\ell^{(i)} := a_\ell^{(i)}/(1+\delta_i) > 0$ and $\sum_\ell b_\ell^{(i)} < \frac{1}{1+\delta_i}$. Moreover, it is easy to verify that for every $\tau_i > 0$ the following equality holds

$$F_i \begin{bmatrix} \tau_i^{1-\bar{n}_i} C_i & & & \\ & \ddots & & \\ & & \tau_i^{-1} C_i & \\ & & & C_i \end{bmatrix} = \begin{bmatrix} \tau_i^{1-\bar{n}_i} C_i & & & \\ & \ddots & & \\ & & \tau_i^{-1} C_i & \\ & & & C_i \end{bmatrix} \begin{bmatrix} \Gamma_i & \tau_i I_{k_i} & & \\ & \Gamma_i & \tau_i I_{k_i} & \\ & & \Gamma_i & \ddots \\ & & & \ddots \quad \tau_i I_{k_i} \\ & & & \Gamma_i \end{bmatrix},$$

where

$$\Gamma_i := \begin{bmatrix} 0 & & & & & b_0^{(i)} \\ \frac{1}{1+\delta_i} & 0 & & & & b_1^{(i)} \\ & \frac{1}{1+\delta_i} & 0 & & & b_2^{(i)} \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & 0 & b_{k_i-2}^{(i)} \\ & & & & \frac{1}{1+\delta_i} & b_{k_i-1}^{(i)} \end{bmatrix} \in \mathbb{R}^{k_i \times k_i}. \tag{19}$$

Once we select positive real numbers ε_i, τ_i, and δ_i satisfying

$$\varepsilon_i < 1 - \lambda_i, \qquad i = 1, 2, \ldots, r, \qquad \tau_i + \frac{1}{1+\delta_i} < 1, \qquad i = 1, 2, \ldots, s,$$

the full row rank matrix

$$C = \left[\begin{array}{c|ccc|c|c|c} 1 & \mathbf{1}_{2n_2} & \cdots & \mathbf{1}_{2n_r} & \mathbf{1}_{k_i\bar{n}_1} & \cdots & \mathbf{1}_{k_s\bar{n}_s} \\ \hline & D(\lambda_2+\varepsilon_2,n_2) & & & & & \\ & & \ddots & & & & \\ & & & D(\lambda_r+\varepsilon_r,n_r) & & & \\ \hline & & & & \tau_1^{-1+\bar{n}_1}C_1 & & \\ & & & & & \ddots & \\ & & & & \tau_1^{-1}C_1 & & \\ & & & & C_1 & & \\ & & & & & \ddots & \\ & & & & & & \tau_s^{-1+\bar{n}_s}C_s \\ & & & & & & \ddots \\ & & & & & & \tau_s^{-1}C_s \\ & & & & & & C_s \end{array}\right]$$

generates a proper polyhedral A-invariant cone $\mathcal{K} := \mathrm{Cone}(C)$, since C is of full row rank and $AC = CP$ for

$$P = \left[\begin{array}{c|ccc|ccc|ccc} 1 & \varepsilon_2\mathbf{1}_{2n_2} & & \varepsilon_r\mathbf{1}_{2n_r} & \mathbf{c}_1 & \mathbf{d}_1 \cdots \mathbf{d}_1 & & \mathbf{c}_s & \mathbf{d}_s \cdots \mathbf{d}_s \\ - & P_2 & & & & & & & \\ & & \ddots & & & & & & \\ & & & P_r & & & & & \\ \hline & & & & \Gamma_1 & \tau_1 I_{k_1} & & & \\ & & & & & \Gamma_1 & & & \\ & & & & & & \ddots & & \\ & & & & & \tau_1 I_{k_1} & & & \\ & & & & & \Gamma_1 & & & \\ & & & & & & \ddots & & \\ & & & & & & & \Gamma_s & \tau_s I_{k_s} \\ & & & & & & & & \Gamma_s \\ & & & & & & & & \ddots \\ & & & & & & & & \tau_s I_{k_s} \\ & & & & & & & & \Gamma_s \end{array}\right]$$

with

$$\begin{aligned} \mathbf{c}_i &:= \left[\, \tfrac{\delta_i}{1+\delta_i} \quad \cdots \quad \tfrac{\delta_i}{1+\delta_i} \quad 1-\textstyle\sum_\ell b_\ell^{(i)} \,\right] > 0, \\ \mathbf{d}_i &:= \left[\, \tfrac{\delta_i}{1+\delta_i}-\tau_i \quad \cdots \quad \tfrac{\delta_i}{1+\delta_i}-\tau_i \quad 1-\textstyle\sum_\ell b_\ell^{(i)}-\tau_i \,\right] > 0, \end{aligned} \qquad i = 1, 2, \ldots, s.$$

To prove that A is $\mathcal{K}$-positive it is sufficient to show that the A-image of every column of C lies in $\mathrm{int}(\mathcal{K})$. By the same reasonings adopted in the proof of Theorem 3.2, the vector $\mathbf{e}_1$, which is the first column of C, is the only eigenvector of A lying in $\mathcal{K}$ and it belongs to $\mathrm{int}(\mathcal{K})$. Moreover, the A-image of the ith column of C, say $\mathbf{y}_i$, is a nonnegative linear combination of the columns of C involving the first column $\mathbf{e}_1$ (as all entries in the first row of P are nonzero). This implies that $\mathbf{y}_i$ belongs to the interior of $\mathcal{K}$ for every i. □

Remarks. The above theorem not only provides a complete spectral characterization of a matrix A for which a proper polyhedral cone $\mathcal{K}$ can be found such that A is $\mathcal{K}$-positive ($\mathcal{K}$-primitive), but it also particularizes to polyhedral cones the well-known result [6] that A is $\mathcal{K}$-primitive for some proper cone $\mathcal{K}$ if and only if there is a proper cone $\mathcal{K}'$ such that A is $\mathcal{K}'$-positive.

Moreover, up to now [6, 19] the pair of conditions (i) and (ii) on $\sigma(A)$ was known as necessary and sufficient for the existence of a proper cone $\mathcal{K}$ such that A is $\mathcal{K}$-positive [19]. Under this point of view, Theorem 3.3 represents a strengthening of this

result, and it proves that a matrix that is $\mathcal{K}$-positive for some proper cone $\mathcal{K}$ is also $\mathcal{K}'$-positive for some proper polyhedral cone $\mathcal{K}'$.

Example 3.2. Consider the following matrix:

$$A = \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & \frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{array}\right].$$

A satisfies conditions (i) and (ii) of Theorem 3.3, and its spectrum is $\sigma(A) = (\lambda_1, \sigma_1 + j\omega_1, \sigma_1 - j\omega_1) = (1, +j/2, -j/2)$. By assuming $\delta_1 = 1/2$, we get that

$$\begin{bmatrix} (1+1/2)0 & (1+1/2)1/2 \\ (1+1/2)(-1/2) & (1+1/2)0 \end{bmatrix}$$

is still Hurwitz stable. So, by applying the procedure described in the previous proof, we can construct a proper polyhedral cone left invariant by A, namely, the one generated by the columns of the matrix

$$C = \left[\begin{array}{c|cccc} 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & -\frac{9}{16} & 0 \\ 0 & 0 & -\frac{3}{4} & 0 & \frac{27}{64} \end{array}\right],$$

and by solving equation $AC = CP$ one can obtain the solution

$$P = \left[\begin{array}{c|ccc|c} 1 & \frac{\delta_1}{1+\delta_1} & \frac{\delta_1}{1+\delta_1} & \frac{\delta_1}{1+\delta_1} & 1 - \sum_{i=0}^{3} b_i \\ \hline 0 & 0 & 0 & 0 & b_0 \\ 0 & \frac{1}{1+\delta_1} & 0 & 0 & b_1 \\ 0 & 0 & \frac{1}{1+\delta_1} & 0 & b_2 \\ 0 & 0 & 0 & \frac{1}{1+\delta_1} & b_3 \end{array}\right]$$

with

$$b_0 = \frac{51}{128}, \quad b_1 = \frac{1}{16}, \quad b_2 = \frac{1}{3}, \quad b_3 = \frac{1}{9},$$

which implies A to be $\mathcal{K}$-positive and hence $\mathcal{K}$-primitive. Notice that A also leaves invariant the proper not polyhedral cone

$$\mathcal{K}_1 = \left\{ \mathbf{v} \in \mathbb{R}^n : \mathbf{v} = \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ b \end{bmatrix}, a^2 + b^2 \leq \alpha^2, \alpha \geq 0 \right\}$$

for which it is $\mathcal{K}_1$-positive.

4. Concluding remarks. To conclude the paper, it is worthwhile clarifying the relevance of our constructive procedure by means of a few additional comments and of an applicative example.

By referring to the constrained control problem, described in the introduction, consider the typical situation when the specific set S where we want to constrain the state evolution is naturally defined by the physical constraint acting on the system. As remarked in [8], such a set is not typically an invariant, and hence we are obliged to look for an invariant set (as large as possible) that is strictly included in S. To reach this goal, the constructive procedure described within the proof of Theorem 3.1 (or

Theorems 3.2 or 3.3, if we want to endow the invariant set with additional properties) can be exploited as follows. Assume without loss of generality that A is in real Jordan form (a situation that, as we have seen, can always be obtained, possibly by means of a suitable change of basis within the state space).

We can mechanically employ the algorithm and later rescale the generating vectors of the polyhedral cone $\mathcal{K}$ in order to make either $\mathcal{K}$ or any of its projections, according to the specific set S we started with, be included in S. Such a rescaling can be performed by applying to the block triangular matrix C, whose columns generate the cone $\mathcal{K}$, a suitable block diagonal matrix

$$D = \text{diag}\{d_1 I_m, d_2 I_{n_2}, \ldots, d_r I_{n_r}, \bar{d}_1 I_{2\bar{n}_1}, \ldots, \bar{d}_s I_{2\bar{n}_s}\},$$

where $m, n_2, \ldots, n_r, 2\bar{n}_1, \ldots, 2\bar{n}_s$ are the sizes of the Jordan blocks (see the proof of Theorem 3.1). In fact, if $\mathcal{K} = \text{Cone}(C)$ is invariant for the given matrix, then $\mathcal{K}' = \text{Cone}(DC)$ also is.

A significant improvement of such a procedure can be obtained by simply performing the above algorithm several times, by exploiting the freedom degrees in the choice of the vectors $\mathbf{v}_1^{(i)}$ (cf. the proof of Theorem 3.1), and hence obtaining different invariant polyhedral cones (or invariant polytope) included in S. As a final invariant region, then, we can assume the convex sum of the invariant regions obtained at every sweep of the algorithm.

Finally, it is also worthwhile to remark that the constructive procedure can lead to good results, even though performed only once, if suitably tuned to the specific region S we are considering. In fact, once we have chosen as first generating vector of $\mathcal{K}$ the first canonical vector (which is, of course, the dominant eigenvector, and hence must belong to every A-invariant region) the specific nonzero coefficients of the remaining columns of C, in particular, the aforementioned vectors $\mathbf{v}_1^{(i)}$, can be properly chosen in order to fall in S or on the boundary of S. Of course, also in this situation, a final rescaling will be necessary.

To better understand the meaning of these considerations, let us analyze the following example.

Consider the linear continuous time model of order 2, with a single input

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = A \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + Bu(t)$$

describing a direct current electric motor, where x_1 represents the armature current, x_2 the armature speed, and $u(t)$ the armature voltage. As in [8], we assume that due to load torque disturbances, the admissible fluctuations from the set point of the state are 100% of the nominal value for the armature current and 20% of the nominal value for the speed. No sensible fluctuations are assumed to affect the control variable u. As it is customary, we will resort to the Euler approximating system in order to solve the problem. This way, we come up with a discrete time system described by a pair of matrices A_d and B_d, for which reasonable values (taken from Example 3 in [8]) are the following ones:

$$A_d = \begin{bmatrix} 0.93 & -0.86 \\ 0.06 & 0.9915 \end{bmatrix}, \qquad B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The above conditions on the fluctuations of the system variables translate into the following constraining set for the state variables of the discretized system

$$S = \{(x_1, x_2) : |x_1| \leq 1,\ |x_2| \leq 0.2\}.$$

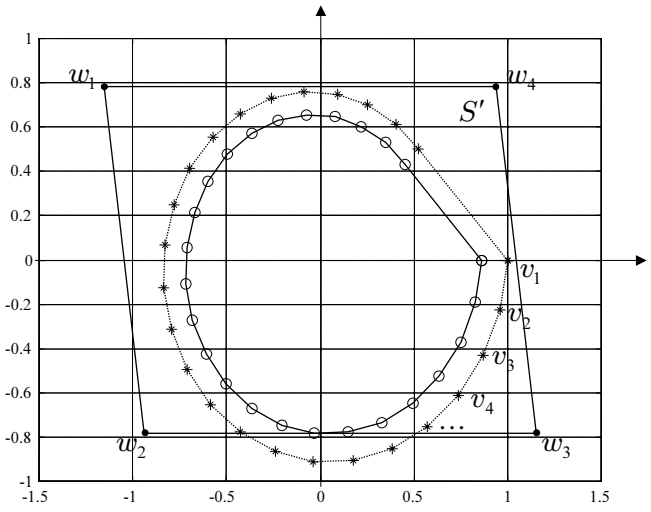


FIG. 1. *Invariant polytope inside S' and its rescaled version.*

A_d has two complex conjugate eigenvalues, which are strictly included in the (open) unitary circle, and its (real) Jordan form is

$$J = \begin{bmatrix} 0.9607 & 0.2251 \\ -0.2251 & 0.9607 \end{bmatrix}.$$

The matrix T, representing the coordinate of the Jordan basis with respect to the original basis, is

$$T = \begin{bmatrix} -0.9579 & -0.1309 \\ 0 & 0.2554 \end{bmatrix}.$$

As a consequence, the new state vector $z = T^{-1}x$ is constrained within the (convex) polytope S' defined by the following four vertices:

$$w_1 = \begin{bmatrix} -1.151 \\ 0.7831 \end{bmatrix}, \quad w_2 = \begin{bmatrix} -0.9369 \\ -0.7831 \end{bmatrix}, \quad w_3 = -w_1, \quad w_4 = -w_2.$$

Upon considering the extended matrix

$$\begin{bmatrix} 1 & 0 \\ 0 & J \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9607 & 0.2251 \\ 0 & -0.2251 & 0.9607 \end{bmatrix},$$

we can construct, by means of the procedure described within the proof of Theorem 3.1, an invariant cone. Starting from $\mathbf{v}_1 = [1\ 0]^T$ and by operating as in the proof of Theorem 3.1 (in the part corresponding to complex conjugate eigenvalues) we can obtain the recursive vector sequence $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots$.

It turns out that $\mathbf{v}_{25}$ is the first vector of the sequence that belongs to the convex polytope generated by the previous ones (see the dashed polytope in Figure 1, where the $*$'s represent the vectors of the sequence $\{\mathbf{v}_i\}$). Being interested only in the cone

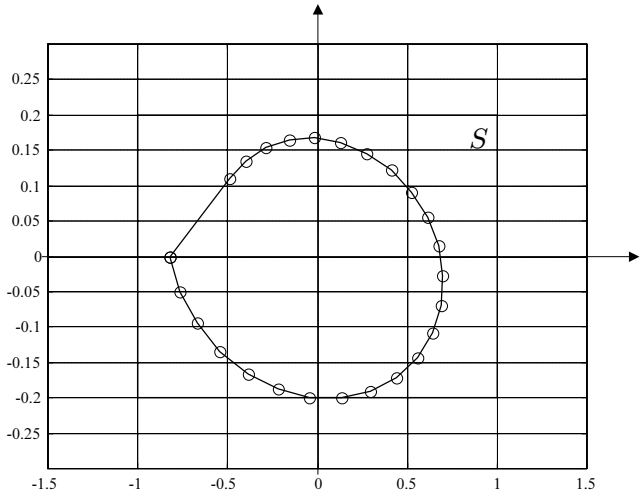


FIG. 2. *The resulting invariant polytope inside S.*

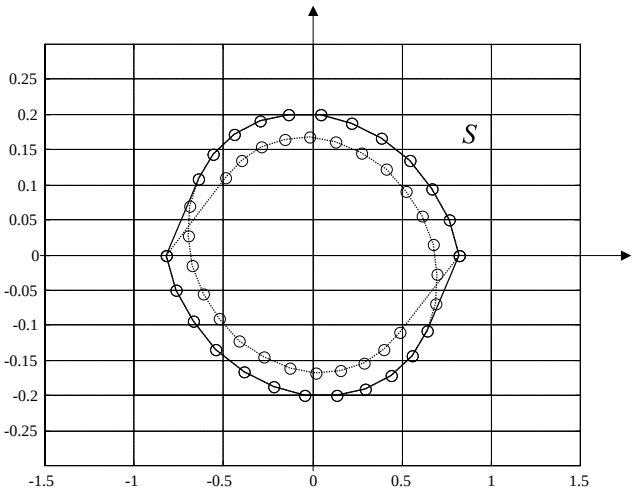


FIG. 3. *Invariant polytope inside S obtained after two runs of the proposed constructive procedure.*

projection on the plane including all these vectors, we can arrest the algorithm at this point. We have obtained, in this way, a convex polytope that is not included in S. However, by assuming as rescaling matrix

$$D = \begin{bmatrix} d & 0 \\ 0 & d \end{bmatrix}, \qquad d = 0.86,$$

we obtain an invariant polytope included in S'. (See the polytope drawn with continuous lines in Figure 1: in this case the $*$'s are mapped into the small circles o.) Of course, if we now refer to the representation with respect to the original state basis, we obtain the invariant polytope depicted in Figure 2.

Finally, if we perform our constructive procedure starting from a different initial vector, i.e., for $\mathbf{v}_1 = [-1\ 0]^T$, and follow the same steps just described, by putting

together the results of the two runs, we finally obtain the invariant region (included in S) depicted in Figure 3.

REFERENCES

[1] B.D.O. Anderson, M. Deistler, L. Farina, and L. Benvenuti, *Nonnegative realization of a linear system with nonnegative impulse response*, IEEE Trans. Circuits Systems I. Fund. Theory Appl., 43 (1996), pp. 134–142.

[2] G.P. Barker, *On matrices having an invariant cone*, Czechoslovak Math. J., 22 (1972), pp. 49–68.

[3] G.P. Barker, *Theory of cones*, Linear Algebra Appl., 39 (1981), pp. 263–291.

[4] G.P. Barker and R.E.L. Turner, *Some observations on the spectra of cone preserving maps*, Linear Algebra Appl., 6 (1973), pp. 149–153.

[5] A. Berman, M. Neumann, and R.J. Stern, *Nonnegative Matrices in Dynamic Systems*, John Wiley & Sons, New York, 1989.

[6] A. Berman and R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[7] G. Birkhoff, *Linear transformations with invariant cones*, Amer. Math. Monthly, 74 (1967), pp. 274–276.

[8] F. Blanchini, *Constrained control for systems with unknown disturbances*, Control Dyn. Syst. Adv. Theory Appl., 51 (1992), pp. 129–182.

[9] R.A. Brualdi and H.J. Ryser, *Combinatorial Matrix Theory*, Cambridge Univ. Press, Cambridge, UK, 1991.

[10] C. Burgat, A. Benzaouia, and S. Tarbouriech, *Positively invariant sets of discrete-time systems with constrained inputs*, Internat. J. Systems Sci., 21 (1990), pp. 1249–1271.

[11] F. Burns, M. Fiedler, and E. Haynswort, *Polyhedral cones and positive operators*, Linear Algebra Appl., 8 (1974), pp. 547–559.

[12] L. Elsner, *Monotonie und randspektrum bei vollstetigen operatoren*, Arch. Ration. Mech. Anal., 36 (1970), pp. 356–365.

[13] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[14] M.G. Krein and M.A. Rutman, *Linear operators leaving invariant a cone in Banach space*, Amer. Math. Soc. Trans. Ser., 10 (1950), pp. 199–325.

[15] H. Minc, *Nonnegative Matrices*, John Wiley & Sons, New York, 1988.

[16] J.W. Nieuwenhuis, *Some results about a Leontieff-type model*, in Frequency Domain and State Space Methods for Linear Systems, C.I. Byrnes and A. Lindquist, eds., Elsevier Science, Amsterdam, 1986, pp. 213–225.

[17] H.H. Schaefer, *Banach Lattices and Positive Operators*, Springer, New York, 1974.

[18] H.H. Schaefer, *Topological Vector Spaces*, 4th ed., Springer, New York, 1980.

[19] R. Stern and H. Wolkowicz, *Invariant ellipsoidal cones*, Linear Algebra Appl., 150 (1991), pp. 81–106.

[20] B.S. Tam, *On the distinguished eigenvalues of a cone-preserving map*, Linear Algebra Appl., 131 (1990), pp. 17–37.

[21] B.S. Tam and H. Schneider, *On the core of a cone-preserving map*, Trans. Amer. Math. Soc., 343 (1994), pp. 479–524.

[22] S. Tarbouriech and C. Burgat, *Positively invariant sets for continuous-time systems with the cone-preserving property*, Internat. J. Systems Sci., 24 (1993), pp. 1037–1047.

[23] A.A. ten Dam, *Representations of dynamic systems described by behavioural inequalities*, in the Proceedings of ECC'93, Groningen, The Netherlands, 1993, pp. 1780–1783.

[24] J.S. Vandergraft, *Spectral properties of matrices which have invariant cones*, SIAM J. Appl. Math., 16 (1968), pp. 1208–1222.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 472–478

HOW CLOSE CAN THE LOGARITHMIC NORM OF A MATRIX PENCIL COME TO THE SPECTRAL ABSCISSA?*

INMACULADA HIGUERAS† AND BERTA GARCÍA-CELAYETA†

Abstract. Given a least upper bound norm, the usefulness of the concept of logarithmic norm depends on how closely the logarithmic norm approximates the spectral abscissa. To study this problem, Ström introduced in 1975 the concepts of logarithmically optimal norm and ε-logarithmically optimal norm with respect to a matrix A. Recently Higueras and García-Celayeta have done an extension of the concept of logarithmic norm for matrix pencils and, in a similar way, the usefulness of the concept depends on how closely the logarithmic norm of the pencil approximates the spectral abscissa. In this paper we study this problem and extend the concepts by Ström to matrix pencils.

Key words. matrix pencil, logarithmic norm, Lyapunov stability, differential algebraic system

AMS subject classifications. 15A22, 34D20, 65L07

PII. S0895479898346296

1. Introduction.

The concept of logarithmic norm for a matrix was introduced in 1958 by Dahlquist and Lozinskij as a tool for studying the growth of solutions of ODEs and the error growth in discretization methods for their approximate solution. For a matrix A, the logarithmic norm is defined by

$$\mu[A] = \lim_{\Delta \to 0^+} \frac{\|I + \Delta A\| - 1}{\Delta}.$$

The norm here is generally assumed to be a least upper bound (l.u.b.) norm. The actual numerical value of $\mu[A]$ depends on the norm on which $\mu[A]$ is based. If $\alpha(A)$ denotes the spectral abscissa of A (i.e., the maximum real part of the eigenvalues of A), it is known that given a linear constant coefficient ODE, $x'(t) = Ax(t)$, the solution is asymptotically stable if and only if $\alpha(A) < 0$. A matrix A with $\alpha(A) < 0$ is called a stable matrix. The solution is stable if and only if $\alpha(A) \leq 0$ and no eigenvalues λ with $\mathrm{Re}(\lambda) = 0$ are defective. An eigenvalue λ of A is defective if it has a noncomplete set of eigenvectors. In terms of elementary divisors, an eigenvalue is nondefective if the elementary divisors associated with it are simple. A matrix satisfying these conditions will be called a weakly stable matrix.

Given a linear variable coefficient ODE, $x'(t) = A(t)x(t)$, the eigenvalues of the matrix do not give any information about the asymptotic stability, but in terms of the logarithmic norm we have the bound

$$\|x(t)\| \leq e^{\int_0^t \mu[A(t)]dt} \|x(0)\|, \qquad t \geq 0.$$

Thus if $\mu[A(t)] < 0$, the solutions are asymptotically stable. It is known that $\alpha(A) \leq \mu[A]$. For details on logarithmic norms, see [2].

*Received by the editors October 27, 1998; accepted for publication (in revised form) by V. Mehrmann January 12, 2000; published electronically August 9, 2000. This work was supported by the Gobierno de Navarra, project "Técnicas de aproximación en la resolución de problemas diferenciales y en la representación de superficies" (O.F. 508/1997).

http://www.siam.org/journals/simax/22-2/34629.html

†Departamento de Matemática e Informática, Universidad Pública de Navarra, 31006 Pamplona, Spain (higueras@unavarra.es, berta@unavarra.es).

The usefulness of the concept of logarithmic norm depends on how closely $\mu[A]$ approximates $\alpha(A)$. In [8] Ström introduces the concept of logarithmic inefficiency of a norm with respect to the matrix A as

$$q[A] = \mu[A] - \alpha(A) \geq 0 \tag{1.1}$$

and gives the following definition.

DEFINITION 1.1. *A norm is called logarithmically optimal with respect to A if $q[A] = 0$; it is called logarithmically ε-efficient if $q[A] \leq \varepsilon$.*

Given a vector norm $\|\cdot\|$ and a nonsingular matrix T, we define the norm $\|x\|_T = \|Tx\|$. The corresponding l.u.b. norm satisfies $\|A\|_T = \|TAT^{-1}\|$ and the logarithmic norm $\mu_T[A] = \mu[TAT^{-1}]$. It is said that $\|\cdot\|$ and $\|\cdot\|_T$ are similar.

In [8] the following theorem is proved.

THEOREM 1.2. *Given $\varepsilon > 0$, any monotonic norm $\|\cdot\|$ and the $n \times n$ matrix A, we may find a logarithmically ε-efficient norm similar to $\|\cdot\|$.*

For the rest of the paper we assume that the norms are monotonic. Thus from Theorem 1.2, if the solution is asymptotically stable, then there exists a norm such that $\mu[A] < 0$. In terms of stable matrices, the following theorem can be stated.

THEOREM 1.3. *A matrix A is stable if and only if $\mu[A] < 0$ for some norm.*

Another theorem proved in [8] is the following one.

THEOREM 1.4. *Given A, there exists a logarithmically optimal norm if and only if no eigenvalue λ of A with $\mathrm{Re}\lambda = \alpha(A)$ is defective. The logarithmically optimal norm can be chosen in a similar manner as the euclidean norm.*

Thus from Theorem 1.4, if the solution is stable, then there exists a norm such that $\mu[A] \leq 0$. In terms of weakly stable matrices, the following theorem can be stated.

THEOREM 1.5. *A matrix A is weakly stable if and only if $\mu[A] \leq 0$ for some norm.*

We consider now linear constant coefficient differential algebraic equations

$$Ax'(t) + Bx(t) = f(t)\,. \tag{1.2}$$

In order to get uniqueness for the solution, the pencil (A, B) is assumed to be regular, i.e., there is a value λ such that $\det(\lambda A + B) \neq 0$. Observe that for ODEs we have the pencil $(I, -A)$. If k is the index of the pencil, then we have

$$\mathbb{R}^n = \ \mathrm{Ker}(\hat{A}^k) \oplus \mathrm{Im}(\hat{A}^k)\,,$$

with $\hat{A} = (cA + B)^{-1}A$, $\hat{B} = (cA + B)^{-1}B$ for any c such that $cA + B$ is regular. To get the solution, we decompose

$$x(t) = \hat{A}^D\hat{A}x(t) + (I - \hat{A}^D\hat{A})x(t) = y(t) + z(t),$$

where A^D is the Drazin inverse. One of the properties of the Drazin inverse is that $\hat{A}^D\hat{A}$ is a projector onto $\mathrm{Im}(\hat{A}^k)$ along $\mathrm{Ker}(\hat{A}^k)$. In the homogeneous case, $z(t) = 0$ and the solution $x(t) \in \mathrm{Im}(\hat{A}^D\hat{A})$. Actually the homogeneous DAE (1.2) is an ODE on the lower dimension subspace $\mathrm{Im}(\hat{A}^D\hat{A})$. For details, see [6], [1].

For a regular pencil, $\det(\lambda A + B)$ is a polynomial of degree less than or equal to n. The complex values λ such that $\det(\lambda A + B) = 0$ are called finite eigenvalues of the pencil. Observe that not every pencil has eigenvalues. It is said that infinite is an eigenvalue of the pencil (A, B) if $\det(A) = 0$. ODEs have only finite eigenvalues; DAEs have infinite eigenvalues and may have finite ones. We will consider only the set

of finite eigenvalues of the pencil and denote it by $\sigma(A,B)$, finite spectrum of (A,B). For $\lambda \in C$, the vectors $v \in C^n$ such that $(\lambda A + B)v = 0$ are called eigenvectors associated with the finite eigenvalue λ. The eigenvectors associated with the infinite eigenvalue are all the vectors in $\mathrm{Ker}(A)$. The spectral radius, denoted by $\rho(A,B)$, is defined as

$$\rho(A,B) = \max\{|\lambda| \,/\, \lambda \in \sigma(A,B)\}\,.$$

The spectral abscissa of the pencil (A,B), denoted by $\alpha(A,B)$, is defined as

$$\alpha(A,B) = \max\{\ \mathrm{Re}\ (\lambda) \,/\, \lambda \in \sigma(A,B)\}\,.$$

For a regular matrix pencil (A,B), there are regular matrices P and Q such that

$$A = Q\begin{pmatrix} I_r & \\ & U \end{pmatrix} P, \qquad B = Q\begin{pmatrix} J & \\ & I_{n-r} \end{pmatrix} P,$$

where U is a nilpotent $(n-r)\times(n-r)$ matrix and J, an $r \times r$ matrix, is formed by Jordan blocks associated with the finite eigenvalues of the pencil. The index of nilpotency of the DAE (or index) is the order of nilpotency of the matrix U. The pencil $(\mathrm{diag}(I_r,U), \mathrm{diag}(J,I_{n-r}))$ is called the Kronecker canonical form of (A,B). Denoting $M = (cI_r + J)^{-1}$, $N = (cU + I_{n-r})^{-1}U$ we obtain

$$\hat{A} = P^{-1}\begin{pmatrix} M & \\ & N \end{pmatrix} P, \qquad \hat{B} = P^{-1}\begin{pmatrix} I_r - cM & \\ & I_{n-r} - cN \end{pmatrix} P \tag{1.3}$$

with M regular and N nilpotent with order the index of nilpotency of the pencil. A simple computation gives $-M^{-1} + cI_r = -J$.

For the homogeneous DAE (1.2) the solution is asymptotically stable if and only if $\alpha(A,B) < 0$. A matrix pencil (A,B) such that $\alpha(A,B) < 0$ will be called a stable matrix pencil. The solution is stable if and only if $\alpha(A,B) \le 0$ and the elementary divisors associated with the eigenvalues λ of the pencil (A,B) with $\mathrm{Re}(\lambda) = 0$ are simple [4]. A matrix pencil (A,B) satisfying these conditions will be called a weakly stable matrix pencil.

For homogeneous linear variable coefficient DAEs, the eigenvalues of the pencil $(A(t),B(t))$ do not give any information about the asymptotic stability. In [3] a definition of logarithmic norm for a matrix pencil (A,B) was given,

$$\mu_V[A,B] = \lim_{\Delta\to 0^+} \frac{\|A, A+\Delta B\|_V - 1}{\Delta},$$

where V is a subspace such that

$$V \neq \{0\} \qquad \text{and} \qquad V \cap \ker(A) = 0\,, \tag{1.4}$$

and

$$\|A,B\|_V = \max_{x\in V\,,\, x\neq 0} \frac{\|Bx\|}{\|Ax\|}\,. \tag{1.5}$$

A subspace V satisfying (1.4) will be called admissible subspace. Observe that for $V = \mathbb{R}^n$, $\mu_V[I,-B] = \mu[B]$. If A is regular, $\mu_V[A,B] = \mu_V[I, BA^{-1}]$. If the norm is an inner product norm, then

$$\mu_V[A,B] = \max_{\substack{Ax\neq 0\\ x\in V}} \frac{\langle Ax, -Bx\rangle}{\langle Ax, Ax\rangle}\,.$$

If the eigenvectors of the pencil (A, B) are in V (or at least any eigenvector corresponding to an eigenvalue which gives the spectral radius), then

$$\alpha(A, B) \leq \mu_V[A, B]\,.$$

For $x(t)$ the solution of the homogeneous linear variable coefficient DAE, and for any admissible V such that $x(t) \in V$, we have the bound

$$\|A(t)x(t)\| \leq e^{\int_0^t \mu_V[A(t),B(t)-A'(t)]\,dt}\|Ax(0)\|, \qquad t \geq 0, \tag{1.6}$$

for any consistent initial condition $x(0)$. For details on logarithmic norms for matrix pencils, see [3].

In particular, for the homogeneous constant coefficient case, the solution is in $\mathrm{Im}(\hat{A}^D\hat{A})$, and it can be proved [3] that it satisfies (1.4); thus we can take $V = \mathrm{Im}(\hat{A}^D\hat{A})$. Inequality (1.6) shows that if $\mu_V[A, B] < 0$ for $V = \mathrm{Im}(\hat{A}^D\hat{A})$, then

$$\lim_{t\to\infty} \|Ax(t)\| = 0,$$

but as

$$x(t) = \hat{A}^D\hat{A}x(t) = \hat{A}^D(cA + B)^{-1}Ax(t)\,,$$

it is equivalent to the asymptotic stability of the solution. Again the usefulness of the concept of logarithmic norm depends on how closely $\mu_V[A, B]$ approximates $\alpha(A, B)$.

For the seminorm (1.5) it is also possible to give the concept of similarity. Given a vectorial norm $\|\cdot\|$ and nonsingular matrices T and $\tilde{T}$, we have

$$\|A, B\|_{V,T} = \max_{x\in V\,,\,x\neq 0} \frac{\|Bx\|_T}{\|Ax\|_T} = \max_{x\in V\,,\,x\neq 0} \frac{\|TBx\|}{\|TAx\|} = \max_{y\in \tilde{T}\,V\,,\,y\neq 0} \frac{\|TB\tilde{T}^{-1}y\|}{\|TA\tilde{T}^{-1}y\|};$$

thus we can define

$$\|A, B\|_{V,T} = \|TA, TB\|_V = \|TA\tilde{T}^{-1}, TB\tilde{T}^{-1}\|_{\tilde{T}\,V},$$

where $\tilde{T}\,V = \{\tilde{T}x \,/\, x \in V\}$. We also get

$$\mu_{V,T}[A, B] = \mu_V[TA, TB] = \mu_{\tilde{T}\,V}[TA\tilde{T}^{-1}, TB\tilde{T}^{-1}]\,.$$

The rest of the paper is organized as follows. In section 2 we extend the concepts of logarithmically ε-efficiency norm and logarithmically optimal norm and prove the analogous Theorems 1.2 and 1.4 for matrix pencils. In section 3, we use these theorems to prove in an easier way a result in [7].

2. Logarithmic efficiency. We begin with the extension of (1.1).

DEFINITION 2.1. *The logarithmic inefficiency of a norm and a subspace V with respect to the matrix pencil (A, B) is given by*

$$q_V[A, B] = \mu_V[A, B] - \alpha(A, B)\,.$$

Observe that now we have $q_V[A, B] \geq 0$ only if any eigenvector corresponding to an eigenvalue that gives the spectral radius is in V.

DEFINITION 2.2. *A norm and a subspace V are logarithmically optimal with respect to (A, B) if and only if $q_V[A, B] = 0$ and logarithmically ε-efficient if $0 \leq q_V[A, B] \leq \varepsilon$.*

We state and prove the extension to Theorem 1.2.

THEOREM 2.3. *Consider the pencil (A,B) and the admissible subspace $V = \mathrm{Im}(A^DA)$ if A and B commute and $V = \mathrm{Im}(\hat{A}^D\hat{A})$ otherwise. Given $\varepsilon > 0$ and any monotonic norm $\|\cdot\|$, there exists a vectorial norm similar to $\|\cdot\|$ such that $0 \le q_V[A,B] \le \varepsilon$.*

Proof. If the matrices A and B do not commute, consider the matrices $\hat{A} = (cA+B)^{-1}A$ and $\hat{B} = (cA+B)^{-1}B$ for any $c \in \mathbb{R}$ such that $(cA+B)$ is regular. Thus $\mu_V[A,B] = \mu_{V,(cA+B)}[\hat{A},\hat{B}]$. Using (1.3), the Drazin inverse of $\hat{A}$ is

$$\hat{A}^D = P^{-1}\begin{pmatrix} M^{-1} & \\ & 0 \end{pmatrix} P\,.$$

We consider $V = \mathrm{Im}(\hat{A}^D\hat{A})$ and compute

$$\begin{aligned}
\mu_{V,(cA+B)}[\hat{A},\hat{B}] =& \mu_{V,(cA+B)}\left[P^{-1}\begin{pmatrix} M & \\ & N \end{pmatrix}P, P^{-1}\begin{pmatrix} I_r - cM & \\ & I_{n-r} - cN \end{pmatrix}P\right]\\
=& \mu_{V,(cA+B)P^{-1}}\left[\begin{pmatrix} M & \\ & N \end{pmatrix}P, \begin{pmatrix} I_r - cM & \\ & I_{n-r} - cN \end{pmatrix}P\right]\\
=& \mu_{V,(cA+B)P^{-1}}\left[\begin{pmatrix} M & \\ & 0 \end{pmatrix}P, \begin{pmatrix} I_r - cM & \\ & 0 \end{pmatrix}P\right]\\
=& \mu_{\mathbb{R}^r,(cA+B)P^{-1}}[M, I_r - cM] = \mu_{\mathbb{R}^r,(cA+B)P^{-1}}[I_r, (I_r - cM)M^{-1}]\\
=& \mu_{(cA+B)P^{-1}}[-M^{-1} + cI_r],
\end{aligned}$$

where for $x \in \mathbb{R}^r$ and W, a regular $n \times n$ matrix, $\|x\|_W$, is defined as $\|x\|_W = \|(x,0)^t\|_W$. For matrices, we know by Theorem 1.2 that there exists a logarithmically ε-efficient norm in $\mathbb{R}^r$ such that

$$\mu[-M^{-1} + cI_r] - \alpha(-M^{-1} + cI_r) < \varepsilon\,.$$

We simply have to extend this norm to $\mathbb{R}^n$. To get the desired result, observe that

$$\alpha(A,B) = \alpha(\hat{A},\hat{B}) = \alpha(M, I_r - cM) = \alpha(-M^{-1} + cI_r). \qquad \square$$

COROLLARY 2.4. *If $\alpha(A,B) < 0$, then there exists a vectorial norm and a subspace V, namely, $V = \mathrm{Im}(\hat{A}^D\hat{A})$ such that*

$$\mu_V[A,B] < 0.$$

It is not strange to have to work on the subspace $V = \mathrm{Im}(\hat{A}^D\hat{A})$ because actually with the homogeneous DAE we are working in this lower-dimension subspace.

In terms of stable pencils, we can state the following result analogous to Theorem 1.3.

THEOREM 2.5. *The pencil (A,B) is stable if and only if for $V = \mathrm{Im}(\hat{A}^D\hat{A})$ $\mu_V[A,B] < 0$ for some norm.*

We can also extend Theorem 1.4 as follows.

THEOREM 2.6. *Consider the pencil (A,B) and the admissible subspace $V = \mathrm{Im}(A^DA)$ if A and B commute and $V = \mathrm{Im}(\hat{A}^D\hat{A})$ otherwise. There exists a logarithmically optimal norm if and only if no eigenvalue λ with $\mathrm{Re}\lambda = \alpha(A,B)$ is defective. The logarithmically optimal norm may be chosen similar to the euclidean norm.*

Proof. As in Theorem 2.3, $\mu_V[A,B] = \mu_{(cA+B)P^{-1}}[-M^{-1} + cI]$; remember that $\alpha(A,B) = \alpha(-M^{-1} + cI) = \alpha(-J)$. We also have to relate the elementary divisors

of the eigenvalues λ of (A, B) with the elementary divisors of the eigenvalues λ of $-J$. As a regular pencil is strictly equivalent to its Kronecker canonical form and strictly equivalent pencils have the same finite and infinite elementary divisors [5], the elementary divisors (finite and infinite) of the pencil (A, B) are the same as the elementary divisors (finite and infinite) of the pencil $(\mathrm{diag}(I_r, N), \mathrm{diag}(J, I_{n-r}))$. The finite elementary divisors of this pencil are the elementary divisors of $-J$. Thus, if λ is a finite eigenvalue of (A, B), the elementary divisors associated with it, necessarily finite, are the same as those of $-J$. Now by Theorem 1.4 we have the desired result. ∎

In terms of weakly stable pencils, we can state the following result analogous to Theorem 1.5.

THEOREM 2.7. *The pencil (A, B) is weakly stable if and only if for $V = \mathrm{Im}(\hat{A}^D \hat{A})$ $\mu_V[A, B] \leq 0$ for some norm.*

3. Some other applications. We have assumed that the pencil (A, B) is regular, but we can drop this condition if, for example, the Kronecker canonical form is not needed.

PROPOSITION 3.1. *Given any monotonic norm $\|\cdot\|$ and the pencil $(\Pi, -M)$, such that*

$$\Pi = P \begin{pmatrix} I_r & \\ & 0 \end{pmatrix} P^{-1}, \qquad M = P \begin{pmatrix} \tilde{M} & \\ & 0 \end{pmatrix} P^{-1}$$

with $\tilde{M}$ a $r \times r$ matrix such that $\alpha(\tilde{M}) < 0$, there exists a norm similar to $\|\cdot\|$ such that for $V = \mathrm{Im}(\Pi)$

$$\mu_V[\Pi, -M] < 0.$$

Proof. A simple computation gives

$$\mu_V[\Pi, -M] = \mu_{\mathbb{R}^r, P}[I, -\tilde{M}] = \mu[\tilde{M}]$$

and we can apply Theorem 1.2. ∎

As a corollary we can obtain Lemma 4.2 in [7].

COROLLARY 3.2. *Given the $m \times m$ real matrices M and Π such that $\Pi^2 = \Pi$, $M = M\Pi = \Pi M$ and $\mathrm{rank}\Pi = r$. Let M have r nontrivial eigenvalues $\lambda_1, \ldots, \lambda_r$ and let them all have negative real part. Then there is a constant $\beta > 0$ and a regular matrix C such that*

$$\langle Mz, z\rangle_C \leq -\beta \langle z, z\rangle_C \qquad \forall z \in \mathrm{Im}(\Pi), \tag{3.1}$$

where $\langle x, y\rangle_C = y^t C^t C x$.

Proof. Expression (3.1) is simply $\mu_V[\Pi, -M] \leq -\beta < 0$ for an inner product norm and $V = \mathrm{Im}(\Pi)$. ∎

REFERENCES

[1] S.L. Campbell and C.D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Dover, New York, 1991.
[2] K. Dekker and J.G. Verwer, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, Amsterdam, 1984.
[3] I. Higueras and B. García-Celayeta, *Logarithmic norms for matrix pencils*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 646–666.
[4] I. Higueras and B. García-Celayeta, *Stability for Linear DAEs*, Preprint 15, Departamento de Matemática e Informática, Universidad Pública de Navarra, Pamplona, Spain, 1997.
[5] F.R. Gantmacher, *The Theory of Matrices*. Vol. II, Chelsea, New York, 1989.
[6] E. Griepentrog and R. März, *Differential algebraic equations and their numerical treatment*, Teubner Texte zur Mathematik 88, Teubner, Leipzig, Germany, 1986.
[7] R. März, *On Quasilinear Index* 2 *Differential Algebraic Equations*, Preprint 269, Fachbereich Mathematik, Humboldt Universität zu Berlin, Germany, 1990.
[8] T. Ström, *On logarithmic norms*, SIAM. J. Numer. Anal., 12 (1975), pp. 741–753.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 479–500

UPDATING A GENERALIZED URV DECOMPOSITION*

MICHAEL STEWART† AND PAUL VAN DOOREN‡

Abstract. An updating scheme for a quotient type generalization of a URV decomposition of two matrices is introduced. This decomposition allows low complexity updating as rows are added to two rectangular matrices, determining the dimension of three distinct subspaces. One of these subspaces is the intersection of the range space of the two matrices—information which leads to a potential application in subspace algorithms for system identification.

Key words. generalized SVD, URV decomposition, system identification

AMS subject classifications. 65F15, 65F20

PII. S0895479897320460

1. Introduction.

The quotient singular value decomposition of two matrices, A and B, with an equal number of rows m, has been described in several ways. The justification for the name is most obvious when A and B are both square and of the same size and when A has full rank. Suppose an application demands knowledge of the singular values of $A^{-1}B$. It is well known in the context of the generalized eigenvalue problem, in which the goal is to find the eigenvalues of $A^{-1}B$, that the best approach is to compute the eigenvalues by applying orthogonal transformations to A and B without explicitly computing $A^{-1}B$. This also applies to the singular value problem and, instead of computing $A^{-1}B$, a more reasonable approach is to directly compute invertible X and orthogonal V_A and V_B such that

$$X^{-1}AV_A = \Sigma_A, \qquad X^{-1}BV_B = \Sigma_B,$$

where Σ_B and Σ_A are diagonal. This solves the problem, since

$$A^{-1}B = V_A\Sigma_A^{-1}\Sigma_B V_B^T$$

is clearly the singular value decomposition of $A^{-1}B$. If X is required to be orthogonal, then the best that can be done is to make Σ_A and Σ_B triangular. An appropriate choice of orthogonal X, V_A, and V_B guarantees that $\Sigma_A^{-1}\Sigma_B$ will be diagonal.

More generally, when A and B are possibly rank deficient $m \times n_a$ and $m \times n_b$ matrices, the generalized SVD [10, 13] has been defined by

$$X^{-1}AV_1 = \begin{bmatrix}\Sigma_A \\ 0\end{bmatrix}\begin{matrix} r \\ m-r \end{matrix}, \qquad X^{-1}BV_2 = \begin{bmatrix}\Sigma_B \\ 0\end{bmatrix}\begin{matrix} r \\ m-r \end{matrix}, \tag{1.1}$$

where

$$\Sigma_A = \begin{bmatrix} I_A & & \\ & S_A & \\ & & 0_A \end{bmatrix}, \qquad \Sigma_B = \begin{bmatrix} 0_B & & \\ & S_B & \\ & & I_B \end{bmatrix}$$

*Received by the editors April 23, 1997; accepted for publication (in revised form) by L. Eldén September 11, 1998; published electronically August 9, 2000. This work was supported by ARPA grant 60NANB2D1272 and NSF grant CCR-9209349.

http://www.siam.org/journals/simax/22-2/32046.html

†Computer Sciences Laboratory, RSISE, Australian National University, Canberra, ACT 0200, Australia (stewart@discus.anu.edu.au).

‡CESAME, Université Catholique de Louvain, Louvain-la-Neuve, Belgium (vdooren@anma.ucl.ac.be).

with diagonal positive definite S_A and S_B satisfying $S_A^2 + S_B^2 = I$ and where r is the rank of $\begin{bmatrix} A & B \end{bmatrix}$. The partitionings are such that S_A and S_B are the same size, r_3. The identity matrices I_B and I_A are $r_2 \times r_2$ and $(r_1 - r_3) \times (r_1 - r_3)$, where r_1 is the rank of A. The zero blocks 0_A and 0_B are $(r - r_1) \times (n_a - r_1)$ and $(r - r_2 - r_3) \times (n_b - r_2 - r_3)$. The decomposition reveals that r_3 is the dimension of the intersection of the range spaces of A and B.

In the rectangular case in which A has full rank the decomposition reveals the singular values of $A^\dagger B$, where $A^\dagger$ denotes the pseudoinverse of A. If A is rank deficient, then the decomposition reveals singular values associated with a quotient formed from the B-weighted pseudoinverse of A [4, 2].

An early development of the generalized SVD was given in [10]. A general description suitable for adaptation to a URV decomposition is as follows: the $m \times (n_a + n_b)$ matrix $\begin{bmatrix} A & B \end{bmatrix}$ is decomposed as

$$U^T \left[A \parallel B \right] V = U^T \left[A \parallel B \right] \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \left[\begin{array}{ccc|ccc} R_{11} & 0 & & S_{13} & R_{14} & 0 \\ 0 & 0 & & R_{23} & 0 & 0 \\ 0 & 0 & & 0 & 0 & 0 \end{array}\right], \tag{1.2}$$

where R_{11} is $r_1 \times r_1$ and upper triangular with full rank and R_{23} is $r_2 \times r_2$ and upper triangular with full rank. The rectangular block

$$R_{14} = \begin{bmatrix} \hat{R}_{14} \\ 0 \end{bmatrix}$$

is $r_1 \times r_3$ with full column rank r_3, and $\hat{R}_{14}$ is square and upper triangular. Clearly the first r_3 columns of U form a basis for the intersection of the range spaces. Since the rank of A is clearly r_1 and the rank of B is $r_2 + r_3$, this decomposition reveals the same rank information as the quotient SVD.

Further, if we define $\hat{R}_{11}$ to be the $r_3 \times r_3$ leading principal submatrix of R_{11} and

$$\hat{R}_{11}^{-1} \hat{R}_{14} = V_{11} \Sigma_R V_{14}^T$$

to be the SVD of $\hat{R}_{11}^{-1} \hat{R}_{14}$, then

$$\hat{R}_{11} V_{11} = \hat{R}_{14} V_{14} \Sigma_R^{-1},$$

and consequently there exists an orthogonal U_R such that $U_R^T \hat{R}_{11} V_{11}$ and $U_R^T \hat{R}_{14} V_{14}$ are both upper triangular. Applying U_R, V_{11}, and V_{14} to the relevant rows and columns of (1.2) will maintain the structure of (1.2) while ensuring that $\hat{R}_{11}^{-1} \hat{R}_{14}$ will be diagonal. Note that the singular values of $\hat{R}_{11}^{-1} \hat{R}_{14}$ are not changed by these further orthogonal transformations.

If U and V are required to be orthogonal, then this is the most condensed form one can obtain. However, if we define

$$X^{-1} = \begin{bmatrix} R_{11}^{-1} & -R_{11}^{-1} S_{13} R_{23}^{-1} & 0 \\ 0 & R_{23}^{-1} & 0 \\ 0 & 0 & I \end{bmatrix} U^T,$$

then

$$X^{-1} \left[A \parallel B \right] V = \left[\begin{array}{ccc|ccc} I & 0 & 0 & 0 & \hat{R}_{11}^{-1} \hat{R}_{14} & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

If the original orthogonal transformations were chosen so that $\hat{R}_{11}^{-1}\hat{R}_{14}$ is diagonal and positive definite, then clearly $\hat{R}_{11}^{-1}\hat{R}_{14} = S_A^{-1}S_B$. Thus up to permutations and scaling by S_A^{-1} we recover the quotient SVD as presented in [10] and we can conclude that the quotient singular values of A and B are the singular values of $\hat{R}_{11}^{-1}\hat{R}_{14}$ even when $\hat{R}_{11}^{-1}\hat{R}_{14}$ is not diagonal. This is the justification for viewing the decomposition as a quotient type generalization of the URV decomposition.

In this paper we do not require diagonality of $\hat{R}_{11}^{-1}\hat{R}_{14}$ and we present an algorithm to efficiently update a rank revealing decomposition that is related to (1.2) when rows are added to the matrices A and B. An obvious application is recursive identification of MIMO systems. The algorithm in [7] requires the intersection of the range spaces of two matrices and may be adapted for use with the decomposition. A summary of the main idea will be presented in section 5. Further details are in [12].

Other papers have considered updating for a quotient generalization of the ULV decomposition [5] in the case in which A and B are $n_a \times m$ and $n_b \times m$ with $n_a, n_b \geq m$ and the update involves the addition of rows to A and B. In the formulation chosen in this paper in which the matrices have an equal number of rows, this is equivalent to updating under the addition of columns to A and B. The method was first proposed in [5] for the case in which A has full rank and extended in [6] to the rank deficient case. A natural application for these decompositions is in prewhitening of colored noise in signal processing [4]. Because of the assumptions on the dimensions of the matrices in [5] and [6] and the difference between updating under the addition of columns and rows, the algorithms considered in this paper are substantially different from the previous work on generalized ULV decompositions.

The set of all rank deficient matrices is a subset of measure zero in the set of all matrices. If $m \geq n_a + n_b$, then A and B can have nonempty range space intersection only when $\begin{bmatrix} A & B \end{bmatrix}$ is rank deficient. Moreover, A or B can be rank deficient only when $\begin{bmatrix} A & B \end{bmatrix}$ is rank deficient. It follows that when $m \geq n_a + n_b$ the decomposition (1.2) has the form

$$U^T \left[A \parallel B \right] V = U^T \left[A \parallel B \right] \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \left[\begin{array}{c||c} R_{11} & S_{13} \\ 0 & R_{23} \\ 0 & 0 \end{array}\right] \tag{1.3}$$

except on a measure zero set. This represents the special case of (1.2) when $r_1 = n_a$, $r_2 = n_b$, and $r_3 = 0$. Because of numerical errors or noisy data, we can always expect a quotient URV to have the trivial structure (1.3).

Thus an exact quotient URV gives no real information about the relation between the numerical range spaces of A and B. Instead of computing the exact quotient URV, we will attempt to compute a rank-revealing decomposition that shows when small perturbations give a nontrivial quotient URV structure of the type (1.2). The perturbations we allow will take the form of small nonzero elements in some of the blocks of (1.2) that were previously zero. We constrain U and V to be orthogonal and drop the requirement that $\hat{R}_{11}^{-1}\hat{R}_{14}$ must be diagonal. The result is

$$U^T \left[A \parallel B \right] \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \left[\begin{array}{cc||ccc} R_{11} & E_{12} & S_{13} & R_{14} & E_{15} \\ 0 & E_{22} & R_{23} & E_{24} & E_{25} \\ 0 & E_{32} & 0 & F_{34} & E_{35} \\ 0 & E_{42} & 0 & 0 & F_{45} \\ 0 & F_{52} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{1.4}$$

The blocks R_{11} and R_{23} are square and upper triangular. In (1.2), R_{14} was upper triangular with full column rank. To make the updating easier, we modify this in (1.4): we allow R_{14} to have potentially large elements arranged in the form of an upper triangular matrix that has had its columns reversed. The rest of the elements can be nonzero but are constrained to be below a prespecified tolerance. The elements below the cross diagonal are kept small enough that they will not cause the block to become rank deficient when the triangular part R_{14} is kept suitably well conditioned by an appropriate method for deflating small singular values. Letting r represent a potentially large element and e represent an element which is small relative to the tolerance, the 4×3 case of R_{14} looks like

$$R_{14} = \begin{bmatrix} r & r & r \\ r & r & e \\ r & e & e \\ e & e & e \end{bmatrix}.$$

Although the structure of R_{14} may seem odd, it turns out that both the permuted triangular structure and the possibility of having small nonzeros below the cross diagonal significantly simplify the updating algorithm.

In further examples, we will follow the convention used for describing the structure of R_{14}. An r will always represent a potentially large element and an e will represent an element which is small relative to the tolerance. The algorithm will keep the elements which should be small from growing inappropriately.

Each F block of the decomposition is an upper triangular matrix with norm of the order of the tolerance. Each E block is an arbitrary matrix, also with norm of the order of the tolerance. The S block is an arbitrary matrix. With sufficiently small E and F blocks, the decomposition gives estimates of the numerical range spaces of A and B, along with an estimate of the numerical intersection in the form of the basis provided by the first r_3 columns of U.

The justification for the small nonzero blocks in (1.4) is that they allow us to find a nontrivial quotient URV structure associated with a slightly perturbed pair of matrices. The locations of these blocks are chosen to facilitate updating. The algorithm will be designed to perform deflations of small singular values using a tolerance that will keep these elements suitably small.

However, in some applications this might not be sufficient. If we wish to reliably identify the most rank deficient nearby quotient SVD structure corresponding to the smallest ranks for A, B, and $\begin{bmatrix} A & B \end{bmatrix}$, then it is natural to expect these perturbations, and hence the magnitudes of the E and F blocks, to be not much larger than the smallest perturbations to A and B required to give a quotient URV structure of the form (1.2). Inordinately large elements in this blocks might cause the tolerance to be reached too early in the deflation process, leading to an overestimate of the ranks of A and B and an underestimate of the intersection dimension.

Unfortunately, standard quotient QR and URV algorithms fail by this standard and the method of this paper suffers from a similar problem. The difficulty centers on the fact that R_{23} is a part of a URV decomposition,

$$\begin{bmatrix} R_{23} & E_{24} & E_{25} \\ 0 & F_{34} & E_{35} \\ 0 & 0 & F_{45} \end{bmatrix}, \tag{1.5}$$

that estimates a numerical rank for $P_A^{\perp} B$. The exact decomposition (1.2) reveals the

exact rank of $P_A^{\perp}B$. When the range space of A is sufficiently sensitive to perturbations, small perturbations of A can lead to large changes in the small singular values of $P_A^{\perp}B$ that correspond to the exactly zero singular values shown in the unperturbed (1.2). Thus, even when dealing with small perturbations to a matrix pair with the exact structure (1.2), the E and F blocks in (1.5) might be significantly larger than the original perturbations to the data.

We illustrate the problem with the matrix pair

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \delta \\ 0 & \epsilon \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{1.6}$$

where $0 < \epsilon < \delta < 1$. We suppose that δ is significantly smaller than 1 but that it is large enough that A can be considered to have full numerical rank. We assume that the perturbing quantity ϵ is small enough that it is of the same order as the tolerance used in rank decisions. A perturbation of norm ϵ to A clearly results in two full rank matrices with an exact one-dimensional row subspace intersection.

Consider the orthogonal transformation given by the QR factorization of A,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\delta}{\sqrt{\delta^2+\epsilon^2}} & \frac{\epsilon}{\sqrt{\delta^2+\epsilon^2}} & 0 \\ 0 & \frac{-\epsilon}{\sqrt{\delta^2+\epsilon^2}} & \frac{\delta}{\sqrt{\delta^2+\epsilon^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \delta & 1 & 0 \\ 0 & \epsilon & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \sqrt{\delta^2+\epsilon^2} & \frac{\delta}{\sqrt{\delta^2+\epsilon^2}} & 0 \\ 0 & 0 & \frac{-\epsilon}{\sqrt{\delta^2+\epsilon^2}} & 0 \\ 0 & 0 & 0 & 1 \end{array}\right].$$

This decomposition gives the SVD of $P_A^{\perp}B$. The smallest singular value is $\epsilon/\sqrt{\delta^2+\epsilon^2}$. If δ is sufficiently small, we would conclude that $P_A^{\perp}B$ has full rank. This would imply that $r_i = 0$, so that the algorithm completely misses the possibility that there is a nontrivial range space intersection achieved by matrices within $O(\epsilon)$ of A and B. The end result is a misleadingly partitioned quotient URV that fails to reveal an interesting and potentially useful feature of A and B.

Sensitivity in rank decisions is fundamental to any generalized URV or generalized QR algorithm that starts with an estimate of the range space of A and proceeds with a rank decision for $P_A^{\perp}B$, including the methods described in [9, 1]. The algorithms in [5, 6] are somewhat different in that they make rank decisions on a matrix with singular values equal to the generalized singular values of A and B. Since generalized singular values can be sensitive to perturbations [8], the rank decisions in these methods can also be difficult.

Although the updating problem considered here is more involved, the basic tools needed to update this decomposition have already been developed for the problem of updating a URV decomposition in [11]. The algorithm can be broken into two stages. The first restores the form of the decomposition when new rows are added to A and B. After this stage of the update, the decomposition has the same general form, but the triangular R matrices are potentially of different size and might no longer have full rank. The second stage looks for small singular values of the R blocks and recursively deflates these blocks, using the scheme described in [11], until they have full rank.

When it is obvious after representing the new rows in the bases provided by V_1 and V_2 that the new information does not increase the ranks of any of the full rank blocks, parts of the updating algorithm are not needed. This is essentially the same simplification as appears in [11]. To avoid giving the details of too many special cases,

r	r	r	r	r	r_2	r	r	r	r	r	r_1
r	r	r	r	e	e	r	r	r	r	e	e
0	r	r	r	e	e	r	r	r	e	e	e
0	0	r	r	e	e	r	r	e	e	e	e
0	0	0	r	e	e	r	r	e	e	e	e
0	0	0	0	e	e	r	r	e	e	e	e
0	0	0	0	e	e	0	r	e	e	e	e
0	0	0	0	e	e	0	0	e	e	e	e
0	0	0	0	e	e	0	0	0	e	e	e
0	0	0	0	e	e	0	0	0	0	e	e
0	0	0	0	e	e	0	0	0	0	0_1	e
0	0	0	0	e	e	0	0	0	0	0	0
0	0	0	0	0_2	e	0	0	0	0	0	0

FIG. 2.1. *An example.*

we deal only with the most general and the most difficult case here. This algorithm applies in every contingency, but if it is immediately obvious that the new data will not significantly change the estimates of the subspaces, then some of the steps are unnecessary. We will be more precise about which steps can be skipped in section 3.

2. The updating algorithm. We first describe how to restore the general structure after new rows are added to A and B, leaving the discussion of deflation for section 3. We also ignore initialization issues by assuming that at some stage the decomposition has already been computed and we are simply interested in computing the update. This does not evade the description of an essential step since the algorithm applies in degenerate cases when the sizes of some of the triangular matrices are zero (although this might involve the elimination of superfluous steps). Thus, the process can be initialized by setting the decomposition to zero, setting the unitary matrices to the identity, and starting the algorithm with the first rows of A and B. It could also start at some later point by applying a more conventional generalized SVD algorithm to compute the initial decomposition.

If two rows, a^T and b^T, are added to A and B, respectively, and each row of the old matrix is weighted by $0 < \alpha < 1$, then

$$\begin{bmatrix} 1 & 0 \\ 0 & U^T \end{bmatrix} \left[\begin{array}{c|c} a^T & b^T \\ \alpha A & \alpha B \end{array}\right] \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \left[\begin{array}{cc|ccc} a_1^T & a_2^T & b_3^T & b_4^T & b_5^T \\ R_{11} & E_{12} & S_{13} & R_{14} & E_{15} \\ 0 & E_{22} & R_{23} & E_{24} & E_{25} \\ 0 & E_{32} & 0 & F_{34} & E_{35} \\ 0 & E_{42} & 0 & 0 & F_{45} \\ 0 & F_{52} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{2.1}$$

where a_i^T and b_i^T are the obvious partitionings of $a^T V_1$ and $b^T V_2$. The blocks of the decomposition shown here are the same as those shown in (1.4) but weighted by α. The problem is to update the orthogonal matrices U and V to restore the structure of the decomposition and to deal with possible rank changes in the R matrices.

To illustrate how we can restore the appropriate structure, we take as an example Figure 2.1. This shows an extra row added to the top of a matrix that has the general form described in (1.4). We may efficiently restore the original structure through the

application of sequences of Givens rotations of the form

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $c^2 + s^2 = 1$, to the appropriate rows and columns of the matrix in Figure 2.1. A description of how to compute Givens rotations to introduce zeros in a numerically reliable manner may be found in [3].

When showing the updating of the decomposition, the approach taken here for dealing with Givens rotations is to mark the elements that are to be eliminated with a number indicating their order and to give any additional information in the text. Such information includes whether the rotation acts on the row or column containing the marked element, along with which other row or column the rotation also acts on. In this algorithm, the rotations will always act on a row or column that is adjacent to the numbered row or column in addition to the numbered row or column itself. The identity of the adjacent row or column is not always explicitly mentioned in the text, but the examples should make this detail of the procedure clear.

Many of the rotations will destroy the structure of a block of the decomposition so that it is sometimes necessary to apply additional rotations to fix this damage. The elements that must be eliminated to fix the structure will be marked with the same number as the rotation that originally did the damage. Occasionally there will be a sequence of two such fixes beyond the original rotation. The fix will always be applied on the opposite side of the rotation which originally did the damage. Thus damage caused by rotations applied on the left are fixed by rotations applied on the right and vice versa. All such rotations can be easily spotted, since they always correspond to either a marked zero element or one of the small elements of the R_{14} block—elements for which no rotation would be needed if they had not been made potentially large by another rotation.

In Figure 2.1 the numbered elements represent two sequences of right rotations to zero the marked elements. Rotation 1 acts on the column of the numbered element and the preceding column to eliminate the marked element, destroying the triangularity of the F_{45} block. It can be restored after the Givens rotation is applied from the right by a rotation from the left. Rotation 2 destroys the triangular structure of the F_{52} block. This can also be maintained through the appropriate use of a left rotation. In a more general setting, rotations 1 and 2 would each be replaced by multiple rotations that are intended to zero all but the first element of a_2^T and b_5^T, respectively, and, after each rotation, it would be necessary to apply an additional rotation to fix F_{52} and F_{45}

The result of these rotations is the matrix shown in Figure 2.2. Now that rotations have been applied to concentrate large elements from the new rows into a region in which they can damage at most two columns, we can take advantage of the permuted triangular structure of the overall decomposition and apply a sequence of rotations that are essentially the same as those used in QR updating. Each numbered rotation, except for those in R_{14}, acts on the numbered row and the preceding row to introduce the necessary zeros. The only additional complication is the need to preserve the triangular structure of R_{14}. Figure 2.2 marks the elements to be eliminated by left rotations. Each left rotation operates on the row marked and the previous row to eliminate the marked element. The first r_3 rotations, rotations 1 and 2 in this example, will destroy the structure of the R_{14} block. For rotation 1, we may fix the damage to R_{14} by using a right rotation on the numbered column and the one before it and then

r	r	r	r	r	0	r	r	r	r	r	0
r_1	r	r	r	e	e	r	r	r	r	e_1	e
0	r_2	r	r	e	e	r	r	r	e_2	e	e
0	0	r_3	r	e	e	r	r	e	e	e	e
0	0	0	r_4	e	e	r	r	e	e	e	e
0	0	0	0	e	e	r_5	r	e	e	e	e
0	0	0	0	e	e	0	r_6	e	e	e	e
0	0	0	0	e	e	0	0	e_7	e	e	e
0	0	0	0	e	e	0	0	0_2	e_8	e	e
0	0	0	0	e	e	0	0	0	0_1	e_9	e
0	0	0	0	e	e	0	0	0	0	0	e_{10}
0	0	0	0	e_{11}	e	0	0	0	0	0	0
0	0	0	0	0	e_{12}	0	0	0	0	0	0

FIG. 2.2. *A sequence of left rotations.*

r	r	r	r	r	e	r	r	r	r	r	e
0	r	r	r	r	e	r	r	r	r	e	e
0	0	r	r	r	e	r	r	r	e	e	e
0	0	0	r	r	e	r	r	r	e	e	e
0	0	0	0	r	e	r	r	r	e	e	e
0	0	0	0	r_6	e	0	r	r	e	e	e
0	0	0	0	r_5	e	0	0	r	e	e	e
0	0	0	0	r_4	e	0	0	0	e	e	e
0	0	0	0	r_3	e	0	0	0	0	e	e
0	0	0	0	r_2	e	0	0	0	0	0	e
0	0	0	0	r_1	e	0	0	0	0	0	0
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 2.3. *Left rotations to repartition* R_{11}.

fix the damage to F_{34} by using a left rotation. We allow rotations 3 and 4 to fill the first column of R_{14} with potentially large elements, eventually moving down to add a potentially large column to the beginning of E_{24}, which results in the pattern shown in Figure 2.3.

At this point, it is necessary to repartition the matrix to prepare for a deflation. The reason for the peculiar reversed triangular structure of R_{14} becomes clear. This figure shows a sequence of elements to be eliminated by left rotations. These rotations will add extra elements into the subdiagonals of R_{23}, F_{34}, and F_{45} to give a matrix that has essentially the same form as the original decomposition. This matrix suggests a natural repartitioning. We expand the size of the square blocks R_{11} and R_{23} by one. The rest of the matrix can be repartitioned along these lines, except that an extra column is added onto the right of R_{14}. The new partitioning is marked in Figure 2.3 and in Figure 2.4. In Figure 2.4 it is clear that the general form of the decomposition has been restored.

The matrix has now been repartitioned so that it has its original form, but it is possible that some of the R blocks will not have full rank. To finish the update we need a general procedure to take a matrix of the correct form, (1.4), and determine

r	r	r	r	r	e	r	r	r	r	r	e
0	r	r	r	r	e	r	r	r	r	e	e
0	0	r	r	r	e	r	r	r	e	e	e
0	0	0	r	r	e	r	r	r	e	e	e
0	0	0	0	r	e	r	r	r	e	e	e
0	0	0	0	0	e	r	r	r	e	e	e
0	0	0	0	0	e	0	r	r	e	e	e
0	0	0	0	0	e	0	0	r	e	e	e
0	0	0	0	0	e	0	0	0	e	e	e
0	0	0	0	0	e	0	0	0	0	e	e
0	0	0	0	0	e	0	0	0	0	0	e
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 2.4. *The repartitioned matrix with the correct form restored.*

an appropriate size for all triangular matrices. The procedure is the deflation method introduced in [11], applied to R_{11}, R_{23}, and R_{14}, along with additional rotation to fix any damage done to the structure by the deflations.

An implementation of this algorithm in MATLAB code is given in the appendix. This implementation deals with special cases ($r_i = 0$ for $i = 1, 2, 3$ and/or A or B has full rank) which were glossed over in the description of the algorithm. For the most part, these special cases involve omitting only the unnecessary steps. To avoid producing an overwhelming quantity of code, the deflation procedures described in the next section are hidden in function calls. The implementation of these deflations is fairly straightforward, given an understanding of the basic methods of the updating algorithm. We will not present codes for these functions. Finally, we note that the algorithm can be made more efficient by comparing components of the new rows with the tolerance and avoiding certain steps (and the deflations) when the new rows cannot change the rank of A or B.

3. The deflation process. The deflation process for each block proceeds by finding a small singular value associated with the block, using knowledge of an associated singular vector to apply transformations forcing the rightmost column to have elements only of the order of this singular value, and continuing the process recursively on a smaller triangular matrix. The overall process proceeds by recursively deflating R_{11} until an appropriate rank is found, then deflating R_{23} in a similar manner, and then finally deflating R_{14}. We will describe the deflations in this order. The basic idea behind the procedure in [11] is to find a vector $\|w\|_2 = 1$ such that

$$\|R_{11}w\| \approx \sigma_{\min}(R_{11}),$$

where $\sigma_{\min}(R_{11})$ denotes the smallest singular value of R_{11}. The literature on condition estimation contains reliable methods which find such a w with $O(r_1^2)$ complexity; the precise method is not particularly important for our explanation of the procedure.

If $\|R_{11}w\|$ is small enough to be considered a null vector within the tolerance, then R_{11} is nearly rank deficient and must be deflated. A sequence of rotations is constructed, zeroing the elements of w in order until the last component is reached. While at the same time applying left rotations, also to R_{11} and in the manner de-

scribed in [11] to preserve triangularity, we obtain

$$R_{11} = \begin{bmatrix} r & r & r & r & e \\ 0 & r & r & r & e \\ 0 & 0 & r & r & e \\ 0 & 0 & 0 & r & e \\ 0 & 0 & 0 & 0 & e \end{bmatrix}.$$

Then pattern of e elements holds since

$$\sigma_{\min}(R) \approx \|Rw\| = \|\hat{U}^T R \hat{V} \hat{V}^T w\| = \|\hat{U}^T R \hat{V} e_n\|,$$

while $\hat{U}^T R \hat{V} e_n$ is the last column of the new R which results from this deflation procedure. As described in relation to Figure 2.2, while it is convenient we fix the effects of the left rotation on R_{14}, but after a certain point, we let the first column of R_{14} fill with large elements. The result of this is shown in Figure 3.1. The matrix has to be repartitioned to make R_{11} smaller. This can be done by eliminating the elements marked in Figure 3.1 with left rotations which act on the numbered row and the row just before it. The result is Figure 3.2.

The effect of the deflating R_{11} on the sizes of the other R blocks is simple to see. Each time the size of R_{11} decreases, the size of R_{23} potentially increases and the size of R_{14} potentially decreases.

The deflation of R_{23} is performed next and the deflation of R_{14} last. The effects of the deflation on R_{23} are very simple to deal with: none of the rotations damage the overall structure of the decomposition, so all that is needed is the standard deflation procedure from [11], resulting in Figure 3.3. A sequence of left rotations needed to produce zeros in the last column of S_{13} so that it can become the first column of R_{14} is shown, together with right rotations needed to fix the effect of these on R_{11}. Thus, each deflation results in a decrease in the size of R_{23} and an increase in the size of R_{14}. If the ranks are to be restored to their original values, then it will be necessary to carry out two deflations on R_{23}. The result of these two deflations, with the corresponding repartitioning for R_{23} and R_{14}, is shown in Figure 3.4. The assumption that the ranks return to their original values is not essential to the algorithm, and it is adopted here only for convenience on the grounds that the algorithm will typically operate in steady state. The method of deflation is general and applies even without this assumption.

The deflation process for R_{14} is similar, although it is worth taking note of minor differences imposed by the odd structure of the block. First, the methods for finding w usually involve a back substitution. Here we ignore the small nonzero elements in attempting to find

$$\|R_{14} w\| \approx \sigma_{\min}(R_{14}).$$

Assuming that we have such a w, we apply rotations to introduce zeros into all but the last element and apply these rotations, together with rotations to fix the permuted triangular structure of R_{14}. Further rotations will be needed to fix R_{11} and F_{34}. The result of single deflation will be

$$R_{14} = \begin{bmatrix} r & r & e \\ r & r & e \\ r & e & e \\ e & e & e \end{bmatrix}.$$

r	r	r	r	e	e	r	r	r	r	r	e
0	r	r	r	e	e	r	r	r	r	e	e
0	0	r	r	e	e	r	r	r	r	e	e
0	0	0	r	e	e	r	r	r	r	e	e
0	0	0	0	e	e	r	r	r	r	e	e
0	0	0	0	0	e	r_1	r	r	r	e	e
0	0	0	0	0	e	0	r_2	r	r	e	e
0	0	0	0	0	e	0	0	r_3	r	e	e
0	0	0	0	0	e	0	0	0	r_4	e	e
0	0	0	0	0	e	0	0	0	0	e_5	e
0	0	0	0	0	e	0	0	0	0	0	e_6
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 3.1. *After a deflation of* R_{11}.

r	r	r	r	e	e	r	r	r	r	r	e
0	r	r	r	e	e	r	r	r	r	e	e
0	0	r	r	e	e	r	r	r	r	e	e
0	0	0	r	e	e	r	r	r	r	e	e
0	0	0	0	e	e	r	r	r	r	e	e
0	0	0	0	e	e	0	r	r	r	e	e
0	0	0	0	e	e	0	0	r	r	e	e
0	0	0	0	e	e	0	0	0	r	e	e
0	0	0	0	e	e	0	0	0	0	e	e
0	0	0	0	e	e	0	0	0	0	0	e
0	0	0	0	e	e	0	0	0	0	0	0
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 3.2. *After deflation of* R_{11} *and repartitioning.*

r	r	r	r	e	e	r	r	r	r	r	e
0	r	r	r	e	e	r	r	r	r	e	e
0	0_2	r	r	e	e	r	r	r	r_2	e	e
0	0	0_1	r	e	e	r	r	r	r_1	e	e
0	0	0	0	e	e	r	r	r	e	e	e
0	0	0	0	e	e	0	r	r	e	e	e
0	0	0	0	e	e	0	0	r	e	e	e
0	0	0	0	e	e	0	0	0	e	e	e
0	0	0	0	e	e	0	0	0	0	e	e
0	0	0	0	e	e	0	0	0	0	0	e
0	0	0	0	e	e	0	0	0	0	0	0
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 3.3. *After a deflation of* R_{23}.

r	r	r	r	e	e	r	r	r	r	r	e
0	r	r	r	e	e	r	r	r	r	e	e
0	0	r	r	e	e	r	r	r	e	e	e
0	0	0	r	e	e	r	r	e	e	e	e
0	0	0	0	e	e	r	r	e	e	e	e
0	0	0	0	e	e	0	r	e	e	e	e
0	0	0	0	e	e	0	0	e	e	e	e
0	0	0	0	e	e	0	0	0	e	e	e
0	0	0	0	e	e	0	0	0	0	e	e
0	0	0	0	e	e	0	0	0	0	0	e
0	0	0	0	e	e	0	0	0	0	0	0
0	0	0	0	0	e	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

FIG. 3.4. *After two deflations of* R_{23}.

A sequence of left rotations will easily transform this to

$$R_{14} = \begin{bmatrix} r & r & e \\ r & e & e \\ e & e & e \\ e & e & e \end{bmatrix},$$

and the damage that the left rotations do to R_{11} can easily be fixed by right rotations. This completes the deflation of all of the triangular blocks.

Since many of the basic principles were illustrated by appealing to an example, it is worth noting that the deflation process for each block is quite general and does not depend on the sizes of the blocks. The deflation might have to be done for each block several times, but because each deflation returns the matrix to its correct form, they can be performed recursively for each block until the proper ranks are determined. As explained above, the deflation process is first applied recursively to R_{11} until its rank has been determined, and then we do the same for R_{23} and finally R_{14}.

There is one difficulty that has not yet been mentioned. The code given in the appendix assumes that prior to the update $r_1 \geq r_3$. This is generally not a problem, but if the size of R_{23} drops sufficiently after several deflations, then r_3 will temporarily increase by a corresponding amount before R_{14} is deflated. After the first such deflation on R_{23} for which this is a problem, we will have R_{14} of the form

$$R_{14} = \begin{bmatrix} r & r & r & r \\ r & r & r & e \\ r & r & e & e \end{bmatrix}.$$

Right rotations should be applied to compress this into the first 3 columns. The effect on F_{34} may be fixed with left rotations. This will keep $r_3 = r_1$ while the appropriate size of R_{23} is being determined.

The algorithm in this paper is essentially an extension of the URV algorithm as described in [11] and it inherits the simplicity with which the URV decomposition can deal with updates that do not increase any of the ranks. If after applying V_1 and V_2 to the new row vectors it is apparent that none of the ranks increase, the generalized URV updating can proceed in a manner similar to the simplified URV algorithm by skipping the rotations shown in Figures 2.1 and 2.3 and by skipping the deflation of

R_{11} and R_{14}. Because of the way the rotations in Figure 2.2 change the partitioning of R_{23} and R_{14} it will still be necessary to do a deflation on R_{23}. This is all in contrast to the ULV algorithm in which substantial additional computations are needed to avoid a deflation.

In the sample code in the appendix, we have hidden the deflations in functions which are not presented in this paper. Since the deflations involve fewer special cases and parallel the method of [11] more closely than the update, we have left them out.

4. Complexity. The decomposition has a fairly involved structure and, from the description given here, it might be assumed that the algorithm is computationally intensive. In fact, considering the difficulty of the problem, this is not the case; the computational complexity is surprisingly reasonable. The exact numbers will depend on r_1, r_2, and r_3 in addition to the number of columns in A and B. To simplify matters for comparison, we assume ranks which are reasonable in the context of the identification algorithm of [7]. In particular, we assume that A and B each have $2i$ columns and that $r_1 = i + n$, $r_2 = i$, and $r_3 = n$. We assume that i is slightly larger than n. If left rotations are not accumulated to form U (knowledge of U is not required by the updating algorithm), then the complexity involved in updating this decomposition when none of the ranks change is at worst $325i^2 + 120in + 6n^2$ flops. If it is apparent in the first stage of the update that the new rows do not increase the ranks of the R blocks, then the update can be computed with much lower complexity.

These numbers look very bad, but when the level of difficulty inherent in the problem is taken into account, they are quite reasonable. If i is only slightly larger than n, so that the difference can be absorbed into lower-order terms, then the complexity is $451(4i)^2/16$. Thus, since the decomposition involves a matrix combining both A and B with a total of $4i$ columns, the worst-case complexity involved is really expressed more reasonable as approximately $28(4i)^2$ flops.

This is certainly large when compared with the updating of a QR decomposition of a matrix of the same size. The QR decomposition involves only $3(4i)^2$ flops, but an ordinary URV decomposition is a different matter. Just to compute a URV decomposition of A involves $71i^2 + 6in + 3n^2$ flops. Again, assuming i and n are close and ignoring lower-order terms, the complexity is roughly $5(4i)^2$. Thus updating the quotient URV is about a factor of three more costly than computing URV decompositions for A and B separately. Depending on the ranks involved, it is often not much more costly than updating the URV decomposition of a single matrix with $4i$ columns.

The generalized URV decomposition is similar in spirit to the generalized QR factorization of [9]. However, a generalized QR factorization does not lend itself to updating. In terms of computational complexity, the use of the URV updating method is justifiable only when updates are needed. The method is not competitive for finding the subspaces associated with the generalized SVD of a single matrix.

5. An application to system identification. Consider the state space model

$$x_{k+1} = A_k x_k + B_k u_k, \tag{5.1}$$

$$y_k = C_k x_k + D_k u_k,$$

where x_k is $n \times 1$, u_k is $m \times 1$, and y_k is $p \times 1$.

Assuming we have observations of the input and output vectors, u_k and y_k, the identification problem is to find an order, n, and time-varying system matrices $\{A_k, B_k, C_k, D_k\}$ that satisfy, or approximately satisfy, (5.1) for some $n \times 1$ state sequence x_k.

If the output vectors are generated by a time-invariant model $\{A, B, C, D\}$ and observations are corrupted by noise, we want the estimated model to converge to $\{A, B, C, D\}$ or to some model $\{SAS^{-1}, SB, CS^{-1}, D\}$ given by a change of basis for x_k and having identical input/output behavior.

More realistically, it is often assumed that the state space model is slowly time-varying and that there is small noise on the observed input and output vectors. Under those circumstances, we wish to provide an algorithm to track variations in the model.

The generalized URV decomposition applies naturally to a system identification algorithm developed in [7]. The approach can be characterized by two steps: find an estimate of the state sequence x_k, and then obtain the system matrices from the least squares problem

$$\begin{aligned}\begin{bmatrix} x_{k+i+j-1} & \cdots & x_{k+i+1} \\ y_{k+i+j-2} & \cdots & y_{k+i} \end{bmatrix} W_{j-1} \\ = \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix} \begin{bmatrix} x_{k+i+j-2} & \cdots & x_{k+i} \\ u_{k+i+j-2} & \cdots & u_{k+i} \end{bmatrix} W_{j-1},\end{aligned} \tag{5.2}$$

where W_j is a diagonal weighting matrix defined by

$$W_j = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_j W_{j-1} \end{bmatrix}$$

for $|\alpha_j| < 1$ and $W_1 = 1$. The index k is the time at which observations begin and $k + i + j - 1$ is the time at which the latest observations have been made. Indices k and i are fixed, but j grows as more observations are made. To keep the notation compact, the indexing of the system matrices will show only the dependence on j, although $\{A_j, B_j, C_j, D_j\}$ will depend on observation up to $u_{k+i+j-1}$ and $y_{k+i+j-1}$.

Define the $mi \times j$ block Toeplitz matrices

$$U_K = \begin{bmatrix} u_{k+j-1} & u_{k+j-2} & \cdots & u_k \\ u_{k+j} & u_{k+j-1} & \cdots & u_{k+1} \\ \vdots & \vdots & & \vdots \\ u_{k+j+i-2} & u_{k+j+i-3} & \cdots & u_{k+i-1} \end{bmatrix},$$

$$Y_k = \begin{bmatrix} y_{k+j-1} & y_{k+j-2} & \cdots & y_k \\ y_{k+j} & y_{k+j-1} & \cdots & y_{k+1} \\ \vdots & \vdots & & \vdots \\ y_{k+j+i-2} & y_{k+j+i-3} & \cdots & y_{k+i-1} \end{bmatrix},$$

and

$$T_k = \begin{bmatrix} U_k \\ Y_k \end{bmatrix}.$$

The following theorem from [7] provides a means for generating an appropriate sequence of state vectors.

THEOREM 5.1. *Let the vectors u_k and y_k be generated by*

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k + Du_k, \end{aligned}$$

where the rank of

$$\begin{bmatrix} C^T & A^T C^T & \cdots & (A^T)^{n-1} C^T \end{bmatrix} \tag{5.3}$$

is n.

Let

$$X_k = \begin{bmatrix} x_{k+j-1} & x_{k+j-2} & \cdots & x_k \end{bmatrix}$$

and

$$X_{k+i} = \begin{bmatrix} x_{k+i+j-1} & x_{k+i+j-2} & \cdots & x_{k+i} \end{bmatrix}.$$

For $i \geq n$, if rank(X_k)= rank$(X_{k+i}) = n$ and the matrices

$$\begin{bmatrix} U_k \\ X_k \end{bmatrix}, \qquad \begin{bmatrix} U_{k+i} \\ X_{k+i} \end{bmatrix}, \qquad \begin{bmatrix} U_k \\ U_{k+i} \\ X_k \end{bmatrix} \tag{5.4}$$

all have full rank $mi + n$, $mi + n$, and $2mi + n$, respectively, then T_k and T_{k+i} both have rank $mi + n$ and the intersection of the span of the rows of T_k and T_{k+i} has dimension n. Further, there is a basis, X, of the intersection for which

$$X = \begin{bmatrix} x_{k+i+j-1} & x_{k+i+j-2} & \cdots & x_{k+i} \end{bmatrix},$$

and different bases for this space correspond to state vector sequences of models with equivalent input/output behavior under a transformation of the form

$$\{SAS^{-1}, SB, CS^{-1}, D\}.$$

The rank condition on (5.3) implies the observability of the linear system; without this assumption the full information contained in the state sequence will not be seen in the output and any identification scheme can be expected to fail. The condition on the rank of X_k and X_{k+i} implies that the input fully excites all modes of the system. This is also a standard and necessary assumption in system identification.

The rank assumption involving (5.4) is stronger: it clearly implies the rank condition on X_k and X_{k+i}. The key point is to make sure that U_k and U_{k+i} have full rank and to make sure that X_k is not contained in their span. A full rank condition on the inputs is standard in identification. The joint condition on X_k is less standard, but it can be verified that it is satisfied generically and the probability that it fails decreases when j is increased. More details about the implications of these assumptions together with a proof of the theorem may be found in [7].

If $\alpha \neq 1$, we look for the intersection of the span of the rows of $T_k W_j$ and $T_{k+i} W_j$. In that case, the theorem shows that the intersection is the weighted state vector sequence required by (5.2).

The generalized URV algorithm can be used to update the intersection of the range spaces of T_k^T and T_{k+i}^T as new observations are made and new rows are added to the matrices. This leaves the solution of (5.2) to complete the identification process. It is possible to efficiently update the QR decomposition needed to solve (5.2) while updating the generalized URV decomposition. Further details are contained in [12].

In order to show the effectiveness of the decomposition, we consider the system defined by

$$A = \begin{bmatrix} .4 & 0 & .8 \\ .4 & .4 & -.4 \\ .4 & 0 & .4 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ -4 & 2 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & -1 & 0 \\ 1 & -2 & -1 \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The system can be verified to be stable with its largest eigenvalue having magnitude strictly less than 1. The observability condition is also easily verified.

We generated a sequence of input vectors u_k with elements that were randomly generated according to zero mean normal distribution with variance 1. An initial state vector was chosen as $x_1 = 0$. A sequence of output vectors, y_k, was generated by (5.1).

Before applying the identification algorithm, each component of the input and output vectors was perturbed by zero mean unit variance normal noise scaled by .01, resulting in a noise component two orders of magnitude below the signal component. The tolerance for deflation of all three triangular blocks was set to an absolute value of .5. To give an idea of the relative significance of this tolerance, the data matrices satisfy

$$\left\| \begin{bmatrix} T_1^T & T_{i+1}^T \end{bmatrix} \right\| \approx 100$$

for $j = 50$. The value of this norm for $j = 20$ is approximately 50. We used $i = 3$ and data were taken for $j = 24, \ldots, 50$.

For each j the generalized URV decomposition correctly identified the order $n = 3$. When the identified model was driven by the inputs u_k starting with the earliest identified state vector x_{i+1}, the difference between the outputs produced by the identified model and the original model was of the same order of magnitude as the noise. We define

$$Y = \begin{bmatrix} y_{i+1} & y_{i+2} & \cdots & y_{100} \end{bmatrix}$$

as a matrix of original, unperturbed outputs and

$$\hat{Y}_j = \begin{bmatrix} \hat{y}_{i+1} & \hat{y}_{i+2} & \cdots & \hat{y}_{100} \end{bmatrix}$$

as the matrix of simulated outputs produced by the system identified using j columns of T_1 and T_{i+1}. The errors

$$\frac{\|Y - \hat{Y}_j\|_2}{\|Y\|_2}$$

are shown in Figure 5.1.

Clearly the algorithm is successful in handling this level of noise. However, if we increase the noise level by a factor of 10, the algorithm fails dramatically; it is not possible to find a tolerance for which the ranks r_2 and r_3 are estimated reliably. Nevertheless the rank r_1 and the sum $r_2 + r_3 = r_1$ are both estimated reliably for a choice of absolute tolerance of 1.

The problem is the inherent sensitivity of the generalized SVD computation as characterized by a simple perturbation analysis in section 1. For rank estimation, the

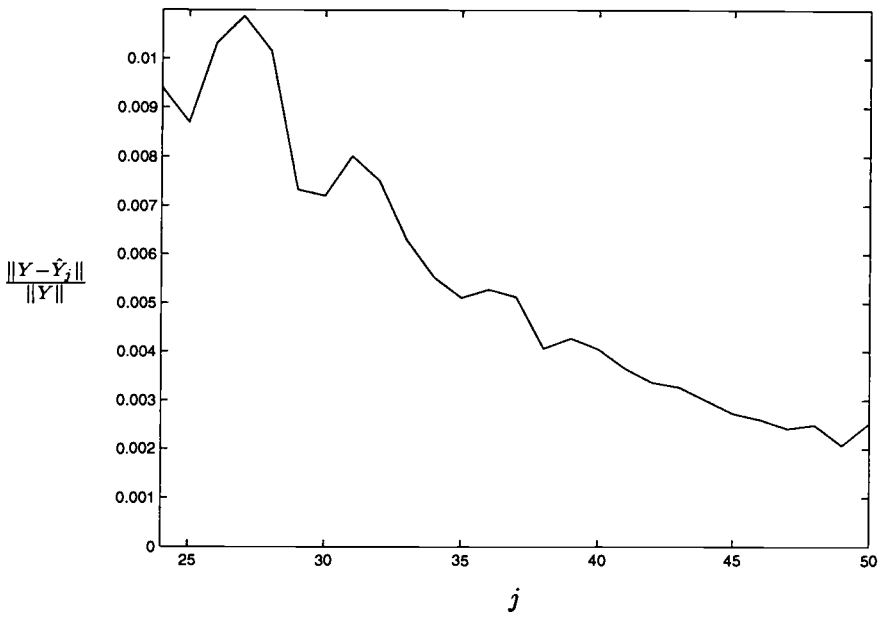


FIG. 5.1. *The identification output residual for varying j.*

relevant singular values of T_1 for $j = 50$ are

$$\sigma_9(T_1) = 5.3, \qquad \sigma_{10}(T_1) = .78.$$

This drop in the singular values is likely to be spotted as significant. By theorem 5.1, we know that $\text{rank}(T_1) = mi + n = 6 + n$ so that we can deduce that $n = 3$ simply from an accurate estimate of r_1. Similarly the rank of T_{i+1} is easily determined to be $6 + n$ and, given a correct estimate of r_1, we know that r_2 should be 6 and r_3 should be 3.

Unfortunately, determining the correct values of r_2 and r_3 is often not as easy as determining their sum. The reason is that determining r_2 is a secondary rank decision that depends on a prior estimate of the range space of T_1^T. Although the singular values of T_1 and T_2 are perfectly conditioned with respect to noise perturbations, the estimate of the range space of T_1^T can be sensitive if R_{11} is ill-conditioned. An accurate estimate of this range space is required to determine r_2.

The problem can occur even when using more refined algorithms. Given the SVD of T_1^T

$$T_1^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix},$$

where U_2 is $j \times (j - r_1)$, the problem of estimating r_2 is equivalent to determining the rank of $U_2^T T_{i+1}^T$. We know that this rank should be 6. But it turns out that

$$\sigma_6(U_2^T T_{i+1}) = 5.0, \qquad \sigma_7(U_2^T T_{i+1}) = 4.2.$$

This is a virtually impossible rank decision. Decreasing the noise variance for the same inputs verifies that the predictions of Theorem 5.1 are correct and that the artificially high value of $\sigma_7(U_2^T T_{i+1}^T)$ is due solely to noise.

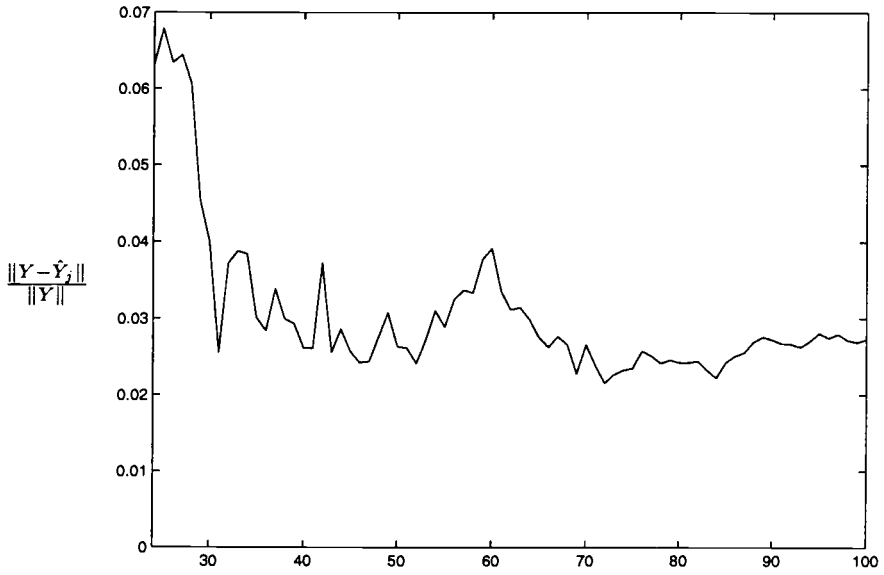


FIG. 5.2. *Output Residuals: High noise with n determined from* r_1.

Because generalized singular values can be sensitive to perturbations [8] and their computation depends on a nonorthogonal transformation, a method that looks directly at generalized singular values can also involve difficult decisions. Thus the failure of the algorithm is not really due to a difficulty in estimating rank using the imperfect URV methods. The problem is inherent in the rank decisions being made. This difficulty is one reason that the most reliable quotient SVD algorithms avoid making rank decisions whenever there is an alternative. Even with the most stable of algorithms, deciding which generalized singular value pairs belong to S_A and S_B or to I_A and I_B can be a difficult problem.

Fortunately, Theorem 5.1 provides a means for obtaining a suboptimal solution with the correct order even in the higher noise case. We use noise that is .1 times zero mean unit variance normal noise and set the absolute tolerance to 1 for determining r_1. Instead of estimating r_2 and r_3 using the standard deflation procedure, we determine r_2 and r_3 from the relations deduced from Theorem 5.1

$$r_3 = r_1 - mi, \qquad r_2 = mi.$$

We apply deflations to R_{23} and R_{14} to enforce these relations even when it results in slightly too large values in E or F blocks. The result is an algorithm that recovers the correct n but in which the intersection can be corrupted by the relaxation of the tolerance in the deflation of R_{23} and R_{14}. In practice, accepting a slightly corrupted intersection is not as harmful as estimating the order incorrectly. As noted, any algorithm for computing intersections applied to this problem is likely to be faced with this sort of choice. The resulting output errors are shown in Figure 5.2.

6. Conclusions. In this paper, we have presented an updating algorithm for a quotient type generalization of the URV decomposition. Three ranks must be estimated, and consequently the details of the algorithm are quite involved. However,

depending on the rank of the matrices, the overall computational complexity is often not much worse than computing a URV decomposition of equal size.

With respect to reliability, the algorithm performs very well in an application to system identification in which inputs and outputs are perturbed by moderate noise. The sequence of three interdependent rank decisions can be difficult in a high noise setting, but experiments with SVD based methods and the sensitivity results in [8] suggest that the difficulty is inherent in estimating ranks associated with generalized SVD and is not a result of a flaw in the algorithm.

Appendix. Sample code. In this appendix, we present a sample code for updating a quotient URV decomposition. To allow better illustration of the general outline of the algorithm, the code for deflation has been hidden in three function calls. This code deals with degenerate cases in which any of the three relevant ranks becomes zero or one. Handling these cases adds significantly to the complexity of the code; for reasons of brevity they were not described in the text.

```
function [R,U,V1,V2,r1,r2,r3]=gurv(R,U,V1,V2,r1,r2,r3,x1,x2,tol)
%
% For a 2n by 2n matrix R, m by 2n U and n by n V1 and V2 for which
%
% [A B]=U*R*[V1' 0; 0 V2']
%
% with A and B m by n, this code updates the decomposition by adding
% rows x1 and x2 to obtain the decomposition for [x1 x2;A B].
% Within the tolerance specified by tol:
% r1 is the rank of A.
% r3 is the dimension of the intersection.
% r2+r3 is the rank of B.
n2=max(size(R)); n=n2/2; rb=r2+r3; [m1,m2]=size(U);
if (max(m1,m2)==0)
  U=[1 zeros(1,2*n)];
else
  U=[1 zeros(1,m2);zeros(m1,1) U];
end
x=[x1*V1,x2*V2]; R=[x;R];
if (r1<n-1)
  for i=n:-1:r1+2
    G=givens(R(1,i-1),R(1,i));
    R(:,i-1:i)=R(:,i-1:i)*G';
    V1(:,i-1:i)=V1(:,i-1:i)*G';
    G=givens(R(n+i,i-1),R(n+i+1,i-1));
    R(n+i:n+i+1,i-1:n)=G*R(n+i:n+i+1,i-1:n);
    U(:,n+i:n+i+1)=U(:,n+i:n+i+1)*G';
  end
end
% If B was not full rank then adding the new row gives a triangular
% structure to R14 after the appropriate zeros have been introduced
% into the row.
if (rb<n-1)
  for i=n:-1:rb+2
    G=givens(R(1,n+i-1),R(1,n+i));
```

```
    R(:,n+i-1:n+i)=R(:,n+i-1:n+i)*G';
    V2(:,i-1:i)=V2(:,i-1:i)*G';
    G=givens(R(r1+i,n+i-1),R(r1+i+1,n+i-1));
    R(r1+i:r1+i+1,n+i-1:2*n)=G*R(r1+i:r1+i+1,n+i-1:2*n);
    U(:,r1+i:r1+i+1)=U(:,r1+i:r1+i+1)*G';
    if (r1<n)
      R(r1+i:r1+i+1,r1+1:n)=G*R(r1+i:r1+i+1,r1+1:n);
    end
  end
end
% If B is full rank then we make R14 triangular using left rotations.
if ((rb==)&(r3>0))
  for i=1:r3
    G=givens(R(i,2*n-i+1),R(i+1,2*n-i+1));
    R(i:i+1,:)=G*R(i:i+1,:);
    U(:,i:i+1)=U(:,i:i+1)*G';
    if (i>1)
      G=givens(R(i+1,i),R(i+1,i-1));
      R(1:i+1,i-1:i)=R(1:i+1,i-1:i)*G;
      V1(:,i-1:i)=V1(:,i-1:i)*G;
    end
  end
end
if (rb<n)
  r3=r3+1;
  rb=rb+1;
end
% We have to handle the case of full-rank B, r2+r3=n, separately.
% The case r2+r3=n-1 doesn't require any work at this stage at all.
% This restores the ``triangular'' structure of R11.
if (r1>0)
  if(r3>1)
   for i=1:r3-1
     G=givens(R(i,i),R(i+1,i));
     R(i:i+1,:)=G*R(i:i+1:);
     U(:,i:i+1)=U(:,i:i+1)*G';
     G=givens(R(i+1,n+rb-i),R(i+1,n+rb+1-i));
     R(:,n+rb-i:n+rb+1-i)=R(:,n+rb-i:n+rb+1-i)*G';
     V2(:,rb-i:rb+1-i)=V2(:,rb-i:rb+1-i)*G';
     G=givens(R(r1+rb-i+1,n+rb-i),R(r1+rb-i+2,n+rb-i));
     R(r1+rb-i+1:r1+rb-i+2,n+rb-i:2*n)=G*R(r1+rb-i+1:r1+rb-i+2,
                                           n+rb-i:2*n);
     U(:,r1+rb-i+1:r1+rb-i+2)=U(:,r1+rb-i+1:r1+rb-i+2)*G';
     if (r1<n)
       R(r1+rb-i+1:r1+rb-i+2,r1+1:n)=G*R(r1+rb-i+1:r1+rb-i+2,r1+1:n);
     end
   end
% Continue to restore the triangular structure, but now don't worry
% about fixing R14.
```

```
    for i=r3:r1
      G=givens(R(i,i),R(i+1,i));
      R(i:i+1,:)=G*R(i:i+1,:);
      U(:,i:i+1)=U(:,i:i+1)*G';
    end
  else
% The case of r3<2; we don't need to fix anything.
    for i=r:r1
      G=givens(R(i,i),R(i+1,i));
      R(i:i+1,:)=G*R(i:i+1,:);
      U(:,i:i+1)=U(:,i:i+1)*G';
    end
  end
end
% Now restore the triangular structure for R23, F34, F45.
for i=r1+1:r1+n
  G=givens(R(i,n+i-r1),R(i+1,n+i-r1));
  R(i:i+1,n+i-r1:2*n)=G*R(i:i+1,n+i-r1:2*n);
  U(:,i:i+1)=U(:,i:i+1)*G';
  if (r1<n)
    R(i:i+1,r1+1:n)=G*R(i:i+1,r1+1:n);
  end
end
% We restore the triangular structure of F52.
for i=r1+n+1:2*n
  G=givens(R(i,i-n),R(i+1,i-n)):
  R(i:i+1,i-n:n)=G*R(i:i+1,i-n:n);
  U(:,i:i+1)=U(:,i:i+1)*G';
end
if (r2<n)
  r2=r2+1;
  if (r3>0)
     r3=r3-1;
  end
end
% And we repartition R11.
if (r1<n)
  for i=r1+n:-1:r1+1
    G=givens(R(i,r1+1),R(i+r,r1+1));
    R(i:i+1,r1+1:2*n)=G*R(i:i+1,r1+1:2*n);
    U(:,i:i+1)=U(:,i:i+1)*G';
  end
  r1=r1+1;
end
U=U(:,1:2*n);
R=R(1:2*n,:);
[R,U,V1,V2,r1,r2,r3]=deflateR11(R,U,V1,V2,r1,r2,r3,n,tol);
[R,U,V1,V2,r1,r2,r3]=deflateR23(R,U,V1,V2,r1,r2,r3,n,tol);
[R,U,V1,V2,r1,r2,r3]=deflateR14(R,U,V1,V2,r1,r2,r3,n,tol);
```

Acknowledgments. This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

REFERENCES

[1] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[2] L. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–502.
[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.
[4] P. C. Hansen, *Rank-deficient prewhitening with quotient SVD and ULV decomposition*, BIT, 38 (1998), pp. 34–43.
[5] F. T. Luk and S. Qiao, *A new matrix decomposition for signal processing*, Automatica J. IFAC, 30 (1994), pp. 39–43.
[6] F. T. Luk and S. Qiao, *An adaptive algorithm for interference cancelling in array processing*, in SPIE Advanced Signal Processing Algorithms, Architectures and Implementations VI, 1996, pp. 151–161.
[7] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle, *On- and off-line identification of linear state-space models*, Internat. J. Control, 49 (1989), pp. 993–1014.
[8] C. C. Paige, *A note on a result of Sun Ji-guang: Sensitivity of the CS and GSV decomposition*, SIAM J. Numer. Anal., 21 (1984), pp. 186–191.
[9] C. C. Paige, *Some aspects of generalized QR factorizations*, in Reliable Numerical Computation, M. G. Cox and S. J. Hamarling, eds., Clarendon Press, Oxford, 1990, pp. 71–91. Cited in Å. Björck, available via anonymous ftp from math.liu.se in pub/references.
[10] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
[11] G. W. Stewart, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Process., 40 (1992), pp. 1535–1541.
[12] M. Stewart and P. Van Dooren, *An updating algorithm for on-line MIMO system identification*, in SVD in Signal Processing III, Algorithms and Applications, M. Moonen and B. De Moore, eds., Elsevier, New York, 1995.
[13] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 501–526

ON THE CHOLESKY FACTORIZATION OF THE GRAM MATRIX OF MULTIVARIATE FUNCTIONS*

TIM N. T. GOODMAN†, CHARLES A. MICCHELLI‡, GIUSEPPE RODRIGUEZ§, AND SEBASTIANO SEATZU¶

Abstract. We study the Cholesky factorization of certain bi-infinite matrices and related finite matrices. These results are applied to show that if the uniform translates of a suitably decaying multivariate function are orthonormalized by the Gram–Schmidt process over certain increasing finite sets, then the resulting functions converge to translates of a fixed function which is obtained by a global orthonormalization procedure. This convergence is also illustrated numerically.

Key words. Cholesky factorization, Gram–Schmidt process, orthonormal, multivariate

AMS subject classifications. 15A23, 41A63, 42C05

PII. S0895479899343274

1. Introduction.

We begin with a function $\phi : \mathbb{R}^d \to \mathbb{R}$ and form the translates of this function by vectors in the lattice $\mathbb{Z}^d$. We order these translates in some fashion and use the standard Gram–Schmidt process to orthonormalize them relative to Lebesgue measure on some bounded domain of $\mathbb{R}^d$. These orthonormalized functions are *not* generally the translates of a fixed function relative to our chosen ordering of $\mathbb{Z}^d$. That is, the Gram–Schmidt process will not preserve this property of the original functions to which it is applied. However, we believe under mild conditions on the function ϕ, the ordering of the translates and the growth of the domain relative to the finite number of translates orthonormalized, there will emerge a *limiting profile*. By our definition a limiting profile is a function such that its translates formed from the chosen ordering of $\mathbb{Z}^d$ will *asymptotically* represent the result of the Gram–Schmidt process. This could be observed computationally [16] in the special case when ϕ was a low degree univariate B-spline and positive integer shifts are orthonormalized relative to an interval $[0, n]$, where n grows large. The motivation for this investigation was to use orthogonal splines in various applications; see also [14]. Subsequently, we proved first in [11] for compactly supported function, including B-splines of arbitrary degree, and then later for functions which decay exponentially [13], for example the Gaussian, again for positive integer translates on $[0, n]$ that indeed a limiting profile emerges from the Gram–Schmidt process. The intent of this paper is to do the same for multivariate functions which can have less than exponential decay. The multivariate case leads to a richer theory.

*Received by the editors March 3, 1999; accepted for publication (in revised form) by U. Helmke April 12, 2000; published electronically August 9, 2000. The first and second authors were partially supported by NATO grant CRG 950849. The third and fourth authors were partially supported by the Italian Ministry of University and Scientific and Technological Research and a University of Cagliari coordinated research project.

http://www.siam.org/journals/simax/22-2/34327.html

†Department of Mathematical Sciences, University of Dundee, Dundee DD1 4HN, Scotland, UK (tgoodman@mcs.dundee.ac.uk).

‡IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (cam@watson.ibm.com).

§Dipartimento di Matematica, Università di Roma *Tor Vergata*, Via della Ricerca Scientifica, 00133 Roma, Italy (rodriguez@mat.uniroma2.it).

¶Dipartimento di Matematica, Università di Cagliari, Viale Merello 92, 09123 Cagliari, Italy (seatzu@unica.it).

In this paper we extend our previous results about the existence of a limiting profile in two ways. First we provide multivariate examples relative to certain orderings of the lattice $\mathbb{Z}^d$, and second, we allow for functions that have algebraic decay. Now let us review the steps we take in our analysis and connect them to existing theory in the matrix theory literature.

The first step of the analysis performed in [11] was to identify the form of the limiting profile as a linear combination of negative integer translates of ϕ whose coefficients form a lower triangular Toeplitz matrix that provides a Cholesky factorization of the inverse of the bi-infinite Gram matrix of all integer translates of the function ϕ. However, for bi-infinite matrices there are *many* Cholesky factorizations. The realization that the one we wanted is the (finite) *minimal phase* factorization, as it is known in the engineering literature (see, for example, [3, 21]), of the symbol of the banded Toeplitz matrix (the trigonometric polynomial formed from the diagonal elements of the Toeplitz matrix) led us to resolve the form of the limiting profile. With this information in hand, the complete analysis requires estimates for the decay of the elements of the inverse of the Toeplitz Gram matrix formed from all translates of ϕ, the assumed decay of the function ϕ, and the difference between the inverse of finite sections of the bi-infinite minimal phase Cholesky factor and the Cholesky factors of the inverse of the finite Gram matrix that appears in the Gram–Schmidt process. For the latter estimate, we required perturbation results for Cholesky factorization which are *independent* of the order of the matrix [23]. The decay of the inverse of a banded Toeplitz matrix is easily seen. This fact has been the subject of some interest in the literature in the general case of banded matrices. Using an idea from [10], this was done in [4] for a banded matrix coming from a problem in spline interpolation. Later this was extended to any banded matrix in [8] and then improved further in [9]. A study of Cholesky factorization of an arbitrary positive definite symmetric bi-infinite matrix was done in [6]. In that paper a uniquely distinguished Cholesky factor was characterized which in the case of a banded block Toeplitz bi-infinite matrix reduces to the finite minimal phase factorization.

In retrospect, the link between the limiting profile and finite minimal phase factorization rests with work by Bauer [1, 2] where he demonstrated that the Cholesky factors of the principal sections of semi-infinite Toeplitz matrix will converge to a banded lower triangular matrix whose elements yield the finite minimal phase factor for the bi-infinite matrix, thereby providing a numerical algorithm for the construction of the finite minimal phase factorization. Although the problem studied here has no connection to wavelet analysis, the appearance of finite minimal phase factorization in wavelet analysis (see, for example, [19]) provides a weak link between them. Motivated by this connection and also by some work of Schoenberg on orthonormalizing cardinal splines [22] we tested several of the most well-known algorithms to find this factorization on these two cases [12]. One of them that differs from the Bauer procedure which is important to us here is called the *cepstral* method; see [3, 21]. This algorithm is based on the work done independently in [5] and [15]; see also [12] and [24] for further information on this issue. We then extended the results of [11] to include functions that decay exponentially in [13]. Although it is not directly related to the concern in this paper we point out that Bauer's work was extended in [26] to the block Toeplitz case; see also [25] for further developments concerning block Toeplitz matrices. These ideas are useful in the study of the asymptotic behavior of the Gram–Schmidt process applied to a finite set of univariate functions and their integer translates; see [18].

Unfortunately, a finite minimal phase factorization does *not* exist in the multivariate case. In fact, there are bivariate nonnegative trigonometric polynomials which cannot be written as the modulus squared of an algebraic polynomial restricted to the torus; see, for example, [17]. Therefore, this excludes an identification with the limiting profile in the multivariate case when ϕ is compactly supported. Fortunately, though, it is the basis of the cepstral algorithm that resolves our problem. Specifically, we decompose the logarithm of the symbol as an appropriate sum of two terms relative to our chosen partial order of $\mathbb{Z}^d$. One of the summands is discarded and the other is exponentiated to form the desired factor that yields the limiting profile. When ϕ has some assumed decay at infinity we then estimate the decay of the coefficients of this factorization by following a technique in [20]. This is all done in detail in section 2 of the paper. In this section we also study the connection between the bi-infinite Gram matrix and the finite Gram matrices for certain subsets of functions restricted to certain finite domains. Section 3 then considers in greater generality the connection between Cholesky factors of certain bi-infinite and related finite matrices. Specializing these results to the situation of section 2, section 4 gives the required results on convergence of orthonormal functions on increasing finite sets to translates of a globally orthonormalized function.

The convergence results are *illustrated* numerically in section 5 for the case of a linear bivariate box spline. The numerical procedure for the factorization of the symbol follows the method used in section 2. The Gram–Schmidt orthonormalization depends on the ordering of the translates of the function, and numerical results are given for both the ordering considered in the previous sections and for another ordering. For both cases we observe numerically an exponential rate of convergence to the limiting profile.

2. Spectral factorization and limiting profile. Before studying the limiting profile corresponding to the Gram–Schmidt orthonormalization process of uniform translates of a multivariate function over increasing intervals, we need to state some properties of the infinite Gram matrix associated with the mentioned functions.

Hence, taking an integer $d \geq 1$, suppose that $\phi : \mathbb{R}^d \to \mathbb{R}$ is a measurable function satisfying

$$|\phi(x)| \leq \beta(|x|), \quad x \in \mathbb{R}^d, \tag{2.1}$$

where

$$|x| := |x_1| + |x_2| + \cdots + |x_d|$$

and β is a decreasing, positive function in $[0, \infty)$ satisfying

$$\int_0^\infty t^{d-1}\beta(t)dt < \infty. \tag{2.2}$$

We note that (2.1) and (2.2) imply that ϕ is bounded and integrable on $\mathbb{R}^d$.

For $j \in \mathbb{Z}^d$ we write $\phi_j := \phi(\cdot - j)$. We shall describe a procedure for orthonormalizing $\{\phi_j\}_{j\in\mathbb{Z}^d}$. For $j, k \in \mathbb{Z}^d$, let

$$T_{jk} = \int_{\mathbb{R}^d} \phi_j(x)\phi_k(x)dx = t_{j-k}, \tag{2.3}$$

where

$$t_j = t_{-j} = \int_{\mathbb{R}^d} \phi(x)\phi_j(x)dx.$$

We let T denote the Gram matrix

$$T = (T_{jk})_{j,k\in\mathbb{Z}^d}.$$

LEMMA 2.1. *There are constants $A > 0$, $a > 0$, such that $|t_j| \le A\beta(a|j|)$ for all j in $\mathbb{Z}^d$.*

Proof. Take j in $\mathbb{Z}^d$ and assume, for some ℓ, that $j_\ell \ge |j_k|$, $k = 1, \dots, d$. Then

$$\begin{aligned}\int_{x_\ell \ge \frac{1}{2}j_\ell} |\phi(x)||\phi(x-j)|dx &\le \int_{x_\ell \ge \frac{1}{2}j_\ell} \beta\left(\frac{1}{2}j_\ell\right)|\phi(x-j)|dx \\ &\le \beta\left(\frac{1}{2}j_\ell\right)\int_{\mathbb{R}^d}|\phi(x-j)|dx \le \beta\left(\frac{|j|}{2d}\right)\int_{\mathbb{R}^d}|\phi(x)|dx.\end{aligned}$$

Also, we observe that

$$\begin{aligned}\int_{x_\ell \le \frac{1}{2}j_\ell} |\phi(x)||\phi(x-j)|dx &= \int_{x_\ell \ge \frac{1}{2}j_\ell} |\phi(j-x)||\phi(-x)|dx \\ &\le \int_{x_\ell \ge \frac{1}{2}j_\ell} |\phi(j-x)|\beta\left(\frac{1}{2}j_\ell\right)dx \le \beta\left(\frac{|j|}{2d}\right)\int_{\mathbb{R}^d}|\phi(x)|dx,\end{aligned}$$

and therefore we conclude that

$$|t_j| \le \int_{\mathbb{R}^d}|\phi(x)||\phi(x-j)|dx \le 2\beta\left(\frac{|j|}{2d}\right)\int_{\mathbb{R}^d}|\phi(x)|dx. \qquad \square$$

From Lemma 2.1 and (2.2) we have that the sequence $\{t_j\}$ belongs to $\ell_1(\mathbb{Z}^d)$, that is,

$$\sum_{j\in\mathbb{Z}^d}|t_j| < \infty.$$

We can then associate with ϕ the symbol of the Toeplitz matrix T, that is, the trigonometric series

$$t_\phi(x) = \sum_{j\in\mathbb{Z}^d} t_j e^{ijx}, \quad x \in \mathbb{R}^d,$$

and we assume that

$$t_\phi(x) > 0, \quad x \in \mathbb{R}^d.$$

Take $\rho \ge 0$ and let

$$f(x) = \sum_{j\in\mathbb{Z}^d} a_j e^{ijx}, \quad a_j \in \mathbb{R}, \quad x \in \mathbb{R}^d$$

with

$$\|f\|_\rho := \sum_{j\in\mathbb{Z}^d}|a_j|(1+|j|)^\rho \quad \text{and} \quad |f|_\infty = \max_{x\in\mathbb{R}^d}|f(x)|.$$

Then $\|f\cdot g\|_\rho \le \|f\|_\rho \cdot \|g\|_\rho$, since $\beta_j = (1+j)^\rho$, $j \in \mathbb{Z}_+$ is such that $\beta_{j+k} \le \beta_j\beta_k$.

We shall need the following result.

LEMMA 2.2. *There exist a positive constant $C > 0$ and an integer $k \geq 1$, depending only on d and ρ, such that*

$$|f|_\infty \leq \|f\|_\rho \leq |f|_\infty + C \max_{\substack{|\alpha|=k \\ \alpha \in \mathbb{Z}_+^d}} |D^\alpha f|_\infty, \tag{2.4}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ and

$$D^\alpha := \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}.$$

Proof. Below K_1, K_2, and C represent some constants. There is a $K_1 > 1$ such that

$$|x|^2 \leq K_1 |x|_2^2, \quad x = (x_1, x_2, \ldots, x_d)^T \in \mathbb{R}^d$$

and

$$|x|_2^2 = \sum_{j=1}^d x_j^2,$$

so that

$$\begin{aligned} \left(\sum_{j \in \mathbb{Z}^d \setminus \{0\}} |a_j| (1 + |j|)^\rho \right)^2 &\leq K_2 \left(\sum_{j \in \mathbb{Z}^d \setminus \{0\}} |a_j| |j|_2^\rho \right)^2 \\ &\leq K_2 \sum_{j \in \mathbb{Z}^d \setminus \{0\}} |j|_2^{-d-1} \sum_{j \in \mathbb{Z}^d \setminus \{0\}} |a_j|^2 |j|_2^{2\rho+d+1} \end{aligned}$$

by the Cauchy–Schwarz inequality.

Set $k := \lceil \rho + \frac{d+1}{2} \rceil$ and bound the last sum above by

$$\begin{aligned} C \max_{\substack{|\alpha|=k \\ \alpha \in \mathbb{Z}_+^d}} \sum_{j \in \mathbb{Z}^d \setminus \{0\}} |a_j|^2 j_1^{2\alpha_1} \ldots j_d^{2\alpha_d} &= C \max_{\substack{|\alpha|=k \\ \alpha \in \mathbb{Z}_+^d}} \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^d} |D^\alpha f(x)|^2 dx \\ &\leq C \max_{\substack{|\alpha|=k \\ \alpha \in \mathbb{Z}_+^d}} |D^\alpha f|_\infty^2. \end{aligned}$$

Since $|a_0| \leq |f|_\infty \leq \|f\|_\rho$, this proves (2.4). ∎

The following result and its proof extend that of Newman [20].

THEOREM 2.3. *Suppose $\rho \geq 0$ and let*

$$f(x) = \sum_{j \in \mathbb{Z}^d} a_j e^{ijx}, \quad x \in \mathbb{R}^d,$$

where

$$\|f\|_\rho < \infty.$$

Let F be an analytic function on a neighborhood of the range of f (which is a compact set). Then, $\|F \circ f\|_\rho < \infty$.

Proof. Let Θ be a bounded open set containing the range R_f of f and on which F is analytic. Choose a closed rectifiable curve Γ surrounding R_f and contained in Θ of distance $\epsilon_0 > 0$ from R_f. Thus $|\zeta - z| \geq \epsilon_0$, $\zeta \in \Gamma$, and $z \in R_f$. For any $\delta < 1/2$ choose a trigonometric polynomial P such that $\|P - f\|_\rho \leq \delta\epsilon_0$, and range $R_P \subset \Theta$. Since $|P - f|_\infty \leq \|P - f\|_\rho \leq \delta\epsilon_0$, we have for $\zeta \in \Gamma$ and $z \in R_P$ that

$$\begin{aligned} |\zeta - z| &= |\zeta - P(x_0)| = |\zeta - P(x_0) + f(x_0) - f(x_0)| \\ &\geq |\zeta - f(x_0)| - |P(x_0) - f(x_0)| \\ &\geq (1 - \delta)\epsilon_0. \end{aligned}$$

Hence by Cauchy's integral formula for $z \in R_P$,

$$\frac{F^{(k)}(z)}{k!} = \frac{1}{2\pi i} \int_\Gamma \frac{F(\zeta)}{(\zeta - z)^{k+1}} d\zeta,$$

we have

$$\left| \frac{F^{(k)}(z)}{k!} \right| \leq \frac{1}{2\pi} \left(\frac{1}{(1 - \delta)\epsilon_0} \right)^{k+1} \max_{x \in \Theta} |F(x)|.$$

For $n \geq 0$ we set for $x \in \mathbb{R}^d$

$$f_n(x) = (F^{(n)} \circ P)(x).$$

Then for each $k \in \mathbb{Z}_+$, by the chain rule, there is a positive constant K_1 (depending on P and k) such that for $|\alpha| = k$,

$$|D^\alpha f_n|_\infty \leq K_1 \max_{0 \leq j \leq k} \left| F^{(n+j)} \circ P \right|_\infty .$$

Thus there is a constant $K_2 > 0$ such that

$$\max_{|\alpha| = k} \frac{|D^\alpha f_n|_\infty}{n!} \leq \frac{K_2(1 + n^k)}{((1 - \delta)\varepsilon_\circ)^n}.$$

Hence, for k as in Lemma 2.2, there is a constant $K_3 > 0$ with

$$\frac{\|f_n\|_\rho}{n!} \leq \frac{K_3(1 + n^k)}{((1 - \delta)\epsilon_0)^n}.$$

For $x \in \mathbb{R}^d$, since F is analytic on Θ, a set which contains both $f(x)$ and $P(x)$, we have that

$$(F \circ f)(x) = \sum_{n=0}^{\infty} \frac{f_n(x)}{n!} (f(x) - P(x))^n.$$

Thus

$$\|F \circ f\|_\rho \leq \sum_{n=0}^{\infty} \frac{\|f_n\|_\rho}{n!} \|(f - P)^n\|_\rho \leq K_3 \sum_{n=0}^{\infty} (1 + n^k) \left(\frac{\delta}{1 - \delta} \right)^n < \infty. \qquad \square$$

To state the main theorem of this section we shall need a total ordering on $\mathbb{Z}^d$, denoted by the symbol $\preceq$, having the additional property

$$\alpha \succeq 0 \quad \text{if and only if} \quad -\alpha \preceq 0,$$

for any $\alpha \in \mathbb{Z}^d$. We write $\alpha \prec \beta$ whenever $\alpha \neq \beta$ and $\alpha \preceq \beta$.

THEOREM 2.4. *Suppose $\rho \geq 0$ and assume that β is a decreasing, positive function on $[0,\infty)$ satisfying*

$$\int_0^\infty t^{\rho+d-1}\beta(t)dt < \infty. \tag{2.5}$$

Then t_ϕ has a spectral factorization

$$t_\phi(x) = h(-x)h(x), \quad x \in \mathbb{R}^d, \tag{2.6}$$

where

$$h(x) = \sum_{j\succeq 0} \sigma_j e^{ijx}, \quad x \in \mathbb{R}^d, \tag{2.7}$$

$\sigma_j \in \mathbb{R}$, *for* $j \succeq 0$, *and*

$$\sum_{j\succeq 0} |\sigma_j|(1+|j|)^\rho < \infty.$$

Moreover,

$$\frac{1}{h(x)} = \sum_{j\succeq 0} \gamma_j e^{ijx}, \quad x \in \mathbb{R}^d, \tag{2.8}$$

where $\gamma_j \in \mathbb{R}$, $j \succeq 0$, *and*

$$\sum_{j\succeq 0} |\gamma_j|(1+|j|)^\rho < \infty.$$

Furthermore, if β decays exponentially, i.e., there are constants $c > 0$ and λ in $(0,1)$ with $\beta(t) \leq c\lambda^t$, $t \geq 0$, then the sequences $\{\sigma_j : j \succeq 0\}$, $\{\gamma_j : j \succeq 0\}$ in (2.7) *and* (2.8) *also decay exponentially, i.e., there are constants $k > 0$ and μ in $(0,1)$ with $|\sigma_j|, |\gamma_j| \leq k\mu^{|j|}$, $j \succeq 0$.*

Proof. By (2.5) and Lemma 2.1 we have that

$$\sum_{j\in\mathbb{Z}^d} |t_j|(1+|j|)^\rho < \infty.$$

Since the range of t_ϕ lies in $(0,\infty)$ we may apply Theorem 2.3 to show that the function $f := \log t_\phi$ has a trigonometric series expansion

$$f(x) = \sum_{j\in\mathbb{Z}^d} f_j e^{ijx}, \quad x \in \mathbb{R}^d, \tag{2.9}$$

where $f_j = f_{-j}$, $f_j \in \mathbb{R}$, $j \in \mathbb{Z}^d$, and

$$\sum_{j\in\mathbb{Z}^d} |f_j|(1+|j|)^\rho < \infty.$$

Now, decompose f as $f = f_- + f_+$, where

$$f_-(x) = \frac{1}{2}f_0 + \sum_{j\prec 0} f_j e^{ijx}, \quad f_+(x) = \frac{1}{2}f_0 + \sum_{j\succ 0} f_j e^{ijx}, \quad x \in \mathbb{R}^d,$$

and the symbol $\succ$ denotes the order specified before.

Theorem 2.3 may be applied to show that the function $h := \exp f_+$ has a trigonometric series (2.7), where $\sigma_j \in \mathbb{R}$, $j \succeq 0$, and

$$\sum_{j \succeq 0} |\sigma_j|(1+|j|)^\rho < \infty.$$

Then for x in $\mathbb{R}^d$,

$$\begin{aligned} t_\phi(x) &= \exp f(x) = \exp f_-(x) \exp f_+(x) \\ &= \exp f_+(-x) \exp f_+(x) = h(-x)h(x), \end{aligned}$$

which gives (2.6). For $x \in \mathbb{R}^d$, $t_\phi(x) > 0$ and so $h(x) \neq 0$. Thus we can apply Theorem 2.3 again to show that the function $1/h$ has a trigonometric series (2.8), where $\gamma_j \in \mathbb{R}$, $j \succeq 0$, and

$$\sum_{j \succeq 0} |\gamma_j|(1+|j|)^\rho < \infty.$$

Moreover, when $\beta(t) \leq c\lambda^t$, $t \geq 0$, where $c > 0$ and $\lambda \in (0,1)$, then for j in $\mathbb{Z}^d$, by Lemma 2.1,

$$|t_j| \leq Ac\lambda^{a|j|}$$

for some constants $A, a > 0$. Thus there are two constants $k > 0$ and $\mu \in (\lambda, 1)$ such that $|t_j| \leq k\mu^{|j|}$. In this case t_ϕ can be extended to a function

$$T_\phi(z) = \sum_{j \in \mathbb{Z}^d} t_j z^j$$

analytic on some neighborhood of the torus

$$T^d := \big\{ z = (z_1, z_2, \dots, z_d) \in \mathbb{C}^d : |z_1| = |z_2| = \cdots = |z_d| = 1 \big\}.$$

Since $T_\phi(z) > 0$ on some neighborhood of T^d, it follows that $F(z) := \log T_\phi(z)$ is analytic on some neighborhood of T^d. With $F(e^{ix}) = f(x)$, the coefficients $\{f_j\}$ in (2.9) decay exponentially as $|j| \to \infty$. Let

$$F_+(z) = \frac{1}{2} f_0 + \sum_{j \succ 0} f_j z^j$$

and $H(z) = \exp F_+(z)$. Then H and $1/H$ are analytic on a neighborhood of T^d with $H(e^{ix}) = h(x)$, and the coefficients $\{\sigma_j : j \succeq 0\}$ and $\{\gamma_j : j \succeq 0\}$ in (2.7) and (2.8) decay exponentially. Also $T_\phi(z) = H(z^{-1})H(z)$ holds on some neighborhood of T^d which gives (2.6). □

In the one-dimensional case ($d = 1$), spectral factorization problems of the type (2.6) have been studied at length by Krein [15] and by Calderon, Spitzer, and Widom [5]. In particular, for $\rho = 0$ and $d = 1$, the existence of (2.6) is an immediate consequence of the necessary and sufficient conditions for spectral factorizability stated in the aforementioned papers.

Setting $\sigma_j = 0$ for $j \prec 0$ we let $L_{jk} = \sigma_{j-k}$ for $j, k \in \mathbb{Z}^d$ and let L denote the matrix $L = (L_{jk})_{j,k \in \mathbb{Z}^d}$. Note that L is lower triangular, i.e., $L_{jk} = 0$ for $j \prec k$. Also for j in $\mathbb{Z}^d$,

$$\sum_{k \in \mathbb{Z}^d} |L_{jk}| = \sum_{k \in \mathbb{Z}^d} |\sigma_k| < \infty,$$

and so L gives a bounded operator on $L_\infty(\mathbb{Z}^d)$. Similarly the Gram matrix T defined in (2.3) is a bounded operator on $L_\infty(\mathbb{Z}^d)$.

From (2.6) and (2.7) we have

$$t_j = \sum_{\ell\in\mathbb{Z}^d} \sigma_{j+\ell}\sigma_\ell, \quad j \in \mathbb{Z}^d,$$

and so, for $j, k \in \mathbb{Z}^d$,

$$T_{jk} = t_{j-k} = \sum_{\ell\in\mathbb{Z}^d} \sigma_{j-k+\ell}\sigma_\ell = \sum_{\ell\in\mathbb{Z}^d} \sigma_{j-\ell}\sigma_{k-\ell} = \sum_{\ell\in\mathbb{Z}^d} L_{j\ell}L_{k\ell} = (LL^T)_{jk}.$$

Thus $T = LL^T$, which gives the Cholesky factorization of the Gram matrix T, relative to the ordering $\prec$.

Now, (2.7) and (2.8) give for j in $\mathbb{Z}^d$,

$$\sum_{\ell\in\mathbb{Z}^d} \sigma_{j-\ell}\gamma_\ell = \delta_{j,0},$$

and so

$$\sum_{\ell\in\mathbb{Z}^d} L_{j\ell}\gamma_{\ell-k} = \delta_{j,k} = \sum_{\ell\in\mathbb{Z}^d} \gamma_{j-\ell}L_{\ell k},$$

where $\gamma_j = 0$ if $j \prec 0$. Thus L has the lower triangular inverse L^{-1} given by

$$(L^{-1})_{jk} = \gamma_{j-k}, \quad j, k \in \mathbb{Z}^d. \tag{2.10}$$

Since

$$\sum_{j\in\mathbb{Z}^d} |\gamma_j| < \infty,$$

L^{-1} gives a bounded inverse for L on $L_\infty(\mathbb{Z}^d)$. As a result, since L and L^{-1} are lower triangular matrices both bounded in $\ell_\infty(\mathbb{Z}^d)$, the Cholesky factorization $T = LL^T$ gives the spectral factorization of T with respect to the fixed ordering.

Now we define

$$\psi(x) = \sum_{j\succeq 0} \gamma_i \phi(x + j), \quad x \in \mathbb{R}^d$$

and write $\psi_j := \psi(\cdot - j)$, $j \in \mathbb{Z}^d$. Recall that

$$\sum_{j\succeq 0} |\gamma_j| < \infty$$

and ϕ is bounded and integrable on $\mathbb{R}^d$. It follows that ψ is bounded and integrable on $\mathbb{R}^d$. Moreover, for j in $\mathbb{Z}^d$, x in $\mathbb{R}^d$,

$$\psi_j(x) = \sum_{k\succeq 0} \gamma_k \phi(x - j + k) = \sum_{k\in\mathbb{Z}^d} L^{-1}_{jk}\phi(x - k),$$

and so, for $j, k \in \mathbb{Z}^d$,

$$\int_{\mathbb{R}^d} \psi_j(x)\psi_k(x)dx = \sum_{\ell,m\in\mathbb{Z}^d} L^{-1}_{j\ell}L^{-1}_{km}\int_{\mathbb{R}^d} \phi_\ell(x)\phi_m(x)dx$$
$$= (L^{-1}T(L^{-1})^T)_{jk} = (L^{-1}LL^T(L^{-1})^T)_{jk} = \delta_{j,k}.$$

Therefore the set of functions $\{\psi_j : j \in \mathbb{Z}^d\}$ are orthonormal.

We shall now consider orthonormalizing a finite subset of the functions $\{\phi_j : j \in \mathbb{Z}^d\}$ restricted to a bounded subset of $\mathbb{R}^d$. With this aim we now define the ordering $\preceq$ on $\mathbb{Z}^d$ as the lexicographical ordering with respect to

$$\alpha_1 + \alpha_2 + \cdots + \alpha_d, \alpha_2 + \alpha_3 + \cdots + \alpha_d, \ldots, \alpha_{d-1} + \alpha_d, \alpha_d,$$

i.e., $\alpha \prec \beta$ if for some k, $1 \le k \le d$,

$$\alpha_k + \alpha_{k+1} + \cdots + \alpha_d < \beta_k + \beta_{k+1} + \cdots + \beta_d$$

and for $1 \le j < k$,

$$\alpha_j + \alpha_{j+1} + \cdots + \alpha_d = \beta_j + \beta_{j+1} + \cdots + \beta_d.$$

Fix a number η in $(0, \frac{1}{d})$. For $n = 1, 2, \ldots$, let $J_n = \{j \in \mathbb{Z}^d_+ : |j| \le (1-\eta)n\}$ and $\Omega_n = \{x \in \mathbb{R}^d_+ : |x| \le n\}$. For j in J_n, let ϕ^n_j denote ϕ_j restricted to Ω_n.

We construct orthonormal functions ψ^n_j, $j \in J_n$, on Ω_n by the Gram–Schmidt process applied to ϕ^n_j, $j \in J_n$, with respect to the prefixed ordering $\preceq$. Thus for $j \in J_n$,

$$\phi^n_j = \sum_{k\in J_n} L^n_{jk}\psi^n_k$$

for a nonsingular lower triangular matrix $L^n = (L^n_{jk})_{j,k\in J_n}$.

Let G^n denote the Gram matrix $(G^n_{jk})_{j,k\in J_n}$:

$$G^n_{jk} = \int_{\Omega_n} \phi^n_j(x)\phi^n_k(x)dx = \int_{\Omega_n} \phi_j(x)\phi_k(x)dx. \tag{2.11}$$

Then

$$G^n_{jk} = \sum_{\ell,m\in J_n} L^n_{j\ell}L^n_{km}\int_{\Omega_n} \psi^n_\ell(x)\psi^n_m(x)dx = \sum_{\ell\in J_n} L^n_{j\ell}L^n_{k\ell},$$

and so $G^n = L^n(L^n)^T$.

We shall show at the end of section 3 that, under certain conditions on β, for all large enough n and j in J_n with $j_\ell \ge \eta n$, $\ell = 1, \ldots, d$,

$$\sup_{x\in\Omega_n} \left|\psi^n_j(x) - \psi_j(x)\right| \to 0 \quad \text{as} \quad n \to \infty,$$

with a convergence rate depending on β. Let us begin the demonstration by recording some matrix theoretic facts. To this end, take j in J_n with $j_\ell \ge \eta n$, $\ell = 1, \ldots, d$.

Then for all $x \in \Omega_n$,

$$\begin{aligned}\left|\psi_j^n(x) - \psi_j(x)\right| &= \left|\sum_{k\in J_n} (L^n)^{-1}_{jk}\phi_k^n(x) - \sum_{k\in\mathbb{Z}^d} L^{-1}_{jk}\phi_k(x)\right| \\ &\le \sum_{k\in J_n}\left|\left((L^n)^{-1}_{jk} - L^{-1}_{jk}\right)\phi_k(x)\right| + \sum_{k\notin J_n}\left|L^{-1}_{jk}\phi_k(x)\right| \\ &\le K\sum_{k\in J_n}\left|(L^n)^{-1}_{jk} - L^{-1}_{jk}\right| + K\sum_{k\notin J_n}\left|L^{-1}_{jk}\right| \end{aligned} \tag{2.12}$$

for some constant $K > 0$. Now, suppose for $\rho > 0$ that (2.5) is satisfied. We note for $k \preceq j$, $k \notin J_n$, that we must have $k_\ell < 0$ for some ℓ, $1 \le \ell \le d$, and so

$$|j - k| \ge \min\{j_\ell : \ell = 1, \dots, d\} \ge \eta n.$$

By (2.10) we obtain the estimate

$$\sum_{k\notin J_n} |L^{-1}_{jk}| = \sum_{k\preceq j} |\gamma_{j-k}| \le \sum_{k\preceq j} |\gamma_{j-k}| \frac{(1+|j-k|)^\rho}{(\eta n)^\rho} \le Cn^{-\rho} \tag{2.13}$$

for a constant $C > 0$, by Theorem 2.4. Also, if β decays exponentially, then Theorem 2.4 shows that $|L^{-1}_{jk}| \le c\mu^{|j-k|}$ for constants $c > 0$ and μ in $(0, 1)$. Consequently, we obtain that

$$\begin{aligned}\sum_{k\notin J_n} |L^{-1}_{jk}| &\le c\sum_{\ell=1}^{d}\sum_{\substack{j\in\mathbb{Z}^d \\ j_\ell<0}} \mu^{|j-k|} \\ &= c\sum_{\ell=1}^{d}\sum_{m=1}^{\infty} \mu^{j_\ell+m}\left\{\sum_{r=-\infty}^{\infty}\mu^{|r|}\right\}^{d-1} \le c_1\mu^{\eta n}\end{aligned} \tag{2.14}$$

for some $c_1 > 0$.

In summary, we have considered the decay of the second term in (2.12) and it remains to consider the first term in (2.12). We recall that L is the Cholesky factor of the bi-infinite Gram matrix T, while L^n is the Cholesky factor of the finite Gram matrix G^n. In the next section we shall consider in more generality the connection between Cholesky factors of bi-infinite and related finite matrices. In order to apply these results we shall need to examine the connection between the matrices T, G^n and the matrix $G = (G_{jk})_{j,k\in\mathbb{Z}^d}$ defined by

$$G_{jk} = \int_{[0,\infty)^d} \phi_j(x)\phi_k(x)dx. \tag{2.15}$$

First, we note some simple properties of any decreasing positive function β on $[0, \infty)$.

LEMMA 2.5. *Let β be the function defined in section 2 and $i, j \in \mathbb{Z}_+$.*

(i) *If $x \le \beta(i) + \beta(j)$ and $x \le \beta(|i - j|)$, then $x \le 2\beta(\frac{1}{4}(i + j))$.*

(ii) *If $x \le \beta(i)$ and $x \le \beta(j)$, then $x \le \beta(\frac{1}{2}(i + j))$.*

Proof. To prove (i), without loss of generality we suppose $i \le j$. If $i \ge \frac{1}{2}j$, then $x \le \beta(i) + \beta(j) \le 2\beta(\frac{1}{2}j) \le 2\beta(\frac{1}{4}(i + j))$. If $i \le \frac{1}{2}j$, then $x \le \beta(|i - j|) = \beta(j - i) \le \beta(\frac{1}{2}j) \le \beta(\frac{1}{4}(i + j))$.

Moreover, $\max\{i,j\} \geq \frac{1}{2}(i+j)$, and so $x \leq \beta(\max\{i,j\}) \leq \beta(\frac{1}{2}(i+j))$. This proves (ii). ∎

Furthermore, we note that Lemma 2.1 can be expressed as follows: there are constants $A, a > 0$ such that

$$|T_{ij}| \leq A\beta(a|j-i|) \tag{2.16}$$

for all $i, j \in \mathbb{Z}^d$.

LEMMA 2.6. *There are numbers $B, b > 0$ such that*

$$|T_{ij} - G_{ij}| \leq B \sum_{\ell=1}^{d} \beta\big(b(i_\ell + j_\ell + |j-i|)\big), \quad \forall i, j \in \mathbb{Z}^d, \quad i, j \geq 0.$$

Proof.

$$\begin{aligned} |T_{ij} - G_{ij}| &= \left| \int_{\mathbb{R}^d \setminus [0,\infty)^d} \phi(x-i)\phi(x-j)dx \right| \\ &\leq \sum_{\sigma} \int_{[0,\infty)^d} |\phi(\sigma x - i)|\, |\phi(\sigma x - j)| dx, \end{aligned}$$

where $\sigma x = (\pm x_1, \ldots, \pm x_d)$ and we sum over all σ except $\sigma x = (x_1, \ldots, x_d)$. Take any such σ and choose ℓ with $(\sigma x)_\ell = -x_\ell$. Then

$$\begin{aligned} \int_{[0,\infty)^d} |\phi(\sigma x - i)|\, |\phi(\sigma x - j)| dx &\leq \int_{[0,\infty)^d} \beta(x_\ell + i_\ell) |\phi(\sigma x - j)| dx \\ &\leq \beta(i_\ell) \int_{\mathbb{R}^d} |\phi(x)| dx. \end{aligned}$$

Similarly, we have that

$$\int_{[0,\infty)^d} |\phi(\sigma x - i)|\, |\phi(\sigma x - j)| dx \leq \beta(j_\ell) \int_{\mathbb{R}^d} |\phi(x)| dx.$$

Consequently, there exists a $C > 0$ with

$$|T_{ij} - G_{ij}| \leq C \sum_{\ell=1}^{d} (\beta(i_\ell) + \beta(j_\ell)).$$

Also, we note that

$$|T_{ij} - G_{ij}| \leq 2 \int_{\mathbb{R}^d} |\phi(x-i)\phi(x-j)| dx \leq 2A\beta(a|j-i|),$$

as in (2.16). The result now follows from Lemma 2.5. ∎

LEMMA 2.7. *There exist numbers $B, b > 0$ such that*

$$|G_{ij} - G^n_{ij}| \leq B \sum_{\ell=1}^{d} \beta\big(b(2n - i_\ell - j_\ell + |j-i|)\big)$$

for all $i, j \in J_n$.

Proof. Since

$$\begin{aligned}|G_{ij} - G^n_{ij}| &\le \int_{[0,\infty)^d \setminus [0,n]^d} |\phi(x-i)|\,|\phi(x-j)|dx \\ &= \int_{(-\infty,n]^d \setminus [0,n]^d} |\phi(n-x-i)|\,|\phi(n-x-j)|dx \\ &\le \int_{\mathbb{R}^d \setminus [0,\infty)^d} |\phi(-x+n-i)|\,|\phi(-x+n-j)|dx,\end{aligned}$$

the result follows as in Lemma 2.6. ∎

LEMMA 2.8. *There are numbers $C, c > 0$ such that*

$$|T_{ij} - G^n_{ij}| \le C \sum_{\ell=1}^{d} \beta\big(c(i_\ell + j_\ell + |j-i|)\big)$$

for all i, j in J_n.

Proof. For $\ell = 1, \dots, d$, when $i_\ell + j_\ell \le 2(1-\eta)n$ it follows that $2n \ge \frac{i_\ell + j_\ell}{1-\eta}$ and $2n - i_\ell - j_\ell \ge (\frac{1}{1-\eta} - 1)(i_\ell + j_\ell)$. Therefore we conclude that

$$\beta\big(b\,(2n - i_\ell - j_\ell + |j-i|)\big) \le \beta\big(b(((1-\eta)^{-1} - 1)(i_\ell + j_\ell) + |j-i|)\big),$$

from which the result follows from Lemmas 2.6 and 2.7. ∎

LEMMA 2.9. *Let $G_{J_n} = (G_{ij})_{i,j \in J_n}$. Then*

$$\|G_{J_n} - G^n\|_2 \le K n^d \beta(kn)$$

for constants $K, k > 0$.

Proof. As $i_\ell + j_\ell \le 2(1-\eta)n$, $\ell = 1, \dots, d$, we conclude that

$$\beta\big(b(2n - i_\ell - j_\ell + |j-i|)\big) \le \beta\big(b(2n - (i_\ell + j_\ell))\big) \le \beta(2b\eta n).$$

Invoking Lemma 2.7, we obtain the estimate

$$|G_{ij} - G^n_{ij}| \le Bd\beta(2b\eta n),$$

and therefore

$$\|G_{J_n} - G^n\|_2^2 \le \sum_{i,j \in J_n} |G_{ij} - G^n_{ij}|^2 \le n^{2d} B^2 d^2 \beta(2b\eta n)^2,$$

which gives the result. ∎

3. Cholesky factorization. As before we require the function β to satisfy (2.2). We note that for some constant K,

$$\sum_{k \in \mathbb{Z}^d} \beta(|k|) \le K \sum_{j=1}^{\infty} j^{d-1} \beta(j) \le K \int_0^\infty t^{d-1} \beta(t) dt,$$

which is finite by (2.2). We write

$$\|\beta\| := \sum_{k \in \mathbb{Z}^d} \beta(|k|),$$

and for any subset I of $\mathbb{Z}^d$ we let M_I denote the set $\{A : A = (A_{jk})_{j,k\in I}\}$ of real matrices on I. For A in M_I we let

$$\|A\|_2 = \sup\{\|Ax\|_2 : \|x\|_2 \le 1\}$$

be the usual $\ell_2(I)$ operator norm and define the norm

$$\|A\|_T = \sup\{|A_{jk}|/\beta(|j-k|) : j,k \in I\}.$$

LEMMA 3.1. *For any $I \subseteq \mathbb{Z}^d$ and A in M_I,*

$$\|A\|_2 \le \|\beta\|\, \|A\|_T. \tag{3.1}$$

Proof. We may extend any vector $x = (x_j : j \in I)$ in $\ell_2(I)$ to a vector in $\ell_2(\mathbb{Z}^d)$ by setting $x_j = 0$ for $j \notin I$. Similarly, we set $A_{jk} = 0$ for $(j,k) \notin I^2$. This process does not alter $\|x\|_2$ or $\|A\|_2$. Now, take x in $\ell_2(I)$ and for each k in $\mathbb{Z}^d$ define a vector $y_k = (A_{j,j+k}x_{j+k} : j \in \mathbb{Z}^d)$. Then

$$\|y_k\|_2^2 = \sum_{j\in\mathbb{Z}^d} |A_{j,j+k}x_{j+k}|^2 \le \sum_{j\in\mathbb{Z}^d} \|A\|_T^2\beta(|k|)^2|x_{j+k}|^2 = \|A\|_T^2\beta(|k|)^2\|x\|_2^2.$$

Hence we get

$$\|Ax\|_2 = \left\| \sum_{k\in\mathbb{Z}^d} y_k \right\|_2 \le \sum_{k\in\mathbb{Z}^d} \|y_k\|_2 \le \|A\|_T\|x\|_2 \sum_{k\in\mathbb{Z}^d} \beta(|k|),$$

which implies (3.1). □

REMARK 3.1. *This result holds for the ℓ^p-norm $\|A\|_p$, $1 \le p \le \infty$, as well.*

For any $I \subseteq \mathbb{Z}^d$, we define $I_+ = \{j \in I : j_1, \dots, j_d \ge 0\}$. For any matrix A in M_I we define A_+ in M_I to be the restriction of A to I_+. For $I \subseteq \mathbb{Z}_+^d$, A in M_I, and $c > 0$ we define

$$\|A\|_{H,c} = \sup\left\{ |A_{jk}| \,/\, \sum_{l=1}^{d} \beta\left(c\max\{j_l + k_l, |j-k|\}\right) : j,k, \in I \right\}.$$

Since

$$\|A\|_{H,c} \ge \frac{1}{d} \sup\left\{|A_{jk}|/\beta(c|j-k|) : j,k \in I\right\},$$

Lemma 3.1 gives the inequality

$$\|A\|_2 \le d\, \|\beta(c\,\cdot)\|\, \|A\|_{H,c}. \tag{3.2}$$

LEMMA 3.2. *For any $I \subseteq \mathbb{Z}^d$ and A, B in M_I,*

$$\|(AB)_+ - A_+B_+\|_{H,\frac{1}{2}} \le 2\, \|\beta\|\, \|A\|_T\|B\|_T.$$

Proof. For any j, k in I_+ we have that

$$\begin{aligned} |((AB)_+)_{jk} - (A_+B_+)_{jk}| &\le \sum_{\ell\in I\setminus I_+} |A_{j\ell}B_{\ell k}| \\ &\le \|A\|_T\|B\|_T \sum_{i=1}^{d} \sum_{\substack{\ell_i<0 \\ \ell\in\mathbb{Z}^d}} \beta(|j-\ell|)\,\beta(|\ell-k|). \end{aligned} \tag{3.3}$$

For any ℓ in $\mathbb{Z}^d$, either $|j-\ell| \ge \frac{1}{2}|j-k|$ or $|\ell-k| \ge \frac{1}{2}|j-k|$, and so

$$\sum_{\substack{\ell_i<0\\ \ell\in\mathbb{Z}^d}} \beta(|j-\ell|)\,\beta(|\ell-k|) \le 2\,\|\beta\|\,\beta\left(\frac{1}{2}|j-k|\right). \tag{3.4}$$

When $1 \le i \le d$ and $\ell_i < 0$ it follows that $|j-\ell| \ge j_i - \ell_i \ge j_i$, and so

$$\sum_{\substack{\ell_i<0\\ \ell\in\mathbb{Z}^d}} \beta(|j-\ell|)\,\beta(|\ell-k|) \le \beta(j_i)\,\|\beta\|.$$

Similarly this sum is bounded by $\beta(k_i)\|\beta\|$ and therefore is also bounded by $\beta(\frac{1}{2}(j_i + k_i))\|\beta\|$. Thus we conclude that

$$\begin{aligned}\sum_{i=1}^{d}\sum_{\substack{\ell_i<0\\ j\in\mathbb{Z}^d}} \beta(|j-\ell|)\,\beta(|\ell-k|) &\le 2\,\|\beta\|\sum_{i=1}^{d}\min\left\{\beta\left(\frac{1}{2}(j_i+k_i)\right),\beta\left(\frac{1}{2}|j-k|\right)\right\}\\ &= 2\|\beta\|\sum_{i=1}^{d}\beta\left(\max\left\{\frac{1}{2}(j_i+k_i),\frac{1}{2}|j-k|\right\}\right).\end{aligned}$$

Using inequality (3.3) we obtain

$$|((AB)_+)_{jk} - (A_+B_+)_{jk}| \le 2\,\|A\|_T\|B\|_T\|\beta\|\sum_{i=1}^{d}\beta\left(\frac{1}{2}\max\{j_i+k_i,|j-k|\}\right),$$

which gives the result. ∎

LEMMA 3.3. *Suppose $0 \le c \le 1$. Then for any $I \subseteq \mathbb{Z}^d_+$ and A, B in M_I*

$$\begin{aligned}\|AB\|_{H,\frac{1}{2}c} &\le 2\,\|\beta(c\,\cdot)\|\,\|A\|_T\|B\|_{H,c},\\ \|AB\|_{H,\frac{1}{2}c} &\le 2\,\|\beta(c\,\cdot)\|\,\|B\|_T\|A\|_{H,c}.\end{aligned}$$

Proof. For j,k in I we have that

$$\begin{aligned}|(AB)_{jk}| \le{}& \sum_{\ell\in I}|A_{j\ell}B_{\ell k}| \le \|A\|_T\|B\|_{H,c}\\ &\times\sum_{\ell\in\mathbb{Z}^d_+}\beta(|j-\ell|)\sum_{i=1}^{d}\beta\big(c\,\max\{\ell_i+k_i,|\ell-k|\}\big).\end{aligned} \tag{3.5}$$

Now, for $1 \le i \le d$, we observe that

$$\begin{aligned}\sum_{\ell\in\mathbb{Z}^d_+}\beta(|j-\ell|)\,\beta\big(c\,\max\{\ell_i+k_i,|\ell-k|\}\big) &\le \sum_{\ell\in\mathbb{Z}^d_+}\beta(c|j-\ell|)\,\beta(c|\ell-k|)\\ &\le 2\,\|\beta(c\,\cdot)\|\,\beta\left(\frac{1}{2}\,c|j-k|\right),\end{aligned} \tag{3.6}$$

as in (3.4). Also, if $\ell_i \le \frac{1}{2}j_i - \frac{1}{2}k_i$, then $|j-\ell| \ge j_i - \ell_i \ge \frac{1}{2}j_i + \frac{1}{2}k_i$, from which it follows that

$$\beta(|j-\ell|) \le \beta\left(\frac{1}{2}(j_i+k_i)\right) \le \beta\left(\frac{1}{2}c(j_i+k_i)\right).$$

Moreover, if $\ell_i \geq \frac{1}{2}j_i - \frac{1}{2}k_i$, then $\ell_i + k_i \geq \frac{1}{2}j_i + \frac{1}{2}k_i$ and so we obtain

$$\sum_{\ell \in \mathbb{Z}_+^d} \beta(|j-\ell|)\, \beta\big(c \max\{\ell_i + k_i, |\ell - k|\}\big) \leq 2\, \|\beta(c)\|\, \beta\left(\frac{1}{2}c(j_i + k_i)\right).$$

Using (3.6) and (3.5) successively we see that

$$\sum_{\ell \in \mathbb{Z}_+^d} \beta(|j-\ell|)\, \beta\big(c \max\{\ell_i + k_i, |\ell - k|\}\big) \leq 2\, \|\beta(c\,\cdot)\|\, \beta\left(\frac{1}{2}\, c\, \max\{j_i + k_i, |j-k|\}\right)$$

and

$$\|AB\|_{H,\frac{1}{2}c} \leq 2\, \|\beta(c\,\cdot)\|\, \|A\|_T \|B\|_{H,c}.$$

The other inequality follows similarly. □

To prove our next result we shall need the following theorem.

THEOREM A (see Sun [23]). *Let A be a finite positive real symmetric matrix with $A = LL^T$ its Cholesky factorization. If E is a symmetric matrix satisfying $\|E\|_F \leq \frac{1}{2}\|A^{-1}\|_2^{-1}$, then the unique Cholesky factorization of $A+E = (L+G)(L+G)^T$ satisfies*

$$\|G\|_F \leq \sqrt{2}\, \|A\|_2^{\frac{1}{2}} \|A^{-1}\|_2 \|E\|_F.$$

For $n \in \mathbb{Z}_+$, we let $I_n = \{j \in \mathbb{Z}_+^d : |j| \leq n\}$.

LEMMA 3.4. *For $n \in \mathbb{Z}_+$, let A_n be a symmetric positive definite matrix in $M_{I_{K_n}}$, $K_n \in \mathbb{Z}_+$, such that $\{\|A_n^{-1}\|_2 : n \in \mathbb{Z}_+\}$ is bounded. Suppose that $\|A_n - I\|_{H,1}$ is uniformly bounded and A_n has Cholesky factorization $A_n = L_n L_n^T$. We assume $t^d\beta(t) \to 0$ as $t \to \infty$ and take $\rho > 1$. Then there are constants M, K so that for any $m \geq M$ and $m \leq K_n \leq \rho m$,*

$$\sum_{j,k \in I_{K_n}} \big\{|(L_n)_{jk} - \delta_{jk}|^2 : j_\ell \geq m,\ \ell = 1, \ldots, d \text{ or } k_\ell \geq m,\ \ell = 1, \ldots, d\big\} \leq K m^{2d} \beta(m)^2.$$

Proof. When $m \in \mathbb{Z}_+$, we define

$$J_m = \{j \in \mathbb{Z}^d : j_\ell \leq m-1 \text{ for some } \ell,\ \ell = 1, \ldots, d\}.$$

For $0 \leq m \leq K_n \leq \rho m$, we introduce the matrix $A_{n,m}$ in $M_{I_{K_n}}$ defined by

$$(A_{n,m})_{jk} = \begin{cases} (A_n)_{jk}, & j,k \in I_{K_n} \cap J_m, \\ \delta_{j,k} & \text{otherwise.} \end{cases}$$

Let $E_{n,m} = A_n - A_{n,m}$ and observe that $(E_{n,m})_{jk} = 0$ when $j,k \in I_{K_n} \cap J_m$. Suppose $j,k \in I_{K_n}$, $j \notin J_m$, or $k \notin J_m$, then

$$\begin{aligned} |(E_{n,m})_{jk}| &= |(A_n - I)_{j,k}| \\ &\leq \|A_n - I\|_{H,1} \sum_{\ell=1}^d \beta\big(\max\{j_\ell + k_\ell,\, |j-k|\}\big) \\ &\leq C \sum_{\ell=1}^d \beta(j_\ell + k_\ell) \end{aligned}$$

for some $C > 0$. Now, either $j_\ell \geq m$, $\ell = 1, \ldots, d$ or $k_\ell \geq m$, $\ell = 1, \ldots, d$. Therefore for $\ell = 1, \ldots, d$, $j_\ell + k_\ell \geq m$ and $\beta(j_\ell + k_\ell) \leq \beta(m)$, from which it follows that

$$|(E_{n,m})_{j,k}| \leq Cd\beta(m)$$

and also

$$\|E_{n,m}\|_F \leq Cd\beta(m)K_n^d \leq C_1 m^d \beta(m) \tag{3.7}$$

for a constant $C_1 > 0$. Since

$$\lim_{m\to\infty} m^d \beta(m) = 0$$

and $\{\|A_n^{-1}\|_2 : n \in \mathbb{Z}_+\}$ is bounded, we can choose M so that for all $m \geq M$ and $m \leq K_n \leq \rho m$,

$$\|E_{n,m}\|_F < \frac{1}{2}\,\|A_n^{-1}\|_2^{-1}.$$

Now, the Cholesky factor of $A_{n,m}$ is $L_{n,m}$, obtained from L_n by the same process as we obtained $A_{n,m}$ from A_n. Since $\|A_n - I\|_{H,1}$ is uniformly bounded, we see from (3.2) that $\|A_n - I\|_2$ is uniformly bounded and hence so is $\|A_n\|_2$. We can apply Theorem A to obtain

$$\|L_n - L_{n,m}\|_F \leq C_2\|E_{n,m}\|_F \leq C_3 m^d \beta(m)$$

for constants $C_2, C_3 \geq 0$, by (3.7). Thus we conclude that

$$\sum_{j,k\in I_{K_n}} \left\{|(L_n)_{jk} - \delta_{jk}|^2 : j_\ell \geq m,\ \ell = 1, \ldots, d \text{ or } k_\ell \geq m,\ \ell = 1, \ldots, d\right\}$$
$$\leq C_3^2 m^{2d}\beta(m)^2. \qquad \square$$

THEOREM 3.5. *For $n \in \mathbb{Z}_+$, let A_n be a symmetric positive definite matrix in $M_{I_{K_n}}$, $K_n \in \mathbb{Z}_+$, such that $\{\|A_n^{-1}\|_2 : n \in \mathbb{Z}_+\}$ is bounded. Suppose that A_n has Cholesky factorization $A_n = L_n L_n^T$. Let L in $M_{\mathbb{Z}_+^d}$ be lower triangular with $\|L\|_T$, $\|L^{-1}\|_T$ finite and $\|A_n - (LL^T)_{I_{K_n}}\|_{H,1}$ uniformly bounded. We assume $t^d\beta(t) \to 0$ as $t \to \infty$ and take $\rho > 1$. Then there are constants M, K so that for $m \geq M$ and j, k in I_{K_n} with $m \leq K_n \leq \rho m$, the following hold.*

If either $j_\ell \geq \frac{5}{4}m$, $\ell = 1, \ldots, d$ or $k_\ell \geq m$, $\ell = 1, \ldots, d$, then

$$|(L_n - L)_{jk}| \leq Km^d\beta\left(\frac{m}{4}\right). \tag{3.8}$$

If $j_\ell \geq m$, $\ell = 1, \ldots, d$, then

$$|(L_n^{-1} - L^{-1})_{jk}| \leq Km^d\beta\left(\frac{m}{4}\right). \tag{3.9}$$

Proof. Let V_n be the restriction of L to I_{K_n}. Since L is lower triangular, L^{-1} restricted to I_{K_n} is V_n^{-1}, and so we obtain that

$$\|V_n\|_T \leq \|L\|_T, \qquad \|V_n^{-1}\|_T \leq \|L^{-1}\|_T.$$

Now,

$$\begin{aligned}\|V_n^{-1}A_n(V_n^{-1})^T - I_n\|_{H,\frac{1}{4}} &= \|V_n^{-1}(A_n - V_nV_n^T)(V_n^{-1})^T\|_{H,\frac{1}{4}}\\ &= \|V_n^{-1}(A_n - (LL^T)_{I_{K_n}})(V_n^{-1})^T\|_{H,\frac{1}{4}}\\ &\le C\ \|V_n^{-1}\|_T\ \|A_n - (LL^T)_{I_{K_n}}\|_{H,1}\ \|(V_n^{-1})^T\|_T\end{aligned}$$

for some $C > 0$, by Lemma 3.3, and so is uniformly bounded. Note that

$$V_n^{-1}A_n(V_n^{-1})^T = (V_n^{-1}L_n)(V_n^{-1}L_n)^T.$$

Also, from Lemma 3.1 we see that $\|(V_n^{-1}A_n(V_n^{-1})^T)^{-1}\|_2$ is uniformly bounded. Hence by Lemma 3.4, there are constants M, K so that for any $m \ge M$ and $m \le K_n \le \rho m$

$$\begin{aligned}&\sum_{j,k\in I_{K_n}} \left\{|(V_n^{-1}L_n)_{jk} - \delta_{jk}|^2 : j_\ell \ge m,\ \ell = 1,\dots,d \text{ or } k_\ell \ge m,\ \ell = 1,\dots,d\right\}\\ &\qquad\qquad \le K^2 m^{2d}\beta\left(\frac{m}{4}\right)^2.\end{aligned} \tag{3.10}$$

Now, for $j, k \in I_{K_n}$,

$$\begin{aligned}|(L_n - L)_{jk}| = |(L_n - V_n)_{jk}| &= \left|(V_n(V_n^{-1}L_n - I))_{jk}\right|\\ &= \left|\sum_{\ell\le j} L_{j\ell}((V_n^{-1}L_n)_{\ell k} - \delta_{\ell k})\right|\\ &\le \sum_{\ell\le j}\|L\|_T\beta(|j-\ell|)\ \left|(V_n^{-1}L_n)_{\ell k} - \delta_{\ell k}\right|.\end{aligned}$$

If $k_i \ge m$, $i = 1,\dots,d$, then by (3.10)

$$|(L_n - L)_{jk}| \le \|L\|_T\|\beta\|\, Km^d\beta\left(\frac{m}{4}\right).$$

Now, suppose we have that

$$j_i \ge \frac{5}{4}m, \quad i = 1,\dots,d.$$

If $\ell_i \ge m$, $i = 1,\dots,d$, then by (3.10),

$$\left|(V_n^{-1}L_n)_{\ell k} - \delta_{\ell k}\right| \le Km^d\beta\left(\frac{m}{4}\right),$$

otherwise $\ell_i < m$ for some $1 \le i \le d$ and so $|j - \ell| \ge |j_i - \ell_i| \ge \frac{m}{4}$. Thus we have confirmed that

$$|(L_n - L)_{jk}| \le \|L\|_T\|\beta\|Km^d\beta\left(\frac{m}{4}\right) + \sum_{l\le j}\|L\|_T\beta\left(\frac{m}{4}\right)\ \left\|V_n^{-1}L_n - I\right\|_2.$$

Since $\|A_n\|_2$ is uniformly bounded, it follows that $\|L_n\|_2$ is also uniformly bounded. Also, $\|V_n^{-1}\|_2$ is uniformly bounded by Lemma 3.1, and so $\|V_n^{-1}L_n - I\|_2$ is uniformly bounded. Thus for some $C > 0$, we obtain that

$$|(L_n - L)_{jk}| \le Cm^d\beta\left(\frac{m}{4}\right).$$

Now, take j, k in I_{K_n} with $j_i \geq m$, $i = 1, \ldots, d$. Then

$$\begin{aligned}\left|(L_n^{-1} - L^{-1})_{jk}\right| &= \left|((I - V_n^{-1}L_n)L_n^{-1})_{jk}\right| \\ &\leq \sum_{\ell \in I_{K_n}} \left|(I - V_n^{-1}L_n)_{j\ell}\right| \left|(L_n^{-1})_{\ell k}\right| \\ &\leq \left\{ \sum_{\ell \in I_{K_n}} \left|(I - {V_n}^{-1}L_n)_{j\ell}\right|^2 \right\}^{\frac{1}{2}} \left\|L_n^{-1}\right\|_2 \\ &\leq C_1 m^d \beta\left(\frac{m}{4}\right)\end{aligned}$$

for some $C_1 > 0$, by (3.10) and the fact that $\|A_n^{-1}\|_2$ is uniformly bounded. □

COROLLARY 3.6. *If in Theorem* 3.5 *we let* L *be in* $M_{\mathbb{Z}^d}$ *rather than in* $M_{\mathbb{Z}_+^d}$, *then* (3.8) *and* (3.9) *hold with* $\beta(\frac{m}{4})$ *replaced by* $\beta(\frac{m}{8})$.

Proof. In this case we use the inequality

$$\left\|A_n - (L_+L_+^T)_{I_{K_n}}\right\|_{H,\frac{1}{2}} \leq \left\|A_n - (LL^T)_{I_{K_n}}\right\|_{H,\frac{1}{2}} + \left\|(LL^T)_+ - L_+L_+^T\right\|_{H,\frac{1}{2}}.$$

Applying Lemma 3.2, noting that $\|\cdot\|_{H,\frac{1}{2}} \leq \|\cdot\|_{H,1}$, we see that $\|A_n - (L_+L_+^T)_{I_{K_n}}\|_{H,\frac{1}{2}}$ is uniformly bounded. We can now apply Theorem 3.5 with L replaced by L_+ and β replaced by $\beta(\frac{1}{2})$ to give the result. □

We now return to the situation of section 2.

COROLLARY 3.7. *Suppose that the matrix* G *in* (2.15) *is invertible in* $\ell_2(\mathbb{Z}_+^d)$ *and that the sequences* $\{\sigma_j : j \succeq 0\}$, $\{\gamma_j : j \succeq 0\}$ *in Theorem* 2.4 *satisfy* $|\sigma_j|, |\gamma_j| \leq \beta(c|j|)$, $j \succeq 0$, *for* $c > 0$. *We assume that* $t^d\beta(t) \to 0$ *as* $t \to \infty$ *and take* $\rho > 1$. *Then there is a constant* M *such that for all* $m \geq M$ *and* j, k *in* I_{K_n} *with* $K_n \leq \rho m$, $j_\ell \geq m$, $\ell = 1, \ldots, d$,

$$\left|(L^n)_{jk}^{-1} - L_{jk}^{-1}\right| \leq C_1 m^d \beta(Cm), \tag{3.11}$$

and

$$\sum_{k \in I_{kn}} \left|(L^n)_{jk}^{-1} - L_{jk}^{-1}\right| \leq C_2 m^{2d} \beta(Cm) \tag{3.12}$$

for constants $C, C_1, C_2 \geq 0$.

Proof. We apply Corollary 3.6 with $A_n = G^n$. By Lemma 2.9 and the assumption on G, $\|(G^n)^{-1}\|_2$ is uniformly bounded. Recalling Lemma 2.8 we see that all the conditions of Corollary 3.6 are satisfied with β replaced by $\beta(k)$ for some $k > 0$. Then (3.11) follows from Corollary 3.6 and (3.12) follows from (3.11) on recalling that $K_n \leq \rho m$. □

4. Gram–Schmidt asymptotics. We are now ready to use the material developed in the previous two sections to state conditions under which the Gram–Schmidt process described in section 2 converges when $n \to \infty$.

It is convenient to consider two cases. First take $p > d$ and suppose that for all $q < p$, there is a constant $c_1 > 0$ with $|\phi(x)| \leq c_1(1 + |x|)^{-q}$, $x \in \mathbb{R}^d$. Then, from Theorem 2.4 we see that for all $r < p - d$, there exists c_2 with $|\sigma_j|, |\gamma_j| \leq c_2(1 + |j|)^{-r}$, $j \succeq 0$. In Corollary 3.7 we choose $\beta(t) = (1 + t)^{-r}$, $t > 0$, for any $r < p - d$ with $r > d$. Then (3.12) gives, for any $s < p - 3d$, a constant c_3 so that

$$\sum_{k \in J_n} \left|(L^n)_{jk}^{-1} - L_{jk}^{-1}\right| \leq c_3(1 + n)^{-s}$$

for all j in J_n, $j_\ell \geq \eta n$, $\ell = 1, \ldots, d$. We have already seen in (2.13) that in this case, for any $r < p - d$ there is a constant c_4 with

$$\sum_{k \notin J_n} \left| L^{-1}_{jk} \right| \leq c_4 n^{-r}.$$

Thus we have established the following result.

THEOREM 4.1. *Suppose that $\phi : \mathbb{R}^d \to \mathbb{R}$ is a measurable function satisfying, for some $p > 3d$, that for all $q < p$ there is a constant c_1 with*

$$|\phi(x)| \leq c_1(1 + |x|)^{-q}, \quad x \in \mathbb{R}^d.$$

Suppose that G in (2.15) is invertible on $\ell_2(\mathbb{Z}^d_+)$. Take $s < p - 3d$ and any η in $(0, \frac{1}{d})$. Then there exist N and $C > 0$ such that for all $n \geq N$ and x in $\Omega_n := \{x \in \mathbb{R}^d : x_1, \ldots, x_d \geq 0, |x| \leq n\}$, and any j in $\mathbb{Z}^d$ with $|j| \leq (1 - \eta)n$ and $j_\ell \geq \eta n$, $\ell = 1, \ldots, d$, we have the estimate

$$\left| \psi^n_j(x) - \psi_j(x) \right| \leq C n^{-s}.$$

The next case we consider is a function ϕ which decays exponentially. We have seen in this case in Theorem 2.4 that the sequences $\{\sigma_j :\succeq 0\}$, $\{\gamma_j :\succeq 0\}$ also decay exponentially. Recalling (2.14), we can apply Corollary 3.7 with $\beta(t) = c\mu^t$, $t > 0$, for some $c > 0$ and μ in $(0, 1)$ to give the following result.

THEOREM 4.2. *Suppose that $\phi : \mathbb{R}^d \to \mathbb{R}$ is a measurable function satisfying for some $c > 0$ and λ in $(0, 1)$*

$$|\phi(x)| \leq c\lambda^{|x|}, \quad x \in \mathbb{R}^d.$$

Suppose that G in (2.15) is invertible on $\ell_2(\mathbb{Z}^d_+)$ and $\eta \in (0, \frac{1}{d})$. Then there exist $N, C > 0$ and μ in $(0, 1)$ so that for all $n \geq N$, x in Ω_n, and any j in $\mathbb{Z}^d$ with $|j| \leq (1 - \eta)n$ and $j_\ell \geq \eta n$, $\ell = 1, \ldots, d$, we have the estimate

$$|\psi^n_j(x) - \psi_j(x)| \leq C\mu^n.$$

5. The special case of box splines. For the sake of illustration, in this section we identify the limiting profile in a simple case. Let $\phi(x, y)$, $(x, y) \in \mathbb{R}^2$, the linear bivariate box spline (Figure 1), that is,

$$\phi(x, y) = \begin{cases} 1 - y & x \in [0, 1),\ y \in [x, 1), \\ 1 - x & x \in [0, 1),\ y \in [0, x), \\ 1 - x + y & x \in [0, 1),\ y \in (-1 + x, 0), \\ \phi(-x, -y) & x \in (-1, 0),\ y \in (-1, 1 + x), \\ 0 & \text{otherwise}, \end{cases}$$

and, for $j = (j_1, j_2) \in \mathbb{Z}^2$, we consider the integer translates $\phi_j(x, y) := \phi(x - j_1, y - j_2)$.

Let $T = \{t_{i-j}\}$ be the Gram matrix (2.3) generated by the functions ϕ_j, $j \in \mathbb{Z}^2$. It is straightforward to confirm that

$$t_j = \begin{cases} \frac{1}{2} & \text{for } j = (0, 0), \\ \frac{1}{12} & \text{for } j = (\pm 1, 0),\ (0, \pm 1),\ (1, 1),\ \text{and } (-1, -1), \\ 0 & \text{otherwise}. \end{cases}$$

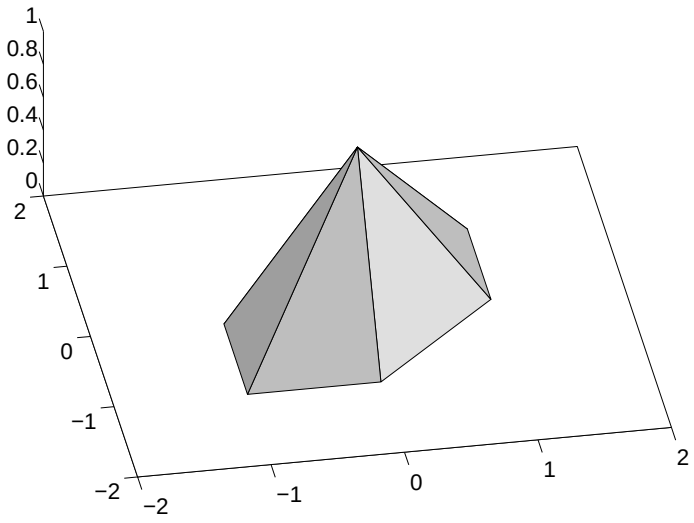


FIG. 1.

Hence the corresponding symbol is

$$t_\phi(x,y) = \sum_{j\in\mathbb{Z}^2} t_j e^{i(j_1 x + j_2 y)} = \frac{1}{6}(3 + \cos x + \cos y + \cos(x+y)), \quad x, y \in \mathbb{R},$$

which is strictly positive on $\mathbb{R}^2$, since $\min_{x,y\in\mathbb{R}} t_\phi(x,y) = \frac{1}{4}$. Furthermore, as ϕ has a compact support, there exists an exponentially decaying function β which satisfies (2.1).

As a result of Theorem 2.4, t_ϕ has a spectral factorization

$$t_\phi(x,y) = h(-x,-y)h(x,y), \quad x, y \in \mathbb{R},$$

which depends on the ordering $\preceq$ fixed on $\mathbb{Z}^2$. Moreover, the coefficients $\{\sigma_j : j \succeq 0\}$ and $\{\gamma_j : j \succeq 0\}$ of the factor $h(x,y)$ and of its reciprocal $1/h(x,y)$ decay exponentially with respect to $|j|$, that is, there are constants $k > 0$ and μ in $(0,1)$ with $|\sigma_j|, |\gamma_j| \le k\mu^{|j|}$, $j \succeq 0$.

These coefficients can be obtained by following the procedure adopted in Theorem 2.4 to prove the existence of the spectral factorization of t_ϕ with respect to the chosen ordering. The computational procedure, which is a generalization of one of the algorithms studied in [12], consists in evaluating the coefficients of the Fourier series of $f := \log t_\phi$. Then, taking into account a fixed ordering on $\mathbb{Z}^2$, the Fourier coefficients $\{\sigma_j : j \succeq 0\}$ of $h := \exp f_+$ are obtained. In this algorithm the bidimensional fast Fourier transform (FFT) and its inverse are used repeatedly. As $1/h := \exp(-f_+)$, the same procedure can be immediately adapted to compute the Fourier coefficients $\{\gamma_j : j \succeq 0\}$ of $1/h$ with little more computational effort.

Note that it is not easy in general to prove the positivity of t_ϕ for an arbitrary function ϕ. Once this assumption is proved, it remains to overcome only a numerical difficulty, consisting in the computation of the coefficients γ_j we have to consider. If

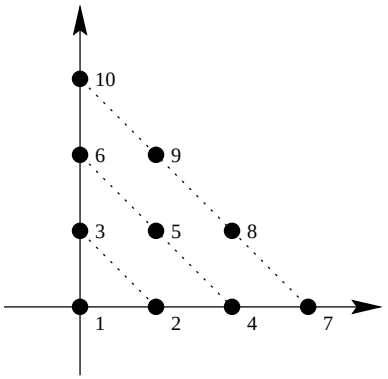


FIG. 2.

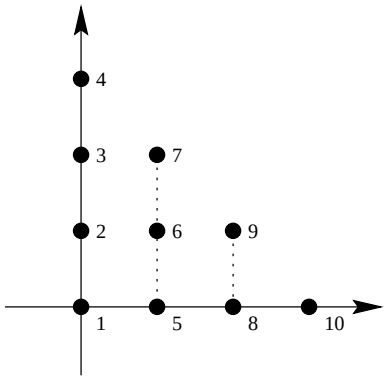


FIG. 3.

ϕ decays exponentially, which is of course the case if it is compactly supported, it is sufficient to compute a relatively small number of coefficients. Otherwise, this number can be large, as it happens in the case of algebraic decaying of ϕ.

In order to highlight how the spectral factorization depends on the ordering $\preceq$ chosen on $\mathbb{Z}^2$, let us consider the following two cases:

(a) the lexicographical ordering with respect to $j_1 + j_2$, j_2, i.e., $\ell \preceq m$ if either $\ell_1 + \ell_2 < m_1 + m_2$ or $\ell_1 + \ell_2 = m_1 + m_2$ and $\ell_2 \le m_2$;

(b) the usual lexicographical ordering on $\mathbb{Z}^2$, so that $\ell \preceq m$ if either $\ell_1 < m_1$ or $\ell_1 = m_1$ and $\ell_2 \le m_2$.

For $n \ge 1$, let $J_n = \{j \in \mathbb{Z}^2_+ : j_1 + j_2 \le n - 1\}$. The orderings (a) and (b) restricted to J_4 are illustrated in Figure 2 and Figure 3, respectively.

Now, let $\Omega_n = \{x \in \mathbb{R}^2_+ : x_1 + x_2 \le n\}$ and let ϕ^n_j be the restriction to Ω_n of the box spline ϕ_j. We then construct a sequence of orthonormal splines ψ^n_j, $j \in J_n$, on Ω_n by the Gram–Schmidt process applied to ϕ^n_j, $j \in J_n$, with respect to the prefixed ordering. Thus, for $j \in J_n$,

$$\psi^n_j = \sum_{k \in J_n} (L^n)^{-1}_{jk} \phi^n_k, \tag{5.1}$$

where L^n is the Cholesky factor of the Gram matrix G^n associated with the box splines ϕ^n_k as in (2.11).

As the functions ϕ^n_k are locally supported, G^n is a sparse matrix. An indication of its sparsity is given by Figure 4 and Figure 5, where the dots denote the position of the nonzero entries of the matrix G^{10} for the two orderings described above. Recalling that the box splines ϕ_j, $j \in \mathbb{Z}^2$, are unconditionally stable [7], we conclude that the corresponding Gram matrix G in (2.15) is positive definite.

Let

$$\psi(x, y) = \sum_{j \succeq 0} \gamma_j \phi_{-j}(x, y), \quad x, y \in \mathbb{R},$$

where $\{\gamma_j : j \succeq 0\}$ are the coefficients of the Fourier series of $1/h(x, y)$.

As the coefficients $\{\gamma_j : j \succeq 0\}$ decay exponentially with respect to $|j|$, for all practical applications ψ can be considered a locally supported function. Figure 6 and Figure 7 show the entries of the matrix $\Gamma = \{\gamma_{i-j} : i, j \in \mathbb{Z}^2\}$ which have modulus greater than 10^{-16} (small dots) and 10^{-8} (big dots) for the two orderings considered.

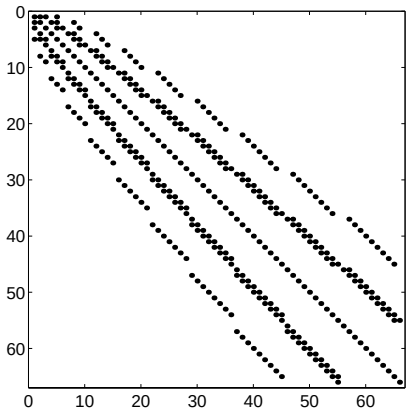


FIG. 4.

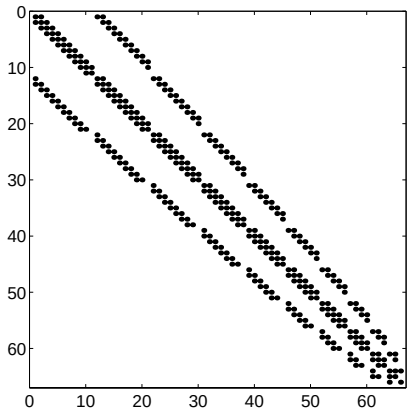


FIG. 5.

In both cases bidimensional FFTs were computed using 64 points on each axis, that is, based upon 4096 coefficients. For the first ordering (second ordering), 817 (785) have modulus greater than 10^{-16} and 205 (204) greater than 10^{-8}. Moreover, for both orderings

$$\gamma_j < 10^{-16} \quad \text{for} \quad |j| > 41 \quad \text{and} \quad \gamma_j < 10^{-8} \quad \text{for} \quad |j| > 22.$$

A graph of the limiting profile $\psi(x, y)$ related to the first ordering is depicted in Figure 8. This graph is hardly distinguishable from the one obtained by the second ordering.

For ordering (a), Theorem 4.2 states that for fixed η in $(0, \frac{1}{2})$ and $|j| \leq (1 - \eta)n$, $j_1, j_2 > \eta n$, (x, y) in Ω_n, $|\psi_j^n(x, y) - \psi_j(x, y)|$ decays exponentially as $n \to \infty$. Numerically, we consider the quantity

$$\epsilon_n = \max_{(x,y)\in\Omega_n} \left|\psi_{j_n}^n(x, y) - \psi_{j_n}(x, y)\right|,$$

which we approximate on a finite grid. Here j_n is chosen to make ϵ_n close to minimum. According to our experience, a good value of j_n is $\left(\lceil \frac{n}{2} \rceil - 1, \ \lceil \frac{n}{2} \rceil - 1\right)$.

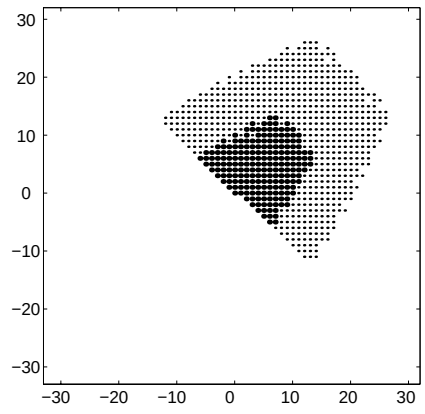


FIG. 6.

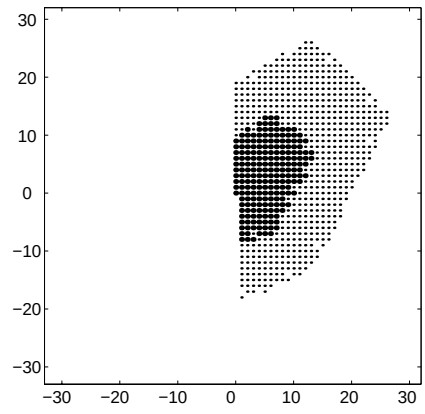


FIG. 7.

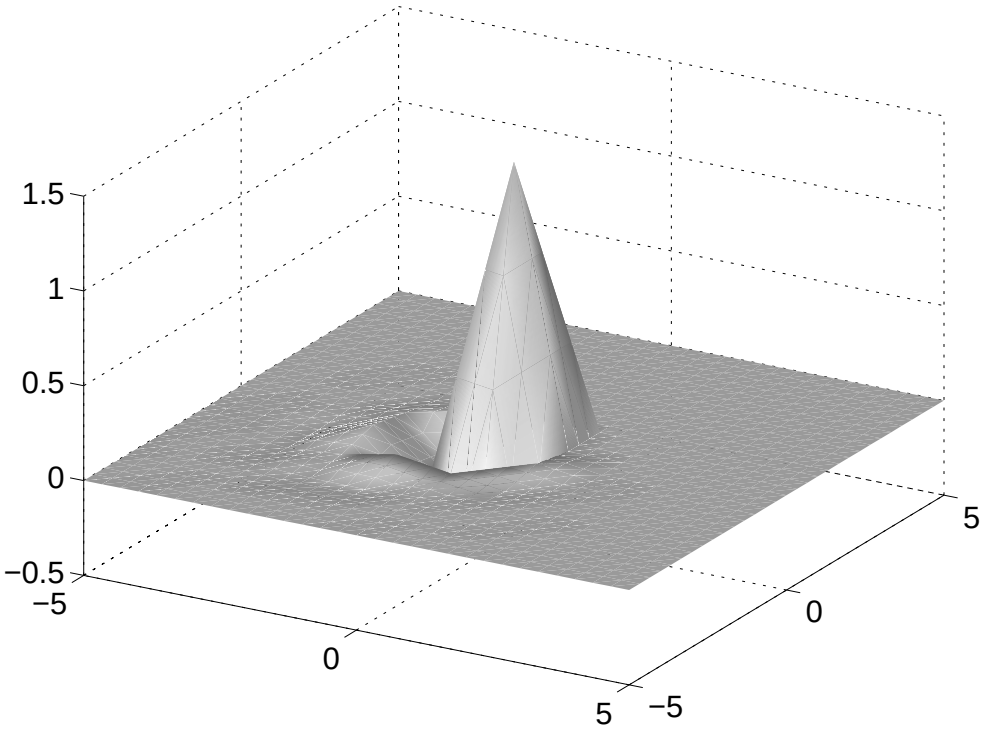


FIG. 8.

For ordering (b) we have no theoretical convergence result, but we shall also consider ϵ_n as defined above. From experience, ϵ_n is close to minimum when $j_n = \left(\lceil \frac{n}{3} \rceil, \lceil \frac{n}{3} \rceil\right)$.

As $n \to \infty$, ϵ_n gives a measure of the distance of $\psi_{j_n}^n$ from the appropriate translate of the limiting profile ψ. We found out that, in both cases, ϵ_n decays

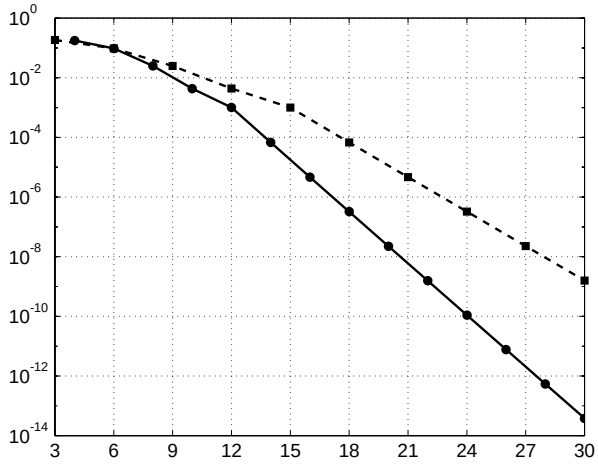


FIG. 9.

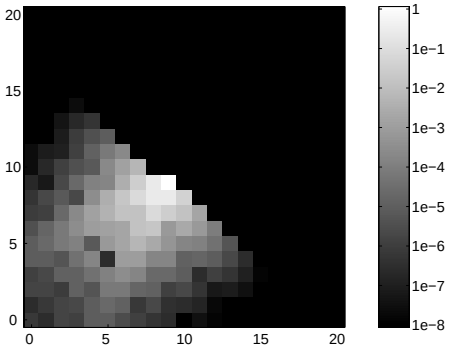


FIG. 10.

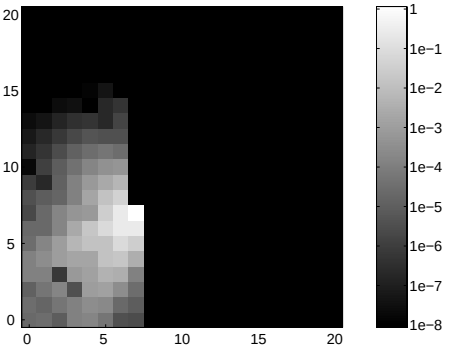


FIG. 11.

exponentially, but the two decay rates are quite different. Figure 9, where dots and squares indicate the values of ϵ_n with respect to the first and the second ordering, highlights this property.

This result is due to the fact that the sequence of domains Ω_n that we use in the orthonormalization process favors the first ordering. In order to justify this statement, we consider $n = 20$ and, for each of the two orderings, we take j_n as above specified, that is $j_{20} = (9, 9)$ for the first ordering and (7,7) for the second. Then we consider the coefficients $(L^n)^{-1}_{j_n k}$ of $\psi^n_{j_n}$ and we depict the magnitude of their modulus, in grayscale, in Figure 10 and Figure 11, respectively. Note the different level of gray on the edges of Ω_n in the two figures. The lighter shade of Figure 11 indicates that in the second ordering we ignore substantial coefficients and, as a result, for moderately high values of n the approximation of ψ_{j_n} by $\psi^n_{j_n}$ is worse in this case than in the other one.

REFERENCES

[1] F. Bauer, *Ein direktes Iterations Verfahren zur Hurwitz-zerlegung eines Polynoms*, Arch. Elektr. Uebertragung, 9 (1955), pp. 285–290.

[2] F. Bauer, *Beiträge zur Entwicklung numerischer Verfahren für programmgesteuerte Rechenanlagen, ii. Direkte Faktorisierung eines Polynoms*, Sitz. Ber. Bayer. Akad. Wiss., (1956), pp. 163–203.

[3] B. Bogert, M. Healy, and J. Tukey, *The quefrency alanysis of time series for echoes: Cepstrum pseudo-autocovariance, cross-cepstrum and saphe cracking*, in Proceedings of the Symposium for Time Series Analysis, M. Rosenblatt, ed., New York, John Wiley and Sons, 1963, pp. 209–243.

[4] C. D. Boor, *A bound on the L_∞-norm of L_2-approximation by splines in terms of a global mesh ratio*, Math. Comp., 30 (1976), pp. 765–771.

[5] A. Calderón, F. Spitzer, and H. Widom, *Inversion of Toeplitz matrices*, Illinois J. Math., 3 (1959), pp. 490–498.

[6] C. Chui, P. Smith, and J. Ward, *Cholesky factorization of positive definite bi-infinite matrices*, Numer. Funct. Anal. Optim., 5 (1982), pp. 1–20.

[7] W. Dahmen and C. Micchelli, *Translates of multivariate splines*, Linear Algebra Appl., 52 (1983), pp. 217–234.

[8] S. Demko, *Inverses of band matrices and local convergence of spline projections*, SIAM J. Numer. Anal., 14 (1977), pp. 616–619.

[9] S. Demko, W. Moss, and P. Smith, *Decay rates for inverses of band matrices*, Math. Comp., 43 (1984), pp. 491–499.

[10] J. Douglas Jr., T. Dupont, and L. Wahlbin, *Optimal L_∞ error estimates for Galerkin approximations to solutions of two-point boundary value problems*, Math. Comp., 29 (1975), pp. 475–483.

[11] T. Goodman, C. Micchelli, G. Rodriguez, and S. Seatzu, *On the Cholesky factorization of the Gram matrix of locally supported functions*, BIT, 35 (1995), pp. 233–257.
[12] T. Goodman, C. Micchelli, G. Rodriguez, and S. Seatzu, *Spectral factorization of Laurent polynomials*, Adv. Comput. Math., 7 (1997), pp. 429–454.
[13] T. Goodman, C. Micchelli, G. Rodriguez, and S. Seatzu, *On the limiting profile arising from orthonormalizing shifts of exponentially decaying functions*, IMA J. Numer. Anal., 18 (1998), pp. 331–354.
[14] A. Innocenti, G. Rodriguez, and S. Seatzu, *Orthogonal Splines with Applications to Multivariate Least Squares and Integral Equations of the First Kind*, Tech. Report CRS4-APPMATH-93-15, CRS4, Cagliari, Italy, 1993.
[15] M. Krein, *Integral equations on the half-line with kernel depending on the difference of the arguments*, Uspehi Mat. Nauk, 13 (1958), pp. 3–120. (in Russian); AMS Translations, 22 (1962), pp. 163–288 (in English).
[16] J. Mason, G. Rodriguez, and S. Seatzu, *Orthogonal splines based on B-splines—with applications to least squares, smoothing and regularization problems*, Numer. Algorithms, 5 (1993), pp. 25–40.
[17] J. McClellan, *Multidimensional spectral estimation*, Proc. IEEE, 70 (1982), pp. 1029–1039.
[18] C. V. D. Mee, G. Rodriguez, and S. Seatzu, *Block Cholesky factorization of infinite matrices and orthonormalization of vectors of functions*, in Advances in Computational Mathematics, Z. Chen, Y. Li, C. Micchelli, and Y. Xu, eds., Lecture Notes in Pure and Appl. Math. 202, M. Dekker, New York, Basel, 1998, pp. 423–455.
[19] C. Micchelli and T. Sauer, *Regularity of multiwavelets*, Adv. Comput. Math., 7 (1997), pp. 455–545.
[20] D. Newman, *A simple proof of Wiener's* $1/f$ *theorem*, Proc. Amer. Math. Soc., 48 (1975), pp. 264–265.
[21] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Prentice Hall Signal Processing Series, Prentice Hall, Englewood Cliffs, NJ, 1989.
[22] I. Schoenberg, *Cardinal Spline Interpolation*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 12, SIAM, Philadelphia, PA, 1973.
[23] J.-G. Sun, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.
[24] H. Widom, *Inversion of Toeplitz matrices*. II, Illinois J. Math., 4 (1960), pp. 88–99.
[25] H. Widom, *Asymptotic behavior of block Toeplitz matrices and determimants*. II, Adv. Math., 21 (1976), pp. 1–29.
[26] D. Youla and N. Kazanjian, *Bauer-type factorization of positive matrices and the theory of matrix polynomials orthogonal on the unit circle*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 25 (1978), pp. 57–69.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 527–532

PRINCIPAL PIVOTING METHOD FOR SOLVING COLUMN SUFFICIENT COMPLEMENTARITY PROBLEMS*

A. L. N. MURTHY† AND G. S. R. MURTHY†

Abstract. The linear complementarity problem (q, A) is to find, for a given real square matrix A of order n and a real column vector q of order n, a nonnegative vector z such that $Az + q \geq 0$ and $z^t(Az + q) = 0$. It is known that when A is a positive semidefinite matrix, one can use a principal pivoting method to compute a solution to (q, A) if it has one and to conclude that the problem has no solution otherwise. Cottle, Pang, and Venkateswaran [*Linear Algebra Appl.*, 114/115 (1989), pp. 231–249] introduced the class of sufficient matrices and widened the scope of a principal pivoting algorithm to solve linear complementarity problems with row sufficient matrices. Our main result in this article is to show that this algorithm can be extended to solve even the problems with column sufficient matrices.

Key words. linear complementarity problem, column sufficiency, principal pivoting

AMS subject classification. 90C33

PII. S0895479899363526

1. Introduction.

The linear complementarity problem (LCP) with data $A \in \boldsymbol{R}^{n\times n}$ and $q \in \boldsymbol{R}^n$, denoted by (q, A), is to find a vector z such that

$$Az + q \geq 0, \quad z \geq 0, \quad \text{and } z^t(Az + q) = 0.$$

LCPs have a wide range of applications both in theory and practice (see [2, 9]). Though algorithms are available to solve special classes of LCPs, computing the solution to a general LCP (q, A) or exhibiting that the problem has no solution has remained as an open question. Lemke and Howson [6] produced an algorithm that can process many classes of LCPs often encountered in practice. An algorithm is said to process a problem if it either finds a solution to the problem or concludes that the problem has no solution. There are other algorithms to solve the special cases of LCPs, such as Cottle and Dantzig's principal pivoting algorithms, Murty's least index method, and so on (see [2] and [9] for details). These algorithms are different from Lemke's algorithm and are based on principal pivoting.

In this article our main concern is to establish that a principal pivoting algorithm can be used to solve LCPs (q, A) in which A is a column sufficient matrix (see below for definitions of matrix classes). Using a result of Murthy and Parthasarathy [7], we will show that if A is a column sufficient $\boldsymbol{Q_0}$-matrix, then (q, A) can be processed using the algorithm in question. This result is useful in deriving a number of results.

The question of whether an algorithm processes a given LCP (q, A) depends upon the properties of the matrix A, and this has led to the evolution of a variety of matrix classes in the theory of LCP. We shall define some of the matrix classes which will be relevant to this article.

In section 2 we shall present the necessary background and the key result from [7] along with its applications. In section 3, we shall present our main results.

*Received by the editors November 5, 1999; accepted for publication (in revised form) by R. Brualdi March 9, 2000; published electronically August 9, 2000.
http://www.siam.org/journals/simax/22-2/36352.html
†Indian Statistical Institute, Street No. 8, Habsiguda, Hyderabad, India (simhaaln@hotmail.com, murthygsr@hotmail.com).

2. Key result and its applications. For $A \in \boldsymbol{R}^{n\times n}$ and $q \in \boldsymbol{R}^n$, define the sets $F(q, A)$ and $S(q, A)$ as

$$F(q, A) \;=\; \{\, z \in \boldsymbol{R}^n_+ \;:\; Az + q \geq 0 \;\}$$

and

$$S(q, A) \;=\; \{\, z \in F(q, A) \;:\; z^t(Az + q\,) \;=\; 0\,\}.$$

Note that $S(q, A)$ is the set of solutions to LCP (q, A). The matrix A is said to be (i) a $\boldsymbol{Q_0}$-matrix if $S(q, A) \neq \emptyset$, whenever $F(q, A) \neq \emptyset$; (ii) a $\boldsymbol{Q}$-matrix if $S(q, A) \neq \emptyset$ for all q; (iii) a positive semidefinite if $x^tAx \geq 0$ for all x; (iv) a copositive matrix $x^tAx \geq 0$ for all nonnegative x; (v) an $\boldsymbol{E_0}$-matrix if for every nonnegative vector $x \neq 0$ there exists an index i such that $x_i > 0$ and $(Ax)_i \geq 0$; (vi) a $\boldsymbol{P_0}$-matrix if every principal minor of A is nonnegative.

If $A_{\alpha\alpha}$, the principal submatrix of A with respect to the index set α, is nonsingular, then the matrix M defined by

$$M_{\alpha\alpha} = (A_{\alpha\alpha})^{-1},\; M_{\alpha\bar{\alpha}} = -M_{\alpha\alpha}A_{\alpha\bar{\alpha}},\; M_{\bar{\alpha}\alpha} = A_{\bar{\alpha}\alpha}M_{\alpha\alpha},\; \text{and}\; M_{\bar{\alpha}\bar{\alpha}} = A_{\bar{\alpha}\bar{\alpha}} - M_{\bar{\alpha}\alpha}A_{\alpha\bar{\alpha}}$$

is called the principal pivotal transform (PPT) of A with respect to α. For any $q \in \boldsymbol{R}^n$, the vector p defined by

$$p_\alpha = -(A_{\alpha\alpha})^{-1}q_\alpha,\; p_{\bar{\alpha}} = q_{\bar{\alpha}} - A_{\bar{\alpha}\alpha}(A_{\alpha\alpha})^{-1}q_\alpha$$

is called PPT of q with respect to A and α.

The new LCP (p, M) is called the PPT of (q, A) with respect to α. There is a one-to-one correspondence between the solution set of (q, A) and that of (p, M). The reader may refer to [2] for details of notation and preliminary results. Consider (q, A) and let (p, M) be its PPT with respect $\alpha \neq \emptyset$.

DEFINITION 1. *We say that (p, M) is obtained by a single (resp., double) pivot if $|\alpha|$, the size of α, is equal to* 1 *(resp.,* 2*).*

Many algorithms have been proposed to solve LCPs based on principal pivoting methods. See [2, 9] for detailed discussions on this topic. But these algorithms have the limitation that they can process only certain classes of LCPs. Given below is one such algorithm which uses only single or double pivots. Graves [5] proposed a lexicographic procedure and showed that LCPs involving positive semidefinite matrices can be processed by this algorithm (see [9] for a complete description and proof).

ALGORITHM 2.

Step 0. Input $M = A$ and $p = q$.

Step 1. If $p \geq 0$, then $z = 0$ is a solution to (p, M); obtain a solution of (q, A) using this and stop.

Step 2. If there exists an index i such that $p_i < 0$ and ith row of M is nonpositive, then conclude that (q, A) has no solution and stop.

Step 3. Choose i with $p_i < 0$ using the *lexicographic* rule. If $m_{ii} > 0$, then replace (p, M) by its PPT with respect to $\alpha = \{i\}$. If $m_{ii} = 0$, then choose j from $\{k : m_{ki} < 0\}$ using the lexicographic rule and replace (p, M) by its PPT with respect to $\alpha = \{i,\; j\}$. Go to Step 1.

Cottle, Pang, and Venkateswaran [3] introduced the class of sufficient matrices (see below for definitions) and expanded the scope of Algorithm 2 to LCPs involving row sufficient matrices.

DEFINITION 3. *A matrix $A \in \boldsymbol{R}^{n\times n}$ is said to be a column sufficient matrix ($\boldsymbol{C_s}$-matrix) if for every $x \in \boldsymbol{R}^n$, $[x_i(Ax)_i \leq 0$ for all $i]$ implies $[x_i(Ax)_i = 0$ for all $i]$. The matrix A is said to be a row sufficient matrix ($\boldsymbol{C_r}$-matrix) if A^t is a $\boldsymbol{C_s}$-matrix, and A is said to be sufficient if it is both row and column sufficient.*

Remark 4. An equivalent characterization of $\boldsymbol{C_s}$-matrices is that A is a $\boldsymbol{C_s}$-matrix if and only if $S(q, A)$ is convex for all q (see Theorem 3.5.8 of [2]).

Murthy and Parthasarathy [8] introduced the class of fully copositive matrices, denoted as $\boldsymbol{C_0^f}$-matrices (a matrix is said to be fully copositive if all its PPTs are copositive), and showed that LCP (q, A) can be processed by Algorithm 2 when A is a $\boldsymbol{C_0^f} \cap \boldsymbol{Q_0}$-matrix . In this article we will show that this algorithm works even in the case of $\boldsymbol{C_s} \cap \boldsymbol{Q_0}$-matrices. A key result in establishing this result is the following theorem.

THEOREM 5 (see Murthy and Parthasarathy [7]). *Suppose $A \in \boldsymbol{R}^{n\times n} \cap \boldsymbol{E_0} \cap \boldsymbol{Q_0}$. Assume that for some $i,\ j,\ a_{ii} = 0$ and $a_{ij} > 0$. Then there exists a k such that $a_{ki} < 0$.*

The above theorem is an extension of Pang's result for $\boldsymbol{E_0}\cap\boldsymbol{Q}$-matrices to $\boldsymbol{E_0}\cap\boldsymbol{Q_0}$-matrices. Pang's result states that if A is an $\boldsymbol{E_0}\cap\boldsymbol{Q}$-matrix, then any nonzero solution of $(0, A)$ must have at least two nonzero coordinates (see [10]). Theorem 5 has very interesting applications. It has been used to prove that Stone's conjecture is true for matrices of order up to $n \leq 5$. Stone's conjecture states that every $\boldsymbol{E_0^f} \cap \boldsymbol{Q_0}$-matrix is a $\boldsymbol{P_0}$-matrix (A is said to be an $\boldsymbol{E_0^f}$-matrix if every PPT of A is an $\boldsymbol{E_0}$-matrix). See [4, 11, 7, 1] for details. Theorem 5 is also used in proving (i) $\boldsymbol{C_0^f}\cap\boldsymbol{Q_0}$-matrices are $\boldsymbol{P_0}$-matrices, (ii) $\boldsymbol{C_0^f}\cap\boldsymbol{Q_0}$-matrices are completely $\boldsymbol{Q_0}$ (i.e., all the principal submatrices are $\boldsymbol{Q_0}$), all their PPTs are completely $\boldsymbol{Q_0}$-matrices, and (iii) (q, A) can be processed by Algorithm 2 when A is a $\boldsymbol{C_0^f} \cap \boldsymbol{Q_0}$-matrix.

3. Main results. We shall present our two main results in this section. The first result states that if A is in $\boldsymbol{C_s}\cap\boldsymbol{Q_0}$, then for any q, (q, A) is processible by Algorithm 2. In our second result we show that (q, A) can be processed by Algorithm 2 even without the assumption of $\boldsymbol{Q_0}$ on A. The main requirement for (q, A) to be processible by Algorithm 2 is contained in the following result.

THEOREM 6. *Let $A \in \boldsymbol{R}^{n\times n}$ be such that for any PPT M of A the following conditions hold:*

(i) $m_{ii} \geq 0$ *for all* i,

(ii) *for any* $i,\ j$, *if* $m_{ii} = 0$ *and* $m_{ij} > 0$, *then there exists a* k *such that* $m_{ki} < 0$, *and*

(iii) *for any* i, j *satisfying* $m_{ii} = 0,\ m_{ji} < 0$, *then* $m_{ij} > 0$.

Then A is a $\boldsymbol{Q_0}$-matrix.

Refer to [9] for a proof of the above theorem. Though the proof in [9] is presented for positive semidefinite matrices, what is required essentially is that the matrix A should satisfy the assumptions of Theorem 6. Positive semidefinite matrices satisfy these conditions because if A is a positive semidefinite matrix, then for any i with $a_{ii} = 0$, we have $a_{ij} + a_{ji} = 0$ for all j. These assumptions are satisfied by a larger class of matrices introduced by Cottle, Pang, and Venkateswaran [3] (see also section 3.5 of [2]). Murthy and Parthasarathy [8] showed that the assumptions are satisfied by $\boldsymbol{C_0^f}\cap\boldsymbol{Q_0}$-matrices. Our next result states that any matrix satisfying the conditions of Theorem 6 is a $\boldsymbol{Q_0}$-matrix.

THEOREM 7. *Suppose $A \in \boldsymbol{R}^{n\times n}$ satisfies the conditions of Theorem* 6. *Then A is a $\boldsymbol{Q_0}$-matrix.*

Proof. Suppose Algorithm 2 is applied to (q, A). Since no cycling occurs, from the assumptions of the theorem the algorithm terminates either in Step 1 or in Step 2 in a finite number of iterations. If the termination is in Step 1, then (q, A) has a solution, and if the algorithm terminates in Step 2, then $F(q, A) = \emptyset$. Therefore, it follows that A is a $\boldsymbol{Q_0}$-matrix. □

The conditions of Theorem 6 are sufficient only for a matrix to be a $\boldsymbol{Q_0}$-matrix. To see that these conditions are not necessary, one can give the trivial example, namely, that any negative (entrywise) matrix is a $\boldsymbol{Q_0}$-matrix. In our next theorem we show that any $\boldsymbol{C_s} \cap \boldsymbol{Q_0}$-matrix satisfies the conditions of Theorem 6.

THEOREM 8. *Suppose $A \in \boldsymbol{R}^{n\times n}$ is a $\boldsymbol{C_s} \cap \boldsymbol{Q_0}$-matrix. Then for any $q \in \boldsymbol{R}^n$, Algorithm 2 processes (q, A).*

Proof. It suffices to show that A satisfies the conditions of Theorem 6. If A is in $\boldsymbol{C_s} \cap \boldsymbol{Q_0}$, then so are all PPTs of A [2]. Let M be any PPT of A. Since every column sufficient matrix is a $\boldsymbol{P_0}$-matrix, $m_{ii} \geq 0$ for all i. The fact that M satisfies (ii) follows from the fact that every $\boldsymbol{P_0}$-matrix is an $\boldsymbol{E_0}$-matrix and from Theorem 5.

Next, suppose $m_{ii} = 0$ and $m_{ji} < 0$ for some i and j. Since $M \in \boldsymbol{P_0}$, $m_{ij} \geq 0$. If $m_{ij} = 0$, then M cannot be a column sufficient matrix. Therefore, M satisfies condition (iii) also. Thus, A satisfies the conditions of Theorem 6. □

It is known that $\boldsymbol{C_s}$ is not a subset of $\boldsymbol{Q_0}$. One might ask the question whether the $\boldsymbol{Q_0}$ condition can be dropped from Theorem 8. The answer is no. The matrix below serves as a counterexample.

Example 9. Consider the matrix

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

It is easy to check that A is copositive and a $\boldsymbol{C_s}$-matrix. Since every copositive matrix is an $\boldsymbol{E_0}$-matrix, it follows from Theorem 5 that A is not a $\boldsymbol{Q_0}$-matrix. Taking $q = (-1, 2, 3)^t$, if we try to use Algorithm 2 to solve (q, A), we get stuck in the very first iteration itself. However, this problem can be processed by applying Algorithm 2 to an augmented problem.

THEOREM 10. *Suppose $A \in \boldsymbol{R}^{n\times n}$. Consider the augmented matrix*

$$M = \begin{bmatrix} A & I \\ -I & 0 \end{bmatrix}, \tag{1}$$

where I is the identity matrix of order n. Then M is a $\boldsymbol{C_s}$-matrix if and only if A is a $\boldsymbol{C_s}$-matrix.

Proof. Since every principal submatrix of a $\boldsymbol{C_s}$-matrix is also in $\boldsymbol{C_s}$, we need only prove the "if" part. Let $x, y \in \boldsymbol{R}^n$ and $z^t = (x^t, y^t)$. Suppose $z_i(Mz)_i \leq 0$ for all i. Then we have $x_i(Ax)_i + x_i y_i \leq 0$ for all i and $-x_i y_i \leq 0$ for all i. From these two inequalities it follows that $x_i(Ax)_i \leq -x_i y_i \leq 0$ for all i. Since $A \in \boldsymbol{C_s}$, $x_i(Ax)_i = 0$ for all i. It follows from the first inequality that $x_i y_i \leq 0$, and from the second inequality that $x_i y_i = 0$ for all i. It follows that M is a $\boldsymbol{C_s}$-matrix. □

When A is a $\boldsymbol{C_s}$-matrix, the augmented matrix defined in (1) is used to solve (q, A) using iterative (convergence) procedures (see Theorem 5.9.7 and the discussion following it in [2]). This procedure is applied to the augmented problem $(\bar{q}, M)$, where $\bar{q}^t = (q^t, p^t)$ with p being a very large positive vector. The following result is used in recovering a solution of (q, A) from the solution of $(\bar{q}, M)$.

THEOREM 11. *Suppose $A \in \boldsymbol{R}^{n\times n}$ and let M be the augmented matrix defined in* (1). *Let $p,\ q \in \boldsymbol{R}^n$ and let $\bar{q}^t = (q^t, p^t)$. Assume that $S(q, A) \neq \emptyset$. Then $(x^t, y^t)^t$, $x, y \in \boldsymbol{R}^n$, is a solution to $(\bar{q}, M)$ for all large positive vectors p if and only if $x \in S(q, A)$ and $y = 0$.*

For a proof of this theorem, see Theorem 3.7.17 of [2]. We are now ready to establish our main result.

THEOREM 12. *Suppose $A \in \boldsymbol{R}^{n\times n}$ is a $\boldsymbol{C_s}$-matrix. Then the augmented matrix M defined in* (1) *is a column sufficient $\boldsymbol{Q_0}$-matrix.*

Proof. Column sufficiency of M is already established in Theorem 10. To show that M is a $\boldsymbol{Q_0}$-matrix, we shall show that M satisfies the conditions of Theorem 6. The matrix M satisfies conditions (i) and (iii) because it is a $\boldsymbol{C_s}$-matrix. So it suffices to show that it satisfies condition (ii). Clearly M satisfies condition (ii). It remains to show that every PPT of M satisfies this condition. Partition the matrix A as

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix},$$

where B and E are square matrices (the case where B is equal to A is also included). We need to consider two types of PPTs—one with respect to $\left[\begin{smallmatrix} B & I \\ -I & 0 \end{smallmatrix}\right]$ and the other with respect to B. First let us consider the case of PPT with respect to $\left[\begin{smallmatrix} B & I \\ -I & 0 \end{smallmatrix}\right]$. This is given by

$$G = \begin{bmatrix} 0 & 0 & -I & 0 \\ 0 & E & -D & I \\ I & -C & B & 0 \\ 0 & -I & 0 & 0 \end{bmatrix}.$$

From the structure of G it is clear that condition (ii) is satisfied. Next, let us consider the PPT with respect to B. This is given by

$$H = \begin{bmatrix} B^{-1} & -B^{-1}C & -B^{-1} & 0 \\ DB^{-1} & E - DB^{-1}C & -DB^{-1} & I \\ -B^{-1} & B^{-1}C & B^{-1} & 0 \\ 0 & -I & 0 & 0 \end{bmatrix}.$$

Since B is nonsingular, every row and every column of B^{-1} has a nonzero entry. Using this and the structure of H it is easy to see that every column of H (except the last block of columns) has a negative entry. Therefore, H satisfies condition (ii). This completes the proof of the theorem. □

COROLLARY 13. *Suppose $A \in \boldsymbol{R}^{n\times n}$ is column sufficient and $q \in \boldsymbol{R}^n$. Then the problem (q, A) can be processed by Algorithm* 2.

Proof. Solve the augmented problem $(\bar{q}, M)$ and use Theorem 11 to obtain a solution to (q, A), if it has one; otherwise, conclude that (q, A) has no solution. □

We shall solve the problem (q, A) mentioned in Example 9 by applying Algorithm 2 to the augmented problem. The augmented matrix M and the augmented vector $\bar{q}$ are given by

$$M = \begin{bmatrix} 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix},\ \bar{q} = \begin{bmatrix} -1 \\ 2 \\ 3 \\ \lambda \\ \lambda \\ \lambda \end{bmatrix},$$

where λ is a large positive number. Following Algorithm 2, the first PPT is with respect to $\alpha = \{1, 4\}$. Let $(\bar{q}^1, M^1)$ denote the PPT of $(\bar{q}, M)$ with respect to $\alpha = \{1, 4\}$. Then

$$M^1 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \ \bar{q}^1 = \begin{bmatrix} \lambda \\ 2 \\ 3+\lambda \\ 1 \\ \lambda \\ \lambda \end{bmatrix}.$$

Since $\bar{q}^1 \geq 0$, a solution to $(\bar{q}, M)$ is given by $(\lambda, 0, 0, 1, 0, 0)^t$. From Theorem 11, we conclude that (q, A) has no solution.

It is to be observed that when Algorithm 2 is applied to problems involving positive semidefinite matrices or row sufficient matrices, the terminal conclusion is that one either has a solution or that the problem has no *feasible* solution (that is, $F(q, A) = \emptyset$). However, in the above example, the problem has a feasible solution, yet we are able to conclude that the problem has no solution.

We shall conclude this paper with the following question. The augmented matrix given in (1) is $\boldsymbol{Q_0}$ when A is a column sufficient matrix. Is this true that the augmented matrix is a $\boldsymbol{Q_0}$-matrix for an arbitrary square matrix A? The authors are not aware of an answer to this problem.

Acknowledgment. The authors wish to thank Professor T. Parthasarathy for some useful discussions they had during his recent visit to the Indian Statistical Institute.

REFERENCES

[1] A. K. Biswas and G. S. R. Murthy, *A note on* $\boldsymbol{E_0^f} \cap \boldsymbol{Q_0}$*-matrices*, in Game Theoretical Applications to Economics and Operations Research, T. Parthasarathy, B. Dutta, J. A. M. Potters, T. E. S. Raghavan, D. Ray, and A. Sen, eds., Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 149–152.

[2] R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[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] R. W. Cottle and R. E. Stone, *On the uniqueness of solutions to linear complementarity problems*, Math. Programming, 27 (1983), pp. 191–213.

[5] R. L. Graves, *A principal pivoting simplex algorithm for linear and quadratic programming*, Oper. Res., 15 (1967), pp. 482–494.

[6] C. E. Lemke and J. T. Howson, Jr., *Equilibrium points of bimatrix games*, SIAM J. Appl. Math., 12 (1964), pp. 413–423.

[7] G. S. R. Murthy and T. Parthasarathy, *Some properties of fully semimonotone* $\boldsymbol{Q_0}$*-matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1268–1286.

[8] G. S. R. Murthy and T. Parthasarathy, *Fully copositive matrices*, Math. Programming, 82 (1998), pp. 401–411.

[9] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, Berlin, 1988.

[10] J. S. Pang, On $\boldsymbol{Q}$-matrices, Math. Programming, 17 (1979), pp. 243–247.

[11] R. E. Stone, *Geometric Aspects of Linear Complementarity Problem*, Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, 1981.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 533–553

FAST STRUCTURED TOTAL LEAST SQUARES ALGORITHM FOR SOLVING THE BASIC DECONVOLUTION PROBLEM*

NICOLA MASTRONARDI†, PHILIPPE LEMMERLING‡, AND SABINE VAN HUFFEL‡

Abstract. In this paper we develop a fast algorithm for the basic deconvolution problem. First we show that the kernel problem to be solved in the basic deconvolution problem is a so-called structured total least squares problem. Due to the low displacement rank of the involved matrices and the sparsity of the generators, we are able to develop a fast algorithm. We apply the new algorithm on a deconvolution problem arising in a medical application in renography. By means of this example, we show the increased computational performance of our algorithm as compared to other algorithms for solving this type of structured total least squares problem. In addition, Monte-Carlo simulations indicate the superior statistical performance of the structured total least squares estimator compared to other estimators such as the ordinary total least squares estimator.

Key words. deconvolution, structured total least squares, displacement rank, structured total least norm, generalized Schur algorithm

AMS subject classifications. 15A03, 62P10, 65C05

PII. S0895479898345813

1. Introduction.

Deconvolution problems occur in many areas such as reflection seismology, telecommunications, and medical applications [5, 23, 3, 12], to name just a few. In this paper we develop a fast algorithm for the basic deconvolution problem. The latter problem is depicted in Figure 1.1, where $u(k)$ represents the measured input (whereas $u_0(k)$ is the true but unmeasured input) and $y(k)$ represents the measured output (whereas $y_0(k)$ is the true but unmeasured output) at time k; $n_u(k)$ and $n_y(k)$ are i.i.d. white Gaussian measurement noise added, respectively, to the input and to the output. The system, represented by its transfer function $X_0(z)$, is a linear time-invariant system with impulse response $x \in \mathbb{R}^{n\times 1}$. The basic deconvolution problem can now be formulated as follows:

> Given the noisy (we assume Gaussian i.i.d. additive noise of equal variance) measurements $u(k), k = 1, \ldots, m+n-1$ and $y(k), k = 1, \ldots, m$, of the linear system, find a maximum likelihood (ML) estimate for the system impulse response $x_0(i), i = 1, \ldots, n$.

*Received by the editors October 8, 1998; accepted for publication (in revised form) by G. Golub April 10, 2000; published electronically August 9, 2000. This work was supported by the National Research Council of Italy under the Short Term Mobility Program, by the EC "Training and Mobility for Researchers" project entitled "Advanced Signal Processing for Medical Magnetic Resonance Imaging and Spectroscopy" (contract ERBFMRXCT970160), by the Belgian Programme on Interuniversity Poles of Attraction (IUAP-4/2 & 24), initiated by the Belgian State, Prime Minister's Office for Science, and by a Concerted Research Action (GOA) project of the Flemish Community, entitled "Model-based Information Processing Systems."

http://www.siam.org/journals/simax/22-2/34581.html

†Dipartimento di Matematica, Università della Basilicata, via N. Sauro 85, 85100 Potenza, Italy (mastronardi@unibas.it).

‡Department of Electrical Engineering, ESAT-SISTA, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Heverlee, Belgium (philippe.lemmerling@esat.kuleuven.ac.be, sabine.vanhuffel@esat.kuleuven.ac.be). The second author is a Ph.D. student funded by the IWT (Flemish Institute for Support of Scientific-Technological Research in Industry). The third author is a Senior Research Associate with the F.W.O. (Fund for Scientific Research-Flanders).

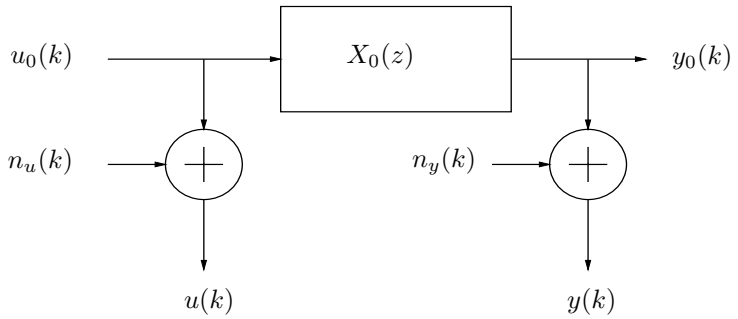


FIG. 1.1. *This figure shows a schematic outline of the basic deconvolution problem. The goal is to estimate the impulse response x_0 starting from the noisy input ($u(k)$) and the noisy output ($y(k)$).*

In the next section we show that the ML estimator for the basic deconvolution problem is a so-called structured total least squares (STLS) problem [19]. The STLS problem is an extension of the ordinary total least squares (TLS) problem [11, 29]. The ordinary TLS problem can be formulated as follows:

$$\begin{aligned} &\min_{\Delta A, \Delta y, x} \|[\Delta A \;\; \Delta y]\|_F^2 \\ &\text{such that (s.t.)} \;\; (A + \Delta A)x = y + \Delta y, \end{aligned} \tag{1.1}$$

with $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^{m \times 1}$. The STLS formulation additionally imposes a structure on the correction matrix $[\Delta A \;\; \Delta y]$ (e.g., a Hankel structure), hence its name *structured* TLS. Furthermore, it is possible that the *different* elements in $[\Delta A \;\; \Delta y]$ get a user-defined weighting, different from the one represented by the Frobenius norm of $[\Delta A \;\; \Delta y]$.

In recent years many problem formulations and associated solution methods have been devised for the STLS problem: the structured total least norm (STLN) approach [24, 25, 30], the constrained total least squares (CTLS) approach [1, 2], and the Riemannian singular value decomposition (RiSVD) approach [7]. We will use the straightforward optimization approach adopted in the STLN framework, since the other approaches either do not have efficient algorithms to solve them (e.g., the RiSVD approach) or they introduce numerical inaccuracies by forming products involving the data matrix $[A \;\; y]$ and its transpose (e.g., the CTLS approach). The basic deconvolution problem, formulated as $Ax \approx y$ and described in section 2, is solved using the 2-norm STLN algorithm described in [25], with A Toeplitz-structured and y unstructured. This iterative STLN algorithm solves as kernel problem in each iteration step a least squares problem involving a higher-dimensional structured and sparse data matrix. However, the STLN algorithm in [25], requiring $O\left((m+n)^3\right)$ operations, does not exploit these matrix features. A fast implementation requiring $O(mn^2 + m^2)$ operations which partially exploits the matrix structure was presented in [24]. In this paper we present a computationally faster algorithm of $O(mn + n^2)$ operations based on the generalized Schur algorithm [16]. The improved efficiency is obtained by fully exploiting the low displacement rank of the involved matrices (displacement rank 5) and the sparsity of the corresponding generators throughout all computations. In addition, we prove that this STLS estimator provides statistically better estimates of the impulse response given input and output data affected by i.i.d. zero mean noise with equal variance, compared to the ordinary TLS estimator. These

are the main contributions of this paper.

The paper is organized as follows. Section 2 describes the basic deconvolution problem and also outlines the 2-norm STLN algorithm with unstructured right-hand side, as described in [25], for solving this problem. Section 3 describes a fast algorithm for solving the kernel problem of the STLN approach applied to the basic deconvolution problem: a least squares (LS) problem involving structured matrices. As will be shown, the algorithm is based on the low displacement rank of the involved matrices. Section 4 describes some examples of typical deconvolution problems in order to demonstrate the dependency of the number of flops on the problem size and to show the increased computational efficiency w.r.t. existing implementations. To illustrate the statistical properties of the STLS estimator, a simulation experiment based on a medical application in renography is described. By means of a Monte-Carlo simulation, using several noise levels, we show the improved statistical accuracy of the deconvolution results obtained with the STLS estimator as compared to other estimators such as the TLS estimator that do not impose a structure on the correction matrix $[\Delta A \;\; \Delta y]$.

2. The basic deconvolution problem. Starting from the problem formulation of the basic deconvolution problem in section 1, it is straightforward to show that a ML estimate can be found as the solution of the following problem (for a proof, see [2]):

$$\begin{aligned} &\min_{E,x} \alpha^T \alpha + \beta^T \beta \\ &\text{s.t. } (A+E)x = y + \beta, \end{aligned} \tag{2.1}$$

with

$$A = \begin{bmatrix} u(n) & u(n-1) & \dots & u(1) \\ u(n+1) & u(n) & \dots & u(2) \\ \vdots & \ddots & & \vdots \\ u(m+n-1) & u(m+n-2) & \dots & u(m) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$E = \begin{bmatrix} \alpha(n) & \alpha(n-1) & \dots & \alpha(1) \\ \alpha(n+1) & \alpha(n) & \dots & \alpha(2) \\ \vdots & \ddots & & \vdots \\ \alpha(m+n-1) & \alpha(m+n-2) & \dots & \alpha(m) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$\beta = (A+E)x - y \in \mathbb{R}^{m \times 1},$$

with E the correction applied to A, β the correction applied to y, $y \in \mathbb{R}^{m \times 1}$ the output, and $x \in \mathbb{R}^{n \times 1}$ the impulse response. Problem (2.1) is a STLS problem, since corrections can be applied to the left-hand side matrix A of the constraints in (2.1) (implying that it is a *total* LS type problem) and in addition the corresponding correction matrix E is structured (implying that we have to deal with a *structured* TLS problem). As already mentioned in the introduction, we will apply the STLN approach, implying that we solve (2.1) as an optimization problem. Using the zeroth and first order terms of the Taylor series expansion of $\beta = (A + E(\alpha))x - y$ (where we use the notation $E(\alpha)$ to denote the dependence of E on α) around $[\alpha^T \;\; x^T]^T$, we obtain the Gauss–Newton method for solving (2.1) (for a proof, see [25]). The outline of the basic deconvolution algorithm, which is equivalent to the 2-norm STLN algorithm with unstructured right-hand side [25] for A Toeplitz, is then as follows:

Basic Deconvolution Algorithm

Input: extended data matrix $[A \;\; y] \in \mathbb{R}^{m\times(n+1)}$ $(m > n)$ of full rank $n+1$.
Output: correction vector α and parameter vector x s.t. $\alpha^T\alpha + \beta^T\beta$ is as small as possible and $\beta = (A + E(\alpha))x - y$.
Step 1: $\alpha \leftarrow 0$
$x \leftarrow A\backslash y$
Step 2: while stop criterion not satisfied
Step 2.1: $\min_{\Delta x,\Delta\alpha} \left\| M \begin{bmatrix} \Delta\alpha \\ \Delta x \end{bmatrix} + \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \right\|_2$
with $M = \begin{bmatrix} X & A+E \\ I & 0 \end{bmatrix} \in \mathbb{R}^{(2m+n-1)\times(m+2n-1)}$
Step 2.2: $x \leftarrow x + \Delta x$
$\alpha \leftarrow \alpha + \Delta\alpha$
end

with $X \in \mathbb{R}^{m\times(m+n-1)}$ defined such that $X\alpha = Ex$. Taking E as defined previously in the deconvolution problem (2.1), X becomes

$$X = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(1) & 0 & \cdots & \cdots & 0 \\ 0 & x(n) & x(n-1) & \cdots & x(1) & 0 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & x(n) & x(n-1) & \cdots & \cdots & x(1) \end{bmatrix}.$$

Note that $A\backslash y$ in Step 1 is a shorthand notation for the LS solution of the overdetermined system of equations $Ax \approx y$. As described in [17], more advanced initialization steps are possible. They yield better starting values in the sense that convergence takes place in fewer iterations and to a better local minimum. However the price to be paid is an increase in computational complexity of the initialization. Due to the nature of the problem we consider, the simple LS estimate will turn out to be sufficient in the considered application.

We use the following stop criterion in our implementation of the algorithm:

$$\|[\Delta\alpha^T \;\; \Delta x^T]\|_2 < 10^{-6}.$$

3. Fast algorithm. In this section we describe a fast algorithm for solving the LS problem in Step 2.1 of the basic deconvolution algorithm described in the previous section. It basically consists of a fast triangularization of the matrix M, followed by the solution of the normal equations

$$M^TM[\Delta\alpha^T \;\; \Delta x^T]^T = R^TR[\Delta\alpha^T \;\; \Delta x^T]^T = -M^T[\beta^T \;\; \alpha^T]^T,$$

with $M = Q[R^T \;\; 0]^T$ the QR factorization of M. The triangularization of the matrix M can be obtained by means of the QR decomposition of M with a computational complexity of $O((m+n)^3)$. The algorithm considered in [25] requires $O(mn^2 + m^2)$ flops. We propose an algorithm for computing the matrix R, based on the generalized Schur algorithm, exploiting displacement representation [14, 8] of the matrix M^TM that requires $O(mn + n^2)$ flops. First we describe briefly the displacement representation of a matrix (see [8, 14] for more details).

Let $Z_k \in \mathbb{R}^{k\times k}$ be the lower shift matrix, that is

$$Z_k = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$$

and $x \in \mathbb{R}^k$. We denote by $L_k(x)$ the so-called Krylov matrix generated by x :

$$L_k(x) = \left[x, Z_k x, Z_k^2 x, \ldots, Z_k^{k-1} x\right].$$

The following lemma holds [8].

LEMMA 3.1. *For an arbitrary matrix* $A \in \mathbb{R}^{k\times k}$,

$$A - Z_k A Z_k^T = \sum_{i=1}^{\hat{\delta}} g_i h_i^T \text{ if and only if } A = \sum_{i=1}^{\hat{\delta}} L_k(g_i) L_k(h_i)^T,$$

$g_i, h_i \in \mathbb{R}^k,\ i = 1, \ldots, \hat{\delta}$.

The matrix pair $G_{\hat{\delta}}(A) = \{X, Y\}$, where $X = \left[g_1, \ldots, g_{\hat{\delta}}\right]$ and $Y = \left[h_1, \ldots, h_{\hat{\delta}}\right]$, is called a *generator* of A. Generators are clearly not unique and can be of different lengths. The smallest possible length is called the *displacement rank* of A and is denoted by $\delta(A)$.

Remark 3.1. A symmetric matrix A has a symmetric generator, in the sense that $g_i = \pm h_i, i = 1, \ldots, \hat{\delta}$. Hence, its displacement representation has the symmetric form

$$A = \sum_{i=1}^{p} L_k(g_i) L_k(g_i)^T - \sum_{i=p+1}^{\hat{\delta}} L_k(g_i) L_k(g_i)^T.$$

The matrix

$$G = \begin{bmatrix} g_1^T \\ \vdots \\ g_{\hat{\delta}}^T \end{bmatrix}$$

is called a generator matrix.

DEFINITION 3.1. *A generator matrix is said to be in* proper form *if its first nonzero column has a single nonzero entry, i.e.,*

$$G = \begin{bmatrix} 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & * & \cdots & * \\ 0 & * & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix},$$

where the elements denoted by "$*$*" are generally different from zero. The row corresponding to the nonzero entry is called the* pivot.

Given a displacement representation of $M^T M$, it is possible to compute a factorization with a computational complexity proportional to the displacement rank of $M^T M$. Since $M^T M$ is a block-Toeplitz-like matrix it is more natural to consider its displacement representation with respect to the block-shift matrix $Z = Z_{m+n-1} \oplus Z_n$ [13]. For clarity of exposition, this is illustrated in Appendix A for a 5×5 matrix.

3.1. Generators for M^TM. For the sake of brevity, in the following we indicate the Krylov matrices $L_{m+2n-1}(x)$ by $L(x)$. The displacement rank of M^TM is 5, that is, $\delta(M^TM) = 5$.

LEMMA 3.2. *Let* $w = M(2 : 2m+n-1, m+n)/\|M(2 : 2m+n-1, m+n)\|_2$ *and* $t = M(2 : 2m+n-1, 2 : m+2n-1)^T w$. *The following vectors form a generator of* M^TM*:*

$$\begin{aligned} g_1 &= M(1,:)^T, \\ g_2 &= e_1, \\ g_3 &= [0, t^T]^T, \\ g_4 &= [0, t(1 : m+n-2)^T, 0, t(m+n : m+2n-2)^T]^T, \\ g_5 &= [0, M(m, 1 : m+n-2)^T, 0, M(m, m+n : m+2n-2)^T]^T, \end{aligned}$$

where $e_1 = [1, \underbrace{0, \ldots, 0}_{m+2n-2}]^T$. *Then*

$$M^TM = \sum_{i=1}^{3} L(g_i)L(g_i)^T - \sum_{i=4}^{5} L(g_i)L(g_i)^T. \tag{3.1}$$

Proof. Construct M^TM and ZM^TMZ^T; then straightforward manipulations show that $M^TM - ZM^TMZ^T$ can be expressed as a sum of five rank 1 matrices. ∎

Following the technique described in [8, 22] we can easily construct the generalized Schur algorithm for the computation of R with computational complexity of $O(m^2 + mn + n^2)$.

In the following section we consider a fast version of the generalized Schur algorithm that requires only $18mn + 34.5n^2$ flops, taking into account the *"sparsity"* of the vectors $g_i, i = 1, \ldots, 5$.

3.2. Description of the algorithm. A matrix Θ is said to be J_{M^TM}-orthogonal if $\Theta^T J_{M^TM} \Theta = J_A$, where $J_{M^TM} = \mathrm{diag}(1, 1, 1, -1, -1)$.

Let

$$G_0 = \begin{bmatrix} g_1^T \\ g_2^T \\ g_3^T \\ g_4^T \\ g_5^T \end{bmatrix}.$$

Denoting by G_{i-1} the generator matrix at the beginning of the ith iteration of the algorithm, a J_{M^TM}-orthogonal matrix Θ_i is chosen such that $H_{i-1} = \Theta_i G_{i-1}$ is in proper form, having the first row as pivot.

The generator matrix G_i is updated in the following way:

$$\begin{aligned} G_i(1,:) &= H_{i-1}(1,:)Z^T, \\ G_i([2:5],:) &= H_{i-1}([2:5],:). \end{aligned}$$

Furthermore, $H(i-1,:)$ becomes the ith row of R.

Starting from the vectors $g_i, i = 1, \ldots, 5$, and following the same steps of the method proposed in [8, 22] (see also Appendix A), we transform these vectors in the following way:

$$\begin{bmatrix} g_i^T \\ g_j^T \end{bmatrix} := Q \begin{bmatrix} g_i^T \\ g_j^T \end{bmatrix}, \tag{3.2}$$

where Q is either a Givens rotation (updating) if $L(g_i)$ and $L(g_j)$ have the same sign in the sum (3.1) or a hyperbolic rotation (downdating) if these terms have opposite sign in the sum (3.1). We perform the downdating step by means of a stabilized hyperbolic rotation [27], since the latter is more stable. Furthermore, at the kth iteration, the matrix Q is chosen to annihilate the kth entry of the resulting vector g_j. At the end of the kth iteration, we have $g_j(k) = 0, j \neq 1$. Then $g_1(k : m + 2n - 1)$ is the kth row of R and we set $g_1 := Zg_1$.

We divide the algorithm into four phases:
(1) initialization: $i = 1$,
(2) the iterations for $i = 2 : m$,
(3) the iterations for $i = m + 1 : m + n - 1$,
(4) the iterations for $i = m + n : m + 2n - 1$.

3.2.1. Initialization: $i = 1$. The only vectors with the first entry different from 0 are g_1 and g_2. The new vectors $\tilde{g}_1$ and $\tilde{g}_2$ are computed as

$$\begin{bmatrix} \tilde{g}_1^T \\ \tilde{g}_2^T \end{bmatrix} = G \begin{bmatrix} g_1^T \\ g_2^T \end{bmatrix},$$

where G is the Givens rotation chosen to annihilate the first element of g_2. The first row of R is $\tilde{g}_1^T$. In the following we denote by $g_k^{(i)}, k = 1, \ldots, 5$, the vectors at the ith iteration. Then we define

$$\begin{aligned} g_1^{(1)} &= Z\tilde{g}_1 = [0, \tilde{g}_1(1 : m + n - 2)^T, 0, \tilde{g}_1(m + n : m + 2n - 2)^T]^T, \\ g_2^{(1)} &= \tilde{g}_2, \\ g_3^{(1)} &= g_3, \\ g_4^{(1)} &= g_4, \\ g_5^{(1)} &= g_5. \end{aligned}$$

The number of flops for this phase is $4n$.

3.2.2. Iterations for $i = 2 : m$. In each iteration of this phase $L(g_1^{(i-1)})$ is updated with $L(g_2^{(i-1)})$, and $L(g_3^{(i-1)})$, $L(g_4^{(i-1)})$ is updated with $L(g_5^{(i-1)})$, and $L(g_1^{(i-1)})$ is downdated with $L(g_4^{(i-1)})$. Since g_5 has m initial zeros, $L(g_5^{(1)})$ does not contribute to this phase.

Moreover, the structure of the vectors $g_k^{(i-1)}, k = 1, 2$, at the beginning of the ith iteration is

$$\begin{aligned} g_1^{(i-1)} &= [\underbrace{0, \ldots, 0}_{i-1}, \underbrace{*, \cdots, *, *}_{n}, \underbrace{0, \ldots, 0}_{m-i+1}, \underbrace{*, \cdots, *}_{n-1}]^T, \\ g_2^{(i-1)} &= [\underbrace{0, \ldots, 0}_{i-1}, \underbrace{*, \cdots, *}_{n-1}, \underbrace{0, \ldots, 0, 0}_{m-i+1}, \underbrace{*, \cdots, *}_{n}]^T, \end{aligned}$$

where the entries $*$ are in general different from 0. Then updating $L(g_1^{(i-1)})$ with $L(g_2^{(i-1)})$ fills in only the $(m + n)$th entry of the updated $g_1^{(i-1)}$. To complete the ith iteration, $L(g_1^{(i-1)})$ must be updated with $L(g_3^{(i-1)})$ and downdated with $L(g_4^{(i-1)})$. To explain this computation, we describe the iteration for $i = 2$, recalling that the vectors $g_3^{(1)}$ and $g_4^{(1)}$ are equal, except for the $(m + n)$th element (the entries of these vectors are generally different from 0). Let

$$g_1^{(1)} = [0, \xi_2, \ldots, \xi_{n+1}, \underbrace{0, \ldots, 0}_{m-2}, \xi_{m+n}, \ldots, \xi_{m+2n-1}]^T, \tag{3.3}$$
$$g_3^{(1)} = [0, \zeta_2, \ldots, \zeta_{m+n-1}, \zeta_{m+n}, \zeta_{m+n+1}, \ldots, \zeta_{m+2n-1}]^T,$$
$$g_4^{(1)} = [0, \zeta_2, \ldots, \zeta_{m+n-1}, \mu_{m+n}, \zeta_{m+n+1}, \ldots, \zeta_{m+2n-1}]^T.$$

We observe that the $(m+n)$th entry of $g_4^{(1)}$ is equal to 0. The Givens rotation used during the updating is

$$G = \begin{bmatrix} c_G^{(1)} & s_G^{(1)} \\ -s_G^{(1)} & c_G^{(1)} \end{bmatrix}, \text{ with } c_G^{(1)} = \frac{\xi_2}{\sqrt{\xi_2^2 + \zeta_2^2}} \text{ and } s_G^{(1)} = \frac{\zeta_2}{\sqrt{\xi_2^2 + \zeta_2^2}}.$$

The updated vectors $\tilde{g}_1^{(1)}$ and $\tilde{g}_3^{(1)}$ are

$$\tilde{g}_1^{(1)} = c_G^{(1)} g_1^{(1)} + s_G^{(1)} g_3^{(1)}, \tag{3.4}$$

$$\tilde{g}_3^{(1)} = -s_G^{(1)} g_1^{(1)} + c_G^{(1)} g_3^{(1)}, \tag{3.5}$$

with $\tilde{g}_1^{(1)} = [0, \tilde{\xi}_2, \ldots, \tilde{\xi}_{m+2n-1}]^T \left(\tilde{\xi}_2 = \sqrt{\xi_2^2 + \zeta_2^2}\right)$. Moreover, we point out that, for (3.3),

$$\tilde{g}_3^{(1)}(n+2 : n+m-1) = c_G^{(1)} g_3^{(1)}(n+2 : n+m-1). \tag{3.6}$$

To finish the iteration for $i = 2$, $L(\tilde{g}_1^{(1)})$ must be downdated with $L(g_4^{(1)})$ by means of the stabilized hyperbolic rotation

$$H = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1-\rho^2}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \rho \\ 0 & 1 \end{bmatrix},$$

where ρ is such that $H[\tilde{\xi}_2, \zeta_2]^T = [\hat{\xi}_2, 0]^T$. Taking (3.4) into account, it is straightforward to see that

$$\rho = -s_G^{(1)} \text{ and } \sqrt{1-\rho^2} = c_G^{(1)}.$$

The downdated vectors $\hat{g}_1^{(1)}$ and $\tilde{g}_4^{(1)}$ are

$$\hat{g}_1^{(1)} = \frac{\tilde{g}_1^{(1)} - s_G^{(1)} g_4^{(1)}}{c_G^{(1)}} = g_1^{(1)} + \frac{s_G^{(1)} \left(g_3^{(1)} - g_4^{(1)}\right)}{c_G^{(1)}} \tag{3.7}$$

and

$$\tilde{g}_4^{(1)} = -s_G^{(1)} \hat{g}_1^{(1)} + c_G^{(1)} g_4^{(1)}. \tag{3.8}$$

Hence,

$$\begin{aligned} \tilde{g}_4^{(1)} &= -s_G^{(1)} \hat{g}_1^{(1)} + c_G^{(1)} g_4^{(1)} \\ &= -s_G^{(1)} g_1^{(1)} + \frac{{c_G^{(1)}}^2 g_4^{(1)} - {s_G^{(1)}}^2 (g_3^{(1)} - g_4^{(1)})}{c_G^{(1)}} \\ &= -s_G^{(1)} g_1^{(1)} + \frac{g_4^{(1)} - (1 - {c_G^{(1)}}^2) g_3^{(1)}}{c_G^{(1)}}. \end{aligned} \tag{3.9}$$

From (3.5) and (3.9), $\tilde{g}_3^{(1)}$ and $\tilde{g}_4^{(1)}$ continue to be equal, except for the $(m+n)$th entry. Furthermore, from (3.7), we observe that $g_1^{(1)}$ and $\hat{g}_1^{(1)}$ differ in their $(n+m)$th entry. $\hat{g}_1^{(1)}$ now becomes the 2nd row of R, and, for the next iteration, the updated vectors are

$$\begin{aligned}
g_1^{(2)} &= Z\hat{g}_1^{(1)} = \left[0, 0, \hat{g}_1^{(1)}(2 : m+n-2)^T, 0, \hat{g}_1^{(1)}(m+n : m+2n-2)^T\right]^T, \\
g_5^{(2)} &= g_5^{(1)}, \\
g_2^{(2)} &= g_2^{(1)}, \\
g_3^{(2)} &= \tilde{g}_3^{(1)}, \\
g_4^{(2)} &= [\tilde{g}_3^{(1)}(1 : m+n-1)^T, \gamma, \tilde{g}_3^{(1)}(m+n+1 : m+2n-1)^T]^T,
\end{aligned}$$

where $\gamma = -s_G^{(1)}\hat{g}_1^{(1)}(m+n) + c_G^{(1)}g_4^{(1)}(m+n)$. To reduce the computational cost of this phase, we observe that it is not necessary to calculate $g_3^{(2)}(n+3 : m+n-1)$ since at the next iteration the corresponding entries of the vector $g_1^{(2)}$ are equal to 0. Hence for the vector $g_3^{(3)}(n+4 : m+n-1)$ the following relation holds:

$$g_3^{(3)}(n+4 : m+n-1) = c_G^{(2)}c_G^{(1)}g_3^{(1)}(n+4 : m+n-1)$$

and, at the ith iteration,

$$g_3^{(i-1)}(n+i : m+n-1) = c_G^{(i-2)} \cdots c_G^{(2)}c_G^{(1)}g_3^{(1)}(n+i : m+n-1).$$

Hence it is sufficient to store the partial product

$$c_G^{(i-2)} \cdots c_G^{(2)}c_G^{(1)} \tag{3.10}$$

into a temporary variable and multiply $g_3^{(1)}(n+i-1)$ with this variable at the beginning of the ith iteration. At the end of each iteration we set $R(i, i : m+2n-1) = \hat{g}_1^{(i-1)}(i : m+2n-1)$, $g_1^{(i)} = Z\hat{g}_i^{(i-1)}$. Hence the number of flops of this phase is $18mn$.

3.2.3. Iterations for $i = m+1 : m+n-1$. The only difference of this phase with the previous one is that $L(g_5^{(i-1)})$ must also be downdated from $L(g_1^{(i-1)})$. Let $\hat{g}_1^{(i-1)}$ be the first generator at the end of the ith iteration. We set $R(i, i : m+2n-1) = \hat{g}_1^{(i-1)}(i : m+2n-1)$, $g_1^{(i)} = Z\hat{g}_i^{(i-1)}$. The number of flops of this phase is $22.5n^2$.

3.2.4. Iterations for $i = m+n : m+2n-1$. This phase is similar to the previous one. The only difference is that the vector $g_4^{(i-1)}$ must also be computed, since now it differs from $g_3^{(i-1)}$. Let $\hat{g}_1^{(i-1)}$ be the first generator at the end of the ith iteration. Also after each iteration of this phase we set $R(i, i : m+2n-1) = \hat{g}_1^{(i-1)}(i : m+2n-1)$, $g_1^{(i)} = Z\hat{g}_i^{(i-1)}$. The number of flops of this phase is $12n^2$.

The Matlab-like code to compute the R factor by means of the described algorithm can be found in Appendix B.

3.3. Modified problem. We will now consider a slightly modified problem (2.1). The modification consists of the introduction of an error-free zero upper triangular part in the matrix A and by consequence E also has a zero upper triangular part. This modified problem typically arises if the system is assumed to be causal with zero initial state, implying that its inputs $u(t)$ are zero for $t \leq 0$. In this case,

the first $n-1$ inputs are zero followed by m nonzero values. The latter means that $A \in \mathbb{R}^{m\times n}$ and $E \in \mathbb{R}^{m\times n}$ are as follows:

$$A = \begin{bmatrix} u(1) & 0 & \dots & 0 \\ u(2) & u(1) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & u(1) \\ \vdots & & & \vdots \\ u(m) & u(m-1) & \dots & u(m-n+1) \end{bmatrix},$$

$$E = \begin{bmatrix} \alpha(1) & 0 & \dots & 0 \\ \alpha(2) & \alpha(1) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \alpha(1) \\ \vdots & & & \vdots \\ \alpha(m) & \alpha(m-1) & \dots & \alpha(m-n+1) \end{bmatrix}.$$

By consequence, X becomes (remember that X is defined by $X\alpha = Ex$)

$$X = \begin{bmatrix} x(1) & & & & & \\ x(2) & x(1) & & & & \\ \vdots & \ddots & \ddots & & & \\ x(n) & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & x(n) & \cdots & x(2) & x(1) \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

Again we define M as

$$M = \begin{bmatrix} X & A+E \\ I & 0 \end{bmatrix} \in \mathbb{R}^{2m\times(m+n)},$$

with $I \in \mathbb{R}^{m\times m}$ the identity matrix and $0 \in \mathbb{R}^{m\times n}$ the null matrix. The displacement rank of M^TM with respect to the block shift matrix $Z = Z_m \oplus Z_n$ is 5. Let $w_1 = M(:,1)/\|M(:,1)\|_2$ and $t_1 = M^Tw_1$. Let $w_2 = M(:,m+1)/\|M(:,m+1)\|_2$ and $t_2 = M^Tw_2$. Then the generators of M^TM with respect to Z are

$$\begin{aligned} g_1 &= t_1, \\ g_2 &= [0, t_2(2:m+n)^T]^T, \\ g_3 &= [0, t_2(2:m)^T, 0, t_2(m+2:m+n)^T]^T, \\ g_4 &= [0, t_1^T(2:m+n)]^T, \\ g_5 &= [0, M(m,1:m-1)^T, 0, M(m,m+1:m+n-1)^T]^T, \end{aligned}$$

and

$$M^TM = \sum_{i=1}^{2} L(g_i)L(g_i)^T - \sum_{i=3}^{5} L(g_i)L(g_i)^T.$$

Also in this case, taking into account the sparsity of the vectors, g_1, g_4, g_5, and since g_2 and g_3 differ only in the $(m+1)$th entry, following the same technique described in section 3.2, it is possible to construct an algorithm for the fast triangularization of M requiring $18mn + 16.5n^2$ flops. As a matter of fact, the algorithm for solving this modified deconvolution problem[1] (which will be referred to as the *modified deconvolution algorithm*) has the same outline as described in section 2, when the appropriate matrices A, E, M, and X are used. For the sake of brevity, we omit the Matlab-like code. The corresponding Matlab m-files can be obtained from the authors upon request.

3.4. Stability of the computation of the R factor. The stability of the proposed algorithm is studied in [20]. In [27] the generalized Schur algorithm is proved to be backward stable, provided one hyperbolic rotation is performed in a stable way [4] at each step. In phase 1 and phase 2 of the algorithm described in the section 3, two hyperbolic rotations are performed in a stable way. In [20] it is proved that also in this case the generalized Schur algorithm performs reliably and the following result holds.

THEOREM 3.1. *Let G be the generator matrix of M. Let R be the matrix of the QR factorization of M computed by means of the generalized Schur algorithm applying a sequence of Givens rotations and two mixed hyperbolic rotations per iteration. Then*

$$\|M^T M - R^T R\|_F \leq 62(m+n-1)(m+n)\varepsilon \left(2\sqrt{m+n}\|R\|_F + \|G\|_F^2\right).$$

4. Numerical experiments. In this section we illustrate by means of two examples the efficiency of the algorithms described in section 3. To illustrate the statistical properties of the STLS estimator, a simulation experiment based on a medical application in renography is introduced. The next subsection describes the three examples that will be used in this section.

4.1. Examples. The first example is a typical deconvolution problem. Referring to Figure 1.1, we start from the exact impulse response $x_0 = [1.9\ 3.3\ 4.4\ 5.4\ 5.9\ 6.2\ 6.3\ 6.4\ 6.5\ 6.45\ 6.3\ 6.2\ 6.1\ 5.9\ 5.8\ 5.7\ 5.6\ 5.4\ 5.3\ 5.0\ 4.85\ 4.6\ 4.0\ 3.4\ 2.8\ 1.7\ 1.3\ 1.0\ 0.2\ 0.0]^T \in \mathbb{R}^{30\times 1}$. The true but unmeasured input u_0 is i.i.d. Gaussian noise of unit variance, except for the first 29 entries which equal 0. The latter implies that the first example is an illustration of the modified problem described in section 3.3. The true but unmeasured output y_0 is calculated as a convolution of x_0 and u_0. The measured input u and output y are obtained by perturbing u_0 and y_0 with i.i.d. Gaussian noise of variance $1e-4$.

The second example is similar to the previous one but now the exact impulse response x_0 equals $[1.9\ 3.3\ 4.4\ 5.4\ 5.9\ 6.2\ 6.3\ 6.1\ 5.8\ 5.6\ 5.3\ 5.0\ 4.85\ 4.6\ 4.0\ 3.4\ 1.8\ 1.0\ 0.2\ 0.0]^T \in \mathbb{R}^{20\times 1}$. Again u_0 is i.i.d. Gaussian noise of unit variance, but this time there are no initial zeros. Thus the latter serves as an example for the basic deconvolution problem (and not the modified one). The measured input u and output y are obtained by perturbing u_0 and y_0 with i.i.d. Gaussian noise of variance $1e-4$.

The third example is a deconvolution problem that occurs in renography [9]. The goal here is to determine via deconvolution the so-called renal retention function of the kidney, which in system theoretic terms corresponds to the impulse response x_0 of the system in Figure 1.1. This retention function visualizes the mean whole kidney

[1] Note that the problem described here is basically also a basic deconvolution problem, with a specific input sequence. However, a specific name is given to this problem and its corresponding algorithm so that they can easily be referred to.

transit time of one unit of a tracer injected into the patient, and it enables a physician to evaluate the renal function and renal dysfunction severity after transplantation. In order to obtain this impulse response, the following experiment is conducted. A radioactive tracer is injected in an artery of the patient. The measured input of the system (u in Figure 1.1) is the by noise perturbed arterial concentration of the radioactive tracer as a function of time. This concentration is measured by means of a gamma camera, and thus in discretized time $u(k)$ represents the number of counts registered in the vascular region at the entrance of the kidney under study during the kth sampling interval. The measured output $y(k)$ (the so-called renogram) represents the (by noise perturbed) number of counts registered in the whole kidney region by the gamma camera in the kth sampling interval. Deconvolution analysis of the renogram is based on modeling the kidney as a linear time-invariant causal system with zero-initial state. This is why we consider the modified problem, described in section 3.3. Since this third example will be used to investigate the statistical properties of the STLS estimator, we have to use a simulation example instead of a real in vivo measurement. We will use the same simulation example as described in [31]. The noiseless input is described as follows:

$$u_0(k+1) = Ae^{(-a_1 k\Delta t)} + Be^{(-a_2 k\Delta t)} + Ce^{(-a_3 k\Delta t)}, k = 0, 1, 2, \ldots, (t_{obs}/\Delta t) - 1,$$

with $A = 40.3$, $B = 45.2$, $C = 15.2$, $a_1 = 1.8$, $a_2 = 0.43$, and $a_3 = 0.035$. Δt represents the sampling interval, expressed in minutes, and t_{obs} is the total observation time. The exact impulse response is characterized by the following function:

$$x_0(k+1) = \begin{cases} 1 & k = 0, 1, 2, \ldots, (t_1/\Delta t) - 1 \\ (k\Delta t - t_2)/(t_1 - t_2) & k = t_1/\Delta t, (t_1/\Delta t) + 1, \ldots, (t_2/\Delta t) - 1 \end{cases},$$

with $t_1 = 3$ minutes, $t_2 = 5$ minutes, and $t_{obs} = 20$ minutes. The noiseless output y_0 is then obtained by convolution of the noiseless input u_0 with the proposed impulse response x_0. In matrix format we have that

$$Ax_0 = y_0,$$

where

$$A = \begin{bmatrix} u_0(1) & 0 & \ldots & 0 \\ u_0(2) & u_0(1) & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_0(1) \\ \vdots & \ddots & \ddots & \vdots \\ u_0(m) & u_0(m-1) & \ldots & u_0(m-n+1) \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

For $\Delta t = 1/3$ minutes, we have that $m \equiv 3t_{obs} = 60$ and $n \equiv 3t_2 = 15$. As described in [10], these functions and constants are a realistic simulation of real in vivo measurements.

4.2. Efficiency. In this subsection we first consider the dependence of the computational cost of the modified deconvolution algorithm for solving the modified problem described in section 3.3 on the problem size (i.e., the dimensions $m \times n$ of the matrix A). Given the iterative nature of the algorithm, the total number of floating point operations (flops) is a multiple of the flops necessary to execute Step 2.1 of

TABLE 4.1

This table shows the number of flops for the different parts of Step 2.1 *of the modified deconvolution algorithm, as well as their sum. The bold numbers show the flop count as measured by the Matlab* flops *command, whereas the other numbers represent the theoretical flop count.*

$m \times n$	part 1	part 2	part 3	part 4	part 5	total
600×30	**78243**	**73478**	**352652**	**73831**	**75036**	**653240**
	78243	73478	338850	73831	75036	639438
1200×30	**155643**	**147878**	**690452**	**147631**	**150036**	**1291640**
	155643	147878	662850	147631	150036	1264038
600×15	**41718**	**38168**	**179507**	**37816**	**39021**	**336230**
	41718	38168	165713	37816	39021	322436

the modified deconvolution algorithm. Therefore, we will analyze this step in further detail.

In Step 2.1 we solve a LS problem by solving the corresponding normal equations

$$R^T \left(R \begin{bmatrix} \Delta\alpha \\ \Delta x \end{bmatrix} \right) = -M^T \begin{bmatrix} \beta \\ \alpha \end{bmatrix}, \tag{4.1}$$

where R is the triangular factor of the QR factorization of M. From (4.1), the description of the fast QR algorithm in section 3.3 and the specific structure of M and R, we obtain the following theoretical number of computations for Step 2.1:

part 1: Calculation of the generators: $4mn + 9m + 0.5n^2 + 12.5n + 18$.

part 2: Construction of the right-hand side (see (4.1)): $4mn + 4m - n^2 - n + 8$.

part 3: Fast QR: $18mn + 16.5n^2$.

part 4: Solving a lower triangular system (see (4.1)): $4mn + 3m + n + 1$.

part 5: Solving an upper triangular system (see (4.1)): $4mn + 5m + n + 6$.

This means that overall the algorithm is $O(mn)$, when $m >> n$, as is mostly the case. To illustrate the dependence on m and n, we consider three different cases based on the first example described in subsection 4.1. The first case (first row of Table 4.1) is constructed by generating $u_0 \in \mathbb{R}^{600\times 1}$. The second case (second row of Table 4.1) is constructed by generating $u_0 \in \mathbb{R}^{1200\times 1}$. The third case (third row of Table 4.1) is similar to the first case (and thus also derived from the first example described in subsection 4.1), but now only the first 15 elements of x_0 are used to generate y_0 with $u_0 \in \mathbb{R}^{600\times 1}$. In Table 4.1 we give the number of flops of the different parts of Step 2.1, as well as their sum. Flops are calculated using the Matlab command *flops*. Note that each cell of the table is divided into two parts: the upper part (bold number) contains the flop count obtained using the Matlab command *flops*, whereas the lower part gives the theoretical number of flops as determined at the beginning of this subsection. Note the close resemblance between the upper and lower part of each cell. Furthermore, from the table we clearly can see that the computational cost of Step 2.1 is $O(mn)$.

To illustrate the better computational efficiency of the newly presented implementation of the basic deconvolution algorithm (referred to as alg_1), we compare its efficiency with that of the standard STLN approach [25, 30] (referred to as alg_2) and the faster STLN algorithm for Toeplitz STLS problems described in [24] (referred to as alg_3). As an example we take the second example described in the previous subsection because this is a basic deconvolution problem (see also [30]). We use an example different from the previous paragraph since the algorithm alg_3 solves the basic deconvolution problem and not the modified one. As a consequence alg_1 is the algorithm described in section 3.2. Table 4.2 shows three cases. The first ($m \times n = 300 \times 20$) and

TABLE 4.2

This table shows the following ratios: the number of flops for the calculation of the R factor in alg_2 w.r.t. the number of flops for the calculation of the R factor in alg_1 ($flops_{alg_2}/flops_{alg_1}$) and the number of flops for the calculation of the R factor in alg_3 w.r.t. the number of flops for the calculation of the R factor in alg_1 ($flops_{alg_3}/flops_{alg_1}$), for matrices A of dimension $m \times n$. The bold numbers shows the flop count as measured by the Matlab flops *command, whereas the other numbers come from the theoretical flop count.*

$m \times n$	$flops_{alg_2}/flops_{alg_1}$	$flops_{alg_3}/flops_{alg_1}$
300×20	**1327.67**	**10.41**
	1432.27	10.93
600×20	**4949.15**	**13.19**
	5362.55	13.94
300×10	**2290.15**	**8.70**
	2671.12	9.76

the second case ($m \times n = 600 \times 20$) are generated using the second example described in subsection 4.1, by generating, respectively, $u_0 \in \mathbb{R}^{319\times 1}$ and $u_0 \in \mathbb{R}^{619\times 1}$. The third case is also based on the second example described in subsection 4.1, but now $u_0 \in \mathbb{R}^{309\times 1}$ and only the first 10 elements of the impulse response x_0 are used to generate y_0. Since the algorithms only differ in the part where the fast (Q)R factorization is performed, Table 4.2 shows only the following figures: the number of flops for the calculation of the R factor in alg_2 w.r.t. the number of flops for the calculation of the R factor in alg_1 ($flops_{alg_2}/flops_{alg_1}$) and the number of flops for the calculation of the R factor in alg_3 w.r.t. the number of flops for the calculation of the R factor in alg_1 ($flops_{alg_3}/flops_{alg_1}$). Note that from section 3 and from the implementation in [24] we have the following theoretical flop counts for the (Q)R factorization:

alg_1: $18mn + 34.5n^2$,

alg_2: $3p^2(p - q/3), p = 2m + n - 1, q = m + 2n - 1$,

alg_3: $17mn + 3m^2 + 8mn^2$.

Again each cell in Table 4.2 is subdivided into an upper and a lower part. The upper part (bold numbers) contains the abovementioned ratios obtained using the Matlab command *flops*, whereas the lower part contains the abovementioned ratios based on the theoretical flop counts. We clearly see the computational advantage of alg_1 over alg_2 and alg_3, for different sizes $m \times n$ of the matrix A. Also notice the close resemblance between the theoretical and Matlab-based flop count.

4.3. Accuracy. We compare the statistical accuracy of the STLS estimator with that of the TLS estimator which in [31] was shown to outperform other estimators by far. To this end, we perform for each noise standard deviation σ_ν a Monte-Carlo simulation consisting of 100 runs. In every run, we add a different realization of i.i.d. white Gaussian noise with standard deviation σ_ν to the noiseless input u_0 and the noiseless output y_0 of the previously described medical simulation example. The obtained noisy vectors u and y are used as input to the modified deconvolution algorithm described at the beginning of section 3.3. To compare the performance of both estimators at a noise level σ_ν, we average for both estimators the relative error $\frac{\|x-x_0\|_2}{\|x_0\|_2}$ over the different runs. Table 4.3 shows that in the case of the STLS estimator, the relative errors are 9% to 14% lower than in the case of the TLS estimator, confirming the statistical superior performance of the STLS estimator.

5. Conclusions. We have proposed new fast implementations of the algorithms that solve the basic and modified deconvolution problem in a ML sense. The algorithms solve the corresponding STLS problems in $O(mn)$ flops by exploiting the low

TABLE 4.3
This table shows the relative error $\frac{\|x-x_0\|_2}{\|x_0\|_2}$ *for the TLS and STLS estimator, averaged over* 100 *runs, for different noise standard deviations* σ_ν.

σ_ν	TLS	STLS	σ_ν	TLS	STLS
0.05	0.0010	0.0009	0.5	0.0104	0.0091
0.071	0.0015	0.0013	0.707	0.0138	0.0123
0.087	0.0018	0.0016	0.866	0.0177	0.0153
0.1	0.0021	0.0018	1	0.0204	0.0180
0.158	0.0032	0.0029	1.58	0.0313	0.0279
0.224	0.0046	0.0041	2.24	0.0454	0.0402
0.274	0.0056	0.0049	2.74	0.0576	0.0513
0.316	0.0064	0.0057	3.16	0.0660	0.0601

displacement rank of the matrices involved in the basic deconvolution problem **and** the sparsity of the corresponding generators.

By means of a deconvolution example, we illustrate the improved efficiency of our implementation of the basic deconvolution algorithm as compared to other implementations of algorithms for solving this type of STLS problem. We use a medical example in renography to illustrate the superior statistical performance of the STLS estimator as compared to other estimators.

6. Appendix A. Triangularization of a symmetric positive definite matrix expressed by its displacement representation. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix with displacement representation

$$A = \sum_{i=1}^{q} L(g_i)L(g_i)^T - \sum_{i=q+1}^{p} L(g_i)L(g_i)^T, \quad p \le n, \tag{6.1}$$

where $g_i \in \mathbb{R}^n, i = 1, \ldots, p$. We can write (6.1) in the following way:

$$\begin{aligned} A &= [L(g_1), \ldots L(g_q), L(g_{q+1}), \ldots, L(g_p)] \begin{bmatrix} I & & & & & & \\ & \ddots & & & & & \\ & & I & & & & \\ & & & -I & & & \\ & & & & \ddots & & \\ & & & & & -I & \end{bmatrix} \begin{bmatrix} L(g_1)^T \\ \vdots \\ L(g_q)^T \\ L(g_{q+1})^T \\ \vdots \\ L(g_p)^T \end{bmatrix} \\ &= [L(g_1), \ldots L(g_q), L(g_{q+1}), \ldots, L(g_p)]J \begin{bmatrix} L(g_1)^T \\ \vdots \\ L(g_q)^T \\ L(g_{q+1})^T \\ \vdots \\ L(g_p)^T \end{bmatrix}, \end{aligned}$$

where I is the identity matrix of order n. We say that a matrix Q is J-orthogonal if $J = QJQ^T$. To compute the Cholesky factor of A, we have to construct a J-orthogonal

matrix Q such that

$$Q^T \begin{bmatrix} L(g_1)^T \\ \vdots \\ L(g_m)^T \\ L(g_{m+1})^T \\ \vdots \\ L(g_p)^T \end{bmatrix} = \begin{bmatrix} R \\ O \\ \vdots \\ \vdots \\ O \end{bmatrix},$$

where $R, O \in \mathbb{R}^{n\times n}$, R is upper triangular, and O is the null matrix.

As an example, we briefly describe how to obtain the matrix R in case $A \in \mathbb{R}^{5\times 5}$, and its displacement representation is given by

$$A = L(g_1)L(g_1)^T + L(g_2)L(g_2)^T - L(g_3)L(g_3)^T - L(g_4)L(g_4)^T.$$

Then

$$J = \begin{bmatrix} I & & & & \\ & I & & & \\ & & -I & & \\ & & & -I & \end{bmatrix},$$

where I is the identity matrix of order 5, and

$$L^T = \begin{bmatrix} g_1(1) & g_1(2) & g_1(3) & g_1(4) & g_1(5) \\ & g_1(1) & g_1(2) & g_1(3) & g_1(4) \\ & & g_1(1) & g_1(2) & g_1(3) \\ & & & g_1(1) & g_1(2) \\ & & & & g_1(1) \\ g_2(1) & g_2(2) & g_2(3) & g_2(4) & g_2(5) \\ & g_2(1) & g_2(2) & g_2(3) & g_2(4) \\ & & g_2(1) & g_2(2) & g_2(3) \\ & & & g_2(1) & g_2(2) \\ & & & & g_2(1) \\ g_3(1) & g_3(2) & g_3(3) & g_3(4) & g_3(5) \\ & g_3(1) & g_3(2) & g_3(3) & g_3(4) \\ & & g_3(1) & g_3(2) & g_3(3) \\ & & & g_3(1) & g_3(2) \\ & & & & g_3(1) \\ g_4(1) & g_4(2) & g_4(3) & g_4(4) & g_4(5) \\ & g_4(1) & g_4(2) & g_4(3) & g_4(4) \\ & & g_4(1) & g_4(2) & g_4(3) \\ & & & g_4(1) & g_4(2) \\ & & & & g_4(1) \end{bmatrix}. \tag{6.2}$$

Denote by $G_{i,j}, H_{i,j}$ the Givens and hyperbolic rotation of order 20, respectively, where $G_{i,j}, H_{i,j}$ are the identity matrices except for the following four entries:

$$\begin{aligned} &G_{i,j}(i,i) = G_{i,j}(j,j) = c;\ G_{i,j}(i,j) = s;\ G_{i,j}(j,i) = -s; \\ &H_{i,j}(i,i) = H_{i,j}(j,j) = c;\ H_{i,j}(i,j) = -s;\ H_{i,j}(j,i) = -s. \end{aligned}$$

The following matrices are J-orthogonal:

$$\begin{array}{ll} G_{i,j}, & 1 \le i,j \le 10, \text{ or } \ 11 \le i,j \le 20, \\ H_{i,j}, & 1 \le i \le 10, \text{ and } \ 11 \le j \le 20. \end{array}$$

Now we describe briefly how to annihilate the matrices $L(g_i)^T,\ i = 2,\dots,4$, to obtain R. At the first step, we consider the Givens rotation $G_{1,6}$ chosen to annihilate the entries $g_2(1)$ of L. We can construct the Givens matrices $G_{i,i+5},\ i = 2,\dots,5$, such that the multiplication of these matrices with L annihilates the diagonal elements of the matrix $L(g_2)^T$ without any additional computation, because of the block Toeplitz structure of L. We remark that these matrices are J-orthogonal. Let $Q_{1,2} = \prod_{i=1}^{5} G_{i,i+5}$. Then

$$Q_{1,2}L^T = \begin{bmatrix} \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) & \tilde{g}_1(4) & \tilde{g}_1(5) \\ & \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) & \tilde{g}_1(4) \\ & & \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) \\ & & & \tilde{g}_1(1) & \tilde{g}_1(2) \\ & & & & \tilde{g}_1(1) \\ 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) & \tilde{g}_2(5) \\ & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) \\ & & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) \\ & & & 0 & \tilde{g}_2(2) \\ & & & & 0 \\ g_3(1) & g_3(2) & g_3(3) & g_3(4) & g_3(5) \\ & g_3(1) & g_3(2) & g_3(3) & g_3(4) \\ & & g_3(1) & g_3(2) & g_3(3) \\ & & & g_3(1) & g_3(2) \\ & & & & g_3(1) \\ g_4(1) & g_4(2) & g_4(3) & g_4(4) & g_4(5) \\ & g_4(1) & g_4(2) & g_4(3) & g_4(4) \\ & & g_4(1) & g_4(2) & g_4(3) \\ & & & g_4(1) & g_4(2) \\ & & & & g_4(1) \end{bmatrix}.$$

We call the multiplication of $Q_{1,2}$ times L an *update* between $L(g_1)$ and $L(g_2)$. Then we set $L := Q_{1,2}L$. We remark that the newly computed matrices $L(g_1)$ and $L(g_2)$ continue to have the Toeplitz structure. To compute this update it is sufficient to update the vectors g_1 and g_2 (instead of $L(g_1)$ and $L(g_2)$),

$$\begin{pmatrix} \tilde{g}_1^T \\ \tilde{g}_2^T \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} {g_1}^T \\ {g_2}^T \end{pmatrix},$$

where c and s are the same "Givens coefficients" of the matrix $G_{1,6}$. In the same way, we compute the Givens rotation $G_{11,16}$ chosen to annihilate the $(16,1)$ entry of the new L and the corresponding Givens matrices $G_{i+10,i+15},\ i = 2,\dots,5$. Let

$Q_{3,4} = \prod_{i=1}^{5} G_{i+10,i+15}$. Then

$$Q_{3,4}L^T = \begin{bmatrix} \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) & \tilde{g}_1(4) & \tilde{g}_1(5) \\ & \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) & \tilde{g}_1(4) \\ & & \tilde{g}_1(1) & \tilde{g}_1(2) & \tilde{g}_1(3) \\ & & & \tilde{g}_1(1) & \tilde{g}_1(2) \\ & & & & \tilde{g}_1(1) \\ 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) & \tilde{g}_2(5) \\ & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) \\ & & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) \\ & & & 0 & \tilde{g}_2(2) \\ & & & & 0 \\ \tilde{g}_3(1) & \tilde{g}_3(2) & \tilde{g}_3(3) & \tilde{g}_3(4) & \tilde{g}_3(5) \\ & \tilde{g}_3(1) & \tilde{g}_3(2) & \tilde{g}_3(3) & \tilde{g}_3(4) \\ & & \tilde{g}_3(1) & \tilde{g}_3(2) & \tilde{g}_3(3) \\ & & & \tilde{g}_3(1) & \tilde{g}_3(2) \\ & & & & \tilde{g}_3(1) \\ 0 & \tilde{g}_4(2) & \tilde{g}_4(3) & \tilde{g}_4(4) & \tilde{g}_4(5) \\ & 0 & \tilde{g}_4(2) & \tilde{g}_4(3) & \tilde{g}_4(4) \\ & & 0 & \tilde{g}_4(2) & \tilde{g}_4(3) \\ & & & 0 & \tilde{g}_4(2) \\ & & & & 0 \end{bmatrix}. \tag{6.3}$$

We call the multiplication of $Q_{3,4}$ times L an update between $L(g_3)$ and $L(g_4)$, and we define $L := Q_{3,4}L$. Also in this case, because of the Toeplitz structure of these matrices, in order to compute the new L it is sufficient to update the vectors g_3 and g_4 only, i.e., $\begin{pmatrix} \tilde{g}_3^T \\ \tilde{g}_4^T \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_3{}^T \\ g_4{}^T \end{pmatrix}$, where c and s are the same "Givens coefficients" of the matrix $G_{11,16}$. To complete an iteration we consider the hyperbolic rotation $H_{1,11}$ constructed to annihilate the $(11,1)$ entry of the matrix (6.3). Without any computation we also obtain from $H_{1,11}$ the associated hyperbolic rotations $H_{i,i+10},\ i = 2,\ldots,5$. Let $S_{1,3} = \prod_{i=1}^{5} H_{i,i+10}$. Then

$$S_{1,3}L^T = \begin{bmatrix} \hat{g}_1(1) & \hat{g}_1(2) & \hat{g}_1(3) & \hat{g}_1(4) & \hat{g}_1(5) \\ & \hat{g}_1(1) & \hat{g}_1(2) & \hat{g}_1(3) & \hat{g}_1(4) \\ & & \hat{g}_1(1) & \hat{g}_1(2) & \hat{g}_1(3) \\ & & & \hat{g}_1(1) & \hat{g}_1(2) \\ & & & & \hat{g}_1(1) \\ 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) & \tilde{g}_2(5) \\ & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) & \tilde{g}_2(4) \\ & & 0 & \tilde{g}_2(2) & \tilde{g}_2(3) \\ & & & 0 & \tilde{g}_2(2) \\ & & & & 0 \\ 0 & \hat{g}_3(2) & \hat{g}_3(3) & \hat{g}_3(4) & \hat{g}_3(5) \\ & 0 & \hat{g}_3(2) & \hat{g}_3(3) & \hat{g}_3(4) \\ & & 0 & \hat{g}_3(2) & \hat{g}_3(3) \\ & & & 0 & \hat{g}_3(2) \\ & & & & 0 \\ 0 & \tilde{g}_4(2) & \tilde{g}_4(3) & \tilde{g}_4(4) & \tilde{g}_4(5) \\ & 0 & \tilde{g}_4(2) & \tilde{g}_4(3) & \tilde{g}_4(4) \\ & & 0 & \tilde{g}_4(2) & \tilde{g}_4(3) \\ & & & 0 & \tilde{g}_4(2) \\ & & & & 0 \end{bmatrix}. \tag{6.4}$$

We observe that the matrix $S_{1,3}Q_{3,4}Q_{1,2}$ is J-orthogonal. We call the multiplication of $S_{1,3}$ times L a *downdate* between $L(g_1)$ and $L(g_3)$. Also in this case, since these matrices have a Toeplitz structure, it is sufficient to downdate the vectors g_1 and g_3 in order to compute the multiplication $S_{1,3}L$,

$$\begin{pmatrix} \hat{g}_1^T \\ \hat{g}_3^T \end{pmatrix} = \begin{pmatrix} c & -s \\ -s & c \end{pmatrix} \begin{pmatrix} \tilde{g}_1^T \\ \tilde{g}_3^T \end{pmatrix},$$

where c and s are the same "hyperbolic coefficients" of the matrix $H_{1,11}$. Then, the first row of (6.4) is the first row of R. To compute the other rows of R, we apply the previous procedure to the matrix $L(2:20, 2:5)$.

In general, $6n$ flops are required to update or downdate two vectors of length n. Hence $3(\delta-1)n^2$ flops are required to compute the triangularization of a symmetric positive definite matrix of order n and displacement rank δ.

7. Appendix B. Matlab-like program for the basic deconvolution problem. The algorithm is summarized in the function `FTriang. generator` is a Matlab function that, given the matrix M, computes the generator $g_i, i = 1\ldots,5$ of M^TM. The functions `givens` computes the coefficients of the involved Givens rotation. The variables t_1, t_2, t_3, t_4 are temporary variables like *temp* in which the partial product (3.10) is stored.

`function`$[R]$=`FTriang`(M, m, n)
$(g_1, g_2, g_3, g_4, g_5) =$ `generator`(M);
$temp = 1; mn1 = m + 2n - 1; mn2 = m + 2n - 2;$
% Initialization
$(c, s) =$ `givens`$(g_1(1), g_2(1))$;

$$\begin{pmatrix} g_1^T([1:n, m+n:mn1]) \\ g_2^T([1:n, m+n:mn1]) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_1^T([1:n, m+n:mn1]) \\ g_2^T([1:n, m+n:mn1]) \end{pmatrix}$$

$R(1, 1:mn1) = g_1^T;\ g_1^T(2:mn1) = g_1^T(1:mn2); g_1(m+n) = 0;$
% Phase 1
`for` $i = 2:m,$

$$(c, s) = \texttt{givens}(g_1(i), g_2(i));$$

$$\begin{pmatrix} g_1^T([i:n+i-1, m+n:mn1]) \\ g_2^T([i:n+i-1, m+n:mn1]) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_1^T([i:n+i-1, m+n:mn1]) \\ g_2^T([i:n+i-1, m+n:mn1]) \end{pmatrix}$$

$$t_3 = g_3(n+m); t_4 = g_4(n+m);$$
$(c, s) =$ `givens`$(g_1(i), g_3(i))$;

$$\begin{aligned} g_3^T([i+1:n+i-1, m+n:mn1]) = \ & -sg_1^T([i+1:n+i-1, m+n:mn1]) \\ & +cg_3^T([i+1:n+i-1, m+n:mn1]); \end{aligned}$$

$$g_1(m+n) = g_1(m+n) + s(t_3 - t_4)/c;$$
$$g_4(m+n) = -sg_1(m+n) + ct_4;$$
$$temp = temp \times c;$$
`if` $i < m,$
$$g_3(n+i) = temp \times g_3(n+i);$$
`end`
$$R(i, i:mn1) = g_1(i:mn1);$$
$$g_1(i+1:mn1) = g_1(i:mn2);\ g_1(m+n) = 0;$$
`end`
% Phase 2
`for` $i = m+1 : m+n-1,$
$(c, s) =$ `givens`$(g_1(i), g_2(i))$;

$$\begin{pmatrix} g_1^T(i:mn1) \\ g_2^T(i:mn1) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_1^T(i:mn1) \\ g_2^T(i:mn1) \end{pmatrix}$$

$$t_3 = g_3(n+m); t_4 = g_4(n+m);$$
$(c, s) =$ `givens`$(g_1(i), g_3(i))$;
$$g_3^T(i+1:mn1) = -sg_1^T(i+1:mn1) + cg_3^T(i+1:mn1);$$
$$g_1(m+n) = g_1(m+n) + s(t_3 - t_4)/c;$$
$$g_4(m+n) = -sg_1(m+n) + ct_4;$$
$$s = g_5(i)/g_1(i);$$
$$c_1 = \sqrt{g_1(i)^2 - g_5(i)^2};$$
$$c = c_1/g_1(i);$$
% $\rho = -s/c$

$g_1^T(i+1:m+n) = \left(g_1^T(i+1:m+n) - sg_5^T(i+1:m+n)\right)/c;$
$g_5^T(i+1:m+n) = -sg_1^T(i+1:m+n) + cg_5^T(i+1:m+n);$
$g_1(i) = c_1;$
$R(i, i:mn1) = g_1^T(i:mn1);\ g_1^T(i+1:mn1) = g_1^T(i:mn2);\ g_1(m+n) = 0;$
end
$g_4(n+m) = t_4; g_4(m+n+1:mn1) = g_3(m+n+1:mn1);$
% Phase 3
for $i = m+n:mn1,$
$(c, s) =$ **givens**$(g_1(i), g_2(i));$
$\begin{pmatrix} g_1^T(i:mn1) \\ g_2^T(i:mn1) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_1^T(i:mn1) \\ g_2^T(i:mn1) \end{pmatrix}$
$(c, s) =$ **givens**$(g_1(i), g_3(i));$
$\begin{pmatrix} g_1^T(i:mn1) \\ g_3^T(i:mn1) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_1^T(i:mn1) \\ g_3^T(i:mn1) \end{pmatrix}$
$(c, s) =$ **givens**$(g_5(i), g_4(i));$
$\begin{pmatrix} g_5^T(i:mn1) \\ g_4^T(i:mn1) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} g_5^T(i:mn1) \\ g_4^T(i:mn1) \end{pmatrix}$
$s = g_5(i)/g_1(i);$
$c_1 = \sqrt{g_1(i)^2 - g_5(i)^2};$
$c = c_1/g_1(i);$
% $\rho = -s/c$
$g_1^T(i+1:m+n) = \left(g_1^T(i+1:m+n) - sg_5^T(i+1:m+n)\right)/c;$
$g_5^T(i+1:m+n) = -sg_1^T(i+1:m+n) + cg_5^T(i+1:m+n);$
$g_1(i) = c_1;$
$R(i, i:mn1) = g_1^T(i:mn1);\ g_1^T(i+1:mn1) = g_1^T(i:mn2);$
end

Acknowledgment. The first author would like to acknowledge the hospitality of the Dept. Elektrotechniek (ESAT), Katholieke Universiteit Leuven, where this work took place.

REFERENCES

[1] T. J. Abatzoglou and J. M. Mendel, *Constrained total least squares*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, 1987, pp. 1485–1488.

[2] T. J. Abatzoglou, J. M. Mendel, and G. A. Harada, *The constrained total least squares technique and its applications to harmonic superresolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1086.

[3] M. T. Bajen, R. Puchal, A. Gonzalez, J. M. Grinyo, A. Castelao, J. Mora, and J. Martin Comin, *MAG3 renogram deconvolution in kidney transplantation: Utility of the measurement of initial tracer uptake*, J. Nucl. Med., 38 (1997), pp. 1295–1299.

[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. Stat. Comput., 8 (1987), pp. 210–221.

[5] A. Capderou, D. Douguet, T. Similowski, A. Aurengo, and M. Zelter, *Non-invasive assessment of technetium-99m albumin transit time distribution in the pulmonary circulation by first-pass angiocardiography*, Eur. J. Nucl. Med., 24 (1997), pp. 745–753.

[6] S. Chandrasekaran and Ali H. Sayed, *Stabilizing the generalized Schur algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 950–983.

[7] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3004–3113.

[8] J. Chun, T. Kailath, and H. Lev–ari, *Fast parallel algorithms for QR and triangular factorization*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 899–913.

[9] J. S. Fleming and B. A. Goddard, *A technique for the deconvolution of the renogram*, Phys. Med. Biol., 19 (1974), pp. 546–549.

[10] J. S. Fleming, *Measurement of Hippuran plasma clearance using a gamma camera*, Phys. Med.

Biol., 22 (1977), pp. 526–530.
[11] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., The John Hopkins University Press, Baltimore, MD, 1996.
[12] R. Howman Giles, A. Moase, K. Gaskin, and R. Uren, *Hepatobiliary scintigraphy in a pediatric population: Determination of hepatic extraction fraction by deconvolution analysis*, J. Nucl. Med., 34 (1993), pp. 214–221.
[13] T. Kailath and J. Chun, *Generalized displacement structure for block–Toeplitz, Toeplitz–block, and Toeplitz–derived matrices,* SIAM J. Matrix Anal. Appl., 15 (1994), pp. 114–128.
[14] T. Kailath, S. Kung, and M. Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.
[15] S. Y. Kung and K. S. Arun, and D. V. Bhaskar Rao, *State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem*, J. Opt. Soc. Amer., 73 (1983), pp. 1799–1811.
[16] T. Kailath and Ali H. Sayed, *Displacement structure: Theory and applications*, SIAM Rev., 37 (1995), pp. 297–386.
[17] P. Lemmerling, I. Dologlou, and S. Van Huffel, *Speech compression based on exact modeling and structured total least norm optimization*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98), Vol. 1, Seattle, WA, 1998, pp. 353–356.
[18] P. Lemmerling, I. Dologlou, and S. Van Huffel, *Variable rate speech compression based on exact modeling and waveform vector quantization*, in Proceedings of the Signal Processing Symposium (SPS'98), IEEE Benelux Signal Processing Chapter, Leuven, Belgium, 1998, pp. 127–130.
[19] P. Lemmerling, S. Van Huffel, and B. De Moor, *Structured total least squares problems: Formulations, algorithms and applications*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 215–223.
[20] N. Mastronardi, P. Van Dooren, and S. Van Huffel, *On the stability of the generalized Schur algorithm*, ESAT-SISTA Report TR 99-82, ESAT Laboratory, Katholieke Universiteit Leuven, Belgium, 1999.
[21] J.G. Nagy, *Fast inverse QR factorization for Toeplitz matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 1174–1193.
[22] H. Park and L. Eldén, *Stability analysis and fast triangularization of Toeplitz matrices,* Numer. Math, 76 (1997), pp 383–402.
[23] J. J. Pedroso de Lima, *Nuclear medicine and mathematics*, Eur. J. Nucl. Med., 23 (1996), pp. 705–719.
[24] J. B. Rosen, H. Park, and J. Glick, *Total Least Norm Problem: Formulation and Algorithms*, Preprint 94–041, Army High Performance Computing Research Center, University of Minnesota, 1993, revised 1994.
[25] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.
[26] P. Stoica, R. L. Moses, B. Friedlander, and T. Söderström, *Maximum likelihood estimation of the parameters of multiple sinusoids from noisy measurements*, IEEE Trans. Acoust., Speech Signal Process., 37 (1989), pp. 378–391.
[27] M. Stewart and P. Van Dooren, *Stability issues in the factorization of structured matrices*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 104–118.
[28] S. Van Huffel, C. Decanniere, H. Chen, and P. Van Hecke, *Algorithm for time-domain NMR data fitting based on total least squares*, J. Magn. Reson., A110 (1994), pp. 228–237.
[29] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.
[30] S. Van Huffel, H. Park, and J. Ben Rosen, *Formulation and solution of structured total least norm problems for parameter estimation*, IEEE Trans. Signal Process., SP-44 (1996), pp. 2464–2474.
[31] S. Van Huffel, J. Vandewalle, M. Ch. De Roo, and J. L. Willems, *Reliable and efficient deconvolution technique based on total linear least squares for calculating the renal retention function*, Med. & Biol. Eng. & Comput., 25 (1987), pp. 26–33.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 554–568

MULTISPLITTING METHODS: OPTIMAL SCHEMES FOR THE UNKNOWNS IN A GIVEN OVERLAP*

R. E. WHITE†

Abstract. Consider a linear algebraic problem where the set of unknowns is a union of subsets. Let the coefficient matrix have a splitting associated with each subset. The traditional multisplitting method forms a weighted sum, over the overlapping unknowns, of the iterates for each such splitting to obtain a single parallel iterative method. An optimal alternative to the weighted sums will be presented. Convergence of this new form of multisplitting (MS) method can be studied for both symmetric positive definite (SPD) matrices and M-matrices. Applications to PDEs and the equilibrium equations for fluid flow in a driven cavity will be presented.

Key words. multisplitting, comparison, fluid flow

AMS subject classifications. 65F10, 65N20

PII. S0895479898334678

1. Introduction.

Consider a linear algebraic system

$$Au = b, \tag{1}$$

where A is an $n \times n$ matrix. Let $A = B_k - C_k$, where $k = 1, \ldots, K$ are K splittings of A with B_k nonsingular. Let D_k be diagonal matrices with nonnegative components such that $\Sigma D_k = I$. The summation will always be understood to be from $k = 1$ to $k = K$. O'Leary and White [5] introduced the following weighted multisplitting (MS) algorithm where the next value of u is denoted by u^+.

Weighted MS algorithm.

$$\begin{aligned} u^+ &= u + \left(\Sigma D_k {B_k}^{-1}\right) r(u), \qquad r(u) = b - Au \\ &= \left(\Sigma D_k {B_k}^{-1} C_k\right) u + \left(\Sigma D_k {B_k}^{-1}\right) b \\ &= Hu + Gb. \end{aligned}$$

Under appropriate conditions one can show that G is nonsingular so that upon defining $B = G^{-1}$ and $A = B - C, H$ will become $B^{-1}C$. If this new splitting is weak regular when A is an M-matrix (see [5]) or if it is P-regular when A is a symmetric positive definite (SPD) matrix (see White [9] or Benassi and White [1]), then the spectral radius of H will be less than 1 and the weighted MS algorithm will converge.

The purpose of the weighted MS algorithm is to be able to use multiprocessing computers, and therefore, the computations in the summations are to be done concurrently.

The diagonal matrices serve both as a "weighting" of the overlapping computations and as a "masking" of the nodes not associated with block k. The latter role ensures that the work done by each splitting will be small relative to the main problem with n unknowns.

*Received by the editors February 27, 1998; accepted for publication (in revised form) by M. Eiermann June 3, 1999; published electronically August 9, 2000.
http://www.siam.org/journals/simax/22-2/33467.html
†Department of Mathematics, Box 8205, North Carolina State University, Raleigh, NC 27695-8205 (white@math.ncsu.edu).

The purpose of this paper is to propose an optimal alternative to the diagonal matrices. Optimal can be interpreted in three ways. First, choose iteration matrix, H, so that the spectral radius of H is a minimum. Second, choose H so that the "residual"

$$R(u) = r(u)^T r(u) \tag{2}$$

is a minimum for $u = u^+$. The third case is for A being SPD ($A^T = A$, and $x^T Ax > 0$ for $x \neq 0$) so that $Au = b$ is equivalent to minimizing the quadratic form

$$J(u) = 1/2\ u^T Au - u^T b. \tag{3}$$

In this case we want to choose H so that $J(u^+)$ is a minimum.

In this paper we consider two variations on the weighted MS algorithm, the MinR MS and the MinJ MS algorithms. We shall study convergence of these when A is either an M-matrix or an SPD matrix. Section 2 contains the definitions and equivalent forms of these variations. Sections 3 and 4 have the convergence theorems for the SPD matrices and the M-matrices, respectively. In the last section we present an application to elliptic partial differential equations and to the driven cavity problem. In the driven cavity problem we use the new MS algorithm as a preconditioner in the restarted GMRES algorithm.

2. Optimal MS algorithms. Let $P_k \subset \{1, \ldots, n\}$ with $k = 0, 1, \ldots, K$ and with empty intersections. Let $S_k = P_0 \cup P_k$ for $k = 1, \ldots, k$ represent overlapping blocks of unknowns u_i, where $i \in S_k$. Associated with each block S_k is a splitting $A = B_k - C_k$, where B_k is nonsingular. In Figure 1 we let $K = 2$.

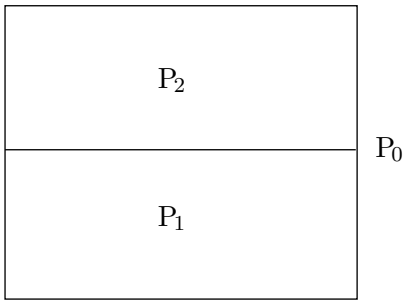


FIG. 1. *Blocks of nodes.*

Example. Consider a discretization of $-\Delta u + a_1 u_x + a_2 u_y + cu = f$ on $(0,1) \times (0,1)$. Let $K = 2$ and P_1 and P_2 be associated with the lower and upper blocks of $(0,1) \times (0,1)$.

$$A = \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix},$$

$$A_{ii} = M_i - N_i,$$

$$B_1 = \begin{bmatrix} M_0 & & \\ A_{10} & M_1 & \\ & & M_2 \end{bmatrix},$$

and

$$B_2 = \begin{bmatrix} M_0 & & \\ & M_1 & \\ A_{20} & & M_2 \end{bmatrix}.$$

In order to define the new version of the MS algorithm, we use the $n \times n_k$ matrix

$$E_k = \begin{bmatrix} 0 \dots 0 & I_k & 0 \dots 0 \end{bmatrix}^T,$$

where n_k is the cardinality of P_k and I_k is the $n_k \times n_k$ identify matrix. Define the new u for the block k to be

$$u_k{}^+ = E_k{}^T(u + B_k{}^{-1} r(u)).$$

Define the $N \times 1$ column vector, v, where $n = n_0 + N$ and $N = \Sigma n_k$,

$$v = \begin{bmatrix} u_1{}^{+T} \dots & u_K{}^{+T} \end{bmatrix}^T.$$

Finally, we must choose $u_0{}^+$ so that the $n \times 1$ vector, u^+, will be "optimal":

$$u^+ = \begin{bmatrix} u_0{}^{+T} v^T \end{bmatrix}^T.$$

Let A be written as a 2×2 block matrix where A_{00} is $n_0 \times n_0$ and Z_{11} is $N \times N$:

$$A = \begin{bmatrix} A_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{bmatrix}. \tag{4}$$

THEOREM 1. *Let A be an SPD matrix. Let $J(u)$ be the quadratic form* (3) *associated with the algebraic system* (1). *$u_0{}^+$ satisfies the inequality for all $n_0 \times 1$ vectors w,*

$$J\left(\begin{bmatrix} u_0{}^+ \\ v \end{bmatrix} \right) \le J\left(\begin{bmatrix} w \\ v \end{bmatrix} \right)$$

if and only if

$$A_{00}\, u_0{}^+ = b_0 - Z_{01} v,$$

where $b = [b_0{}^T b_1{}^T]^T$ with b_0 a $n_0 \times 1$ vector.

Proof. Let

$$\begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ v \end{bmatrix} + \begin{bmatrix} w \\ 0 \end{bmatrix}.$$

$$\begin{aligned} J\left(\begin{bmatrix} w \\ v \end{bmatrix} \right) &= 1/2 \begin{bmatrix} w \\ v \end{bmatrix}^T \begin{bmatrix} A_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} - \begin{bmatrix} w \\ v \end{bmatrix}^T \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ &= 1/2 \begin{bmatrix} 0 \\ v \end{bmatrix}^T \begin{bmatrix} A_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{bmatrix} \begin{bmatrix} 0 \\ v \end{bmatrix} + \begin{bmatrix} w \\ 0 \end{bmatrix}^T \begin{bmatrix} A_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{bmatrix} \begin{bmatrix} 0 \\ v \end{bmatrix} \\ &\quad + 1/2 \begin{bmatrix} w \\ 0 \end{bmatrix}^T \begin{bmatrix} A_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{bmatrix} \begin{bmatrix} w \\ 0 \end{bmatrix} - \begin{bmatrix} w \\ 0 \end{bmatrix}^T \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} - \begin{bmatrix} 0 \\ v \end{bmatrix}^T \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ &= (1/2 v^T Z_{11} v - v^T b_1) + (1/2 w^T A_{00} w - w^T (b_0 - Z_{01} v)). \end{aligned}$$

Since v is fixed, it suffices to minimize the second term. Because A is an SPD matrix, A_{00} is also an SPD matrix. Therefore, the second term is a minimum if and only if $w = u_0{}^+$ satisfies $A_{00} u_0{}^+ = b_0 - Z_{01} v$.

MinJ MS algorithm. Let $A = B_k - C_k$ be K splittings. Let $A = [A_{ij}]$ be represented by $(K+1)\times(K+1)$ blocks associated with the partition of the unknown nodes so that in (4), $Z_{01} = \begin{bmatrix} A_{01} \dots A_{0K} \end{bmatrix}$, $Z_{10} = \begin{bmatrix} {A_{10}}^T \dots {A_{K0}}^T \end{bmatrix}^T$, and Z_{11} is a $K \times K$ block matrix whose block components are A_{ij} with $i, j = 1, \dots, K$:

$$\begin{aligned} {u_k}^+ &= {E_k}^T(u + {B_k}^{-1} r(u)) && \text{(concurrent computations)},\\ A_{00}\, {u_0}^+ &= b_0 - \Sigma A_{0k} {u_k}^+ && \text{(serial computations)},\\ u^+ &= \begin{bmatrix} {u_0}^{+T} {u_1}^{+T} \dots {u_K}^{+T} \end{bmatrix}^T . \end{aligned}$$

If A is not an SPD matrix, then the linear algebraic problem (1) will not be equivalent to minimizing $J(u)$ in (3) so that the above ${u_0}^+$ is not applicable. An alternative is to minimize the residual $R(u)$ as defined in (2).

THEOREM 2. *Let A be represented by a 2×2 block as in* (4). *Assume the first block column $[{A_{00}}^T {Z_{10}}^T]^T$ has full column rank. ${u_0}^+$ satisfies the inequality for all $n_0 \times 1$ vectors w*

$$R\left(\begin{bmatrix} {u_0}^+ \\ v \end{bmatrix}\right) \le R\left(\begin{bmatrix} w \\ v \end{bmatrix}\right)$$

if and only if ${u_0}^+$ satisfies the normal equations

$$\begin{bmatrix} {A_{00}}^T & {Z_{10}}^T \end{bmatrix} \begin{bmatrix} A_{00} \\ Z_{10} \end{bmatrix} {u_0}^T = \begin{bmatrix} {A_{00}}^T & {Z_{10}}^T \end{bmatrix} \begin{bmatrix} b_0 - Z_{01} v \\ b_1 - Z_{11} v \end{bmatrix},$$

where $b = [{b_0}^T {b_1}^T]^T$ with b_0 a $n_0 \times 1$ vector.

Proof.

$$\begin{aligned} r\left(\begin{bmatrix} w \\ v \end{bmatrix}\right) &= b - A \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} b_0 - A_{00} w - Z_{01} v \\ b_1 - Z_{10} w - Z_{11} v \end{bmatrix} \\ &= \begin{bmatrix} b_0 - Z_{01} v \\ b_1 - Z_{11} v \end{bmatrix} - \begin{bmatrix} A_{00} \\ Z_{10} \end{bmatrix} w. \end{aligned}$$

Since v is fixed, it suffices to apply the normal equations to

$$\begin{bmatrix} A_{00} \\ Z_{10} \end{bmatrix} w = \begin{bmatrix} b_0 - Z_{01} v \\ b_1 - Z_{11} v \end{bmatrix}.$$

MinR MS algorithm. Let $A = B_k - C_k$ be K splittings. Let $A = [A_{ij}]$ be represented by $(K+1)\times(K+1)$ blocks associated with the partition of the unknown nodes so that in (4), $Z_{01} = \begin{bmatrix} A_{01} \dots A_{0K} \end{bmatrix}$, $Z_{10} = \begin{bmatrix} {A_{10}}^T \dots {A_{K0}}^T \end{bmatrix}^T$, Z_{11} is a $K \times K$ block matrix whose block components are A_{ij} with $i, j = 1, \dots, K$, and $b = [{b_0}^T {b_1}^T \dots {b_K}^T]^T$:

$$\begin{aligned} {u_k}^+ &= {E_k}^T(u + {B_k}^{-1} r(u)) && \text{(concurrent computations)},\\ ({A_{00}}^T A_{00} + \Sigma {A_{k0}}^T A_{k0}) {u_0}^+ &= {A_{00}}^T (b_0 - \Sigma A_{0k} {u_k}^+) + \Sigma {A_{i0}}^T (b_k - \Sigma A_{ik} {u_k}^+) \\ & && \text{(serial computations)},\\ u^+ &= \begin{bmatrix} {u_0}^{+T} {u_1}^{+T} \dots {u_K}^{+T} \end{bmatrix}^T . \end{aligned}$$

The serial computations in the MinR MS algorithm are more elaborate than the serial computations in the MinJ MS algorithm. For most of this paper we will focus on the MinJ MS algorithm. Even though it was formulated for the SPD matrices, one can establish convergence results for the M-matrix case that may not be SPD.

3. SPD matrices and the MinJ MS algorithm. Before presenting several examples we will find it convenient to use the following representation lemma for the MinJ MS algorithm.

REPRESENTATION LEMMA. *Let $A = B_k - C_k$ be K splittings and assume B_k and A_{00} are nonsingular. The MinJ MS algorithm may be represented by*

$$u^+ = u + \mathrm{Gr}(u),$$

where

$$G = \begin{bmatrix} {A_{00}}^{-1}{E_0}^T - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1} \\ {E_1}^T{B_1}^{-1} \\ \vdots \\ {E_K}^T{B_K}^{-1} \end{bmatrix}.$$

If H is defined via $GA = I - H$, then $u^+ = Hu + Gb$ and the MinJ algorithm will converge if H has a spectral radius more than 1. Moreover, if G is nonsingular, then the MinJ MS algorithm may be represented by a single splitting $A = B - C$, where $B = G^{-1}$.

Proof. For $k > 0$ we have ${u_k}^+ = {E_k}^T(u + {B_k}^{-1}r(u))$. For $k = 0$, ${u_0}^+$ is defined by

$$\begin{aligned} A_{00}{u_0}^+ &= b_0 - \Sigma A_{0k}{u_k}^+ \\ &= b_0 - \Sigma A_{0k}{E_k}^T(u + {B_k}^{-1}r(u)) - A_{00}u_0 + A_{00}u_0 \\ &= {E_0}^T r(u) - \Sigma A_{0k}{E_k}^T{B_k}^{-1}r(u) + A_{00}u_0. \\ {u_0}^+ &= ({A_{00}}^{-1}{E_0}^T - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1})r(u) + u_0. \end{aligned}$$

Since for $k > 0$, ${E_k}^T u = u_k$, we have

$$u^+ = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_K \end{bmatrix} + \begin{bmatrix} {A_{00}}^{-1}{E_0}^T - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1} \\ {E_1}^T{B_1}^{-1} \\ \vdots \\ {E_K}^T{B_K}^{-1} \end{bmatrix} r(u).$$

Let A be an SPD matrix, and therefore, its diagonal blocks are SPD. O'Leary and White [5] observed that if all the MSs were P-regular ($A = B - C$ symmetric with B nonsingular and $B^T + C$ positive definite), then the weighted MS algorithm may or may not converge. A modification of this example shows this is also the case for the MinJ MS algorithm.

Example. This example illustrates the importance of assuming appropriate structure of the MSs so as to be able to establish convergence of the MinJ MS algorithm.

The appropriate structure was discussed in White [9] for the weighted MS algorithm:

$$A = \begin{bmatrix} 1 & & \\ & 3/4 & \\ & & 3/4 \end{bmatrix} \text{ with } A_{00} = [1] \text{ and } A_{11} = \begin{bmatrix} 3/4 & \\ & 3/4 \end{bmatrix};$$

$$B_1 = \begin{bmatrix} 1 & & \\ & 4 & 1 \\ & -1 & 1/2 \end{bmatrix} \text{ and } B_2 = \begin{bmatrix} 1 & & \\ & 1/2 & -1 \\ & 1 & 4 \end{bmatrix} \text{ with}$$

$$E_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \text{ and } E_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T.$$

By the Representation Lemma

$$G = \begin{bmatrix} 1 & 0 & 0 \\ & {E_1}^T {B_1}^{-1} & \\ & {E_2}^T {B_2}^{-1} & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/6 & -1/3 \\ 0 & -1/3 & 1/6 \end{bmatrix}.$$

H is defined via $GA = I - H$ so that

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 7/8 & 1/4 \\ 0 & 1/4 & 7/8 \end{bmatrix}.$$

The eigenvalues of H are 0 and $\frac{7}{8} \pm \frac{1}{4}$, and therefore, the MinJ MS algorithm will not converge for this choice of B_k and E_k. However, If B_1 and B_2 are interchanged, then the new iteration matrix will be

$$\hat{H} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1/4 \\ 0 & -1/4 & 0 \end{bmatrix}.$$

This iteration matrix has eigenvalues equal to 0 and $\pm\frac{1}{4}$, and so, the MinJ MS will converge.

In order to impose some structure on A and the MSs and to keep the notation from becoming a burden, we let $K = 2$. Results from $K > 2$ are straightforward and may be formulated by decomposing the last diagonal block. We will consider two cases, the block diagonal MinJ MS and the incomplete block Gauss–Seidel MinJ MS.

Block diagonal MinJ MS. Let $A = B_k - C_k$, where for $k > 0, A_{kk} = M_k - N_k$.

$$B_k = \begin{bmatrix} A_{00} & & \\ & M_1 & \\ & & M_2 \end{bmatrix} \text{ for both } k = 1 \text{ and } k = 2,$$

$${E_1}^T {B_1}^{-1} = \begin{bmatrix} 0 & {M_1}^{-1} & 0 \end{bmatrix},$$

$${E_2}^T {B_2}^{-1} = \begin{bmatrix} 0 & 0 & {M_2}^{-1} \end{bmatrix}.$$

By the Representation Lemma

$$G = \begin{bmatrix} \begin{bmatrix} {A_{00}}^{-1} & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & {A_{00}}^{-1}A_{01}{M_1}^{-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & {A_{00}}^{-1}A_{02}{M_2}^{-1} \end{bmatrix} \\ \begin{bmatrix} 0 & {M_1}^{-1} & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & {M_2}^{-1} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} {A_{00}}^{-1} & -{A_{00}}^{-1}A_{01}{M_1}^{-1} & -{A_{00}}^{-1}A_{02}{M_2}^{-1} \\ 0 & {M_1}^{-1} & 0 \\ 0 & 0 & {M_2}^{-1} \end{bmatrix}.$$

Example. This example illustrates a MinJ MS algorithm that converges faster than a corresponding weighted MS algorithm:

$$A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix} \text{ with } B_1 = B_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The weighting matrices for the weighted MS algorithm will be

$$D_1 = \begin{bmatrix} 1/2 & & \\ & 1 & \\ & & 0 \end{bmatrix} \text{ and } D_2 = \begin{bmatrix} 1/2 & & \\ & 0 & \\ & & 1 \end{bmatrix}.$$

The G matrix for the weighted MS algorithm will be

$$G_w = D_1{B_1}^{-1} + D_2B_2^{-1} = \begin{bmatrix} 1/2 & & \\ & 1/2 & \\ & & 1/2 \end{bmatrix}.$$

The H matrix for the weighted MS algorithm will be

$$H_w = I - G_wA = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of H_w are $\pm\frac{1}{\sqrt{2}}$ and 0.

The G matrix for the block diagonal MinJ MS algorithm is

$$G = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix}.$$

The H matrix for the MinJ MS algorithm is

$$H = I - GA = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of H are $\frac{1}{2}, 0$ and 0. Therefore, for these same splittings the MinJ MS will converge faster than the weighted MS algorithm.

THEOREM 3. *Consider the block diagonal MinJ MS algorithm, where $K = 2$. If A is SPD and*

$$\begin{bmatrix} {M_1}^T + N_1 & -A_{12} \\ -A_{21} & {M_2}^T + N_2 \end{bmatrix}$$

is positive definite, then the block diagonal MinJ MS algorithm will converge to the solution of (1).

Proof. We shall show that the algorithm is given by a single P-regular splitting. Note for the block diagonal MinJ MS algorithm that G is nonsingular and a block elementary matrix

$$G = B^{-1} \text{ and } B = \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ & M_1 & \\ & & M_2 \end{bmatrix}.$$

Thus, $A = B - C$ and

$$C = -A + B = -\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} + \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ & M_1 & \\ & & M_2 \end{bmatrix}.$$

Because A is symmetric and $A_{kk} = M_k - N_k$ for $k > 0$,

$$C = \begin{bmatrix} 0 & 0 & 0 \\ -A_{10} & N_1 & -A_{12} \\ -A_{20} & -A_{21} & N_2 \end{bmatrix}$$

and

$$B^T + C = \begin{bmatrix} A_{00} & 0 & 0 \\ 0 & {M_1}^T + N_1 & -A_{12} \\ 0 & -A_{21} & {M_2}^T + N_2 \end{bmatrix}.$$

Since A is SPD, A_{00} is SPD. Therefore, $B^T + C$ will be positive definite if the assumption is true.

Another more complicated MS is the following incomplete block Gauss–Seidel MS. Here the word incomplete is used because we delete some of the blocks in the lower triangular part of A so as to obtain concurrent calculations. For the MinJ MS this will possibly generate a symmetric G, but the iteration matrix, H, for this scheme will have the same eigenvalues as the iteration matrix for the block diagonal MS algorithm!

Incomplete block Gauss–Seidel MinJ MS.

$$B_1 = \begin{bmatrix} A_{00} & 0 & 0 \\ A_{10} & M_1 & 0 \\ 0 & 0 & M_2 \end{bmatrix} \text{ and } B_2 = \begin{bmatrix} A_{00} & 0 & 0 \\ 0 & M_1 & 0 \\ A_{20} & 0 & M_2 \end{bmatrix},$$

$${E_1}^T {B_1}^{-1} = \begin{bmatrix} -{M_1}^{-1} A_{10} {A_{00}}^{-1} & {M_1}^{-1} & 0 \end{bmatrix},$$

$${E_2}^T B_2^{-1} = \begin{bmatrix} -{M_2}^{-1} A_{20} {A_{00}}^{-1} & 0 & {M_2}^{-1} \end{bmatrix}.$$

By the Representation Lemma

$$G = \begin{bmatrix} \begin{bmatrix} {A_{00}}^{-1} & 0 & 0 \end{bmatrix} - {A_{00}}^{-1}A_{01}{E_1}^T{B_1}^{-1} - {A_{00}}^{-1}A_{02}{E_2}^T{B_2}^{-1} \\ {E_1}^T B_1^{-1} \\ {E_2}^T B_2^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} X & -{A_{00}}^{-1}A_{01}{M_1}^{-1} & -{A_{00}}^{-1}A_{02}{M_2}^{-1} \\ -{M_1}^{-1}A_{10}{A_{00}}^{-1} & {M_1}^{-1} & 0 \\ -{M_2}^{-1}A_{20}{A_{00}}^{-1} & 0 & {M_2}^{-1} \end{bmatrix},$$

where

$$X = {A_{00}}^{-1} + {A_{00}}^{-1}A_{01}{M_1}^{-1}A_{10}{A_{00}}^{-1} + {A_{00}}^{-1}A_{02}{M_2}^{-1}A_{20}{A_{00}}^{-1}.$$

A more compact way of writing the both block MinJ MSs for A is as follows, where A_{00} and M are nonsingular:

$$Z = \begin{bmatrix} A_{01} & A_{02} \end{bmatrix} \text{ and } W = \begin{bmatrix} {A_{10}}^T & {A_{20}}^T \end{bmatrix},$$

$$M = \begin{bmatrix} M_1 & \\ & M_2 \end{bmatrix} \text{ and } N = \begin{bmatrix} N_1 & -A_{12} \\ -A_{21} & N_2 \end{bmatrix},$$

$$A = \begin{bmatrix} A_{00} & Z \\ W^T & M - N \end{bmatrix}.$$

For the block diagonal MinJ MS we have

$$G_d = {B_d}^{-1} = \begin{bmatrix} {A_{00}}^{-1} & -{A_{00}}^{-1}ZM^{-1} \\ 0 & M^{-1} \end{bmatrix}.$$

For the incomplete block Gauss–Seidel MinJ MS we have

$$G_{\text{igs}} = {B_{\text{igs}}}^{-1} = \begin{bmatrix} {A_{00}}^{-1} + {A_{00}}^{-1}ZM^{-1}W^T{A_{00}}^{-1} & -{A_{00}}^{-1}ZM^{-1} \\ -M^{-1}W^T{A_{00}}^{-1} & M^{-1} \end{bmatrix}.$$

Note, if M is symmetric and $W = Z$, then G_{igs} will be symmetric.

THEOREM 4. *Let A be possibly nonsymmetric with nonsingular A_{00} and M. The iteration matrices for the block diagonal MinJ MS, H_d, and for the incomplete block Gauss–Seidel MinJ MS, H_{igs}, are*

$$H_d = \begin{bmatrix} {A_{00}}^{-1}ZM^{-1}W^T & -{A_{00}}^{-1}ZM^{-1}N \\ -M^{-1}W^T & M^{-1}N \end{bmatrix}$$

and

$$H_{\text{igs}} = \begin{bmatrix} 0 & -{A_{00}}^{-1}Z(M^{-1}N + M^{-1}W^T{A_{00}}^{-1}Z) \\ 0 & M^{-1}N + M^{-1}W^T{A_{00}}^{-1}Z \end{bmatrix}.$$

Moreover, H_d and H_{igs} have the same eigenvalues.

Proof. The definition of the iteration matrix is given by the formula $GA = I - H$. Thus, $H_d = I - G_dA$ and $H_{\text{igs}} = I - G_{\text{igs}}A$, and the formulas follow by block matrix products.

By the block row operations we may write H_d as a product

$$H_d = UL = \begin{bmatrix} I & -{A_{00}}^{-1}Z \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -M^{-1}W^T & M^{-1}N \end{bmatrix}.$$

Then H_d is similar to LU and

$$U^{-1}H_dU = LU = \begin{bmatrix} 0 & 0 \\ -M^{-1}W^T & M^{-1}W^T{A_{00}}^{-1}Z + M^{-1}N \end{bmatrix}.$$

Since both H_{igs} and $U^{-1}H_dU$ have the same nonzero diagonal block and have a block row or column equal to zero, they must have the same eigenvalues.

4. M-matrices and the MinJ MS algorithm. Even though the MinJ MS was designed for the SPD case, it will also work for the M-matrix case. But the optimal sense will not be realized because the algebraic problem (1) is not equivalent to the quadratic minimization problem (3). When the MSs are weak regular ($A = B - C$ with $B^{-1} \geq 0$ and $B^{-1}C \geq 0$), we establish convergence similar to O'Leary and White [5]. When the MSs are regular ($A = B - C$ with $B^{-1} \geq 0$ and $C \geq 0$), we establish comparison results similar to Woznicki [10], and Csordas and Varga [2], and for MSs Elsner [3] and Marek and Szyld [4].

THEOREM 5. *Let A be a nonsingular M-matrix ($A^{-1} \geq 0$ and A has nonpositive off-diagonal components). If $A = B_k - C_k$ for $k = 1, \ldots, K$ are weak regular splittings, then $G, H \geq 0$, where G is given by the Representation Lemma for MinJ MS and $H = I - GA$. Moreover, the following are true and equivalent:*

1. *G is nonsingular,*
2. *$A = B - C$, where $B = G^{-1}$ is a weak regular splitting and represents the MinJ MS, and*
3. *MinJ MS algorithm is convergent.*

Proof. Since A is a nonsingular M-matrix, A_{00} is also a nonsingular M-matrix. Thus, ${A_{00}}^{-1} \geq 0$ and for $k > 0, -A_{0k} \geq 0$. Since each splitting is weak regular, both ${B_k}^{-1}$ and ${B_k}^{-1}C_k \geq 0$. Simply apply the Representation Lemma to obtain

$$G = \begin{bmatrix} {A_{00}}^{-1}{E_0}^T - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1} \\ {E_1}^T{B_1}^{-1} \\ \vdots \\ {E_K}^T{B_K}^{-1} \end{bmatrix} \geq 0.$$

In order to show $H \geq 0$, use $H = I - GA$ and the Representation Lemma:

$$H = I - \begin{bmatrix} {A_{00}}^{-1}{E_0}^T - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1} \\ {E_1}^T{B_1}^{-1} \\ \vdots \\ {E_K}^T{B_K}^{-1} \end{bmatrix} A$$

$$= I - \begin{bmatrix} {A_{00}}^{-1}{E_0}^T A - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1}A \\ {E_1}^T{B_1}^{-1}A \\ \vdots \\ {E_K}^T{B_K}^{-1}A \end{bmatrix}.$$

${E_0}^T A = \begin{bmatrix} A_{00} & A_{01} & \dots & A_{0K} \end{bmatrix}$. Since $A = B_k - C_k, {B_k}^{-1}$, $A = I - {B_k}^{-1}C_k$.

$$H = I - \begin{bmatrix} {A_{00}}^{-1}\begin{bmatrix} A_{00} & A_{01} & \dots & A_{0K} \end{bmatrix} - \Sigma {A_{00}}^{-1}A_{0k}{E_k}^T(I - {B_k}^{-1}C_k) \\ {E_1}^T(I - {B_1}^{-1}C_1) \\ \vdots \\ {E_K}^T(I - {B_K}^{-1}C_K) \end{bmatrix}$$

$$= \begin{bmatrix} -\Sigma {A_{00}}^{-1}A_{0k}{E_k}^T{B_k}^{-1}C_k \\ {E_1}^T{B_1}^{-1}C_1 \\ \vdots \\ {E_K}^T{B_K}^{-1}C_K \end{bmatrix} \geq 0.$$

The proof of the truth and equivalence of statements 1, 2, and 3 follows from Theorem 1 in White [8]. This theorem is applicable because A^{-1}, G, and $H \geq 0$. It suffices to show that each row of G has some nonzero component. The zero block row of G is given by

$${E_0}^T G = \begin{bmatrix} {A_{00}}^{-1} - {A_{00}}^{-1}A_{01}{E_1}^T{B_1}^{-1} \dots - {A_{00}}^{-1}A_{0K}{E_K}^T{B_K}^{-1} \end{bmatrix}.$$

Since ${A_{00}}^{-1}, -A_{0k}$, ${B_k}^{-1} \geq 0$ for $k > 0, {E_0}^T G \geq \begin{bmatrix} {A_{00}}^{-1} & 0 & \dots & 0 \end{bmatrix}$. Because A is a nonsingular M-matrix, A_{00} is also and must be nonsingular. Therefore, each row of ${A_{00}}^{-1}$ must have a nonzero component. Because of the matrix inequality ${E_0}^T G \geq \begin{bmatrix} {A_{00}}^{-1} & 0 & \dots & 0 \end{bmatrix}$, each row in the zero block row ${E_0}^T G$ must have a nonzero component. The k block row of G is ${E_k}^T G = {E_k}^T {B_k}^{-1}$. Since B_k are nonsingular, ${B_k}^{-1}$ are nonsingular and each of its rows has a nonzero component and so does each row of ${E_k}^T {B_k}^{-1}$.

The classical comparison result for iterative methods may be stated as follows: If $A^{-1} \geq 0$ and $A = B - C = B' - C'$ are two regular splittings such that $B^{-1} \geq {B'}^{-1}$, then $\rho({B'}^{-1}C') \geq \rho(B^{-1}C)$. In Elsner [3], it was proved that if one of these splittings is weak regular and the other is regular, then the conclusion will also be true. The next theorem is for the MinJ MS algorithm in which two sets of MSs satisfy the inequality ${B_k}^{-1} \geq {B'}_k{}^{-1}$.

THEOREM 6. *Let A be a nonsingular M-matrix. Let $A = B_k - C_k = {B'}_k - {C'}_k$ be weak regular splittings. Consider the two MinJ MS algorithms where $A = B - C = B' - C'$ are the single splittings associated with the two MinJ MS algorithms and assume either C or $C' \geq 0$. If ${B_k}^{-1} \geq {B'}_k{}^{-1}$, then*

1. $B^{-1} = G \geq G' = {B'}^{-1}$, *and*
2. $\rho({B'}^{-1}C') \geq \rho(B^{-1}C)$.

Proof. The first conclusion follows directly from the Representation Lemma for the MinJ MS algorithm. The second conclusion is just an application of Elsner's comparison result for weak regular splittings and of Theorem 5 of this paper. The additional assumption that C or $C' \geq 0$ implies that one of the splittings $A = B - C = B' - C'$ is a regular splitting.

COROLLARY. *Let A be a nonsingular M-matrix. Consider either the block diagonal or block incomplete Gauss–Seidel versions of the MinJ MS algorithm. If $A_{kk} = M_k - N_k = {M'}_k - {N'}_k$ are regular splittings, then $A = B_k - C_k = {B'}_k - {C'}_k$ are regular splittings. Also, if ${M_k}^{-1} \geq {M'}_k{}^{-1}$, then ${B_k}^{-1} \geq {B'}_k{}^{-1}$ and $C, C' \geq 0$ so that the conclusions of the theorem must be true.*

Proof. Because of the special structure of these two cases, it is easy to see that $A = B_k - C_k = {B'}_k - {C'}_k$ are regular splittings and that ${B_k}^{-1} \geq {B'}_k{}^{-1}$. In order to show $C, C' \geq 0$, we consider $C = -A + B$ for the two cases. The block diagonal case follows from

$$C = -A + B = -\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} + \begin{bmatrix} A_{00} & A_{01} & A_{02} \\ & M_1 & \\ & & M_2 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 0 \\ -A_{10} & N_1 & -A_{12} \\ -A_{20} & -A_{21} & N_2 \end{bmatrix}.$$

By the assumptions, each of the blocks in C are nonnegative.

In order to show that C is nonnegative for the block incomplete Gauss–Seidel case, we use the compact notation for A which was introduced before Theorem 4:

$$Z = \begin{bmatrix} A_{01} & \dots & A_{0K} \end{bmatrix} \text{ and } W = \begin{bmatrix} {A_{10}}^T & \dots & {A_{K0}}^T \end{bmatrix},$$
$$C = -A + B$$
$$= -\begin{bmatrix} A_{00} & Z \\ W^T & M - N \end{bmatrix} + \begin{bmatrix} {A_{00}}^{-1} + {A_{00}}^{-1} Z M^{-1} W^T {A_{00}}^{-1} & -{A_{00}}^{-1} Z M^{-1} \\ -M^{-1} W^T {A_{00}}^{-1} & M^{-1} \end{bmatrix}^{-1}.$$

B^{-1} can be factored, and so one can compute its inverse:

$$B^{-1} = \begin{bmatrix} {A_{00}}^{-1} + {A_{00}}^{-1} Z M^{-1} W^T {A_{00}}^{-1} & -{A_{00}}^{-1} Z M^{-1} \\ -M^{-1} W^T {A_{00}}^{-1} & M^{-1} \end{bmatrix}$$
$$= \begin{bmatrix} I & -{A_{00}}^{-1} Z \\ 0 & I \end{bmatrix} \begin{bmatrix} {A_{00}}^{-1} & 0 \\ -M^{-1} W^T {A_{00}}^{-1} & M^{-1} \end{bmatrix},$$
$$B = \begin{bmatrix} {A_{00}}^{-1} & 0 \\ -M^{-1} W^T {A_{00}}^{-1} & M^{-1} \end{bmatrix}^{-1} \begin{bmatrix} I & -{A_{00}}^{-1} Z \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} A_{00} & 0 \\ W^T & M \end{bmatrix} \begin{bmatrix} I & {A_{00}}^{-1} Z \\ 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} A_{00} & Z \\ W^T & M + W^T {A_{00}}^{-1} Z \end{bmatrix}.$$

So, C is easily computed:

$$C = \begin{bmatrix} 0 & 0 \\ 0 & N + W^T {A_{00}}^{-1} Z \end{bmatrix} \geq 0.$$

It must be nonnegative because $N, -W, -Z$, and ${A_{00}}^{-1}$ are nonnegative.

5. Applications. We discuss two applications to partial differential equations. The first is the elliptic problem mentioned in section 2. The second is the Stokes problem associated with the driven cavity problem. For both applications, $K = 2$ and the implementations were done in MATLAB so as to illustrate the qualitative properties of these MS algorithms. No numerical studies are given that attempt to optimize the algorithms for a particular computer architecture.

Elliptic partial differential equation. Consider a steady state problem with given boundary value and two variables

$$-\Delta u + a_1 u_x + a_2 u_y + cu = f \text{ on } (0,1) \times (0,1).$$

Suppose there are N unknowns in the x-direction and $2N+1$ unknowns in the y-direction. Thus, the total unknowns is $n = N(2N+1)$. Use the classical order starting with the bottom grid row and going from left to right and from the bottom to top. Consider the following partition:

$$P_0 = \{N^2+1, \ldots, N^2+N\},$$
$$P_1 = \{1, \ldots, N^2\},$$

and

$$P_2 = \{N^2+N+1, \ldots, 2N^2+N\}.$$

Use centered finite difference for second-order derivatives and upwind finite differences for the first-order derivatives. This will generate an SPD coefficient matrix if there are no first-order terms, or an M-matrix if there are first-order terms (see Proposition 2.4.14 in [6]).

In the computations in Table 1 we used $c = 1, N = 5$, and variables a_1 and a_2. The calculation for the weighted MS used the Gauss–Seidel splitting $M_k = \text{tril}(A_{kk})$ = the lower triangular part of A_{kk} with $k = 0, 1, 2$.

TABLE 1
Spectral radii and multisplittings.

Different MS methods	Spectral radius of H $a_1 = 10, a_2 = 1$	Spectral radius of H $a_1 = 0, a_2 = 0$
Weighted MS	.6428	.7393
Block diagonal MinJ with $M_k = \text{tril}(A_{kk})$	.5971	.7052
Block diagonal MinJ with $M_k = A_{kk}$	.2906	.4683
Block inc. G–S MinJ with $M_k = \text{tril}(A_{kk})$	.5971	.7052
Block inc. G–S MinJ with $M_k = A_{kk}$	.2906	.4683

The first and second row computations indicates that the MinJ MS for this nonsymmetric matrix has better convergence properties than the weighted MS. In a single weighted MS step there are two concurrent N^2+N solves and an average over N nodes; in the block diagonal G–S MinJ there are two concurrent N^2 solves and one N solve. The second and third row computation is an illustration of the comparison theorem, Theorem 6. Computations in rows 2 and 4 and in rows 3 and 5 illustrate Theorem 4.

Driven cavity problem. Consider a two space variable driven cavity problem with unknowns (u, v, P) at each point in space for $u = x$-direction velocity component, $v = y$-direction velocity component, and $P =$ pressure. A semi-implicit time discretization of the Navier–Stokes equations for flow in two dimensions, which is incompressible, is

$$u/\Delta t - 1/\,\text{Re}\,\Delta u + a_1 u_x + a_2 u_y + P_x = f, \qquad x\text{-momentum},$$
$$v/\Delta t - 1/\,\text{Re}\,\Delta v + b_1 v_x + b_2 v_y + P_y = g, \qquad y\text{-momentum},$$
$$u_x + v_y = 0, \qquad \text{incompressible}.$$

Now discretize the space variables so that A_1 is the y-momentum coefficient matrix,

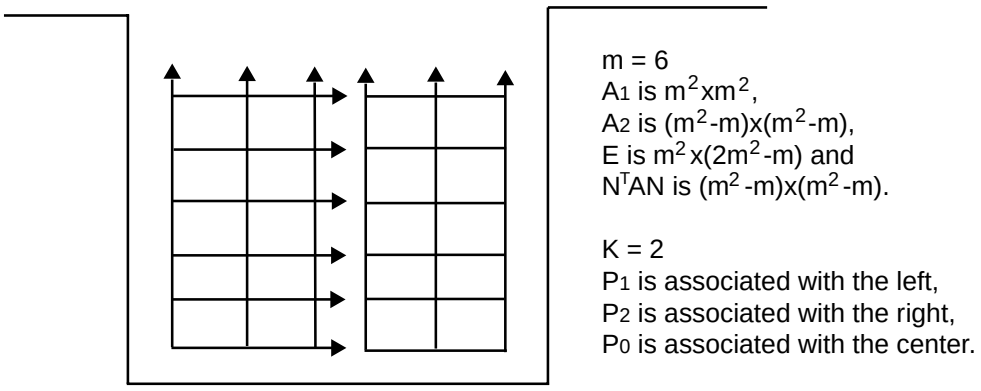


FIG. 2. *Driven cavity and partition of unknowns.*

A_2 is the x-momentum coefficient matrix, and $E = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$ is the coefficient matrix for the incompressible equation. Order all the y-velocity components, v, first and all the x-velocity components, u, second. The above then becomes

$$\begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} + \begin{bmatrix} {R_1}^T \\ {R_2}^T \end{bmatrix} P = \begin{bmatrix} f \\ g \end{bmatrix}$$

and

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = 0.$$

Or, $Ax + E^T y = s$, where $x = \left[\begin{smallmatrix} v \\ u \end{smallmatrix}\right]$ and $y = P$, $Ex = t$.
The last two equations are known as the equilibrium equations, which also evolve from other important applications such as structures and circuits. More details can be found in Plemmons and White [7].

If the nodes are carefully ordered, then R_1 in E will be nonsingular. Then the nullspace of E will be given by a basis consisting of the columns of N. This enables one to solve the equilibrium equations by the nullspace method:

Form $N = \begin{bmatrix} {R_1}^{-1} R_2 \\ -I \end{bmatrix}$,
Solve $Ex_p = t$,
Set $x = Nx_0 + x_p$ and
Solve for x_0, $N^T A N x_0 = N^T(s - Ax_p)$.

The last equation is formed by inserting the nullspace representation of x into $Ax + E^T = s$ and multiplying by N^T. We solve this system via the restarted GMRES algorithm.

In our calculations let m equal the number of grid points in both the x and y directions. Then there will be m unknown v components in each vertical subset of grid points, and there will be $m-1$ unknown u components in each horizontal subset of grid points. In Figure 2 the unknown v components are the vertical line segments, and the unknown u components are the horizontal line segments. Here $m = 6$ and $K = 2$.

The first set of calculations uses $m = 12$, the symmetric case, $\mathrm{Re} = 10$, $\Delta t = 10$, and a restart after three inner iterations. The preconditioners are the block incomplete

Gauss–Seidel MinJ and MinR MSs, where $K = 2$ and $M_k = A_{kk}$. The error for the MinJ was $((x^1 - x)^T A(x^1 - x))^{1/2}$. These calculations are in Table 2.

TABLE 2
MinJ and MinR errors.

Outer GMRES Iteration = 1	MinJ MS error $m = 12$, inner = 3	MinR MS error $m = 12$, inner = 3
1	1.0616	.1047
2	.3235	.0429
3	.1097	.0127
4	.0416	.0055
5	.0135	.0016

The second set of calculations is for the nonsymmetric case with coefficients of the first order terms equal to 10. Here we used just the MinR MS as a preconditioner for GMRES with variable m and $(m/5 + 1)$ restarts. The error of the MinR was $(r(x^1)^T r(x^1)/b^T b)^{1/2}$. These calculations are in Table 3.

TABLE 3
MinR and variable unknowns.

Outer GMRES Iteration=1	MinR MS error $m = 10$, inner = 3	MinR MS error $m = 20$, inner = 5	MinR MS error $m = 30$, inner = 7
1	.1059	.1125	.0864
2	.0174	.0277	.0175
3	.0036	.0024	.0038
4	.0009	.0008	.0016
5	.0001	.0002	.0001

REFERENCES

[1] M. BENASSI AND R. E. WHITE, *Parallel numerical solution of variational inequalities*, SIAM J. Numer. Anal., 31 (1994), pp. 813–830.

[2] G. CSORDAS AND R. S. VARGA, *Comparison of regular splitting matrices*, Numer. Math., 44 (1984), pp. 23–35.

[3] L. ELSNER, *Comparison of weak regular splittings and multisplitting methods*, Numer. Math., 56 (1989), pp. 283–289.

[4] I. MAREK AND D. SZYLD, *Comparison theorems for weak splittings of bounded operators*, Numer. Math., 58 (1990), pp. 387–397.

[5] D. P. O'LEARY AND R. E. WHITE, *Multisplittings of matrices and parallel solution of linear systems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 630–640.

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

[7] R. J. PLEMMONS AND R. E. WHITE, *Substructuring methods for computing the nullspace for equilibrium matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 1–22.

[8] R. E. WHITE, *Multisplittings with different weighting schemes*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 481–493.

[9] R. E. WHITE, *Multisplittings of a symmetric positive definite matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 69–82.

[10] Z. WOZNICKI, *Two-Sweep Iterative Methods for Solving Large Linear Systems and Their Application to the Numerical Solution of Multi-Group Multi-Dimensional Neutron Diffusion*, Ph.D. thesis, Institute of Nuclear Research, Swierk k/Otwocha, Poland, 1973.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 569–573

SOME NORM INEQUALITIES FOR COMPLETELY MONOTONE FUNCTIONS*

JASPAL SINGH AUJLA†

Abstract. Let A, B be $n \times n$ complex positive semidefinite matrices, and let f be a completely monotone function on $[0, \infty)$. We prove that $2|||f(A+B)||| \leq |||f(2A)+f(2B)|||$ for all unitarily invariant norms $|||\cdot|||$. The corollary $2^{1-r}|||(A+B)^r||| \leq |||A^r+B^r|||$, $r \leq 0$, supplements recent results of Ando, Bhatia, Kittaneh, and Zhan.

Key words. positive definite matrix, unitarily invariant norm, completely monotone function

AMS subject classifications. 47A30, 47B15, 15A60

PII. S0895479800369761

1. Introduction.

Let $\mathcal{M}_n$ denote the set of $n \times n$ complex matrices. We denote by $\mathcal{H}_n$ the set of all Hermitian matrices in $\mathcal{M}_n$ and by $\mathcal{S}_n$ the set of all positive semidefinite matrices in $\mathcal{M}_n$. The set of all positive definite matrices in $\mathcal{M}_n$ shall be denoted by $\mathcal{P}_n$.

A norm $|||\cdot|||$ on $\mathcal{M}_n$ is called unitarily invariant or symmetric if

$$|||UAV||| = |||A|||$$

for all $A \in \mathcal{M}_n$ and for all unitaries U, V. A unitarily invariant norm is monotone in the sense that $0 \leq A \leq B$ implies $|||A||| \leq |||B|||$. The most basic unitarily invariant norms are the Ky Fan norms $\|\cdot\|_{(k)}, (k = 1, 2, \ldots, n)$, defined as

$$\| A \|_{(k)} = \sum_{j=1}^{k} \sigma_j(A), (k = 1, 2, \ldots, n),$$

where $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_n(A)$ are the singular values of A.

The maximum principle of Ky Fan says that for $k = 1, 2, \ldots, n$, we have

$$\sum_{j=1}^{k} \sigma_j(A) = \max \sum_{j=1}^{k} | \langle Au_j, v_j \rangle |,$$

where the maximum is taken over all choices of orthonormal vectors $\{u_1, u_2, \ldots, u_k\}$ and $\{v_1, v_2, \ldots, v_k\}$. If $A \in \mathcal{H}_n$, we can choose $u_j = v_j$. The Ky Fan dominance principle says that

$$\| A \|_{(k)} \geq \| B \|_{(k)},$$

$k = 1, 2, \ldots, n$ if and only if

$$|||A||| \geq |||B|||$$

*Received by the editors March 15, 2000; accepted for publication (in revised form) by R. Bhatia March 28, 2000; published electronically August 9, 2000.
http://www.siam.org/journals/simax/22-2/36976.html
†Department of Applied Mathematics, Regional Engineering College, Jalandhar-144011, Punjab, India (jsaujla@hotmail.com).

for all unitarily invariant norms $|||\cdot|||$.

A real valued function f on $[0,\infty)$ is said to be completely monotone if $(-1)^k f^{(k)}(x) \geq 0$ for all $k = 0, 1, \ldots$, and all $x \in [0,\infty)$. Here $f^{(0)} = f$ and $f^{(k)}, k = 1, 2, \ldots$, denotes the kth derivative of f. There is a theorem of Bernstein (see [4]) that says that a function f is completely monotone on $[0,\infty)$ if and only if there exists a positive measure μ such that

$$f(x) = \int_0^\infty e^{-\lambda x} d\mu(\lambda). \tag{1}$$

Let f be a real valued function defined on an interval I, and let $A \in \mathcal{H}_n$ have its spectrum in I. Then $f(A)$ is defined by the familiar functional calculus. The function f is called matrix monotone increasing if $A \geq B$ implies $f(A) \geq f(B)$ for $A, B \in \mathcal{H}_n$ with spectrum in I. The function f is called matrix monotone decreasing if $-f$ is matrix monotone increasing. If f is a matrix monotone increasing function on $[0,\infty)$, then f admits the integral representation

$$f(x) = \alpha + \beta x + \int_0^\infty \left(\frac{\lambda}{\lambda^2 + 1} - \frac{1}{x + \lambda} \right) d\mu(\lambda), \tag{2}$$

where α is a real number, $\beta \geq 0$, and μ is a positive measure (see [2]). From the integral representation (2) it follows that the derivative of a matrix monotone increasing function is completely monotone.

Bhatia and Kittaneh [3] proved that for all $A, B \in \mathcal{P}_n$ and for all $1 \leq r < \infty$,

$$\| A^r + B^r \| \leq \| (A + B)^r \|,$$

where $\|\cdot\|$ is the operator norm. They further proved that for any positive integer m,

$$|||A^m + B^m||| \leq |||(A + B)^m|||$$

for all unitarily invariant norms $|||\cdot|||$, and they conjectured that this inequality remains true when m is replaced by any real number $r \geq 1$. Ando and Zhan [1] affirmatively settled this question and proved that

$$|||(A + B)^r||| \leq |||A^r + B^r|||$$

for $0 \leq r \leq 1$ and

$$|||A^r + B^r||| \leq |||(A + B)^r|||$$

for $1 \leq r < \infty$, $A, B \in \mathcal{S}_n$, and for all unitarily invariant norms $|||\cdot|||$. It is natural to ask what happens for negative values of r. We settle this problem and prove a more general result for completely monotone functions. We prove that if the function f is completely monotone on $[0,\infty)$, then

$$2|||f(A + B)||| \leq |||f(2A) + f(2B)|||$$

for all $A, B \in \mathcal{S}_n$ and all unitarily invariant norms $|||\cdot|||$. This in particular includes the inequality

$$2^{1-r}|||(A + B)^r||| \leq |||A^r + B^r|||$$

for all $-\infty < r \leq 0$ and $A, B \in \mathcal{P}_n$.

2. Main results. The following lemmas will be used in what follows. The reader may refer to [2, Theorem IX.3.7 and Theorem IX.4.5] for their proofs.

LEMMA 2.1. *Let $A, B \in \mathcal{H}_n$. Then*

$$|||e^{A+B}||| \ \leq \ |||e^A e^B|||$$

for all unitarily invariant norms $|||\cdot|||$.

LEMMA 2.2. *Let $A, B, X \in \mathcal{M}_n$. Then*

$$|||A^*XB||| \ \leq \ \frac{1}{2}|||AA^*X + XBB^*|||$$

for all unitarily invariant norms $|||\cdot|||$.

THEOREM 2.3. *Let f be a completely monotone function on $[0,\infty)$, and let $A, B \in \mathcal{S}_n$. Then*

$$2|||f(A+B)||| \ \leq \ |||f(2A) + f(2B)|||$$

for all unitarily invariant norms $|||\cdot|||$.

Proof. First we prove the result for the function $f_\lambda(x) = e^{-\lambda x}, \lambda, x > 0$. Using Lemma 2.1 and Lemma 2.2, respectively, one gets

$$\begin{aligned} 2|||f_\lambda(A+B)||| &= 2|||e^{-\lambda(A+B)}||| \\ &\leq 2|||e^{-\lambda A}e^{-\lambda B}||| \\ &\leq \ |||e^{-\lambda 2A} + e^{-\lambda 2B}||| \\ &= \ |||f_\lambda(2A) + f_\lambda(2B)|||. \end{aligned}$$

Now let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $A+B$, and let $u_1, u_2, \ldots, u_n$ be the corresponding orthonormal eigenvectors. Further, let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be the eigenvalues of $f_\lambda(2A) + f_\lambda(2B)$, and let $v_1, v_2, \ldots, v_n$ be the corresponding orthonormal eigenvectors. The above inequality for the Ky Fan norms gives

$$2 \ \| \ f_\lambda(A+B) \ \|_{(k)} \ \leq \ \| \ f_\lambda(2A) + f_\lambda(2B) \ \|_{(k)},$$

which implies

$$2\sum_{j=1}^{k}\langle f_\lambda(A+B)u_j, u_j\rangle \ \leq \ \sum_{j=1}^{k}\langle (f_\lambda(2A) + f_\lambda(2B))v_j, v_j\rangle,$$

since $u_1, u_2, \ldots, u_n$ are the eigenvectors of $f_\lambda(A+B)$ corresponding to the eigenvalues $f_\lambda(\lambda_1) \ \geq \ f_\lambda(\lambda_2) \ \geq \cdots \geq f_\lambda(\lambda_n)$. Therefore via the integral representation (1) for the completely monotone function f, we arrive at

$$2\sum_{j=1}^{k}\langle f(A+B)u_j, u_j\rangle \ \leq \ \sum_{j=1}^{k}\langle (f(2A) + f(2B))v_j, v_j\rangle \tag{3}$$

for $k = 1, 2, \ldots, n$. Again, since $u_1, u_2, \ldots, u_n$ are the eigenvectors of $f(A+B)$ corresponding to the eigenvalues $f(\lambda_1) \ \geq \ f(\lambda_2) \ \geq \cdots \geq f(\lambda_n)$, we have by definition

$$\| \ f(A+B) \ \|_{(k)} \ = \ \sum_{j=1}^{k}\langle f(A+B)u_j, u_j\rangle.$$

On the other hand by the Ky Fan maximum principle, we have

$$\sum_{j=1}^{k}\langle(f(2A)+f(2B))v_j,v_j\rangle \;\leq\; \parallel f(2A)+f(2B)\parallel_{(k)}.$$

Therefore from inequality (3), we get the inequality

$$2\;\parallel f(A+B)\parallel_{(k)} \;\leq\; \parallel f(2A)+f(2B)\parallel_{(k)}$$

for $k=1,2,\ldots,n$. Hence by the Ky Fan dominance principle, one gets

$$2|||f(A+B)||| \;\leq\; |||f(2A)+f(2B)|||.$$

This completes the proof of the theorem. □

Since the derivative of a matrix monotone increasing function on $[0,\infty)$ is completely monotone, we have the following corollary.

COROLLARY 2.4. *Let f be a matrix monotone increasing function on $[0,\infty)$ and $A,B\in\mathcal{S}_n$. Then*

$$2|||f^{(1)}(A+B)||| \;\leq\; |||f^{(1)}(2A)+f^{(1)}(2B)|||$$

for all unitarily invariant norms $|||\cdot|||$.

COROLLARY 2.5. *Let $A,B\in\mathcal{P}_n$ and $-\infty<r\leq 0$. Then*

$$2^{1-r}|||(A+B)^r||| \;\leq\; |||A^r+B^r|||$$

for all unitarily invariant norms $|||\cdot|||$.

Proof. Since the function $f(x)=x^r$, $-\infty<r\leq 0$ is completely monotone on $(0,\infty)$, one has the desired inequality by Theorem 2.3. □

THEOREM 2.6. *Let f be a nonnegative matrix monotone decreasing function on $[0,\infty)$, and let $A,B\in\mathcal{S}_n$. Then*

$$2|||f(A+B)||| \;\leq\; |||f(A)+f(B)|||$$

for all unitarily invariant norms $|||\cdot|||$.

Proof. Observe that

$$f(A+B)\leq f(A)$$

and

$$f(A+B)\leq f(B).$$

Consequently,

$$2f(A+B)\leq f(A)+f(B).$$

Now the result follows, since a unitarily invariant norm is monotone. □

Remark 2.7. If for a nonnegative function f on $[0,\infty)$ with $f(0)=0$, the inequality

$$|||f(A+B)||| \;\leq\; |||f(A)+f(B)|||$$

holds, then f is midpoint concave. To see this one may take $A = \begin{pmatrix} x & \sqrt{xy} \\ \sqrt{xy} & y \end{pmatrix}$ and $B = \begin{pmatrix} x & -\sqrt{xy} \\ -\sqrt{xy} & y \end{pmatrix}$. Then using the given inequality for the trace norm one gets the inequality that shows the midpoint concavity of f. Further, if for a nonnegative function f on $[0, \infty)$ the inequality

$$|||f(A+B)||| \;\geq\; |||f(A)+f(B)|||$$

holds, then f is midpoint convex with $f(0) = 0$. This can be seen similarly.

REFERENCES

[1] T. Ando and X. Zhan, *Norm inequalities related to operator monotone functions*, Math. Ann., 315 (1999), pp. 771–780.

[2] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[3] R. Bhatia and F. Kittaneh, *Norm inequalities for positive operators*, Letters Math. Phys., 43 (1998), pp. 225–231.

[4] D. V. Widder, *Laplace Transforms*, Princeton University Press, Princeton, NJ, 1968.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 574–579

IMPROVED UPPER BOUNDS FOR THE REAL PART OF NONMAXIMAL EIGENVALUES OF NONNEGATIVE MATRICES*

REINHARD NABBEN†

Abstract. In this note we improve two upper bounds for the real part of nonmaximal eigenvalues of nonnegative irreducible matrices. We also improve an upper bound for the spectral radius of principal submatrices of nonnegative matrices.

Key words. nonnegative matrices, doubly stochastic matrices, nonmaximal eigenvalues, Cheeger-type bounds

AMS subject classifications. 15A18, 15A42, 15A48, 15A51

PII. S0895479899367755

1. Introduction.

Let $A = [a_{i,j}] \in \mathbb{R}^{n,n}$ be a nonnegative irreducible matrix with positive right and left eigenvectors u and v. We arrange the eigenvalues of A as

$$\rho(A) = \lambda_n(A) > \mathrm{Re}(\lambda_{n-1}(A)) \geq \cdots \geq \mathrm{Re}(\lambda_1(A)).$$

In many applications of nonnegative matrices, such as finite Markov chains, algebraic connectivity of graphs, and convergence rates of iterative methods, one needs bounds for $\lambda_{n-1}(A)$ or $\mathrm{Re}(\lambda_{n-1}(A))$. Recently, some new bounds, so-called Cheeger-type bounds, were established. These bounds use different Cheeger constants of nonnegative matrices.

It is proved by Friedland and Gurvits in [4] that

$$\mathrm{Re}(\lambda_{n-1}(A)) \leq \rho(A) - \frac{1}{2}\left(\rho(A) - \max_i a_{ii}\right)\epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v)^2, \tag{1.1}$$

where

$$\epsilon_s(A,u,v) := \min_{\emptyset \neq V \subset \langle n \rangle, |V| \leq s} \frac{\sum_{i \in V, j \in \langle n \rangle \setminus V} a_{ij} v_i u_j + a_{ji} v_j u_i}{\sum_{i \in V} 2(\rho(A) - a_{ii}) v_i u_i}$$

and $\epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v)$ is a Cheeger constant of A.

In this note we improve (1.1). We show that

$$\mathrm{Re}(\lambda_{n-1}(A)) \leq \rho(A) - \left(\rho(A) - \max_i a_{ii}\right)\left(1 - \sqrt{1 - i(A,u,v)^2}\right),$$

where $i(A,u,v)$ is another Cheeger constant (see (2.5)) that satisfies $i(A,u,v) \geq \epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v)$. Thus we obtain an improved upper bound.

A different Cheeger-type bound for $\mathrm{Re}(\lambda_{n-1})$ is given by Berman and Zhang. They proved in [1] that

$$Re(\lambda_{n-1}) \leq \sqrt{\rho(A)^2 - \frac{h(A,u,v)^2}{\max_i (u_i v_i)^2}}, \tag{1.2}$$

*Received by the editors December 29, 1999; accepted for publication (in revised form) by R. Brualdi January 31, 2000; published electronically August 9, 2000.
http://www.siam.org/journals/simax/22-2/36775.html

†Fakultät für Mathematik, Universität Bielefeld, Postfach 10 01 31, 33 501 Bielefeld, Germany (nabben@mathematik.uni-bielefeld.de).

where

$$h(A,u,v) = \min_{\emptyset \neq U \subset \langle n \rangle, |U| \leq \lfloor \frac{n}{2} \rfloor} \frac{\sum_{i \in U, j \in \langle n \rangle \setminus U} a_{ij} v_i u_j + a_{ji} v_j u_i}{2|U|}. \tag{1.3}$$

Here we establish that

$$\begin{aligned} &\mathrm{Re}(\lambda_{n-1}(A)) \leq \rho(A)\sqrt{1 - l(A,u,v)^2}, \\ &\mathrm{Re}(\lambda_{n-1}(A)) \leq \min_i a_{ii} + \sqrt{(\rho(A) - \min_i a_{ii})^2 - \frac{h(A,u,v)^2}{\max_i (u_i v_i)^2}}, \end{aligned}$$

where $l(A,u,v)$ is defined similarly as $i(A,u,v)$; see (2.6). We obtain that both upper bounds above are smaller than the bound (1.2). Thus we also improve the bound (1.2) due to Berman and Zhang.

The right-hand side of (1.1) can also be used to get an upper bound for the spectral radius of principal submatrices of A. Let U be a nonempty subset of $\langle n \rangle = \{1, \ldots, n\}$ and define

$$\rho_s(A) = \max_{U \subset \langle n \rangle, |U| = s} \rho(A(U)),$$

where $A(U)$ is the submatrix of A whose rows and columns are in U. The values $\rho_s(A)$ are used to define a partition of the class of Z-matrices by Fiedler and Markham in [2] (see also [7] for more details). Moreover, it is established in [5] that for any real eigenvalue λ of A different from $\rho(A)$ it holds that

$$\lambda \leq \rho_{\lfloor \frac{n}{2} \rfloor}(A).$$

It is then proved in [5] that

$$\rho_s(A) \leq \rho(A) - \frac{1}{2}\left(\rho(A) - \max_i a_{ii}\right)\epsilon_s(A,u,v)^2. \tag{1.4}$$

Here we show that

$$\begin{aligned} \rho_s(A) &\leq \rho(A) - \left(\rho(A) - \max_i a_{ii}\right)\left(1 - \sqrt{1 - \epsilon_s(A,u,v)^2}\right) \\ &\leq \rho(A) - \frac{1}{2}\left(\rho(A) - \max_i a_{ii}\right)\epsilon_s(A,u,v)^2. \end{aligned}$$

2. Improved bounds. Our results are mainly based on different Cheeger-type lower bounds for the second smallest eigenvalue of the Laplacian matrix of a graph. A comparison between different well-known bounds and some new bounds is given in [6].

Let $G_w = (V, E(G_w)$ be a connected, undirected weighted graph. Note that G_w may have loops. Here $w_{i,j}$ denotes the positive weight $w_{i,j}$ of $(i,j) = (j,i) \in E(G_w)$. The Laplacian matrix $L_w = [l_{i,j}]$ is defined as

$$\begin{aligned} l_{i,j} &= -w_{i,j} \quad \text{for } i \neq j, \\ l_{i,i} &= \delta_i - w_{i,i}, \end{aligned}$$

where

$$\delta_i = \sum_{(i,j) \in E(G_w)} w_{i,j}$$

are the weighted degrees. The established bounds use different Cheeger constants $h(G_w)$ and $i(G_w)$:

$$h(G_w) = \min_{\emptyset \neq U \subset V} \frac{|E(U,\bar{U})|}{\min(|U|,|\bar{U}|)} = \min_{\substack{\emptyset \neq U \subset V \\ 0 \neq |U| \leq \lfloor \frac{n}{2} \rfloor}} \frac{|E(U,\bar{U})|}{|U|},$$

$$i(G_w) = \min_{\emptyset \neq U \subset V} \frac{|E(U,\bar{U})|}{\min(|E(U)|,|E(\bar{U})|)} = \min_{\substack{\emptyset \neq U \subset V \\ |E(U)| \leq \frac{1}{2}|E(V)|}} \frac{|E(U,\bar{U})|}{|E(U)|},$$

where

$$|E(U)| = \sum_{i \in U} \delta_i,$$

$\bar{U} = \langle n \rangle \backslash U$, and $E(U,\bar{U})$ are the edges connecting vertices of U with vertices in $\bar{U}$.

Now let $D = \operatorname{diag}(d_1, \ldots, d_n)$ be a positive diagonal matrix. Then Theorem 2.1 of [6] gives

$$\lambda_2(D^{-1}L_w) \geq \min_i \frac{\delta_i}{d_i} \left(1 - \sqrt{1 - i(G_w)^2}\right) \tag{2.1}$$

while Theorem 2.2 of [1] says

$$\lambda_2(D^{-1}L_w) \geq \frac{1}{\max_i d_i} \left(\bar{\delta} - \sqrt{\bar{\delta}^2 - h(G_w)^2}\right). \tag{2.2}$$

Now it is worthwhile to mention that weighted loops do not influence the Laplacian matrix. Thus we can think of L_w as a Laplacian matrix of a weighted graph with loops or as the Laplacian matrix of a weighted graph without loops. The inequalities (2.1) and (2.2) hold for both cases. (This is not mentioned in [1] for (2.2).) But of course one gets different bounds since one has different weighted degrees and a different constant $i(G_w)$. It is shown in [6] that for (2.1) loops can increase or decrease the bound. However, since $h(G_w)$ is independent of loops we obtain with $\bar{v} = \max_i v_i$ for $v = [v_i] \in \mathbb{R}^n$.

PROPOSITION 2.1. *Let G_w be a weighted, undirected, connected graph and $V = \langle n \rangle$. Let $D = \operatorname{diag}(d_1, \ldots, d_n)$ be a positive diagonal matrix. Then*

$$\lambda_2(D^{-1}L_w) \geq \frac{1}{\bar{d}} \left(\bar{\gamma} - \sqrt{\bar{\gamma}^2 - h(G_w)^2}\right) \tag{2.3}$$

$$\geq \frac{1}{\bar{d}} \left(\bar{\delta} - \sqrt{\bar{\delta}^2 - h(G_w)^2}\right), \tag{2.4}$$

where

$$\gamma = [\gamma_i]_{i=1}^n \quad \text{with} \quad \gamma_i = \delta_i - w_{i,i}.$$

Proof. The proof of Theorem 2.2 in [1] works for both cases, loops or no loops. Thus we only have to show the second inequality. But the function

$$f(t) = \frac{1}{\bar{d}} \left(t - \sqrt{t^2 - h(G_w)^2}\right)$$

decreases for $t \geq h(G_w)$. ∎

We now consider nonnegative matrices $A = [a_{i,j}] \in \mathbb{R}^{n,n}$. Let $U \subset \langle n \rangle$ and define for convenience

$$||U||_{nl} := \sum_{\substack{i \in U, j \in \langle n \rangle \\ i \neq j}} a_{ij}u_iv_j + a_{ji}u_jv_i = \sum_{i \in U} 2(\rho(A) - a_{ii})v_iu_i,$$

$$||U||_{l} := \sum_{i \in U, j \in \langle n \rangle} a_{ij}u_iv_j + a_{ji}u_jv_i = \sum_{i \in U} 2\rho(A)v_iu_i.$$

Moreover, let

$$i(A,u,v) := \min_{\emptyset \neq U \subset \langle n \rangle} \frac{\sum_{i \in U, j \in \langle n \rangle \setminus U} a_{ij}v_iu_j + a_{ji}v_ju_i}{\min(||U||_{nl}, ||\bar{U}||_{nl})} = \min_{\substack{\emptyset \neq U \subset \langle n \rangle \\ ||U||_{nl} \leq \frac{1}{2}||\langle n \rangle||_{nl}}} \frac{\sum_{i \in U, j \in \langle n \rangle \setminus U} a_{ij}v_iu_j + a_{ji}v_ju_i}{\sum_{i \in U} 2(\rho(A) - a_{ii})v_iu_i}, \tag{2.5}$$

$$l(A,u,v) := \min_{\emptyset \neq U \subset \langle n \rangle} \frac{\sum_{i \in U, j \in \langle n \rangle \setminus U} a_{ij}v_iu_j + a_{ji}v_ju_i}{\min(||U||_{l}, ||\bar{U}||_{l})} = \min_{\substack{\emptyset \neq U \subset \langle n \rangle \\ ||U||_{l} \leq \frac{1}{2}||\langle n \rangle||_{l}}} \frac{\sum_{i \in U, j \in \langle n \rangle \setminus U} a_{ij}v_iu_j + a_{ji}v_ju_i}{\sum_{i \in U} 2\rho(A)v_iu_i}, \tag{2.6}$$

$$\epsilon(A,U,u,v) := \min_{\emptyset \neq W \subset U} \frac{\sum_{i \in W, j \in \langle n \rangle \setminus W} a_{ij}v_iu_j + a_{ji}v_ju_i}{\sum_{i \in W} 2(\rho(A) - a_{ii})v_iu_i}.$$

We easily obtain

$$\epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v) \leq i(A,u,v), \tag{2.7}$$

$$h(A,u,v) \leq \max_i u_iv_i \ \rho(A)l(A,u,v). \tag{2.8}$$

Then we get the following theorem.

THEOREM 2.2. *Let $A = [a_{i,j}] \in \mathbb{R}^{n,n}$ be a nonnegative irreducible matrix with positive right and left eigenvectors u and v. Then*

$$\mathrm{Re}(\lambda_{n-1}(A)) \leq \rho(A) - \left(\rho(A) - \max_i a_{ii}\right)\left(1 - \sqrt{1 - i(A,u,v)^2}\right) \tag{2.9}$$

$$\leq \rho(A) - \left(\rho(A) - \max_i a_{ii}\right)\left(1 - \sqrt{1 - \epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v)^2}\right) \tag{2.10}$$

$$\leq \rho(A) - \frac{1}{2}\left(\rho(A) - \max_i a_{ii}\right)\epsilon_{\lfloor \frac{n}{2} \rfloor}(A,u,v)^2, \tag{2.11}$$

and

$$\mathrm{Re}(\lambda_{n-1}(A)) \leq \rho(A)\sqrt{1 - l(A,u,v)^2} \tag{2.12}$$

$$\leq \sqrt{\rho(A)^2 - \frac{h(A,u,v)^2}{\max_i (u_iv_i)^2}}. \tag{2.13}$$

Moreover,

$$\mathrm{Re}(\lambda_{n-1}(A)) \leq \min_i a_{ii} + \sqrt{(\rho(A) - \min_i a_{ii})^2 - \frac{h(A,u,v)^2}{\max_i (u_iv_i)^2}} \tag{2.14}$$

$$\leq \sqrt{\rho(A)^2 - \frac{h(A,u,v)^2}{\max_i (u_iv_i)^2}}. \tag{2.15}$$

For $s = 1, \ldots, n-1$,

$$\rho_s(A) \leq \rho(A) - \left(\rho(A) - \max_i a_{ii}\right)\left(1 - \sqrt{1 - \epsilon_s(A,u,v)^2}\right) \tag{2.16}$$

$$\leq \rho(A) - \frac{1}{2}\left(\rho(A) - \max_i a_{ii}\right)\epsilon_s(A,u,v)^2. \tag{2.17}$$

Proof. First assume that $A = A^T$ and $u = v$. We consider the weighted graph G_w, without loops, whose weighted Laplacian matrix is

$$L_w = Q(\rho(A)I - A)Q,$$

where $Q = \operatorname{diag}(u_1, \ldots, u_n)$. The weighted degrees δ_i are given by

$$\delta_i = \sum_{1 \leq j \leq n, j \neq i} a_{ij}u_iu_j = (\rho(A) - a_{ii})u_i^2.$$

We get for the second smallest eigenvalue of $Q^{-2}L_w$

$$\begin{aligned}\lambda_2(Q^{-2}L_w) &= \lambda_2(Q^{-1}(\rho(A) - A)Q) = \lambda_2(\rho(A) - A)\\ &= \rho(A) - \lambda_{n-1}(A).\end{aligned}$$

Now let U be a subset of $\langle n\rangle$ with $|U| = s$. Similarly we obtain for the smallest eigenvalue of $Q^{-2}L_w(U)$

$$\lambda_1(Q^{-2}L_w(U)) = \rho(A) - \rho(A(U)).$$

Now we apply (2.1) and Lemma 2.3 of [6] with $D = Q^2$ and get

$$\rho(A) - \lambda_{n-1}(A) \geq \min_i \frac{\delta_i}{d_i}\left(1 - \sqrt{1 - i(A,u,v)^2}\right),$$

$$\rho(A) - \rho(A(U)) \geq \min_i \frac{\delta_i}{d_i}\left(1 - \sqrt{1 - \epsilon_s(A,u,v)^2}\right).$$

Thus (2.9) and (2.16) hold for symmetric matrices.

Now assume that A is not symmetric. Let F be the unique positive diagonal matrix such that $Fu = F^{-1}v = x$ and consider $\tilde{A} = FAF^{-1}$. Note that the spectrum of A and $\tilde{A}$ are the same. Moreover, $i(A,u,v) = i(\tilde{A},x,x)$ and $\epsilon(A,U,u,v) = \epsilon(\tilde{A},U,x,x)$ for any $U \subset \langle n\rangle$.

Next consider $B = (\tilde{A} + \tilde{A}^T)/2$. The arguments of [4] yield

$$\lambda_{n-1}(B) \geq \operatorname{Re}(\lambda_{n-1}(\tilde{A})).$$

We also have $\rho(B) = \rho(\tilde{A})$. Moreover, $i(B,x,x) = i(\tilde{A},x,x)$ and $\epsilon(B,U,x,x) = \epsilon(\tilde{A},U,x,x)$ for all $U \subset \langle n\rangle$. The maximal characterization of $\rho(B(U))$ implies the inequality $\rho(B(U)) \geq \rho(\tilde{A}(U))$. In particular, $\rho_s(B) \geq \rho_s(A)$. Thus (2.9) and (2.16) also hold for nonsymmetric matrices. With (2.7) we get the inequality (2.10). Since

$$1 - \sqrt{1 - \epsilon_s(A,u,v)^2} \geq \frac{1}{2}\epsilon_s(A,u,v)^2,$$

we obtain (2.11) and (2.17). To prove (2.12) and (2.14) we consider just the weighted graph G_w with loops and use arguments similar to those above. The inequality (2.13) follows from (2.8). Moreover, (2.15) follows from Proposition 2.1. ∎

Theorem 2.2 is also true for reducible matrices. But then we have to consider eigenvalues different from $\rho(A)$.

Next we consider doubly stochastic matrices, i.e., nonnegative matrices A for which $Ae = e$ and $e^T A = e^T$, where $e = (1, \dots, 1)^T$.

Note that we have $||\langle n \rangle||_{nl} = 2\,(n - \sum_i^n a_{ii})$ and $||\langle n \rangle||_l = 2n$. We obtain the following corollary.

COROLLARY 2.3. *Let A be a doubly stochastic matrix. Then for any eigenvalue λ of A different from 1 we obtain*

$$\mathrm{Re}(\lambda) \le \max_i a_{ii} + \left(1 - \max_i a_{ii}\right) \sqrt{1 - i(A,e,e)^2}$$

and

$$\begin{aligned} \mathrm{Re}(\lambda) &\le \sqrt{1 - l(A,e,e)^2} \\ &\le \sqrt{1 - h(A,e,e)^2}. \end{aligned}$$

Moreover,

$$\begin{aligned} \mathrm{Re}(\lambda) &\le \min_i a_{ii} + \sqrt{(1 - \min_i a_{ii})^2 - h(A,e,e)^2} \\ &\le \sqrt{1 - h(A,e,e)^2}. \end{aligned}$$

Proof. For doubly stochastic matrices we have

$$\sum_{i \in U, j \in \bar{U}} a_{ij} = \sum_{i \in U, j \in \bar{U}} a_{ji}.$$

Thus, all inequalities follow from Theorem 2.2. ∎

REFERENCES

[1] A. BERMAN, X.-D.ZHANG, *Lower bounds for the eigenvalues of Laplacian matrices*, Linear Algebra Appl., to appear.

[2] M. FIEDLER AND T.L. MARKHAM, *A classification of matrices of class Z*, Linear Algebra Appl., 173 (1992), pp. 115–124.

[3] S. FRIEDLAND, *Lower bounds for the first eigenvalue of certain M-matrices associated with graphs*, Linear Algebra Appl., 172 (1992), pp. 71–84.

[4] S. FRIEDLAND AND L. GURVITS, *An upper bound for the real part of nonmaximal eigenvalues of nonnegative irreducible matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1015–1017.

[5] S. FRIEDLAND AND R. NABBEN, *On the second real eigenvalue of nonnegative and Z-matrices*, Linear Algebra Appl., 255 (1997), pp. 303–313.

[6] S. FRIEDLAND AND R. NABBEN, *On Cheeger-Type Inequalities for Weighted Graphs*, submitted.

[7] R. NABBEN, *Z–matrices and inverse Z–matrices*, Linear Algebra Appl., 256 (1997), pp. 31–48.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 580–601

ON THE COMPUTATION OF THE RESTRICTED SINGULAR VALUE DECOMPOSITION VIA THE COSINE-SINE DECOMPOSITION*

DELIN CHU†, LIEVEN DE LATHAUWER‡, AND BART DE MOOR‡

Abstract. In this paper, we show that the restricted singular value decomposition of a matrix triplet $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$ can be computed by means of the cosine-sine decomposition. In the first step, the matrices A, B, C are reduced to a lower-dimensional matrix triplet $\mathcal{A}, \mathcal{B}, \mathcal{C}$, in which $\mathcal{B}$ and $\mathcal{C}$ are nonsingular, using orthogonal transformations such as the QR-factorization with column pivoting and the URV decomposition. In the second step, the components of the restricted singular value decomposition of A, B, C are derived from the singular value decomposition of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. Instead of explicitly forming the latter product, a link with the cosine-sine decomposition, which can be computed by Van Loan's method, is exploited. Some numerical examples are given to show the performance of the presented method.

Key words. singular value decomposition, restricted singular value decomposition, cosine-sine decomposition, QR-factorization, URV decomposition

AMS subject classifications. 65F15, 65H15

PII. S0895479898346983

1. Introduction.

In past years, several generalizations of the singular value decomposition (SVD), related to a sequence of matrices in product/quotient form, have been proposed and their properties analyzed. These generalized SVDs (GSVDs) are essential numerical linear algebraic tools in signal processing and identification. Possible applications include source separation, stochastic realization, generalized Gauss–Markov estimation problems, generalized total linear least squares problems, open and closed loop balancing, etc. [7, 6].

Most well known are the product SVD (PSVD) (discussed in [11, 13]), the quotient SVD (QSVD) (introduced in [17] and refined in [14, 21]), and the restricted SVD (RSVD) (introduced in its explicit form in [19] and further developed and discussed in [4, 5]). The PSVD of a pair of matrices A, B is related to the SVD of A^TB, the QSVD of a pair of matrices A, C corresponds to the SVD of AC^{-1} if C is nonsingular, and the RSVD of a matrix triplet A, B, C shows the SVD of $B^{-1}AC^{-1}$ if B and C are nonsingular.

*Received by the editors November 6, 1998; accepted for publication (in revised form) by L. Eldén May 30, 2000; published electronically September 7, 2000. This research was partially supported by the Flemish Government through (1) the Research Council K.U. Leuven: Concerted Research Actions GOA-MIPS (Model-Based Information Processing Systems) and GOA-MEFISTO-666 (Mathematical Engineering for Information and Communication Systems Technology), (2) the Fund for Scientific Research–Flanders (F.W.O.) project G.0256.97 (Numerical Algorithms for Subspace System Identification, Extension to Special Cases), (3) the F.W.O. Research Communities ICCoS (Identification and Control of Complex Systems) and ANMMM (Advanced Numerical Methods for Mathematical Modelling); and by the Belgian State, Prime Minister's Office—Federal Office for Scientific, Technical and Cultural Affairs through the Interuniversity Poles of Attraction Programmes IUAP P4-02 (Modeling, Identification, Simulation and Control of Complex Systems) and IUAP P4-24 (Intelligent Mechatronic Systems (IMechS)). The scientific responsibility is assumed by the authors.

http://www.siam.org/journals/simax/22-2/34698.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@math.nus.edu.sg). This author was a visiting researcher with the K.U. Leuven during part of this research.

‡Department of Electrical Engineering, Research Group SISTA, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium (lieven.delathauwer@esat.kuleuven.ac.be, bart.demoor@esat.kuleuven.ac.be)

As far as the GSVDs related to a matrix inverse is concerned, the calculation of the QSVD has been extensively studied. A key idea, which is also exploited in this paper, is the link between the cosine-sine decomposition (CSD) [12] of a partitioned column-orthogonal matrix $\left[\begin{smallmatrix} A \\ C \end{smallmatrix}\right]$ and the QSVD of the couple (A, C): if the CSD of a column-orthogonal matrix $\left[\begin{smallmatrix} A \\ C \end{smallmatrix}\right]$, with $A \in \mathbf{R}^{n\times m}$ and $C \in \mathbf{R}^{p\times m}$, where we assume that $n \geq m$, is given by

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I_{n-p} & 0 \\ 0 & C \\ 0 & -S \end{bmatrix} W,$$

in which $U, W \in \mathbf{R}^{n\times n}$ and $V \in \mathbf{R}^{p\times p}$ are orthogonal and

$$\begin{aligned} C &= \operatorname{diag}(c_1, \ldots, c_p) \in \mathbf{R}^{p\times p}, & c_i &= \cos\theta_i, \\ S &= \operatorname{diag}(s_1, \ldots, s_p) \in \mathbf{R}^{p\times p}, & s_i &= \sin\theta_i, \end{aligned}$$

with $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_j \leq \pi/2$, then we have that the SVD of $AC^\dagger$ is given by

$$AC^\dagger = -UCS^\dagger W^T.$$

(An analogous link can be derived for the case $n \leq m$.) This idea forms the basis for two backward stable algorithms proposed by Stewart [16] and Van Loan [18]. Recently, Zha has studied the computation of the QSVD via the CSD and the Lanczos bidiagonalization process. The Kogbetliantz algorithm has been generalized for the computation of the QSVD by Paige [15]; his elegant implementation initially transforms A and C into a pair of triangular matrices and preserves the triangular form in the iterative phase by using suitable plane rotations. A variation of Paige's algorithm, also backward stable, is given by Bai and Demmel [2]. This technique is implemented as the LAPACK procedure STGSJA() [1]. High relative accuracy Jacobi-type algorithms for the PSVD and the QSVD are developed in Drmač [8, 9, 10].

For the computation of the RSVD, the Kogbetliantz algorithm has been generalized by Zha [20]. However, it is well known that Jacobi-type algorithms, which use Jacobi transformations, typically have a higher complexity than QR-type algorithms, which are based on QR-factorizations [3, 12, 15]. So far, nobody has proposed a QR-type algorithm for the RSVD based on the CSD. In this paper, we show that such an algorithm can in fact be derived in an easy way.

The paper is organized as follows. In section 2 the RSVD theorem is recalled. In section 3 we present our numerical method for computing the RSVD. Some numerical examples are given in section 4. Conclusions are drawn in section 5.

2. The RSVD theorem. The RSVD is a simultaneous decomposition of three matrices with compatible dimensions to quasi-diagonal forms, i.e., to matrices consisting of diagonal and zero blocks with at most one nonvanishing diagonal block per block row and column. In this section we briefly recall its definition and importance.

Given matrices A, B, C with compatible size, we denote

$$\begin{aligned} &r_a = \operatorname{rank}(A), \quad r_b = \operatorname{rank}(B), \quad r_c = \operatorname{rank}(C), \\ &r_{ab} = \operatorname{rank}\begin{bmatrix} A & B \end{bmatrix}, \quad r_{ac} = \operatorname{rank}\begin{bmatrix} A \\ C \end{bmatrix}, \quad r_{abc} = \operatorname{rank}\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}, \\ &k_1 = r_{abc} - r_b - r_c, \quad k_2 = r_{ac} + r_b - r_{abc}, \quad k_3 = r_{ab} + r_c - r_{abc}, \\ &\mu = r_{abc} - r_{ab}, \quad \nu = r_{abc} - r_{ac}, \quad \eta = r_{abc} + r_a - r_{ab} - r_{ac}. \end{aligned}$$

The RSVD is now described in the following theorem.

THEOREM 2.1 (RSVD theorem). *Given $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$, there exist nonsingular matrices $X \in \mathbf{R}^{n\times n}, Y \in \mathbf{R}^{m\times m}$ and orthogonal matrices $U \in \mathbf{R}^{l\times l}, V \in \mathbf{R}^{p\times p}$ such that*

$$XAY = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{array} \\ \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & S_A & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$XBU = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{array} \begin{array}{c} \begin{array}{ccc} l-r_b & k_2 & \nu \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \tag{2.1}$$

$$VCY = \begin{array}{c} \\ p-r_c \\ k_3 \\ \mu \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \end{array},$$

in which $S_A = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}$, with $\Sigma \in \mathbf{R}^{\eta\times\eta}$ positive diagonal.

Equation (2.1) can be interpreted as a decomposition of A, relative to the column space of B and the row space of C. With an appropriate partitioning of X^{-1} and Y^{-1} it can be written as

$$A = (X^{-1})_1(Y^{-1})_1 + (X^{-1})_2(Y^{-1})_2 + (X^{-1})_3(Y^{-1})_3 + (X^{-1})_4\Sigma(Y^{-1})_4,$$

in which the four terms can geometrically be classified as follows:

	In column space B	Not in column space B
In row space C	$(X^{-1})_4\Sigma(Y^{-1})_4$	$(X^{-1})_3(Y^{-1})_3$
Not in row space C	$(X^{-1})_2(Y^{-1})_2$	$(X^{-1})_1(Y^{-1})_1$

Obviously, the term $(X^{-1})_4\Sigma(Y^{-1})_4$ represents the *restriction* of the linear operator represented by the matrix A to the column space of B and the row space of C; the term $(X^{-1})_1(Y^{-1})_1$ is the restriction of A to the orthogonal complements of these spaces, and so on.

The decomposition can also be seen as an ordinary SVD with different inner products in row and column space. Consider the maximization of the bilinear form $\phi(x, y) = x^T Ay$ over all vectors x, y subject to $x^T BB^T x = 1$ and $y^T C^T Cy = 1$. Assuming $\eta \neq 0$, one can show that the maximum is equal to the largest diagonal element of Σ and the optimizing vectors are $x = x_1$ (first column vector of X) and $y = y_1$ (first column vector of Y). ($\eta = 0$ is a special case, in which either the maximum is 0 or the norm constraints cannot be satisfied.) Other extrema of the objective function, constrained to lie in subspaces that are orthogonal to x_1 with respect to the inner product defined by BB^T and orthogonal to y_1 with respect to the

inner product defined by C^TC, can be found in an obvious recursive manner. These extrema correspond to the columns of X and Y.

Considering the fact that knowledge of the column space of B and the row space of C is present in (2.1), it comes as no surprise that the RSVD plays a crucial role in a basic problem such as the analysis of the rank of $A + BDC$ for varying $D \in \mathbf{R}^{l\times p}$; also the influence of changing D on the rank of

$$M = \begin{bmatrix} A & B \\ C & D^T \end{bmatrix}$$

can be examined by means of the RSVD. Another application is the minimization of $\|y\|^2 + \|x\|^2$ over all vectors x, y, z satisfying

$$b = Ax + By, \quad z = Cx,$$

where A, B, C, b are given. In this variant of the conventional linear least squares problem the matrix B allows us to take into account the geometrical distribution of the noise. For example, the error term By is restricted to the column space of B; other components of the observation vector b are considered as error-free. The matrix C represents a weighting of the components of x, e.g., due to a priori information that some components are more likely or less costly than others.

The RSVD was introduced in [19]. A discussion of its properties and applications can be found in [5].

3. Computation of the RSVD via the CSD. [19] contained a constructive proof of Theorem 2.1. However, as pointed out in [19], this procedure is not useful from a numerical point of view, since in this constructive proof nonorthogonal transformations are used to scale nonsingular matrices to identity matrices and to eliminate certain submatrices in some *intermediate steps* during the decomposition. This will cause numerical instability if the underlying matrices or the pivoting matrices are ill-conditioned. In [20], Zha generalized the Kogbetliantz algorithm to compute the RSVD. In this section we show that the RSVD can be computed via the CSD.

Our method consists of three different stages. Given an arbitrary matrix triplet A, B, C, the first stage is a preprocessing QR reduction step which produces a possibly lower-dimensional triplet $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular (section 3.1). This preprocessing indicates very clearly how we can get the nonsingular matrices X and Y in the RSVD (2.1). The second stage is the actual calculation of the SVD of matrix $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ via the CSD (section 3.2). The third stage consists of the backtransformation of the results to the original matrix spaces (section 3.3). An outline of the overall algorithm is presented in section 3.4.

3.1. Preprocessing reduction. Before we give our preprocessing reduction for the RSVD, we need to recall the QR-factorization with column pivoting and URV decomposition [12], which will be the building blocks of our constructive proof of Lemmas 3.1 and 3.2 below.

It is well known that any matrix $A \in \mathbf{R}^{m\times n}$ can be factorized as

$$UA = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} \Pi, \tag{3.1}$$

where U and Π are orthogonal matrix and permutation matrix, respectively, R_1 is nonsingular and upper triangular. The factorization (3.1) is called the QR-factorization

of A with column pivoting. Note that we do not need memory to store the matrix Π during the computation of (3.1). Moreover, if we denote

$$R := \begin{bmatrix} R_1 & R_2 \end{bmatrix} \Pi,$$

then R is of full row rank and

$$UA = \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

If we continue to squeeze $\begin{bmatrix} R_1 & R_2 \end{bmatrix}$ into upper triangular form by applying a sequence of Householder transformations, then we have the following URV decomposition of A, i.e., we get an orthogonal matrix V such that

$$UAV = \begin{bmatrix} \mathcal{R} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.2}$$

with $\mathcal{R}$ nonsingular and upper triangular.

To compute the RSVD (2.1) of the triplet A, B, and C, first we reduce A, B, and C to lower dimensional submatrices $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular, applying orthogonal transformations to compress the rows and/or columns of certain matrices in such a way that the RSVD of A, B, C can easily be derived from the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. The idea is that a compression of the rows and/or columns of the matrices that are involved allows for a more efficient formulation of the actual core computations.

Preprocessing reduction step is necessary for computing QSVD and RSVD; see [2, 3, 15, 20]. In all of the above-cited papers and in our paper, the QR-factorization with column pivoting and the URV decomposition are used to compress the rows and/or columns of a matrix [12]. (Of course, the SVD method can also be used to compress rows and/or columns of a matrix, but it is much more expensive.) The rank determination of a matrix is an ill-conditioned problem, which always arises in the computation of the QSVD and RSVD. The common and effective tools are QR-factorization with column pivoting, URV decomposition, and SVD.

We will now state two lemmas and one theorem on which our preprocessing step is based. These three results describe how the given matrices can be brought in a desired form by means of orthogonal transformations. Lemma 3.1 deals with two matrices; it is a tool to prove Lemma 3.2, which discusses the actual situation of three matrices. Theorem 3.4 is a refinement of Lemma 3.2.

LEMMA 3.1. *Given matrices $A \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{p\times m}$, there exist three orthogonal matrices $U \in \mathbf{R}^{n\times n}, W \in \mathbf{R}^{m\times m}, V \in \mathbf{R}^{p\times p}$ such that*

$$\begin{aligned} UAW &= \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & 0 \end{bmatrix} \end{matrix}, \\ VCW &= \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{22} & 0 \end{bmatrix} \end{matrix}, \end{aligned} \tag{3.3}$$

where A_{11} and C_{22} are nonsingular.

Proof. We prove Lemma 3.1 constructively by the following algorithm.

ALGORITHM 1.

Input: $A \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{p\times m}$.

Output: *Orthogonal matrices* $U \in \mathbf{R}^{n\times n}, V \in \mathbf{R}^{p\times p}, W \in \mathbf{R}^{m\times m}$ *and the form* (3.3).

Step 1. *Perform a URV decomposition of* C *to get orthogonal matrices* V *and* W *such that*

$$VCW =: \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} m-r_c & r_c \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & C_{22} \end{bmatrix} \end{matrix}$$

with C_{22} *nonsingular. Set*

$$AW =: \begin{matrix} \begin{matrix} m-r_c & r_c \end{matrix} \\ \begin{bmatrix} A_1 & A_2 \end{bmatrix} \end{matrix}.$$

Step 2. *Perform a URV decomposition of* A_1 *(note that* $\operatorname{rank}(A_1) = r_{ac} - r_c$*) to get orthogonal matrices* U *and* W_2 *such that*

$$UA_1W_2 =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix} \end{matrix}$$

with A_{11} *nonsingular. Set*

$$UA_2 =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix}, \; W := W \begin{bmatrix} W_2 & \\ & I_{r_c} \end{bmatrix} \begin{bmatrix} I & & \\ & 0 & I_{m-r_{ac}} \\ & I_{r_c} & \end{bmatrix}.$$

Step 3. *Output* U, V, W *and* UAW, VCW.

It is easy to see that (U^TAW, V^TCW) are in the form (3.3). □

Based on Lemma 3.1, we have the following lemma.

LEMMA 3.2. *Given* $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$, *there exist orthogonal matrices* $P \in \mathbf{R}^{n\times n}, Q \in \mathbf{R}^{m\times m}, U_b \in \mathbf{R}^{l\times l}, V_c \in \mathbf{R}^{p\times p}$ *such that*

$$PAQ = \begin{matrix} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & 0 \\ 0 & A_{22} & A_{23} & A_{24} & 0 \\ 0 & 0 & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix},$$

$$PBU_b = \begin{matrix} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} l-r_b & k_2 & \nu \end{matrix} \\ \begin{bmatrix} 0 & 0 & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & 0 \\ 0 & 0 & B_{43} \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}, \tag{3.4}$$

$$V_cCQ = \begin{matrix} \\ p-r_c \\ k_3 \\ \mu \end{matrix} \begin{matrix} \begin{matrix} k_1 & k_2 & k_3 & \mu & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{23} & 0 & 0 \\ 0 & 0 & C_{33} & C_{34} & 0 \end{bmatrix} \end{matrix},$$

where A_{11}, A_{22}, A_{33}, B_{22}, B_{43}, C_{23}, *and* C_{34} *are nonsingular.*

Proof. Similarly to the proof of Lemma 3.1, we prove Lemma 3.2 constructively by the following algorithm.

ALGORITHM 2.

Input: $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$.

Output: *Orthogonal matrices* $P \in \mathbf{R}^{n\times n}, Q \in \mathbf{R}^{m\times m}, U_b \in \mathbf{R}^{l\times l}, V_c \in \mathbf{R}^{p\times p}$ *and the form* (3.4).

Step 1. *Apply Algorithm* 1 *to* (A, C) *to get orthogonal matrices* P, V_c, *and* Q *such that*

$$PAQ =: \begin{matrix} \\ r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} A_{11} & A_{13} & 0 \\ 0 & A_{33} & 0 \end{bmatrix} \end{matrix},$$

$$V_cCQ =: \begin{matrix} \\ p-r_c \\ r_c \end{matrix} \begin{matrix} \begin{matrix} r_{ac}-r_c & r_c & m-r_{ac} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{23} & \end{bmatrix} \end{matrix},$$

where A_{11} *and* C_{23} *are nonsingular. Denote*

$$PB =: \begin{matrix} r_{ac}-r_c \\ n+r_c-r_{ac} \end{matrix} \begin{bmatrix} B_1 \\ B_3 \end{bmatrix}.$$

Then we have that $\operatorname{rank}\begin{bmatrix} A_{33} & B_3 \end{bmatrix} = r_{ab}+r_c-r_{ac} =: t$.

Step 2. *Apply Algorithm* 1 *to* (A_{33}^T, B_3^T) *to get orthogonal matrices* U_b, P_2, *and* Q_2 *such that*

$$P_2A_{33}Q_2 =: \begin{matrix} \\ t-x \\ x \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{33} & 0 \\ A_{43} & A_{44} \\ 0 & 0 \end{bmatrix} \end{matrix}, \quad P_2B_3U_b =: \begin{matrix} \\ t-x \\ x \\ n-r_{ab} \end{matrix} \begin{matrix} \begin{matrix} l-x & x \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & B_{43} \\ 0 & 0 \end{bmatrix} \end{matrix}$$

with A_{33} *and* B_{43} *nonsingular. Set*

$$B_1U_b =: \begin{matrix} \begin{matrix} l-x & x \end{matrix} \\ \begin{bmatrix} B_{11} & B_{13} \end{bmatrix} \end{matrix}, \quad A_{13}Q_2 =: \begin{matrix} \begin{matrix} t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{13} & A_{14} \end{bmatrix} \end{matrix}, \quad C_{23}Q_2 =: \begin{matrix} \begin{matrix} t-x & x \end{matrix} \\ \begin{bmatrix} C_{23} & C_{24} \end{bmatrix} \end{matrix},$$

$$P := \begin{bmatrix} I_{r_{ac}-r_c} & \\ & P_2 \end{bmatrix} P, \quad Q := Q \begin{bmatrix} I_{r_{ac}-r_c} & & \\ & Q_2 & \\ & & I_{m-r_{ac}} \end{bmatrix}.$$

We have that $\begin{bmatrix} C_{23} & C_{24} \end{bmatrix}$ *is nonsingular.*

Step 3. *Perform a QR-factorization of* C_{24} *with column pivoting to get orthogonal matrix* V_3 *such that*

$$V_3C_{24} =: \begin{matrix} t-x \\ r_c+x-t \end{matrix} \begin{bmatrix} 0 \\ C_{34} \end{bmatrix}$$

with C_{34} *nonsingular. Set*

$$V_3C_{23} =: \begin{matrix} t-x \\ r_c+x-t \end{matrix} \begin{bmatrix} C_{23} \\ C_{33} \end{bmatrix}, \quad V_c := \begin{bmatrix} I_{p-r_c} & \\ & V_3 \end{bmatrix} V_c.$$

Here, C_{23} *is nonsingular.*

Step 4. *Perform a URV decomposition of* B_{11} *(note that* $\operatorname{rank}(B_{11}) = r_b - x$*) to get orthogonal matrices* P_4 *and* U_4 *such that*

$$P_4 B_{11} U_4 =: \begin{matrix} \\ r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{matrix} \begin{matrix} l-r_b & r_b-x \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & B_{22} \end{bmatrix} \end{matrix}$$

with B_{22} *nonsingular. Set*

$$P_4 B_{13} =: \begin{matrix} r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{bmatrix} B_{13} \\ B_{23} \end{bmatrix},$$

$$P_4 \begin{bmatrix} A_{11} & A_{13} & A_{14} \end{bmatrix} =: \begin{matrix} \\ r_{ac}+x-r_b-r_c \\ r_b - x \end{matrix} \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x & t-x & r_c+x-t \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \end{bmatrix} \end{matrix},$$

$$P := \begin{bmatrix} P_4 & \\ & I_{n+r_c-r_{ac}} \end{bmatrix}, \quad U_b := U_b \begin{bmatrix} U_4 & \\ & I_x \end{bmatrix}.$$

We know that $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ *is nonsingular.*

Step 5. *Perform a QR-factorization of* $\begin{bmatrix} A_{21} & A_{22} \end{bmatrix}^T$ *with column pivoting to get orthogonal matrix* Q_5 *such that*

$$\begin{bmatrix} A_{21} & A_{22} \end{bmatrix} Q_5 =: \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x \end{matrix} \\ \begin{bmatrix} 0 & A_{22} \end{bmatrix} \end{matrix}$$

with A_{22} *nonsingular. Set*

$$\begin{bmatrix} A_{11} & A_{12} \end{bmatrix} Q_5 =: \begin{matrix} \begin{matrix} r_{ac}+x-r_b-r_c & r_b-x \end{matrix} \\ \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \end{matrix}, \quad Q := Q \begin{bmatrix} Q_5 & \\ & I_{m+r_c-r_{ac}} \end{bmatrix}.$$

It is easy to see that A_{11} *is nonsingular.*

Step 6. *Output* $P, Q, U_b, V_c, PAQ, PBU_b$*, and* $V_c CQ$*.*

A simple calculation yields that $r_{abc} = r_{ac} + x$, i.e., $x = r_{abc} - r_{ac}$. Hence, matrices PAQ, PBU_b, and $V_c CQ$ are in the form (3.4). ∎

Algorithm 2 is implemented using only orthogonal transformations; hence, it is numerically stable.

The reduction procedure above is similar to the one proposed in Zha [20], but it is more convenient for the computation of the RSVD, because the condensed form (3.4) gives an explicit way to get the nonsingular matrix X and Y in the RSVD (2.1). Note that in Algorithm 2

$$P_2 \in \mathbf{R}^{(n+r_c-r_{ac})\times(n+r_c-r_{ac})}, \quad Q_2, V_3 \in \mathbf{R}^{r_c \times r_c},$$
$$U_4 \in \mathbf{R}^{(l+r_{ac}-r_{abc})\times(l+r_{ac}-r_{abc})}, \quad P_4, Q_5 \in \mathbf{R}^{(r_{ac}-r_c)\times(r_{ac}-r_c)},$$
$$n + r_c - r_{ac} \le n, \quad r_{ac} - r_c \le r_a \le n, \quad r_c \le m,\ l + r_{ac} - r_{abc} \le l,$$

so, besides the necessary arrays for matrices A, B, C, P, Q, U_b, and V_c, two extra arrays of at most $n \times n$ and $\max(l,m) \times \max(l,m)$ orders are enough for the temporal storage of the "temporal" matrices Q_2, P_2, V_3, P_4, U_4, and Q_5. So, the memory requirement of Algorithm 2 is the same as that in the reduction procedure of [20]; hence, it is acceptable.

In order to get the nonsingular matrices X and Y in the RSVD (2.1) we have to eliminate submatrices A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, A_{43}, B_{13}, B_{23}, and C_{33} in (3.4). Similar to Zha [20], these submatrices can be eliminated using nonsingular matrices A_{11}, A_{22}, A_{33}, B_{43}, and C_{34} by solving some coupled linear matrix equations as follows:

$$X_1 B_{43} = \begin{bmatrix} B_{13} \\ B_{23} \end{bmatrix}, \tag{3.5}$$

$$A_{11} Y_1 = A_{12}, \ \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} Y_2 = \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix} - X_1 \begin{bmatrix} A_{43} & A_{44} \end{bmatrix}, \tag{3.6}$$

$$C_{34} Y_3 = C_{33}, \ X_2 A_{33} = A_{43} - A_{44} Y_3. \tag{3.7}$$

However, if some of matrices A_{11}, A_{22}, A_{33}, B_{43}, and C_{34} are highly ill-conditioned, the numerical computations of linear matrix equations (3.5), (3.6), and (3.7) will encounter numerical difficulties and lead to large computational errors. Fortunately, we will show that we can refine (3.4) and eliminate submatrices A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, B_{13}, B_{23}, and C_{33} using only orthogonal transformations such as QR-factorization with column pivoting.

LEMMA 3.3. *Given matrices*

$$\mathcal{A}_{11} \in \mathbf{R}^{n_1 \times n_1}, \quad \mathcal{A}_{12} \in \mathbf{R}^{n_1 \times n_2}, \quad \mathcal{A}_{21} \in \mathbf{R}^{\tilde{n}_2 \times n_1}, \quad \mathcal{A}_{22} \in \mathbf{R}^{\tilde{n}_2 \times n_2}$$

with $\mathcal{A}_{11}$ nonsingular, let $\mathcal{P}_{21} \in \mathbf{R}^{\tilde{n}_2 \times n_1}$ and $\mathcal{P}_{22} \in \mathbf{R}^{\tilde{n}_2 \times \tilde{n}_2}$ satisfy

$$\begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}^T = I_{\tilde{n}_2}.$$

Denote

$$\begin{bmatrix} \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \tilde{\mathcal{A}}_{22}.$$

Then $\mathcal{P}_{22}$ is nonsingular,

$$\begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{12} \\ \tilde{\mathcal{A}}_{22} \end{bmatrix}, \tag{3.8}$$

and, furthermore,

$$\mathcal{A}_{22} - \mathcal{A}_{21} \mathcal{A}_{11}^{-1} \mathcal{A}_{12} = (\mathcal{P}_{22}^T - \mathcal{P}_{12}^T \mathcal{P}_{11}^{-T} \mathcal{P}_{21}^T) \tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}^{-1} \tilde{\mathcal{A}}_{22}. \tag{3.9}$$

Consequently, if $\mathcal{A}_{12} = 0$, $\tilde{n}_2 = n_2$, and $\mathcal{A}_{22}$ is nonsingular, then $\tilde{\mathcal{A}}_{22}$ is also nonsingular.

Proof. Equation (3.8) is obvious. Let $\mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}$ be orthogonal and denote

$$\begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \tilde{\mathcal{A}}_{12}.$$

Then we have

$$\mathcal{A}_{11} = P_{11}^T \tilde{\mathcal{A}}_{11}.$$

Hence, $\mathcal{P}_{11}$ and $\tilde{\mathcal{A}}_{11}$ are nonsingular. Note that $\mathcal{P}$ is orthogonal, so $\mathcal{P}_{22}$ is also nonsingular. Moreover,

$$\mathcal{P}_{21}\mathcal{P}_{11}^T + \mathcal{P}_{22}\mathcal{P}_{12}^T = 0, \quad \mathcal{P}_{21}\mathcal{P}_{21}^T + \mathcal{P}_{22}\mathcal{P}_{22}^T = I,$$

which, with the nonsingularity of $\mathcal{P}_{11}$ and $\mathcal{P}_{22}$, gives

$$\begin{aligned} \mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T &= \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T - \mathcal{P}_{22}\mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T) \\ &= \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T + \mathcal{P}_{21}\mathcal{P}_{21}^T) \\ &= \mathcal{P}_{22}^{-1}. \end{aligned} \tag{3.10}$$

On other hand, we also have

$$\mathcal{A}_{21} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{11}, \quad \mathcal{A}_{22} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{22}^T\tilde{\mathcal{A}}_{22}, \quad \mathcal{A}_{12} = \mathcal{P}_{11}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{21}^T\tilde{\mathcal{A}}_{22}.$$

Hence, we have

$$\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} = (\mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T)\tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}^{-1}\tilde{\mathcal{A}}_{22},$$

i.e., (3.9) is true. Moreover, if $\mathcal{A}_{12} = 0, \tilde{n}_2 = n_2$ and $\mathcal{A}_{22}$ is nonsingular, then, $\tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}\mathcal{A}_{22}$ is nonsingular. □

In general, for matrix $\mathcal{A}_{ij}(i,j = 1,2)$ with suitable sizes, if $\mathcal{A}_{11}$ is nonsingular but very ill-conditioned, the computation of $\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12}$ will not be numerically stable. Fortunately, (3.9) in the proof of Lemma 3.3 gives that we can cancel "the instability factor" of $\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12}$ by multiplying $\mathcal{P}_{22}$. Now, although $\begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}$ is not orthogonal, it and the form (3.8) are computed by only orthogonal transformations, for example, the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix}$ with column pivoting.

If we apply Lemma 3.3 to the form (3.4), we have the following result.

THEOREM 3.4. *Given $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, and $C \in \mathbf{R}^{p\times m}$, there exist nonsingular matrices $\mathcal{X} \in \mathbf{R}^{n\times n}$, $\mathcal{Y} \in \mathbf{R}^{m\times m}$ and orthogonal matrices $U_b \in \mathbf{R}^{l\times l}$ and $V_c \in \mathbf{R}^{p\times p}$ such that*

$$\mathcal{X}A\mathcal{Y} = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n - r_{ab} \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m - r_{ac} \end{array} \\ \begin{bmatrix} \mathcal{A}_{11} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{A}_{22} & 0 & 0 & 0 \\ 0 & 0 & \mathcal{A}_{33} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{A} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$\mathcal{X}BU_b = \begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \\ \nu \\ n - r_{ab} \end{array} \begin{array}{c} \begin{array}{ccc} l - r_b & k_2 & \nu \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathcal{B}_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathcal{B} \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \tag{3.11}$$

$$V_cC\mathcal{Y} = \begin{array}{c} \\ p - r_c \\ k_3 \\ \mu \end{array} \begin{array}{c} \begin{array}{ccccc} k_1 & k_2 & k_3 & \mu & m - r_{ac} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{C}_{23} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{C} & 0 \end{bmatrix} \end{array},$$

where $\mathcal{A}_{11}, \mathcal{A}_{22}, \mathcal{A}_{33}$, $\mathcal{B}_{22}$, $\mathcal{C}_{23}$, $\mathcal{B}$, *and* $\mathcal{C}$ *are all nonsingular. Moreover,* $\mathcal{X}$, $\mathcal{Y}$, *and the condensed form* (3.11) *are computed using only orthogonal transformations which are numerically stable.*

Proof. We prove Theorem 3.4 constructively by the following algorithm.

ALGORITHM 3.

Input: $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, *and* $C \in \mathbf{R}^{p\times m}$.

Output: *Nonsingular matrices* $\mathcal{X} \in \mathbf{R}^{n\times n}$ *and* $\mathcal{Y} \in \mathbf{R}^{m\times m}$, *orthogonal matrices* $U_b \in \mathbf{R}^{l\times l}$ *and* $V_c \in \mathbf{R}^{p\times p}$, *and the form* (3.11).

Step 1. *Perform Algorithm* 2 *to get orthogonal matrices* P, Q, U_b, *and* V_c *and the form* (3.4). *Set*

$$\mathcal{X} := P, \quad \mathcal{Y} =: Q.$$

Step 2. *Perform the QR-factorization of* $\left[\begin{smallmatrix} B_{13} \\ B_{43} \end{smallmatrix}\right]$ *with column pivoting to get* $\mathcal{U}_{11}^{(2)} \in \mathbf{R}^{k_1\times k_1}$ *and* $\mathcal{U}_{12}^{(2)} \in \mathbf{R}^{k_1\times\nu}$ *such that*

$$\begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} B_{13} \\ B_{43} \end{bmatrix} =: 0, \quad \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix}^T = I_{k_1}.$$

Set

$$\mathcal{A}_{11} := \mathcal{U}_{11}^{(2)} A_{11}, \quad A_{12} := \mathcal{U}_{11}^{(2)} A_{12}, \quad \begin{matrix} k_3 & \mu \end{matrix} \\ \begin{bmatrix} A_{13} & A_{14} \end{bmatrix} := \begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \begin{bmatrix} A_{13} & A_{14} \\ A_{43} & A_{44} \end{bmatrix},$$

and

$$\mathcal{X} := \begin{bmatrix} \mathcal{U}_{11}^{(2)} & 0 & \mathcal{U}_{12}^{(2)} & 0 \\ 0 & I_{k_2+k_3} & 0 & 0 \\ 0 & 0 & I_\nu & 0 \\ 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix} \mathcal{X}.$$

Since A_{11} *is nonsingular, Lemma* 3.3 *gives that* $\mathcal{A}_{11}$ *is also nonsingular.*

Step 3. *Perform the QR-factorization of* $\left[\begin{smallmatrix} B_{23} \\ B_{43} \end{smallmatrix}\right]$ *with column pivoting to get* $\mathcal{U}_{11}^{(3)} \in \mathbf{R}^{k_2\times k_2}$ *and* $\mathcal{U}_{12}^{(3)} \in \mathbf{R}^{k_2\times\nu}$ *such that*

$$\begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} B_{23} \\ B_{43} \end{bmatrix} =: 0, \quad \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix}^T = I_{k_2}.$$

Set

$$\begin{matrix} k_3 & \mu \end{matrix} \\ \begin{bmatrix} A_{23} & A_{24} \end{bmatrix} := \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \begin{bmatrix} A_{23} & A_{24} \\ A_{43} & A_{44} \end{bmatrix}, \quad \tilde{\mathcal{A}}_{22} := \mathcal{U}_{11}^{(3)} A_{22}, \quad \mathcal{B}_{22} := \mathcal{U}_{11}^{(3)} B_{22},$$

and

$$\mathcal{X} := \begin{bmatrix} I_{k_1} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{U}_{11}^{(3)} & 0 & \mathcal{U}_{12}^{(3)} & 0 \\ 0 & 0 & I_{k_3} & 0 & 0 \\ 0 & 0 & 0 & I_\nu & 0 \\ 0 & 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix} \mathcal{X}.$$

Since A_{22} and B_{22} are nonsingular, Lemma 3.3 gives that $\tilde{\mathcal{A}}_{22}$ and $\mathcal{B}_{22}$ are also nonsingular.

Step 4. Perform the QR-factorization of $\begin{bmatrix} C_{33} & C_{34} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{11}^{(4)} \in \mathbf{R}^{k_3 \times k_3}$ *and* $\mathcal{V}_{21}^{(4)} \in \mathbf{R}^{\mu \times k_3}$ *such that*

$$\begin{bmatrix} C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} = I_{k_3}.$$

Denote

$$\begin{array}{c} \\ k_1 \\ k_2 \\ \nu \end{array} \begin{array}{c} \begin{array}{cc} k_3 & \mu \end{array} \\ \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \\ A_{43} & A_{44} \end{bmatrix} \end{array} := \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \\ A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix}, \quad \tilde{\mathcal{A}}_{33} := A_{33}\mathcal{V}_{11}^{(4)}, \quad \tilde{\mathcal{C}}_{23} := C_{23}\mathcal{V}_{11}^{(4)},$$

and

$$\mathcal{Y} := \mathcal{Y} \begin{bmatrix} I_{k_1+k_2} & 0 & 0 & 0 \\ 0 & \mathcal{V}_{11}^{(4)} & 0 & 0 \\ 0 & \mathcal{V}_{21}^{(4)} & I_\mu & 0 \\ 0 & 0 & 0 & I_{m-r_{ac}} \end{bmatrix}.$$

A_{33} and C_{23} are nonsingular, so we have from Lemma 3.3 that $\tilde{\mathcal{A}}_{33}$ and $\tilde{\mathcal{C}}_{23}$ are nonsingular.

Step 5. Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{14} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{24} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{12}^{(5)} \in \mathbf{R}^{(k_1+k_2)\times\mu}$ *and* $\mathcal{V}_{22}^{(5)} \in \mathbf{R}^{\mu\times\mu}$ *such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{14} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{24} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{12}^{(5)} \\ \mathcal{V}_{22}^{(5)} \end{bmatrix} = I_\mu.$$

Denote

$$\tilde{\mathcal{A}}_{44} := A_{44}\mathcal{V}_{22}^{(5)}, \quad \mathcal{C} := C_{34}\mathcal{V}_{22}^{(5)},$$

and

$$\mathcal{Y} := \mathcal{Y} \begin{bmatrix} I_{k_1+k_2} & 0 & \mathcal{V}_{12}^{(5)} & 0 \\ 0 & I_{k_3} & 0 & 0 \\ 0 & 0 & \mathcal{V}_{22}^{(5)} & 0 \\ 0 & 0 & 0 & I_{m-r_{ac}} \end{bmatrix}.$$

Since C_{34} is nonsingular, by Lemma 3.3, $\mathcal{C}$ is nonsingular.

Step 6. Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{13} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{23} \end{bmatrix}^T$ *with column pivoting to get* $\mathcal{V}_{12}^{(6)} \in \mathbf{R}^{(k_1+k_2)\times k_3}$ *and* $\mathcal{V}_{22}^{(6)} \in \mathbf{R}^{k_3\times k_3}$ *such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} & A_{13} \\ 0 & \tilde{\mathcal{A}}_{22} & A_{23} \end{bmatrix} \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix}^T \begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} = I_{k_3}.$$

Denote

$$\mathcal{A}_{33} := \tilde{\mathcal{A}}_{33}\mathcal{V}_{22}^{(6)}, \quad A_{43} := A_{43}\mathcal{V}_{22}^{(6)}, \quad \mathcal{C}_{23} := \tilde{\mathcal{C}}_{23}\mathcal{V}_{22}^{(6)},$$

and

$$\mathcal{Y} := \mathcal{Y}\begin{bmatrix} I_{k_1+k_2} & \mathcal{V}_{12}^{(6)} & 0 \\ 0 & \mathcal{V}_{22}^{(6)} & 0 \\ 0 & 0 & I_{\mu+m-r_{ac}} \end{bmatrix}.$$

Since $\tilde{\mathcal{A}}_{33}$ and $\tilde{\mathcal{C}}_{23}$ are nonsingular, by Lemma 3.3*, $\mathcal{A}_{33}$ and $\mathcal{C}_{23}$ are nonsingular.*

Step 7. *Perform the QR-factorization of $\begin{bmatrix} \mathcal{A}_{11} & A_{12} \end{bmatrix}^T$ with column pivoting to get $\mathcal{V}_{12}^{(7)} \in \mathbf{R}^{k_1\times k_2}$ and $\mathcal{V}_{22}^{(7)}\mathbf{R}^{k_2\times k_2}$ such that*

$$\begin{bmatrix} \mathcal{A}_{11} & A_{12} \end{bmatrix}\begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix}^T\begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} = I_{k_2}.$$

Denote

$$\mathcal{A}_{22} := \tilde{\mathcal{A}}_{22}\mathcal{V}_{22}^{(7)}, \quad \mathcal{Y} := \mathcal{Y}\begin{bmatrix} I_{k_1} & \mathcal{V}_{12}^{(7)} & 0 \\ 0 & \mathcal{V}_{22}^{(7)} & 0 \\ 0 & 0 & I_{k_3+\mu+m-r_{ac}} \end{bmatrix}.$$

Note that $\tilde{\mathcal{A}}_{22}$ is nonsingular, so Lemma 3.3 *implies that $\mathcal{A}_{22}$ is nonsingular.*

Step 8. *Perform the QR factorization of $\begin{bmatrix} \mathcal{A}_{33} \\ A_{43} \end{bmatrix}$ with column pivoting to get $\mathcal{U}_{21}^{(8)} \in \mathbf{R}^{\nu\times k_3}$ and $\mathcal{U}_{22}^{(8)} \in \mathbf{R}^{\nu\times\nu}$ such that*

$$\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}\begin{bmatrix} \mathcal{A}_{33} \\ A_{43} \end{bmatrix} = 0, \quad \begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix}^T = I_\nu.$$

Denote

$$\mathcal{A} := \mathcal{U}_{22}^{(8)}\tilde{\mathcal{A}}_{44}, \quad \mathcal{B} := \mathcal{U}_{22}^{(8)}B_{43}, \quad \mathcal{X} := \begin{bmatrix} I_{k_1+k_2} & 0 & 0 & 0 \\ 0 & I_{k_3} & 0 & 0 \\ 0 & \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} & 0 \\ 0 & 0 & 0 & I_{n-r_{ab}} \end{bmatrix}.$$

Since B_{43} is nonsingular, so by Lemma 3.3*, $\mathcal{B}$ is nonsingular.*

Step 9. *Output $\mathcal{X}, \mathcal{Y}$, U_b, V_c, $\mathcal{X}A\mathcal{Y}$, $\mathcal{X}BU_b$, and $V_cC\mathcal{Y}$.*

A simple calculation using Lemma 3.3 yields that $(\mathcal{X}A\mathcal{Y}, \mathcal{X}BU_b, V_cC\mathcal{Y})$ are in the form (3.11). □

It should be pointed out that Algorithm 3 is also implemented using only orthogonal transformations, so it is numerically stable.

It is easy to see that

$$\begin{bmatrix} \mathcal{U}_{11}^{(2)} & \mathcal{U}_{12}^{(2)} \end{bmatrix} \in \mathbf{R}^{k_1\times(k_1+\nu)}, \ \begin{bmatrix} \mathcal{U}_{11}^{(3)} & \mathcal{U}_{12}^{(3)} \end{bmatrix} \in \mathbf{R}^{k_2\times(k_2+\nu)},$$

$$\begin{bmatrix} \mathcal{U}_{21}^{(8)} & \mathcal{U}_{22}^{(8)} \end{bmatrix} \in \mathbf{R}^{\nu\times(k_3+\nu)}, \tag{3.12}$$

$$\begin{bmatrix} \mathcal{V}_{11}^{(4)} \\ \mathcal{V}_{21}^{(4)} \end{bmatrix} \in \mathbf{R}^{(k_3+\mu)\times k_3}, \ \begin{bmatrix} \mathcal{V}_{11}^{(5)} \\ \mathcal{V}_{21}^{(5)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2+\mu)\times\mu}, \tag{3.13}$$

$$\begin{bmatrix} \mathcal{V}_{12}^{(6)} \\ \mathcal{V}_{22}^{(6)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2+k_3)\times k_3}, \ \begin{bmatrix} \mathcal{V}_{12}^{(7)} \\ \mathcal{V}_{22}^{(7)} \end{bmatrix} \in \mathbf{R}^{(k_1+k_2)\times k_2}, \tag{3.14}$$

$$\begin{aligned} &\max(k_1, k_2, k_3, \nu, k_1+\nu, k_2+\nu, k_3+\nu, k_1+k_2, k_1+k_2+k_3) \le n, \\ &\max(k_3, \mu, k_3+\mu, k_1+k_2+\mu) \le m. \end{aligned} \tag{3.15}$$

Thus, the arrays for P and Q in Algorithm 2 can be used for $\mathcal{X}$ and $\mathcal{Y}$ in Algorithm 3, and the two arrays for the "temporal" matrices Q_2, P_2, V_3, P_4, U_4, and Q_5 in Algorithm 2 can be used for the temporal storage of the "temporal" matrices in (3.12), (3.13), and (3.14) in Algorithm 3. Hence, the memory requirements for Algorithm 2 and Algorithm 3 are the same, i.e., when we refine the form (3.4) to get the refined form (3.11), we do not need extra memory.

Obviously, we can scale $\mathcal{A}_{11}, \mathcal{A}_{22}, \mathcal{A}_{33}$, $\mathcal{B}_{22}$, $\mathcal{B}$, $\mathcal{C}_{23}$, and $\mathcal{C}$ in (3.11) to identity matrices by SVD or QR-factorization. As with Zha [20], this step cannot be avoided since the complete RSVD is computed, just like the computation of the complete QSVD [15]. Therefore, now the main problem for the RSVD (3.3) is how to compute the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, which is the main work of the next subsection.

3.2. CSD stage. An obvious way to compute the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is first forming $\mathcal{Z} = \mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly and then computing the SVD of $\mathcal{Z}$. As is well known, this approach is not numerically stable and will lead to large numerical errors when $\mathcal{B}$ or $\mathcal{C}$ is ill-conditioned, so generally it is not reliable, and hence not recommended.

In the following we show how the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ can be computed without forming $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly by solving a CSD problem.

LEMMA 3.5. *Let $\mathcal{A} \in \mathbf{R}^{\nu\times\mu}, \mathcal{B} \in \mathbf{R}^{\nu\times\nu}, \mathcal{C} \in \mathbf{R}^{\mu\times\mu}$ with $\mathcal{B}$ and $\mathcal{C}$ nonsingular. Let $\mathcal{P}_1 \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{P}_2 \in \mathbf{R}^{\mu\times\nu}$, $\tilde{\mathcal{Q}}_1 \in \mathbf{R}^{\nu\times\mu}$, $\tilde{\mathcal{Q}}_2 \in \mathbf{R}^{\mu\times\mu}$ satisfy*

$$\mathcal{P}_1^T\mathcal{P}_1 + \mathcal{P}_2^T\mathcal{P}_2 = I_\nu, \qquad \begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} = 0,$$

$$\tilde{\mathcal{Q}}_1^T\tilde{\mathcal{Q}}_1 + \tilde{\mathcal{Q}}_2^T\tilde{\mathcal{Q}}_2 = I_\mu, \qquad \begin{bmatrix} \tilde{\mathcal{Q}}_1^T & \tilde{\mathcal{Q}}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix} = 0;$$

then we have that

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = \tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1}.$$

Proof. Let $\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2 \in \mathbf{R}^{\mu\times\mu}$ be such that $\begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix}$ is an orthogonal matrix; then

$$\begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix} \begin{bmatrix} \mathcal{R}_1 \\ 0 \end{bmatrix}$$

for some matrix $\mathcal{R}_1 \in \mathbf{R}^{\mu\times\mu}$. As $\mathcal{C}$ is nonsingular, $\tilde{\mathcal{P}}_2$ and $\mathcal{R}_1$ are nonsingular as well. Since

$$\begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = 0,$$

we have that

$$\begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{P}}_1 & \mathcal{P}_1 \\ \tilde{\mathcal{P}}_2 & \mathcal{P}_2 \end{bmatrix} \begin{bmatrix} \mathcal{R}_2 \\ 0 \end{bmatrix} \quad \text{for some matrix } \mathcal{R}_2 \in \mathbf{R}^{\mu\times\mu}.$$

Because $\mathcal{B}$ is nonsingular, thus

$$\operatorname{rank} \begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \mu.$$

Consequently, $\mathcal{R}_2$ is nonsingular, which, along with the nonsingularity of $\tilde{\mathcal{P}}_2$, also yields that $\tilde{\mathcal{Q}}_2$ is nonsingular. Hence,

$$\begin{bmatrix} \mathcal{B}\tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix} \mathcal{R}_1^{-1}\mathcal{R}_2,$$

that is,

$$\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix} \mathcal{R}_2^{-1}\mathcal{R}_1 = \begin{bmatrix} \mathcal{B}^{-1}\mathcal{A} \\ \mathcal{C} \end{bmatrix}.$$

Therefore, we have that

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = \tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1}. \qquad \square$$

According to Lemma 3.5, the CSD of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$ reveals the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. On the other hand, it is also possible to obtain the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ from the orthogonal complement of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$. To explain this, let us consider $\mathcal{Q}_1 \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{Q}_2 \in \mathbf{R}^{\mu\times\nu}$ such that

$$\mathcal{Q}_1^T\mathcal{Q}_1 + \mathcal{Q}_2^T\mathcal{Q}_2 = I_\nu, \quad \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} \mathcal{R},$$

in which $\mathcal{R}$ is nonsingular. Since $\tilde{\mathcal{Q}}_2$ is nonsingular, $\mathcal{Q}_1$ is nonsingular as well. If the CSD of $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$ is given by

$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{U} & \\ & \mathcal{V} \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \mathcal{W},$$

where $\mathcal{U}, \mathcal{W} \in \mathbf{R}^{\nu\times\nu}$, $\mathcal{V} \in \mathbf{R}^{\mu\times\mu}$ are orthogonal, $\mathcal{U}\Sigma_1\mathcal{W}$ and $\mathcal{V}\Sigma_2\mathcal{W}$ are the SVDs of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively, then we have that

$$\tilde{\mathcal{Q}}_1\tilde{\mathcal{Q}}_2^{-1} = (-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T,$$

that is,

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = (-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T. \tag{3.16}$$

Equivalently, the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by $(-\mathcal{U})(\Sigma_2\Sigma_1^{-1})\mathcal{V}^T$. $\square$

Depending on the dimensions μ and ν, we choose to base the RSVD algorithm on either $\mathcal{Q}_1$ and $\mathcal{Q}_2$, or $\tilde{\mathcal{Q}}_1$ and $\tilde{\mathcal{Q}}_2$. If $\mu < \nu$, we use $\tilde{\mathcal{Q}}_1$ and $\tilde{\mathcal{Q}}_2$. If $\mu > \nu$, we use $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Since the computation of $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$ itself is cheaper than that of $\begin{bmatrix} \tilde{\mathcal{Q}}_1 \\ \tilde{\mathcal{Q}}_2 \end{bmatrix}$, we also resort to $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in the case $\mu = \nu$.

3.3. The RSVD components. After the computation of the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, the only thing that remains to be done is the backtransformation of the results to the original matrix spaces of A, B, and C.

Assume that orthogonal matrices U_b, V_c and nonsingular matrices $\mathcal{X}, \mathcal{Y}$ satisfy (3.11), the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by (3.16), and set

$$\begin{aligned} X &= \operatorname{diag}\{\mathcal{A}_{11}^{-1}, \mathcal{B}_{22}^{-1}, \mathcal{C}_{23}\mathcal{A}_{33}^{-1}, -\mathcal{U}^T\mathcal{B}^{-1}, I_{n-r_{ab}}\}\mathcal{X}, \\ Y &= \mathcal{Y}\operatorname{diag}\{I_{k_1}, \mathcal{A}_{22}^{-1}\mathcal{B}_{22}, \mathcal{C}_{23}^{-1}, \mathcal{C}^{-1}\mathcal{V}^T, I_{m-r_{ac}}\}, \\ U &= U_b\operatorname{diag}\{I_{l-r_b}, I_{k_2}, -\mathcal{U}\}, \quad V = V_c\operatorname{diag}\{I_{p-r_c}, I_{k_3}, \mathcal{V}\}; \end{aligned} \tag{3.17}$$

then it is easy to verify that the RSVD (2.1) of (A, B, C) is given by (XAY, XBU, VCY).

3.4. A numerical algorithm for the RSVD. We have now the following overall numerical method[1] for the computation of the RSVD of a matrix triplet A, B, C.

ALGORITHM 4.

Input: *Matrices* $A \in \mathbf{R}^{n\times m}, B \in \mathbf{R}^{n\times l}, C \in \mathbf{R}^{p\times m}$.

Output: *RSVD* (2.1) *of the matrix triplet* (A, B, C).

Step 1: *Compute the condensed form* (3.11).

Step 2: *Perform the QR-factorization of* $\begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix}$ *with column pivoting to get an orthonormal basis of the following null space:*

$$I = \mathcal{P}_1^T\mathcal{P}_1 + \mathcal{P}_2^T\mathcal{P}_2, \qquad \mathcal{P}_1 \in \mathbf{R}^{\nu\times\nu}, \mathcal{P}_2 \in \mathbf{R}^{\mu\times\nu},$$
$$0 = \begin{bmatrix} \mathcal{P}_1^T & \mathcal{P}_2^T \end{bmatrix} \begin{bmatrix} \mathcal{A} \\ \mathcal{C} \end{bmatrix}.$$

Step 3: *Perform the QR-factorization of* $\begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix}$ *to get an orthonormal basis of the following range space:*

$$I = \mathcal{Q}_1^T\mathcal{Q}_1 + \mathcal{Q}_2^T\mathcal{Q}_2, \qquad \mathcal{Q}_1 \in \mathbf{R}^{\nu\times\nu}, \mathcal{Q}_2 \in \mathbf{R}^{\mu\times\nu},$$
$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} \mathcal{R} = \begin{bmatrix} \mathcal{B}^T\mathcal{P}_1 \\ \mathcal{P}_2 \end{bmatrix},$$

where $\mathcal{R} \in \mathbf{R}^{\mu\times\mu}$ *is a nonsingular triangular matrix.*

Step 4: *Compute the CSD of* $\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix}$*:*

$$\begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{U} & \\ & \mathcal{V} \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \mathcal{W}.$$

Step 5: *Compute matrices* X, Y, U, *and* V *using* (3.17), *in which* $\mathcal{A}_{ii}^{-1}$, $i = 1, 2, 3$, $\mathcal{B}_{22}^{-1}$, $\mathcal{B}^{-1}$, $\mathcal{C}_{23}^{-1}$, *and* $\mathcal{C}^{-1}$ *are computed using their SVDs or QR-factorizations, such that* (XAY, XBU, VCY) *are in the form* (2.1).

Output XAY, XBU, VCY, X, Y, U, *and* V.

In Algorithm 4, Steps 1, 2, and 3 amount to the QR-factorizations with column pivoting and URV decomposition, which can be implemented in a stable and efficient way (see [1]); Step 5 can be carried out by SVD or QR-factorization; Step 4 is the critical stage in Algorithm 4; one can make use of Van Loan's CSD algorithm here [18].

4. Numerical examples. We will verify only the numerical performance of the core part of Algorithm 4 formed by Steps 2, 3, and 4, which computes the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$. Steps 2 and 3 are implemented using QR-factorization with column pivoting and URV decomposition. Step 4 is implemented as Van Loan's CSD algorithm [18], in which we set the parameter $\tau = \frac{1}{\sqrt{2}}$, which minimizes a backward error bound [3]. All calculations were carried out in MATLAB 5.0 on an HP 712/80 workstation with IEEE standard (machine accuracy $\epsilon \cong 10^{-16}$).

We assume that $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbf{R}^{\mu\times\mu}$ are square. To quantify the accuracy of the results, we define the residuals

$$resSVD = \frac{\|\hat{\Sigma} - \Sigma\|_2}{\mu\|\hat{\Sigma}\|_2},$$

[1]MATLAB code is available upon request.

in which

$$\hat{\Sigma} = \mathrm{diag}\{\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_\mu\}, \quad \Sigma = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_\mu\},$$

where $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_\mu$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\mu$ are the exact and the computed singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, respectively.

In a first setup, consisting of eight examples, the triplet $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is of the form

$$\mathcal{A} = R_1\Sigma_1 R_2, \quad \mathcal{B} = R_1\Sigma_2 W_1, \quad \mathcal{C} = W_2\Sigma_3 R_2,$$

in which R_1, R_2, W_1, and W_2 are randomly chosen orthogonal matrices (obtained from a QR-factorization of a matrix of which the entries are drawn from a uniform distribution over $[0, 1)$); the way in which the entries of

$$\Sigma_i = \mathrm{diag}\{\sigma_{i1}, \ldots, \sigma_{i\mu}\}, \quad i = 1, 2, 3,$$

are chosen is specific for each example. Obviously, the exact SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is given by

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1} = W_1^T \Sigma W_2^T \text{ with } \Sigma = \Sigma_2^{-1}\Sigma_1\Sigma_3^{-1}.$$

We will choose the diagonal elements of Σ_1, Σ_2, and Σ_3 in such a way that different situations in terms of the condition numbers of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are to be dealt with. In the description below, we use the symbol η to denote a value drawn from a uniform distribution over $[0, 1)$; $\kappa(M)$ is the 2-norm condition number of matrix M.

EXAMPLE 1.

$$\sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 2, 3, \quad j = 1, \ldots, \mu.$$

This is a situation in which $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are well-conditioned.

EXAMPLE 2.

$$\sigma_{1j} = (1 + \eta_{1j}) * 10^{-\frac{9}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 2, 3, \quad j = 1, \ldots, \mu.$$

This means that $\mathcal{B}$ and $\mathcal{C}$ are well-conditioned, but, on the other hand, $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^9).$$

EXAMPLE 3.

$$\sigma_{2j} = (2 + \eta_{2j}) * 10^{\frac{3}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 3, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ and $\mathcal{C}$ are well-conditioned; $\mathcal{B}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ have a moderate condition number.

EXAMPLE 4.

$$\sigma_{3j} = (3 + \eta_{3j}) * 10^{\frac{4}{\mu}j}, \quad \sigma_{ij} = (j + i) + \eta_{ij}, \quad i = 1, 2, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ and $\mathcal{B}$ are well-conditioned; $\mathcal{C}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ have a moderate condition number.

EXAMPLE 5.

$$\sigma_{3j} = (j+3)+\eta_{3j}, \quad \sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{C}$ is well-conditioned; $\mathcal{B}$ has a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6).$$

EXAMPLE 6.

$$\sigma_{2j} = (j+2)+\eta_{2j}, \quad \sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{B}$ is well-conditioned; $\mathcal{C}$ has a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6).$$

EXAMPLE 7.

$$\sigma_{1j} = (j+1)+\eta_{1j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{\frac{3}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{A}$ is well-conditioned; $\mathcal{B}$ and $\mathcal{C}$ have a moderate condition number; $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is ill-conditioned, with $\kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^6)$.

EXAMPLE 8.

$$\sigma_{1j} = (1+\eta_{1j})*10^{-\frac{9}{\mu}j}, \quad \sigma_{2j} = (2+\eta_{2j})*10^{\frac{4}{\mu}j}, \quad \sigma_{3j} = (3+\eta_{3j})*10^{-\frac{3}{\mu}j}, \quad j = 1, \ldots, \mu.$$

$\mathcal{B}$ and $\mathcal{C}$ have a moderate condition number; $\mathcal{A}$ and $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are ill-conditioned, with

$$\kappa(\mathcal{A}) = O(10^9), \quad \kappa(\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}) = O(10^{10}).$$

In each of these examples above, the dimension μ was varied between 51 and 100. The values of *resSVD* that were obtained in Examples 1–4 are plotted in Figure 4.1 and Examples 5–8 are plotted in Figure 4.2. From these figures, we see that our results are satisfactory.

In a second setup, we choose

$$\mathcal{B} = (\text{wilkinson}(\mu) - \mu * I_\mu) * V(\mu), \quad \mathcal{A} = (\text{wilkinson}(\mu) - \mu * I_\mu) * \text{hilb}(\mu),$$
$$\mathcal{C} = U(\mu) * \text{hilb}(\mu),$$

in which $U(\mu), V(\mu) \in \mathbf{R}^{\mu\times\mu}$ are randomly chosen orthogonal matrices and wilkinson(μ) and hilb(μ) are the μth-order Wilkinson matrix and Hilbert matrix, respectively (which can be generated by the "wilkinson" and "hilb" command in MATLAB). The exact singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ are obviously equal to 1. They are estimated in 4 different ways:

- By using the algorithm developed in this paper.
- By means of the MATLAB command

  $$\text{svd}(\text{inv}(\mathcal{B}) * \mathcal{A} * \text{inv}(\mathcal{C})),$$

  in which $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ is formed explicitly. This technique is referred to as "direct method 1."
- By means of the MATLAB commands

  $$[U1, S1, V1] = \text{svd}(\mathcal{B}), \quad \tilde{\mathcal{B}} = V1 * \text{inv}(S1) * U1',$$

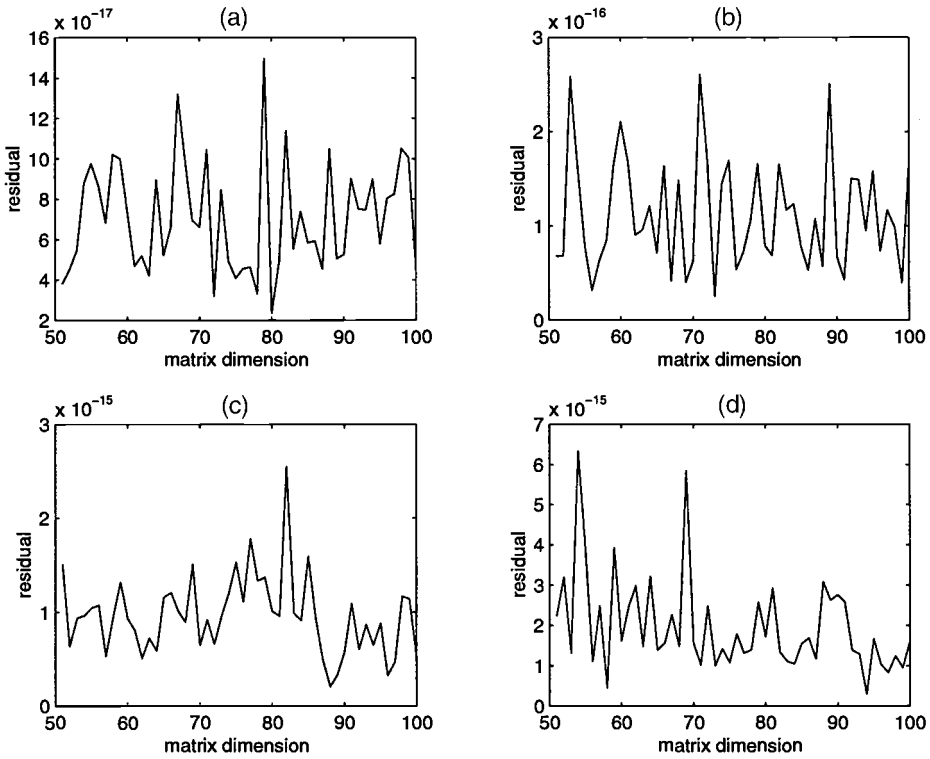


FIG. 4.1. (a)–(d) *depict Examples* 1–4, *respectively.*

$$[U2, S2, V2] = \text{svd}(\mathcal{C}), \quad \tilde{\mathcal{C}} = V2 * \text{inv}(S2) * U2',$$

and

$$\text{svd}(\tilde{\mathcal{B}} * \mathcal{A} * \tilde{\mathcal{C}}).$$

This technique is referred to as "direct method 2."

- By means of the MATLAB command

$$\text{gsvd}(\tilde{\mathcal{A}}, \mathcal{C}),$$

in which

$$\tilde{\mathcal{A}} = \text{hilb}(\mu).$$

It is easy to see that with this particular definition of $\tilde{\mathcal{A}}$, the singular values of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ and $\tilde{\mathcal{A}}\mathcal{C}^{-1}$ are the same. The "gsvd" routine in MATLAB is an implementation of the QSVD algorithm by Van Loan [18]. This technique is referred to as the "GSVD method."

In the following table, we list the values of the residue $resSVD$ obtained by these 4 methods, for $\mu = 8, 9, \ldots, 12$.

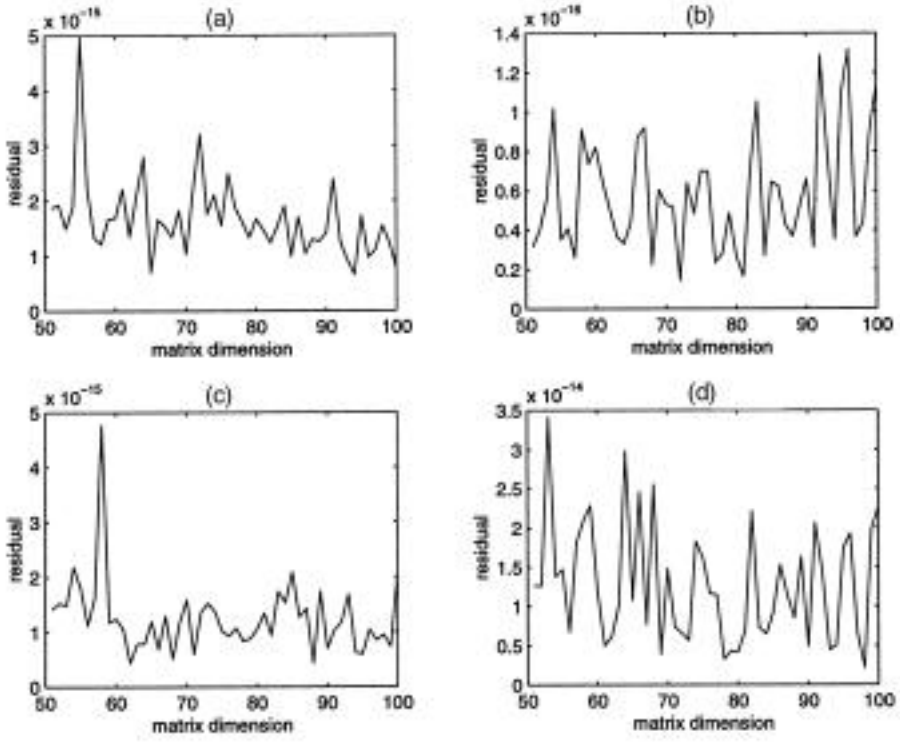


FIG. 4.2. (a)–(d) *depict Examples* 5–8, *respectively.*

	Our method	GSVD method	Direct method 1	Direct method 2
$\mu = 8$	$2.446643637e-08$	$1.106089772e-08$	$1.591006943e-07$	$2.241213379e-07$
$\mu = 9$	$4.635656435e-07$	$3.593259342e-07$	$1.009544058e-05$	$4.035071152e-06$
$\mu = 10$	$1.328935076e-05$	$2.534763366e-06$	$2.875451264e-04$	$1.094955938e-04$
$\mu = 11$	$2.659354824e-04$	$3.018865472e-04$	$2.008595385e-02$	$2.639692842e-03$
$\mu = 12$	$1.686164914e-02$	$1.683184121e-02$	$2.138542385e+00$ Matrix is close to singular! when $\mu = 12$	$1.189748962e-01$

These results clearly show that our method is comparable with the GSVD method and performs better than the direct method 1 and the direct method 2. One of the main reasons for the differences in performance is that Hilbert matrix is very ill-conditioned. For example, for a Hilbert matrix of order μ we have

μ	8	9	10	11	12
Condition number	$1.5258e+10$	$4.9315e+11$	$1.6025e+13$	$5.2196e+14$	$1.712120390592042e+16$

5. Conclusions. We explained how the RSVD of an arbitrary matrix triplet $A \in \mathbf{R}^{n\times m}$, $B \in \mathbf{R}^{n\times l}$, $C \in \mathbf{R}^{p\times m}$ can be computed using a CSD-based QR-type method.

First, the matrices A, B, C are reduced to a lower-dimensional triplet $\mathcal{A}, \mathcal{B}, \mathcal{C}$, with $\mathcal{B}$ and $\mathcal{C}$ nonsingular, using orthogonal transformations such as the QR-factorization with column pivoting and the URV decomposition. Next, the restricted singular values and vectors, defined by the SVD of $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$, are computed using Van Loan's CSD algorithm, without having to form $\mathcal{B}^{-1}\mathcal{A}\mathcal{C}^{-1}$ explicitly. Some numerical examples were given to illustrate the performance of the presented method.

Our algorithm is of the QR-type. It is well known that, in general, QR-type algorithms do not have a high relative accuracy. On the other hand, the Jacobi-type QSVD algorithm proposed by Drmač [8, 9, 10] does have a high relative accuracy. How this procedure can be generalized for the RSVD is an important topic of further research.

Acknowledgments. We would like to thank Prof. Lars Eldén and the anonymous referees for their kind and detailed comments on an early version of the text. This paper benefited a lot from their valuable suggestions.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bishof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, SIAM, Philadelphia, 1992.

[2] Z.J. Bai and J.W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.

[3] Z. Bai, *The CSD, GSVD, their Applications and Computations*, IMA Preprint 958, University of Minnesota, Minneapolis, 1992.

[4] B. De Moor and G.H. Golub, *Generalized Singular Value Decompositions: A Proposal for a Standardized Nomenclature*, Tech. Report 89-10, SISTA, E.E. Dept. (ESAT), K.U. Leuven, Leuven, Belgium, 1989.

[5] B.L.R. De Moor and G.H. Golub, *The restricted singular value decomposition: Properties and applications*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 401–425.

[6] B. De Moor and P. Van Dooren, *Generalizations of the singular value and QR decomposition*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 993–1014.

[7] B. De Moor, *On the structure of generalized singular value and QR decompositions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 347–358.

[8] Z. Drmač, *Fast and accurate algorithms for PSVD and GSVD*, in SIAM Annual Meeting, Stanford University, Stanford, CA, 1997.

[9] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.

[10] Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal., 35 (1998), pp. 1804–1832.

[11] K.V. Fernando and S. Hammarling, *A product induced singular value decomposition for two matrices and balanced realization*, in Linear Algebra in Signals, Systems, and Control, B.N. Datta et al., eds., SIAM, Philadelphia, 1988, pp. 128–140.

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[13] M.T. Heath, A.J. Laub, C.C. Paige, and R.C. Ward, *Computing the singular value decomposition of a product of two matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1147–1159.

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

[15] C.C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

[16] G.W. Stewart, *Computing the CS-decomposition of a partitioned orthogonal matrix*, Numer. Math., 40 (1982), pp. 297–306.

[17] C.F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

[18] C.F. Van Loan, *Computing the CS and the generalized singular value decomposition*, Numer. Math., 46 (1985), pp. 479–491.

[19] H. Zha, *The restricted singular value decomposition of matrix triplets*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 172–194.
[20] H. Zha, *A numerical algorithm for computing the restricted singular value decomposition of matrix triplets*, Linear Algebra Appl., 168 (1992), pp. 1–25.
[21] H. Zha, *Computing the generalized singular values/vectors of large sparse or structured matrix pairs*, Numer. Math., 72 (1996), pp. 391–417.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 602–616

THICK-RESTART LANCZOS METHOD FOR LARGE SYMMETRIC EIGENVALUE PROBLEMS*

KESHENG WU† AND HORST SIMON†

Abstract. In this paper, we propose a restarted variant of the Lanczos method for symmetric eigenvalue problems named the thick-restart Lanczos method. This new variant is able to retain an arbitrary number of Ritz vectors from the previous iterations with a minimal restarting cost. Since it restarts with Ritz vectors, it is simpler than similar methods, such as the implicitly restarted Lanczos method. We carefully examine the effects of the floating-point round-off errors on stability of the new algorithm and present an implementation of the partial reorthogonalization scheme that guarantees accurate Ritz values with a minimal amount of reorthogonalization. We also show a number of heuristics on deciding which Ritz pairs to save during restart in order to maximize the overall performance of the thick-restart Lanczos method.

Key words. thick-restart, Lanczos eigenvalue method, partial reorthogonalization

AMS subject classifications. 65F15, 65F25

PII. S0895479898334605

1. Introduction.

Given an $n \times n$ matrix A, its eigenvalue λ and the corresponding eigenvector x are defined by $Ax = \lambda x$. If the matrix size is large and only a smaller number of eigenvalues are wanted, a projection-based method is usually used [16, 18]. These types of methods usually build orthogonal bases first and then perform the Rayleigh–Ritz projection to extract approximate solutions. There are some alternative projection methods, such as the harmonic Ritz value method [14], but the most significant difference among the projection eigenvalue methods is how they generate their bases. For this reason, most of the eigenvalue methods are named after their basis generation procedures.

When the matrix is symmetric, the Lanczos method (see Algorithm 1, [11, 16, 20]) is the most commonly used method. Other frequently used methods include the Arnoldi method (see Algorithm 2, [1, 20, 24]) and the Davidson method [6, 7, 23]. The Arnoldi method and the Lanczos method are mathematically equivalent on symmetric eigenvalue problems. The Lanczos method is used more frequently because it takes advantage of the fact that most coefficients $h_{j,i}$ computed in step (c) of Algorithm 2 are zero ($h_{j,i} = 0, j = 1, \ldots, i-2$), and the matrix formed from $h_{j,i}$ is symmetric ($\beta_{i-1} \equiv h_{i-1,i} = h_{i,i-1}$, $\alpha_i \equiv h_{i,i}$). This allows the Lanczos method to avoid a significant amount of arithmetic operations. The Davidson method offers more functionality, such as preconditioning, flexible restarting options, etc., but it also uses more arithmetic operations per iteration and more computer memory.

There are many different variations of the Lanczos method depending on factors such as restarting, reorthogonalization, storage schemes for the Lanczos vectors q_i, and

*Received by the editors February 26, 1998; accepted for publication (in revised form) by A. Greenbaum April 30, 2000; published electronically September 15, 2000. This work was supported by the Director, Office of Science, Office of Laboratory Policy and Infrastructure Management, of the U.S. Department of Energy under contract DE-AC03-76SF00098. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.
http://www.siam.org/journals/simax/22-2/33460.html
†Lawrence Berkeley National Laboratory/NERSC, Berkeley, CA 94720 (kwu@lbl.gov, hdsimon@lbl.gov).

ALGORITHM 1. The Lanczos iterations starting with r_0. Let $\beta_0 = \|r_0\|$, and $q_0 = 0$.

For $i = 1, 2, \ldots,$
(a) $q_i = r_{i-1}/\|r_{i-1}\|$,
(b) $p = Aq_i$,
(c) $\alpha_i = q_i^T p$,
(d) $r_i = p - \alpha_i q_i - \beta_{i-1} q_{i-1}$,
(e) $\beta_i = \|r_i\|$.

ALGORITHM 2. The Arnoldi iterations starting with r_0.

For $i = 1, 2, \ldots,$
(a) $q_i = r_{i-1}/\|r_{i-1}\|$,
(b) $p = Aq_i$,
(c) $h_{j,i} = q_j^T p$, $j = 1, \ldots, i$,
(d) $r_i = p - \sum_{j=1}^{i} h_{j,i} q_j$,
(e) $h_{i,i+1} = \|r_i\|$.

so on. This paper will discuss a restarted Lanczos method, and in the remainder of this section we will discuss the motivations for restarting and briefly review the commonly used restarting schemes. We plan to store the Lanczos vectors in a computer's main memory. This significantly reduces the scope of the discussion. The issues related to reorthogonalization and details of how to restart are discussed later in the paper.

In both Algorithms 1 and 2 a new vector q_i is generated in each iteration. These vectors are needed when performing reorthogonalization and computing the Ritz vectors. Often a large number of iterations are needed to compute an eigenvalue. On most machines, there is not enough computer memory to store the Lanczos vectors. In addition, the number of arithmetic operations associated with the reorthogonalization and the Rayleigh–Ritz projection grows as the number of vectors increases. A restarted method avoids these difficulties by limiting the maximum number of vectors it generates at any time. When the maximum number is reached, a set of new starting vectors is computed and the method is restarted. Typically the restarted Lanczos method stores the Lanczos vectors in core, which allows fast access. Because the maximum number of vectors is usually modest, the arithmetic operations required by reorthogonalizations and the Rayleigh–Ritz projection are reasonably small. These features allow a restarted method to execute efficiently.

There are a number of ways to restart the Lanczos method. Since Algorithm 1 can start with only one vector, the most straightforward way is to use the Ritz vector if one eigenvalue is wanted. If more than one eigenvalue is wanted, we can lock the converged ones and combine the rest into one starting vector [18]. Typically, a restarted Lanczos method with one of these restarting schemes needs significantly more iterations to compute a solution than the nonrestarted version. Recently, there has been a number of significant developments in restarted methods. The implicit restarting scheme is a successful strategy that has been applied to both the Arnoldi method [24] and the Lanczos method [2]. Another successful technique is the dynamic thick-restart scheme [7, 10, 15, 25]. Because the thick-restart scheme uses Ritz pairs directly, it is also known as an explicit restarting scheme. The most commonly used thick-restart method is the thick-restart Davidson method. When used with identity preconditioner, this method is mathematically equivalent to the implicitly restarted Arnoldi method with exact shifts [25]. Since the implicit restart scheme does not restart with Ritz vectors, a separate postprocessing step is required to compute the final Ritz vectors when the users need eigenvectors. This postprocessing step is not needed with an explicit restarting scheme that keeps the latest Ritz vectors in the current Krylov subspace. The thick-restart procedure is only slightly different from the Rayleigh–Ritz projection, whereas the implicit restarting procedure is more complex. The implicitly restarted Arnoldi method is known to have stability concerns [13]; ex-

plicitly restarted methods do not have the same concerns. For these reasons, we set out to develop and study a thick-restart Lanczos method for eigenvalue problems in this paper.

2. Thick-restart Lanczos algorithm. We have briefly reviewed the features of the thick-restart procedure and the Lanczos method. In this section we will show how the two may be combined to form an eigenvalue method.

Before any restarting takes place, the restarted Lanczos method proceeds as described in Algorithm 1. Assume that m iterations are allowed before restarting. After m iterations, the vectors q_i satisfy the Lanczos recurrence

$$AQ_m = Q_m T_m + \beta_m q_{m+1} e_m^T, \tag{2.1}$$

where $Q_m = [q_1, \ldots, q_m]$, e_m is the last column of the identity matrix with m columns, and $T_m \equiv Q_m^T A Q_m$ is an $m \times m$ symmetric tridiagonal matrix constructed from α_i and β_i, $t_{i,i} = \alpha_i, t_{i,i+1} = t_{i+1,i} = \beta_i$. Using the Rayleigh–Ritz projection, we can produce approximate solutions to the eigenvalue problem. Let (λ, y) be a pair of eigenvalue and eigenvector, i.e., an eigenpair, of T_m; then λ is an approximate eigenvalue of A and $x = Q_m y$ is the corresponding approximate eigenvector. They are also known as the Ritz value and the Ritz vector. The residual of this approximate solution is defined to be $Ax - \lambda x$. For symmetric eigenvalue problems, the norm of this residual is a good indicator of the solution quality.

When restarting, we first determine an appropriate number of Ritz vectors to save, say, k, then choose k eigenvectors of T_m, say, Y, and compute k Ritz vectors, $\hat{Q}_k = Q_m Y$. The following derivation can be carried out by assuming Y to be any orthonormal basis of a k-dimensional invariant subspace of T_m. Since the matrix T_m is symmetric, there is no apparent advantage to use any basis set other than the eigenvectors. If Y are eigenvectors of T_m, the vectors saved during restart $\hat{Q}_k$ are Ritz vectors. To distinguish the quantities before and after restart, we denote the quantities after restart with a hat. For example, the projected matrix T_m after restart is $\hat{T}_k \equiv Y^T T_m Y$. Since we have chosen to restart with Ritz vectors, the matrix $\hat{T}_k$ is diagonal and the diagonal elements are the Ritz values. Immediately after restart, the new basis vectors satisfy the relation

$$A\hat{Q}_k = \hat{Q}_k \hat{T}_k + \beta_m \hat{q}_{k+1} s^T, \tag{2.2}$$

where $\hat{q}_{k+1} \equiv q_{m+1}$ and $s \equiv Y^T e_m$. We recognize that this equation is an extension of (2.1) because the residual vector of every Ritz vector in $\hat{Q}_k$ is parallel to $\hat{q}_{k+1}$. In Algorithm 1, the Lanczos recurrence is extended one column at a time by augmenting the current basis with q_{i+1}. In the same spirit, we can augment the basis $\hat{Q}_k$ with $\hat{q}_{k+1}$.

To continue extending the basis, we follow the augment Krylov subspace method [3, 19] and use the Gram–Schmidt procedure to enforce the orthogonality of the whole basis. The expression for $\hat{q}_{k+2}$ is

$$\begin{aligned} \hat{\beta}_{k+1}\hat{q}_{k+2} = \hat{r}_{k+1} &\equiv (I - \hat{Q}_{k+1}\hat{Q}_{k+1}^T) A\hat{q}_{k+1} \\ &= (I - \hat{q}_{k+1}\hat{q}_{k+1}^T - \hat{Q}_k \hat{Q}_k^T) A\hat{q}_{k+1} \\ &= (I - \hat{q}_{k+1}\hat{q}_{k+1}^T) A\hat{q}_{k+1} - \hat{Q}_k \beta_m s. \end{aligned} \tag{2.3}$$

The scalar $\hat{\beta}_{k+1}$ in the above equation is equal to the norm of the right-hand side so that $\hat{q}_{k+2}$ has unit norm. Since the vector $\hat{Q}_k^T A \hat{q}_{k+1}$ is known ($= s$), we only need to

compute $\hat{\alpha}_{k+1}$ as in step (c) of Algorithm 1. The vector $\hat{q}_{k+2}$ can be computed by replacing step (d) with $\hat{r}_{k+1} = \hat{p} - \hat{\alpha}_{k+1}\hat{q}_{k+1} - \sum_{j=1}^{k} \beta_m s_j \hat{q}_j$, where $\hat{p} = A\hat{q}_{k+1}$. While computing $\hat{q}_{k+2}$, we also extended the matrix $\hat{T}_k$ by one column and one row, which produces an arrowhead matrix $\hat{T}_{k+1}$. The Lanczos recurrence relation (see (2.1)) is maintained after this step, more specifically, $A\hat{Q}_{k+1} = \hat{Q}_{k+1}\hat{T}_{k+1} + \hat{\beta}_{k+1}\hat{q}_{k+2}e_{k+1}^T$, where $\hat{\beta}_{k+1} = \|\hat{r}_{k+1}\|$. Even though $\hat{T}_{k+1}$ is not tridiagonal as in the original Lanczos method, further steps of the restarted Lanczos algorithm can be carried out using three-term recurrence, as shown next.

After we have computed $\hat{q}_{k+i}$ $(i > 1)$, to compute the next vector $\hat{q}_{k+i+1}$, we again go back to the Gram–Schmidt procedure,

$$\begin{aligned}
\hat{\beta}_{k+i}\hat{q}_{k+i+1} &= (I - \hat{Q}_{k+i}\hat{Q}_{k+i}^T)A\hat{q}_{k+i} \\
&= (I - \hat{q}_{k+i}\hat{q}_{k+i}^T - \hat{q}_{k+i-1}\hat{q}_{k+i-1}^T)A\hat{q}_{k+i} - \hat{Q}_{k+i-2}(A\hat{Q}_{k+i-2})^T\hat{q}_{k+i} \\
&= A\hat{q}_{k+i} - \hat{\alpha}_{k+i}\hat{q}_{k+i} - \hat{\beta}_{k+i-1}\hat{q}_{k+i-1}, \qquad (2.4)
\end{aligned}$$

where by definition $\hat{\alpha}_{k+i}$ is $\hat{q}_{k+i}^T A\hat{q}_{k+i}$ and $\hat{\beta}_{k+i}$ is the norm of the right-hand side. The above equation is true for any i greater than 1. From this equation we see that computing $\hat{q}_{k+i}$ $(i > 2)$ requires the same amount of arithmetic work as in the original Lanczos algorithm; see Algorithm 1. The matrix $\hat{T}_{k+i} \equiv \hat{Q}_{k+i}^T A\hat{Q}_{k+i}$ can be written as follows:

$$\hat{T}_{k+i} = \begin{pmatrix} \hat{T}_k & \beta_m s & & & \\ \beta_m s^T & \hat{\alpha}_{k+1} & \hat{\beta}_{k+1} & & \\ & \hat{\beta}_{k+1} & \hat{\alpha}_{k+2} & \hat{\beta}_{k+2} & \\ & & \ddots & \ddots & \ddots \end{pmatrix}.$$

The above formulas show how to continue the Lanczos iterations after the first restart. The derivation is based on the facts that the Lanczos vectors are orthogonal and that they satisfy the Lanczos recurrence. Since the vectors resulting from the above formulas also satisfy the same conditions, the procedure can be repeatedly restarted. It is clear that this restarted algorithm is cheaper than the straightforward versions of the augmented Krylov methods. If k vectors $(\hat{Q}_k)$ are saved, the augmented Krylov subspace method needs the projection matrix $\hat{Q}_k^T A\hat{Q}_k$ in order to proceed. The crucial step here is determining how to generate $A\hat{Q}_k$. Usually one either explicitly multiplies A and $\hat{Q}_k$ or computes $A\hat{Q}_k$ from stored AQ_m of previous iterations. However, because of (2.2), $\hat{Q}_k^T A\hat{Q}_k$ is available without performing any matrix-vector multiplication with A or storing AQ_m in computer memory. Similar to the nonrestarted Lanczos method, using the Lanczos recurrence relation we can compute the residual norms of the approximate eigenpairs without explicitly computing the residual vectors. This allows us to measure the quality of the solutions efficiently.

The matrix T_m is no longer tridiagonal after the first restart but can still be stored in an efficient manner. As mentioned before, the matrix $\hat{T}_k$ is diagonal, and its nonzero values can be stored as $\hat{\alpha}_1, \ldots, \hat{\alpha}_k$. The array $(\beta_m s)$ is of size k and can be stored as $\hat{\beta}_1, \ldots, \hat{\beta}_k$. In short, the arrays $\hat{\alpha}_i$ and $\hat{\beta}_i$ are

$$\hat{\alpha}_i = \lambda_i, \qquad \hat{\beta}_i = \beta_m y_{m,i}, \qquad i = 1, \ldots, k, \qquad (2.5)$$

where λ_i is the ith saved eigenvalue of T_m, the corresponding eigenvector is the ith column of Y, and $y_{m,i}$ is the mth element of the ith column. After restart the first k basis vectors satisfy the following relation:

$$A\hat{q}_i = \hat{\alpha}_i \hat{q}_i + \hat{\beta}_i \hat{q}_{k+1}, \quad i = 1, \ldots, k.$$

It is easy to arrange the algorithm so that $\hat{q}_i$ and q_i are stored in the same memory location. The hat symbol is dropped in the following description of the algorithm.

ALGORITHM 3.

The thick-restart Lanczos iterations starting with k Ritz vectors and a residual vector r_k such that $Aq_i = \alpha_i q_i + \beta_i q_{k+1}$, $i = 1, \ldots, k$, and $q_{k+1} = r_k/\|r_k\|$. The value k may be zero, in which case α_i and β_i are uninitialized and r_0 is the initial guess.

1. Initialization.

(a) $q_{k+1} = r_k/\|r_k\|$,
(b) $p = Aq_{k+1}$,
(c) $\alpha_{k+1} = q_{k+1}^T p$,
(d) $r_{k+1} = p - \alpha_{k+1} q_{k+1} - \sum_{i=1}^{k} \beta_i q_i$,
(e) $\beta_{k+1} = \|r_{k+1}\|$,

2. Iterate. For $i = k+2, k+3, \ldots$,

(a) $q_i = r_{i-1}/\beta_{i-1}$,
(b) $p = Aq_i$,
(c) $\alpha_i = q_i^T p$,
(d) $r_i = p - \alpha_i q_i - \beta_{i-1} q_{i-1}$,
(e) $\beta_i = \|r_i\|$.

The difference between Algorithms 1 and 3 is in the initialization step. When k is zero, the initialization step of the two algorithms are the same. Algorithm 3 can take on an arbitrary number of starting vectors, but Algorithm 1 cannot. When k is greater than zero, the initialization step orthogonalizes Aq_{k+1} against all existing $k+1$ vectors. In all subsequent steps, the same three-term recurrence is used for both Algorithms 1 and 3.

It is easy to see how an existing restarted Lanczos program can be converted to generate orthogonal bases using the above algorithm. To convert a complete eigenvalue program, the Rayleigh–Ritz projection step needs to be modified because the matrix T_m is not tridiagonal in the new method. Our options include treating it as a full matrix, treating it as a banded matrix, and using Givens rotations to reduce it to a tridiagonal matrix. After deciding what to do, we can use an appropriate routine from LAPACK or EISPACK to find all eigenvalues and eigenvectors of T_m. At this point, as in any other version of the Lanczos method, we can evaluate the residual norms of the approximate solutions and perform a convergence test [9, 18]. In addition, based on Ritz values and their residual norms, we can also decide which Ritz pairs to save. This allows us to compute only those Ritz vectors that are needed for restarting.

Following the same argument used to show that the implicitly restarted Arnoldi method is equivalent to the thick-restart Davidson method, it is easy to show that the thick-restart Lanczos method is mathematically equivalent to the implicitly restarted Lanczos method [27]. We derived the thick-restart Lanczos method by using the augmented Krylov subspace concept which may lead to bases that do not span Krylov subspaces. This equivalence property indicates that the bases built by this method in fact span Krylov subspaces, though the starting vectors for the Krylov sequences are not explicitly known. A corollary of this equivalence property is that the new method is a Krylov subspace method. Because of the equivalence relation, we expect the new method to be as effective as the implicitly restarted Lanczos method. One advantage of the new method is that it is simpler to implement as a computer program.

This concludes the description of the new algorithm. In the remainder of this paper, we will focus on two issues related to implementing the eigenvalue method on computers: how to maintain orthogonality among the Lanczos vectors, and which Ritz pairs to save when restarting.

TABLE 3.1
Information about the test matrices used.

Name	N	NNZ	Description
NASASRB	54870	2677324	shuttle rocket booster structure from NASA
S3DKT3M2	90449	3753461	cylindrical shell, uniform triangular mesh
S3DKQ4M2	90449	4820891	cylindrical shell, quadratic elements

3. Orthogonality of the basis. The above description of the thick-restart Lanczos method is accurate only when carried out in exact arithmetic. When implemented as a computer program, the round-off errors of floating-point arithmetic will cause the Lanczos vectors to lose orthogonality. A similar issue also exists for other variants of the Lanczos method and has been extensively studied before. The typical cure for loss of orthogonality is reorthogonalization through the Gram–Schmidt procedure. The commonly used reorthogonalization schemes are no reorthogonalization [4, 26], the selective reorthogonalization [17], the partial reorthogonalization [22], and the full reorthogonalization. In this section we will exam three of the four schemes, i.e., no reorthogonalization, full reorthogonalization, and partial reorthogonalization. We leave out the selective reorthogonalization because it has similar objectives to the partial reorthogonalization scheme, and the latter one was shown to be more effective [21].

If no reorthogonalization is performed, we avoid the arithmetic operations associated with reorthogonalization, and we need to access only the two most recent Lanczos vectors when building the basis. If eigenvectors aren't needed, there is no need to access the earlier Lanczos vectors. Not performing reorthogonalization may reduce both operation count and memory requirement of a nonrestarted Lanczos program. The same is not true for the thick-restart Lanczos method. The thick-restart Lanczos method cannot be implemented without storing the Lanczos vectors since they are an integral part of the restarting process. Not performing reorthogonalization in the thick-restart Lanczos method reduces only the arithmetic operations. In the nonrestarted Lanczos method, loss of orthogonality among the Lanczos vectors leads to spurious solutions, even though the spurious solutions are duplicate copies of the correct eigenvalues [9, 16]. To illustrate the issues related to loss of orthogonality in the thick-restart Lanczos method, we conduct some tests using three symmetric matrices listed in Table 3.1. These matrices are the largest symmetric matrices from the MatrixMarket web site[1] when the tests were conducted.

Figure 3.1 shows the orthogonality level of the bases built by the thick-restart Lanczos method without reorthogonalization. The horizontal axis indicates how many times the Lanczos method restarted, and the vertical axis is the Frobenius norm of $Q^TQ - I$, where Q contains 20 Lanczos vectors. When vectors in Q are orthogonal to machine precision, we would expect $\|Q^TQ - I\|_F$ to be on order of 10^{-15}. As $\|Q^TQ - I\|_F$ becomes close to one, Q is no longer a set of linearly independent vectors. Data in Figure 3.1 show that the loss of orthogonality progressively becomes worse after the first few restarts. Similar loss of orthogonality has been observed in the nonrestarted Lanczos method. Despite the loss of orthogonality, the nonrestarted Lanczos method can still compute the eigenvalues accurately. To see whether the thick-restarted Lanczos method behaves similarly, we conduct further tests. For con-

[1] MatrixMarket URL is http://math.nist.gov/MatrixMarket.

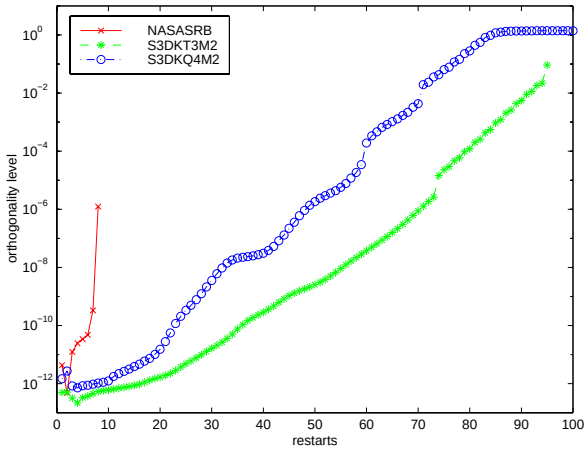


FIG. 3.1. *The orthogonality level* ($\|Q^TQ - I\|_F$) *of the bases built by the thick-restart Lanczos method without reorthogonalization* ($m = 20$).

venience of discussion, we define δ_λ and δ_r to measure the errors caused by the loss of orthogonality:

$$\delta_\lambda \equiv |\lambda - x^T Ax/x^T x|, \qquad \delta_r \equiv \|Ax - \lambda x\| - |\beta_m e_m^T y|.$$

These two quantities together will be called the floating-point errors in this paper. If the Rayleigh–Ritz projection is carried out using exact arithmetic on an exactly orthogonal basis, both δ_λ and δ_r are zero. If the Lanczos vectors are orthogonal to the machine precision (ϵ), then the floating-point errors are on the order of $\epsilon\|A\|$.

Table 3.2 shows the five largest Ritz values and their corresponding δ_λ and δ_r computed by the thick-restart Lanczos method without reorthogonalization ($m = 20$). Since all values of δ_λ are close to $\epsilon\|A\|(= \epsilon\lambda_{max})$, these Ritz values are close to their exact values. Given that the Ritz values are accurate, δ_r indicates errors in the Ritz vectors. If the Ritz vectors are accurate, δ_r is expected to be close to $\epsilon\|A\|$. The values of δ_r in Table 3.2 are several orders of magnitude larger than $\epsilon\|A\|$, which indicates that the eigenvectors are not computed accurately. Similar characteristics are also present in the nonrestarted Lanczos method. What is also similar is that they both generate the same kind of spurious solutions. For example, the largest eigenvalue of S3DKQ4M2 is a simple eigenvalue; however, from Table 3.2, we see it is computed twice.

The most straightforward way to cure the loss of orthogonality problem is to perform full reorthogonalization. Since the full reorthogonalization maintains the orthog-

TABLE 3.2
The five largest eigenvalues computed by the thick-restart Lanczos method without reorthogonalization.

NASASRB			S3DKT3M2			S3DKQ4M2		
λ	δ_λ	δ_r	λ	δ_λ	δ_r	λ	δ_λ	δ_r
2648056755	1E-6	2E2	8798.436369	3E-11	7E-6	4601.653436	6E-11	2E-6
2647979344	1E-6	2E2	8796.715998	1E-11	7E-5	4601.653436	3E-11	7E-7
2634048615	4E-6	7E2	8794.143789	4E-11	1E-3	4600.851648	4E-11	3E-6
2633679289	1E-6	9E2	8793.936155	4E-11	7E-3	4599.515718	3E-12	2E-6
2606151408	3E-6	1E3	8792.317911	9E-12	8E-3	4598.281889	6E-11	2E-5

onality to the machine precision, reorthogonalization is necessary only if $r_{k+1}^T r_{k+1} < \alpha_{k+1}^2 + \sum_{i=1}^{k} \beta_i^2$ after step (1.d) or $r_i^T r_i < \alpha_i^2 + \beta_{i-1}^2$ after step (2.d) [28]. Usually it is necessary only to orthogonalize r_i against Q_i once [16]. If the norm of r_i reduced significantly after the Gram–Schmidt procedure, it indicates that r_i is almost a linear combination of the current basis Q_i. In other words, an invariant subspace has been found. The algorithm can be continued with any unit vector that is orthogonal to Q_i. This is a different form of restarting which happens very infrequently. In this case, β_i should be set to zero. There are many different ways to implement the full reorthogonalization procedure, for example, always performing the Gram–Schmidt procedure once or twice, using a different criteria to determine when to invoke the Gram–Schmidt procedure [5], etc. The scheme we have selected above appears to be inexpensive and works well in tests.

The third reorthogonalization scheme is the partial reorthogonalization scheme which simulates loss of orthogonality using the ω-recurrence and performs reorthogonalization only if the loss of orthogonality exceeds the user-specified limit. It is relatively easy to adopt the ω-recurrence to the thick-restart Lanczos method [28]. Using the ω-recurrence, we can monitor the loss of orthogonality and maintain the orthogonality to any reasonable level desired. Similar to the nonrestarted Lanczos method, it is easy to show that the thick-restart Lanczos method can generate accurate eigenvalues if the actual orthogonality level of the basis is no worse than $\sqrt{\epsilon}$ [28]. The partial reorthogonalization procedure for the thick-restart Lanczos method is very similar to that of the nonrestarted Lanczos method. One important caveat is that the last residual vector before restarting, r_m or equivalently $q_{m+1} \equiv r_m/\|r_m\|$, must be orthogonal to the existing basis vectors to the machine precision. This does not mean that the all earlier vectors need to be orthogonal to machine precision; it merely means that the reorthogonalization process must be invoked in the last iteration before restarting [28]. More details on the partial reorthogonalization can be found in the appendix.

Figure 3.2 plots the orthogonality level of the Lanczos bases built by the thick-restart Lanczos method with the partial reorthogonalization ($m = 20$). The difference between the two curves is that the solid curve is generated with the modification that always reorthogonalizes r_{20} while the dashed curve is generated without this modification. This extra reorthogonalization ensures that r_{20} has no significant error in the directions of the vectors to be discarded during restarting. Since errors in those directions cannot be recovered in the future iterations, avoiding them improves the overall quality of the bases. The two tests shown in Figure 3.2 clearly demonstrate the importance of maintaining orthogonality of the last residual vector of a restarted loop. The figure also demonstrates that when the last residual vector is orthogonal, the orthogonality level of the whole basis can be maintained at a reasonable level.

The next set of tests will demonstrate that the thick-restart Lanczos method with partial reorthogonalization generates accurate solutions [28]. To do this, we compare the floating-point errors generated by the thick-restart Lanczos method with partial reorthogonalization and with full reorthogonalization. To verify the results, we also conduct the same test on an implementation of the implicitly restarted Lanczos method with full reorthogonalization (ARPACK [12]). Table 3.3 shows the results of this set of tests. The five largest eigenvalues of the three matrices are computed. Corresponding to each eigenvalue, there is a pair of δ_λ and δ_r. The errors reported in the table are the maximum errors of the five pairs. In these tests, we have used a relatively small basis size ($m = 10$). The rationale for using a smaller basis size is that the floating-point errors might be larger because more iterations are needed.

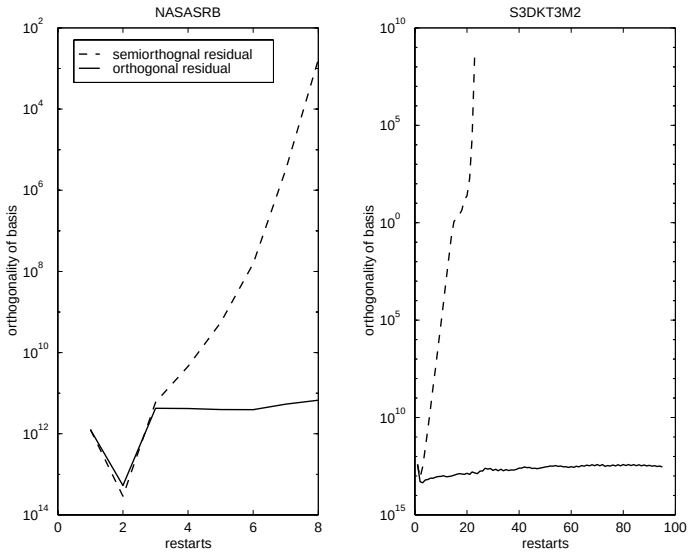


FIG. 3.2. *The orthogonality level* ($\|Q^TQ - I\|_F$) *of the bases prior to each restart.*

Indeed the floating-point errors are slightly larger than when the basis size is 20. The important point to note is that the errors of different methods are roughly the same. Most of the quantities in Table 3.3 are about $10 \sim 100\epsilon\|A\|$, which confirms that the partial reorthogonalization scheme can maintain a very good orthogonality level (see Figure 3.2) and generate accurate solutions.

It is possible for the orthogonality level in the Lanczos method with partial reorthogonalization to rise to the user-specified limit, typically $\sqrt{\epsilon}$, in which case δ_r would be on the order of $\sqrt{\epsilon}\|A\|$, though δ_λ remains on the order of $\epsilon\|A\|$. Table 3.4 shows an example where δ_r is significantly larger than $\epsilon\|A\|$. In this example, the smallest eigenvalues of NASASRB are computed. They are nine orders of magnitude smaller than the largest one, and the relative gap ratio is on the order of 10^{-10}. It takes about 50,000 iterations for the thick-restart Lanczos method to reduce the residual norms of the five smallest Ritz values to 10^{-4}. ARPACK reduces these residual norms to about 10^4 using similar number of iterations and the same basis size. The performance difference is mainly due to the differences in the restarting strategies which will be discussed in next section. The size of $\|r\|$ does not affect the value of

TABLE 3.3

The maximum floating-point errors of the five largest Ritz values computed using basis size (m) 10*, where method* I *is the thick Lanczos method with partial reorthogonalization, method* II *is the thick-restart Lanczos method with full reorthogonalization, and method* III *is ARPACK.*

	NASASRB			S3DKT3M2			S3DKQ4M2		
	I	II	III	I	II	III	I	II	III
max δ_λ	7E-6	2E-5	6E-6	1E-10	4E-11	2E-10	5E-10	4E-10	3E-11
max δ_r	1E-5	2E-6	3E-6	5E-11	2E-10	7E-11	6E-10	3E-10	4E-10
MATVEC	185	185	184	2269	2269	4459	5119	5119	3646
restarts	46	46	39	465	465	1310	1516	1516	1516
time	11.5	12.0	14.6	200	213	577	533	555	504

TABLE 3.4
The maximum floating-point errors of NASASRB's five smallest Ritz values computed using basis size 1000.

	I	II	III
max δ_λ	9E-7	2E-6	5E-7
max δ_r	7E-4	3E-6	4E-6
MATVEC	50471	50467	46761
restarts	51	51	61
time	9798(8PE)	8546(8PE)	8547(16PE)

δ_r. When full orthogonality is maintained, δ_r is on the order of 10^{-6} ($\sim 10\epsilon\|A\|$); see column II and III of Table 3.4. Because the smallest eigenvalues are much harder to compute than the largest ones, many large eigenvalues reach convergence before the smallest ones do. This provides ample opportunities for serious loss of orthogonality to occur. The actual orthogonality level is near $\sqrt{\epsilon}$ in this case.

Tables 3.3 and 3.4 also contain some performance information about the different methods. The CPU time reported is from a Cray T3E-900 at the National Energy Research Supercomputer Center.[2] The time in Table 3.3 is measured on two processors. The numbers of processors used for Table 3.4 are next to the time values. Since the two versions of the thick-restart Lanczos method use the same restarting strategy, the time differences are mainly due to the different reorthogonalization strategies. The results shown in Table 3.3 are representative of the typical case where using partial reorthogonalization saves execution time. With full reorthogonalization, the global reorthogonalization, i.e., the Gram–Schmidt procedure, is often invoked once per matrix-vector multiplication. With partial reorthogonalization, the Gram–Schmidt procedure is invoked only infrequently. The percentage of time saved is relatively small because the Gram–Schmidt procedure can be performed much faster than the parallel matrix-vector multiplications.

Table 3.4 shows an extreme case where using the partial reorthogonalization actually takes more time than using the full reorthogonalization. In this case, the partial reorthogonalization algorithm invokes the global reorthogonalization frequently, about once every three matrix-vector multiplications. Each time the global reorthogonalization is invoked, the Gram–Schmidt procedure is applied on the two most recent Lanczos vectors to make sure both vectors are orthogonal to the proceeding vectors to machine precision. Since there are some Lanczos vectors that are not orthogonal to machine precision against others, the Gram–Schmidt procedure might have to be repeated more times than in the case where all Lanczos vectors are orthogonal to machine precision [28]. All these make the partial reorthogonalization more expensive for solving this test problem.

In short, the loss of orthogonality among the Lanczos vectors has very similar effects on the thick-restart Lanczos method as on other variants of the Lanczos method. Using the partial reorthogonalization, the thick-restart Lanczos method can generate accurate eigenvalues but not always accurate eigenvectors. In most cases, using partial reorthogonalization reduces the CPU time compared to using full reorthogonalization. When implementing a general purpose Lanczos method, we would suggest using full reorthogonalization. The tests performed in the next section use only the thick-restart Lanczos method with full reorthogonalization.

[2]More information about the National Energy Research Supercomputer Center can be found at http://www.nersc.gov.

4. Restarting strategies. The thick-restart Lanczos method offers the flexibility of saving an arbitrary portion of the current basis during restarting. Given this capability, a crucial problem is determining how to take full advantage of it. A number of theoretical tools are available for analyzing restarting strategies used in the implicitly restarted Arnoldi method and the thick-restart Davidson method [2, 8, 15, 24]; however, the most successful strategies for deciding what to save are based on heuristics. For example, ARPACK uses a heuristic strategy based on basis size, number of eigenvalues converged, and so on. This section will briefly summarize our experiences of using three such heuristics.

The first restarting strategy attempts to maximize the reduction of residual norms at every step. This strategy is based on the one used in the dynamic thick-restart Davidson method [25]. It is implemented using a parameter called the effective gap ratio γ. In each step of the restarted Lanczos method, the residual norm is expected to reduce by a factor proportional to $e^{-\sqrt{\gamma}}$ [15]. To maximize the residual norm reduction, we need to maximize γ. If the eigenvalues of the matrix A are $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, the gap ratio for λ_1 is $(\lambda_2 - \lambda_1)/(\lambda_n - \lambda_1)$. When restarting to compute the smallest eigenvalue while saving the eigenvectors corresponding to $\lambda_2, \ldots, \lambda_k$, the effective gap ratio is [15, 25]

$$\gamma = (\lambda_{k+1} - \lambda_1)/(\lambda_n - \lambda_1).$$

When used in the Davidson method or the Lanczos method, the eigenvalues in the above definition are replaced with the computed Ritz values. Because the Ritz values may not be good approximations to their corresponding eigenvalues and because there are far fewer Ritz values than the eigenvalues $m \ll n$, the computed gap ratio γ may be quite different from the actual gap ratio. Typically, when k is large, say, $k \geq 2m/3$, the computed gap ratio γ is significantly larger than the actual gap ratio. Another problem with maximizing γ is that it is a monotonic function of k. The maximum value for k is $m-1$. If this maximum value is used, $m-1$ Ritz pairs are saved during restarting, and the restarting procedure is invoked after every matrix-vector multiplication. Computing $m-1$ Ritz pairs takes a considerable number of arithmetic operations. Since the computed γ is larger than its actual value, the reduction in residual norm is much less than expected. To reduce the number of Ritz pairs saved, the developers of the dynamic thick-restart Davidson method [25] enforce a limit on k, $k \leq m-3$. In our experiences, we found that reducing the size of k often reduces the overall execution time. The following formula is found to be a reasonable choice, $k \leq \max(n_{eig}, (3m+2n_c)/5)$, where n_{eig} is the number of eigenvalues to be computed and n_c is the number of eigenvalues converged so far.

In the dynamic thick-restart Davidson method, instead of maximizing the residual norm reduction of each iteration, we choose to maximize the residual norm reduction of the whole restarted loop, i.e., maximize $\xi \equiv (m-k)\sqrt{\gamma}$. Alternatively, we can also choose to maximize $\mu \equiv (m-k)\gamma$. Both ξ and μ are not monotonic functions of k; we actually need to search through all possible values of k to find their maximums. Typically these strategies yield a smaller k than maximizing γ. However, to avoid potentially choosing exceedingly large k, we also use the same restriction on k as in the previous case.

Table 4.1 shows some examples of how the three restarting strategies work and compares them to a simple restarted Lanczos method (LANSO-locking) and the implicitly restarted Lanczos method implemented in ARPACK. It shows both iteration counts and time used to compute the five largest eigenvalues of the three test matrices.

TABLE 4.1
Time and iterations needed to compute five largest eigenvalues of the test matrices ($m = 20$).

	NASASRB		S3DKT3M2		S3DKQ4M2	
	Iter.	Time	Iter.	Time	Iter.	Time
LANSO-locking	2170	266	2678	464	> 5000	> 1000
ARPACK	145	23.0	973	233	961	257
TRLAN max γ	88	14.0	693	195	784	232
TRLAN max μ	96	14.4	684	167	741	196
TRLAN max ξ	92	12.9	691	165	757	191

The time reported is the number of seconds on a DEC alpha processor running at 450MHz. The simple restarted Lanczos method always restarts with the Ritz vector corresponding to the largest Ritz value that is not converged yet, and it locks the converged Ritz pairs. The program is implemented on top of the LANSO package maintained by Beresford Parlett of UC Berkeley. Table 4.1 shows that the three versions of the thick-restart Lanczos method use less time than the simple restarted Lanczos method and ARPACK on the test problems. The differences among the three versions of the thick-restart Lanczos method are relatively small.

This set of examples clearly demonstrates that the restarting strategy is important to the overall effectiveness of the eigenvalue methods. The strategies suggested here give reasonable performances compared to the existing strategies used in ARPACK. Some of the known techniques not discussed here include saving Ritz pairs from the opposite end of the spectrum and taking into account the residual norms when computing gap ratio. Our tests indicate that there are some advantages to using these techniques in combination with those described earlier in this section [29]. However, the modified strategies do not consistently outperform the simple ones shown in Table 4.1.

5. Summary. In this paper, we described an explicitly restarted Lanczos method for symmetric eigenvalue problems called the thick-restart Lanczos method. It is theoretically equivalent to the implicitly restarted Lanczos method. The main advantage of the new method is that it is simpler to use. We studied three different reorthogonalization schemes and found that the loss of orthogonality has similar effects on this restarted Lanczos method as on the original nonrestarted Lanczos method. In other words, without reorthogonalization, it usually generates accurate eigenvalues; with the partial reorthogonalization, it is guaranteed to generate accurate eigenvalues; only with the full reorthogonalization can it generate both accurate eigenvalues and accurate eigenvectors.

Through some examples, we also demonstrated the importance of employing an effective restarting strategy and suggested a number of restarting heuristics. Tests showed that these strategies are as effective as the best-known strategies.

Appendix. Partial reorthogonalization. This appendix gives more details on how to implement the partial reorthogonalization scheme in the thick-restart Lanczos method. We address three issues in three subsections: ω-recurrence for the thick-restart Lanczos method, global reorthogonalization, and local reorthogonalization.

A.1. Monitoring loss of orthogonality. An important ingredient of the partial reorthogonalization is the ω-recurrence for monitoring loss of orthogonality [22]. This subsection extends the ω-recurrence for the thick-restart Lanczos method. The derivation of the recurrence is relatively straightforward. To start with, we will rewrite

(2.1), (2.3), and (2.4) to accommodate round-off errors during the actual computations:

$$\begin{aligned} Aq_i &= \alpha_i q_i + \beta_i q_{k+1} + d_i && (i \le k),\\ Aq_{k+1} &= \alpha_{k+1} q_{k+1} + \sum_{j=1}^{k} \beta_j q_j + \beta_{k+1} q_{k+2} + d_{k+1},\\ Aq_i &= \alpha_i q_i + \beta_{i-1} q_{i-1} + \beta_i q_{i+1} + d_i && (i > k+1). \end{aligned}$$

In the above equations, d_i represents the error associated with expressing Aq_i in terms of other quantities ($\|d_i\| \le \epsilon \|Aq_i\|$).

The ω-recurrence uses $\omega_{i,j} \equiv q_i^T q_j$ as the measure of loss of orthogonality. For symmetric matrices, we can use the relation $q_j^T Aq_i = q_i^T Aq_j$ and the above three equations to generate a recursion for $\omega_{i,j}$:

$$\begin{aligned} \beta_i \omega_{i+1,j} &= (\alpha_j - \alpha_i)\omega_{i,j} + \beta_j \omega_{i,k+1} - \beta_{i-1}\omega_{i-1,j} + d_j^T q_i - q_j^T d_i && (j \le k),\\ \beta_i \omega_{i+1,k+1} &= (\alpha_i - \alpha_{k+1})\omega_{i,k+1} + \sum_{j=1}^{k} \beta_j \omega_{i,j} + \beta_{k+1}\omega_{i,k+2} - \beta_{i-1}\omega_{i-1,k+1}\\ &\quad + d_{k+1}^T q_i - q_{k+1}^T d_i,\\ \beta_i \omega_{i+1,j} &= (\alpha_j - \alpha_i)\omega_{i,j} + \beta_j \omega_{i,j+1} + \beta_{j-1}\omega_{i,j-1} - \beta_{i-1}\omega_{i-1,j} + d_j^T q_i - q_j^T d_i\\ &\qquad\qquad (k+1 < j \le i-1),\\ \beta_i \omega_{i+1,i} &= \alpha_i(1 - \omega_{i,i}) - \beta_{i-1}\omega_{i,i-1} + q_i^T d_i. \end{aligned}$$

To use the recursion, we need to evaluate the quantities $\pm d_j^T q_i$. In our implementation of the ω recurrence, we simply replace $\pm d_j^T q_i$ with $\epsilon \|Aq_i\|$. The above set of equations can be used only when i is greater than $k+1$. Among the first $k+2$ Lanczos vectors, $q_1, \ldots, q_{k+1}$ are not generated by the current Lanczos iterations, and only q_{k+2} is computed in the current Lanczos iterations; see (2.3). Since the computation of q_{k+2} explicitly involved all previous vectors, the decision of whether to perform reorthogonalization has been studied [5]. We need only apply the ω-recurrence when computing q_i, $i > k+2$.

Let W denote the matrix formed from $\omega_{i,j}$. One important point to note here is that the $(i+1)$st row of W depends only on the two previous rows. This indicates that if the orthogonality levels of q_{k+1} and q_{k+2} are known, the above recurrence can be carried forward. In practice, we try to make q_{k+1} and q_{k+2} orthogonal to previous vectors to machine precision. This can prevent the loss of orthogonality among the first k vectors from significantly affecting the orthogonality level of the new vectors computed later. Since q_{k+1} after restarting is q_{m+1} before restarting, this also explains why q_{m+1}, i.e., r_m, has to be computed accurately; see Figure 3.2. For similar reasons, when q_i needs global reorthogonalization, we should orthogonalize both q_i and q_{i-1} [21].

A.2. Global reorthogonalization. The global reorthogonalization here refers to the process of applying the Gram–Schmidt procedure to explicitly orthogonalize q_i against all previous vectors. This is necessary when q_i has exceeded the user-specified limit on loss of orthogonality, say, $\|(\omega_{i,1}, \ldots, \omega_{i,i})\| \ge \sqrt{\epsilon}$. Using the ω-recurrence to simulate the loss of orthogonality, when to invoke the global reorthogonalization procedure can be easily determined. Typically, if the Gram–Schmidt procedure is invoked, it may be repeated to ensure the desired orthogonality level is achieved. The

remainder of this subsection will briefly examine the decision of how many repetitions to use. To answer this question, we need to find out how effective the Gram–Schmidt procedure is and when to stop.

Let z be an arbitrary vector and Q be a set of Lanczos vectors that are nearly orthogonal; the process of repeatedly applying the Gram–Schmidt procedure can be written as $z_{(i)} = (I - QQ^T)z_{(i-1)}$, where $z_{(0)} \equiv z$. Define $w_{(i)} \equiv Q^T z_{(i)}$; the orthogonality level between $z_{(i)}$ and Q can be measured by $\|w_{(i)}\|$. It is easy to see that $Q^T z_{(i)} = (I - Q^T Q)Q^T z_{(i-1)}$ and $\|w_{(i)}\| \leq \|I - Q^T Q\|\|w_{(i-1)}\|$. If $\|I - Q^T Q\| < 1$, repeating the Gram–Schmidt procedure will eventually reduce $\|w_{(i)}\|$ to a very small number. When the Gram–Schmidt procedure is carried out in exact arithmetic, $z_{(\infty)}$ is exactly the same as orthogonalizing z against an exactly orthonormal basis of Q. When the Gram–Schmidt procedure is carried out in finite-precision arithmetic, it is likely to produce a $z_{(\infty)}$ that is orthogonal to Q to machine precision. However, the solution may not be the same as in the exact arithmetic case.

Previously, it was argued that when q_i needs reorthogonalization, it should be orthogonalized to machine precision [21]. We adopt the same stopping criteria for our global reorthogonalization. Orthogonalizing to machine precision can be expressed as requiring $\|w_{(i)}\| < \epsilon\|z_{(\infty)}\|$. Since $z_{(\infty)}$ is not computed, the above condition can be rewritten as $\|w_{(i)}\| < \epsilon\|z_{(i)}\|$. In the process of computing $z_{(i)}$, $w_{(i-1)}$ is computed. The above equation can be expressed in known quantities as

$$\|I - Q^T Q\|\|w_{(i-1)}\| < \epsilon\|z_{(i)}\|.$$

Using the relation between $\|w_{(i)}\|$ and $\|w_{(i-1)}\|$, we can estimate $\|I - Q^T Q\|$. The above stopping criteria can be implemented efficiently. This stopping test differs from earlier ones in that it takes into account of the orthogonality of Q [5, 12]. This characteristic is important to the partial reorthogonalization since Q may be of varying orthogonality level.

A.3. Local reorthogonalization. To implement a robust Lanczos method, the local orthogonality should always be maintained. When the global reorthogonalization is not necessary, we apply an explicit local reorthogonalization. This local reorthogonalization uses the Gram–Schmidt procedure to orthogonalize q_{i+1} against q_i and q_{i-1}. It needs to be done only once, but doing so has two important consequences. It ensures that $\omega_{i+1,i}$ and $\omega_{i+1,i-1}$ are on the order of ϵ and that α_i and β_i are accurate to the machine precision. Both of these conditions are crucial ingredients in guaranteeing the accuracies of eigenvalues computed by the thick-restarted Lanczos method [28].

Acknowledgments. The authors would like to thank the referees for invaluable suggestions and comments.

REFERENCES

[1] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] D. Calvetti, L. Reichel, and D. Sorensen, *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal., 2 (1994), pp. 1–21.

[3] A. Chapman and Y. Saad, *Deflated and Augmented Krylov Subspace Techniques*, Tech. Rep. UMSI 95/181, Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN, 1995.

[4] J. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Theory*, Progress in Scientific Computing 3, Birkhäuser Boston, Boston, MA, 1985.

[5] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.
[6] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 87–94.
[7] E. R. Davidson, *Super-matrix methods*, Comput. Phys. Comm., 53 (1989), pp. 49–60.
[8] G. de Samblanx, *Filtering And Restarting Projection Methods for Eigenvalue Problems*, Ph.D. thesis, Katholieke Universiteit Leuven, Heverlee, Belgium, 1998.
[9] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
[10] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1999), pp. 94–125.
[11] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[12] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998. ARPACK Software is available at http://www.caam.rice.edu/software/ARPACK/.
[13] R. B. Lehoucq and D. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.
[14] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 156 (1991), pp. 289–309.
[15] R. B. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.
[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics Appl. Math. 20, SIAM, Philadelphia, PA, 1998.
[17] B. N. Parlett and D. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[18] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1993.
[19] Y. Saad, *Analysis of Augmented Krylov Subspace Techniques*, Tech. Rep. UMSI 95/175, Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN, 1995.
[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, 1996.
[21] H. D. Simon, *The Lanczos Algorithm for Solving Symmetric Linear Systems*, Ph.D. thesis, University of California, Berkeley, CA, 1982.
[22] H. D. Simon, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–136.
[23] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
[24] D. S. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[25] A. Stathopoulos, Y. Saad, and K. Wu, *Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods*, SIAM J. Sci. Comput., 19 (1998), pp. 227–245.
[26] L.-W. Wang and A. Zunger, *Large scale electronic structure calculations using the Lanczos method*, Computational Materials Science, 2 (1994), pp. 326–340.
[27] K. Wu, *Preconditioned Techniques for Large Eigenvalue Problems*, Ph.D. thesis, University of Minnesota, Minneapolis, MN, 1997. An updated version also appears as Tech. Rep. TR97-038 at the Computer Science Department, University of Minnesota, Minneapolis, MN.
[28] K. Wu and H. Simon, *Thick-Restart Lanczos Method for Symmetric Eigenvalue Problems*, Tech. Rep. 41412, Lawrence Berkeley National Laboratory, Berkeley, CA, 1998.
[29] K. Wu and H. D. Simon, *Dynamic Restarting Schemes for Eigenvalue Problems*, Tech. Rep. LBNL-42982, Lawrence Berkeley National Laboratory, Berkeley, CA, 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 617–626

MOORE–PENROSE INVERSE OF MATRICES ON IDEMPOTENT SEMIRINGS*

S. PATI†

Abstract. We characterize matrices over an idempotent semiring satisfying some additional necessary conditions for which the Moore–Penrose inverse exists. The "$(\max, \times)$ semiring," defined as the set of nonnegative real numbers $\mathbb{R}^+$, equipped with the operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = ab$ is an example of such a semiring. The "$(\max, +)$ semiring", defined as the set of real numbers including $-\infty$, equipped with the operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = a + b$ is another example. Some of our results generalize known results in the case of the binary boolean algebra (a trivial idempotent semiring). We give an algorithm to compute the Moore–Penrose inverse, when it exists. We also make comparisons with similar results over the conventional algebra.

Key words. Moore–Penrose inverse, $(\max, \times)$ semiring, $(\max, +)$ semiring, boolean algebra

AMS subject classification. 15A09

PII. S0895479899355517

1. Introduction.

A *commutative semiring* is a set $\mathcal{D}$ with two associative, commutative operations $\oplus$ and $\otimes$ (called the sum and the product, respectively), satisfying the following properties:

(1) there exists an element $\mathbf{0}$ (called the "zero" element) such that $a \oplus \mathbf{0} = a$, $\forall\, a \in \mathcal{D}$;
(2) there exists an element $\mathbf{1}$ (called the "identity" element) such that $a \otimes \mathbf{1} = a$, $\forall\, a \in \mathcal{D}$;
(3) $\otimes$ is distributive over $\oplus$, that is, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$, $\forall\, a, b, c \in \mathcal{D}$; and
(4) the zero element absorbs every other element with respect to the product, that is, $a \otimes \mathbf{0} = \mathbf{0}$, $\forall\, a \in \mathcal{D}$.

We shall consider only commutative semirings in which the operation $\oplus$ is idempotent, that is, $a \oplus a = a$, $\forall\, a \in \mathcal{D}$. The operation $\oplus$ induces a partial ordering $\leq$ on the set $\mathcal{D}$ according to the rule $a \leq b$ if and only if $a \oplus b = b$. So, $\mathbf{0}$ is the smallest element in $\mathcal{D}$. All other elements of $\mathcal{D}$ shall be called *positive.* It can be easily seen that $\forall\, a, b, c \in \mathcal{D}$,

(1) $a \leq b \Rightarrow a \oplus c \leq b \oplus c$, and
(2) $a \leq b \Rightarrow a \otimes c \leq b \otimes c$.

The semiring is called *total* if the induced partial order is a total order, in which case $a \oplus b$ is $\max\{a, b\}$.

Example 1. The following are some important examples of semirings. A good overview of other semirings in which $\oplus$ is idempotent can be found in [11].

(1) The set of nonnegative real numbers with the operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = a \times b$ (the usual product). The partial order $\leq$ induced by $\oplus$ is a total order here. This semiring shall be referred as $\mathbb{R}_{\max,\times}$.
(2) The set of real numbers, including $-\infty$, with the operations $a \oplus b = \max\{a, b\}$ and $a \otimes b = a + b$. This is called the "$(\max, +)$ semiring," denoted by $\mathbb{R}_{\max}$. Note that the function $f(x) = e^x$ is an isomorphism from $\mathbb{R}_{\max}$ to $\mathbb{R}_{\max,\times}$.
(3) The "boole algebra" in which the set is $\{0, 1\}$ and the operations are $a \oplus b = \max\{a, b\}$ and $a \otimes b = a \times b$ (the usual product).

*Received by the editors May 4, 1999; accepted for publication (in revised form) by R. Bhatia May 17, 2000; published electronically September 15, 2000.
http://www.siam.org/journals/simax/22-2/35551.html
†Department of Mathematics, University of Regina, Regina, Saskatchewan, Canada, S4S 0A2 (pati@math.uregina.ca).

We note that given any semiring $\mathcal{D}$, the subset $\{\mathbf{0}, \mathbf{1}\}$ of $\mathcal{D}$ with the operations $\oplus$ and $\otimes$ behaves like the boole algebra.

CONDITION 1 (cancellation). *A semiring $\mathcal{D}$ is said to satisfy the cancellation condition if the relations $a \otimes b = a \otimes c$ and $a \neq \mathbf{0}$ imply $b = c$.* Note that if $\mathcal{D}$ satisfies the cancellation condition and $a, b \in \mathcal{D}, a \neq \mathbf{0}, a \otimes b = \mathbf{0}$, then $b = \mathbf{0}$. Thus in $\mathcal{D}$ the product of two nonzero numbers is nonzero. Let a^n denote the product $a \otimes a \otimes \cdots \otimes a$ (n factors). The following observation can be found in [7].

PROPOSITION 1.1. *Suppose $\mathcal{D}$ satisfies the cancellation condition. Suppose n is a natural number and $a \in \mathcal{D}$. If there exists a solution λ to the equation $x^n = a$, then it is unique.*

CONDITION 2 (algebraic completeness). *A semiring $\mathcal{D}$ is called algebraically complete if the equation $x^n = a$ has a solution x for every natural number n and every $a \in \mathcal{D}$.* It follows from Proposition 1.1 that if an algebraically complete semiring $\mathcal{D}$ satisfies the cancellation condition, then the equation $x^n = a$ has a unique solution in $\mathcal{D}$.

CONDITION 3 (stabilization). *A semiring is said to satisfy the stabilization condition if $\forall\ \lambda, \mu, a, b \in \mathcal{D}\ (a \neq 0, b \neq 0)$ there exists an element $c \in \mathcal{D}$ and a positive integer m such that for every $n \geq m$ we have*

$$a \otimes \lambda^n \oplus b \otimes \mu^n = c \otimes (\lambda \oplus \mu)^n.$$

DEFINITION 1.2. *Let $\mathcal{D}$ be a semiring. A set $\mathcal{M}$ is called a* semimodule *over $\mathcal{D}$, if the following conditions are satisfied.*

(1) *There is an operation $\oplus$ defined on $\mathcal{M}$ which is associative and commutative, and there exists a zero element $\mathbf{0}$ with respect to this operation.*
(2) *There is a multiplication $\otimes$ of elements of $\mathcal{M}$ by the elements of $\mathcal{D}$ which satisfies*

$$\lambda \otimes m \in \mathcal{M} \qquad \forall\ \lambda \in \mathcal{D},\ m \in \mathcal{M},$$

$$(\lambda \otimes \mu) \otimes m = \lambda \otimes (\mu \otimes m) \qquad \forall\ \lambda, \mu \in \mathcal{D},\ m \in \mathcal{M},$$

$$\lambda \otimes (m_1 \oplus m_2) = \lambda \otimes m_1 \oplus \lambda \otimes m_2 \qquad \forall\ \lambda \in \mathcal{D},\ m_1, m_1 \in \mathcal{M},$$

$$\mathbf{0} \otimes m = \lambda \otimes \mathbf{0} = \mathbf{0} \qquad \forall\ \lambda \in \mathcal{D},\ m \in \mathcal{M},$$

$$\bigoplus_\alpha \lambda_\alpha \in \mathcal{D},\ m \in \mathcal{M} \Rightarrow \left(\bigoplus_\alpha \lambda_\alpha\right) \otimes m = \bigoplus_\alpha (\lambda_\alpha \otimes m).$$

Henceforth, by $\mathcal{D}$ we will denote a commutative, algebraically complete semiring in which the cancellation and stabilization conditions hold, the operation $\oplus$ is idempotent, and the partial ordering "$\leq$" induced by $\oplus$ is total.

Example 2. Let $\mathcal{D}$ be a semiring. Consider the set $\mathcal{D}^n$ of n-tuples (column vectors) of elements of $\mathcal{D}$. This set is a semimodule over $\mathcal{D}$ with respect to the following operations: If (a_i) and (b_i) are in $\mathcal{D}^n$ and $\lambda \in \mathcal{D}$, then

$$(a_i) \oplus (b_i) = (a_i \oplus b_i) \text{ and } \lambda \otimes (a_i) = (\lambda \otimes a_i).$$

Note also that for any $(a_i) \in \mathcal{D}^n$, $(a_i) \oplus (a_i) = (a_i)$.

DEFINITION 1.3. *If $\mathcal{M}$ and $\mathcal{N}$ are semimodules over the same semiring K, then a mapping $f : \mathcal{M} \to \mathcal{N}$ satisfying*

$$f(\lambda \otimes m_1 \oplus \mu \otimes m_2) = \lambda \otimes f(m_1) \oplus \mu \otimes f(m_2) \qquad \forall\ m_1, m_2 \in \mathcal{M},\ \lambda, \mu \in K$$

is called a homomorphism. *A homomorphism from a semimodule $\mathcal{M}$ to itself is called an* endomorphism.

Let $\mathcal{D}$ be a semiring. An $m \times n$ matrix A can be viewed as a homomorphism from $\mathcal{D}^n$ to $\mathcal{D}^m$, $f(x) = A \otimes x$, where $A \otimes x \in \mathcal{D}^m$ whose ith entry is $\bigoplus_{k=1}^{n} a_{ik} \otimes x_k$, $i = 1, \ldots, m$. We shall write ab to mean $a \otimes b$. The (i, j)th entry of a matrix A is denoted by a_{ij} or $A(i, j)$. A matrix with each entry positive is called *positive*. We define $A \oplus B = C$, where $c_{ij} = a_{ij} \oplus b_{ij}$, when A, B are of the same order. When A is of order $m \times n$ and B is of order $n \times k$, we define $A \otimes B = C$, where $c_{ij} = \bigoplus_r a_{ir} b_{rj}$. The product of two matrices can be viewed as the composition of two homomorphisms. In case there is no ambiguity, AB means $A \otimes B$. For matrices A, B of the same order, $A \geq B$ means $a_{ij} \geq b_{ij}\ \forall\ i, j$. The identity matrix of an appropriate order is denoted by I. The transpose of A, denoted by A^t, is defined in the usual way.

DEFINITION 1.4. *Let A be of order $m \times n$. The* Moore–Penrose inverse *of A, denoted by A^+, is defined to be an $n \times m$ matrix G such that*

$$AGA = A,\ GAG = G,\ (AG)^t = AG,\ \textit{and}\ (GA)^t = GA.$$

It is easy to see that the Moore–Penrose inverse of a matrix A, if it exists, is unique. In fact, given A, if G_1 and G_2 are two matrices satisfying all of the four conditions given above, then

$$\begin{aligned} G_1 &= G_1 \overbrace{A}\, G_1 = \overbrace{G_1 A}\, \overbrace{G_2 A}\, G_1 = \overbrace{A^t G_1^t A^t}\, G_2^t G_1 = \overbrace{A^t G_2^t}\, G_1 \\ &= G_2 \overbrace{A G_1}\, = G_2 G_1^t \overbrace{A^t}\, = G_2 \overbrace{G_1^t A^t}\, \overbrace{G_2^t A^t}\, = G_2 \overbrace{A G_1 A}\, G_2 = G_2 A G_2 = G_2. \end{aligned}$$

The Moore–Penrose inverse over the boole algebra has been well studied (see, for example, [13] for some interesting results). It is well known that a matrix A over the boole algebra admits a Moore–Penrose inverse if and only if $AA^tA = A$. The boole algebra being the trivial subsemiring of the semiring $\mathbb{R}_{\max,\times}$ it is interesting to determine necessary and sufficient conditions for a matrix over the semiring $\mathbb{R}_{\max,\times}$, which admits a Moore–Penrose inverse. In general we ask the following question.

Question. Let $\mathcal{D}$ be a commutative, algebraically complete semiring in which the cancellation and stabilization condition holds and in which $\oplus$ is idempotent and the partial ordering "$\leq$" induced by $\oplus$ is total. Let A be a matrix over $\mathcal{D}$. What are the necessary and sufficient conditions on A so that A admits a Moore–Penrose inverse?

Another motivation for studying the Moore–Penrose inverse over idempotent semirings is the fact that the idempotent semiring $\mathbb{R}_{\max,\times}$ has applications in areas like optimal control and discrete event systems (see, for example, [6],[8],[10],[11]).

In this paper we present a solution to the question asked above. We also provide a program which is a MATLAB .m function that helps to find the Moore–Penrose inverse of a matrix A, if it exists. We compare some of the results here with similar results on nonnegative matrices over the real numbers.

2. The main result. Henceforth, unless otherwise stated, all matrices considered are over $\mathcal{D}$. For any matrix A, $p(A)$ will mean the pattern matrix of A, i.e., the

matrix whose (i,j)th entry is $\mathbf{1}$ if and only if the (i,j)th entry of A is not $\mathbf{0}$; otherwise it is $\mathbf{0}$. Thus $p(A)$ can be treated as a binary boolean matrix.

PROPOSITION 2.1. *Suppose A^+ exists for the matrix A. Then $(p(A))^+$ exists and $(p(A))^+ = p(A^+)$. Thus $(p(A))^+$ exists over the boole algebra.*

Proof. First we show that if A, B are two matrices over $\mathcal{D}$, then $p(AB) = p(A)p(B)$. To see this let the (i,j)th entry $p(AB)_{ij} = \mathbf{1}$. Then $(AB)_{ij} \neq \mathbf{0}$. That is $\bigoplus_k A_{ik}B_{kj} \neq \mathbf{0}$. So there exists s such that $A_{is}B_{sj} \neq \mathbf{0}$. It follows that $A_{is} \neq \mathbf{0}$ and $B_{sj} \neq \mathbf{0}$. Thus $p(A)_{is} = \mathbf{1}$ and $p(B)_{sj} = \mathbf{1}$. Thus $(p(A)p(B))_{ij} = \mathbf{1}$.

Conversely, let $(p(A)p(B))_{ij} = \mathbf{1}$. So there exists s such that $p(A)_{is} = \mathbf{1}$ and $p(B)_{sj} = \mathbf{1}$. Thus $A_{is} \neq \mathbf{0}, B_{sj} \neq \mathbf{0}$, and so the product $A_{is}B_{sj} \neq \mathbf{0}$ (because of the cancellation condition). Hence $\bigoplus_k A_{ik}B_{kj} \neq \mathbf{0}$.

Now, $p(A)p(A^+)p(A) = p(AA^+A) = p(A)$, since $AA^+A = A$. Similar arguments show that $p(A^+)$ satisfies the other three conditions for being the Moore–Penrose inverse of $p(A)$. □

PROPOSITION 2.2. *Let A be of order $m \times n$ and suppose A^+ exists. Then there exist permutation matrices P, Q such that*

$$PAQ = \begin{bmatrix} F_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & F_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & F_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where the blocks F_i are positive but not necessarily square.

Proof. Consider the matrix $p(A)$. By Proposition 2.1, $p(A)^+$ exists over the boole algebra. It is well known (see, for example, [14], [4]) that there exist permutation matrices P, Q such that $Pp(A)Q$ is a direct sum of some "all ones" matrices and a "zero" matrix. The desired result now follows easily. □

It is known (see, for example, Theorem 3.4.3 of [13] or Theorem 1.1(v) of [4]) that if A is a matrix over the boole algebra and A^+ exists, then any two rows of A are either equal or have disjoint sets of 1s. Keeping this in mind it is easy to compute the matrices P, Q, mentioned in Proposition 2.2, for a matrix A over $\mathcal{D}$ for which A^+ exists. We present a MATLAB program to compute P, Q.

Given: A is an $m \times n$ binary boolean matrix and A^+ exists.

```
function [p,q]=temp(a)
        echo off
        [m,n]=size(a); p=eye(m); q=eye(n); s=1; c=0;
        while s<m, k=s; j=k;
        if a(k,:)==0, ult=m+1;
        for v=k+1:m,
        if sum(a(v,:))>0, ult=v; end; end; j=ult; end;
        if j<=m, b=a(k,:); a(k,:)=a(j,:); a(j,:)=b; b=a(k,:);
        pl=eye(m); pl(k,k)=0; pl(j,j)=0; pl(j,k)=1; pl(k,j)=1; p=pl*p; c1=c+1;
        for l=c1:n,
        if b(l)==1, c=c+1; qc=eye(n); qc(l,l)=0; qc(c,c)=0; qc(c,l)=1; qc(l,c)=1;
        q=q*qc; a=a*qc; end; end;
        for i=k+1:m,
        if a(i,:)==a(k,:), s=s+1; pl=eye(m); pl(i,i)=0; pl(s,s)=0; pl(i,s)=1; pl(s,i)=1;
        p=pl*p;
        a=pl*a; end; end; s=s+1; end;
```

if j>m, s=m; end; end

The following is an easy observation which is stated without proof.

PROPOSITION 2.3. *Let A, B be two matrices. Then the Moore–Penrose inverse of the matrix* $\left[\begin{smallmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{smallmatrix}\right]$ *exists if and only if A^+, B^+ exist and, in this case,*

$$\begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}^+ = \begin{bmatrix} A^+ & \mathbf{0} \\ \mathbf{0} & B^+ \end{bmatrix}.$$

We note two simple facts here.

(i) If A has Moore–Penrose inverse A^+, then for any permutation matrices P, Q $(PAQ)^+$ exists and is $Q^tA^+P^t$.

(ii) For a permutation matrix P, the Moore–Penrose inverse is P^t. In view of Propositions 2.2 and 2.3, it is clear that we need to discuss the Moore–Penrose inverse of a positive matrix. The next lemma is an easy observation.

LEMMA 2.4. *Suppose the element $k \in \mathcal{D}$ is invertible (denote the inverse of k by k^{-1}). If A^+ exists, then $(kA)^+ = k^{-1}A^+$, where $kA(i,j) = ka_{ij}$.*

We know that to any square matrix A there corresponds a directed graph $G(A)$ whose weighted adjacency matrix is A. (The edge from the vertex i to the vertex j bears weight w if $a_{ij} = w$; if $a_{ij} = \mathbf{0}$, then there is no edge starting from i and ending at j. The existence of a loop is allowed.) The matrix A is called *irreducible* (the corresponding endomorphism is called *indecomposable*) if there is a directed path between every pair of distinct vertices in the graph $G(A)$. Suppose A is an irreducible matrix of order n. Let $\phi(n)$ denote the least common multiple of the numbers $1, 2, \ldots, n$. Let λ be the root of the equation

$$\lambda^{\phi(n)} = \bigoplus_{\substack{k=1,2,\ldots,n \\ (i_1,i_2,\ldots,i_k)}} [a_{i_1i_2}a_{i_2i_3}\ldots a_{i_ki_1}]^{\frac{\phi(n)}{k}}, \tag{1}$$

where the summation is over k and for each k, over all k-tuples of indices $(i_1, i_2, \ldots, i_k)$.

We note that if $\mathcal{D} = \mathbb{R}_{\max}$ and A is irreducible, then the root of the above equation is nothing but the only eigenvalue of the matrix A, which is also equal to the maximum cycle arithmetic mean. (In the case of $\mathbb{R}_{\max,\times}$ it corresponds to the maximum cycle geometric mean.) For some discussions on eigenvalues and eigenvectors of a matrix over $\mathbb{R}_{\max,\times}$ or $\mathbb{R}_{\max}$, refer to [2], [3], [9], [11], [15]. Karp [12] has supplied an effective algorithm for computing the minimum cycle arithmetic mean in a strongly connected edge-weighted digraph. Given an irreducible matrix A over $\mathbb{R}_{\max}$, the minimum cycle arithmetic mean of the digraph corresponding to the matrix $-A$ is simply the maximum cycle arithmetic mean of the digraph corresponding to A. Thus using Karp's algorithm one can compute the root of (1) for an irreducible matrix over $\mathbb{R}_{\max}$.

The following result can be found in [7].

PROPOSITION 2.5. *Let A be an irreducible matrix and let λ be the root of* (1). *Then there exist natural numbers m_1 and m_2 such that $m_1 > m_2$ and*

$$A^{m_1} = \lambda^{m_1 - m_2}A^{m_2}.$$

The next corollary follows easily from Proposition 2.5.

COROLLARY 2.6. *Let A be a symmetric irreducible matrix and let λ be the root of* (1). *Suppose A^+ exists. Then there exists a natural number $m > 1$ such that*

$$A^m = \lambda^{m-1}A.$$

Proof. By Proposition 2.5, there exist natural numbers m_1 and m_2 such that $m_1 > m_2$ and

$$A^{m_1} = \lambda^{m_1-m_2} A^{m_2}.$$

Let r be the smallest natural number such that

$$A^{m_1} = \lambda^{m_1-r} A^r. \tag{2}$$

If $r = 1$, then we have nothing to prove. So, let $r > 1$. It follows from (2) that $A^{m_1}A^+ = \lambda^{m_1-r}A^rA^+$. Since A^+A and A are both symmetric and $(A^+)^t = (A^t)^+$, it follows that $A^{m_1-1}A^+A = \lambda^{m_1-r}A^{r-1}A^+A$. Since $AA^+A = A$, we have $A^{m_1-1} = \lambda^{m_1-r}A^{r-1}$, a contradiction to the minimality of r. □

The following result is crucial for further developments.

LEMMA 2.7. *Let $A_{n\times n} > \mathbf{0}$ be a symmetric matrix and assume that $a_{11} = \mathbf{1}$ is the largest entry in A. Assume that the entries of A satisfy the following condition: $a_{ij} = \mathbf{1} \Rightarrow a_{ii} = a_{jj} = \mathbf{1}$. Then there exists a natural number m such that $A^m = A^{m+1}$.*

Proof. It suffices to show that for any given i, j there exists a natural number m such that $A^m_{ij} = A^{m+r}_{ij}$, $r \in \mathbb{N}$. To this extent note that A^m_{ij} is the maximum of the weights of walks of length m from i to j in $G(A)$, where the length of a walk is the number of edges contained in it and the weight of a walk is the product of the weights of the edges in it. Let W_{ij} be a walk of length t_0 from i to j passing through the vertex v such that $a_{vv} = \mathbf{1}$. (The existence of such a walk is guaranteed by the hypothesis.) Then

$$\forall t \in \mathbb{N}, t \geq t_0,\ A^t_{ij} \geq \text{weight}(W_{ij}), \tag{3}$$

where weight(W_{ij}) is the weight of the walk W_{ij}. Let b be the second maximal entry ($b \neq \mathbf{1}$) in A. By the stability condition

$$\exists k \in \mathbb{N} \text{ such that } b^{k+r} < \text{ weight}(W_{ij}) \qquad \forall\, r \in \mathbb{N}. \tag{4}$$

Let $m = \max\{t_0, kn\}$. Let W^m be the walk of length m from i to j such that $a^m_{ij} = \text{weight}(W^m)$. The walk W^m passes through a vertex v such that $a_{vv} = \mathbf{1}$, since otherwise $a^m_{ij} = \text{weight}(W^m) \leq b^m < \text{weight}(W_{ij})$, by (4), which contradicts (3). It follows that $A^{m+1}_{ij} \geq A^m_{ij}$.

On the other hand, since W^{m+1} can contain at most k edges of weight less than $\mathbf{1}$, it follows from the pigeonhole principle that W^{m+1} contains a walk W of length at least $n+1$ such that weight$(W) = \mathbf{1}$. Thus W contains a cycle $\Gamma \equiv [u_1, u_2, \ldots, u_1]$. By the hypothesis, $a_{u_1u_1} = \mathbf{1}$. Thus, without loss we can assume $\Gamma = [u_1, u_1, \ldots, u_1]$. Let W' be the walk obtained from W^{m+1} by deleting a single loop at u_1. Clearly W' is a walk of length m from i to j and weight(W') = weight(W^{m+1}). Thus $A^m_{ij} \geq A^{m+1}_{ij}$, and hence, $A^m_{ij} = A^{m+1}_{ij}$. □

We note that in the above lemma $\lambda = \mathbf{1}$ is the root of (1) for A, and thus the lemma discloses a relationship among m_1, m_2 and λ beyond what was observed in Proposition 2.5, under some additional assumptions. An immediate corollary to the above lemma is the following.

COROLLARY 2.8. *Let $A_{n\times n} > \mathbf{0}$ be a symmetric matrix. Then the following are equivalent.*

(i) *The largest entry in A is $\mathbf{1}$, the entries of A satisfy the condition $a_{ij} = \mathbf{1} \Rightarrow a_{ii} = a_{jj} = \mathbf{1}$, and A^+ exists.*

(ii) *A is idempotent.*

Proof. (i) $\Rightarrow$ (ii) follows from Lemma 2.7, by using the technique used in the proof of Corollary 2.6.

To see (ii) $\Rightarrow$ (i) it is easy to check that $A^+ = A$. Let a_{ij} be the maximum among the entries of A. Note that $a_{ij} \geq \mathbf{1}$. Because if $a_{ij} < \mathbf{1}$, then the (i,j)th entry of A^2 is

$$\bigoplus_k a_{ik}a_{kj} < \bigoplus_k a_{ik}\mathbf{1} = a_{ij},$$

which is a contradiction to the fact that $A^2 = A$. Now, from $A^2 = A$ we get that

$$a_{ii} = \bigoplus_k a_{ik}a_{ik} = a_{ij}a_{ij} \geq a_{ij},$$

since $a_{ij} \geq \mathbf{1}$. Thus $a_{ii} \geq \mathbf{1}$ is the maximum entry. Now it is easy to see that $a_{ii} = \mathbf{1}$. ∎

The following is another crucial observation.

LEMMA 2.9. *Let $A > \mathbf{0}$ and suppose A^+ exists. Let k be the maximum among the entries of A. Then k is invertible in $\mathcal{D}$.*

Proof. Note that A^+ exists if and only if $(PAQ)^+$ exists, where P, Q are permutation matrices of suitable orders. Thus, we may assume that a_{11} is the maximum entry of A.

Let $B = AA^+$ and $C = A^+A$. Since B and C are idempotent, by Corollary 2.8, the largest entry in B is $\mathbf{1}$ and the largest entry in C is $\mathbf{1}$. Suppose that $b_{1s} < 1, \forall\ s$. Since $BA = A$, we have

$$a_{11} = \bigoplus_s b_{1s}a_{s1} < \bigoplus_s a_{s1} \leq a_{11},$$

a contradiction.

So, let $b_{1r} = \mathbf{1}$. By Corollary 2.8, $b_{11} = \mathbf{1}$. Thus for some l, $a_{1l}a_{l1}^+ = \mathbf{1}$.

Now,

$$\mathbf{1} \geq c_{l1} = \bigoplus_s a_{ls}^+ a_{s1} \geq a_{l1}^+ a_{11} \geq a_{l1}^+ a_{1l} = \mathbf{1}.$$

So we have $a_{l1}^+ a_{11} = \mathbf{1}$ and a_{11} is invertible. ∎

The following is one of our main results.

THEOREM 2.10. *Suppose $A > 0$ and the largest entry of A is $\mathbf{1}$. Then A^+ exists if and only if $A = AA^tA$. In this case $A^+ = A^t$.*

Proof. Suppose A^+ exists. Let $B = AA^t$. Note that B^+ exists and $B^+ = (A^+)^tA^+$. Thus, by Corollary 2.8, $B^2 = B$. That is, $AA^tAA^t = AA^t$. Thus, $A^+AA^tAA^t = A^+AA^t$. Since A^+A is symmetric, we have $A^t(A^+)^tA^tAA^t = A^t(A^+)^tA^t$. Since $(A^+)^t = (A^t)^+$ and $AA^+A = A$, it follows that $A^tAA^t = A^t$. Thus, $AA^tA = A$. Now observe that $A^+AA^tAA^+ = A^+AA^+$ (by multiplying A^+ on both sides). The left-hand side is equal to $A^t(A^+)^tA^t(A^+)^tA^t = A^t$ and the right-hand side is equal to A^+.

Conversely, given $A = AA^tA$, it is easy to check that A^t satisfies all conditions for A^+. ∎

Remark. Let D_n be the set of $n \times n$ doubly stochastic matrices in the conventional algebra (that is, $(\mathbb{R}, +, \times)$). An element $A \in D_n$ is called regular in D_n if there exists some $G \in D_n$ such that $AGA = A$. It is known (see Theorem 5.5 of Chapter 3 in [5]) that *if $A \in D_n$ is regular in D_n, then $A^+ = A^t$*. Theorem 2.10 may be thought of as a weaker version of this. Considering the matrix $\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$ we observe that a similar statement to that of Theorem 5.5 of Chapter 3 in [5] is not possible.

COROLLARY 2.11. *Suppose $A > 0$ and let k be the largest entry of A. Then A^+ exists if and only if $k^{-2}AA^tA = A$, in which case $A^+ = k^{-2}A^t$.*

Proof. Suppose A^+ exists. By Lemma 2.9, k is invertible. Let $B \equiv k^{-1}A$. Then by Lemma 2.4, B^+ exists and by Theorem 2.10 we have $B^+ = B^t = k^{-1}A^t$. Thus, $BB^+B = B$, that is, $k^{-1}Ak^{-1}A^tk^{-1}A = k^{-1}A$, or $k^{-2}AA^tA = A$.

Conversely, given $k^{-2}AA^tA = A$, it is easy to see that $k^{-2}A^t$ satisfies all conditions to be the Moore–Penrose inverse of A. □

Now we combine all our discussions to state one of the main results.

THEOREM 2.12. *Let A be a matrix on $\mathcal{D}$. The following are equivalent.*

(i) *A^+ exists.*

(ii) *There exist permutation matrices P, Q such that*

$$PAQ = \begin{bmatrix} F_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & F_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & F_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $F_i > \mathbf{0}$ are not necessarily square. Let m_i be the maximal entry in F_i, $i = 1, 2, \ldots, k$, respectively. Then

$$F_iF_i^tF_i = m_i^2F_i \qquad \forall i = 1, 2, \ldots, k.$$

In this case,

$$Q^tA^+P^t = \begin{bmatrix} \frac{F_1^t}{m_1^2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{F_2^t}{m_2^2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{F_k^t}{m_k^2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

(iii) *There exists an invertible diagonal matrix D such that $A^+ = DA^t$.*

Proof. Suppose A^+ exists. From Proposition 2.2, it follows that there exist permutation matrices P, Q such that PAQ is the direct sum of some positive matrices F_i and a zero matrix (not necessarily square). Using Proposition 2.3, we see that each of these positive matrices has a Moore–Penrose inverse and an application of Corollary 2.11 completes the proof of (i) ⇒ (ii).

To see (ii) ⇒ (iii) note that

$$Q^tA^+P^t = D(PAQ)^t = DQ^tA^tP^t = Q^tD'A^tP^t, \tag{5}$$

where D is the diagonal matrix defined as $d_{ii} = \frac{1}{m_j^2}$, if a nonzero entry of the ith row of PAQ is in F_j and $d_{ii} = \mathbf{1}$ otherwise. The last equality in the above equation follows,

since DQ^t can be viewed as Q^tD' for some diagonal matrix D'. Now multiplying by Q on the left and by P on the right in the above equation completes the proof. ∎

Remark. It is known (see [5, Chapter 5, Theorem 5.2]) in the conventional algebra that if A is a nonnegative matrix, then $A^+ \geq 0$ if and only if $A^+ = DA^t$ for some positive diagonal matrix D. It may be noted here that in Theorem 2.12 we have a similar statement. Here, since all the matrices are nonnegative, A^+ exists if and only if $A^+ = DA^t$ for some positive diagonal matrix. Because of (iii), we can now use the program given earlier in this paper to compute the Moore–Penrose inverse of a matrix over $\mathcal{D}$, if it exists. After computing the permutation matrices we only have to look at the diagonal blocks and find the maximums. Then we can compute positive diagonal matrix D as specified in (iii).

Before stating the next result we need some discussion. Let $A > \mathbf{0}$ have the largest entry $\mathbf{1}$ and suppose that A^+ exists. Consider the matrix B defined as $b_{ij} = a_{ij}$ if $a_{ij} = \mathbf{1}$, otherwise $b_{ij} = \mathbf{0}$. Considering B as a binary boolean matrix it is easy to see that B^+ exists. Thus, by Theorem 2.12, there exist permutation matrices P, Q such that

$$PBQ = \begin{bmatrix} J_1 & 0 & \cdots & 0 & 0 \\ 0 & J_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & J_k & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix},$$

where J_i are not necessarily square. Let $C = PAQ$. So,

$$C = \begin{bmatrix} J_1 & C_{12} & \cdots & C_{1k} & C_{1k+1} \\ C_{21} & J_2 & \cdots & C_{2k} & C_{2k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{k1} & C_{k2} & \cdots & J_k & C_{kk+1} \\ C_{k+1,1} & C_{k+1,2} & \cdots & C_{k+1,k} & C_{k+1,k+1} \end{bmatrix}, \tag{6}$$

where the blocks other than $J_1, J_2, \ldots, J_k$ have entries less than $\mathbf{1}$. From the hypothesis that A^+ exists it follows that C^+ exists, and by Theorem 2.10 $C^+ = C^t$. Thus, $CC^tC = C$, and hence,

$$C_{ij} = C_{ij}J_j^tJ_j \qquad \text{if } j \neq k+1, i \neq j,$$

$$C_{ij} = J_iJ_i^tC_{ij} \qquad \text{if } i \neq k+1, i \neq j.$$

Thus we conclude that all the off-diagonal blocks in C are constant blocks.

Let us write C in the following block form:

$$C = \begin{bmatrix} L & Y \\ X^t & C_{k+1,k+1} \end{bmatrix},$$

where $X^t = [C_{k+1,1}, C_{k+1,2}, \ldots, C_{k+1,k}]$ and $Y^t = [C_{1,k+1}, C_{2,k+1}, \ldots, C_{k,k+1}]$.

Since $CC^tC = C$, we get

$$C_{k+1,k+1} = XL^tY \oplus C_{k+1,k+1}Y^tY \oplus XX^tC_{k+1,k+1} \oplus C_{k+1,k+1}C_{k+1,k+1}^tC_{k+1,k+1}. \tag{7}$$

Noting that the entries $X, Y, C_{k+1,k+1}$ are all less than $\mathbf{1}$, it follows from (7) that $C_{k+1,k+1} = X^tL^tY$. It is easy to see that $L^tY = Y$ and $XL^t = X$. This discussion is

summarized in the following theorem which gives the structure of $A > \mathbf{0}$, when A^+ exists.

THEOREM 2.13. *Suppose $A > 0$ has largest entry $\mathbf{1}$ and A^+ exists. Then there exist permutation matrices P, Q such that*

$$PAQ = \begin{bmatrix} L & L^tY \\ X^tL^t & X^tL^tY \end{bmatrix} = \begin{bmatrix} L & Y \\ X^t & X^tY \end{bmatrix},$$

where the blocks in L are produced according to the J-blocks in PAQ. Each diagonal block of L is a J-block. The entries in an off-diagonal block are all equal to a number less than $\mathbf{1}$, and X, Y are two matrices whose largest entry is less than $\mathbf{1}$.

As an immediate consequence we get a description of A, if $A > 0$ and A^+ exists (by considering $\frac{A}{k}$ where k is the maximal entry in A). Hence the structure of the matrices $\frac{F_i}{m_i}$, where $F_i, i = 1, 2, \ldots, k$ are as in Theorem 2.12 and m_i is the largest element of F_i, is the same as that of the matrix PAQ in Theorem 2.13.

3. Conclusions. We have given (in Theorem 2.12) necessary and sufficient conditions for the existence of the Moore–Penrose inverse A^+ of a matrix A over any idempotent semiring. In case A^+ exists, we have given a description of it so that it can be easily computed.

Acknowledgments. The author sincerely thanks an anonymous referee for many helpful comments and suggestions. The author also thanks Prof. R. B. Bapat for suggesting the problem and for many helpful comments.

REFERENCES

[1] F. BACCELI, G. COHEN, G. J. OLSDER, AND J. P. QUADRAT, *Synchronization and Linearity*, John Wiley and Sons, New York, 1992.
[2] R. B. BAPAT, *A max version of the Perron-Frobenius theorem*, Linear Algebra Appl., 275/276 (1998), pp. 3–18.
[3] R. B. BAPAT, D. P. STANFORD, AND P. VAN DEN DRIESSCHE, *Pattern properties and spectral inequalities in max algebra*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 964–976.
[4] R. B. BAPAT, S. K. JAIN, AND S. PATI, *Weighted Moore–Penrose inverse of a boolean matrix*, Linear Algebra Appl., 255 (1997), pp. 267–279.
[5] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
[6] R. A. CUNINGHAME-GREEN, *Minimax Algebra*, Lecture Notes in Econom. and Math. Systems 166, Springer-Verlag, Berlin, Germany, 1979.
[7] P. I. DUDINIKOV AND S. N. SAMBORSKII, *Endomorphisms of finitely generated free semirings*, Advances in Soviet Mathematics, 13 (1992), pp. 65–85.
[8] S. GAUBERT, *Methods and Applications of* (max, +) *Linear Algebra*, Lecture Notes in Comput. Sci. 1200, Springer, Berlin, 1997.
[9] S. GAUBERT, *Théorie des Systémes Linéaires dans des Dioïdes*, Ph.D. Thesis, L'Ecole des Mines de Paris, Paris, France, 1992.
[10] M. GONDRAN AND M. MINOUX, *Graphs and Algorithms*, John Wiley and Sons, Chichester, UK, 1984.
[11] M. GONDRAN AND M. MINOUX, *Linear algebra in dioids: A survey of recent results*, Ann. of Discrete Math., 19 (1984), pp. 147–163.
[12] R. M. KARP, *A characterization of the minimum cycle mean in a digraph*, Discrete Math., 23 (1978), pp. 309–311.
[13] K. H. KIM, *Boolean Matrix Theory and Applications*, Marcel Dekker, New York, 1982.
[14] P.S.S.N.V.P. RAO AND K.P.S.B. RAO, *On generalized inverses of boolean matrices*, Linear Algebra Appl., 11 (1975), pp. 135–153.
[15] W. CHEN, X. QI, AND S. DENG, *The eigen problem and period analysis of the discrete event system*, Systems Sci. Math. Sci., 3 (1990), pp. 243–260.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 2, pp. 627–645

SPECTRAL STRUCTURES OF IRREDUCIBLE TOTALLY NONNEGATIVE MATRICES*

SHAUN M. FALLAT†, MICHAEL I. GEKHTMAN‡, AND CHARLES R. JOHNSON§

Abstract. An n-by-n matrix is called totally nonnegative if every minor of A is nonnegative. The problem of interest is to characterize all possible Jordan canonical forms (Jordan structures) of irreducible totally nonnegative matrices. We show that the positive eigenvalues of such matrices have algebraic multiplicity one, and also demonstrate key relationships between the number and sizes of the Jordan blocks corresponding to zero. These notions yield a complete description of all Jordan forms through $n = 7$, as well as numerous general results. We also define a notion of "principal rank" and employ this idea throughout.

Key words. totally nonnegative matrices, spectral structure, irreducible, principal rank, Jordan blocks, Jordan forms and structures

AMS subject classifications. 15A18, 15A48

PII. S0895479800367014

1. Introduction.

An n-by-n matrix A is called *totally positive* (TP) (*totally nonnegative* (TN)) if *every* minor of A is positive (nonnegative). Such matrices arise in a variety of applications [11], have been studied most of the twentieth century, and have received increasing attention of late.

It has long been known [10, 1] that any TP matrix has only positive eigenvalues that are each of algebraic multiplicity one ("distinct positive eigenvalues"); these distinct positive eigenvalues may be any positive numbers [2]. One proof of the fact that any TP matrix has distinct positive eigenvalues offers a nice application of the so-called Perron–Frobenius theory of entrywise nonnegative matrices. If TP is relaxed to TN (which is the closure of TP [1]), then the eigenvalues are nonnegative, but any multiplicities, and in fact any Jordan structure (i.e., any Jordan canonical form), may occur (because a basic Jordan block associated with a nonnegative eigenvalue is TN and direct summation preserves TN). This naturally leaves a question about irreducible TN matrices. Are they spectrally more like TP matrices or more like general TN matrices? We say that an n-by-n ($n \geq 2$) matrix A is *reducible* if there exists a permutation matrix P so that

$$PAP^T = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix},$$

where the matrix 0 is an $(n-r)$-by-r zero matrix ($n-1 \geq r \geq 1$). Otherwise we say A is an *irreducible* matrix. A hint comes from prior work. A TN matrix is called *oscillatory* if some power of it is TP. It is known [10] that an invertible irreducible

*Received by the editors February 4, 2000; accepted for publication (in revised form) by R. Brualdi May 16, 2000; published electronically October 6, 2000.
http://www.siam.org/journals/simax/22-2/36701.html

†Department of Mathematics and Statistics, University of Regina, Regina, SK, Canada, S4S 0A2 (sfallat@math.uregina.ca). Most of this work has been taken from this author's Ph.D. dissertation while at the College of William and Mary. This research of this author is currently supported by an NSERC research grant.

‡Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-5683 (Michael.Gekhtman.1@nd.edu).

§Department of Mathematics, College of William and Mary, Williamsburg, VA 23187-8795 (crjohnso@math.wm.edu). The research of this author was supported in part by the Office of Naval Research contract N00014-90-J-1739 and NSF grant 92-00899.

TN matrix is oscillatory (the converse is also true). Since an oscillatory matrix must also have distinct positive eigenvalues (obviously), then an invertible irreducible TN matrix is spectrally like a TP matrix.

However, an irreducible TN matrix may easily be singular (in which case the eigenvalue zero occurs), and it is not immediately clear how significant the effect of this singularity may be. Our purpose, then, is to study the possible spectral structure of (singular) irreducible TN matrices. Interestingly, the effect of singularity is, in part, substantial. Our paper will follow two major lines. The eigenvalue zero may not only be multiple but may have very elaborate, but restricted, Jordan structure. On the other hand, the positive eigenvalues must still have multiplicity one. In the process of developing our results some necessary facts of possible independent interest are proven.

2. Preliminaries. For an m-by-n matrix $A = [a_{ij}]$, $\alpha \subseteq \{1, 2, \ldots, m\}$, and $\beta \subseteq \{1, 2, \ldots, n\}$, the submatrix of A lying in rows indexed by α and the columns indexed by β will be denoted by $A[\alpha|\beta]$. Similarly, $A(\alpha|\beta)$ is the submatrix obtained from A by deleting the rows indexed by α and the columns indexed by β. If A is an n-by-n matrix and $\alpha = \beta$, then the principal submatrix $A[\alpha|\alpha]$ is abbreviated to $A[\alpha]$, and the complementary principal submatrix is $A(\alpha)$. If $x = [x_i]$ is an n-vector, then we let $\mathrm{diag}(x_i)$ denote the n-by-n diagonal matrix with main diagonal entries x_i. Recall that the *rank* of a given m-by-n matrix A, denoted by $\mathrm{rank}(A)$, is the size of the largest invertible square submatrix of A. Naturally, the *principal rank* of an n-by-n matrix A, denoted by $\text{p-rank}(A)$, is the size of the largest invertible principal submatrix of A. Note that the inequality $0 \leq \text{p-rank}(A) \leq \mathrm{rank}(A) \leq \min(m, n)$ follows directly from the definitions above. One topic of interest is characterizing all the triples $(n, \mathrm{rank}(A), \text{p-rank}(A))$, where n is the size of a TN square matrix A.

It is not difficult to show that if A is a TN matrix (in fact, this result holds as long as A has nonnegative principal minors, i.e., A is a P_0-matrix), then $\text{p-rank}(A)$ is equal to the number of nonzero (or in this case positive) eigenvalues of A. Hence if A is an n-by-n TN matrix, then $n - \text{p-rank}(A)$ is equal to the sum of the sizes of the Jordan blocks corresponding to the eigenvalue zero (i.e., the algebraic multiplicity of the eigenvalue zero), and $n - \mathrm{rank}(A)$ is equal to the number of Jordan blocks corresponding to zero (i.e., the geometric multiplicity of the eigenvalue zero). Another well-known notion is how to determine the size of the largest Jordan block corresponding to the eigenvalue zero (in fact, it is known for any eigenvalue). In the case of irreducible TN matrices this reduces to the following: if k is the smallest positive integer such that $\mathrm{rank}(A^k) = \text{p-rank}(A)$, then k is equal to the size of the largest Jordan block corresponding to the eigenvalue zero. Observe that k, as defined above, satisfies $k \leq \mathrm{rank}(A) - \text{p-rank}(A) + 1$.

The next notion is needed for our reduction-type results in the next section and is useful for many problems dealing with TN matrices (see [6]). In the following definition and throughout this paper the symbol $*$ in a matrix means the corresponding entry is nonzero.

DEFINITION 2.1. *An m-by-n matrix A with no zero rows or columns is said to be* in double echelon form *if*

(i) *each row of A has one of the following forms:*
   (1) $(*, *, \cdots, *)$,
   (2) $(*, \cdots, *, 0, \cdots, 0)$,
   (3) $(0, \cdots, 0, *, \cdots, *)$, *or*
   (4) $(0, \cdots, 0, *, \cdots, *, 0, \cdots, 0)$;

(ii) *the first and last nonzero entries in row $i+1$ are not to the left of the first and last nonzero entries in row i, respectively ($i = 1, 2, \ldots, m-1$).*

Thus a matrix in double echelon form appears as follows:

$$\begin{bmatrix} * & * & 0 & \cdots & 0 \\ * & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & * \\ 0 & \cdots & 0 & * & * \end{bmatrix}.$$

It is not difficult to see that any TN matrix with no zero rows or columns must be in double echelon form (see also [17]). This implies the following result that was originally observed in [10], and can also be found in [17], stated slightly differently. We present a different proof here for completeness.

LEMMA 2.2. *Suppose $A = [a_{ij}]$ is an n-by-n TN matrix. Then A is irreducible if and only if $a_{ij} > 0$ for all i, j such that $|i - j| \leq 1$.*

Proof. The sufficiency of the condition that $a_{ij} > 0$ for all i, j with $|i - j| \leq 1$ is trivial. On the other hand, suppose A is irreducible. Thus A has no zero lines, and hence it must be in double echelon form. Suppose $a_{ii} = 0$ for some i. Then $a_{st} = 0$ for $1 \leq s \leq i$ and $i \leq t \leq n$, or $a_{st} = 0$ for $i \leq s \leq n$ and $1 \leq t \leq i$, from which it follows that A is reducible, and hence we have a contradiction. Therefore we may assume $a_{ii} > 0$ for all i. Similarly, if $a_{i,i+1} = 0$ for some $i = 1, 2, \ldots, n-1$, then $a_{st} = 0$ for all $1 \leq s \leq i$ and $i \leq t \leq n$. Again A is reducible, another contradiction. This completes the proof. ∎

We call an n-by-n matrix $A = [a_{ij}]$ *tridiagonal* if $a_{ij} = 0$ for all i, j with $|i-j| \geq 2$. Hence tridiagonal matrices and irreducible TN matrices are somewhat related (see Lemma 2.2 above). The following result is well known.

LEMMA 2.3. *Let T be an n-by-n irreducible entrywise nonnegative tridiagonal matrix. Then the eigenvalues of T are real and distinct.*

Before we come to our reduction-type results we wish to make some brief remarks concerning row operations involving TN matrices. For simplicity of notation (see also [4]), we let $G_2 = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 1 \end{smallmatrix}\right]$,and let $F_r(c)$, $c \geq 0$ denote the r-by-r matrix,

$$F_r(c) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0_{r-2} & 0 \\ c & 0 & 1 \end{bmatrix}.$$

Observe that both of the above matrices are TN. We also note that the property TN is not in general preserved under row and column operations. However (see, for example, [19]), if $A = [a_{ij}]$ is TN and say $a_{1j},\ a_{1,j+1} > 0$, and $a_{1k} = 0$ for $k > j+1$, then the matrix A^* obtained from A by subtracting $a_{1,j+1}/a_{1j}$ times column j from column $j+1$ is TN. That is, $A = A^*U$, where U is the upper triangular TN matrix given by $U = I_{j-1} \oplus F_2^T(a_{1,j+1}/a_{1j}) \oplus I_{n-j-1}$. This fact was generalized in [4]. Suppose $A = [a_{ij}]$ is TN, $a_{1j} > 0$, and for some $t > j$, $a_{1t} > 0$, but $a_{1k} = 0$ for all $j < k < t$ and $k > t$. Then, since A is TN it follows that $a_{ik} = 0$ for $1 \leq i \leq n$ and $j < k < t$. Let A^* be obtained from A by subtracting a_{1t}/a_{1j} times column j from column t. Thus $A = A^*U$, where U is the upper triangular TN matrix given by $U = I_{j-1} \oplus F_{t-j+1}^T(a_{1t}/a_{1j}) \oplus I_{n-t}$. It is shown in [4] that the matrix A^* is TN.

Consider the following examples which demonstrate that there do exist nontrivial (larger than one) Jordan blocks corresponding to zero for irreducible TN matrices.

Example 2.4. Let

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}.$$

Then A is a 3-by-3 TP matrix.

Consider the 4-by-4 irreducible TN matrix,

$$B = \left[\begin{array}{c|ccc} 3 & 3 & 2 & 1 \\ 2 & 2 & 3 & 2 \\ 1 & 1 & 2 & 3 \\ \hline 1 & 1 & 2 & 3 \end{array}\right].$$

Then $\text{rank}(B) = \text{rank}(A) = 3$ and $\text{p-rank}(B) = \text{p-rank}(A) - 1 = 2$, from which it follows that B has one 2-by-2 Jordan block corresponding to zero and two distinct positive eigenvalues.

We note here that in general this "asymmetric" bordering of a TN matrix preserves the rank but may change the principal rank. Observe that if we border the matrix B above in a similar manner, then the resulting TN matrix has the same rank and principal rank as B. We will come back to this asymmetric bordering later.

Example 2.5. A matrix $B = [b_{ij}]$ is said to be a *lower Hessenberg matrix* if $b_{ij} = 0$ for all i, j with $i + 1 < j$. Consider the n-by-n lower Hessenberg (0,1)-matrix

$$H = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}.$$

It is not difficult to verify that H is an irreducible TN matrix with rank $n-1$. (Observe that H is singular and that $H[\{1, 2, \ldots, n-1\}|\{2, 3, \ldots, n\}]$ is a nonsingular lower triangular matrix.) A more challenging exercise is to prove that $\text{p-rank}(H) = \lceil \frac{n}{2} \rceil$. To see this observe that if n is odd (even), then $H[\{1, 3, 5, \ldots, n\}]$ ($H[\{2, 4, 6, \ldots, n\}]$) is a nonsingular lower triangular matrix. Hence $\text{p-rank}(H) \geq \lceil \frac{n}{2} \rceil$. To show $\text{p-rank}(H) \leq \lceil \frac{n}{2} \rceil$, suppose there exists an index set α such that $|\alpha| > \lceil \frac{n}{2} \rceil$ and $\det H[\alpha] > 0$. Then α must contain at least one consecutive pair of indices, say i and $i+1$ are in α, where $1 < i < n - 1$. Since H is lower Hessenberg and $\det H[\alpha] > 0$, it follows that index $i + 2 \in \alpha$. Applying the same reasoning to the pair $i + 1$ and $i + 2$, we may conclude $i + 3 \in \alpha$. However, continuing in this manner will show that $H[\alpha]$ is singular, since either both indices $n - 1$ and n will be in α, or the maximum of α is less than n, in which case there will exist a pair of indices $k, k + 1$ in α and $k + 2$ not in α. Thus $\text{p-rank}(H) = \lceil \frac{n}{2} \rceil$. Finally, we conclude this section with the following definition.

DEFINITION 2.6. *Let A and B be two square matrices, not necessarily of the same size. Then we say that A and B have the* same nonzero Jordan structure *if the distinct nonzero eigenvalues of A and B can be put into* 1-1 *correspondence so that each corresponding pair has the same number and sizes of Jordan blocks. Further, if A and B are the same size, we say that A and B have the same* qualitative Jordan structure*, if they have the same nonzero Jordan structure and if the number and sizes of the Jordan blocks corresponding to zero coincide.*

For example, suppose A has eigenvalues two (with one 2-by-2 Jordan block) and three (with one 3-by-3 Jordan block). Then any 5-by-5 matrix with two distinct nonzero eigenvalues, one of which has a 2-by-2 Jordan block and the other with a 3-by-3 Jordan block, has the same nonzero and qualitative Jordan structure as A. Recall that if A is an n-by-m matrix and B is a m-by-n matrix, then AB and BA have the same nonzero Jordan structure [13, Thm. 1.3.20], and two matrices that are similar have the same qualitative Jordan structure (since they both have the same Jordan canonical form [13]).

3. Positive eigenvalues of TN matrices. As noted earlier, the nonzero eigenvalues of a TN matrix are positive. However, more can be said about other properties that they might possess. We note here that for the matrix B in Example 2.4 the two eigenvalues are distinct. In fact, as we shall see, the positive eigenvalues of an irreducible TN matrix are always distinct.

We begin our analysis with a basic lemma, from which the main result may be obtained by a sequential application of this lemma. In [18] a similar "reduction"-type basic lemma was used to prove the following result. If A is an n-by-n nonsingular TN matrix, then there exists a nonsingular TN matrix S and a tridiagonal TN matrix T such that $TS = SA$, and the matrices A and T have the same eigenvalues. Later Cryer [4] extended this result to general (singular and nonsingular) TN matrices. Since the auxiliary assumption of irreducibility is necessary for our analysis (and was not in [18]), we are required to prove a different reduction-type lemma that can be stated as follows.

LEMMA 3.1 (basic lemma). *Suppose that $A = [a_{ij}]$ is an n-by-n irreducible TN matrix such that for some fixed $i < j < n$, $a_{lm} = 0$ for all $l < i$, $m \geq j$, $a_{i,j+1} > 0$, and $a_{it} = 0$ for all $t > j+1$. Then there exists an irreducible TN matrix A' such that*

(i) *$(A')_{lm} = 0$ for all $l < i$, $m \geq j$, $(A')_{i,j+1} = 0$, and*

(ii) *A' is either n-by-n and similar to A or is $(n-1)$-by-$(n-1)$ and has the same nonzero Jordan structure as A.*

Proof. By our assumptions, $a_{lj} = a_{l,j+1} = 0$ for $l < i$. Also, since A is irreducible and $a_{i,j+1} > 0$, a_{ij} is also positive. Use column j to eliminate $a_{i,j+1}$ via the elementary upper triangular bidiagonal nonsingular matrix $S = I - \alpha e_j e_{j+1}^T$. Consider the matrix $A' = S^{-1}AS$. It is shown in [19] that AS is TN, and since S^{-1} is TN, we have $A' = S^{-1}AS$ is TN. Clearly the $(i, j+1)$ entry of A' is zero. Observe that A' will be in double echelon form, unless A' contains a zero column, which necessarily must be column $j+1$. Assume for now that column $j+1$ of A' is nonzero. Then we must show that A' is irreducible. Note that, by construction, and the irreducibility of A, $(A')_{k,k+1} = a_{k,k+1} > 0$ for $k \neq j$, $(A')_{k,k-1} = a_{k,k-1} > 0$ for $k \neq j, j+1$, and $(A')_{j,j-1} = a_{j,j-1} + \alpha a_{j+1,j-1} > 0$ for $k \neq j$. Thus, by Lemma 2.2, we only need to show that $(A')_{j,j+1} > 0$ and $(A')_{j+2,j+1} > 0$. Since $(A')_{j,j+2} = a_{j,j+2} + \alpha a_{j+1,j+2}$ is positive, so is $(A')_{j,j+1}$ (recall that A' is in double echelon form). Now consider $(A')_{j+2,j+1} = a_{j+2,j+1} - \alpha a_{j+2,j}$. Then either $a_{j+2,j} = 0$, and therefore $(A')_{j+2,j+1} > 0$, or both $(A')_{j+2,j} = a_{j+2,j}$ and $(A')_{j+2,j+2} = a_{j+2,j+2}$ are positive, from which the positivity of $(A')_{j+2,j+1}$ follows from the double echelon form of A'. Thus, A' is irreducible. Finally, suppose the $(j+1)$th column of A' is zero. Then (as in [18] and [4]) consider the matrix obtained by deleting the $(j+1)$th row and column of A', and denote it by A''. It is not hard to see that A' is similar to a matrix $\left[\begin{smallmatrix} A'' & 0 \\ * & 0 \end{smallmatrix}\right]$. Therefore, the nonzero Jordan structure of A'' is the same as A', which in turn is the same as A. Moreover, since A'' is a submatrix of A', A'' is TN. The only point remaining to prove is that A'' is irreducible. To this end, it suffices to show that $(A')_{j+2,j}$ and $(A')_{j,j+2}$

are positive. But, $(A')_{j,j+2} = a_{j,j+2} + \alpha a_{j+1,j+2} > 0$, as A is irreducible. For the same reason, since $(A')_{j+2,j+1} = a_{j+2,j+1} - \alpha a_{j+2,j} = 0$, we have $(A')_{j+2,j} = a_{j+2,j} > 0$ and the proof is complete. ∎

We are now in a position to state our main results concerning the eigenvalues of an irreducible TN matrix.

THEOREM 3.2. *Let A be an irreducible TN matrix. Then there exists an irreducible tridiagonal TN matrix T (not necessarily of the same size as A) with the same nonzero Jordan structure as A. Moreover, T is obtained from A by a sequence of similarity transformations and projections.*

Proof. Successive application of the basic lemma (Lemma 3.1) results in a k-by-k ($k \leq n$) irreducible lower Hessenberg TN matrix L, which has the same nonzero Jordan structure as A. Consider the matrix $U = L^T$. Clearly, U is upper Hessenberg. Since this property is preserved under similarity transformations by upper triangular matrices, if a zero column is produced via application of the procedure described in the proof of Lemma 3.1, it must be the last column. In this case, deleting the last column and row then produces an upper Hessenberg matrix of smaller dimension with the same nonzero Jordan structure as U. Applying Lemma 3.2 repeatedly, we finally obtain an irreducible TN tridiagonal matrix T^T with the same nonzero Jordan structure as A. Then, clearly, T satisfies all the conditions of the theorem. ∎

THEOREM 3.3. *Let A be an n-by-n irreducible TN matrix. Then the positive eigenvalues of A are distinct.*

Proof. By the previous theorem there exists an irreducible TN tridiagonal matrix T, with the same nonzero Jordan structure as A. By Lemma 2.3 the positive eigenvalues of T are distinct, hence the positive eigenvalues of A are distinct. ∎

COROLLARY 3.4 (see [10]). *The eigenvalues of a TP matrix are real positive and distinct.*

We note here that the size of the tridiagonal matrix obtained in Theorem 3.2 is either the same as the number of nonzero eigenvalues of A or is this number plus one. In the next section we will see that this quantity (namely, the number of nonzero eigenvalues of A) will play a central role in our analysis of the qualitative Jordan structures of TN matrices.

4. Jordan structures of TN matrices. There has been a considerable amount of work accomplished on factorizations of TN matrices particularly into upper and lower elementary bidiagonal matrices (see [4, 12]). By definition, an *elementary bidiagonal matrix* is an n-by-n matrix whose main diagonal entries are all equal to one, and there is at most one nonzero off-diagonal entry and this entry must occur on the super- or subdiagonal. To this end, we denote by $E_k(\mu) = [c_{ij}]$ ($2 \leq k \leq n$) the lower elementary bidiagonal matrix whose elements are given by

$$c_{ij} = \begin{cases} 1 & \text{if } i = j, \\ \mu & \text{if } i = k,\ j = k-1, \text{ that is, } E_k(\mu) = \\ 0 & \text{otherwise,} \end{cases} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \mu & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}.$$

The next result may be found in many places (see, for example, [4] where a bidiagonal factorization is proven for general (i.e., singular and nonsingular) TN matrices, or see [19] for a statement of the key lemma used to prove the next theorem), but see [12] for the version stated here.

THEOREM 4.1. *Let A be an n-by-n nonsingular TN matrix. Then A can be written as*

$$\begin{aligned} A = &(E_2(l_k))(E_3(l_{k-1})E_2(l_{k-2}))\cdots(E_n(l_{n-1})\cdots E_3(l_2)E_2(l_1))D \\ &(E_2^T(u_1)E_3^T(u_2)\cdots E_n^T(u_{n-1}))\cdots(E_2^T(u_{k-2})E_3^T(u_{k-1}))(E_2^T(u_k)), \end{aligned} \tag{4.1}$$

where $k = \binom{n}{2}$, $l_i, u_j \geq 0$ for all $i, j \in \{1, 2, \ldots, k\}$, and D is a positive diagonal matrix.

An excellent treatment of the combinatorial and algebraic aspects of bidiagonal factorizations of TN matrices along with generalizations for TP elements is reductive Lie groups as given in [3] and [7]. One of the main tools used in these papers is a graphical representation of the bidiagonal factorization in terms of planar diagrams that can be described as follows.

An n-by-n diagonal matrix $\operatorname{diag}(d_1, d_2, \ldots, d_n)$ is represented by the diagram in Figure 4.1. Associated with an elementary lower bidiagonal matrix $E_k(l)$ is the diagram in Figure 4.2, while an elementary upper bidiagonal matrix $E_j^T(u)$ is represented by the diagram in Figure 4.3.

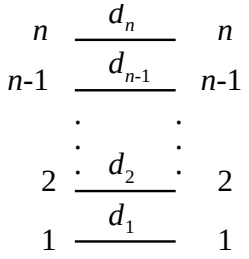


FIG. 4.1. *Diagram for a diagonal matrix.*

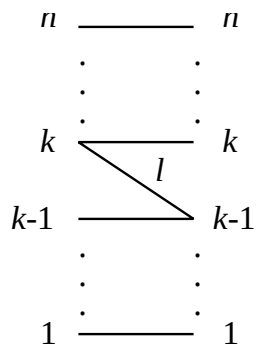


FIG. 4.2. *Diagram for a lower bidiagonal matrix.*

Each horizontal edge in Figures 4.2 and 4.3 has a weight of 1. It is not hard to verify that if A is a matrix represented by any of the diagrams in Figures 4.1, 4.2, and 4.3, then $\det A[\{i_1, i_2, \ldots, i_k\}|\{j_1, j_2, \ldots, j_k\}]$ is nonzero if and only if in the corresponding diagram there is a family of k vertex-disjoint paths joining the vertices $\{i_1, i_2, \ldots, i_k\}$ on the left side of the diagram with the vertices $\{j_1, j_2, \ldots, j_k\}$ on the right side. Moreover, in this case this family of paths is unique and $\det A[\{i_1, i_2, \ldots, i_k\}|\{j_1, j_2, \ldots, j_k\}]$ is equal to the product of all the weights assigned to the edges that form this family.

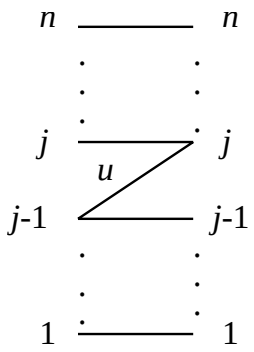


FIG. 4.3. *Diagram for a upper bidiagonal matrix.*

Now, given a product $A = A_1 A_2 \cdots A_k$ in which each matrix A_i is either a diagonal matrix or an elementary (upper or lower) bidiagonal matrix, a corresponding diagram is obtained by concatenation left to right of the diagrams associated with the matrices $A_1, A_2, \ldots, A_k$. For a given collection of vertex-disjoint paths joining the vertices $\{i_1, i_2, \ldots, i_k\}$ on the left side of the diagram with the vertices $\{j_1, j_2, \ldots, j_k\}$ on the right side we define the weight of such a collection to be the product of all the weights assigned to the edges of the paths that form this collection. Then, by the Cauchy–Binet identity (see [13]), the $\det A[\{i_1, i_2, \ldots, i_k\}|\{j_1, j_2, \ldots, j_k\}]$ is equal to the sum of all the weights of all collections of vertex-disjoint paths joining the vertices $\{i_1, i_2, \ldots, i_k\}$ with the vertices $\{j_1, j_2, \ldots, j_k\}$.

As the most important example, consider the bidiagonal factorization of an arbitrary nonsingular TN matrix A from Theorem 4.1. This factorization translates into the following diagram (see Figure 4.4).

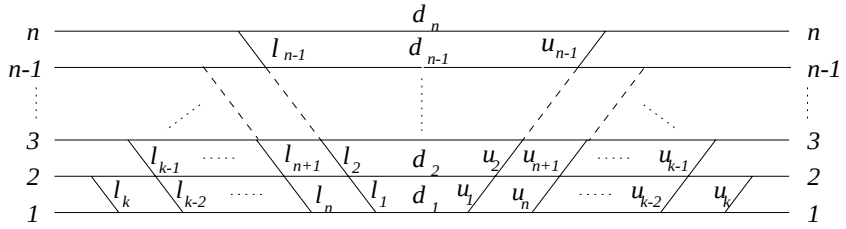


FIG. 4.4. *General n-by-n diagram.*

In the 3-by-3 case, for example, this diagram can be represented as in Figure 4.5.

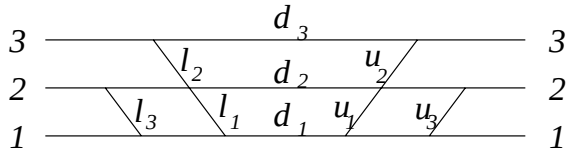


FIG. 4.5. *General 3-by-3 diagram.*

Then, for example, $\det A[\{2,3\}|\{1,2\}] = (l_2 d_2)(l_3 d_1)$, while the principal minor $\det A[\{2,3\}] = d_2 d_3 + (l_2 d_2 u_2)(l_3 d_1 u_3)$.

The fact that any TN matrix has a bidiagonal factorization (see [4]) can also

be employed to tackle certain problems concerning Jordan structures. In particular, the quantities rank and principal rank of a given matrix and irreducibility can all be interpreted via the diagram associated with a particular bidiagonal factorization of a TN matrix. For example, from a given diagram we can verify whether or not the associated TN matrix is irreducible by determining if there exists a path in this diagram from any index i to each of $i-1$, i, and $i+1$ (ignoring $i-1$ when $i=1$, and $i+1$ when $i=n$). If such a path exists, then for each i, $a_{i,i-1}$, a_{ii}, and $a_{i,i+1}$ are all positive, which implies by Lemma 2.2, that the associated TN matrix is irreducible. Since rank and principal rank are defined in terms of nonsingular submatrices, it follows that rank and principal rank can be interpreted as the largest collection of vertex disjoint paths beginning on the left and terminating on the right, in the diagram, and the largest collection of vertex disjoint paths which begin and terminate in the same index set, respectively.

We begin our analysis by considering the triple $(n, \text{rank}, \text{p-rank})$ among the class of irreducible TN matrices. First, observe that the triples $(n,1,1)$ and (n,n,n) certainly exist, for all $n \geq 1$, by considering the matrix J of all ones, and any n-by-n TP matrix, respectively. Thus for $n \leq 2$, we have completely characterized all possible triples. However, for $n=3$, the triples $(3,2,2)$ and $(3,2,1)$ have not been shown to be realizable. First, consider the triple $(3,2,2)$, and suppose that $A = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ is a 2-by-2 TP matrix. Then the matrix

$$B = \left[\begin{array}{cc|c} a & b & b \\ c & d & d \\ \hline c & d & d \end{array}\right]$$

is a 3-by-3 irreducible TN matrix with $\text{rank}(B) = \text{p-rank}(B) = 2$. Hence the triple $(3,2,2)$ is realizable. What about the triple $(3,2,1)$? Suppose there exists an irreducible TN matrix $A = [a_{ij}]$ with $\text{rank}(A) = 2$ and $\text{p-rank}(A) = 1$. Then $a_{ij} > 0$, for all i,j with $|i-j| \leq 1$. Observe that multiplying A by a positive diagonal matrix does not affect rank or principal rank. Hence we may assume that $a_{11} = a_{22} = a_{33} = 1$ and $a_{12} = a_{21}$, $a_{23} = a_{32}$. Thus

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{12} & 1 & a_{23} \\ a_{31} & a_{23} & 1 \end{bmatrix}.$$

Since $\text{p-rank}(A) = 1$, it follows that $a_{12} = a_{23} = 1$ and $a_{13}a_{31} = 1$. It is not difficult to determine, in this case, that if $a_{13}a_{31} = 1$, then $a_{13} = a_{31} = 1$, as A is TN. However, in this case $A = J$, which is a contradiction since $\text{rank}(A) = 2$. Thus the triple $(3,2,1)$ is not realizable. This leads us to our first result on these triples, with fixed principal rank. Recall that the equation $l_i = 0$ (or $u_j = 0$) means that the corresponding edge does not appear in a diagram.

PROPOSITION 4.2. *The triple $(n,k,1)$ is realizable by an n-by-n irreducible TN matrix if and only if $k = 1$.*

Proof. We have already seen that the triple $(n,1,1)$ is realizable for all n. Now assume the triple $(n,k,1)$ is realizable by an n-by-n irreducible TN matrix A. Then A has a bidiagonal factorization, which can be represented by an associated diagram. Since A is irreducible there exists a path from index n to n. Let P denote a shortest such path from n to n. Then we claim that P must intersect the bottom row of this diagram. If not, then since there always exists a path from index 1 to 1 by going along this bottom row, it follows that the principal rank of A would be at least 2, which

is a contradiction (see diagram on the left in Figure 4.6). Otherwise, P intersects the bottom row (see diagram on the right in Figure 4.6). Then since P is a shortest path, it follows that any maximal collection of vertex disjoint paths must intersect P. Hence the rank of A is at most one. Since $\{P\}$ is one such maximal collection the rank of A is one. □

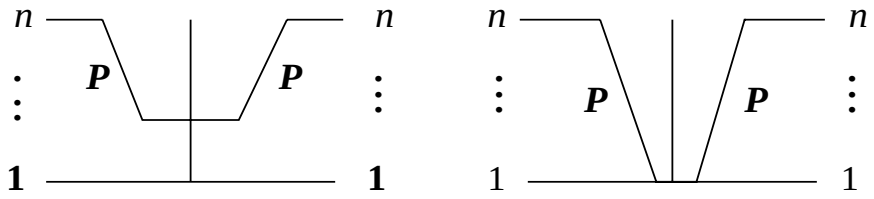


FIG. 4.6. *Fixed principal rank one.*

We note here that the above proposition could also have been proved in a similar manner as proving that the triple $(3, 2, 1)$ is not realizable.

Observe that if $\text{rank}(A) = \text{p-rank}(A)$, then A has $n - \text{p-rank}(A)$, 1-by-1 Jordan blocks corresponding to zero, and if $\text{rank}(A) = \text{p-rank}(A) + 1$, then A has exactly one 2-by-2 Jordan block and $n - \text{p-rank}(A) - 1$, 1-by-1 Jordan blocks corresponding to zero.

We now move on to the case when the principal rank is two.

PROPOSITION 4.3. *Suppose the triple $(n, k, 2)$ is realizable by an n-by-n irreducible TN matrix. Then $2 \leq k \leq \lceil \frac{n+1}{2} \rceil$. Moreover, each such k is realizable.*

Proof. First observe that $k \geq 2$ is obvious. As before we may assume that the given matrix has a bidiagonal factorization and an associated diagram. Choose a shortest path P from n to n. Since $\text{p-rank}(A) = 2$, this path cannot intersect the bottom row. Suppose the path P uses the edge with weight d_i. Since $\text{p-rank}(A) = 2$, it follows that any path beginning at $i-1$ which is disjoint from P must intersect the bottom row. Hence any maximal collection of vertex-disjoint paths from the indices $\{1, 2, \ldots, i-1\}$ that are disjoint from P contains at most one path. In this case it follows that $\text{rank}(A) = k \leq \max_{2 \leq i \leq n}\{\min(i, n - i + 2)\}$. Hence $k \leq \lceil \frac{n+1}{2} \rceil$. To show that every such triple $(n, k, 2)$ with $2 \leq k \leq \lceil \frac{n+1}{2} \rceil$ can be realized, consider the diagram in Figure 4.7 for $i = 2, 3, \ldots, n$. □

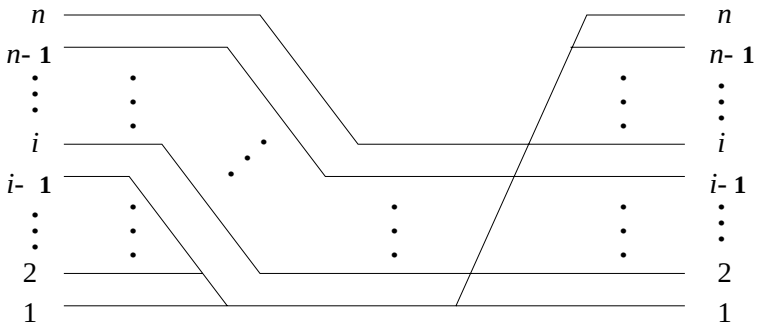


FIG. 4.7. *Fixed principal rank two.*

We can also apply similar techniques to prove the next result. Recall that the (0,1) lower Hessenberg matrix in Example 2.5 has rank equal to $n - 1$ and principal

rank equal to $\lceil \frac{n}{2} \rceil$. The next result shows that $\lceil \frac{n}{2} \rceil$ is the smallest possible value for principal rank in the case when rank is $n-1$.

PROPOSITION 4.4. *Suppose the triple $(n, n-1, k)$ $(n \geq 2)$ is realizable by an n-by-n irreducible TN matrix. Then $\lceil \frac{n}{2} \rceil \leq k \leq n-1$. Moreover, for each such k, the triple is realizable.*

Proof. First observe that the inequality $k \leq n-1$ is obvious. The proof is by induction on n. The claim has already been verified for $n \leq 3$. Let A be an n-by-n irreducible TN matrix with rank equal to $n-1$. As before we assume A has a bidiagonal factorization and an associated diagram, and suppose P denotes the shortest path from n to n. Then there are three cases to consider: (1) P uses the edge with weight d_i with $i \leq n-2$; (2) P uses the edge with weight d_{n-1}; or (3) P uses the edge with weight d_n.

Case 1: If P uses the edge with weight d_i with $i \leq n-2$, then it is not difficult to check that, since P was a shortest such path, the rank of A is at most $n-2$, a contradiction.

Case 2: If P uses the edge with weight d_{n-1}, then since A has rank $n-1$ the shortest path from each index j to itself $(1 \leq j \leq n-2)$ can drop at most one level. Moreover, the shortest path from index $n-2$ to itself cannot intersect P (otherwise $\operatorname{rank}(A) < n-1$). Consider the diagram induced by the vertices $\{1, 2, \ldots, n-2\}$. Observe that for this diagram the associated TN matrix A' (which is not a submatrix of A), satisfies $n-3 \leq \operatorname{rank}(A') \leq n-2$. If $\operatorname{rank}(A') = n-2$, then since the vertex disjoint paths that constitute $\operatorname{rank}(A')$ do not intersect P we have $\text{p-rank}(A) = n-1$. If $\operatorname{rank}(A') = n-3$, then by induction $\lceil \frac{n-2}{2} \rceil \leq \text{p-rank}(A') \leq n-3$, and every such value is achievable. Hence $\lceil \frac{n-2}{2} \rceil + 1 \leq \text{p-rank}(A) \leq n-3+1$ or $\lceil \frac{n}{2} \rceil \leq \text{p-rank}(A) \leq n-2$, and every such value is realizable.

Case 3: Suppose P uses the edge with weight d_n (i.e., P goes straight across the diagram). Consider the diagram induced by the indices $\{1, 2, \ldots, n-1\}$. By induction, the associated TN matrix A' is $(n-1)$-by-$(n-1)$ with $\operatorname{rank}(A') = n-2$; hence $\lceil \frac{n-1}{2} \rceil \leq \text{p-rank}(A') \leq n-2$, and every such value is achievable. Thus it follows that $\lceil \frac{n-1}{2} \rceil + 1 \leq \text{p-rank}(A) \leq n-1$, and every such value is achievable. This completes the proof. ∎

We are now in a position to classify all possible triples for $n = 4$.

Example 4.5. Suppose the triple $(4, k, p)$ is realizable by a 4-by-4 irreducible TN matrix. The triples (4,4,4) and (4,1,1) are certainly realizable. (By Proposition 4.2, the triple (4,1,1) is the only possible triple with principal rank equal to one.) If the rank (k) is fixed to be three, then, by Proposition 4.4, the only realizable triples are (4,3,3) and (4,3,2). Similarly, if the rank is fixed at two, then by Proposition 4.3 the triple (4,2,2) is the only realizable triple. Hence all possible triples have been determined.

We now move on to a more general result on the Jordan structure of TN matrices. Recall that if A is an n-by-n matrix and D is a positive diagonal matrix, then the Jordan structure of A and AD can be vastly different. (For example, it is known that if A is a matrix with positive principal minors, then there exists a positive diagonal matrix D so that the eigenvalues of AD are positive and distinct, even though A may not even be diagonalizable.) However, in the case of irreducible TN matrices it turns out that the Jordan structure of A and AD (D a positive diagonal matrix) coincide. We first state some necessary notation for compound matrices. Given an m-by-n matrix A we define the kth *compound of* A, which we denote by $C_k(A)$, to be the $\binom{m}{k}$-by-$\binom{n}{k}$ matrix whose (i,j)th entry is $\det A[\alpha_i | \beta_j]$, where α_i and β_j are k-subsets,

order lexicographically, of $\{1, 2, \ldots, m\}$ and $\{1, 2, \ldots, n\}$, respectively (see also [13]). We begin with the following lemma.

LEMMA 4.6. *Let A be an n-by-n TN matrix and suppose D is an n-by-n positive diagonal matrix. Then* $\operatorname{rank}((AD)^k) = \operatorname{rank}(A^k)$ *and* $\text{p-rank}((AD)^k) = \text{p-rank}(A^k)$, *where $k \geq 1$.*

Proof. Let $C_j(A)$ denote the jth compound of A. Since D is a positive diagonal matrix, it follows that $C_j(D) = D'$ is a positive diagonal matrix for all j. Hence $C_j(AD) = C_j(A)C_j(D) = C_j(A)D'$ where the first equality follows from the Cauchy–Binet identity for determinants. Since D' is a positive diagonal matrix the zero/nonzero patterns of $C_j(AD)$ and $C_j(A)$ are the same. Moreover, since $C_j(A)$ and $C_j(AD)$ are entrywise nonnegative matrices and $C_j(A^k) = (C_j(A))^k$, it follows that the zero/nonzero pattern of each $C_j(A^k)$ is completely determined by $C_j(A)$. Since the zero/nonzero patterns of $C_j(AD)$ and $C_j(A)$ are the same, it follows that the zero/nonzero patterns of $C_j(A^k)$ and $C_j((AD)^k)$ are the same. Observe that the rank and the principal rank of a given matrix are given by the largest j, such that jth compound is nonzero, and the largest j, such that the jth compound has a nonzero diagonal, respectively. Hence it follows that $\operatorname{rank}((AD)^k) = \operatorname{rank}(A^k)$ and $\text{p-rank}((AD)^k) = \text{p-rank}(A^k)$, where $k \geq 1$. This completes the proof. ∎

We are now in a position to prove that the Jordan structure of A and AD are the same, whenever A is TN and irreducible.

THEOREM 4.7. *Suppose A is an n-by-n irreducible TN matrix and D is a positive diagonal matrix. Then A and AD have the same qualitative Jordan structure.*

Proof. Since A is irreducible (and hence AD is) and since $\text{p-rank}(AD) = \text{p-rank}(A)$, we have that the number of distinct positive eigenvalues of A and AD are equal. Moreover, since the number and sizes of the Jordan blocks corresponding to zero are completely determined by the ranks of powers, it follows that A and AD have the same qualitative Jordan structure, since $\operatorname{rank}((AD)^k) = \operatorname{rank}(A^k)$, for $k \geq 1$ (by Lemma 4.6). ∎

The assumption of irreducibility in the above result is necessary as seen by the following example. Let

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

Then A is TN and is itself a Jordan block, and hence is not diagonalizable. However, if $D = \operatorname{diag}(1, 2, 3)$, then

$$AD = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix},$$

which has distinct eigenvalues and hence is diagonalizable. Thus A and AD do not have the same qualitative Jordan structure.

We now present of couple of interesting consequences to the above theorem.

COROLLARY 4.8. *Suppose A is an n-by-n irreducible TN matrix partitioned as follows: $A = \left[\begin{smallmatrix} A_{11} & a_{12} \\ a_{21} & a_{22} \end{smallmatrix}\right]$, where A_{11} is $(n-1)$-by-$(n-1)$, and a_{22} is a scalar. Define the $(n+1)$-by-$(n+1)$ matrix B as follows:*

$$B = \left[\begin{array}{cc|c} A_{11} & a_{12} & a_{12} \\ a_{21} & a_{22} & a_{22} \\ \hline a_{21} & a_{22} & a_{22} \end{array}\right].$$

Then B is an irreducible TN matrix with $\operatorname{rank}(B) = \operatorname{rank}(A)$ *and* $\text{p-rank}(B) = \text{p-rank}(A)$*, and the Jordan structure of B is the same as A, except B has one more* 1*-by-*1 *Jordan block associated with the eigenvalue zero.*

Proof. The fact that B is TN is trivial, and since $a_{22} > 0$ (because A is irreducible), B is irreducible. Also by the symmetry of the bordering scheme, it follows that $\operatorname{rank}(B) = \operatorname{rank}(A)$ and $\text{p-rank}(B) = \text{p-rank}(A)$. Let $S = E_n(-1)$, the n-by-n elementary bidiagonal matrix with a -1 in the $(n, n-1)$ entry. Then an easy calculation reveals that

$$SBS^{-1} = \begin{bmatrix} A_{11} & 2a_{12} & a_{12} \\ a_{21} & 2a_{22} & a_{22} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} AD & \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} \\ 0 & 0 \end{bmatrix},$$

where $D = I \oplus [2]$. Observe that $\operatorname{rank}(B^k) = \operatorname{rank}(SB^kS^{-1}) = \operatorname{rank}((SBS^{-1})^k)$. Since

$$(SBS^{-1})^k = \begin{bmatrix} (AD)^k & (AD)^{k-1}\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} \\ 0 & 0 \end{bmatrix},$$

and $\left[\begin{smallmatrix} a_{12} \\ a_{22} \end{smallmatrix}\right]$ is in the span of AD, it follows that

$$\operatorname{rank}\left(\begin{bmatrix} (AD)^k & (AD)^{k-1}\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} \\ 0 & 0 \end{bmatrix}\right) = \operatorname{rank}((AD)^k).$$

By Theorem 4.7, we have $\operatorname{rank}(B^k) = \operatorname{rank}((AD)^k) = \operatorname{rank}(A^k)$. The result now follows easily. ∎

COROLLARY 4.9. *If the triple (n, k, p) is realizable by an irreducible TN matrix, then the triple $(n+1, k, p)$ is also realizable by an irreducible TN matrix.*

Using the above results we can now classify all possible triples for $n = 5$ and 6.

Example 4.10. Suppose the triple $(5, k, p)$ is realizable by a 5-by-5 irreducible TN matrix. The triples (5,5,5) and (5,1,1) are certainly realizable. (By Proposition 4.2, the triple (5,1,1) is the only possible triple with a principal rank equal one realization.) If the rank (k) is fixed to be four, then, by Proposition 4.4, the only realizable triples are (5,4,4) and (5,4,3). Similarly, if the rank is fixed at two, then by Proposition 4.3 the triple (5,2,2) is the only realizable triple. Finally, suppose the rank is fixed to be three. Then there are only two possible values for the principal rank: two or three. Recall from Example 4.5 that the triples (4,3,2) and (4,3,3) were both realizable. Hence by Corollary 4.9, the triples (5,3,2) and (5,3,3) are both realizable.

For the case $n = 6$, the arguments are much the same as above and are omitted here. Following is the list of all the triples that are realizable by 6-by-6 irreducible TN matrices: (6,6,6); (6,5,5), (6,5,4), (6,5,3); (6,4,4), (6,4,3), (6,4,2); (6,3,3), (6,3,2); (6,2,2); (6,1,1). Hence all possible triples for $n = 5$ and 6 have been determined.

We now turn our attention to proving some general results on the triples, $(n, \text{rank}, \text{p} - \text{rank})$.

PROPOSITION 4.11. *For $n \geq 1$ and $r \leq n$, the triple (n, r, r) is realizable by an irreducible n-by-n TN matrix.*

Proof. Let A be an r-by-r TP matrix. Then the triple (r, r, r) is realizable by an irreducible TN matrix (namely, A). Hence, by Corollary 4.9, the triple (n, r, r) is realizable, since $r \leq n$. ∎

Recall that any matrix with the triple (n, r, r), has $n - r$, 1-by-1 Jordan blocks corresponding to zero.

The asymmetric bordering notion used in Example 2.4 may also be used to prove the existence of a general class of triples.

PROPOSITION 4.12. *For $r \geq 3$ and $r < n$, the triple $(n, r, r-1)$ is realizable by an irreducible n-by-n TN matrix.*

Proof. We first prove the following claim: The triple $(r+1, r, r-1)$ is realizable for $r \geq 3$. To prove this claim let A be an r-by-r TP matrix partitioned as follows:

$$A = \begin{bmatrix} a_{11} & A_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

where A_{12} is $(r-1)$-by-$(r-1)$. Define the $(r+1)$-by-$(r+1)$ matrix A' as

$$A' = \left[\begin{array}{c|cc} a_{11} & a_{11} & A_{12} \\ a_{21} & a_{21} & a_{22} \\ \hline a_{21} & a_{21} & a_{22} \end{array}\right].$$

Then A' is an irreducible TN matrix, and it is clear that $\operatorname{rank}(A') = \operatorname{rank}(A) = r$. What about the p-rank$(A')$? First observe that

$$\det A'[\{2, 3, \ldots, r\}] = \det A[\{2, 3, \ldots, r\} | \{1, 2, \ldots, r-1\}] > 0,$$

since A is TP. Thus $r \geq$ p-rank$(A') \geq r-1$. Suppose p-rank$(A') = r$, and let $\alpha \subseteq \{1, 2, \ldots, r+1\}$, with $|\alpha| = r$ and $\det A[\alpha] > 0$. There are two cases to consider: $1 \in \alpha$, or $1 \notin \alpha$. Suppose $1 \in \alpha$. Then $2 \notin \alpha$ since $\det A[\alpha] > 0$, so $\alpha = \{1, 3, 4 \ldots, r+1\}$. But since rows r and $r+1$ of A' are the same, it follows that $\det A[\alpha] = 0$, which is a contradiction. Otherwise, suppose $1 \notin \alpha$. Then $\alpha = \{2, 3, \ldots, r+1\}$, and again $\det A[\alpha] = 0$, which is also a contradiction. Hence p-rank$(A') = r-1$. Thus the triple $(r+1, r, r-1)$ is realizable. Then, by Corollary 4.9, the triple $(n, r, r-1)$ is realizable for all $n > r$. This completes the proof. ∎

We note that the requirement that $r \geq 3$ is necessary since if the principal rank is equal to one, then the rank is necessarily equal to one. Similarly, $r < n$ is also necessary. Recall that any matrix with the triple $(n, r, r-1)$ has one 2-by-2 and $n-r-1$ 1-by-1 Jordan blocks corresponding to zero. We now consider a more general result whose proof follows slightly the proof of the previous result.

THEOREM 4.13. *For $k \geq 0$, $r \geq k+2$, and $n \geq r+k$, the triple $(n, r, r-k)$ is realizable by an irreducible n-by-n TN matrix.*

Proof. We first prove that the triple $(r+k, r, r-k)$ is realizable, from which the general result will follow by Corollary 4.9. Let A be an r-by-r TP matrix. Let $A^{(k)}$ be the $(r+k)$-by-$(r+k)$ irreducible TN matrix obtained from A by k successive applications of the asymmetric bordering scheme used in the proof of the previous result. Then $\operatorname{rank}(A^{(k)}) = r$. Moreover, $\det A^{(k)}[\{k+1, \ldots, r\}] = \det A[\{k+1, \ldots, r\} | \{1, 2, \ldots, r-k\}] > 0$, since A is TP. Hence $r \geq$ p-rank$(A^{(k)}) \geq r-k$. Finally, suppose $\alpha \subseteq \{1, 2, \ldots, r+k\}$ with $\det(A^{(k)}[\alpha]) > 0$. Then by the construction of $A^{(k)}$ it follows that α can contain at most one index from $\{1, 2, \ldots, k+1\}$ and at most one index from $\{r, r+1, \ldots, r+k\}$. In other words $|\alpha| \leq |\{k+2, \ldots, r-1\}| + 2 = r-k$. Hence p-rank$(A^{(k)}) = r-k$. This completes the proof. ∎

It is worth mentioning that while this asymmetric bordering scheme has proved useful for determining certain triples, it is not clear how this bordering scheme affects the Jordan structure of an irreducible TN matrix. We are currently making progress on this issue.

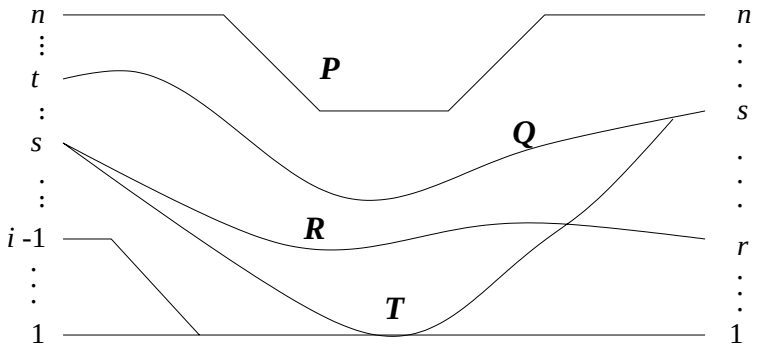


FIG. 4.8. *Principal rank two.*

We are now in a position to prove the next result for the case when the principal rank is fixed to be two.

THEOREM 4.14. *Let A be an irreducible TN matrix with principal rank equal to two. Then the size of the largest Jordan block corresponding to zero is at most two.*

Proof. As before we assume that A has a bidiagonal factorization and an associated diagram. Let P be a shortest path from index n on the left to index n on the right. Since p-rank$(A) = 2$, it follows that this path P does not intersect the bottom of this diagram. Suppose P drops to level i, that is, does not use any edges of the diagram induced by the set $\{1, 2, \ldots, i-1\}$, then (as in the case of Proposition 4.3) any path from any index $2, 3, \ldots, i-1$, disjoint from P, must intersect the bottom row, otherwise p-rank$(A) > 2$. To show that the size of the largest Jordan block is at most two we will show rank$(A^2) = 2$. To prove this it is enough to show that any path from any of the indices $\{i, i+1, \ldots, n-1\}$ to $\{1, 2, \ldots, n\}$ must either intersect P, or the bottom row, or terminate among the indices $\{1, 2, \ldots, i-1\}$. Suppose there exists a path Q originating at some index $t \in \{i, i+1, \ldots, n-1\}$ and terminating at $s \in \{i, i+1, \ldots, n-1\}$ without intersecting P or the bottom row. Since Q does not intersect P it must drop below level i, as P was a shortest path. Assume $t \geq s$ (the argument for $s \leq t$ is similar). Since A is irreducible there exists a path from s to $s+1$, but in this case such a path must intersect Q. We claim that any path from s, disjoint from P, must intersect the bottom level. To see this suppose there exists a path R from s that does not intersect the bottom level (see also Figure 4.8). Recall that any path T from s to s, disjoint from P, must intersect the bottom level (since p-rank$(A) = 2$), and hence any such path must intersect R. Thus there exists a path from s to s that does not intersect P and is disjoint from the bottom level: Take R until the intersection of R and T, then follow T until it intersects Q (which may be at s), and then proceed to s. This contradicts that fact that p-rank$(A) = 2$. Therefore any path originating from $\{i, i+1, \ldots, n-1\}$ must satisfy one of the following: (1) intersects P; (2) intersects the bottom row; (3) terminates in $\{1, 2, \ldots, i-1\}$; or (4) if it terminates at $s \geq i$, then any path beginning at s that is disjoint from P must intersect the bottom row. (We note that these cases are not mutually exclusive.) It now follows that the rank of A^2 is two. Certainly, the rank$(A^2) \geq 2$, as p-rank$(A) = 2$. Suppose there exists at least three vertex disjoint paths constituting the rank of A^2. Since P was chosen to be a shortest such path, at most one of these paths can intersect P. Moreover, since these paths are vertex-disjoint, at most one can terminate among the vertices $\{1, 2, \ldots, i-1\}$ (which also includes that case of a

path intersecting the bottom level). Thus the only possibility left is case (4). But in this case, any path beginning from s that is disjoint from P must intersect the bottom level. Hence these paths cannot be disjoint for the diagram representing A^2 (which is obtained simply by concatenating two diagrams associated with A). This completes the proof. □

COROLLARY 4.15. *Let A be an n-by-n irreducible TN matrix with* p-rank$(A) = 2$. *Then* rank$(A^2) = 2$.

Through $n = 6$ it is not difficult to show that given a complete description of all the triples $(n, \text{rank, p-rank})$, and using Theorem 4.14, we can completely characterize all possible Jordan structures (or all possible Jordan canonical forms) for every n-by-n irreducible TN matrix with $n \leq 6$. For example, when $n = 6$ and rank is fixed at four, the triples (6,4,4), (6,4,3), and (6,4,2) are the only realizable triples. Since $n - \text{rank}(A) = 2$, we know that there must exist two Jordan blocks corresponding to zero. By the remarks following Propositions 4.11 and 4.12, it follows that for (6,4,4) there are two 1-by-1 Jordan blocks corresponding to zero, and for (6,4,3) there is one 2-by-2 and one 1-by-1 Jordan block corresponding to zero. For the case (6,4,2) there are two possible Jordan structures: (1) one 3-by-3 and one 1-by-1 Jordan block corresponding to zero, or (2) two 2-by-2 Jordan blocks corresponding to zero. By Theorem 4.14 it follows that case (1) cannot occur, hence any 6-by-6 irreducible TN matrix with rank equal to four and principal rank equal to two (which do exist) must have two 2-by-2 Jordan blocks corresponding to zero.

In the following list we use JB to mean Jordan block corresponding to zero. Also in this list we do not include the cases when rank$(A) = 1$ or n. This list represents a complete classification of all possible Jordan structures through $n = 6$:

(1) $n = 3$:
  (a) rank$(A) = 2$
    (i) p-rank$(A) = 2 \Rightarrow$ one 1-by-1 JB;
(2) $n = 4$:
  (a) rank$(A) = 3$
    (i) p-rank$(A) = 3 \Rightarrow$ one 1-by-1 JB;
    (ii) p-rank$(A) = 2 \Rightarrow$ one 2-by-2 JB;
  (b) rank$(A) = 2$
    (i) p-rank$(A) = 2 \Rightarrow$ two 1-by-1 JBs;
(3) $n = 5$:
  (a) rank$(A) = 4$
    (i) p-rank$(A) = 4 \Rightarrow$ one 1-by-1 JB;
    (ii) p-rank$(A) = 3 \Rightarrow$ one 2-by-2 JB;
  (b) rank$(A) = 3$
    (i) p-rank$(A) = 3 \Rightarrow$ two 1-by-1 JBs;
    (ii) p-rank$(A) = 2 \Rightarrow$ one 2-by-2 JB and one 1-by-1 JB;
  (c) rank$(A) = 2$
    (i) p-rank$(A) = 2 \Rightarrow$ three 1-by-1 JBs;
(4) $n = 6$:
  (a) rank$(A) = 5$
    (i) p-rank$(A) = 5 \Rightarrow$ one 1-by-1 JB;
    (ii) p-rank$(A) = 4 \Rightarrow$ one 2-by-2 JB;
    (iii) p-rank$(A) = 3 \Rightarrow$ one 3-by-3 JB;
  (b) rank$(A) = 4$
    (i) p-rank$(A) = 4 \Rightarrow$ two 1-by-1 JBs;

(ii) p-rank$(A) = 3 \Rightarrow$ one 2-by-2 JB and one 1-by-1 JB;
(iii) p-rank$(A) = 2 \Rightarrow$ two 2-by-2 JBs;
(c) rank$(A) = 3$
(i) p-rank$(A) = 3 \Rightarrow$ three 1-by-1 JBs;
(ii) p-rank$(A) = 2 \Rightarrow$ one 2-by-2 JB and two 1-by-1 JBs;
(d) rank$(A) = 2$
(i) p-rank$(A) = 2 \Rightarrow$ four 1-by-1 JBs.

For the case $n = 7$, we can use the previous results to classify all possible triples. (We ignore the trivial triples (7,7,7) and (7,1,1) in this discussion.) For instance, by Proposition 4.4 the triples (7,6,6), (7,6,5), and (7,6,4) are the only realizable triples when the rank is fixed at six. All of the remaining realizable triples (which are listed below) follow from the list for $n = 6$ and Corollary 4.9, and also by Proposition 4.3. The realizable triples for $n = 7$, are (7,6,6), (7,6,5), (7,6,4); (7,5,5), (7,5,4), (7,5,3); (7,4,4), (7,4,3), (7,4,2); (7,3,3), (7,3,2); (7,2,2). In the case when $n = 7$, using the complete list of triples above and Theorem 4.14 it follows that all possible Jordan structures can be characterized (see list to follow) with the exception of one case, namely the triple (7,5,3). For this particular triple there are two possible Jordan structures: (1) one 3-by-3 JB and one 1-by-1 JB, or (2) two 2-by-2 JBs. The Jordan structure in case (1) is possible by considering a matrix which realizes the triple (6,5,3) (which exists), and then using Corollary 4.8 to construct a 7-by-7 irreducible TN matrix with the desired Jordan structure. For case (2), we do not know of a general technique to rule out or guarantee such a Jordan structure. However, consider the following matrix:

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

Then A is a 7-by-7 irreducible TN matrix with rank$(A) = 5$ and p-rank$(A) = 3$, and has Jordan structure

$$\left[\begin{array}{ccc|cc|cc} * & & & & & & \\ & * & & & & & \\ & & * & & & & \\ \hline & & & 0 & 1 & & \\ & & & 0 & 0 & & \\ \hline & & & & & 0 & 1 \\ & & & & & 0 & 0 \end{array}\right].$$

Hence the second possible Jordan structure does indeed occur for some 7-by-7 irreducible TN matrices. The complete list of all possible Jordan structures for $n = 7$ is given below.

- $n = 7$:
  (1) rank$(A) = 6$
  (a) p-rank$(A) = 6 \Rightarrow$ one 1-by-1 JB;
  (b) p-rank$(A) = 5 \Rightarrow$ one 2-by-2 JB;
  (c) p-rank$(A) = 4 \Rightarrow$ one 3-by-3 JB;

(2) rank$(A) = 5$
  (a) p-rank$(A) = 5 \Rightarrow$ two 1-by-1 JBs;
  (b) p-rank$(A) = 4 \Rightarrow$ one 2-by-2 JB and one 1-by-1 JB;
  (c) p-rank$(A) = 3 \Rightarrow$ one 3-by-3 JB and one 1-by-1 JB, or two 2-by-2 JBs;

(3) rank$(A) = 4$
  (a) p-rank$(A) = 4 \Rightarrow$ three 1-by-1 JBs;
  (b) p-rank$(A) = 3 \Rightarrow$ one 2-by-2 JB and two 1-by-1 JBs;
  (c) p-rank$(A) = 2 \Rightarrow$ two 2-by-2 JBs and one 1-by-1 JB;

(4) rank$(A) = 3$
  (a) p-rank$(A) = 3 \Rightarrow$ four 1-by-1 JBs;
  (b) p-rank$(A) = 2 \Rightarrow$ one 2-by-2 JB and three 1-by-1 JBs;

(5) rank$(A) = 2$
  (a) p-rank$(A) = 2 \Rightarrow$ five 1-by-1 JBs.

We conclude this section with a discussion about some future work along these lines and a couple of open problems, which we feel are not only interesting but are important for continuing progress in this area.

First, and the most important issue, is classifying all possible Jordan structures for n-by-n irreducible TN matrices. By the results presented thus far we have completed this classification through $n = 7$. (In fact, we are now close to completing this classification through $n = 8$.) We are in the process of working on many new and worthwhile ideas to continue this classification.

A related, but apparently less difficult (although by no means easy), problem is determining which triples $(n, \text{rank, p-rank})$ are realizable by the class of n-by-n irreducible TN matrices. Again it follows from the analysis in this section that this issue has been completely settled through $n = 7$, along with some general existence results. (Applying arguments similar to those throughout this section we can extend this to $n = 8$.) It seems that, at least thus far, a general result for all realizable triples is very possible, and we continue to develop new techniques and ideas in hopes of obtaining such a general result.

There are two problems that we wish to touch upon here for a couple of reasons. First, answering these questions will definitely shed some light onto the previous two unresolved issues, and second, they were both unexpected and still (for the most part) remain unexplained. The first problem is concerned with the size of the largest Jordan block corresponding to zero for an n-by-n irreducible TN matrix. Through $n = 7$ (and nearly for $n = 8$) the size of the largest Jordan block corresponding to zero is at most the principal rank. Moreover, in general this result holds when p-rank$(A) \geq \lceil \frac{n}{2} \rceil$, or when p-rank$(A) \geq$ rank$(A) - 1$, or also when p-rank$(A) = 1$ or 2 (by Theorem 4.14). Recall that the size of the largest Jordan block corresponding to zero is at most rank(A) − p-rank$(A) + 1$. Thus if p-rank$(A) \geq \frac{\text{rank}(A)+1}{2}$, then the result holds. At this point this question is still unresolved in general, and we do not know of a good reason why such a result should hold. As a final note on this problem, this claim for the size of the largest Jordan block is equivalent to the following equality: rank$(A^{\text{p-rank}(A)}) =$ p-rank(A).

The next and final problem we discuss here is concerned with the existence of the triple, $(n, \text{rank, p-rank})$. Consider the triple (6,4,2), which was shown to be a realizable triple. (Observe that the Jordan structure associated with such a triple must consist of 2 positive distinct eigenvalues, and two, 2-by-2 Jordan blocks corresponding to zero.) Then note that the triple $(6, 6 - \mathbf{2} + 1, 6 - \mathbf{4} + 1) = (6, 5, 3)$ is also a realizable

triple. Moreover, this particular rearrangement gives rise to realizable triples for every known realizable triple (compare Propositions 4.3 and 4.4). If the triple (n, k, p) can be realized, then it seems that the triple $(n, n-p+1, n-k+1)$ can also be realized by an n-by-n irreducible TN matrix. Again we have little to offer about why such a result should be true, but nevertheless, it is an interesting property that these triples seem to possess.

REFERENCES

[1] T. Ando, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.

[2] W. W. Barrett and C. R. Johnson, *Possible spectra of totally positive matrices*, Linear Algebra Appl., 62 (1984), pp. 231–233.

[3] A. Berenstein, S. Fomin, and A. Zelevinsky, *Parameterizations of canonical bases and totally positive matrices*, Adv. Math., 122 (1996), pp. 49–149.

[4] C. W. Cryer, *Some properties of totally positive matrices*, Linear Algebra Appl., 15 (1976), pp. 1–25.

[5] S. P. Eveson, *The eigenvalue distribution of oscillatory and strictly sign-regular matrices*, Linear Algebra Appl., 246 (1996), pp. 17–21.

[6] S. M. Fallat, *Totally Nonnegative Matrices*, Ph.D. dissertation, Department of Mathematics, College of William and Mary, Williamsburg, VA, 1999.

[7] S. Fomin and A. Zelevinsky, *Double bruhat cells and total positivity*, J. Amer. Math. Soc., 12 (1999), pp. 335–380.

[8] S. Friedland, *Weak interlacing properties of totally positive matrices*, Linear Algebra Appl., 71 (1985), pp. 247–266.

[9] F. R. Gantmacher and M. G. Krein, *Sur les matrices complement non-negatives et oscillatories*, Compositio Math., 4 (1937), pp. 445–476.

[10] F. R. Gantmacher and M. G. Krein, *Oszillationsmatrizen, Oszillationskerne und kleine Schwingungen Mechanischer Systeme*, Akademie-Verlag, Berlin, 1960.

[11] M. Gasca and C. A. Micchelli, *Total Positivity and its Applications*, Math. Appl. 359, Kluwer Academic, Dordrecht, The Netherlands, 1996.

[12] M. Gasca and J. M. Peña, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, Math. Appl. 359, Kluwer Academic, Dordrecht, The Netherlands, 1996, pp. 109–130.

[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[14] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[15] S. Karlin, *Total Positivity* I, Stanford University Press, Stanford, 1968.

[16] A. Pinkus, *An interlacing property of eigenvalues of strictly totally positive matrices*, Linear Algebra Appl., 279 (1998), pp. 201–206.

[17] C. E. Radke, *Classes of matrices with distinct, real characteristic values*, SIAM J. Appl. Math., 16 (1968), pp. 1192–1207.

[18] J. W. Rainey and G. J. Halbetler, *Tridiagonalization of completely nonnegative matrices*, Math. Comp., 26 (1972), pp. 121–128.

[19] A. Whitney, *A reduction theorem for totally positive matrices*, J. Anal. Math., 2 (1952), pp. 88–92.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 647–665

PRECONDITIONERS FOR NONDEFINITE HERMITIAN TOEPLITZ SYSTEMS*

RAYMOND H. CHAN†, DANIEL POTTS‡, AND GABRIELE STEIDL§

Abstract. This paper is concerned with the construction of circulant preconditioners for Toeplitz systems arising from a piecewise continuous generating function with sign changes.

If the generating function is given, we prove that for any $\varepsilon > 0$, only $\mathcal{O}(\log N)$ eigenvalues of our preconditioned Toeplitz systems of size $N \times N$ are not contained in $[-1-\varepsilon, -1+\varepsilon] \cup [1-\varepsilon, 1+\varepsilon]$. The result can be modified for trigonometric preconditioners. We also suggest circulant preconditioners for the case that the generating function is not explicitly known and show that only $\mathcal{O}(\log N)$ absolute values of the eigenvalues of the preconditioned Toeplitz systems are not contained in a positive interval on the real axis.

Using the above results, we conclude that the preconditioned minimal residual method requires only $\mathcal{O}(N \log^2 N)$ arithmetical operations to achieve a solution of prescribed precision if the spectral condition numbers of the Toeplitz systems increase at most polynomial in N. We present various numerical tests.

Key words. nondefinite Toeplitz matrices, circulant matrices, Krylov space methods, preconditioners

AMS subject classifications. 65F10, 65F15, 65T50

PII. S0895479899362521

1. Introduction.

Let $\mathcal{L}_{2\pi}$ be the space of 2π-periodic Lebesgue integrable real-valued functions and let $C_{2\pi}$ be the subspace of 2π-periodic real-valued continuous functions with norm

$$\|f\|_\infty := \max_{t \in [-\pi, \pi]} |f(t)| (f \in C_{2\pi}).$$

The Fourier coefficients of $f \in \mathcal{L}_{2\pi}$ are given by

$$a_k = a_k(f) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \mathrm{e}^{-\mathrm{i}kt} \, \mathrm{d}t (k \in \mathbb{Z}),$$

and the sequence $\{\boldsymbol{A}_N(f)\}_{N=1}^{\infty}$ of (N, N)-Toeplitz matrices generated by f is defined by

$$\boldsymbol{A}_N = \boldsymbol{A}_N(f) := (a_{j-k}(f))_{j,k=0}^{N-1}.$$

Since $f \in \mathcal{L}_{2\pi}$ is real-valued, the matrices $\boldsymbol{A}_N(f)$ are Hermitian.

We are interested in the iterative solution of Toeplitz systems

$$\boldsymbol{A}_N(f) \, \boldsymbol{x} = \boldsymbol{b}, \tag{1.1}$$

*Received by the editors October 25, 1999; accepted for publication (in revised form) by L. Reichel April 30, 2000; published electronically October 25, 2000. This research was supported in part by the Hong Kong–German Joint Research Collaboration grant from the Deutscher Akademischer Austauschdienst and Hong Kong Research Grants Council grant CUHK4212/99P.

http://www.siam.org/journals/simax/22-3/36252.html

†Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong (rchan@math.cuhk.edu.hk).

‡Medizinische Universität Lübeck, Institut für Mathematik, Wallstraße 40, D-23560 Lübeck, Germany (potts@math.mu-luebeck.de).

§Universität Mannheim, Fakultät für Mathematik und Informatik, D-68131 Mannheim, Germany (steidl@math.uni-mannheim.de).

where the generating function $f \in \mathcal{L}_{2\pi}$. To be more precise, we are looking for good preconditioning strategies so that Krylov space methods applied to the preconditioned system converge in a small number of iteration steps. Note that by the Toeplitz structure of $\boldsymbol{A}_N$ each iteration step requires only $\mathcal{O}(N \log N)$ arithmetical operations by using fast Fourier transforms.

Preconditioning techniques for Toeplitz systems have been well studied in the past 10 years. However, most of the papers in this area are concerned with the case where the generating function f is either positive or nonnegative; see, for instance, [5, 3, 20, 7, 17, 10] and the references therein. In this paper, we consider f that has sign changes. The method we propose here will also work for generating functions that are positive or nonnegative.

Up to now iterative methods for Toeplitz systems with generating functions having different signs were only considered in [20, 19, 22, 24] and in connection with non-Hermitian systems in [8, 6]. In [8], we have constructed circulant preconditioners for non-Hermitian Toeplitz matrices with known generating function of the form

$$f = p\, h,$$

where p is an arbitrary trigonometric polynomial and h is a function from the Wiener class with $|h| > 0$. We proved that the preconditioned matrices have *singular values* properly clustered at 1. Then, if the *spectral condition number* of $\boldsymbol{A}_N(f)$ fulfills $\kappa_2(\boldsymbol{A}_N(f)) = \mathcal{O}(N^\alpha)$, the *conjugate gradient (CG) method* applied to the normal equation requires only $\mathcal{O}(\log N)$ iteration steps to produce a solution of fixed precision. However, in general, nothing can be said about the eigenvalues of the preconditioned matrix.

In this paper, we consider real-valued functions $f \in \mathcal{L}_{2\pi}$ of the form

$$f = p_s h, \tag{1.2}$$

where

$$p_s(t) := \prod_{j=1}^{\mu} (2 - 2\cos(t - t_j))^{s_j}, \quad s := \sum_{j=1}^{\mu} s_j \tag{1.3}$$

is a trigonometric polynomial with a finite number of zeros $t_j \in [-\pi, \pi)$ $(j = 1, \dots, \mu)$ of even order $2s_j$ and where $h \in \mathcal{L}_{2\pi}$ is a piecewise continuous function with simple discontinuities at ξ_j $(j = 1, \dots, \nu)$, i.e., there exist $h(\xi_j \pm 0)$ and $h(\xi_j + 0) - h(\xi_j - 0) = \alpha_j \neq 0$. For simplicity let $h(\xi_j) = (h(\xi_j - 0) + h(\xi_j + 0))/2$. Further, we assume that

$$\overline{\{|h(t)| : t \in [-\pi, \pi);\, |h(t)| > 0\}} \subseteq [h_-, h_+], \tag{1.4}$$

where $0 < h_- \leq h_+ < \infty$. In particular, we are interested in the Heaviside function h.

A similar setting was also considered in [20]. Serra Capizzano suggested the application of band-Toeplitz preconditioners $\boldsymbol{A}_N(p_s)$ in combination with CG applied to the normal equation. He proved, beyond a more general result which cannot directly be used for preconditioning, that at most $o(N)$ eigenvalues of the preconditioned matrix $\boldsymbol{A}_N(p_s)^{-1}\boldsymbol{A}_N(f)$ have absolute values not contained in a positive interval on the real axis.

The same author suggested in [19] a preconditioning method based on the Sherman–Morrison–Woodbery formula and some kind of normal equation for generating functions with zeros of odd order.

A result with $o(N)$ outliers was also obtained in [23], where the application of preconditioned GMRES was examined.

In the following, we construct circulant preconditioners for the *minimal residual method* (MINRES). Note that preconditioned MINRES avoids the transformation of the original system to the normal equation but requires Hermitian positive definite preconditioners. Then, the preconditioned matrices are again Hermitian, so that the absolute values of their eigenvalues coincide with their singular values. If the generating function is given, we prove that for any $\varepsilon > 0$, only $\mathcal{O}(\log N)$ singular values of the preconditioned matrices are not contained in $[1-\varepsilon, 1+\varepsilon]$. We also construct circulant preconditioners for the case that the generating function of the Toeplitz matrices is not explicitly known. For this, we use positive reproducing kernels with special properties previously applied by the authors in [17, 10] and show that $\mathcal{O}(\log N)$ singular values of the preconditioned matrices are not contained in a positive interval on the real axis. Then, if in addition $\kappa_2(\boldsymbol{A}_N(f)) = \mathcal{O}(N^\alpha)$, preconditioned MINRES converges in at most $\mathcal{O}(\log N)$ iteration steps. In summary, the proposed algorithm requires only $\mathcal{O}(N \log^2 N)$ arithmetical operations.

Note that the theoretical verification of the above assumption on the condition number of $\boldsymbol{A}_N(f)$ is not straightforward. See [4] for examples of banded indefinite Toeplitz matrices with exponentially (or even faster) growing condition numbers.

This paper is organized as follows. In section 2, we introduce circulant preconditioners for (1.1) under the assumption that the generating function of the sequence of Toeplitz matrices is known and prove clustering results for the eigenvalues of the preconditioned matrices. Section 3 deals with the construction of preconditioners if the generating function of the Toeplitz matrices is not explicitly known. In section 4, we modify the results of section 2 with respect to trigonometric preconditioners. The convergence of MINRES applied to our preconditioned Toeplitz systems is considered in section 5. Finally, we present numerical results in section 6.

2. Circulant preconditioners involving generating functions. First we introduce some basic notation. By $\boldsymbol{R}_N(M)$ we denote arbitrary (N,N)-matrices of rank *at most* M. Let $\boldsymbol{M}_N(g)$ be the circulant (N,N)-matrix

$$\boldsymbol{M}_N(g) := \boldsymbol{F}_N \operatorname{diag}\left(g\left(\frac{2\pi l}{N}\right)\right)_{l=0}^{N-1} \boldsymbol{F}_N^*,$$

where $\boldsymbol{F}_N$ denotes the Nth *Fourier matrix*

$$\boldsymbol{F}_N := \frac{1}{\sqrt{N}}\left(\mathrm{e}^{-2\pi \mathrm{i} jk/N}\right)_{j,k=0}^{N-1}$$

and where $\boldsymbol{F}^*$ is the transposed complex conjugate matrix of $\boldsymbol{F}$. For a trigonometric polynomial $q(t) := \sum_{k=-n_1}^{n_2} q_k \mathrm{e}^{\mathrm{i}kt}$, the matrices $\boldsymbol{A}_N(q)$ and $\boldsymbol{M}_N(q)$ are related by

$$\boldsymbol{A}_N(q) = \boldsymbol{M}_N(q) + \boldsymbol{R}_N(n_1+n_2) \tag{2.1}$$

(see [15]). For a function g with a finite number of zeros we define the set $I_N(g)$ by

$$I_N(g) := \left\{ l = 0, \ldots, N-1 : \; g\left(\frac{2\pi l}{N}\right) \neq 0 \right\}$$

and the points $x_{N,l}(g)$ $(l = 0, \ldots, N-1)$ by

$$x_{N,l}(g) := \begin{cases} \frac{2l\pi}{N} & \text{if} \quad l \in I_N(g)\,, \\ \frac{2\tilde{l}\pi}{N} & \text{otherwise,} \end{cases}$$

where $\tilde{l} \in \{0, \ldots, N-1\}$ is the next higher index to l so that $\tilde{l} \in I_N(g)$. For N large enough we can simply choose $\tilde{l} = l+1 \mod N$. By $\boldsymbol{M}_{N,g}(f)$ we denote the circulant matrix

$$\boldsymbol{M}_{N,g}(f) := \boldsymbol{F}_N \operatorname{diag}\left(f(x_{N,l}(g))\right)_{l=0}^{N-1} \boldsymbol{F}_N^*. \tag{2.2}$$

If g has m zeros, then we have by construction that

$$\boldsymbol{M}_N(f) = \boldsymbol{M}_{N,g}(f) + \boldsymbol{R}_N(m). \tag{2.3}$$

Now assume that the sequence $\{\boldsymbol{A}_N(f)\}_{N=1}^{\infty}$ of nonsingular Toeplitz matrices is generated by a known piecewise continuous function $f \in \mathcal{L}_{2\pi}$ of the form (1.2)–(1.4). Then we suggest the Hermitian positive definite circulant matrix $\boldsymbol{M}_{N,f}(|f|)$ as preconditioner for MINRES.

We examine the distribution of the eigenvalues of $\boldsymbol{M}_{N,f}(|f|)^{-\frac{1}{2}} \boldsymbol{A}_N(f) \boldsymbol{M}_{N,f}(|f|)^{-\frac{1}{2}}$.

The following theorem is Lemma 10 of [26] written with respect to our notation.

THEOREM 2.1. *Let $h \in \mathcal{L}_{2\pi}$ be a piecewise continuous function having only simple discontinuities at $\xi_j \in [-\pi, \pi)$ $(j = 1, \ldots, \nu)$. By $\mathcal{F}_N$ we denote the* Fejér kernel

$$\mathcal{F}_N(t) := \sum_{k=-(N-1)}^{N-1} \left(1 - \left|\frac{k}{N}\right|\right) \mathrm{e}^{\mathrm{i}kt} = 1 + 2\sum_{k=1}^{N-1} \left(1 - \frac{k}{N}\right) \cos kt \tag{2.4}$$

$$= \begin{cases} \frac{1}{N}\left(\sin\left(\frac{Nt}{2}\right) / \sin\left(\frac{t}{2}\right)\right)^2, & t \neq 0, \\ \frac{1}{N}, & t = 0, \end{cases} \tag{2.5}$$

and by $\mathcal{F}_N * h$ we denote the cyclic convolution *of $\mathcal{F}_N$ and h. Then, for any $\varepsilon > 0$, there exist constants $0 < c_1 \leq c_2 < \infty$ independent of N so that the number $\nu(\varepsilon; \boldsymbol{A}_N)$ of eigenvalues of $\boldsymbol{A}_N(h) - \boldsymbol{M}_N(\mathcal{F}_N * h)$ with absolute value exceeding ε can be estimated by*

$$c_1 \log\,(N) \leq \nu(\varepsilon; \boldsymbol{A}_N) \leq c_2 \log\,(N).$$

In other words, we have by Theorem 2.1 that

$$\boldsymbol{A}_N(h) = \boldsymbol{M}_N(\mathcal{F}_N * h) + \boldsymbol{V}_N + \boldsymbol{U}_N, \tag{2.6}$$

where $\boldsymbol{V}_N$ is a matrix of spectral norm $\leq \varepsilon$ and where

$$c_1 \log N \leq \operatorname{rank}\,(\boldsymbol{U}_N) \leq c_2 \log N.$$

Using Theorem 2.1, we can prove the following lemma.

LEMMA 2.2. *Let $f = p_s h \in \mathcal{L}_{2\pi}$ be given by* (1.2)–(1.4). *Then, for any $\varepsilon > 0$ and sufficiently large N, the number of singular values of $\boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{A}_N(h) \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}}$ which are not contained in the interval $[1-\varepsilon, 1+\varepsilon]$ is $\mathcal{O}(\log N)$.*

Proof. By (2.6) and since the eigenvalues of $\boldsymbol{M}_{N,f}(|h|)$ are restricted from below by h_-, it remains to show that for any $\varepsilon > 0$ and sufficiently large N, except for $\mathcal{O}(\log N)$ eigenvalues, all eigenvalues of $\boldsymbol{M}_{N,f}(|h|)^{-1} \boldsymbol{M}_N(\mathcal{F}_N * h)$ have absolute values in $[1-\varepsilon, 1+\varepsilon]$. Indeed we will prove that there are only $\mathcal{O}(1)$ outliers.

For this we mainly follow the lines of proof of Gibb's phenomenon. Without loss of generality we assume that $h \in \mathcal{L}_{2\pi}$ has only one jump at $\xi_1 = 0$ of height α_1.

First we examine $\mathcal{F}_N * g$, where g is given by

$$g(x) := \begin{cases} \frac{1}{2}(\pi - x), & x \in (0, \pi), \\ \frac{1}{2}(-x - \pi), & x \in (-\pi, 0), \\ 0, & x = 0. \end{cases}$$

By (2.4) and since g has Fourier series

$$g(x) \sim \sum_{k=1}^{\infty} \frac{1}{k} \sin kx,$$

we obtain

$$\int_0^x \mathcal{F}_N(t)\,\mathrm{d}t = x + 2\sum_{k=1}^{N-1}\left(\frac{1}{k} - \frac{1}{N}\right)\sin kx \;\; = x + 2\,(\mathcal{F}_N * g)(x),$$

and further by (2.5)

$$\begin{aligned}(\mathcal{F}_N * g)(x) &= \frac{1}{2N}\int_0^x \left(\frac{\sin\frac{Nt}{2}}{\sin\frac{t}{2}}\right)^2 \mathrm{d}t - \frac{x}{2} \\ &= \frac{1}{2N}\int_0^x \left(\frac{\sin\frac{Nt}{2}}{\frac{t}{2}}\right)^2 \mathrm{d}t + \frac{1}{2N}\int_0^x \left(\frac{1}{(\sin\frac{t}{2})^2} - \frac{1}{(\frac{t}{2})^2}\right)\left(\sin\frac{Nt}{2}\right)^2 \mathrm{d}t - \frac{x}{2} \\ &= \int_0^{\frac{Nx}{2}} \left(\frac{\sin t}{t}\right)^2 \mathrm{d}t + \mathcal{O}(N^{-1}) - \frac{x}{2},\end{aligned}$$

and by partial integration and definition of g

$$(\mathcal{F}_N * g)(x) - g(x) = \frac{-(\sin\frac{Nx}{2})^2}{\frac{Nx}{2}} + \mathrm{si}\,(Nx) - \frac{\pi}{2} + \mathcal{O}(N^{-1}) \quad (x \in (0, \pi)),$$

where $\mathrm{si}\,(y) := \int_0^y \frac{\sin t}{t}\,\mathrm{d}t$. We are interested in the behavior of

$$(\mathcal{F}_N * g)\left(\frac{2\pi l}{N}\right) - g\left(\frac{2\pi l}{N}\right) = \mathrm{si}\,(2\pi l) - \frac{\pi}{2} + \mathcal{O}(N^{-1}) \quad \left(l = 0, \ldots, \left\lceil\frac{N}{2}\right\rceil - 1\right).$$

Here $\lceil x \rceil$ denotes the smallest integer $\geq x$. It is well known that $\lim_{x\to\infty} \mathrm{si}\,(x) = \frac{\pi}{2}$. Thus, if $l = l(N) \to \infty$ for $N \to \infty$, then, for any $\varepsilon > 0$, there exists $N_0 = N_0(\varepsilon)$ so that

$$\left|(\mathcal{F}_N * g)\left(\frac{2\pi l}{N}\right) - g\left(\frac{2\pi l}{N}\right)\right| < \frac{\pi h_-}{2\alpha_1}\,\varepsilon \quad \text{for all } N \geq N_0. \tag{2.7}$$

The same holds if we approach 0 from the left, i.e., if we consider $2\pi l/N$ for $l = \lceil\frac{N}{2}\rceil, \ldots, N-1$.

Next we have by definition of g and h that

$$\tilde{h}(x) := h(x) - \frac{\alpha_1}{\pi}\,g(x)$$

is a continuous function. Since $\mathcal{F}_N$ is a reproducing kernel, for any $\varepsilon > 0$, there exists $\tilde{N}_0 = \tilde{N}_0(\varepsilon)$ so that for all $l \in \{0, \dots, N-1\}$

$$\left| (\mathcal{F}_N * \tilde{h}) \left(\frac{2\pi l}{N}\right) - \tilde{h}\left(\frac{2\pi l}{N}\right) \right| < \frac{\varepsilon}{2} h_- \quad \text{for all } N \geq \tilde{N}_0. \tag{2.8}$$

Assume that $l = l(N) \to \infty$ for $N \to \infty$ ($l \in \{0, \dots, \lceil \frac{N}{2} \rceil - 1\}$). Then we obtain by (2.7) and (2.8) that for any $\varepsilon > 0$ there exists $N(\varepsilon) = \max(N_0, \tilde{N}_0)$ so that

$$\begin{aligned} \left| (\mathcal{F}_N * h) \left(\frac{2\pi l}{N}\right) - h\left(\frac{2\pi l}{N}\right) \right| &\leq \left| (\mathcal{F}_N * \tilde{h}) \left(\frac{2\pi l}{N}\right) - \tilde{h}\left(\frac{2\pi l}{N}\right) \right| \\ &\quad + \frac{\alpha_1}{\pi} \left| (\mathcal{F}_N * g) \left(\frac{2\pi l}{N}\right) - g\left(\frac{2\pi l}{N}\right) \right|, \\ \left| (\mathcal{F}_N * h) \left(\frac{2\pi l}{N}\right) - h\left(\frac{2\pi l}{N}\right) \right| &\leq \varepsilon\, h_- \qquad \text{for all } N \geq N(\varepsilon), \end{aligned}$$

and consequently, since $|h\left(\frac{2\pi l}{N}\right)| \geq h_-$ $(l \in I_N(f))$,

$$1 - \varepsilon \leq \frac{\left|(\mathcal{F}_N * h)\left(\frac{2\pi l}{N}\right)\right|}{\left|h\left(\frac{2\pi l}{N}\right)\right|} \leq 1 + \varepsilon \qquad (l \in I_N(f)). \tag{2.9}$$

Let $m \leq \mu + \nu$ denote the number of zeros of f which are equal to one of the points $2\pi l/N$ $(l = 0, \dots, N-1)$. Then the set

$$\left\{ \frac{\left|(\mathcal{F}_N * h)\left(\frac{2\pi l}{N}\right)\right|}{\left|h\left(\frac{2\pi l}{N}\right)\right|} : l \in I_N(f) \right\}$$

contains at least $N - m$ absolute values of eigenvalues of $\boldsymbol{M}_{N,f}(|h|)^{-1}\boldsymbol{M}_N(\mathcal{F}_N * h)$ and we conclude by (2.9) that except for $\mathcal{O}(1)$ eigenvalues and sufficiently large N, all eigenvalues of $\boldsymbol{M}_{N,f}(|h|)^{-1}\boldsymbol{M}_N(\mathcal{F}_N * h)$ have absolute values contained in $[1-\varepsilon, 1+\varepsilon]$. This completes the proof. ∎

Remark 2.3. In a similar way as above we can prove that for any $\varepsilon > 0$ and N sufficiently large, the number of eigenvalues of $\boldsymbol{A}_N(h)$ with absolute values not in the interval $[h_- - \varepsilon, h_+]$ is $\mathcal{O}(\log N)$.

Note that the property that at most $o(N)$ eigenvalues of $\boldsymbol{A}_N(h)$ have absolute values not contained in $[h_- - \varepsilon, h_+]$ follows simply from the fact that the singular values of $\boldsymbol{A}_N(h)$ are distributed as $|h|$ [14, 23]. ∎

THEOREM 2.4. *Let $f = p_s h \in \mathcal{L}_{2\pi}$ be given by* (1.2)–(1.4). *Then, for any $\varepsilon > 0$ and sufficiently large N, except for $\mathcal{O}(\log N)$ singular values, all singular values of*

$$\boldsymbol{M}_{N,f}(|f|)^{-\frac{1}{2}}\boldsymbol{A}_N(f)\boldsymbol{M}_{N,f}(|f|)^{-\frac{1}{2}}$$

are contained in $[1 - \varepsilon, 1 + \varepsilon]$.

Proof. The polynomial p_s in (1.3) can be rewritten as

$$p_s = p\bar{p},$$

where

$$p(t) := \prod_{j=1}^{\mu} (1 - \mathrm{e}^{-\mathrm{i}t_j}\,\mathrm{e}^{\mathrm{i}t})^{s_j}, \quad \sum_{j=1}^{\mu} s_j = s,$$

and $\bar{p}(t)$ is the complex conjugate of $p(t)$. By straightforward computation it is easy to check that

$$\begin{aligned}\boldsymbol{A}_N(f) &= \boldsymbol{A}_N(p_s\, h) \,=\, \boldsymbol{A}_N(p\, h\, \bar{p}) \\ &= \boldsymbol{A}_N(p\, h)\, \boldsymbol{A}_N(\bar{p}) \,+\, \boldsymbol{R}_N^c(s) \\ &= \boldsymbol{A}_N(p)\, \boldsymbol{A}_N(h)\, \boldsymbol{A}_N(\bar{p}) \,+\, \boldsymbol{R}_N^r(s)\, \boldsymbol{A}_N(\bar{p}) \,+\, \boldsymbol{R}_N^c(s) \\ &= \boldsymbol{A}_N(p)\, \boldsymbol{A}_N(h)\, \boldsymbol{A}_N(\bar{p}) \,+\, \boldsymbol{R}_N(2s), \end{aligned} \tag{2.10}$$

where only the first s columns (rows) of $\boldsymbol{R}_N^{c\,(r)}(s)$ are nonzero columns (rows).

Since $|f| = p\,\bar{p}\,|h|$ the eigenvalues of $\boldsymbol{M}_{N,f}(|f|)^{-1}\boldsymbol{A}_N(f)$ coincide with the eigenvalues of

$$\boldsymbol{B}_N(f) := \boldsymbol{M}_{N,f}(|h|)^{-1/2}\, \boldsymbol{M}_{N,f}(p)^{-1}\, \boldsymbol{A}_N(f) \boldsymbol{M}_{N,f}(\bar{p})^{-1}\, \boldsymbol{M}_{N,f}(|h|)^{-1/2}. \tag{2.11}$$

Now we obtain by (2.10), (2.1), and (2.3) that

$$\begin{aligned}\boldsymbol{B}_N(f) &= \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{M}_{N,f}(p)^{-1} \boldsymbol{A}_N(p) \boldsymbol{A}_N(h) \boldsymbol{A}_N(\bar{p}) \boldsymbol{M}_{N,f}(\bar{p})^{-1} \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \\ &\quad + \boldsymbol{R}_N(2s) \\ &= \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{M}_{N,f}(p)^{-1} (\boldsymbol{M}_{N,f}(p) + \boldsymbol{R}_N(s+m)) \boldsymbol{A}_N(h) \\ &\quad \cdot (\boldsymbol{M}_{N,f}(\bar{p}) + \boldsymbol{R}_N(s+m)) \boldsymbol{M}_{N,f}(\bar{p})^{-1} \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} + \boldsymbol{R}_N(2s) \\ &= \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{A}_N(h) \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \,+\, \boldsymbol{R}_N(4s+2m). \end{aligned} \tag{2.12}$$

By Lemma 2.2, for any $\varepsilon > 0$ and N sufficiently large, except for $\mathcal{O}(\log N)$ singular values, all singular values of $\boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{A}_N(h) \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}}$ are contained in $[1-\varepsilon, 1+\varepsilon]$. Now the assertion follows by (2.12) and Weyl's interlacing theorem [13, p. 184]. □

3. Circulant preconditioners involving positive kernels. In many applications we know only the entries $a_k(f)$ of the Toeplitz matrices $\boldsymbol{A}_N(f)$ and not the generating function itself. In this case, we use even positive reproducing kernels $K_N \in C_{2\pi}$. These are trigonometric polynomials of the form

$$K_N(t) := c_{N,0} + 2 \sum_{k=1}^{N-1} c_{N,k} \cos kt, \quad c_{N,k} = a_k(K_N) \in \mathbb{R}$$

satisfying $K_N \geq 0$,

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} K_N(t)\, \mathrm{d}t = 1, \tag{3.1}$$

and the *reproducing property*

$$\lim_{N\to\infty} \|f - K_N * f\|_\infty = 0 \quad \text{for all } f \in C_{2\pi}.$$

Since

$$(K_N * f)(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) K_N(x-t)\, \mathrm{d}t = \sum_{k=-(N-1)}^{N-1} a_k(f)\, c_{N,k}\, \mathrm{e}^{\mathrm{i}kx},$$

the cyclic convolution of K_N and f is determined by the first N Fourier coefficients of f. As a preconditioner which can be constructed from the entries of $\boldsymbol{A}_N(f)$ without explicit knowledge of f we suggest the circulant matrix $\boldsymbol{M}_{N,K_N*f}(|K_N * f|)$.

In order to obtain a suitable distribution of the eigenvalues of the preconditioned matrices, we need kernels with a special property which is related to the order

$$\sigma := \max_{j=1,\dots,\mu} s_j$$

of the zeros of p_s.

The *generalized Jackson kernels* $\mathcal{J}_{m,N}$ of degree $\le N-1$ are defined by

$$K_{m,N}(t) = \mathcal{J}_{m,N}(t) := \lambda_{m,N} \left(\frac{\sin(nt/2)}{\sin(t/2)} \right)^{2m} \quad (m \in \mathbb{N}), \tag{3.2}$$

where $n := \lfloor \frac{N-1}{m} \rfloor + 1$ and where $\lambda_{m,N}$ is determined by (3.1). Here $\lfloor t \rfloor$ denotes the largest integer $\le t$. In particular, we have that

$$\lambda_{m,N} \sim N^{1-2m},$$

i.e., there exist positive constants c_1, c_2 so that $c_1\, N^{1-2m} \le \lambda_{m,N} \le c_2\, N^{1-2m}$. See [11, pp. 203–204]. A possibility for the construction of the Fourier coefficients of $\mathcal{J}_{m,N}$ is prescribed in [10].

The *B-spline kernels* $\mathcal{B}_{m,N}$ of degree $\le N-1$ are defined by

$$K_{m,N}(t) = \mathcal{B}_{m,N}(t) := \frac{N}{m}\, \frac{1}{M_{2m}(0)} \sum_{r \in Z} \left(\operatorname{sinc} \left(\frac{N}{m} \left(\frac{t+2\pi r}{2} \right) \right) \right)^{2m}, \tag{3.3}$$

where M_m denotes the *centered cardinal B-spline* of order m and

$$\operatorname{sinc} t := \begin{cases} \frac{\sin t}{t}, & t \ne 0\,, \\ 1, & t = 0. \end{cases}$$

See [17, 9]. Since

$$\mathcal{B}_{m,N}(t) := 1 + \frac{2}{M_{2m}(0)} \sum_{k=1}^{N-1} M_{2m}\left(\frac{mk}{N} \right) \cos kt,$$

the Fourier coefficients of $\mathcal{B}_{m,N}$ are given by values of centered cardinal B-splines. Note that $\mathcal{J}_{1,N} = \mathcal{B}_{1,N}$ is just the Fejér kernel $\mathcal{F}_N$.

The above kernels have the following important property.

THEOREM 3.1. *Let $f = p_s h \in \mathcal{L}_{2\pi}$ be given by* (1.2)–(1.4). *Assume that for all t_j $(j \in \{1,\dots,\mu\})$ with $t_j = \xi_k$ for some $k \in \{1,\dots,\nu\}$ and $\operatorname{sgn} h(\xi_k+0) \ne \operatorname{sgn} h(\xi_k-0)$ there exists a neighborhood $[t_j - \varepsilon_j, t_j + \varepsilon_j]$ $(\varepsilon_j > 0)$ of t_j so that f is a monotone function in this neighborhood and moreover $f(t_j - t) = -f(t_j + t)$ $(0 \le t \le \varepsilon_j)$. Let $K_N = K_{m,N}$ be given by* (3.2) *or* (3.3), *where*

$$m \ge \sigma + 1.$$

Then there exist $0 < \alpha \le \beta < \infty$ so that for $N \to \infty$, except for $\mathcal{O}(1)$ points, all points of the set $\{2\pi l/N : l \in I_N(f)\}$ fulfill

$$\frac{1}{\beta} \le \frac{|(K_N * f)(\frac{2\pi l}{N})|}{|f(\frac{2\pi l}{N})|} \le \frac{1}{\alpha}. \tag{3.4}$$

Proof. (1) First we consider the upper bound. Since p_s and K_N are nonnegative, we obtain

$$\begin{aligned}|(K_N * f)(x)| &\le \frac{1}{2\pi}\int_{-\pi}^{\pi} |h(t)|\, p_s(t)\, K_N(x-t)\,\mathrm{d}t \\ &\le h_+\,\frac{1}{2\pi}\int_{-\pi}^{\pi} p_s(t) K_N(x-t)\,\mathrm{d}t \;=\; h_+\,(K_N * p_s)(x).\end{aligned}$$

In [17, 10], we proved that $m \ge \sigma + 1$ implies that for all $x \in I_N(p_s) \supseteq I_N(f)$, there exists a constant $0 < c < \infty$ so that

$$\frac{(K_N * p_s)(x)}{p_s(x)} \le c.$$

Thus, since $|h(x)| \ge h_-$ for $(x \in I_N(f))$, we obtain

$$\frac{|(K_N * f)(x)|}{|f(x)|} \le \frac{h_+}{h_-}\,\frac{(K_N * p_s)(x)}{p_s(x)} \le \frac{h_+}{h_-}\,c \qquad (x \in I_N(f)).$$

(2) Next we deal with the lower bound.

(2.1) Let $x \in I_N(f)$ be not in the neighborhood of t_j $(j = 1, \ldots, \mu)$, i.e., there exist $b_j > 0$ independent of N so that $|x - t_j| \ge b_j > 0$ $(j = 1, \ldots, \mu)$. Then $|f(x)| \ge c > 0$ for all $x \in I_N(f)$. Further, since K_N is a reproducing kernel and by using the same arguments as in the proof of Lemma 2.2 if x is in the neighborhood of some ξ_k $(k = 1, \ldots, \nu)$, we obtain that, for any $\varepsilon > 0$ there exists $N(\varepsilon)$, so that except for at most a constant number of points, all considered points $x \in I_N(f)$ satisfy

$$|(K_N * f)(x) \;-\; f(x)| \le c\,\varepsilon \qquad (N \ge N(\varepsilon)),$$

and thus

$$\frac{|(K_N * f)(x)|}{|f(x)|} \ge 1 - \frac{c\,\varepsilon}{|f(x)|} \ge 1 - \varepsilon.$$

(2.2) It remains to consider the points $x = x(N) \in I_N(f)$ with $\lim_{N\to\infty} x(N) = t_j$ $(j = 1, \ldots, \mu)$.

For simplicity we assume that

$$p_s(t) = (2 - 2\cos t)^s = (2\,\sin(t/2))^{2s},$$

i.e., p_s has only a zero of order $2s$ at $t_1 = 0$. Let $x = x(N) \in I_N(f)$ with

$$\lim_{N\to\infty} x(N) = 0.$$

For any fixed $0 < b < \pi$ we obtain

$$\begin{aligned}(K_N * f)(x) &= \frac{1}{2\pi}\left(\int_{-b}^{b} f(t)K_N(x-t)\,\mathrm{d}t \;+\; \int_{-\pi}^{-b} f(t)K_N(x-t)\,\mathrm{d}t \right.\\ &\qquad\left. +\int_{b}^{\pi} f(t)K_N(x-t)\,\mathrm{d}t\right)\\ &= \frac{1}{2\pi}\left(\int_{-b}^{b} f(t)K_N(x-t)\,\mathrm{d}t \;+\; \int_{b+x}^{\pi+x} f(x-t)K_N(t)\,\mathrm{d}t \right.\\ &\qquad\left. +\int_{b-x}^{\pi-x} f(x+t)K_N(t)\,\mathrm{d}t\right),\end{aligned}$$

and since f is bounded

$$(K_N * f)(x) - \frac{1}{2\pi} \int_{-b}^{b} f(t) K_N(x-t)\,\mathrm{d}t \;\sim\; \left(\int_{b+x}^{\pi+x} + \int_{b-x}^{\pi-x} \right) K_N(t)\,\mathrm{d}t.$$

By definition of K_N we see that for any fixed $0 < \tilde{b} \le \pi$

$$\int_{\tilde{b}}^{\pi} K_N(t)\,\mathrm{d}t \;\le\; \text{const}\; N^{-2m+1}, \tag{3.5}$$

so that we get for small x (e.g., $x < b/2$)

$$(K_N * f)(x) = \frac{1}{2\pi} \int_{-b}^{b} f(t) K_N(x-t)\,\mathrm{d}t \;+\; \mathcal{O}(N^{-2m+1}). \tag{3.6}$$

(2.2.1) Assume that h has no jump at $t_1 = 0$ with sign change. Then there exists $\varepsilon > 0$ so that $h(t) \ge h_-$ or $h(t) \le -h_-$ for $t \in [-\varepsilon, \varepsilon]$. We restrict our attention to the case $h \ge h_-$. Since $0 < h_- p_s(t) \le f(t) \le h_+ p_s(t)$ $(t \in [-\varepsilon, \varepsilon])$ and p_s is monotone increasing on $(0, \pi)$, we obtain for $x(N) \in (0, \varepsilon) \cap I_N(f)$ and N sufficiently large that

$$\begin{aligned} \int_{-\varepsilon}^{\varepsilon} \frac{f(t)}{f(x(N))} K_N(t - x(N))\,\mathrm{d}t &\ge \int_{x(N)}^{\varepsilon} \frac{f(t)}{f(x(N))} K_N(t - x(N))\,\mathrm{d}t \\ &\ge \frac{h_-}{h_+} \int_{x(N)}^{\varepsilon} \frac{p_s(t)}{p_s(x(N))} K_N(t - x(N))\,\mathrm{d}t \\ &\ge \frac{h_-}{h_+} \int_{0}^{\varepsilon - x(N)} \frac{p_s(t)}{p_s(x(N))} K_N(t)\,\mathrm{d}t \;\ge\; c \end{aligned} \tag{3.7}$$

with a positive constant c independent of N. On the other hand, we have by definition of p_s and by assumption $s \le m-1$ that $f(x(N)) \ge h_- \tilde{c}\, N^{-2s} \ge h_- \tilde{c}\, N^{-2m+2}$. Then we obtain by (3.6) with $b = \varepsilon$ and (3.7) that for N large enough

$$\frac{(K_N * f)(x(N))}{f(x(N))} \ge \text{const}$$

with a positive constant const independent of N.

The proof for $x(N) \in (-\varepsilon, 0) \cap I_N(f)$ follows the same lines.

(2.2.2) Finally, we assume that h has a jump at $t_1 = 0$ with $\operatorname{sgn} h(0+0) \ne \operatorname{sgn} h(0-0)$. Without loss of generality let $h(0+0) > 0$. Then, by assumption on f, there exists $\varepsilon_1 > 0$ so that $h(t) = -h(-t)$ for $t \in [0, \varepsilon_1]$. Thus,

$$\int_{-\varepsilon_1}^{\varepsilon_1} f(t) K_N(x-t)\,\mathrm{d}t = \int_{0}^{\varepsilon_1} f(t)(K_N(t-x) - K_N(t+x))\,\mathrm{d}t. \tag{3.8}$$

We consider points of the form

$$y = y_k(N) := \frac{2\pi m}{N\gamma}\, k \qquad (k \in \mathbb{N})$$

with $\lim_{N\to\infty} y_k(N) = 0$, where $\gamma := mn/N$ in case of Jackson kernels and $\gamma := 1$ in case of B-spline kernels. Then we have for $t \in [0, \varepsilon_1]$ that

$$\mathcal{J}_{m,N}(t-y) - \mathcal{J}_{m,N}(t+y) = \lambda_{m,N} \left(\left(\frac{\sin(nt/2)}{\sin((t-y)/2)} \right)^{2m} - \left(\frac{\sin(nt/2)}{\sin((t+y)/2)} \right)^{2m} \right), \tag{3.9}$$

and consequently, for sufficiently small ε_1 and y, since sin is odd and monotone increasing on $(0, \pi/2)$ we have that

$$\mathcal{J}_{m,N}(t-y) - \mathcal{J}_{m,N}(t+y) > 0 \quad \text{for all } t \in (0, \varepsilon_1).$$

Further, by definition of the B-spline kernels

$$\mathcal{B}_{m,N}(t-y) - \mathcal{B}_{m,N}(t+y) = \mathcal{B}^0_{m,N}(t-y) - \mathcal{B}^0_{m,N}(t+y) \ + \ \mathcal{O}(N^{-2m+1}),$$

where $\mathcal{B}^0_{m,N}(t) := \frac{N}{m} \frac{1}{M_{2m}(0)} \left(\operatorname{sinc}\left(\frac{N}{m}\frac{t}{2}\right)\right)^{2m}$, and similarly as in (3.9) we see that

$$\mathcal{B}^0_{m,N}(t-y) - \mathcal{B}^0_{m,N}(t+y) > 0 \quad \text{for all } t \in (0, \varepsilon_1).$$

By assumption h does not change the sign in $(0, \varepsilon_1)$. Then we obtain by (3.8), monotonicity of p_s in $(0, \pi)$ and $m \geq s+1$ that

$$\int_{-\varepsilon_1}^{\varepsilon_1} \frac{f(t)}{f(y)} K_N(y-t)\,\mathrm{d}t \geq \frac{h_-}{h_+} \int_y^{\varepsilon_1} K^0_N(t-y) - K^0_N(t+y)\,\mathrm{d}t \ + \ \mathcal{O}(N^{-1}), \tag{3.10}$$

where $K^0_N \in \{\mathcal{J}_{m,N}, \mathcal{B}^0_{m,N}\}$. Set $w = w(N) := \frac{2\pi m}{N\gamma}$. Then $y_k = y_k(N) = w\,k$ and there exist $r = r(N) \in \mathbb{N}$ $(r > k)$ so that $\varepsilon_1 = w\,r + \tilde{\varepsilon}_1$, where $0 \leq \tilde{\varepsilon}_1 = \tilde{\varepsilon}_1(N) < w$. Now it follows that

$$\begin{aligned}
\int_{y_k}^{wr} K^0_N(t-y_k) - K^0_N(t+y_k)\,\mathrm{d}t &= \sum_{l=0}^{r-k-1} \int_{y_k+wl}^{y_k+w(l+1)} K^0_N(t-y_k) - K^0_N(t+y_k)\,\mathrm{d}t \\
&= \sum_{l=0}^{2k-1} \int_{wl}^{w(l+1)} K^0_N(t)\,\mathrm{d}t \ - \ \sum_{l=r-k}^{r+k-1} \int_{wl}^{w(l+1)} K^0_N(t)\,\mathrm{d}t \\
&\geq \int_0^w K^0_N(t)\,\mathrm{d}t \ - \ \int_{\varepsilon_1+y_k-w}^{\varepsilon_1+y_k} K^0_N(t)\,\mathrm{d}t,
\end{aligned}$$

and further by (3.5) and since $\lim_{N\to\infty} y_k = 0$,

$$\int_{y_k}^{\varepsilon_1} K^0_N(t-y_k) - K^0_N(t+y_k)\,\mathrm{d}t \geq \int_0^w K^0_N(t)\,\mathrm{d}t \ + \ \mathcal{O}(N^{-2m+1}).$$

Straightforward computation yields

$$\int_0^{2\pi m/(N\gamma)} K^0_N(t)\,\mathrm{d}t \ \geq \ \text{const} \int_0^{\pi} \left(\frac{\sin u}{u}\right)^{2m} \mathrm{d}u \ \geq \ \text{const}.$$

Hence we get for N large enough that

$$\int_{y_k}^{\varepsilon_1} K^0_N(t-y_k) - K^0_N(t+y_k)\,\mathrm{d}t \ \geq \ \text{const}$$

and by (3.10) that

$$\int_{-\varepsilon_1}^{\varepsilon_1} \frac{f(t)}{f(y_k)} K_N(y_k-t)\,\mathrm{d}t \ \geq \ \text{const} \tag{3.11}$$

with positive constants const independent of N.

Now we consider $x(N) \in I_N(f)$ with $y_k(N) \le x(N) < y_{k+1}(N)$.
Let $z(N) := x(N) - y_k(N) > 0$. Then

$$\begin{aligned}\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - x(N))\,\mathrm{d}t &= \int_{-\varepsilon_1 - z(N)}^{\varepsilon_1 - z(N)} f(t + z(N))\, K_N(t - y_k(N))\,\mathrm{d}t \\ &= \int_{-\varepsilon_1}^{\varepsilon_1 - z(N)} f(t + z(N))\, K_N(t - y_k(N))\,\mathrm{d}t \\ &\quad + \int_{-\varepsilon_1 - z(N)}^{-\varepsilon_1} f(t + z(N))\, K_N(t - y_k(N))\,\mathrm{d}t,\end{aligned}$$

and since f is by assumption monotone increasing on $[-\varepsilon_1, \varepsilon_1]$

$$\begin{aligned}\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - x(N))\,\mathrm{d}t &\ge \int_{-\varepsilon_1}^{\varepsilon_1 - z(N)} f(t)\, K_N(t - y_k(N))\,\mathrm{d}t \\ &\quad + \int_{-\varepsilon_1}^{-\varepsilon_1 + z(N)} f(t)\, K_N(t - x(N))\,\mathrm{d}t \\ &= \int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - y_k(N))\,\mathrm{d}t \\ &\quad + \int_{-\varepsilon_1}^{-\varepsilon_1 + z(N)} f(t)\, K_N(t - x(N))\,\mathrm{d}t \\ &\quad - \int_{\varepsilon_1 - z(N)}^{\varepsilon_1} f(t)\, K_N(t - y_k(N))\,\mathrm{d}t,\end{aligned}$$

and by (3.5) and since f is bounded

$$\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - x(N))\,\mathrm{d}t \;\ge\; \int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - y_k(N))\,\mathrm{d}t \;+\; \mathcal{O}(N^{-2m+1}). \tag{3.12}$$

By assumption $x(N) = \zeta\, y_k(N)$ $(0 < \zeta < 2)$. Thus

$$\frac{\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - x(N))\,\mathrm{d}t}{f(x(N))} \;\ge\; \text{const}\, \frac{\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - y_k(N))\,\mathrm{d}t}{f(y_k(N))},$$

and since $f(y_k(N) \ge \text{const}\, N^{-2s}$ and $m \ge s + 1$ we obtain by (3.12), (3.11) that for N large enough

$$\int_{-\varepsilon_1}^{\varepsilon_1} f(t)\, K_N(t - x(N))\,\mathrm{d}t \,/\, f(x(N)) \;\ge\; \text{const}$$

with a nonnegative constant const independent of N. Finally, we use (3.6) with $b = \varepsilon_1$ and again $m \ge s + 1$ to finish the proof. □

To show our main result we also need the following lemma.

LEMMA 3.2. *Let $\boldsymbol{A} \in \mathbb{C}^{N,N}$ be a Hermitian positive definite matrix having $N - n_1$ eigenvalues in $[a_-, a_+]$, where $0 < a_- \le a_+ < \infty$. Let $\boldsymbol{B} \in \mathbb{C}^{N,N}$ be a Hermitian matrix with $N - n_2$ singular values in $[b_-, b_+]$, where $0 < b_- \le b_+ < \infty$. Then at least $N - 4n_1 - n_2$ eigenvalues of $\boldsymbol{A}\,\boldsymbol{B}$ are contained in $[-a_+b_+, -a_-b_-] \cup [a_-b_-, a_+b_+]$.*

Proof. (1) Assume first that $n_1 = 0$, i.e., $\boldsymbol{A}$ has only eigenvalues in $[a_-, a_+]$. Let $\lambda_j(\boldsymbol{B})$ denote the jth eigenvalue of the matrix $\boldsymbol{B}$. We consider the eigenvalues of

$\boldsymbol{B} - t\boldsymbol{A}^{-1}$ with respect to $t \in \boldsymbol{R}$. By Weyl's interlacing theorem (see [13, p. 184]) we obtain for $t \geq 0$ that

$$\lambda_j(\boldsymbol{B}) - \frac{t}{a_-} \leq \lambda_j\left(\boldsymbol{B} - t\boldsymbol{A}^{-1}\right) \leq \lambda_j(\boldsymbol{B}) - \frac{t}{a_+} \tag{3.13}$$

and for $t < 0$ that

$$\lambda_j(\boldsymbol{B}) - \frac{t}{a_+} \leq \lambda_j\left(\boldsymbol{B} - t\boldsymbol{A}^{-1}\right) \leq \lambda_j(\boldsymbol{B}) - \frac{t}{a_-}. \tag{3.14}$$

Let $\lambda_j(\boldsymbol{B}) \in [-b_+, -b_-]$. Then we obtain by (3.13) and (3.14) that $\lambda_j\left(\boldsymbol{B} - t\boldsymbol{A}^{-1}\right) < 0$ for all $t > -a_-b_-$. On the other hand, we see by (3.13) and (3.14) that $\lambda_j\left(\boldsymbol{B} - t\boldsymbol{A}^{-1}\right) > 0$ for all $t < -a_+b_+$. Thus, since $\lambda_j\left(\boldsymbol{B} - t\boldsymbol{A}^{-1}\right) = \lambda_j(t)$ is a continuous function in $t \in \boldsymbol{R}$, there exists $t_j \in [-a_+b_+, -a_-b_-]$ such that $\lambda_j\left(\boldsymbol{B} - t_j\boldsymbol{A}^{-1}\right) = 0$. This implies that $t_j \in [-a_+b_+, -a_-b_-]$ is an eigenvalue of $\boldsymbol{AB}$. Consequently, every $\lambda_j(\boldsymbol{B}) \in [-b_+, -b_-]$ corresponds to an eigenvalue $t_j \in [-a_+b_+, -a_-b_-]$ of $\boldsymbol{AB}$. (Eigenvalues are called with multiplicities.)

The examination of $\lambda_j(\boldsymbol{B}) \in [a_-b_-, a_+b_+]$ follows the same lines.

In summary, $N - n_2$ eigenvalues of $\boldsymbol{AB}$ are contained in $[-a_+b_+, -a_-b_-] \cup [a_-b_-, a_+b_+]$.

(2) Let n_1 eigenvalues of $\boldsymbol{A}$ be outside $[a_-, a_+]$. Then, since $\boldsymbol{A}$ is positive definite, the matrix can be split as

$$\boldsymbol{A}^{1/2} = \tilde{\boldsymbol{A}}^{1/2} + \boldsymbol{R}(n_1)\,, \tag{3.15}$$

where $\tilde{\boldsymbol{A}}^{1/2}$ is Hermitian with all eigenvalues in $[a_-^{1/2}, a_+^{1/2}]$ and $\boldsymbol{R}(n_1)$ is a Hermitian matrix of rank n_1. The eigenvalues of $\boldsymbol{AB}$ coincide with the eigenvalues of $\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2}$. Hence it remains to show that at most $4n_1 + n_2$ singular values of $\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2}$ are not contained in $[a_-b_-, a_+b_+]$. By (3.15) we have

$$\begin{aligned} \boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2} &= \tilde{\boldsymbol{A}}^{1/2}\boldsymbol{B}\tilde{\boldsymbol{A}}^{1/2} + \boldsymbol{R}(2n_1)\,, \\ \left(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2}\right)^2 &= \left(\tilde{\boldsymbol{A}}^{1/2}\boldsymbol{B}\tilde{\boldsymbol{A}}^{1/2}\right)^2 + \boldsymbol{R}(4n_1). \end{aligned} \tag{3.16}$$

By (1) all but n_2 singular values of $\tilde{\boldsymbol{A}}^{1/2}\boldsymbol{B}\tilde{\boldsymbol{A}}^{1/2}$ are contained in $[a_-b_-, a_+b_+]$. Then (3.16) and Weyl's interlacing theorem yield the assertion. ∎

THEOREM 3.3. *Let $f = p_s h \in \mathcal{L}_{2\pi}$ be given by* (1.2)–(1.4). *Assume that for all t_j $(j \in \{1, \dots, \mu\})$ with $t_j = \xi_k$ for some $k \in \{1, \dots, \nu\}$ and $\operatorname{sgn} h(\xi_k + 0) \neq \operatorname{sgn} h(\xi_k - 0)$ there exists a neighborhood $[t_j - \varepsilon_j, t_j + \varepsilon_j]$ $(\varepsilon_j > 0)$ of t_j so that f is a monotone function in this neighborhood and moreover $f(t_j - t) = -f(t_j + t)$ $(0 \leq t \leq \varepsilon_j)$. Let $K_N = K_{m,N}$ be given by* (3.2) *or* (3.3), *where*

$$m \geq \sigma + 1.$$

By α, β we denote the constants from Theorem 3.1.

Then, for any $\varepsilon > 0$ and sufficiently large N, except for $\mathcal{O}(\log N)$ singular values, all singular values of $\boldsymbol{M}_N(|K_N * f|)^{-\frac{1}{2}}\boldsymbol{A}_N(f)\boldsymbol{M}_N(|K_N * f|)^{-\frac{1}{2}}$ are contained in $[\alpha - \varepsilon, \beta + \varepsilon]$.

Proof. Let $\boldsymbol{B}_N(f)$ be defined by (2.11). Then we obtain by (2.12) that

$$\begin{aligned}
(3.17)\quad & \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}}\, \boldsymbol{A}_N(f)\, \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}} \\
&= \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}}\, \boldsymbol{M}_{N,f}(p)\, \boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}}\ \boldsymbol{B}_N(f) \\
&\quad \cdot \boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}}\, \boldsymbol{M}_{N,f}(\bar{p})\ \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}} \\
&= \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}}\, \boldsymbol{M}_{N,f}(p)\, \boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}} \\
&\quad \cdot \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{A}_N(h) \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \\
(3.18)\quad &\quad \cdot \boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}}\, \boldsymbol{M}_{N,f}(\bar{p})\ \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}} \ +\ \boldsymbol{R}(4s+2m).
\end{aligned}$$

The distribution of the eigenvalues of $\boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}} \boldsymbol{A}_N(h) \boldsymbol{M}_{N,f}(|h|)^{-\frac{1}{2}}$ is known by Lemma 2.2. It remains to examine the eigenvalues of the Hermitian positive definite matrix

$$\boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}}\, \boldsymbol{M}_{N,f}(\bar{p})\, \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-1}\, \boldsymbol{M}_{N,f}(p)\, \boldsymbol{M}_{N,f}(|h|)^{\frac{1}{2}}.$$

These eigenvalues coincide with the reciprocal eigenvalues of $\boldsymbol{M}_{N,f}(|f|)^{-1}\, \boldsymbol{M}_{N,K_N*f}(|K_N * f|)$. By definition of $\boldsymbol{M}_{N,g}$ and since K_N is a reproducing kernel, except for $\mathcal{O}(1)$ eigenvalues, all eigenvalues of $\boldsymbol{M}_{N,f}(|f|)^{-1}\, \boldsymbol{M}_{N,K_N*f}(|K_N * f|)$ are given by $|(K_N * f)(2\pi l/N)|/|f(2\pi l/N)|$ $(l \in I_N(f))$. Thus, by Theorem 3.1, for $N \to \infty$ only $\mathcal{O}(1)$ eigenvalues of $\boldsymbol{M}_{N,f}(|f|)\, \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-1}$ are not contained in $[\alpha, \beta]$. Consequently, by (3.18), Lemma 2.2, Lemma 3.2, and Weyl's interlacing theorem at most $\mathcal{O}(\log N)$ singular values of $\boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}} \boldsymbol{A}_N(f) \boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-\frac{1}{2}}$ are not contained in $[\alpha - \varepsilon, \beta + \varepsilon]$. ∎

4. Trigonometric preconditioners. In addition to section 2, we suppose that the Toeplitz matrices $\boldsymbol{A}_N \in \mathbb{R}^{N,N}$ are symmetric, i.e., the generating function $f \in \mathcal{L}_{2\pi}$ is even. This suggests the application of so-called trigonometric preconditioners. Note that in the symmetric case the multiplication of a vector with $\boldsymbol{A}_N$ can be realized using *fast trigonometric transforms* instead of fast Fourier transforms (see [15]). In this way complex arithmetic can be completely avoided in the iterative solution of (1.1). This is one of the reasons to look for preconditioners which can be diagonalized by trigonometric matrices corresponding to fast trigonometric transforms instead of the Fourier matrix $\boldsymbol{F}_N$.

In practice, four discrete sine transforms (DST I–IV) and four discrete cosine transforms (DCT I–IV) were used (see [25]). Any of these eight trigonometric transforms can be realized with $\mathcal{O}(N \log N)$ arithmetical operations. Likewise, we can define preconditioners with respect to any of these transforms.

In this paper, we restrict our attention to the so-called discrete cosine transform of type II (DCT-II) and discrete sine transform of type II (DST-II), which are determined by the following transform matrices:

$$\begin{aligned}
\text{DCT–II}: \quad \boldsymbol{C}_N^{II} &:= \left(\frac{2}{N}\right)^{1/2} \left(\epsilon_j^N \cos \frac{j(2k+1)\pi}{2N}\right)_{j,k=0}^{N-1} \in \mathbb{R}^{N,N}, \\
\text{DST–II}: \quad \boldsymbol{S}_N^{II} &:= \left(\frac{2}{N}\right)^{1/2} \left(\epsilon_{j+1}^N \sin \frac{(j+1)(2k+1)\pi}{2N}\right)_{j,k=0}^{N-1} \in \mathbb{R}^{N,N},
\end{aligned}$$

where $\epsilon_k^N := 2^{-1/2} (k = 0, N)$ and $\epsilon_k^N := 1$ $(k = 1, \ldots, N-1)$. We propose the

preconditioners

$$\text{DCT}-\text{II}: \qquad \boldsymbol{M}_{N,f}(|f|,\boldsymbol{C}_N^{II}) \;:=\; (\boldsymbol{C}_N^{II})'\,\text{diag}\,(|f(\tilde{x}_{N,l})|)_{l=0}^{N-1}\;\boldsymbol{C}_N^{II},$$

$$\text{DST}-\text{II}: \qquad \boldsymbol{M}_{N,f}(|f|,\boldsymbol{S}_N^{II}) \;:=\; (\boldsymbol{S}_N^{II})'\,\text{diag}\,(|f(\tilde{x}_{N,l})|)_{l=1}^{N}\;\boldsymbol{S}_N^{II},$$

where

$$\tilde{x}_{N,l} := \begin{cases} \frac{l\pi}{N} & \text{if} \quad f\left(\frac{l\pi}{N}\right) \neq 0, \\ \frac{\tilde{l}\pi}{N} & \text{otherwise} \end{cases}$$

and where $\tilde{l} \in \{0,\dots,N-1\}$ is the next higher index to l such that $|f(\tilde{x}_{N,l})| > 0$. (See [16].)

Then we can prove in a completely similar way as in section 2 that for any $\varepsilon > 0$ and sufficiently large N except for $\mathcal{O}(\log N)$ singular values, all singular values of

$$\boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{-\frac{1}{2}}\boldsymbol{A}_N(f)\boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{-\frac{1}{2}} \quad (\boldsymbol{O} \in \{\boldsymbol{S}_N^{II},\boldsymbol{C}_N^{II}\})$$

are contained in $[1-\varepsilon, 1+\varepsilon]$.

5. Convergence of preconditioned MINRES. In order to prescribe the convergence behavior of preconditioned MINRES with our preconditioners of the previous sections, we have to estimate the smaller outliers for increasing N.

LEMMA 5.1. *Let $f \in \mathcal{L}_{2\pi}$ be defined by (1.2)–(1.4). Assume that $\kappa_2(\boldsymbol{A}_N(f)) = \mathcal{O}(N^\alpha)$ $(\alpha > 0)$. Then the smallest absolute values of the eigenvalues of*

$$\boldsymbol{M}_{N,f}(|f|)^{-1}\,\boldsymbol{A}_N(f)$$

and

$$\boldsymbol{M}_{N,K_N*f}(|K_N * f|)^{-1}\,\boldsymbol{A}_N(f)$$

behave for $N \to \infty$ as $\mathcal{O}(N^{-\alpha})$.

Proof. Since

$$\|\boldsymbol{A}_N(f)^{-1}\boldsymbol{M}_{N,f}(|f|)\|_2 \leq \frac{\|\boldsymbol{M}_{N,f}(|f|)\|_2}{\|\boldsymbol{A}_N(f)\|_2}\,\kappa_2(\boldsymbol{A}_N(f)),$$

$$\|\boldsymbol{A}_N(f)^{-1}\boldsymbol{M}_{N,K_N*f}(|K_N * f|)\|_2 \leq \frac{\|\boldsymbol{M}_{N,K_N*f}(|K_N * f|)\|_2}{\|\boldsymbol{A}_N(f)\|_2}\,\kappa_2(\boldsymbol{A}_N(f)),$$

and both $\|\boldsymbol{M}_{N,f}(|f|)\|_2$ and $\|\boldsymbol{M}_{N,K_N*f}(|K_N * f|)\|_2$ are restricted from above, it remains to show that there exists a constant $c > 0$ independent of N so that

$$\|\boldsymbol{A}_N(f)\|_2 \;>\; c.$$

The above inequality follows immediately from the fact that the singular values of $\boldsymbol{A}_N(f)$ are distributed as $|f|$ (see [14, 23]). ∎

We want to combine our knowledge of the distribution of the eigenvalues of our preconditioned matrices with results concerning the convergence of MINRES.

THEOREM 5.2. *Let $\boldsymbol{A} \in \mathbb{C}^{N,N}$ be a Hermitian matrix with p and q isolated large and small singular values, respectively:*

$$0 < \sigma_1 \le \sigma_2 \le \cdots \le \sigma_q < a \le \sigma_{q+1} \le \cdots \sigma_{N-p} \le b < \sigma_{N-p+1} \le \sigma_{N-p+2} \le \cdots \le \sigma_N \quad (0 < a \le b < \infty).$$

Let $\nu(k) := 0$ if $k - p - q \equiv 0 \bmod 2$ and $\nu(k) := 1$ otherwise. Then MINRES requires for the solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

$$k \le 2 \left(\ln \frac{2}{\tau} + \sum_{k=1}^{q} \ln \left(1 + \frac{b}{\sigma_k} \right) + p \ln 2 \right) / \left(\ln \frac{1 + (\frac{a}{b})}{1 - (\frac{a}{b})} \right) + p + q + \nu(k)$$

iteration steps to achieve precision τ, i.e., $\frac{||\boldsymbol{r}^{(k)}||_2}{||\boldsymbol{r}^{(0)}||_2} \le \tau$, where $\boldsymbol{r}^{(k)} := \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}$ and $x^{(k)}$ is the kth iterate.

The theorem can be proved by using the same technique as in [1, pp. 569–573]. Namely, based on the known estimate

$$\frac{||\boldsymbol{r}^{(k)}||_2}{||\boldsymbol{r}^{(0)}||_2} \le \min_{p_k \in \Pi_k^0} \max_{\lambda_j} |p_k(\lambda_j)|,$$

where Π_k^0 denotes the space of polynomials of degree $\le k$ with $p_k(0) = 1$ and λ_j are the eigenvalues of $\boldsymbol{A}$, we choose p_k as the product of the linear polynomials passing through the $p + q$ outliers and the modified Chebyshev polynomials

$$T_{\lfloor (k-p-q)/2 \rfloor} \left(1 + 2\,\frac{a^2 - x^2}{b^2 - a^2} \right) / T_{\lfloor (k-p-q)/2 \rfloor} \left(1 + 2\,\frac{a^2}{b^2 - a^2} \right).$$

The above summand $p \ln 2$ can be further reduced if we use polynomials of higher degree for the larger outliers.

Note that a similar estimate can be given for the CG method applied to the normal equation $\boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^*\boldsymbol{b}$. Here we need

$$k \le \left(\ln \frac{2}{\tau} + \sum_{k=1}^{q} \ln \left(\frac{b}{\sigma_k^2} \right) \right) / \left(\ln \frac{1 + (\frac{a}{b})}{1 - (\frac{a}{b})} \right) + p + q$$

iteration steps to achieve precision $\frac{\|\boldsymbol{e}^{(k)}\|_{\boldsymbol{A}}}{||\boldsymbol{e}^{(0)}||_{\boldsymbol{A}}} \le \tau$, where $\boldsymbol{e}^{(k)} := \boldsymbol{x}_* - \boldsymbol{x}^{(k)}$. Note that the latter method requires two matrix-vector multiplications in each iteration step.

By Theorem 2.4, Theorem 3.3, and Lemma 5.1 our preconditioned MINRES with both preconditioners $\boldsymbol{M}_{N,f}(|f|)$ and $\boldsymbol{M}_{N,K_N * f}(|K_N * f|)$ produces a solution of (1.1) of prescribed precision in $\mathcal{O}(\log N)$ iteration steps and with $\mathcal{O}(N \log^2 N)$ arithmetical operations. The same holds for preconditioned CG applied to the normal equation.

6. Numerical results. In this section, we test our circulant and trigonometric preconditioners in connection with different iterative methods on a SGI O2 work station. As transform length we use $N = 2^n$, as right-hand side $\boldsymbol{b}$ of (1.1) we use the vector consisting of N entries "1," and as start vector we use the zero vector.

We begin with a comparison of MINRES applied to

$$\boldsymbol{M}_{N,f}(|f|, \boldsymbol{O})^{-1}\, \boldsymbol{A}_N(f)\, \boldsymbol{x} = \boldsymbol{M}_{N,f}(|f|, \boldsymbol{O})^{-1}\, \boldsymbol{b}, \tag{6.1}$$

TABLE 1
$f(t) = h_1(t)\, t^2 \quad h_1(t) = (t^2+1)\ sgn\,(t) \quad (t \in [-\pi, \pi))$.

Method	$\boldsymbol{M}_{N,f}$	4	5	6	7	8	9	10
MINRES	$\boldsymbol{I}_N$	23	71	277	*	*	*	*
MINRES	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{F}_N)$	15	17	17	19	21	23	23
MINRES	$\boldsymbol{M}_{N,\mathcal{F}_N * f}(\lvert \mathcal{F}_N * f\rvert, \boldsymbol{F}_N)$	19	31	35	41	43	47	51
MINRES	$\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{F}_N)$	19	23	23	25	25	27	29
CGNE	$\boldsymbol{I}_N$	11	37	164	*	*	*	*
CGNE	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{F}_N)$	8	8	9	9	9	10	10

where $\boldsymbol{O} \in \{\boldsymbol{F}_N, \boldsymbol{C}_N^{II}, \boldsymbol{S}_N^{II}\}$ and CGNE (Craig's method) (cf. [18, p. 239]) applied to

(6.2)
$$(\boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{-\frac{1}{2}}\, \boldsymbol{A}_N(f)\, \boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{-\frac{1}{2}})\, (\boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{\frac{1}{2}}\, \boldsymbol{x}) = \boldsymbol{M}_{N,f}(|f|,\boldsymbol{O})^{-\frac{1}{2}}\, \boldsymbol{b}.$$

For both algorithms we have used MATLAB implementations of Fischer (see also [12]). In particular, his implementation of preconditioned MINRES avoids the splitting (6.2).

In order to make the following computations with MINRES and CGNE comparable, we have stopped both computations if

$$||\boldsymbol{b} - \boldsymbol{A}_N \boldsymbol{x}^{(k)}||_2 \,/\, ||\boldsymbol{b}||_2 < 10^{-7}.$$

Example 1. We begin with Hermitian Toeplitz matrices $\boldsymbol{A}_N(f)$ arising from the generating function

$$f_1(t) = h_1(t)\, t^2 \quad \text{with } h_1(t) = (t^2+1)\ \text{sgn}\,(t) \quad (t \in [-\pi, \pi)).$$

Table 1 presents the number of iterations for circulant preconditioners. The first row of the table contains the exponent n of the transform length $N = 2^n$. According to Theorem 2.4 and Theorem 5.2, the preconditioners $\boldsymbol{M}_N(|f|, \boldsymbol{F}_N)$ lead to very good results. As expected, the preconditioners $\boldsymbol{M}_{N,K_N * f}(|K_N * f|, \boldsymbol{F}_N)$ with the Fejér kernels $K_N = \mathcal{F}_N$ are not suitable for (1.1) (cf. also [17]), while the preconditioners with $K_N = \mathcal{B}_{2,N}$ do their job.

Further, CGNE needs half the number of iterations but twice the number of matrix-vector multiplications per iteration as MINRES needs. See also section 5.

Example 2. Next, we consider the symmetric Toeplitz matrices $\boldsymbol{A}_N(f)$ arising from the generating function

$$f_2(t) = h_2(t)\, (\cos(t+2)+1)\, (\cos(t-2)+1)$$

with

$$h_2(t) = \text{sgn}(t-\pi+2)\, \text{sgn}(t+\pi-2).$$

Table 2 presents the number of iterations for trigonometric preconditioners. The results are similar to those of Example 1, except that CGNE requires nearly the same number of iterations as MINRES.

TABLE 2
$f_2(t) = h_2(t)\,(\cos(t+2)+1)\,(\cos(t-2)+1) \quad (t \in [-\pi,\pi))$.

Method	$\boldsymbol{M}_N$	4	5	6	7	8	9	10
MINRES	$\boldsymbol{I}_N$	9	17	45	142	401	*	*
MINRES	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{C}_N^{II})$	8	9	10	11	14	13	16
MINRES	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{S}_N^{II})$	9	10	11	12	14	13	16
MINRES	$\boldsymbol{M}_{N,\mathcal{F}_N * f}(\lvert \mathcal{F}_N * f\rvert, \boldsymbol{C}_N^{II})$	10	15	20	26	30	39	53
MINRES	$\boldsymbol{M}_{N,\mathcal{F}_N * f}(\lvert \mathcal{F}_N * f\rvert, \boldsymbol{S}_N^{II})$	10	15	19	25	30	39	53
MINRES	$\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{C}_N^{II})$	9	15	17	16	20	18	18
MINRES	$\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{S}_N^{II})$	9	14	16	18	19	18	18
CGNE	$\boldsymbol{I}_N$	10	29	99	413	*	*	*
CGNE	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{C}_N^{II})$	7	9	11	11	17	16	17
CGNE	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{S}_N^{II})$	7	7	10	10	12	14	15

TABLE 3
$f_3(t) = ((\frac{t}{\pi})^2 - 1)^2 - 0.9 \quad (t \in [-\pi,\pi))$.

Method	$\boldsymbol{M}_N$	4	5	6	7	8	9	10
MINRES	$\boldsymbol{I}_N$	9	17	33	66	133	*	*
MINRES	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{C}_N^{II})$	6	7	7	8	7	7	7
MINRES	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{S}_N^{II})$	7	8	8	7	9	8	8
MINRES	$\boldsymbol{M}_{N,\mathcal{F}_N * f}(\lvert \mathcal{F}_N * f\rvert, \boldsymbol{C}_N^{II})$	8	11	15	17	16	17	17
MINRES	$\boldsymbol{M}_{N,\mathcal{F}_N * f}(\lvert \mathcal{F}_N * f\rvert, \boldsymbol{S}_N^{II})$	8	11	15	16	15	15	15
MINRES	$\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{C}_N^{II})$	8	10	10	11	9	7	7
MINRES	$\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{S}_N^{II})$	8	10	10	10	9	9	8
CGNE	$\boldsymbol{I}_N$	8	22	65	164	378	*	*
CGNE	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{C}_N^{II})$	5	6	6	8	6	5	6
CGNE	$\boldsymbol{M}_{N,f}(\lvert f\rvert, \boldsymbol{S}_N^{II})$	6	6	6	6	7	7	7

Example 3. Finally, we consider an example from [21, Table 4.9]:

$$f_3(t) = \left(\left(\frac{t}{\pi} \right)^2 - 1 \right)^2 - 0.9 \quad (t \in [-\pi,\pi)) \,.$$

This generating function doesn't fit into our setting (1.2). However, our preconditioning technique leads to very good practical results; see Table 3. For $N = 512$ the preconditioned MINRES with our preconditioner $\boldsymbol{M}_{N,\mathcal{B}_{2,N} * f}(\lvert \mathcal{B}_{2,N} * f\rvert, \boldsymbol{C}_N^{II})$ requires only 7 iterations. For comparison, PCG with the banded Toeplitz preconditioner of *bandwidth* 37 suggested in [21] requires 2 iterations. The theoretical justification of these numerical results is part of further research.

Acknowledgments. We wish to thank B. Fischer for the MATLAB implementations of MINRES and CGNE. Many thanks to the referees for their valuable com-

ments, in particular for pointing out various references to us.

REFERENCES

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1996.
[2] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods,* 2*nd Edition*, SIAM, Philadelphia, PA, 1994.
[3] F. Di Benedetto, G. Fiorentino, and S. Serra, *C. G. preconditioning for Toeplitz matrices*, Comput. Math. Appl., 25 (1993), pp. 35–45.
[4] A. Böttcher and S. M. Grudsky, *Toeplitz band matrices with exponentially growing condition numbers*, Electron. J. Linear Algebra, 5 (1999), pp. 104–125.
[5] R. Chan, *Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions*, IMA J. Numer. Anal., 11 (1991), pp. 333–345.
[6] R. H. Chan and W.-K. Ching, *Toeplitz–circulant preconditioners for Toeplitz systems and their applications to queueing networks with batch arrivals*, SIAM J. Sci. Comput., 17 (1996), pp. 762–772.
[7] R. H. Chan and M. K. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.
[8] R. H. Chan, D. Potts, and G. Steidl, *Preconditioners for Non-Hermitian Toeplitz Systems*, Numer. Linear Algebra Appl., to appear.
[9] R. H. Chan, T. Tso, and H. Sun, *Circulant preconditioners from B-splines*, in Proceedings of the SPIE Symposium, Advanced Signal Processing: Algorithms, Architectures, and Implementations VII, Vol. 3162, San Diego, CA, 1997, F. Luk, ed., Society of Photo-optical Instrumentation Engineers, 1997, pp. 338–347.
[10] R. H. Chan, M. Yip, and M. Ng, *The Best Circulant Preconditioners for Hermitian Toeplitz Matrices*, SIAM J. Numer. Anal., 38 (2000), pp. 876–896.
[11] R. DeVore and G. Lorentz, *Constructive Approximation*, Springer–Verlag, Berlin, Germany, 1993.
[12] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley–Teubner, New York, 1996.
[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[14] S. V. Parter, *On the distribution of singular values of Toeplitz matrices*, Linear Algebra Appl., 80 (1986), pp. 115–130.
[15] D. Potts and G. Steidl, *Optimal trigonometric preconditioners for nonsymmetric Toeplitz systems*, Linear Algebra Appl., 281 (1998), pp. 265–292.
[16] D. Potts and G. Steidl, *Preconditioners for ill–conditioned Toeplitz matrices*, BIT, 39 (1999), pp. 513–533.
[17] D. Potts and G. Steidl, *Preconditioners for Ill–Conditioned Toeplitz Matrices Constructed from Positive Kernels*, SIAM J. Sci. Comput., to appear.
[18] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishers, Boston, MA, 1996.
[19] S. Serra, *New PCG based algorithms for the solution of Hermitian Toeplitz systems*, Calcolo, 32 (1995), pp. 153–176.
[20] S. Serra, *Preconditioning strategies for Hermitian Toeplitz systems with nondefinite generating functions*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 1007–1019.
[21] S. Serra Capizzano, *Toeplitz preconditioners constructed from linear approximation processes*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 446–465.
[22] S. Serra, *How to choose the best iterative strategy for symmetric Toeplitz systems*, SIAM J. Numer. Anal., 36 (1999), pp. 1078–1103.
[23] E. E. Tyrtyshnikov, *A unifying approach to some old and new theorems on distribution and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.
[24] E. E. Tyrtyshnikov, A. Yeremin, and N. Zamarashkin, *Clusters – preconditioners – convergence*, Linear Algebra Appl., 263 (1997), pp. 25–48.
[25] Z. Wang, *Fast algorithms for the discrete W transform and for the discrete Fourier transform*, IEEE Trans. Acoust. Speech Signal Process, 32 (1984), pp. 803–816.
[26] M.-C. Yeung and R. H. Chan, *Circulant preconditioners for Toeplitz matrices with piecewise continuous generating functions*, Math. Comp., 61 (1993), pp. 701–718.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 666–681

AN ORTHOGONALLY BASED PIVOTING TRANSFORMATION OF MATRICES AND SOME APPLICATIONS*

ENRIQUE CASTILLO†, ANGEL COBO†, FRANCISCO JUBETE†, ROSA EVA PRUNEDA†, AND CARMEN CASTILLO†

Abstract. In this paper we discuss the power of a pivoting transformation introduced by Castillo, Cobo, Jubete, and Pruneda [*Orthogonal Sets and Polar Methods in Linear Algebra: Applications to Matrix Calculations, Systems of Equations and Inequalities, and Linear Programming*, John Wiley, New York, 1999] and its multiple applications. The meaning of each sequential tableau appearing during the pivoting process is interpreted. It is shown that each tableau of the process corresponds to the inverse of a row modified matrix and contains the generators of the linear subspace orthogonal to a set of vectors and its complement. This transformation, which is based on the orthogonality concept, allows us to solve many problems of linear algebra, such as calculating the inverse and the determinant of a matrix, updating the inverse or the determinant of a matrix after changing a row (column), determining the rank of a matrix, determining whether or not a set of vectors is linearly independent, obtaining the intersection of two linear subspaces, solving systems of linear equations, etc. When the process is applied to inverting a matrix and calculating its determinant, not only is the inverse of the final matrix obtained, but also the inverses and the determinants of all its block main diagonal matrices, all without extra computations.

Key words. compatibility, determinant, intersection of linear subspaces, linear systems of equations, rank of a matrix, updating inverses

AMS subject classifications. 15A03, 15A06, 15A09, 15A15

PII. S0895479898349720

1. Introduction.

Castillo, Cobo, Fernández-Canteli, Jubete, and Pruneda [2] and Castillo, Cobo, Jubete, and Pruneda [3] have recently introduced a pivoting transformation of a matrix that has important properties and has been shown to be very useful to solve a long list of problems in linear algebra. The aim of this paper is to show the power of this transformation, clarify the meaning of the partial results obtained during the computationl process, and illustrate the wide range of applications of this transformation to solve common problems in linear algebra, such as calculating inverses of matrices, determinants or ranks, solving systems of linear equations, etc.

The reader interested in a classical treatment of these problems can, for example, consult the works of Burden and Faires [1], Golub and Van Loan [5], Gill et al. [6], and Press et al. [8].

The new methods arising from this transformation have complexity identical to that associated with the Gauss elimination method (see Castillo, Cobo, Jubete, and Pruneda [3]). However, they are specially suitable for updating solutions when changes in rows, columns, or variables are done. In fact, when changing a row, column, or variable, a single step of the process allows one to obtain (update) the new solution without the need to start again from scratch. For example, updating the inverse of

*Received by the editors December 22, 1998; accepted for publication (in revised form) by M. Chu May 30, 2000; published electronically October 25, 2000. This work was partially supported by Iberdrola, the Leonardo Torres Quevedo Foundation of the University of Cantabria, and Dirección General de Investigación Científica y Técnica (DGICYT) (project TIC96-0580).

http://www.siam.org/journals/simax/22-3/34972.html

†Department of Applied Mathematics and Computational Sciences, University of Cantabria, 39005 Santander, Spain (castie@unican.es, acobo@besaya.unican.es, rpruneda@ccp-cr.uclm.es, mcastill@platon.ugr.es).

an $n \times n$ matrix when a row is changed requires one instead of n steps, a drastic reduction in computational power.

In this paper we introduce the pivoting transformation and its applications only from the algebraic point of view. Discussing the numerical properties and performance of this method with respect to stability, ill conditioning, etc., which must be done carefully and taking into account its applications (see Demmel [4] and Higham [7]), will be the aim of another paper.

The paper is structured as follows. In section 2 the pivoting transformation is introduced. In section 3 its main properties are discussed. In section 4 an orthogonalization algorithm is derived. In section 5 some applications are given and illustrated with examples. Finally, some conclusions are given in section 6.

2. Pivoting transformation. The main tool to be used in this paper consists of the so-called *pivoting transformation*, which transforms a set of vectors $\mathbf{V}^j = \{\mathbf{v}_1^j, \ldots, \mathbf{v}_n^j\}$ into another set of vectors $\mathbf{V}^{j+1} = \{\mathbf{v}_1^{j+1}, \ldots, \mathbf{v}_n^{j+1}\}$ by

$$\mathbf{v}_k^{j+1} = \begin{cases} \mathbf{v}_k^j / t_j^j & \text{if } k = j, \\ \mathbf{v}_k^j - \dfrac{t_k^j}{t_j^j} \mathbf{v}_j^j & \text{if } k \neq j, \end{cases} \tag{2.1}$$

where $t_j^j \neq 0$ and t_k^j, $k \neq j$, are arbitrary real numbers. In what follows we consider that the vectors above are the columns of a matrix $\mathbf{V}^j$.

This transformation can be formulated in matrix form as follows. Given a matrix $\mathbf{V}^j = [\mathbf{v}_1^j, \ldots, \mathbf{v}_n^j]$, where $\mathbf{v}_i^j, i = 1, \ldots, n$, are column vectors, a new matrix $\mathbf{V}^{j+1}$ is defined via

$$\mathbf{V}^{j+1} = \mathbf{V}^j \mathbf{M}_j^{-1}, \tag{2.2}$$

where $\mathbf{M}_j^{-1}$ is the inverse of the matrix

$$\mathbf{M}_j = (\mathbf{e}_1, \ldots, \mathbf{e}_{j-1}, \mathbf{t}_j, \mathbf{e}_{j+1}, \ldots, \mathbf{e}_n)^T, \tag{2.3}$$

where $\mathbf{e}_i$ is the ith column of the identity matrix, the transpose of $\mathbf{t}_j$ being defined by

$$\mathbf{t}_j^T = \mathbf{u}_j^T \mathbf{V}^j \tag{2.4}$$

for some predestined vector $\mathbf{u}_j$.

Since $t_j^j \neq 0$, the matrix $\mathbf{M}_j$ is invertible. It can be proved that $\mathbf{M}_j^{-1}$ is the identity matrix with its jth row replaced by

$$\mathbf{t}_j^* = \frac{1}{t_j^j} \left(-t_1^j, \ldots, -t_{j-1}^j, 1, -t_{j+1}^j, \ldots, -t_n^j \right).$$

This transformation is used in well-known methods, such as the Gaussian elimination method. However, different selections of the t-values lead to completely different results. In this paper we base this selection on the concept of orthogonality, and assume a sequence of m transformations associated with a set of vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$.

3. Main properties of the pivoting transformation. As we shall see, the pivoting transformation has very important and useful properties that are illustrated in the following theorems.

The first theorem proves that given a matrix $\mathbf{V}$, the pivoting transformation transforms its columns without changing the linear subspace they generate.

THEOREM 3.1. *Let* $\mathcal{L}(\mathbf{V}^j) = \mathcal{L}\{\mathbf{v}_1^j, \ldots, \mathbf{v}_n^j\}$ *be the linear subspace generated or spanned by the set of vectors* $\{\mathbf{v}_1^j, \ldots, \mathbf{v}_n^j\}$. *Consider the pivoting transformation* (2.1) *or* (2.2) *and let* $\mathcal{L}(\mathbf{V}^{j+1}) = \mathcal{L}\{\mathbf{v}_1^{j+1}, \ldots, \mathbf{v}_n^{j+1}\}$; *then* $\mathcal{L}(\mathbf{V}^j) = \mathcal{L}(\mathbf{V}^{j+1})$.

Proof. The relationship (2.2) implies immediately that $\mathcal{L}(\mathbf{V}^{j+1}) \subset \mathcal{L}(\mathbf{V}^j)$. Conversely, the relationship

$$\mathbf{V}^{j+1}\mathbf{M}_j = \mathbf{V}^j$$

implies the other way. In fact, this theorem is true by merely looking at (2.2). ∎

The following theorem shows that the pivoting process (2.2) with the pivoting strategy (2.4) leads to the orthogonal decomposition of the linear subspace generated by the columns of $\mathbf{V}^j$ with respect to vector $\mathbf{u}$.

THEOREM 3.2 (orthogonal decomposition with respect to a given vector). *Assume now a vector* $\mathbf{u}_j \neq \mathbf{0}$ *and let* $t_k^j = \mathbf{u}_j^T\mathbf{v}_k^j, k = 1, \ldots, n$. *If* $t_j^j \neq 0$, *then*

$$\mathbf{u}_j^T\mathbf{V}^{j+1} = \mathbf{e}_j^T. \tag{3.1}$$

In addition, the linear subspace orthogonal to $\mathbf{u}_j$ *in* $\mathcal{L}(\mathbf{V}^j)$ *is*

$$\{\mathbf{v} \in \mathcal{L}(\mathbf{V}^j) | \mathbf{u}_j^T\mathbf{v} = 0\} = \mathcal{L}\left(\mathbf{v}_1^{j+1}, \ldots, \mathbf{v}_{j-1}^{j+1}, \mathbf{v}_{j+1}^{j+1}, \ldots, \mathbf{v}_n^{j+1}\right),$$

and its complement is $\mathcal{L}(\mathbf{v}_j^{j+1})$.

In other words, the transformation (2.2) *gives the generators of the linear subspace orthogonal to* $\mathbf{u}_j$ *and the generators of its complement.*

Proof. This theorem follows quickly from (2.4) and (2.2) because

$$\mathbf{u}^T\mathbf{V}^{j+1} = \mathbf{u}^T\mathbf{V}^j\mathbf{M}_j^{-1} = \mathbf{t}_j^T\mathbf{M}_j^{-1} = \mathbf{e}_j^T.$$

Finally, Theorem 3.1 guarantees that $\mathcal{L}(\mathbf{v}_j^{j+1})$ is the complement. ∎

Remark 1. Note that Theorem 3.2 allows us to obtain the linear subspace orthogonal to a given vector $\mathbf{u}_j$ in any case. If $t_j^j = 0$, we can reorder the $\mathbf{v}$ vectors until we satisfy the condition $t_j^j \neq 0$ or we find that $t_j^j = 0\ \forall j = 1, \ldots, n$, in which case the orthogonal set to $\mathbf{u}_j$ in $\mathcal{L}(\mathbf{V}^j)$ is all $\mathcal{L}(\mathbf{V}^j)$.

The following two theorems show that the pivoting transformation (2.2) allows obtaining the linear space orthogonal to a given linear space, in another linear space.

THEOREM 3.3. *If we sequentially apply the transformation in Theorem* 3.1 *based on a set of linearly independent vectors* $\{\mathbf{u}_1, \ldots, \mathbf{u}_j\}$, *the orthogonalization and normalization properties in* (3.1) *are kept. In other words, we have*

$$\mathbf{u}_r^T\mathbf{v}_k^{j+1} = \delta_{rk} \qquad \forall r \leq j \qquad \forall j, \tag{3.2}$$

where δ_{rk} *are the Kronecker deltas.*

Proof. We prove this by induction over j.

Step 1. The theorem is true for $j = 1$, because from (3.1) we have $\mathbf{u}_1^T\mathbf{v}_k^2 = \delta_{1k}$.

Step j. We assume that the theorem is true for j, that is,

$$\mathbf{u}_r^T \mathbf{v}_k^{j+1} = \delta_{rk} \qquad \forall r \le j \qquad \forall j.$$

Step $j+1$. We prove that it is true for $j+1$. In fact, we have

$$\mathbf{u}^T \mathbf{V}^{j+2} = \mathbf{u}_r^T \mathbf{V}^{j+1} \mathbf{M}_{j+1}^{-1} = \begin{cases} \mathbf{e}_r^T & \text{if } r = j+1, \\ \mathbf{e}_r^T \mathbf{M}_{j+1}^{-1} = \mathbf{e}_r^T & \text{if } r \le j. \end{cases} \qquad \square$$

THEOREM 3.4 (orthogonal decomposition with respect to a given linear subspace). *Assume the linear subspace* $\mathcal{L}\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$. *We can sequentially use Theorem* 3.2 *to obtain the orthogonal set to* $\mathcal{L}\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ *in a given subspace* $\mathcal{L}(\mathbf{V}^1)$. *Let* t_i^j *be the dot product of* $\mathbf{u}_j$ *and* $\mathbf{v}_i^j$. *Then assuming, without loss of generality, that* $t_q^{q-1} \neq 0$, *we obtain*

$$\begin{aligned} \mathcal{L}(\mathbf{V}^{q-1}) &= \mathcal{L}\left(\mathbf{v}_1^{q-1} - \frac{t_1^{q-1}}{t_q^{q-1}} \mathbf{v}_q^{q-1}, \ldots, \mathbf{v}_q^{q-1}, \ldots, \mathbf{v}_n^{q-1} - \frac{t_n^{q-1}}{t_q^{q-1}} \mathbf{v}_q^{q-1}\right) \\ &= \mathcal{L}\left(\mathbf{v}_1^q, \ldots, \mathbf{v}_n^q\right) = \mathcal{L}(\mathbf{V}^q) \end{aligned}$$

and

$$\{\mathbf{v} \in \mathcal{L}(\mathbf{V}^1) | \mathbf{u}_1^T \mathbf{v} = 0, \ldots, \mathbf{u}_q^T \mathbf{v} = 0\} = \mathcal{L}\left(\mathbf{v}_{q+1}^q, \ldots, \mathbf{v}_n^q\right).$$

In addition, we have

$$\mathbf{u}_1^T \mathbf{v}_1^q = 1, \quad \mathbf{u}_1^T \mathbf{v}_i^q = 0 \quad \forall i \neq 1, \ldots, \mathbf{u}_q^T \mathbf{v}_q^q = 1, \mathbf{u}_q^T \mathbf{v}_i^q = 0 \quad \forall i \neq q.$$

The proof can easily be obtained using Theorem 3.3.

The following remarks point out the practical significance of the above four theorems.

Remark 2. The linear subspace orthogonal to the linear subspace generated by vector $\mathbf{u}_j$ is the linear space generated by the columns of $\mathbf{V}^k$ for any $k \ge j+1$ with the exception of its pivot column, and its complement is the linear space generated by this pivot column of $\mathbf{V}^k$ for any $k \ge j+1$.

Remark 3. The linear subspace, in the linear subspace generated by the columns of $\mathbf{V}^1$, orthogonal to the linear subspace generated by any subset $W = \{\mathbf{u}_k | k \in K\}$ is the linear subspace generated by the columns of $\mathbf{V}^\ell, \ell \ge \max_{k \in K} k + 1$, with the exception of all pivot columns associated with the vectors in W, and its complement is the linear subspace generated by the columns of $\mathbf{V}^\ell, \ell \ge \max_{k \in K} k + 1$, which are their pivot columns.

4. The orthogonalization algorithm. In this section we describe an algorithm for obtaining orthogonal decompositions, which is based on Theorem 3.4.

ALGORITHM 1.

- **Input:** Two linear subspaces $\mathcal{L}(\mathbf{V}^1) = \mathcal{L}(\mathbf{v}_1, \ldots, \mathbf{v}_s) \subseteq \mathbb{R}^n$ and $\mathcal{L}(\mathbf{U}) = \mathcal{L}(\mathbf{u}_1, \ldots, \mathbf{u}_m) \subseteq \mathbb{R}^n$.
- **Output:** The orthogonal linear subspace $\mathcal{L}(\mathbf{W}_2)$ to $\mathcal{L}(\mathbf{U})$ in $\mathcal{L}(\mathbf{V}^1)$ and its complement $\mathcal{L}(\mathbf{W}_1)$.

Step 1: Set $\mathbf{W} = \mathbf{V}^1$ (the matrix with $\mathbf{v}_j, j = 1, \ldots, s$, as columns).
Step 2: Let $i = 1$ and $\ell = 0$.
Step 3: Calculate the dot products $t_j^\ell = \mathbf{u}_i^T \mathbf{w}_j, j = 1, \ldots, s$.

TABLE 4.1

Iterations for obtaining the orthogonal decomposition of $\mathcal{L}(\mathbf{V}^1)$ with respect to $\mathcal{L}(\mathbf{U})$. Pivot columns are boldfaced.

Iteration 1						*Iteration* 2					
1	**1**	0	0	0	0	3	*1*	1	**−1**	0	0
−1	**0**	1	0	0	0	−3	*0*	1	**0**	0	0
1	**0**	0	1	0	0	0	*0*	0	**1**	0	0
0	**0**	0	0	1	0	1	*0*	0	**0**	1	0
0	**0**	0	0	0	1	0	*0*	0	**0**	0	1
	1	−1	1	0	0		*3*	0	**−3**	1	0

Modified second table						*Iteration* 3					
3	*1*	**−1**	1	0	0	0	*0*	*1/3*	1	**−1/3**	0
−3	*0*	**0**	1	0	0	0	*0*	*0*	1	**0**	0
0	*0*	**1**	0	0	0	−1	*1*	*−1/3*	0	**1/3**	0
1	*0*	**0**	0	1	0	0	*0*	*0*	0	**1**	0
0	*0*	**0**	0	0	1	1	*0*	*0*	0	**0**	1
	3	**−3**	0	1	0		*−1*	*1/3*	0	**−1/3**	1

Output				
1	*0*	*1*	1	−1
0	*0*	*0*	1	0
0	*0*	*−1*	0	1
−3	*1*	*−3*	0	3
0	*0*	*0*	0	1

Step 4: For $j = \ell+1$ to s locate the pivot column r_ℓ as the first column not orthogonal to $\mathbf{u}_i$, that is, $t^\ell_{r_\ell} \neq 0$. If there is no such a column go to Step 7. Otherwise, continue with Step 5.

Step 5: Increase ℓ in one unit, divide the r_ℓ column by $t^\ell_{r_\ell}$, and if $r_\ell \neq \ell$, switch columns ℓ and r_ℓ and associated dot products $t^\ell_{r_\ell}$ and t^ℓ_ℓ.

Step 6: For $j = 1$ to s and $j \neq r_\ell$ do the following: If $t^\ell_j \neq 0$, do $w_{kj} = w_{kj} - t^\ell_j w_{ki}$ for $k = 1, \ldots, n$.

Step 7: If $i = m$, go to Step 8. Otherwise, increase i in one unit and go to Step 3.

Step 8: Return $\mathcal{L}(\mathbf{W}_2) = \mathcal{L}(\mathbf{w}_\ell, \ldots, \mathbf{w}_s)$ as the orthogonal subspace of $\mathcal{L}(\mathbf{U})$ in $\mathcal{L}(\mathbf{V}^1)$ and $\mathcal{L}(\mathbf{W}_1) = \mathcal{L}(\mathbf{w}_1, \ldots, \mathbf{w}_{\ell-1})$ as its complement.

Remark 4. If the pivoting process were used taking into account numerical considerations, Step 4 should be adequately modified by the corresponding pivot selecting strategy (maximum pivot strategy, for example). In this case, the corresponding permutation in Step 8 is required. Note that in this paper only algebraic considerations are used.

The process described in Algorithm 1 can be organized in a tabular form. A detailed description is given in the following example.

Example 1 (orthogonal decomposition). Consider the linear subspace of $\mathcal{L}(\mathbf{V}^1) = \mathbb{R}^5$:

$$\mathcal{L}(\mathbf{U}) = \mathcal{L}\left\{(1,-1,1,0,0)^T, (3,-3,0,1,0)^T, (0,0,-1,0,1)^T\right\}.$$

We organize the procedure in a tabular form (see Table 4.1).

First, to obtain the orthogonal decomposition of $\mathcal{L}(\mathbf{V}^1)$ with respect to $\mathcal{L}(\mathbf{U})$, we construct the initial tableau (see Iteration 1 in Table 4.1), starting with the identity matrix $\mathbf{V}^1$. The first column of this table is the first generator of $\mathcal{L}(\mathbf{U})$ and the generators of the subspace to be decomposed are in the other columns. The last row contains the inner products of the vector in the first column by the corresponding column vectors.

Next, the first nonnull element in the last row is identified and the corresponding column is selected as the pivot column, which is boldfaced in Iteration 1.

Finally, it is necessary to perform the pivoting process and to update the first column and the last row of the table with the next generator of $\mathcal{L}(\mathbf{U})$ and the new inner products. Then, we get the second table (see Iteration 2 in Table 4.1). In order to select the pivot column, we have to look for the first nonnull element in the last row starting with its second element because we are in the second iteration. Then, the selected column is the third one, and before performing the pivoting process, interchange of second and third columns must be done.

We repeat the pivoting process, incorporate the last generators of $\mathcal{L}(\mathbf{U})$, and obtain the new dot products. We select the pivot column, starting at column three, and look for a nonnull dot product, obtaining the fourth column as the pivot. Next, we repeat the normalization and pivoting processes and, finally, we get the Output tableau in Table 4.1, where the first three vectors are the generators of the complement subspace and the last two are the generators of the orthogonal subspace. Italicized columns are used in all iterations to refer to the complementary subspace.

Thus, the orthogonal decomposition becomes

$$\begin{aligned}\mathbb{R}^5 &= \mathcal{L}\left\{(1,0,0,-3,0)^T,(0,0,0,1,0)^T,(1,0,-1,-3,0)^T\right\}\\ &\quad\oplus\mathcal{L}\left\{(1,1,0,0,0)^T,(-1,0,1,3,1)^T\right\}.\end{aligned}$$

Note that, from the Output tableau, we can obtain the linear subspace orthogonal to the linear subspace generated by any subset of the initial set of vectors. For example, the orthogonal complement of the linear subspace generated by the set $\{(1,-1,1,0,0)^T,(3,-3,0,1,0)^T\}$ is (see Output in Table 4.1)

$$\mathcal{L}(\{(1,-1,1,0,0)^T,(3,-3,0,1,0)^T\})^\perp=\mathcal{L}\{(1,0,-1,-3,0)^T,(1,1,0,0,0)^T,(-1,0,1,3,1)^T\},$$

which can also be written as (see Iteration 3 in Table 4.1)

$$\mathcal{L}(\{(1,-1,1,0,0)^T,(3,-3,0,1,0)^T\})^\perp=\mathcal{L}\{(1,1,0,0,0),(-1/3,0,1/3,1,0),(0,0,0,0,1)\}.$$

Similarly,

$$\mathcal{L}(\{(0,0,-1,0,1)^T\})^\perp=\mathcal{L}\{(1,0,0,-3,0)^T,(0,0,0,1,0)^T,(1,1,0,0,0)^T,(-1,0,1,3,1)^T\}.$$

5. Applications. In addition to obtaining the linear subspace orthogonal to a linear space generated by one or several vectors in a given linear subspace, the proposed orthogonal pivoting transformation allows solving the following problems:

1. calculating the inverse of a matrix,
2. updating the inverse of a matrix after changing a row,
3. determining the rank of a matrix,
4. calculating the determinant of a matrix,
5. updating the determinant of a matrix after changing a row,
6. determining whether or not a set of vectors is linearly independent,
7. obtaining the intersection of two linear subspaces,
8. solving a homogeneous system of linear equations,
9. solving a complete system of linear equations,
10. deciding whether or not a linear system of equations is compatible.

5.1. Calculating the inverse of a matrix. The following theorem shows that Algorithm 1 can be used for obtaining the inverse of a matrix.

THEOREM 5.1. *Assume that Algorithm* 1 *is applied to the rows of matrix* $\mathbf{A}=(\mathbf{a}_1,\ldots,\mathbf{a}_n)^T$ *using a nonsingular initial matrix* $\mathbf{V}^1$. *Then the matrix whose columns*

are in the last tableau $\mathbf{V}^{n+1}$ *is the inverse of matrix* $\mathbf{A}$*. In addition, if we start with* $\mathbf{V}^1$ *being the identity matrix, in the process we obtain the inverses of all block main diagonal matrices.*

Proof. Matrices $\mathbf{V}^j$ for $j = 2, \ldots, n+1$ are obtained, using the transformations

$$\mathbf{V}^{j+1} = \mathbf{V}^j \mathbf{M}_j^{-1}, \qquad j = 1, \ldots, n, \tag{5.1}$$

where $\mathbf{M}_j$ is defined in (2.3) with $\mathbf{t}_j^T = \mathbf{a}_j^T \mathbf{V}^j$. Then it satisfies

$$\mathbf{a}_j^T \mathbf{V}^n = \mathbf{a}_j^T \mathbf{V}^j \mathbf{M}_j^{-1} \cdots \mathbf{M}_n^{-1} = \mathbf{t}_j^T \mathbf{M}_j^{-1} \cdots \mathbf{M}_n^{-1} = \mathbf{e}_j^T \mathbf{M}_{j+1}^{-1} \cdots \mathbf{M}_n^{-1} = \mathbf{e}_j^T. \tag{5.2}$$

This proves that $\mathbf{A}^{-1} = \mathbf{V}^n$; that is, the inverse of $\mathbf{A}$ is the matrix whose columns are in the final tableau obtained using Algorithm 1.

The second part of the theorem is obvious because the lower triangular part of the identity matrix is null and does not affect the dot products and the pivoting transformations involved in the process. ∎

Example 2 (matrix inverses). Consider the following matrix $\mathbf{A}$, where the block main diagonal matrices are shown, and its inverse $\mathbf{A}^{-1}$:

$$\mathbf{A} = \left(\begin{array}{ccc|cc} \underline{1} & 1 & 0 & 1 & 0 \\ \underline{-1} & 1 & -1 & 0 & 0 \\ \hline 0 & 0 & \underline{1} & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \\ \hline 0 & 1 & 0 & -1 & 1 \end{array}\right); \quad \mathbf{A}^{-1} = \begin{pmatrix} 2/7 & -5/7 & -5/7 & 1/7 & 3/7 \\ 3/7 & 3/7 & 3/7 & -2/7 & 1/7 \\ 1/7 & 1/7 & 8/7 & -3/7 & -2/7 \\ 2/7 & 2/7 & 2/7 & 1/7 & -4/7 \\ -1/7 & -1/7 & -1/7 & 3/7 & 2/7 \end{pmatrix}. \tag{5.3}$$

Table 5.1 shows the iterations for inverting $\mathbf{A}$ using Algorithm 1. The inverse matrix $\mathbf{A}^{-1}$ is obtained in the last iteration (see Table 5.1). In addition, Table 5.1 also contains the inverses of the block main diagonal matrices indicated below (see the marked matrices in Iterations 2 to 5 in Table 5.1). The important result is that this is obtained with no extra computation.

Finally, we mention that the 5×5 matrices we obtain in Iterations 2 to 5 in Table 5.1 are the inverses of the matrices that result from replacing in the unit matrix its rows by the rows of matrix $\mathbf{A}$. For example, the matrix in Iteration 4 is such that

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}; \ \mathbf{H}^{-1} = \begin{pmatrix} 1/2 & -1/2 & -1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 & -1/2 & -1/2 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.4}$$

Example 3 (inversion of a matrix starting from a regular matrix). The proposed pivoting process can be done starting with an arbitrary nonsingular matrix. For example, if we start with the matrix

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & -1 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 2 & 1 & 0 & 1 \end{pmatrix}, \tag{5.5}$$

we get the results in Table 5.2, i.e., the same inverse.

TABLE 5.1
Iterations for inverting the matrix in Example 2. Pivot columns are boldfaced. The inverses of all block main diagonal matrices are indicated in Iterations 2 to 5.

Iteration 1						*Iteration 2*					
1	**1**	0	0	0	0	–1	*1*	**–1**	0	–1	0
1	**0**	1	0	0	0	1	*0*	**1**	0	0	0
0	**0**	0	1	0	0	–1	*0*	**0**	1	0	0
1	**0**	0	0	1	0	0	*0*	**0**	0	1	0
0	**0**	0	0	0	1	0	*0*	**0**	0	0	1
	1	1	0	1	0		*–1*	**2**	–1	1	0
Iteration 3						*Iteration 4*					
0	*1/2*	*–1/2*	**–1/2**	–1/2	0	0	*1/2*	*–1/2*	*–1/2*	**–1/2**	1/2
0	*1/2*	*1/2*	**1/2**	–1/2	0	0	*1/2*	*1/2*	*1/2*	**–1/2**	–1/2
1	*0*	*0*	**1**	0	0	0	*0*	*0*	*1*	**0**	–1
0	*0*	*0*	**0**	1	0	1	*0*	*0*	*0*	**1**	0
1	*0*	*0*	**0**	0	1	2	*0*	*0*	*0*	**0**	1
	0	*0*	**1**	0	1		*0*	*0*	*0*	**1**	2
Iteration 5						*Output*					
0	*1/2*	*–1/2*	*–1/2*	*–1/2*	**3/2**		*2/7*	*–5/7*	*–5/7*	*1/7*	*3/7*
1	*1/2*	*1/2*	*1/2*	*–1/2*	**1/2**		*3/7*	*3/7*	*3/7*	*–2/7*	*1/7*
0	*0*	*0*	*1*	*0*	**–1**		*1/7*	*1/7*	*8/7*	*–3/7*	*–2/7*
–1	*0*	*0*	*0*	*1*	**–2**		*2/7*	*2/7*	*2/7*	*1/7*	*–4/7*
1	*0*	*0*	*0*	*0*	**1**		*–1/7*	*–1/7*	*–1/7*	*3/7*	*2/7*
	1/2	*1/2*	*1/2*	*–3/2*	**7/2**						

5.2. Updating the inverse of a matrix after changing a row. In this section we start by giving an interpretation to each tableau obtained in the inversion process of a matrix.

Since, according to Theorem 3.3, the pivoting transformation does not alter the orthogonal properties of previous vectors, we can update the inverse of a matrix after changing a row by an additional pivoting transformation in which the new row vector is used.

To illustrate this, we use the results in Table 5.2. Note that the matrices in Iterations 2 to 5 correspond to the matrices obtained from $\mathbf{B}^{-1}$ after sequentially replacing the row which number coincides with the number of the pivot column by their associated **u**-vectors. In other words, matrices in Table 5.2, Iterations 2 to 5, are the inverses of the following matrices:

$$A_1 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -8 & 7 & 6 & 4 & -2 \\ 6 & -5 & -4 & -3 & 2 \\ -9 & 8 & 7 & 5 & -3 \\ 10 & -9 & -8 & -5 & 3 \end{pmatrix}; \; A_2 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 6 & -5 & -4 & -3 & 2 \\ -9 & 8 & 7 & 5 & -3 \\ 10 & -9 & -8 & -5 & 3 \end{pmatrix};$$

$$A_3 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ -9 & 8 & 7 & 5 & -3 \\ 10 & -9 & -8 & -5 & 3 \end{pmatrix}; \; A_4 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \\ 10 & -9 & -8 & -5 & 3 \end{pmatrix}.$$

5.3. Determining the rank of a matrix. In this section we see that Algorithm 1 also allows one to determine the rank of a matrix.

TABLE 5.2
Iterations for inverting the matrix in Example 2 *when we start with matrix* $\mathbf{A}$ *in* (5.5). *Pivot columns are boldfaced.*

Iteration 1						*Iteration* 2					
1	**1**	0	1	1	0	–1	*1/3*	**–2/3**	1/3	0	–2/3
1	**2**	1	0	0	0	1	*2/3*	**–1/3**	–4/3	–2	–4/3
0	**–1**	–1	1	0	–1	–1	*–1/3*	**–1/3**	5/3	1	–1/3
1	**0**	1	1	2	2	0	*0*	**1**	1	2	2
0	**0**	2	1	0	1	0	*0*	**2**	1	0	1
	3	2	2	3	2		*2/3*	**2/3**	–10/3	–3	–1/3
Iteration 3						*Iteration* 4					
0	*1*	*–1*	**–3**	–3	–1	0	*5/11*	*–7/22*	*–3/11*	**–15/22**	–13/22
0	*1*	*–1/2*	**–3**	–7/2	–3/2	0	*5/11*	*2/11*	*–3/11*	**–13/11**	–12/11
1	*0*	*–1/2*	**0**	–1/2	–1/2	0	*0*	*–1/2*	*0*	**–1/2**	–1/2
0	*–1*	*3/2*	**6**	13/2	5/2	1	*1/11*	*3/22*	*6/11*	**41/22**	37/22
1	*–2*	*3*	**11**	9	2	2	*0*	*1/2*	*1*	**1/2**	1/2
	–2	*5/2*	**11**	17/2	3/2		*1/11*	*25/22*	*28/11*	**63/22**	59/22
Iteration 5						*Output*					
0	*10/21*	*–1/21*	*1/3*	*–5/21*	**1/21**		*2/7*	*–5/7*	*-5/7*	*1/7*	*3/7*
1	*31/63*	*41/63*	*7/9*	*–26/63*	**1/63**		*3/7*	*3/7*	*3/7*	*-2/7*	*1/7*
0	*1/63*	*–19/63*	*4/9*	*–11/63*	**–2/63**		*1/7*	*1/7*	*8/7*	*–3/7*	*–2/7*
–1	*2/63*	*–38/63*	*–10/9*	*41/63*	**–4/63**		*2/7*	*2/7*	*2/7*	*1/7*	*–4/7*
1	*–1/63*	*19/63*	*5/9*	*11/63*	**2/63**		*–1/7*	*–1/7*	*–1/7*	*3/7*	*2/7*
	4/9	*14/9*	*22/9*	*–8/9*	**1/9**						

In an n-dimensional linear space, the rank of a matrix $\mathbf{U}$ coincides with n minus the dimension of its orthogonal complement. Thus, if during the orthogonalization process we start with a nonsingular matrix as columns and we can find a pivot in all iterations, then the corresponding matrix is full rank. Otherwise, the rank is equal to the number of pivot columns we can find.

Example 4 (rank of a matrix). Assume that we are interested in calculating the rank of the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 2 & 1 & 2 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 \end{pmatrix}.$$

In Table 5.3 we show the iterations for obtaining its rank. We can see that the rank of $\mathbf{A}$ is 3, since the third and fifth iterations have no pivots.

5.4. Calculating the determinant of a matrix. The following theorem shows that the determinant of a matrix can be calculated by means of Algorithm 1.

THEOREM 5.2 (determinant of a matrix). *The determinant of a matrix* $\mathbf{A}$ *can be calculated by Algorithm* 1 *by multiplying the normalizing constants* $t_{r_\ell}^{\ell}, l = 1, \ldots, n$, *used in Step* 5 *and* $(-1)^p$, *where* p *is the number of interchanges of columns that have occurred when executing the algorithm. If we start the algorithm with the identity matrix, the determinants of the block main diagonal matrices referred to in section* 5.1 *are the partial products.*

Proof. Assume that we start in Step 1 with an identity matrix, $\mathbf{W} = \mathbf{I}_n$, which has a determinant of one. In the inverting process, we transform this matrix using two different transformations: the pivoting step, which does not alter the determinant

TABLE 5.3
Iterations for calculating the rank of the matrix in Example 4. Pivot columns are boldfaced.

Iteration 1					*Iteration* 2						*Iteration* 3					
1	**1**	0	0	0	0	*1*	**0**	–1	–1	–1	1	*1*	*0*	–1	–1	–1
0	**0**	1	0	0	1	*0*	**1**	0	0	0	1	*0*	*1*	–1	0	–1
1	**0**	0	1	0	1	*0*	**0**	1	0	0	2	*0*	*0*	1	0	0
1	**0**	0	0	1	0	*0*	**0**	0	1	0	1	*0*	*0*	0	1	0
1	**0**	0	0	0	1	*0*	**0**	0	0	1	2	*0*	*0*	0	0	1
	1	0	1	1	1	*0*	**1**	1	0	1		*1*	*1*	0	0	0

Iteration 4						*Iteration* 5					
1	*1*	*0*	**–1**	–1	–1	1	*1/2*	*–1/2*	*1/2*	–1/2	0
1	*0*	*1*	**–1**	0	–1	–1	*–1/2*	*1/2*	*1/2*	1/2	0
0	*0*	*0*	**1**	0	0	0	*1/2*	*1/2*	*–1/2*	–1/2	–1
0	*0*	*0*	**0**	1	0	1	*0*	*0*	*0*	1	0
0	*0*	*0*	**0**	0	1	0	*0*	*0*	*0*	0	1
	1	*1*	**–2**	–1	–2		*1*	*–1*	*0*	0	0

value of the matrix, and the normalization step, which divides its determinant by $t^{\ell}_{r_\ell}$ (see (2.1)). In addition, we multiply it by -1 each time we switch columns. Since $|\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$, we have

$$|\mathbf{A}| = \prod_{i=1}^{n} (-1)^p t^i_{r_i}. \tag{5.6}$$

If we start with an identity matrix, the lower triangular part of the identity matrix is null and does not affect the dot products and the pivoting transformations involved in the process. Thus, the result holds. ∎

Example 5 (determinant of a matrix). The determinant of the matrix in Example 2 is obtained by multiplying the normalizing constants, that is, the last values in the boldfaced columns in Table 5.1. Thus, we have

$$1 \times 2 \times 1 \times 1 \times 7/2 = 7.$$

The determinants of the block main diagonal matrices in (5.3) are $1, 2, 2, 2$, and 7, respectively.

Remark 5. If instead of starting with the identity matrix $\mathbf{I}_n$, we start with a nonsingular matrix $\mathbf{B}$ with determinant $|\mathbf{B}|$, expression (5.6) becomes

$$|\mathbf{A}| = |\mathbf{B}|^{-1} \prod_{i=1}^{n} (-1)^p t^i_{r_i}. \tag{5.7}$$

5.5. Updating the determinant of a matrix after changing a row. According to (5.6) or (5.7), the determinant is updated by multiplying the previous determinant by the dot product of the new row by the associated pivot column.

Example 6 (updating the determinant after changing a row). Consider the matrix $\mathbf{A}$ and its inverse $\mathbf{A}^{-1}$,

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{A}^{-1} = \begin{pmatrix} 2/7 & -5/7 & -5/7 & 1/7 & 3/7 \\ 3/7 & 3/7 & 3/7 & -2/7 & 1/7 \\ 1/7 & 1/7 & 8/7 & -3/7 & -2/7 \\ 2/7 & 2/7 & 2/7 & 1/7 & -4/7 \\ -1/7 & -1/7 & -1/7 & 3/7 & 2/7 \end{pmatrix}, \tag{5.8}$$

TABLE 5.4
Pivoting process to determine whether or not a set of vectors is linearly dependent.

Iteration 1					*Iteration* 2					*Iteration* 3					*Iteration* 4				
1	**1**	0	0	0	2	1	**0**	–1	–1	1	1	0	**–1**	–1	–1	1/4	1/4	1/4	0
0	**0**	1	0	0	–1	0	**1**	0	0	1	2	–1	**–3**	–2	1	–1/4	–1/4	3/4	1
1	**0**	0	1	0	–1	0	**0**	1	0	0	0	0	**1**	0	2	3/4	–1/4	–1/4	–1
1	**0**	0	0	1	0	0	**0**	0	1	–1	0	0	**0**	1	1	0	0	0	*1*
$\mathbf{t}^1$	**1**	0	1	1	$\mathbf{t}^2$	2	**–1**	–3	–2	$\mathbf{t}^3$	3	–1	**–4**	–4	$\mathbf{t}^4$	1	–1	0	0

and assume that we want to calculate the determinant of the matrix

$$\mathbf{B} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ a & b & c & d & e \\ 0 & 1 & 0 & -1 & 1 \end{pmatrix}.$$

Since $|\mathbf{A}| = 7$, we have

$$|\mathbf{B}| = 7 \times (a, b, c, d, e)(1/7, -2/7, -3/7, 1/7, 3/7)^T = a - 2b - 3c + d + 3e.$$

5.6. Determining whether or not a set of vectors is linearly dependent. To know whether or not a set of vectors is linearly independent, we use the property

$$\{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \text{ are linearly dependent } \Leftrightarrow \mathbf{u}_n \in \mathcal{L}\{\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}\}$$

which can be written as

$$\{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \text{ are linearly dependent } \Leftrightarrow \mathbf{u}_n \perp \mathcal{L}\{\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}\}^{\perp}.$$

Thus, the problem reduces to obtaining a set of generators of the orthogonal complement of $\mathcal{L}\{\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}\}$ and checking that the dot products of each of its generators by $\mathbf{u}_n$ are null.

Note that this problem is the same as determining whether or not a vector belongs to a linear subspace.

Example 7 (linear dependence of a set of vectors). Consider the set of vectors

$$\{(1, 0, 1, 1), (2, -1, -1, 0), (1, 1, 0, -1), (-1, 1, 2, 1)\}.$$

If we use the pivoting process (see Table 5.4), we have no problem finding a pivot column for the first three vectors, but there is no pivot column for the fourth vector. This means that the fourth vector is a linear combination of the first three.

5.7. Obtaining the intersection of two linear subspaces. Theorem 3.4 allows us to obtain the intersection of two linear subspaces S_1 and S_2 by noting that

$$S_1 \cap S_2 = S_1 \cap (S_2^{\perp})^{\perp} = S_2 \cap (S_1^{\perp})^{\perp}. \tag{5.9}$$

In fact, we can obtain first $S_2^{\perp}$, the orthogonal to S_2, by letting $\mathcal{L}(\mathbf{V}^1) = \mathbb{R}^n$ in Theorem 3.4 and then find the orthogonal to $S_2^{\perp}$ in S_1, using S_1 as $\mathcal{L}(\mathbf{V}^1)$. Alternatively, we can obtain first $S_1^{\perp}$, the orthogonal to S_1, by letting $\mathcal{L}(\mathbf{V}^1) = R^n$ in Theorem 3.4 and then find the orthogonal to $S_1^{\perp}$ in S_2, using S_2 as $\mathcal{L}(\mathbf{V}^1)$.

Example 8 (intersection of moving subspaces). Consider the linear subspaces $S_1 = \mathcal{L}\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ and $S_2^{\perp} = \mathcal{L}\{\mathbf{v}_1, \mathbf{v}_4\}$, where

$$\begin{aligned}
\mathbf{v}_1 &= (\ 1, \quad \sin 2t, \quad -1, \quad \cos t\ )^T,\\
\mathbf{v}_2 &= (\ \cos t, \quad 1, \quad \sin 2t, \quad -1\ )^T,\\
\mathbf{v}_3 &= (\ -1, \quad \cos t, \quad 1, \quad \sin 2t\ )^T,\\
\mathbf{v}_4 &= (\ \sin 2t, \quad -1, \quad \cos t, \quad 1\ )^T,
\end{aligned}$$

and assume that we wish to

1. determine the intersection $Q_1 = S_1 \cap S_2$ for all values of the time parameter $0 \le t \le 2\pi$;
2. find the t-values for which we have $Q_1 = Q_2$, where $Q_2 = S_1 \cap S_3$ and $S_3^{\perp} = \mathcal{L}\{\mathbf{v}_1, \mathbf{v}_2\}$.

Then we have the following:

1. By definition we can write

$$Q_1 = \{\mathbf{v} \in S_1 | \mathbf{v} \in S_2\}; \mathbf{v} \in S_2 \Leftrightarrow\ \mathbf{v}^T\mathbf{v}_1 = 0 \text{ and } \mathbf{v}^T\mathbf{v}_4 = 0.$$

Using the procedure in Theorem 3.4 and starting with $\mathbf{v}_1$, we get

$$\begin{aligned}
&p = \mathbf{v}_1^T\mathbf{v}_1 = \sin^2 2t + \cos^2 t + 2,\\
&q = \mathbf{v}_1^T\mathbf{v}_3 = 2\sin 2t \cos t - 2,\\
&\mathbf{v}_1^T\mathbf{v}_2 = \mathbf{v}_1^T\mathbf{v}_4 = 0.
\end{aligned} \tag{5.10}$$

Since $p \neq 0\ \forall t$, using the orthogonalization procedure in Theorem 3.4, we obtain

$$\left\{\mathbf{v} \in S_1 | \mathbf{v}^T\mathbf{v}_1 = 0\right\} = \mathcal{L}\{\mathbf{u}_1 = p\mathbf{v}_3 - q\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4)\},$$

and proceeding with $\mathbf{v}_4$ and taking into account that

$$\mathbf{v}_4^T\mathbf{v}_2 = q; \mathbf{v}_4^T\mathbf{v}_4 = p; \mathbf{v}_4^T\mathbf{u}_1 = 0,$$

we have

$$Q_1 = \mathcal{L}\{p\mathbf{v}_3 - q\mathbf{v}_1, p\mathbf{v}_2 - q\mathbf{v}_4\} = \mathcal{L}\{\mathbf{u}_1, \mathbf{u}_2\}.$$

2. By a similar process we get

$$Q_2 = \mathcal{L}\{p\mathbf{v}_3 - q\mathbf{v}_1, p\mathbf{v}_4 - q\mathbf{v}_2\} = \mathcal{L}\{\mathbf{u}_1, \mathbf{u}_3\}.$$

Since Q_2 is orthogonal to $\mathbf{v}_2$ and $Q_1 = Q_2$, then Q_1 is orthogonal to $\mathbf{v}_2$ and, in particular, $\mathbf{u}_2$ is orthogonal to $\mathbf{v}_2$. Similarly, since Q_1 is orthogonal to $\mathbf{v}_4$ and $Q_1 = Q_2$, then Q_2 is orthogonal to $\mathbf{v}_4$ and, in particular, $\mathbf{u}_3$ is orthogonal to $\mathbf{v}_4$; that is,

$$\begin{aligned}
&Q_1 = Q_2 \Rightarrow \mathbf{u}_2^T\mathbf{v}_2 = 0, \mathbf{u}_3^T\mathbf{v}_4 = 0 \Rightarrow p^2 - q^2 = 0\\
&\Rightarrow p = -q \Rightarrow \cos t(2\sin t + 1) = 0 \Rightarrow t \in A = \left\{\frac{\pi}{2}, \frac{3\pi}{2}, \frac{7\pi}{6}, \frac{11\pi}{6}\right\}
\end{aligned}$$

and, conversely,

$$p = -q \Rightarrow \mathbf{u}_2 = \mathbf{u}_3 \Rightarrow Q_1 = Q_2.$$

Thus, we get

$$Q_1 = Q_2 \Leftrightarrow t \in A = \left\{\frac{\pi}{2}, \frac{3\pi}{2}, \frac{7\pi}{6}, \frac{11\pi}{6}\right\}.$$

TABLE 5.5
Pivoting transformations corresponding to Example 9.

Iteration 1						*Iteration 2*					*Iteration 3*				*Final*	
$\mathbf{a}_1$	$\mathbf{v}_1^1$	$\mathbf{v}_2^1$	$\mathbf{v}_3^1$	$\mathbf{v}_4^1$	$\mathbf{v}_5^1$	$\mathbf{a}_2$	$\mathbf{v}_1^2$	$\mathbf{v}_2^2$	$\mathbf{v}_3^2$	$\mathbf{v}_4^2$	$\mathbf{a}_3$	$\mathbf{v}_1^3$	$\mathbf{v}_2^3$	$\mathbf{v}_3^3$	$\mathbf{v}_1$	$\mathbf{v}_2$
1	**1**	0	0	0	0	0	**−1**	1	−1	2	0	**1**	0	0	1	0
1	**0**	1	0	0	0	1	**1**	0	0	0	0	**0**	−1	2	−1	2
−1	**0**	0	1	0	0	0	**0**	1	0	0	1	**1**	0	0	1	0
1	**0**	0	0	1	0	1	**0**	0	1	0	−1	**0**	1	0	1	0
−2	**0**	0	0	0	1	−2	**0**	0	0	1	0	**0**	0	1	0	1
$\mathbf{t}^1$	**1**	1	−1	1	−2	$\mathbf{t}^2$	**1**	0	1	−2	$\mathbf{t}^3$	**1**	−1	0		

5.8. Solving a homogeneous system of linear equations. Consider the homogeneous system of equations

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &=& 0, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &=& 0, \\ \cdots\cdots \quad \cdots\cdots \quad \cdots\cdots \quad \cdots\cdots & & \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &=& 0 \end{array} \tag{5.11}$$

which can be written as

$$\begin{array}{rcl} (a_{11}, \ldots, a_{1n})(x_1, \ldots, x_n)^T &=& 0, \\ (a_{21}, \ldots, a_{2n})(x_1, \ldots, x_n)^T &=& 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots & & \cdots \\ (a_{m1}, \ldots, a_{mn})(x_1, \ldots, x_n)^T &=& 0. \end{array} \tag{5.12}$$

Expression (5.12) shows that $(x_1, \ldots, x_n)$ is orthogonal to the set of vectors

$$\{(a_{11}, \ldots, a_{1n}), (a_{21}, \ldots, a_{2n}), \ldots, (a_{m1}, \ldots, a_{mn})\}.$$

Then, obtaining the solution to system (5.11) reduces to determining the orthogonal complement of the linear subspace generated by the rows of matrix $\mathbf{A}$.

Example 9 (a homogeneous system of linear equations). Consider the system of equations

$$\begin{array}{rcl} x_1 + x_2 - x_3 + x_4 - 2x_5 &=& 0, \\ x_2 \qquad + x_4 - 2x_5 &=& 0, \\ x_3 - x_4 \qquad\quad &=& 0. \end{array} \tag{5.13}$$

To solve this system, we obtain the orthogonal complement of the linear subspace generated by the rows of the system matrix, as shown in Table 5.5. Thus, the solution is

$$(x_1, x_2, x_3, x_4, x_5) = \rho_1 (1, -1, 1, 1, 0) + \rho_2 (0, 2, 0, 0, 1), \tag{5.14}$$

where ρ_1 and ρ_2 are arbitrary real numbers.

5.9. Solving a complete system of linear equations. Now consider the complete system of linear equations:

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &=& b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &=& b_2, \\ \cdots \quad \cdots \quad \cdots \quad \cdots & & \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &=& b_m. \end{array} \tag{5.15}$$

Adding the artificial variable x_{n+1}, it can be written as

$$\begin{array}{rcccccl} a_{11}x_1 & +a_{12}x_2 & +\cdots & +a_{1n}x_n & -b_1x_{n+1} & = & 0, \\ a_{21}x_1 & +a_{22}x_2 & +\cdots & +a_{2n}x_n & -b_2x_{n+1} & = & 0, \\ \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots \\ a_{m1}x_1 & +a_{m2}x_2 & +\cdots & +a_{mn}x_n & -b_mx_{n+1} & = & 0, \\ a_{m1}x_1 & +a_{m2}x_2 & +\cdots & +a_{mn}x_n & -b_mx_{n+1} & = & 0, \end{array} \tag{5.16}$$

$$x_{n+1} = 1. \tag{5.17}$$

System (5.16) can be written as

$$\begin{array}{rcl} (a_{11},\ldots,a_{1n},-b_1)(x_1,\ldots,x_n,x_{n+1})^T & = & 0, \\ (a_{21},\ldots,a_{2n},-b_2)(x_1,\ldots,x_n,x_{n+1})^T & = & 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & & \cdots \\ (a_{m1},\ldots,a_{mn},-b_m)(x_1,\ldots,x_n,x_{n+1})^T & = & 0. \end{array} \tag{5.18}$$

Expression (5.18) shows that $(x_1,\ldots,x_n,x_{n+1})$ is orthogonal to the set of vectors

$$\{(a_{11},\ldots,a_{1n},-b_1),(a_{21},\ldots,a_{2n},-b_2),\ldots,(a_{m1},\ldots,a_{mn},-b_m)\}.$$

Then, it is clear that the solution of (5.16) is the orthogonal complement of the linear subspace generated by the rows of matrix $\mathbf{A}$:

$$\mathcal{L}\{(a_{11},\ldots,a_{1n},-b_1),(a_{21},\ldots,a_{2n},-b_2),\ldots,(a_{m1},\ldots,a_{mn},-b_m)\}^{\perp}.$$

Thus, the solution of (5.15) is the projection on $X_1\times\cdots\times X_n$ of the intersection of the orthogonal complement of the linear subspace generated by

$$\{(a_{11},\ldots,a_{1n},-b_1),(a_{21},\ldots,a_{2n},-b_2),\ldots,(a_{m1},\ldots,a_{mn},-b_m)\}$$

and the set $\{\mathbf{x}|x_{n+1}=1\}$.

Example 10 (a complete system of linear equations). Consider the system of equations

$$\begin{array}{rcccccl} x_1 & +x_2 & -x_3 & +x_4 & = & 2, \\ & x_2 & & +x_4 & = & 2, \\ & & x_3 & -x_4 & = & 0 \end{array} \tag{5.19}$$

which, using the auxiliary variable x_5, can be written as (5.13). Since the solution of the homogeneous system (5.13) was already obtained, now we only need to force $x_5=1$ and return to the initial set of variables. Thus, the solution is

$$(x_1,x_2,x_3,x_4)=(0,2,0,0)+\rho_1\,(1,-1,1,1)\,, \tag{5.20}$$

where ρ_1 is an arbitrary real number.

Assume that now we add to system (5.19) the equation

$$x_2 \;-x_4 \;=\; 0. \tag{5.21}$$

The new solution

$$(x_1,x_2,x_3,x_4)=(1,1,1,1) \tag{5.22}$$

is obtained by an extra pivoting step using the new vector $(0,1,0,-1,0)$ (see Table 5.6).

Finally if in the system (5.19) we eliminate variable $\mathbf{x}_4$, the new solution

$$(x_1,x_2,x_3)=(0,2,0) \tag{5.23}$$

can be obtained by introducing the new equation $x_4=0$, which is equivalent to this elimination. Using a new pivoting step with the vector $(0,0,0,1,0)$, we get the results in Table 5.6, and the solution in (5.23).

TABLE 5.6
New pivoting transformation after adding (5.21) *and after removing variable* x_4 *in system* (5.19).

Iteration 4			*Final*
$\mathbf{a}_4$	$\mathbf{v}_1^4$	$\mathbf{v}_2^4$	$\mathbf{v}_1$
0	**1**	0	1
1	**−1**	2	1
0	**1**	0	1
−1	**1**	0	1
0	**0**	1	1
$\mathbf{t}^4$	**−2**	2	

Iteration 4			*Final*
$\mathbf{a}_4$	$\mathbf{v}_1^4$	$\mathbf{v}_2^4$	$\mathbf{v}_1$
0	**1**	0	0
0	**−1**	2	2
0	**1**	0	0
1	**1**	0	0
0	**0**	1	1
$\mathbf{t}^4$	**1**	0	

5.10. Deciding whether or not a linear system of equations is compatible. In this section we show how to apply the orthogonal methods to analyze the compatibility of a given system of equations.

System (5.15) can be written as

$$x_1 \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix} + x_2 \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix} + \cdots + x_n \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \tag{5.24}$$

Expression (5.24) shows that the vector $(b_1, \ldots, b_m)^T$ is a linear combination of the set of column vectors

$$\{(a_{11}, \ldots, a_{m1})^T, (a_{12}, \ldots, a_{m2})^T, \ldots, (a_{1n}, \ldots, a_{mn})^T\}$$

of the system matrix $\mathbf{A}$. Thus, the compatibility problem reduces to that of section 5.6.

Thus, analyzing the compatibility of the system of equations (5.15) reduces to finding the orthogonal complement $\mathcal{L}\{\mathbf{w}_1, \ldots, \mathbf{w}_p\}$ of $\mathcal{L}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ and checking whether or not $\mathbf{b}\mathbf{W}^T = \mathbf{0}$.

The computational process arising from this procedure has a complexity which coincides exactly with that for the Gauss elimination procedure. However, it has one important advantage: $\mathbf{W}$ is independent of $\mathbf{b}$ and so we can analyze the compatibility of any symbolic vector $\mathbf{u}_{n+1}$ without extra computations.

Example 11 (compatibility of a linear system of equations). Suppose that we are interested in determining the conditions under which the system of equations

$$\begin{aligned} 2x_1 - x_2 + x_3 &= a, \\ x_1 - x_3 &= 3a, \\ x_2 - 3x_3 &= b \end{aligned} \tag{5.25}$$

is compatible. Then, using Algorithm 1, we get (see Table 5.7)

$$\mathbf{W} = \mathcal{L}\left\{(1, -2, 1)^T\right\}, \tag{5.26}$$

which implies the following compatibility condition:

$$\mathbf{w}_1(a, 3a, b)^T = (1, -2, 1)(a, 3a, b)^T = 0 \Rightarrow b - 5a = 0. \tag{5.27}$$

TABLE 5.7
Pivoting process to determine the orthogonal complement of the linear subspace generated by the columns of A.

Iteration 1				*Iteration* 2				*Iteration* 3			
2	**1**	0	0	–1	*1/2*	**–1/2**	0	1	*0*	*–1*	1
1	**0**	1	0	0	*0*	**1**	0	–1	*1*	*2*	–2
0	**0**	0	1	1	*0*	**0**	1	–3	*0*	*0*	1
$\mathbf{t}^1$	**2**	1	0	$\mathbf{t}^2$	–1/2	**1/2**	1	$\mathbf{t}^3$	–1	–3	0

6. Conclusions. A pivoting transformation, based on the orthogonality concept, has been discussed and some of its applications to solve common linear algebra problems have been given. The main advantage of the suggested method with respect to the Gauss elimination method is that the intermediate results arising in the solution process are easily interpretable. This leads to immediate methods to update solutions of several problems, such as calculating the inverse or the determinant of a matrix, solving system of linear equations, etc., when small changes are done (changes in rows, columns, and/or variables). When the method is applied to inverting a matrix and calculating its determinant, not only is the inverse of the final matrix obtained, but also the inverses and determinants of all its block main diagonal matrices, without extra computations.

Acknowledgment. We thank the referee for the constructive comments that allowed an important improvement of the paper.

REFERENCES

[1] R. I. BURDEN AND J. D. FAIRES, *Numerical Analysis*, PWS, Boston, 1985.
[2] E. CASTILLO, A. COBO, A. FERNANDEZ-CANTELI, F. JUBETE, AND R. E. PRUNEDA, *Updating inverses in matrix analysis of structures,* Internat. J. Numer. Methods Engrg., 43 (1998), pp. 1479–1504.
[3] E. CASTILLO, A. COBO, F. JUBETE, AND R. E. PRUNEDA, *Orthogonal Sets and Polar Methods in Linear Algebra: Applications to Matrix Calculations, Systems of Equations and Inequalities, and Linear Programming*, John Wiley, New York, 1999.
[4] J.W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.
[6] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.
[7] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[8] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C. The Art of Scientific Computing,* Cambridge University Press, Cambridge, UK, 1985.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 682–713

AN IMPLICITLY RESTARTED SYMPLECTIC LANCZOS METHOD FOR THE SYMPLECTIC EIGENVALUE PROBLEM*

PETER BENNER† AND HEIKE FAßBENDER‡

Abstract. An implicitly restarted symplectic Lanczos method for the symplectic eigenvalue problem is presented. The Lanczos vectors are constructed to form a symplectic basis. The inherent numerical difficulties of the symplectic Lanczos method are addressed by inexpensive implicit restarts. The method is used to compute some eigenvalues and eigenvectors of large and sparse symplectic operators.

Key words. eigenvalues, symplectic Lanczos method, implicit restarting, symplectic matrix

AMS subject classifications. 65F15, 65F50, 15A18

PII. S0895479898343115

1. Introduction.

We consider the numerical solution of the real symplectic eigenvalue problem

$$Mx = \lambda x, \tag{1.1}$$

where $M \in \mathbb{R}^{2n\times 2n}$ is large and possibly sparse. A matrix M is called *symplectic* iff

$$MJM^T = J, \tag{1.2}$$

or equivalently, $M^T JM = J$, where

$$J^{2n,2n} = \begin{bmatrix} 0 & I^{n,n} \\ -I^{n,n} & 0 \end{bmatrix} \tag{1.3}$$

and $I^{n,n}$ is the $n \times n$ identity matrix. If the dimension of $J^{2n,2n}$ or $I^{n,n}$ is clear from the context, we leave off the superscript. The symplectic matrices form a group under multiplication. The eigenvalues of symplectic matrices occur in reciprocal pairs: If λ is an eigenvalue of M with right eigenvector x, then λ^{-1} is an eigenvalue of M with left eigenvector $(Jx)^T$. The computation of eigenvalues and eigenvectors of such matrices is an important task in applications like the discrete linear-quadratic regulator problem, discrete Kalman filtering, the solution of discrete-time algebraic Riccati equations, and certain large, sparse quadratic eigenvalue problems. See, e.g., [23, 24, 32, 33] for applications and further references. Symplectic matrices also occur when solving linear Hamiltonian difference systems [7].

In order to develop fast and efficient methods, the symplectic structure of the problem should be preserved and exploited. Then the method will be reliable in the sense that the computed solution will have a physical meaning, as important properties of symplectic matrices (e.g., the symmetry of the spectrum) will be preserved and not destroyed by rounding errors. Different structure-preserving methods for

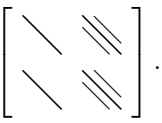
*Received by the editors August 5, 1998; accepted for publication (in revised form) by D. Calvetti May 15, 2000; published electronically October 25, 2000.
http://www.siam.org/journals/simax/22-3/34311.html

†Universität Bremen, Fachbereich 3 – Mathematik und Informatik, Zentrum für Technomathematik, 28334 Bremen, FRG (benner@math.uni-bremen.de).

‡Technische Universität München, Zentrum Mathematik, 80290 München, FRG (fassbend@ma.tum.de).

solving (1.1) have been proposed. In [28] Lin introduces the $S + S^{-1}$-transformation which can be used to compute the eigenvalues of a symplectic matrix by a structure-preserving method similar to Van Loan's square-reduced method for the Hamiltonian eigenvalue problem [42]. Flaschka, Mehrmann, and Zywietz show in [16] how to construct structure-preserving methods based on the SR method [11, 12, 30]. Patel [38, 37] and Mehrmann [31] developed structure-preserving algorithms for the symplectic generalized eigenproblem $L - \lambda N$ where $L, N \in \mathbb{R}^{2n\times 2n}$ and $LJL^T = NJN^T$.

Recently, research by Banse [2] and by Banse and Bunse-Gerstner [3] presented a new condensed form for symplectic matrices. The $2n \times 2n$ condensed matrix is symplectic, contains $8n - 4$ nonzero entries, and is determined by $4n - 1$ parameters. This condensed form, called *symplectic butterfly form*, can be depicted as a symplectic matrix of the following form:



Once the reduction of a symplectic matrix to butterfly form is achieved, the SR algorithm [11, 12, 30] is a suitable tool for computing the eigenvalues/eigenvectors of a symplectic matrix. The SR algorithm preserves the butterfly form in its iterations and can be rewritten in a parameterized form that works with the $4n-1$ parameters instead of the $(2n)^2$ matrix elements in each iteration. Hence, the symplectic structure, which will be destroyed in the numerical process due to roundoff errors, can be restored in each iteration for this condensed form. An analysis of the butterfly SR algorithm can be found in [2, 5, 6].

In [2, 3] an elimination process for computing the butterfly form of a symplectic matrix is given which uses elementary unitary symplectic transformations as well as nonunitary symplectic transformations. Unfortunately, this approach is not suitable when dealing with large and sparse symplectic matrices, as an elimination process cannot make full use of the sparsity. Hence, symplectic Lanczos methods which create the symplectic butterfly form if no breakdown occurs are derived in [2, 5]. Given $v_1 \in \mathbb{R}^{2n}$ and a symplectic matrix $M \in \mathbb{R}^{2n\times 2n}$, these Lanczos algorithms produce a matrix $S^{2n,2k} = [v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k] \in \mathbb{R}^{2n\times 2k}$ which satisfies a recursion of the form

$$MS^{2n,2k} = S^{2n,2k}B^{2k,2k} + r_{k+1}e_{2k}^T, \tag{1.4}$$

where $B^{2k,2k}$ is a butterfly matrix of order $2k \times 2k$, and the columns of $S^{2n,2k}$ are orthogonal with respect to the indefinite inner product defined by J (1.3). The latter property will be called *J-orthogonality* throughout this paper. The residual r_{k+1} depends on v_{k+1} and w_{k+1}; hence, $(S^{2n,2k})^T J r_{k+1} = 0$. Such a symplectic Lanczos method will suffer from the well-known numerical difficulties inherent to any Lanczos method for unsymmetric matrices. In [2], a symplectic look-ahead Lanczos algorithm is presented which overcomes breakdown by giving up the strict butterfly form. Unfortunately, so far there do not exist eigenvalue methods that can make use of that special reduced form. Standard eigenvalue methods such as QR or SR algorithms have to be employed, resulting in a full symplectic matrix after only a few iteration steps.

A different approach to deal with the numerical difficulties of the Lanczos process is to modify the starting vectors by an implicitly restarted Lanczos process (see

the fundamental work in [10, 39]); for the unsymmetric eigenproblem the implicitly restarted Arnoldi method has been implemented very successfully; see [26]. The problems are addressed by fixing the number of steps in the Lanczos process at a prescribed value k which depends upon the required number of approximate eigenvalues. J-orthogonality of the k Lanczos vectors is secured by re-J-orthogonalizing these vectors when necessary. The purpose of the implicit restart is to determine initial vectors such that the associated residual vectors are tiny. Given (1.4), an implicit Lanczos restart computes the Lanczos factorization

$$M\breve{S}^{2k} = \breve{S}^{2k}\breve{B}^{2k,2k} + \breve{r}_{k+1}e_{2k}^T,$$

which corresponds to the starting vector

$$\breve{v}_1 = p(M)v_1$$

(where $p(M) \in \mathbb{R}^{2n\times 2n}$ is a polynomial) without having to explicitly restart the Lanczos process with the vector $\breve{v}_1$. Such an implicit restarting mechanism is derived here analogous to the technique introduced in [20, 39].

Such an implicitly restarted symplectic Lanczos method has been developed by the authors for the Hamiltonian eigenproblem in [4], that is for the eigenproblem $Hx = \lambda x$, where $H \in \mathbb{R}^{2n\times 2n}$ and $(HJ)^T = HJ$. While in [4] the focus was on deriving the method and on discussing its possible application to compute a low-rank-approximation to the solution of continuous-time algebraic Riccati equations, here we derive a theory for the implicitly restarted symplectic Lanczos method for the symplectic eigenvalue problem similar to the one derived in [39, 10] for the implicitly restarted Arnoldi and Lanczos methods. Unfortunately, the Hamiltonian and the symplectic eigenproblems are (despite common belief) quite different. The symplectic eigenproblem is much more difficult than the Hamiltonian one. The relation between the two eigenproblems is best described by comparing it with the relation between symmetric and orthogonal eigenproblems or the Hermitian and unitary eigenproblems. In all these cases, the underlying algebraic structures are an algebra and a group acting on this algebra. For the algebra (Hamiltonian, symmetric, Hermitian matrices), the structure is explicit, i.e., can be read off the matrix by viewing it. In contrast, the structure of a matrix contained in a group (symplectic, orthogonal, unitary matrices) is given only implicitly. It is very difficult to make this structure explicit. (For unitary matrices, this can be done using Schur parameter pencils; for symplectic matrices, this can be achieved using the symplectic butterfly pencils.) If the "group" eigenproblem is to be solved using a method that exploits the given structure, then this is relatively easy for orthogonal or unitary matrices as one works with the standard scalar product. Additional difficulties for the symplectic problem arise from the fact that one has to work with an indefinite inner product. Moreover, it is important to note that the condensed form used for deriving the symplectic Lanczos method for the symplectic eigenproblem differs from the one used for the Hamiltonian eigenproblem. To the best of our knowledge, these forms cannot be transformed into each other using some Moebius transform, like, e.g., the Cayley transformation when turning a Hamiltonian into a symplectic matrix and vice versa. The results in this paper explicitly rely on the symplectic butterfly form, hence they could not be derived from similar results for the Hamiltonian J-Hessenberg form.

Section 2 reviews the symplectic butterfly form and some of its properties that will be helpful for analyzing the symplectic Lanczos method which reduces a symplectic matrix to butterfly form. This symplectic Lanczos method is presented in section 3.

Further, that section is concerned with finding conditions for the symplectic Lanczos method terminating prematurely such that an invariant subspace associated with certain desired eigenvalues is obtained. We will also consider the important question of determining stopping criteria. The implicitly restarted symplectic Lanczos method itself is derived in section 4. Numerical properties of the proposed algorithm are discussed in section 5. In section 6, we present some preliminary numerical examples.

2. The symplectic butterfly form. A symplectic matrix

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} \diagdown & \diagdown\!\diagdown \\ \diagdown & \diagdown\!\diagdown \end{bmatrix}, \qquad \text{where } B_{ij} \in \mathbb{R}^{n\times n},$$

is called a *butterfly* matrix if B_{11} and B_{21} are diagonal, and B_{12} and B_{22} are tridiagonal. Banse in [2] and Banse and Bunse-Gerstner in [3] showed that for every symplectic matrix M, there exist numerous symplectic matrices S such that $B = S^{-1}MS$ is a symplectic butterfly matrix. In [2], an elimination process for computing the butterfly form of a symplectic matrix is presented (see also [5]).

In [5], an *unreduced butterfly matrix* is introduced in which the lower right tridiagonal matrix is unreduced, that is, the subdiagonal elements of B_{22} are nonzero. Using the definition of a symplectic matrix, one easily verifies that if B is an unreduced butterfly matrix, then B_{21} is nonsingular. This allows the decomposition of B into two simpler symplectic matrices:

$$B = \begin{bmatrix} B_{21}^{-1} & B_{11} \\ 0 & B_{21} \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & T \end{bmatrix} = \begin{bmatrix} \diagdown & \diagdown \\ 0 & \diagdown \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & \diagdown\!\diagdown \end{bmatrix}, \tag{2.1}$$

where $T = B_{21}^{-1}B_{22}$ is tridiagonal and symmetric. Hence $4n-1$ parameters that determine the symplectic matrix can be read off directly. The unreduced butterfly matrices play a role analogous to that of unreduced Hessenberg matrices in the standard QR theory [2, 5, 6].

We will frequently make use of the decomposition (2.1) and will denote it by

$$B_1 = \begin{bmatrix} B_{21}^{-1} & B_{11} \\ 0 & B_{21} \end{bmatrix} = \left[\begin{array}{ccc|ccc} a_1^{-1} & & & b_1 & & \\ & \ddots & & & \ddots & \\ & & a_n^{-1} & & & b_n \\ \hline & & & a_1 & & \\ & & & & \ddots & \\ & & & & & a_n \end{array}\right], \tag{2.2}$$

$$B_2^{-1} = \begin{bmatrix} 0 & -I \\ I & B_{21}^{-1}B_{22} \end{bmatrix} = \left[\begin{array}{ccc|ccc} & & & -1 & & \\ & & & & \ddots & \\ & & & & & -1 \\ \hline 1 & & & c_1 & d_2 & \\ & \ddots & & d_2 & \ddots & \ddots \\ & & 1 & & \ddots & d_n \\ & & & & d_n & c_n \end{array}\right], \tag{2.3}$$

$$B = \left[\begin{array}{cccc|cccc} b_1 & & & & b_1c_1 - a_1^{-1} & b_1d_2 & & \\ & \ddots & & & b_2d_2 & \ddots & \ddots & \\ & & \ddots & & & \ddots & \ddots & b_{n-1}d_n \\ & & & b_n & & & b_nd_n & b_nc_n - a_n^{-1} \\ \hline a_1 & & & & a_1c_1 & a_1d_2 & & \\ & \ddots & & & a_2d_2 & \ddots & \ddots & \\ & & \ddots & & & \ddots & \ddots & a_{n-1}d_n \\ & & & a_n & & & a_nd_n & a_nc_n \end{array}\right]. \tag{2.4}$$

Remark 2.1 (see [5]).

(a) Any unreduced butterfly matrix is similar to an unreduced butterfly matrix with $b_i = 1$, $|a_i| = 1$ for $i = 1, \ldots, n$, and $\operatorname{sign}(a_i) = \operatorname{sign}(d_i)$ for $i = 2, \ldots n$.

(b) We will have deflation if $d_j = 0$ for some j. Then the eigenproblem can be split into two smaller ones with unreduced symplectic butterfly matrices.

Eigenvalues and eigenvectors of symplectic butterfly matrices can be computed efficiently by the SR algorithm [8], which is a QR-like algorithm in which the QR decomposition is replaced by the SR decomposition. Almost every matrix $A \in \mathbb{R}^{2n\times 2n}$ can be decomposed into a product $A = SR$, where S is symplectic and R is J-triangular, that is

$$R = \left[\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array}\right] = \left[\begin{array}{cc} \diagdown & \diagdown \\ 0 \diagdown & \diagdown \\ \quad\ddots & \\ \quad 0 & \end{array}\right],$$

where all submatrices $R_{ij} \in \mathbb{R}^{n\times n}$ are upper triangular, and R_{21} is strictly upper triangular [13]. In the following a matrix $D \in \mathbb{R}^{2n\times 2n}$ will be called *trivial* if it is both symplectic and J-triangular. D is trivial iff it has the form

$$D = \left[\begin{array}{cc} C & F \\ 0 & C^{-1} \end{array}\right],$$

where C and F are diagonal matrices with C nonsingular.

If the SR *decomposition* $A = SR$ exists, then other SR decompositions of A can be built from it by passing trivial factors back and forth between S and R. That is, if D is a trivial matrix, $\tilde{S} = SD$ and $\tilde{R} = D^{-1}R$, then $A = \tilde{S}\tilde{R}$ is another SR decomposition of A. If A is nonsingular, then this is the only way to create other SR decompositions. In other words, the SR decomposition is unique up to trivial factors.

The SR algorithm is an iterative algorithm that performs an SR decomposition at each iteration. If B is the current iterate, then a (rational) function q is chosen (such that $q(B) \in \mathbb{R}^{2n\times 2n}$) and the SR decomposition of $q(B)$ is formed, if possible:

$$q(B) = SR.$$

Then the symplectic factor S is used to perform a similarity transformation on B to yield the next iterate, which we will call $\widehat{B}$:

$$\widehat{B} = S^{-1}BS. \tag{2.5}$$

If $\operatorname{rank}(q(B)) = 2n$ and B is an unreduced symplectic butterfly matrix, then so is $\widehat{B}$ in (2.5) [2, 3]. If rank $(p(B)) = 2n - \nu =: 2k$ and B is an unreduced symplectic

butterfly matrix, then $\widehat{B}$ in (2.5) is of the form (see [5])

$$\widehat{B} = \left[\begin{array}{c|c} \diagdown & /\!/ \\ \hline /\!/ & \diagdown \end{array}\right] = \left[\begin{array}{cc|cc} \widehat{B}_{11} & & \widehat{B}_{13} & \\ & \widehat{B}_{22} & & \widehat{B}_{24} \\ \hline \widehat{B}_{31} & & \widehat{B}_{33} & \\ & \widehat{B}_{42} & & \widehat{B}_{44} \end{array}\right] \begin{array}{l} \}k \\ \}n-k \\ \}k \\ \}n-k \end{array} \quad , \tag{2.6}$$

$$\underbrace{}_{k} \quad \underbrace{}_{n-k} \quad \underbrace{}_{k} \quad \underbrace{}_{n-k}$$

where

- $\left[\begin{smallmatrix} \widehat{B}_{11} & \widehat{B}_{13} \\ \widehat{B}_{31} & \widehat{B}_{33} \end{smallmatrix}\right]$ is a symplectic butterfly matrix and
- the eigenvalues of $\left[\begin{smallmatrix} \widehat{B}_{22} & \widehat{B}_{24} \\ \widehat{B}_{42} & \widehat{B}_{44} \end{smallmatrix}\right]$ are just the ν shifts that are eigenvalues of B.

An algorithm for explicitly computing S and R is presented in [9]. As with explicit QR steps, the expense of explicit SR steps comes from the fact that $q(B)$ has to be computed explicitly. A preferred alternative is the implicit SR step, an analogue to the Francis QR step [17, 19, 22]. As the implicit SR step is analogous to the implicit QR step, this technique will not be discussed here (see [5, 6] for details).

A natural way to choose the function q is to choose a polynomial $p_2(\lambda) = (\lambda - \mu)(\lambda - \mu^{-1})$ for $\mu \in \mathbb{R}$ (or $\mu \in \mathbb{C}, |\mu| = 1$) or $p_4(\lambda) = (\lambda-\mu)(\lambda-\mu^{-1})(\lambda-\overline{\mu})(\lambda-\overline{\mu}^{-1})$ for $\mu \in \mathbb{C}$ as these choices make use of the symmetries of the spectrum of symplectic matrices. But, as explained in [6], a better choice is a Laurent polynomial to drive the SR step. For example, instead of $p_4(\lambda)$ we will use

$$q_4(\lambda) = p_4(\lambda)\lambda^{-2} = (\lambda+\lambda^{-1})^2 - (\mu+\mu^{-1}+\overline{\mu}+\overline{\mu}^{-1})(\lambda+\lambda^{-1}) + (\mu+\mu^{-1})(\overline{\mu}+\overline{\mu}^{-1}) - 2.$$

This reduces the size of the bulges that are introduced, thereby decreasing the number of computations required per iteration. Moreover, the use of Laurent polynomials improves the convergence and stability properties of the algorithm by effectively treating each reciprocal pair of eigenvalues as a unit. Using a generalized Rayleigh-quotient strategy, the butterfly SR algorithm is typically cubically convergent [6].

The right eigenvectors of unreduced butterfly matrices have the following property which will be helpful when analyzing the symplectic Lanczos method introduced in the next section.

LEMMA 2.2. *Suppose that $B \in \mathbb{R}^{2n\times 2n}$ is an unreduced butterfly matrix as in (2.4). If $Bx = \lambda x$ with $x \neq 0$, then $e_{2k}^T x \neq 0$.*

In order to prove this lemma we need the following definition. Let P_n be the permutation matrix

$$P_n := [e_1, e_3, \ldots, e_{2n-1}, e_2, e_4, \ldots, e_{2n}] \in \mathbb{R}^{2n\times 2n}. \tag{2.7}$$

If the dimension of P_n is clear from the context, we leave off the subscript.

Proof of Lemma 2.2. The proof is by induction on the size of B. The entries of the eigenvector x will be denoted by x_i; $x = [x_1, x_2, \ldots, x_{2n}]^T$.

Suppose that $n = 2$. The second and fourth row of $Bx = \lambda x$ yield

$$b_2 x_2 + b_2 d_2 x_3 + (b_2 c_2 - a_2^{-1}) x_4 = \lambda x_2, \tag{2.8}$$

$$a_2 x_2 + a_2 d_2 x_3 + a_2 c_2 x_4 = \lambda x_4. \tag{2.9}$$

Since B is unreduced, we know that $a_2 \neq 0$ and $d_2 \neq 0$. If $x_4 = 0$, then from (2.9) we obtain

$$x_2 + d_2 x_3 = 0, \tag{2.10}$$

while (2.8) gives $b_2(x_2 + d_2 x_3) = \lambda x_2$. Using (2.10) we obtain $x_2 = 0$, and further, $x_3 = 0$.

The third row of $Bx = \lambda x$ gives

$$a_1 x_1 + a_1 c_1 x_3 + a_1 d_2 x_4 = \lambda x_3.$$

Using $x_2 = x_3 = x_4 = 0$ and $a_1 \neq 0$, since B is unreduced, we obtain $x_1 = 0$. Thus $x = 0$, which contradicts the assumption $x \neq 0$.

Assume that the lemma is true for matrices of order $2(n-1)$. Let $B^{2n,2n} \in \mathbb{R}^{2n\times 2n}$ be an unreduced butterfly matrix. For simplicity we will consider the permuted equation $B_P^{2n,2n} x_P = \lambda x_P$, where $B_P^{2n,2n} = PB^{2n,2n}P^T$ and $x_P = Px$. Partition $B_P^{2n,2n}, x_P$ as

$$B_P^{2n,2n} = \left[\begin{array}{c|cc} B_P^{2(n-1),2(n-1)} & 0 & d_n(b_{n-1}e_{2n-3} + a_{n-1}e_{2n-2}) \\ \hline b_n d_n e_{2n-2}^T & b_n & b_n c_n - a_n^{-1} \\ a_n d_n e_{2n-2}^T & a_n & a_n c_n \end{array}\right],$$

$$x_P = \begin{bmatrix} y \\ \tilde{x}_{2n-1} \\ \tilde{x}_{2n} \end{bmatrix},$$

where $B_P^{2(n-1),2(n-1)} \in \mathbb{R}^{(2n-2)\times(2n-2)}$ is an unreduced butterfly matrix and $y \in \mathbb{R}^{2n-2}$. Suppose $x_{2n} = \tilde{x}_{2n} = 0$. This implies

$$d_n y_{2n-2} + \tilde{x}_{2n-1} = 0 \tag{2.11}$$

since $a_n \neq 0$ as $B^{2n,2n}$ is unreduced. Further we have

$$b_n(d_n y_{2n-2} + \tilde{x}_{2n-1}) = \lambda \tilde{x}_{2n-1}.$$

Hence, using (2.11) we get $\tilde{x}_{2n-1} = 0$. This implies $B_P^{2n-1,2n-1} y = \lambda y$. Using $\tilde{x}_{2n-1} = \tilde{x}_{2n} = 0$ we further obtain from (2.11) $y_{2n-2} = 0$. This is a contradiction, because by induction hypothesis $e_{2n-2}^T y \neq 0$. □

Remark 2.3. If $Bx = \lambda x$, then $(Jx)^T B = \lambda^{-1}(Jx)^T$. Let y be the right eigenvector of B to λ^{-1}, i.e., $By = \lambda^{-1} y$, then $(Jy)^T B = \lambda (Jy)^T$. From Lemma 2.2 it follows that $e_{2n}^T y \neq 0$, hence the nth component of the left eigenvector of B corresponding to λ is $\neq 0$.

3. A symplectic Lanczos method for symplectic matrices. In this section, we review the symplectic Lanczos method to compute the butterfly form (2.4) for a symplectic matrix M derived in [5]. The usual unsymmetric Lanczos algorithm generates two sequences of vectors. Due to the symplectic structure of M it is easily seen that one of the two sequences can be eliminated here, and thus work and storage can essentially be halved. (This property is valid for a broader class of matrices; see [18].) Further, this section is concerned with finding conditions for the symplectic Lanczos method terminating prematurely such that an invariant subspace associated with certain desired eigenvalues is obtained. Finally, we will consider the important question of determining stopping criteria.

In order to simplify the notation we use in the following sections permuted versions of M and B as in the previous section. Let

$$M_P = PMP^T, \quad B_P = PBP^T, \quad S_P = PSP^T, \quad J_P = PJP^T$$

with the permutation matrix P as in (2.7).

3.1. The symplectic Lanczos factorization. We want to compute a symplectic matrix S such that S transforms the symplectic matrix M to a symplectic butterfly matrix B; in the permuted version $MS = SB$ yields

$$M_P S_P = S_P B_P. \tag{3.1}$$

Equivalently, as $B = B_1 B_2^{-1}$, we can consider

$$M_P S_P (B_2)_P = S_P (B_1)_P, \tag{3.2}$$

where

$$(B_1)_P = \left[\begin{array}{cc|cc|c|cc} a_1^{-1} & -b_1 & & & & & \\ 0 & a_1 & & & & & \\ \hline & & \ddots & & & & \\ & & & \ddots & & & \\ \hline & & & & a_n^{-1} & -b_n \\ & & & & 0 & a_n \end{array}\right], \tag{3.3}$$

$$(B_2)_P = \left[\begin{array}{cc|cc|c|cc} c_1 & 1 & d_2 & 0 & & & \\ -1 & 0 & 0 & 0 & & & \\ \hline d_2 & 0 & c_2 & 1 & \ddots & & \\ 0 & 0 & -1 & 0 & & \ddots & \\ \hline & & \ddots & & \ddots & & d_n \quad 0 \\ & & & \ddots & & \ddots & 0 \quad 0 \\ \hline & & & & d_n & 0 & c_n \quad 1 \\ & & & & 0 & 0 & -1 \quad 0 \end{array}\right]. \tag{3.4}$$

The structure-preserving Lanczos method generates a sequence of permuted symplectic matrices

$$S_P^{2n,2k} = [v_1, w_1, v_2, w_2, \ldots, v_k, w_k] \in \mathbb{R}^{2n \times 2k},$$

satisfying

$$M_P S_P^{2n,2k} = S_P^{2n,2k} B_P^{2k,2k} - d_{k+1}(b_{k+1} v_{k+1} + a_{k+1} w_{k+1}) e_{2k}^T, \tag{3.5}$$

where $B_P^{2k,2k} = P_k B^{2k,2k} {P_k}^T$ is a permuted $2k \times 2k$ symplectic butterfly matrix.

The vector $r_{k+1} := d_{k+1}(b_{k+1} v_{k+1} + a_{k+1} w_{k+1})$ is the *residual vector* and is J_P-orthogonal to the columns of $S_P^{2n,2k}$, the *Lanczos vectors*. The matrix $B_P^{2k,2k}$ is the J_P-orthogonal projection of M_P onto the range of $S_P^{2n,2k}$,

$$B_P^{2k,2k} = J_P^{2k,2k} (S_P^{2n,2k})^T J_P M_P S_P^{2n,2k}.$$

Here $J_P^{2k,2k}$ denotes a permuted $2k \times 2k$ matrix J of the form (1.3). Equation (3.5) defines a length $2k$ Lanczos factorization of M_P. If the residual vector r_{k+1} is the zero vector, then (3.5) is called a *truncated Lanczos factorization* when $k < n$. Note that r_{n+1} must vanish since $(S_P^{2n,2n})^T J_P r_{n+1} = 0$ and the columns of $S_P^{2n,2n}$ form a J_P-orthogonal basis for $\mathbb{R}^{2n}$. In this case the symplectic Lanczos method computes a reduction to permuted butterfly form.

The symplectic Lanczos factorization is, up to multiplication by a trivial matrix, specified by the starting vector v_1 (see [5, Theorem 4.1]).

Let $S_P = [v_1, w_1, v_2, w_2, \ldots, v_n, w_n]$. For a given v_1, a Lanczos method constructs the matrix S_P columnwise from the equations

$$M_P S_P (B_2)_P e_j = S_P (B_1)_P e_j, \qquad j = 1, 2, \ldots.$$

From this we obtain the algorithm given in Table 3.1 (for a more detailed discussion, see [5]).

TABLE 3.1
Symplectic Lanczos method.

Algorithm : Symplectic Lanczos method

Choose an initial vector $\widetilde{v}_1 \in \mathbb{R}^{2n}, \widetilde{v}_1 \neq 0$.
Set $v_0 = 0 \in \mathbb{R}^{2n}$.
Set $d_1 = ||\widetilde{v}_1||_2$ and $v_1 = \frac{1}{d_1}\widetilde{v}_1$.

for m = 1, 2, ... do

(update of w_m)

set

$\widetilde{w}_m = M_P v_m - b_m v_m$
$a_m = v_m^T J_P M_P v_m$
$w_m = \frac{1}{a_m}\widetilde{w}_m$

(computation of c_m)

$c_m = a_m^{-1} v_m^T J_P M_P^{-1} w_m$

(update of v_{m+1})

$\widetilde{v}_{m+1} = -d_m v_{m-1} - c_m v_m + w_m + a_m^{-1} M_P^{-1} v_m$
$d_{m+1} = ||\widetilde{v}_{m+1}||_2$
$v_{m+1} = \frac{1}{d_{m+1}}\widetilde{v}_{m+1}$

Remark 3.1. Using the derived formulae for w_{k+1}, the residual term $r_{k+1} = d_{k+1}(b_{k+1}v_{k+1} + a_{k+1}w_{k+1})$ can be expressed as

$$r_{k+1} = d_{k+1} M_P v_{k+1}.$$

There is still some freedom in the choice of the parameters that occur in this algorithm. Essentially, the parameters b_m can be chosen freely. Here we set $b_m = 1$. Likewise, a different choice of the parameters a_m, d_m is possible.

Note that $M_P^{-1} = -J_P M_P^T J_P$ since M is symplectic. Thus $M_P^{-1} v_m$ is just a matrix-vector product with the transpose of M_P. Making use of this, the algorithm

can be rewritten such that only one matrix-vector product is required for each computed Lanczos vector w_m or v_m. Thus an efficient implementation of this algorithm requires $6n + (4nz + 32n)k$ flops[1], where nz is the number of nonzero elements in M_P and $2k$ is the number of Lanczos vectors computed (that is, the loop is executed k times). The algorithm as given in Table 3.1 computes an odd number of Lanczos vectors; for a practical implementation one has to omit the computation of the last vector v_{k+1} (or one has to compute an additional vector w_{k+1}).

In the symplectic Lanczos method as given above we have to divide by parameters that may be zero or close to zero. If the normalization parameter d_{m+1} is zero, the corresponding vector $\widetilde{v}_{m+1}$ is the zero vector. In this case, a J_P-orthogonal invariant subspace of M_P or equivalently, a symplectic invariant subspace of M is detected. By redefining $\widetilde{v}_{m+1}$ to be any vector satisfying

$$v_j^T J_P \widetilde{v}_{m+1} \;=\; 0, \qquad\qquad w_j^T J_P \widetilde{v}_{m+1} \;=\; 0,$$

for $j = 1, \ldots, m$, the algorithm can be continued. The resulting butterfly matrix is no longer unreduced; the eigenproblem decouples into two smaller subproblems. In the case in which $d_{m+1} \approx 0$, a good approximation to a symplectic invariant subspace of M may have been found (if $||S_P^{2n,2m}||$ is large, then d_{m+1} can not be trusted; see section 3.3 for a discussion); then one can proceed as described above. In the case in which $\widetilde{w}_m$ is zero (close to zero), an invariant subspace of M_P with dimension $2m-1$ is found (may be found). From Table 3.1 it is easy to see that the parameter a_m will be (close to) zero if $\widetilde{w}_m = 0$ ($\widetilde{w}_m \approx 0$). We further obtain from Table 3.1 that in this case $M_P v_m = b_m v_m$, i.e., b_m is an eigenvalue of M_P with corresponding eigenvector v_m. (In the case in which $\widetilde{w}_m \approx 0$, we have $M_P v_m \approx b_m v_m$.) Due to the symmetry of the spectrum of M, we also have that $1/b_m$ is an eigenvalue of M. Computing an eigenvector y of M_P corresponding to $1/b_m$, we can try to augment the $(2m-1)$-dimensional invariant subspace to an M_P-invariant subspace of even dimension. If this is possible, the space can be made J_P-orthogonal by J_P-orthogonalizing y against $\{ v_1, w_1, \ldots, v_{m-1}, w_{m-1} \}$ and normalizing such that $y^T J_P v_m = 1$.

Thus if either v_{m+1} or w_{m+1} vanishes, the breakdown is benign. If $v_{m+1} \neq 0$ and $w_{m+1} \neq 0$ but $a_{m+1} = 0$, then the breakdown is serious. No reduction of the symplectic matrix to a symplectic butterfly matrix with v_1 as first column of the transformation matrix exists. On the other hand, an initial vector v_1 exists so that the symplectic Lanczos process does not encounter serious breakdown. However, determining this vector requires knowledge of the minimal polynomial of M. Thus, no algorithm for successfully choosing v_1 at the start of the computation yet exists.

Moreover, an error analysis of the symplectic Lanczos algorithm in finite-precision arithmetic analogous to the analysis for the unsymmetric Lanczos algorithm presented by Bai [1] can also be derived; see [14]. As to be expected, the computed Lanczos vectors loose $J(J_P)$-orthogonality when some Ritz values begin to converge.

3.2. Truncated symplectic Lanczos factorizations. This section is concerned with finding conditions for the symplectic Lanczos method terminating prematurely. This is a welcome event since in this case we have found an invariant symplectic subspace $S^{2n,2k}$ and the eigenvalues of $B^{2k,2k}$ are a subset of those of M. We will first discuss the conditions under which the residual vector of the symplectic Lanczos factorization will vanish at some step k. Then we will show how the residual vector

[1] Following [19], we define each floating point arithmetic operation together with the associated integer indexing as a flop.

and the starting vector are related. Finally, a result indicating when a particular starting vector generates an exact truncated factorization is given.

First, the conditions under which the residual vector of the symplectic Lanczos factorization will vanish at some step k will be discussed. From the derivation of the algorithm it is immediately clear that if no breakdown occurs, then

$$\begin{aligned}
&\operatorname{span}\{v_1,\ldots,v_{k+1},w_1,\ldots,w_k\}\\
&\qquad=\operatorname{span}\{v_1,M_P^{-1}v_1,\ldots,M_P^{-k}v_1,M_Pv_1,\ldots,M_P^kv_1\}\\
&\qquad=\operatorname{span}\{\operatorname{span}(\mathcal{K}(M_P,v_1,k))\cup\{M_P^{-k}v_1\}\},\\
&\operatorname{span}\{v_1,\ldots,v_{k+1},w_1,\ldots,w_{k+1}\}\\
&\qquad=\operatorname{span}\{v_1,M_P^{-1}v_1,\ldots,M_P^{-k}v_1,M_Pv_1,\ldots,M_P^{k+1}v_1\}\\
&\qquad=\operatorname{span}(\mathcal{K}(M_P,v_1,k+1)),
\end{aligned}$$

where $\mathcal{K}(X,v,k)=\{v,X^{-1}v,X^{-2}v,\ldots,X^{-(k-1)}v,Xv,X^2v,\ldots,X^kv\}$. Further, it is easy to see that

$$\text{(3.6)}\quad \dim\ \mathcal{K}(M_P,v_1,k)=d<2k \quad\Longrightarrow\quad \dim\ \mathcal{K}(M_P,v_1,j)=d \quad\forall\ j>k.$$

If $\dim\ \mathcal{K}(M_P,v_1,k+1)=2k+1$, then

$$w_{k+1}=M_Pv_{k+1}-b_{k+1}v_{k+1}\in\operatorname{span}\{v_1,\ldots,v_{k+1},w_1,\ldots,w_k\}.$$

Hence, there exist real scalars $\alpha_1,\ldots,\alpha_{k+1}$ and $\beta_1,\ldots,\beta_k$ such that

$$M_Pv_{k+1}=\alpha_1v_1+\cdots+\alpha_{k+1}v_{k+1}+\beta_1w_1+\cdots+\beta_kw_k.$$

Using the definition of a_{k+1} as given in Table 3.1 and the above expression we obtain because of J-orthogonality

$$\begin{aligned}
a_{k+1}&=v_{k+1}^TJ_PM_Pv_{k+1}\\
&=\alpha_1v_{k+1}^TJ_Pv_1+\cdots+\alpha_{k+1}v_{k+1}^TJ_Pv_{k+1}+\beta_1v_{k+1}^TJ_Pw_1+\cdots+\beta_kv_{k+1}^TJ_Pw_k\\
&=0.
\end{aligned}$$

As $\widetilde{w}_{k+1}=a_{k+1}w_{k+1}=M_Pv_{k+1}-b_{k+1}v_{k+1}$ (see Table 3.1) it follows that $\widetilde{w}_{k+1}=0$. This implies that an invariant subspace of M_P with dimension $2k+1$ is found.

If $\dim\ \mathcal{K}(M_P,v_1,k+1)=2k$, then $M_P^{-1}v_k\in\operatorname{span}\{v_1,\ldots,v_k,w_1,\ldots,w_k\}$. Hence

$$a_k^{-1}M_P^{-1}v_k=\alpha_1v_1+\cdots+\alpha_kv_k+\beta_1w_1+\cdots+\beta_kw_k$$

for properly chosen α_j,β_j, and from the algorithm in Table 3.1

$$\begin{aligned}
\widetilde{v}_{k+1}=\alpha_1v_1+\cdots+\alpha_{k-2}v_{k-2}+(\alpha_{k-1}-d_k)v_{k-1}+(\alpha_k-c_k)v_k\\
+\beta_1w_1+\cdots+\beta_{k-1}w_{k-1}+(\beta_k+1)w_k.
\end{aligned}$$

Since $[v_1,w_1,\ldots,v_k,w_k]^TJ_P\widetilde{v}_{k+1}=[0,\ldots,0]$ we obtain for $j<k$ and $\ell<k-2$

$$\begin{aligned}
v_j^TJ_P\widetilde{v}_{k+1}&=\beta_jv_j^TJ_Pw_j=\beta_j=0,\\
v_k^TJ_P\widetilde{v}_{k+1}&=(\beta_k+1)v_k^TJ_Pw_k=\beta_k+1=0,\\
w_\ell^TJ_P\widetilde{v}_{k+1}&=\alpha_\ell w_\ell^TJ_Pv_\ell=-\alpha_\ell=0,\\
w_{k-1}^TJ_P\widetilde{v}_{k+1}&=(\alpha_{k-1}-d_k)w_{k-1}^TJ_Pv_{k-1}=d_k-\alpha_{k-1}=0,\\
w_k^TJ_P\widetilde{v}_{k+1}&=(\alpha_k-c_k)w_k^TJ_Pv_k=c_k-\alpha_k=0.
\end{aligned}$$

Therefore, $\widetilde{v}_{k+1} = 0$, and further, $d_{k+1} = 0$. This implies that the residual vector of the symplectic Lanczos factorization will vanish at the first step k such that the dimension of $\mathcal{K}(M, v_1, k+1)$ is equal to $2k$ and hence is guaranteed to vanish for some $k \leq n$.

Next, we will discuss the relation between the residual term and the starting vector. Here, $\hat{v}_1$ will denote the first Lanczos vector after permuting it back, i.e., $\hat{v}_1 = P^T v_1$. If $\dim\ \mathcal{K}(M, \hat{v}_1, n) = 2n$, then

$$MK(M, \hat{v}_1, n) = K(M, \hat{v}_1, n) C_n,$$

where $K(M, \hat{v}_1, n) = [\hat{v}_1, M^{-1}\hat{v}_1, M^{-2}\hat{v}_1, \ldots, M^{-(n-1)}\hat{v}_1, M\hat{v}_1, M^2\hat{v}_1, \ldots, M^n\hat{v}_1] \in \mathbb{R}^{2n\times 2n}$, and C_n is a generalized companion matrix of the form

$$C_n = \left[\begin{array}{ccccc|ccccc} 0 & 1 & & & & & & & & c_1 \\ & \ddots & \ddots & & & & & & & \vdots \\ & & \ddots & 1 & & & & & & \vdots \\ & & & 0 & & & & & & c_n \\ \hline 1 & & & & & 0 & & & & c_{n+1} \\ & & & & & 1 & \ddots & & & \vdots \\ & & & & & & \ddots & 0 & & c_{2n-1} \\ & & & & & & & 1 & & c_{2n} \end{array}\right]$$

(see [2, proof of Satz 3.6]). Thus,

$$MK(M, \hat{v}_1, k) = K(M, \hat{v}_1, k) C_k + (M^{k+1}\hat{v}_1 - K(M, \hat{v}_1, k) C_k e_{2k}) e_{2k}^T. \tag{3.7}$$

Define the residual in (3.7) by

$$f_{k+1} := M^{k+1}\hat{v}_1 - K(M, \hat{v}_1, k) C_k e_{2k}. \tag{3.8}$$

Note that

$$f_{k+1} = p_k(M)\hat{v}_1, \tag{3.9}$$

where

$$p_k(\lambda) := \lambda^{k+1} - \sum_{j=0}^{k-1} (c_{k+j+1}\lambda^{j+1} + c_{j+1}\lambda^{-j}).$$

We will now show that f_{k+1} is up to scaling the residual of the length $2k$ symplectic Lanczos iteration with starting vector $\hat{v}_1$. Together with (3.9) this reveals the relation between residual and starting vectors. Since $\det\ (C_k - \lambda I) = \lambda^{2k} - \sum_{j=0}^{k-1}(c_{k-j}\lambda^j + c_{k+j+1}\lambda^{k+j})$, we obtain

$$p_k(\lambda) = \lambda^{-(k-1)} \det\ (C_k - \lambda I).$$

Let $K(M, \hat{v}_1, k) = S^{2n,2k} R$, where $S^{2n,2k} \in \mathbb{R}^{2n\times 2k}$ with J-orthogonal columns (that is, $(S^{2n,2k})^T J^{2n,2n} S^{2n,2k} = J^{2k,2k}$) and $R \in \mathbb{R}^{2k\times 2k}$ is a J-triangular matrix. Then

$S^{2n,2k}e_1 = \hat{v}_1$. The diagonal elements of R are nonzero iff the columns of $K(M, \hat{v}_1, k)$ are linear independent. Choosing

$$c = \begin{bmatrix} c_1 \\ \vdots \\ c_{2k} \end{bmatrix} = R^{-1}(-J^{2k,2k}(S^{2n,2k})^T J^{2n,2n})M^{k+1}\hat{v}_1$$

assures that $(-J^{2k,2k}(S^{2n,2k})^T J^{2n,2n})f_{k+1} = 0$. Now multiplying (3.7) from the right by R^{-1} yields

$$\begin{aligned} & MK(M, \hat{v}_1, k)R^{-1} - K(M, \hat{v}_1, k)C_k R^{-1} = f_{k+1}e_{2k}^T R^{-1} \\ \iff \quad & MS^{2n,2k} - S^{2n,2k}B = f_{k+1}e_{2k}^T / r_{2k,2k}, \end{aligned} \tag{3.10}$$

where $B = RC_kR^{-1}$ is an unreduced butterfly matrix (see [2, proof of Satz 3.6]) with the same characteristic polynomial as C_k. Equation (3.10) is a valid symplectic Lanczos recursion with starting vector $\hat{v}_1 = S^{2n,2k}e_1$ and residual vector $f_{k+1}/r_{2k,2k}$. By (3.9) and due to the essential uniqueness of the symplectic Lanczos recursion, any symplectic Lanczos recursion with starting vector $\hat{v}_1$ yields a residual vector that can be expressed as a polynomial in M times the starting vector $\hat{v}_1$.

Remark 3.2. From (3.8) it follows that if dim $\mathcal{K}(M, \hat{v}_1, k+1) \leq 2k$, then we can choose $c_1, \ldots, c_{2k}$ such that $f_{k+1} = 0$. This shows that if the Krylov subspace $\mathcal{K}(M, \hat{v}_1, k+1)$ forms a $2k$-dimensional M-invariant subspace, the residual of the symplectic Lanczos recursion will be zero after k Lanczos steps such that the columns of $S^{2n,2k}$ span a symplectic basis for the subspace $\mathcal{K}(M, \hat{v}_1, k+1)$.

The final result of this section will give necessary and sufficient conditions for a particular starting vector to generate an exact truncated factorization in a similar way as stated for the Arnoldi method in [39]. This is desirable since then the columns of $S^{2n,2k}$ form a basis for an invariant symplectic subspace of M and the eigenvalues of $B^{2k,2k}$ are a subset of those of M. Here, $\hat{v}_k$, $\hat{w}_k$ will denote the Lanczos vectors after permuting them back, i.e., $\hat{v}_k = P^T v_k$, $\hat{w}_k = P^T w_k$.

THEOREM 3.3. *Let* $MS^{2n,2k} - S^{2n,2k}B^{2k,2k} = d_{k+1}(b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1})e_{2k}^T$ *be the symplectic Lanczos factorization after* k *steps, with* $B^{2k,2k}$ *unreduced. Then* $d_{k+1} = 0$ *iff* $\hat{v}_1 = Xy$, *where* $MX = XY$ *with rank* $(X) = 2k$ *and* Y *is a Jordan matrix of order* $2k$.

Proof. If $d_{k+1} = 0$, let $B^{2k,2k}\widetilde{X} = \widetilde{X}Y$ be the Jordan canonical form of $B^{2k,2k}$ and put $X = S^{2n,2k}\widetilde{X}$. Then $MX = S^{2n,2k}B^{2k,2k}\widetilde{X} = S^{2n,2k}\widetilde{X}Y = XY$ and $\hat{v}_1 = S^{2n,2k}e_1 = S^{2n,2k}\widetilde{X}\widetilde{X}^{-1}e_1 = Xy$ with $y = \widetilde{X}^{-1}e_1$.

Suppose now that $MX = XY$, rank$(X) = 2k$, and $\hat{v}_1 = Xy$. Then $M^m X = XY^m$ for $m \in \mathbb{N}$ and it follows that

$$M^m\hat{v}_1 = M^m Xy = XY^m y \in \text{Range}(X)$$

for $m \in \mathbb{N}$. Hence by (3.6) dim $\mathcal{K}(M, \hat{v}_1, k+1) \leq$ rank $(X) = 2k$. Since $B^{2k,2k}$ is unreduced, dim $\mathcal{K}(M, \hat{v}_1, j) = 2j$ for $j = 1, \ldots, k$. Hence dim $\mathcal{K}(M, \hat{v}_1, k+1) = 2k$, and therefore, $d_{k+1} = 0$. ∎

A similar result may be formulated in terms of Schur vectors or symplectic Schur vectors (see, e.g., [32, 29] for the real symplectic Schur decomposition of a symplectic matrix). These theorems provide the motivation for the implicit restart developed in the next section. Theorem 3.3 suggests that one might find an invariant subspace by iteratively replacing the starting vector with a linear combination of approximate eigenvectors corresponding to eigenvalues of interest. Such approximations are readily available through the Lanczos factorization.

3.3. Stopping criteria. Now assume that we have performed k steps of the symplectic Lanczos method and thus obtained the identity (after permuting back)

$$MS^{2n,2k} = S^{2n,2k}B^{2k,2k} + d_{k+1}(b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1})e_{2k}^T.$$

If the norm of the residual vector is small, the $2k$ eigenvalues of $B^{2k,2k}$ are approximations to the eigenvalues of M. Numerical experiments indicate that the norm of the residual rarely becomes small by itself. Nevertheless, some eigenvalues of $B^{2k,2k}$ may be good approximations to eigenvalues of M. Let λ be an eigenvalue of $B^{2k,2k}$ with the corresponding eigenvector y. Then the vector $x = S^{2n,2k}y$ satisfies

$$\begin{aligned} ||Mx - \lambda x|| &= ||(MS^{2n,2k} - S^{2n,2k}B^{2k,2k})y|| \\ &= |d_{k+1}|\,|e_{2k}^T y|\,||b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1}||. \end{aligned} \tag{3.11}$$

The vector x is referred to as *Ritz vector* and λ as *Ritz value* of M. If the last component of the eigenvector y is sufficiently small, the right-hand side of (3.11) is small and the pair $\{\lambda, x\}$ is a good approximation to an eigenvalue-eigenvector pair of M. Note that by Lemma 2.2, $|e_{2k}^T y| > 0$ if $B^{2k,2k}$ is unreduced. The pair (λ, x) is exact for the nearby problem

$$(M - E)x = \lambda x, \qquad \text{where} \quad E = d_{k+1}(b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1})e_k^T(S^{2n,2k})^T J^{2n,2n}.$$

A small $||E||$ is not sufficient for the Ritz pair $\{\lambda, x\}$ being a good approximation to an eigenvalue-eigenvector pair of M. The advantage of using the *Ritz estimate* $|d_{k+1}|\,|e_{2k}^T y|\,||b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1}||$ is to avoid the explicit formation of the residual $(MS^{2n,2k} - S^{2n,2k}B^{2k,2k})y$ when deciding about the numerical accuracy of an approximate eigenpair.

It is well known that for nonnormal matrices the norm of the residual of an approximate eigenvector is not by itself sufficient information to bound the error in the approximate eigenvalue. It is sufficient, however, to give a bound on the distance to the nearest matrix to which the given approximation is exact. In the following, we will give a computable expression for the error. Assume that $B^{2k,2k}$ is diagonalizable, i.e., there exists a nonsingular Y such that

$$Y^{-1}B^{2k,2k}Y = \left[\begin{array}{ccc|ccc} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_k & & & \\ \hline & & & \lambda_1^{-1} & & \\ & & & & \ddots & \\ & & & & & \lambda_k^{-1} \end{array}\right] = \Lambda;$$

Y can be chosen symplectic. Let $X = S^{2n,2k}Y = [x_1 \ldots, x_{2k}]$ and denote the residual term $d_{k+1}(b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1})$ by $\hat{r}_{k+1}$. Since $MS^{2n,2k} = S^{2n,2k}B^{2k,2k} + \hat{r}_{k+1}e_{2k}^T$, it follows that

$$MS^{2n,2k}Y = S^{2n,2k}YY^{-1}B^{2k,2k}Y + \hat{r}_{k+1}e_{2k}^T Y$$

or $MX = X\Lambda + \hat{r}_{k+1}e_{2k}^T Y$. Thus,

$$Mx_i = \lambda_i x_i + y_{2k,i}\hat{r}_{k+1} \quad \text{and} \quad Mx_{k+i} = \lambda_i^{-1}x_{k+i} + y_{2k,k+i}\hat{r}_{k+1}$$

for $i = 1, \ldots, k$. The last equation can be rewritten as

$$(Jx_{k+i})^T M = \lambda_i (Jx_{k+i})^T + y_{2k,k+i}\lambda_i \hat{r}_{k+1}^T JM.$$

Using Theorem 2′ of [21] we obtain that $(\lambda_i, x_i, (Jx_{k+i})^T)$ is an eigentriplet of $M - F_{\lambda_i}$, where

$$||F_{\lambda_i}|| = \max \left\{ \frac{||\hat{r}_{k+1}||\ |y_{2k,i}|}{||x_i||}, \frac{||\hat{r}_{k+1}^T JM||\ |y_{2k,k+i}\lambda_i|}{||Jx_{k+i}||} \right\}.$$

Furthermore, when $||F_{\lambda_i}||$ is small enough, then

$$|\theta_i - \lambda_j| \leq \mathrm{cond}(\lambda_j)||F_{\lambda_i}|| + \mathcal{O}(||F_{\lambda_i}||^2),$$

where θ_i is an eigenvalue of M and $\mathrm{cond}(\lambda_j)$ is the condition number of the Ritz value λ_j:

$$\mathrm{cond}(\lambda_j) = \frac{||x_i||_2||Jx_{k+i}||_2}{|x_{k+i}^T Jx_i|} = ||x_i||_2||x_{k+i}||_2.$$

Similarly, we obtain that $\{\lambda_i^{-1}, x_{k+i}, (Jx_i)^T\}$ is an eigentriplet of $M - F_{\lambda_i^{-1}}$, where

$$||F_{\lambda_i^{-1}}||_2 = |d_{k+1}|\ \max_i \left\{ \frac{||\widehat{r}_{k+1}||_2 |y_{2k,k+i}|}{||x_{k+i}||_2}, \frac{||\widehat{r}_{k+1}^T JM||_2 |y_{2k,i}\lambda_i^{-1}|}{||Jx_i||_2} \right\}.$$

Consequently, as λ_i and λ_i^{-1} should be treated alike, the symplectic Lanczos algorithm should be continued until $||F_{\lambda_i}||_2$ and $||F_{\lambda_i^{-1}}||_2$ are small, and until $\mathrm{cond}(\lambda_j)||F_{\lambda_i}||_2$ and $\mathrm{cond}(\lambda_j)||F_{\lambda_i^{-1}}||_2$ are below a given threshold for accuracy. Note that as in the Ritz estimate, in the criteria derived here the essential quantities are $|d_{k+1}|$ and the last component of the desired eigenvectors $|y_{2k,i}|$ and $|y_{2k,k+i}|$.

4. An implicitly restarted symplectic Lanczos method. In the previous sections we have briefly mentioned two algorithms for computing approximations to the eigenvalues of a symplectic matrix M. The symplectic Lanczos algorithm is appropriate when the matrix M is large and sparse. If only a small subset of the eigenvalues is desired, the length k symplectic Lanczos factorization may suffice. The analysis in the last chapter suggests that a strategy for finding $2k$ eigenvalues in a length k factorization is to find an appropriate starting vector that forces the residual r_{k+1} to vanish. The SR algorithm, on the other hand, computes approximations to all eigenvalues and eigenvectors of M. From Theorem 4.1 in [5] (an implicit Q-theorem for the SR case) we know that in exact arithmetic, when using the same starting vector, the SR algorithm and the length n Lanczos factorization generate the same symplectic butterfly matrices (up to multiplication by a trivial matrix). Forcing the residual for the symplectic Lanczos algorithm to zero has the effect of deflating a subdiagonal element during the SR algorithm: by Remark 3.1 $r_{k+1} = -d_{k+1}M_P v_{k+1}$ and from the symplectic Lanczos process we have $d_{k+1} = \|v_{k+1}\|_2$. Hence, a zero residual implies a zero d_{k+1} such that deflation occurs for the corresponding butterfly matrix.

Our goal in this section will be to construct a starting vector that is a member of the invariant subspace of interest. Our approach is to implicitly restart the symplectic Lanczos factorization. This was first introduced by Sorensen [39] in the context of unsymmetric matrices and the Arnoldi process. The scheme is called implicit because

the updating of the starting vector is accomplished with an implicit shifted SR mechanism on $B^{2j,2j}, j \le n$. This allows us to update the starting vector by working with a symplectic matrix in $\mathbb{R}^{2j\times 2j}$ rather than in $\mathbb{R}^{2n\times 2n}$, which is significantly cheaper.

The iteration starts by extending a length k symplectic Lanczos factorization by p steps. Next, $2p$ shifts are applied to $B^{2(k+p),2(k+p)}$ using double or quadruple SR steps. The last $2p$ columns of the factorization are discarded, resulting in a length k factorization. The iteration is defined by repeating this process until convergence.

For simplicity let us first assume that $p = 1$ and that a $2n \times 2(k+1)$ matrix $S_P^{2n,2k+2}$ is known such that

$$M_P S_P^{2n,2k+2} = S_P^{2n,2k+2} B_P^{2k+2,2k+2} + r_{k+2} e_{2k+2}^T \tag{4.1}$$

as in (3.5). Let μ be a real shift and

$$q_2(B^{2k+2,2k+2}) = (B^{2k+2,2k+2} - \mu I)(B^{2k+2,2k+2} - \mu^{-1} I)(B^{2k+2,2k+2})^{-1} = SR.$$

Then (using (2.6))

$$S_P^{-1} B_P^{2k+2,2k+2} S_P$$

will be a permuted butterfly matrix and S_P is an upper triangular matrix with two additional subdiagonals.

With this we can re-express (4.1) as

$$M_P(S_P^{2n,2k+2} S_P) = (S_P^{2n,2k+2} S_P)(S_P^{-1} B_P^{2k+2,2k+2} S_P) + r_{k+2} e_{2k+2}^T S_P.$$

Defining $\breve{S}_P^{2n,2k+2} = S_P^{2n,2k+2} S_P$ and $\breve{B}_P^{2k+2,2k+2} = S_P^{-1} B_P^{2k+2,2k+2} S_P$, this yields

$$M_P \breve{S}_P^{2n,2k+2} = \breve{S}_P^{2n,2k+2} \breve{B}_P^{2k+2,2k+2} + r_{k+2} e_{2k+2}^T S_P. \tag{4.2}$$

The above equation fails to be a symplectic Lanczos factorization since the columns $2k, 2k+1, 2k+2$ of the matrix $d_{k+2}(b_{k+2} v_{k+2} + a_{k+2} w_{k+2}) e_{2k+2}^T S_P$ are nonzero. Let s_{ij} be the (i,j)th entry of S_P. The residual term in (4.2) is

$$d_{k+2}(b_{k+2} v_{k+2} + a_{k+2} w_{k+2})(s_{2k+2,2k} e_{2k}^T + s_{2k+2,2k+1} e_{2k+1}^T + s_{2k+2,2k+2} e_{2k+2}^T).$$

Rewriting (4.2) as

$$M_P \breve{S}_P^{2n,2k+2} = [\breve{S}_P^{2n,2k},\ \breve{v}_{k+1},\ \breve{w}_{k+1},\ v_{k+2},\ w_{k+2}] Z,$$

where Z is blocked as

$$\left[\begin{array}{c|c|c} \breve{B}_P^{2k,2k} & 0 & \breve{d}_{k+1}(\breve{b}_k e_{2k-1} + \breve{a}_k e_{2k}) \\ \hline \begin{matrix}\breve{b}_{k+1}\breve{d}_{k+1} e_{2k}^T \\ \breve{a}_{k+1}\breve{d}_{k+1} e_{2k}^T\end{matrix} & \begin{matrix}\breve{b}_{k+1} \\ \breve{a}_{k+1}\end{matrix} & \begin{matrix}\breve{b}_{k+1}\breve{c}_{k+1} - \breve{a}_{k+1}^{-1} \\ \breve{a}_{k+1}\breve{c}_{k+1}\end{matrix} \\ \hline \begin{matrix}d_{k+2} b_{k+2} s_{2k+2,2k} e_{2k}^T \\ d_{k+2} a_{k+2} s_{2k+2,2k} e_{2k}^T\end{matrix} & \begin{matrix}d_{k+2} b_{k+2} s_{2k+2,2k+1} \\ d_{k+2} a_{k+2} s_{2k+2,2k+1}\end{matrix} & \begin{matrix}d_{k+2} b_{k+2} s_{2k+2,2k+2} \\ d_{k+2} a_{k+2} s_{2k+2,2k+2}\end{matrix} \end{array}\right],$$

we obtain as a new Lanczos identity

$$M_P \breve{S}_P^{2n,2k} = \breve{S}_P^{2n,2k} \breve{B}_P^{2k,2k} + \breve{r}_{k+1} e_{2k}^T, \tag{4.3}$$

where

$$\breve{r}_{k+1} = \breve{d}_{k+1}(\breve{b}_{k+1}\breve{v}_{k+1} + \breve{a}_{k+1}\breve{w}_{k+1}) + d_{k+2}s_{2k+2,2k}(b_{k+2}v_{k+2} + a_{k+2}w_{k+2}).$$

Here, $\breve{a}_{k+1}$, $\breve{b}_{k+1}$, $\breve{d}_{k+1}$ denote parameters of $\breve{B}_P^{2k+2,2k+2}$, while $a_{k+2}, b_{k+2}, d_{k+2}$ are parameters of $B_P^{2k+2,2k+2}$. In addition, $\breve{v}_{k+1}, \breve{w}_{k+1}$ are the last two column vectors from $\breve{S}_P^{2n,2k+2}$, while v_{k+2}, w_{k+2} are the two last column vectors of $S_P^{2n,2k+2}$.

As the space spanned by the columns of $S^{2n,2k+2} = (P_n)^T S_P^{2n,2k+2} P_{k+1}$ is J-orthogonal, and S_P is a permuted symplectic matrix, the space spanned by the columns of $\breve{S}^{2n,2k} = (P_n)^T \breve{S}_P^{2n,2k} P_k$ is J-orthogonal. Thus (4.3) is a valid symplectic Lanczos factorization. The new starting vector is $\breve{v}_1 = \rho q_2(M_P)v_1$ for some scalar $\rho \in \mathbb{R}$. This can be seen as follows: first note that for unreduced butterfly matrices $B^{2k+2,2k+2}$ we have $q_2(B_P^{2k+2,2k+2})e_1 \neq 0$. Hence, from $q_2(B_P^{2k+2,2k+2}) = S_P R_P$ we obtain $q_2(B_P^{2k+2,2k+2})e_1 = \rho S_P e_1$ for $\rho = e_1^T R_P e_1$ as R_P is an upper triangular matrix. As $q_2(B_P^{2k+2,2k+2})e_1 \neq 0$, we have $\rho \neq 0$. Using (4.3) it follows that

$$\begin{aligned}
\breve{S}_P^{2n,2k} e_1 &= S_P^{2n,2k+2} S_P e_1 \\
&= \frac{1}{\rho} S_P^{2n,2k+2} q_2(B_P^{2k+2,2k+2}) e_1 \\
&= \frac{1}{\rho} S_P^{2n,2k+2} (B_P^{2k+2,2k+2} - \mu I)(B_P^{2k+2,2k+2} - \mu^{-1} I)(B_P^{2k+2,2k+2})^{-1} e_1 \\
&= \frac{1}{\rho} (M_P S_P^{2n,2k+2} - r_{k+2} e_{2k+2}^T - \mu S_P^{2n,2k+2})(I - \mu^{-1}(B_P^{2k+2,2k+2})^{-1}) e_1 \\
&= \frac{1}{\rho} (M_P S_P^{2n,2k+2} - \mu S_P^{2n,2k+2})(I - \mu^{-1}(B_P^{2k+2,2k+2})^{-1}) e_1
\end{aligned}$$

as $r_{k+2} e_{2k+2}^T (I - \mu^{-1}(B_P^{2k+2,2k+2})^{-1}) e_1 = 0$. Thus, using (4.3) again we get

$$\begin{aligned}
\breve{S}_P^{2n,2k} e_1 &= \frac{1}{\rho}(M_P - \mu I)(S_P^{2n,2k+2} - \mu^{-1} S_P^{2n,2k+2}(B_P^{2k+2,2k+2})^{-1}) e_1 \\
&= \frac{1}{\rho}(M_P - \mu I)(S_P^{2n,2k+2} - \mu^{-1} M_P^{-1} S_P^{2n,2k+2}) e_1 \\
&\qquad - \frac{1}{\rho} \mu^{-1} M_P^{-1} r_{k+2} e_{2k+2}^T (B_P^{2k+2,2k+2})^{-1} e_1 \\
&= \frac{1}{\rho}(M_P - \mu I)(I - \mu^{-1} M_P^{-1}) S_P^{2n,2k+2} e_1 \\
&= q_2(M_P) v_1
\end{aligned}$$

as $e_{2k+2}^T (B_P^{2k+2,2k+2})^{-1} e_1 = 0$.

Note that in the symplectic Lanczos process the vectors v_j of $S_P^{2n,2k}$ satisfy the condition $||v_j||_2 = 1$ and the parameters b_j are chosen to be one. This is no longer true for the odd-numbered column vectors of S_P generated by the SR decomposition and the parameters $\breve{b}_j$ from $\breve{B}_P^{2k,2k}$, and thus, for the new Lanczos factorization (4.3). Both properties could be forced using trivial factors. Numerical tests indicate that there is no obvious advantage in doing so.

Using standard polynomials as shift polynomials instead of using Laurent polynomials as above results in the following situation: In $p_2(B_P^{2k+2,2k+2}) = (B_P^{2k+2,2k+2} - \mu I)(B_P^{2k+2,2k+2} - \mu^{-1} I) = S_P R_P$ S_P is an upper triangular matrix with four (!) additional subdiagonals. Therefore, the residual term in (4.2) has five nonzero entries.

TABLE 4.1
k-step restarted symplectic Lanczos method.

Algorithm: k-step restarted symplectic Lanczos method

perform k steps of the symplectic Lanczos algorithm to compute $S_P^{2n,2k}$ and $B_P^{2k,2k}$
obtain the residual vector r_{k+1}
while $||r_{k+1}|| > tol$
    perform p additional steps of the symplectic Lanczos method
        to compute $S_P^{2n,2(k+p)}$ and $B_P^{2(k+p),2(k+p)}$
    select p shifts μ_i
    compute $\breve{B}_P^{2k,2k}$ and $\breve{S}_P^{2n,2k}$ via implicitly shifted SR steps
    set $S_P^{2n,2k} = \breve{S}_P^{2n,2k}$ and $B_P^{2k,2k} = \breve{B}_P^{2k,2k}$
    obtain the new residual vector r_{k+1}
end while

Hence, not the last two but the last four columns of (4.2) have to be discarded in order to obtain a new valid Lanczos factorization. That is, we would have to discard wanted information, which is avoided by using Laurent polynomials.

This technique can be extended to the quadruple shift case using Laurent polynomials as the shift polynomials as discussed in section 2. The implicit restart can be summarized as given in Table 4.1. In the course of the iteration we have to choose p shifts $\Delta = \{\mu_1, \ldots, \mu_p\}$ in order to apply $2p$ shifts: choosing a real shift μ_k implies that μ_k^{-1} is also a shift due to the symplectic structure of the problem. Hence, μ_k^{-1} is not added to Δ as the use of the Laurent polynomial q_2 guarantees that μ_k^{-1} is used as a shift once $\mu_k \in \Delta$. In case of a complex shift μ_k, $|\mu_k| = 1$, this implies that $\overline{\mu_k}$ is also a shift not added to Δ. For complex shifts μ_k, $|\mu_k| \neq 1$, we include μ_k, $\overline{\mu_k}$ in Δ.

Numerous choices are possible for the selection of the p shifts. One possibility is to choose p "exact" shifts with respect to $B_P^{2(k+p),2(k+p)}$. That is, first the eigenvalues of $B_P^{2(k+p),2(k+p)}$ are computed (by the SR algorithm), then p unwanted eigenvalues are selected. One choice for this selection might be to sort the eigenvalues by decreasing magnitude. There will be $k+p$ eigenvalues with modulus greater than or equal to 1:

$$\begin{aligned}|\lambda_1| \geq \cdots \geq |\lambda_k| \geq |\lambda_{k+1}| \geq \cdots |\lambda_{k+p}| \geq 1 \\ \geq |\lambda_{k+p}^{-1}| \geq \cdots \geq |\lambda_{k+1}^{-1}| \geq |\lambda_k^{-1}| \geq \cdots \geq |\lambda_1^{-1}|.\end{aligned}$$

Select the $2p$ eigenvalues with modulus closest to 1 as shifts. If λ_{k+1} is complex with $|\lambda_k| = |\lambda_{k+1}| \neq 1$, then we have to choose either $2p+2$ shifts or just $2p-2$ shifts, as λ_{k+1} belongs to a quadruple pair of eigenvalues of $B_P^{2(k+p),2(k+p)}$ and in order to preserve the symplectic structure either λ_k and λ_{k+1} have to be chosen or none.

A different possibility of choosing the shifts is to keep those eigenvalues that are good approximations to eigenvalues of M. That is, eigenvalues for which (3.11) is small. Again, we have to make sure that our set of shifts is complete in the sense described above.

Choosing eigenvalues of $B_P^{2(k+p),2(k+p)}$ as shifts has an important consequence for the next iterate. Assume for simplicity that $B_P^{2(k+p),2(k+p)}$ is diagonalizable. Let $\lambda(B_P^{2(k+p),2(k+p)}) = \{\theta_1, \ldots, \theta_{2k}\} \cup \{\mu_1, \ldots, \mu_{2p}\}$ be a disjoint partition of the spectrum of $B_P^{2(k+p),2(k+p)}$. Selecting the exact shifts $\mu_1, \ldots, \mu_{2p}$ in the implicit restart,

following the rules mentioned above yields a matrix

$$\breve{B}_P^{2(k+p),2(k+p)} = \begin{bmatrix} \breve{B}_P^{2k,2k} & X \\ 0 & Y \end{bmatrix},$$

where $\lambda(\breve{B}_P^{2k,2k}) = \{\theta_1, \ldots, \theta_{2k}\}$ and $\lambda(Y) = \{\mu_1, \ldots, \mu_{2p}\}$. This follows from (2.6). Moreover, the new starting vector has been implicitly replaced by the sum of $2k$ approximate eigenvectors:

$$\breve{v}_1 = S_P^{2n,2(k+p)} S_P e_1 = \frac{1}{\rho} S_P^{2n,2(k+p)} q(B_P^{2(k+p),2(k+p)}) e_1 = \frac{1}{\rho} S_P^{2n,2(k+p)} \sum_{j=1}^{2k} \zeta_j y_j,$$

where $\rho = e_1^T R_P e_1$, $B_P^{2(k+p),2(k+p)} y_j = \theta_j y_j$, and ζ_j is properly chosen. The last equation follows since $q(B_P^{2(k+p),2(k+p)})e_1$ has no component along an eigenvector of $B_P^{2(k+p),2(k+p)}$ associated with $\mu_j, 1 \le j \le 2p$. Hence $\breve{v}_1$ is a linear combination of the $2k$ Ritz vectors associated with the Ritz values that are kept:

$$\breve{v}_1 = \rho \sum_{j=1}^{2k} \zeta_j x_j, \qquad \text{where } S_P^{2n,2(k+p)} y_j = x_j.$$

It should be mentioned that the k-step restarted symplectic Lanczos method as in Table 4.1 with exact shifts builds a J-orthogonal basis for a number of generalized Krylov subspaces simultaneously. The subspace of length $2(k+p)$ generated during a restart using exact shifts contains all the Krylov subspaces of dimension $2k$ generated from each of the desired Ritz vectors; for a detailed discussion, see [15]. A similar observation for Sorensen's restarted Arnoldi method with exact shifts was made by Morgan in [34]. For a discussion of this observation see [34] or [25]. Morgan infers that "the method works on approximations to all of the desired eigenpairs at the same time, without favoring one over the other" [34, p. 1220]. This remark can also be applied to the method presented here.

Moreover, the implicitly restarted symplectic Lanczos method can be interpreted as a nonstationary subspace iteration. An analogous statement for the implicitly restarted Arnoldi method is given in [27]. Assume that we have computed

$$M_P S_P^{2n,2m} = S_P^{2n,2m} B_P^{2m,2m} + r_{m+1} e_{2m}^T, \tag{4.4}$$

a length $m = k + p$ symplectic Lanczos reduction. As p shifts for the implicit restart we have chosen $\{\mu_1, \ldots, \mu_p\}$, where the shifts are sorted such that first all the complex shifts are given so that for a shift $\mu_{2j} \in \mathbb{C}, |\mu_{2j}| \neq 1$ we have $\overline{\mu_{2j-1}} = \mu_{2j}$, and then all real and purely imaginary shifts are given. Hence, we want to apply the Laurent polynomial

$$q_{2p}(B) = (B - \mu_p I)(B - \mu_p^{-1} I)B^{-1} \cdot \ldots \cdot (B - \mu_1 I)(B - \mu_1^{-1} I)B^{-1}$$

during the implicit restart. It is fairly easy to see that

$$q_{2p}(M_P) S_P^{2n,2k} = S_P^{2n,2m} q_{2p}(B_P^{2m,2m})[e_1, e_2, \ldots, e_{2k}] \tag{4.5}$$

(see [15, Lemma 5.19]). Applying $q_{2p}(M_P)$ to the first $2k$ columns of $S_P^{2n,2m}$ is equivalent to the basis representation given by the first $2k$ columns of $S_P^{2n,2m} q_{2p}(B_P^{2m,2m})$.

Applying an implicit restart to (4.4) using the rational function q_{2p}, we essentially apply the SR algorithm with shifts $\mu_1, \mu_1^{-1}, \ldots, \mu_p, \mu_p^{-1}$ to $B_P^{2m,2m}$:

$$B_P^{2m,2m} S_P = S_P \breve{B}_P^{2m,2m}.$$

$S_P \in \mathbb{R}^{2m \times 2m}$ is a symplectic, upper triangular matrix with $m-k$ additional subdiagonals. Write S_P as $S_P = [S_P^{[1]}\ S_P^{[2]}\ S_P^{[3]}]$ with $S_P^{[1]} \in \mathbb{R}^{2m \times 2k}, S_P^{[2]} \in \mathbb{R}^{2m \times 2}, S_P^{[3]} \in \mathbb{R}^{2m \times (2m-2k-2)}$. Then

$$B_P^{2m,2m} S_P^{[1]} = [S_P^{[1]}\ S_P^{[2]}\ S_P^{[3]}] \begin{bmatrix} \breve{B}_P^{2k,2k} \\ \hline \breve{b}_{k+1}\breve{d}_{k+1} e_{2k}^T \\ \breve{a}_{k+1}\breve{d}_{k+1} e_{2k}^T \\ \hline 0 \end{bmatrix}.$$

Postmultiplying (4.4) with $S_P^{[1]}$ and using $e_{2m}^T S_P^{[1]} = 0$, which is due to the special form of S_P (upper triangular with $m-k$ additional subdiagonals), we obtain

$$\begin{aligned} M_P S_P^{2n,2m} S_P^{[1]} &= S_P^{2n,2m} B_P^{2m,2m} S_P^{[1]} + r_{m+1} e_{2m}^T S_P^{[1]} \\ &= \breve{S}_P^{2n,2k} \breve{B}_P^{2k,2k} + \breve{d}_{k+1}(\breve{b}_{k+1} S_P^{2n,2m} S_P^{[2]} e_1 + \breve{a}_{k+1} S_P^{2n,2m} S_P^{[2]} e_2) e_{2k}^T \\ &= \breve{S}_P^{2n,2k} \breve{B}_P^{2k,2k} + \breve{r}_{k+1} e_{2k}^T, \end{aligned}$$

where $\breve{S}_P^{2n,2k} = S_P^{2n,2m} S_P^{[1]}$. This is just the implicitly restarted symplectic Lanczos recursion obtained by applying one implicit restart with the Laurent polynomial q_{2p}. Applying the SR algorithm with shifts $\mu_1, \mu_1^{-1}, \ldots, \mu_p, \mu_p^{-1}$ to $B_P^{2m,2m}$ is equivalent to computing the permuted SR decomposition

$$q_{2p}(B_P^{2m,2m}) = S_P R_P.$$

Substituting this into (4.5) we obtain

$$q_{2p}(M_P) S_P^{2n,2k} = S_P^{2n,2m} S_P R_P [e_1, e_2, \ldots, e_{2k}] = \breve{S}_P^{2n,2k} \breve{R}_P,$$

where $\breve{R}_P$ is a $2k \times 2k$ upper triangular matrix. This equation describes a nonstationary subspace iteration. As one step of the implicitly restarted symplectic Lanczos process computes the new subspace spanned by the columns of $\breve{S}_P^{2n,2k}$ from $S_P^{2n,2k}$, the implicitly restarted symplectic Lanczos algorithm can be interpreted as a nonstationary subspace iteration.

In the above discussion we have assumed that the permuted SR decomposition $q(B_P^{2(k+p),2(k+p)}) = S_P R_P$ exists. Unfortunately, this is not always true. During the bulge-chase in the implicit SR step, it may happen that a diagonal element a_j of B_1 (2.2) is zero (or almost zero). In that case no reduction to symplectic butterfly form with the corresponding first column $\breve{v}_1$ does exist. In the next section we will prove that a serious breakdown in the symplectic Lanczos algorithm is equivalent to such a breakdown of the SR decomposition. Moreover, it may happen that a subdiagonal element d_j of the $(2,2)$-block of B_2^{-1} (2.3) is zero (or almost zero) such that

$$\breve{B}_P^{2(k+p),2(k+p)} = \begin{bmatrix} \breve{B}_P^{2j,2j} & \\ & \hat{B}_P \end{bmatrix}.$$

The matrix $\breve{B}_P^{2(k+p),2(k+p)}$ is split, and an invariant subspace of dimension j is found. If $j \geq k$ and all shifts have been applied, then the iteration is halted. Otherwise we can continue as in the procedure described by Sorensen in [39, Remark 3].

One important property for a stable implicitly restarted Lanczos method is that the Lanczos vectors stay bounded after possibly many implicit restarts. Neither for the symplectic Lanczos method nor for the symplectic SR algorithm can it be proved that the symplectic transformation matrix stays bounded. Hence the symplectic Lanczos vectors $S_P^{2n,2k}$ computed via an implicitly restarted symplectic Lanczos method may not stay bounded; this has to be monitored during the iteration. During the SR step on the $2k \times 2k$ symplectic butterfly matrix, all but $k-1$ transformations are orthogonal. These are known to be numerically stable. For the $k-1$ nonorthogonal symplectic transformations that have to be used, we choose among all possible transformations the ones with optimal (smallest possible) condition number (see [9]).

As the iteration progresses, some of the Ritz values may converge to eigenvalues of M long before the entire set of wanted eigenvalues have. These converged Ritz values may be part of the wanted or unwanted portion of the spectrum. In either case it is desirable to deflate the converged Ritz values and corresponding Ritz vectors from the unconverged portion of the factorization. If the converged Ritz value is wanted, then it is necessary to keep it in the subsequent factorizations; if it is unwanted, then it must be removed from the current and the subsequent factorizations. Lehoucq and Sorensen develop in [25, 40] locking and purging techniques to accomplish this in the context of unsymmetric matrices and the restarted Arnoldi method. These ideas can be carried over to the situation here.

It is well known that for general Lanczos-like methods the stability of the overall process is improved when the norm of the Lanczos vectors is chosen to be equal to 1 [36, 41]. Moreover, without some form of reorthogonalization any Lanczos algorithm is numerically unstable. For more on these two important points see [4, section 6.1], as the discussion given there in the context of a symplectic Lanczos method for the Hamiltonian eigenproblem does not differ from what can be said here.

5. Breakdowns in the *SR* factorization. If there is a starting vector $\breve{v}_1 = \rho q(M)v_1$ for which the explicitly restarted symplectic Lanczos method breaks down, then it is impossible to reduce the symplectic matrix M to symplectic butterfly form with a transformation matrix whose first column is $\breve{v}_1$. Thus, in this situation the SR decomposition of $q(B)$ cannot exist.

As will be shown in this section, this is the only way that breakdowns in the SR decomposition can occur. In the SR step, most of the transformations used are orthogonal symplectic transformations; their computation cannot break down. The only source of breakdown can be one of the symplectic Gaussian eliminations L_j. For simplicity, we will discuss the double shift case. Only the following elementary elimination matrices are used in the implicit SR step:

- elementary symplectic Givens matrices [35]

$$G_k = \begin{bmatrix} C_k & -S_k \\ S_k & C_k \end{bmatrix},$$

  where $C_k = I + (c_k - 1)e_k e_k^T$, $S_k = s_k e_k e_k^T$, and $c_k^2 + s_k^2 = 1$,

- elementary symplectic Gaussian elimination matrices [9]

$$L_k = \begin{bmatrix} W_k & V_k \\ 0 & W_k^{-1} \end{bmatrix},$$

where $W_k = I + (w_k - 1)(e_{k-1}e_{k-1}^T + e_k e_k^T)$, and $V_k = v_k(e_{k-1}e_k^T + e_k e_{k-1}^T)$,

- elementary symplectic Householder transformation

$$H_k = \left[\begin{array}{cc|cc} I^{k-1,k-1} & & & \\ & P & & \\ \hline & & I^{k-1,k-1} & \\ & & & P \end{array}\right], \qquad \text{where } P = I^{n-k+1,n-k+1} - 2\frac{vv^T}{v^Tv}.$$

Assume that k steps of the symplectic Lanczos algorithm are performed, then from (3.5)

$$M_P S_P^{2n,2k} = S_P^{2n,2k} B_P^{2k,2k} + r_{k+1} e_{2k}^T. \tag{5.1}$$

Now an implicit restart is to be performed using an implicit double shift SR step. In the first step of the implicit SR step, a symplectic Householder matrix H_1 is computed such that $H_1^T q(B^{2k,2k})e_1 = \lambda e_1$. H_1 is applied to $B^{2k,2k}$ via $H_1^T B^{2k,2k} H_1$, introducing a small bulge in the butterfly form: additional elements are found in the positions $(2,1)$, $(1,2)$, $(n+2,n+1)$, $(n+1,n+2)$, $(1,n+3)$, $(3,n+1)$, $(n+1,n+3)$, and $(n+3,n+1)$. The remaining implicit transformations perform a bulge-chasing sweep down the subdiagonal to restore the butterfly form. An algorithm for this is given in [2] or [5]; it can be summarized for the situation here as in Table 5.1, where $\tilde{G}_j$ and G_j both denote symplectic Givens transformation matrices acting in the same planes but with different rotation angles.

TABLE 5.1
Reduction to butterfly form—double shift case.

for $\ell = 1 : n-1$
  compute $G_{\ell+1}$ such that $(G_{\ell+1}B^{2k,2k})_{n+\ell+1,\ell} = 0$
  $B^{2k,2k} = G_{\ell+1}B^{2k,2k}G_{\ell+1}^T$
  compute $L_{\ell+1}$ such that $(L_{\ell+1}B^{2k,2k})_{\ell+1,\ell} = 0$
  $B^{2k,2k} = L_{\ell+1}B^{2k,2k}L_{\ell+1}^{-1}$
  compute $\tilde{G}_{\ell+1}$ such that $(B^{2k,2k}\tilde{G}_{\ell+1})_{\ell,\ell+1} = 0$
  $B^{2k,2k} = \tilde{G}_{\ell+1}^T B^{2k,2k}\tilde{G}_{\ell+1}$
  compute $H_{\ell+1}$ such that $(B^{2k,2k}H_{\ell+1})_{\ell,n+\ell+2} = 0$
  $B^{2k,2k} = H_{\ell+1}^T B^{2k,2k} H_{\ell+1}$
end

Suppose that the first $j-1$ Gaussian transformations, $j < k$, exist and that we have computed

$$\widehat{S} = H_1 G_2^T L_2^{-1} \tilde{G}_2 \ldots H_{j-2} G_{j-1}^T L_{j-1}^{-1} \tilde{G}_{j-1} H_{j-1} G_j^T.$$

In order to simplify the notation, we switch to the permuted version and rewrite the permuted symplectic matrix $\widehat{S}_P$ as

$$\widehat{S}_P = \left[\begin{array}{cc} S_P & 0 \\ 0 & I^{2(n-j-1)2(n-j-1)} \end{array}\right],$$

where $S_P \in \mathbb{R}^{(2j+2)\times(2j+2)}$, making use of the fact that the accumulated transformations affect only the rows 1 to j and $n+1$ to $n+j$. The leading $(2j+2)\times(2j+2)$

principal submatrix of $\widehat{S}_P^{-1} B_P^{2k,2k} \widehat{S}_P$ is given by

$$\widetilde{B}_P^{2j+2,2j+2} = \left[\begin{array}{ccc|cc|cc} \breve{B}^{2j-4,2j-4} & 0 & \breve{b}_{j-2}\breve{d}_{j-1}e_{2j-5} & & & & \\ & 0 & \breve{a}_{j-2}\breve{d}_{j-1}e_{2j-4} & & & & \\ \hline \breve{b}_{j-1}\breve{d}_{j-1}e_{2j-4}^T & \breve{b}_{j-1} & \breve{b}_{j-1}\breve{c}_{j-1} - \breve{a}_{j-1}^{-1} & 0 & \hat{x} & & \\ \breve{a}_{j-1}\breve{d}_{j-1}e_{2j-4}^T & \breve{a}_{j-1} & \breve{a}_{j-1}\breve{c}_{j-1} & 0 & \hat{x} & & \\ \hline & 0 & \hat{y}_1 & \hat{b}_j & \hat{x} & \hat{x} & \hat{x} \\ & 0 & \hat{y}_2 & \hat{a}_j & \hat{x} & \hat{x} & \hat{x} \\ \hline & 0 & \hat{x}_1 & \hat{x}_2 & \hat{x} & \hat{x} & \hat{x} \\ & 0 & 0 & 0 & \hat{x} & \hat{x} & \hat{x} \end{array}\right], \tag{5.2}$$

where the hatted quantities denote unspecified entries that would change if the SR update could be continued. Next, the $(2j+1, 2j-1)$ entry should be annihilated by a permuted symplectic Gaussian elimination. This elimination will fail to exist if $\hat{a}_j = 0$; the SR decomposition of $q(B^{2k,2k})$ does not exist.

As will be needed later, $\hat{a}_j = 0$ implies that $\hat{y}_2 = 0$. This follows as $\widetilde{B}_P^{2j+2,2j+2}$ is J_P-orthogonal: From

$$e_{2j-2}^T \widetilde{B}_P^{2j+2,2j+2} J_P (\widetilde{B}_P^{2j+2,2j+2})^T e_{2j} = e_{2j-2}^T J_P e_{2j} = 0$$

we obtain $0 = -\breve{a}_{j-1}\hat{y}_2 - \hat{x}\hat{a}_j$. If $\hat{a}_j = 0$, we have $\hat{y}_2 = 0$ as $\breve{a}_{j-1} \neq 0$. (Otherwise the last Gaussian transformation L_{j-1} did not exist.)

Next, we show that this breakdown in the SR decomposition implies a breakdown in the Lanczos process started with the starting vector $\breve{v}_1 = \rho q(M_P) v_1$.

For this we have to consider (5.1) multiplied from the right by $\widehat{S}_P$. From the derivations in the last section we know that the starting vector of that recursion is given by $\breve{v}_1 = \rho q(M_P) v_1$. As the trailing $(2n-2j-2) \times (2n-2j-2)$ principal submatrix of $\widehat{S}_P$ is the identity, we can just as well consider

$$M_P S_P^{2n,2j+2} = S_P^{2n,2j+2} B_P^{2j+2,2j+2} + r_{j+2} e_{2j+2}^T,$$

multiplied from the right by S_P

$$M_P \breve{S}_P^{2n,2j+2} = \breve{S}_P^{2n,2j+2} \breve{B}_P^{2j+2,2j+2} + r_{j+2} e_{2j+2}^T S_P, \tag{5.3}$$

where $\breve{B}_P^{2j+2,2j+2} = S_P^{-1} B_P^{2j+2,2j+2} S_P$ corresponds to the matrix in (5.2) (no butterfly form!) and $\breve{S}_P^{2n,2j+2} = S_P^{2n,2j+2} S_P = [\breve{v}_1, \breve{w}_1, \ldots, \breve{v}_{j-1}, \breve{w}_{j-1}, \hat{v}_j, \hat{w}_j, \hat{v}_{j+1}, \hat{w}_{j+1}]$. The columns of $\breve{S}_P^{2n,2j+2}$ are J_P-orthogonal, i.e.,

$$(\breve{S}_P^{2n,2j+2})^T J_P \breve{S}_P^{2n,2j+2} = J_P^{2(j+1),2(j+1)}. \tag{5.4}$$

The starting vector of the recursion (5.3) is given by $\breve{v}_1 = \rho q(M_P) v_1$. Deleting the last four columns of $\breve{S}_P^{2n,2j+2}$ in the same way as in the implicit restart we obtain a valid symplectic Lanczos factorization of length $2j-2$.

In order to show that a breakdown in the SR decomposition of $q(B)$ implies a breakdown in the above symplectic Lanczos recursion, we need to show

$$\hat{a}_j = 0 \quad \Longrightarrow \quad \breve{a}_j = \breve{v}_j^T J_P M_P \breve{v}_j = 0.$$

From (5.2) and (5.3) we obtain

$$\begin{aligned} M_P \breve{w}_{j-1} = {} & \breve{b}_{j-2}\breve{d}_{j-1}\breve{v}_{j-2} + \breve{a}_{j-2}\breve{d}_{j-1}\breve{w}_{j-2} + (\breve{b}_{j-1}\breve{c}_{j-1} - \breve{a}_{j-1}^{-1})\breve{v}_{j-1} \\ & + \breve{a}_{j-1}\breve{c}_{j-1}\breve{w}_{j-1} + \hat{y}_1\hat{v}_j + \hat{y}_2\hat{w}_j + \hat{x}_1\hat{v}_{j+1}, \end{aligned} \tag{5.5}$$

and

$$M_P\breve{v}_k = \breve{b}_k\breve{v}_k + \breve{a}_k\breve{w}_k, \qquad k \le j-1. \tag{5.6}$$

Further, we do know from the symplectic Lanczos algorithm

$$\breve{v}_j = -\breve{d}_{j-1}\breve{v}_{j-2} - \breve{c}_{j-1}\breve{v}_{j-1} + \breve{w}_{j-1} + \breve{a}_{j-1}^{-1}M_P^{-1}\breve{v}_{j-1}, \tag{5.7}$$

all of these quantities are already known. Now consider

$$\begin{aligned}\breve{a}_j &= \breve{v}_j^T J_P M_P \breve{v}_j \\ &= -\underbrace{\breve{d}_{j-1}\breve{v}_j^T J_P M_P \breve{v}_{j-2}}_{x_1} - \underbrace{\breve{c}_{j-1}\breve{v}_j^T J_P M_P \breve{v}_{j-1}}_{x_2} + \underbrace{\breve{v}_j^T J_P M_P \breve{w}_{j-1}}_{x_3} + \underbrace{\breve{a}_{j-1}^{-1}\breve{v}_j^T J_P \breve{v}_{j-1}}_{x_4}.\end{aligned}$$

Obviously, $x_4 = 0$. Using (5.6) we obtain $\breve{v}_j^T J_P M_P \breve{v}_k = \breve{b}_k\breve{v}_j^T J_P\breve{v}_k + \breve{a}_k\breve{v}_j^T J_P\breve{w}_k = 0$ for $k = j-1, j-2$. Hence $x_1 = x_2 = 0$. Using (5.5) and (5.4) will see that $x_3 = 0$:

$$\begin{aligned}\breve{v}_j^T J_P M_P \breve{w}_{j-1} &= \breve{b}_{j-2}\breve{d}_{j-1}\breve{v}_j^T J_P\breve{v}_{j-2} + \breve{a}_{j-2}\breve{d}_{j-1}\breve{v}_j^T J_P\breve{w}_{j-2} \\ &\quad + (\breve{b}_{j-1}\breve{c}_{j-1} - \breve{a}_{j-1}^{-1})\breve{v}_j^T J_P\breve{v}_{j-1} + \breve{a}_{j-1}\breve{c}_{j-1}\breve{v}_j^T J_P\breve{w}_{j-1} \\ &\quad + \hat{y}_1\breve{v}_j^T J_P\hat{v}_j + \hat{y}_2\breve{v}_j^T J_P\hat{w}_j + \hat{x}_1\breve{v}_j^T J_P\hat{v}_{j+1} \\ &= \underbrace{\hat{y}_1\breve{v}_j^T J_P\hat{v}_j}_{z_1} + \underbrace{\hat{y}_2\breve{v}_j^T J_P\hat{w}_j}_{z_2} + \underbrace{\hat{x}_1\breve{v}_j^T J_P\hat{v}_{j+1}}_{z_3}.\end{aligned}$$

As $\hat{a}_j = 0$, $\hat{y}_2 = 0$, and therefore, $z_2 = 0$. With (5.7) we get

$$z_1 = -\hat{y}_1\hat{v}_j^T J_P\breve{v}_j = -\hat{y}_1\breve{a}_{j-1}^{-1}\hat{v}_j^T M_P^T J_P\breve{v}_{j-1}.$$

From (5.3) we obtain $M_P\hat{v}_j = \hat{b}_j\hat{v}_j + \hat{a}_j\hat{w}_j + \hat{x}_2\hat{v}_{j+1}$. Hence, using (5.4) yields

$$z_1 = -\hat{y}_1\breve{a}_{j-1}^{-1}(\hat{b}_j\hat{v}_j^T J_P\breve{v}_{j-1} + \hat{a}_j\hat{w}_j^T J_P\breve{v}_{j-1} + \hat{x}_2\hat{v}_{j+1}^T J_P\breve{v}_{j-1}) = 0.$$

Similarly, it follows that $z_3 = 0$. Hence $x_3 = 0$, and therefore, $\breve{v}_j^T J_P M_P\breve{v}_j = 0$.

This derivation has shown that an SR breakdown implies a serious Lanczos breakdown. The opposite implication follows from the uniqueness of the Lanczos factorization. The result is summarized in the following theorem.

THEOREM 5.1. *Suppose the symplectic butterfly matrix $B^{2k,2k}$ corresponding to* (3.5) *is unreduced, and let $\mu \in \mathbb{R}$. Let L_j be the jth symplectic Gauss transformation required in the SR step on $(B^{2k,2k} - \mu I)(B^{2k,2k} - \mu^{-1}I)(B^{2k,2k})^{-1}$. If the first $j-1$ symplectic Gauss transformations of this SR step exist, then L_j fails to exist iff $\breve{v}_j^T J_P M_P\breve{v}_j = 0$ with $\breve{v}_j$ as in* (4.3).

6. Numerical experiments. Some examples to demonstrate the properties of the (implicitly restarted) symplectic Lanczos method are presented. The computational results are quite promising but certainly preliminary. All computations were done using MATLAB Version 5.1 on a Sun Ultra 1 with IEEE double-precision arithmetic and machine precision $\epsilon = 2.2204 \times 10^{-16}$.

Our code implements exactly the algorithm as given in Table 4.1. In order to detect convergence in the restart process, the rather crude criterion

$$||r_{k+1}|| \le ||M|| * 10^{-6}$$

was used. This ad hoc stopping rule allowed the iteration to halt quite early. Usually, the eigenvalues largest in modulus (and their reciprocals) of the wanted part of the spectrum are much better approximated than the ones of smaller modulus. In a black-box implementation of the algorithm this stopping criterion has to be replaced with a more rigorous one to ensure that all eigenvalues are approximated to the desired accuracy (see the discussion in section 3.3). Benign breakdown in the symplectic Lanczos process was detected by the criterion

$$||v_{m+1}|| \leq \epsilon * ||M|| \quad \text{or} \quad ||w_{m+1}|| \leq \epsilon * ||M||,$$

while a serious breakdown was detected by

$$v_{m+1} \neq 0, \quad w_{m+1} \neq 0, \quad |a_{m+1}| \leq \epsilon * ||M||.$$

Our implementation intends to compute the k eigenvalues of M largest in modulus and their reciprocals. In the implicit restart, we used exact shifts where we chose the shifts to be the $2p$ eigenvalues of $B^{2k+p,2k+p}$ closest to the unit circle.

Our observations have been the following.

- Re-J-orthogonalization is necessary; otherwise J-orthogonality of the computed Lanczos vectors is lost after a few steps, and *ghost eigenvalues* (see, e.g., [19]) appear. That is, multiple eigenvalues of $B^{2k,2k}$ correspond to simple eigenvalues of M.
- The implicit restart is more accurate than the explicit one.
- The leading end of the "wanted" Ritz values (that is, the eigenvalues largest in modulus and their reciprocals) converge faster than the tail end (closest to cut off of the sort). The same behavior was observed in [39] for the implicitly restarted Arnoldi method. In order to obtain faster convergence, it seems advisable (similar to the implementation of Sorensen's implicitly restarted Arnoldi method in MATLAB's `eigs`) to increase the dimension of the computed Lanczos factorization. That is, instead of computing $S_P^{2n,2k}$ and $B_P^{2k,2k}$ as a basis for the restart, one should compute a slightly larger factorization, e.g., of dimension $2(k+3)$ instead of dimension $2k$. When 2ℓ eigenvalues have converged, a subspace of dimension $2(k+3+\ell)$ should be computed as a basis for the restart, followed by p additional Lanczos steps to obtain a factorization of length $k+3+\ell+p$. Using implicit SR steps this factorization should be reduced to one of length $k+3+\ell$. If the symplectic Lanczos method would be implemented following this approach, the convergence check could be done using only the k Ritz values of largest modulus (and their reciprocals) or those that yield the smallest Ritz estimate

  $$|d_{k+1}|\,|e_{2k}^T y_j|\,||b_{k+1}\hat{v}_{k+1} + a_{k+1}\hat{w}_{k+1}||,$$

  where the y_j are the eigenvectors of $B^{2k,2k}$.
- It is fairly difficult to find a good choice for k and p. Not for every possible choice of k does there exist an invariant subspace of dimension $2k$ associated to the k eigenvalues λ_i largest in modulus and their reciprocals. If λ_k is complex and $\overline{\lambda_{k+1}} = \lambda_k$, then we cannot choose the $2p$ eigenvalues with modulus closest to the unit circle as shifts because this would tear a quadruple of eigenvalues apart, resulting in a shift polynomial q such that $q(B_P^{2(k+p),2(k+p)})$ would not be real. All we can do is to choose the $2p-2$ eigenvalues with modulus closest to 1 as shifts. In order to get a full set of $2p$ shifts we add as

the last shift the real eigenvalue pair with largest Ritz residual. Depending on how good that real eigenvalue approximates an eigenvalue of M, this strategy worked, but the resulting subspace is no longer the subspace corresponding to the k eigenvalues largest in modulus and their reciprocals. If the real eigenvalue has converged to an eigenvalue of M, it is unlikely to remove that eigenvalue just by restarting; it will keep coming back. Only a purging technique like the one discussed by Lehoucq and Sorensen [25, 40] will be able to remove this eigenvalue. Moreover, there is no guarantee that there is a real eigenvalue of $B_P^{2(k+p),2(k+p)}$ that can be used here. Hence, in a black-box implementation one should either try to compute an invariant subspace of dimension $2(k-1)$ or of dimension $2(k+1)$. As this is not known a priori, the algorithm should adapt k during the iteration process appropriately. This is no problem if, as suggested above, one always computes a slightly larger Lanczos factorization than requested.

Example 6.1. Tests were done using a 100×100 symplectic matrix with the eigenvalues

$$200, 100, 50, 47, \ldots, 4, 3, 2 \pm \mathtt{i}, 1/3, 1/4, \ldots, 1/47, 1/50, 1/100, 1/200.$$

A real symplectic block-diagonal matrix with these eigenvalues was constructed and a similarity transformation with a randomly generated orthogonal symplectic matrix was performed to obtain a symplectic matrix M.

The first test performed concerned the loss of J-orthogonality of the computed Lanczos vectors during the symplectic Lanczos method and the ghost eigenvalue problem (see, e.g., [19]). As expected, when using a random starting vector M's eigenvalues largest in modulus (and the corresponding reciprocals) tend to emerge right from the start, e.g., the eigenvalues of $B^{10,10}$ are

$$199.99997,\ 100.06771,\ 48.71752,\ 26.85083,\ 8.32399,$$

and their reciprocals. Without any form of re-J-orthogonalization, the J-orthogonality of the Lanczos vectors is lost after a few iterations as indicated in Figure 6.1.

The loss of J-orthogonality in the Lanczos vectors results, as in the standard Lanczos algorithm, in ghost eigenvalues. That is, multiple eigenvalues of $B^{2k,2k}$ correspond to simple eigenvalues of M. For example, using no re-J-orthogonalization, after 17 iterations the 6 eigenvalues largest in modulus of $B^{34,34}$ are

$$207.63389,\ 200,\ 100,\ 49.99982,\ 47.04542,\ 45.85367.$$

Using complete re-J-orthogonalization, this effect is avoided:

$$200,\ 100,\ 49.99992,\ 47.02461,\ 45.93018,\ 42.31199.$$

The second test performed concerned the question whether an implicit restart is more accurate than an explicit one. After 9 steps of the symplectic Lanczos method (with a random starting vector) the resulting 18×18 symplectic butterfly matrix $B^{18,18}$ had the eigenvalues (using the MATLAB function `eig`)

```
200.000000000000,      99.999999841718,
 50.070648930465,      41.873264094053,
 35.891491504806,      23.654512559868,
 13.344815062428,       3.679215125563 ±5.750883779240i,
```

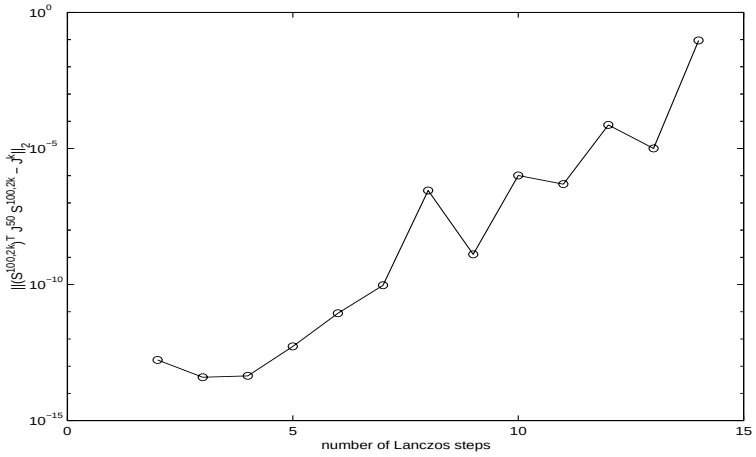


FIG. 6.1. *Loss of J-orthogonality after k symplectic Lanczos steps.*

and their reciprocals. Removing the 4 complex eigenvalues from $B^{18,18}$ using an implicit restart as described in section 4, we obtain a symplectic butterfly matrix $B^{14,14}_{impl}$ whose eigenvalues are

$$\begin{aligned} &200.000000000000, && 99.999999841719,\\ &50.070648930464, && 41.873264094053,\\ &35.891491504806, && 23.654512559868,\\ &13.344815062428, \end{aligned}$$

and their reciprocals. From (2.6) it follows that these have to be the 14 real eigenvalues of $B^{18,18}$ which have not been removed. As can be seen, we lost one digit during the implicit restart (indicated by the "underbar" under the "lost" digits in the above table). Performing an explicit restart with the explicitly computed new starting vector $\breve{v}_1 = (M_P - \mu I)(M_P - \overline{\mu} I)(M_P - \mu^{-1} I)(M_P - \overline{\mu}^{-1} I) M_P^{-2} v_1$ yields a symplectic butterfly matrix $B^{14,14}_{expl}$ whose eigenvalues are

$$\begin{aligned} &200.000000000000, && 99.999999841793,\\ &50.070648885030, && 41.873247045627,\\ &35.891922701991, && 23.654509163541,\\ &13.344810484061, \end{aligned}$$

and their reciprocals. This time we lost up to 9 digits.

The last set of tests performed on this matrix concerned the k-step restarted symplectic Lanczos method as given in Table 4.1. As M has only one quadruple of complex eigenvalues, and these eigenvalues are smallest in magnitude, there is no problem in choosing $k \ll n$. For every such choice there exists an invariant symplectic subspace corresponding to the k eigenvalues largest in magnitude and their reciprocals. In the tests reported here, a random starting vector was used. Figure 6.2 shows a plot of $||r_{k+1}||$ versus the number of iterations performed. Iteration step 1 refers to the norm of the residual after the first k Lanczos steps; no restart is performed. The three lines in Figure 6.2 present three different choices for k and p: $k = p = 8; k = 8, p = 16$; and $k = 5, p = 10$. Convergence was achieved for all three examples (and many more, not shown here). Obviously, the choice $k = 8, p = 2k$ results in faster convergence than the choice $k = p = 8$. Convergence is by no means monotonic; during the major part of the iteration the norm of the residual is changing quite dramatically. But once

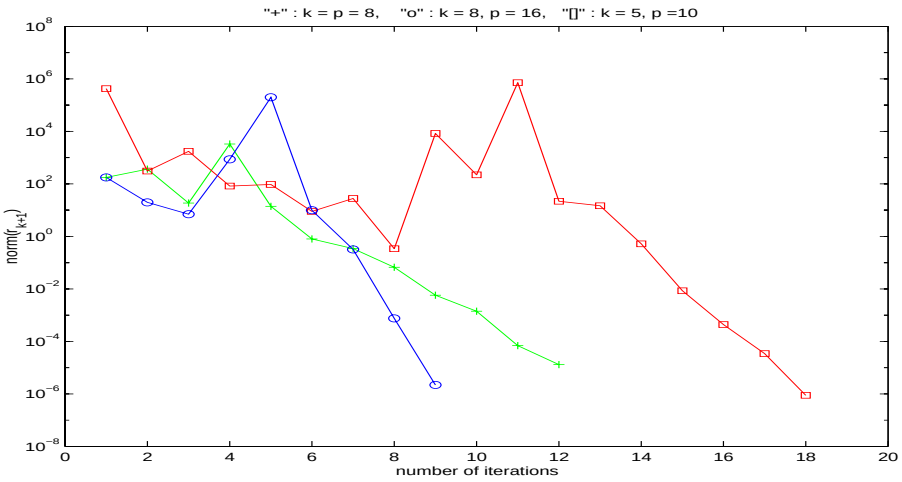


FIG. 6.2. *k-step restarted symplectic Lanczos method with different choices of k and p.*

a certain stage is achieved, the norm of the residual converges. Although convergence for $k = 8, p = k$ or $p = 2k$ was quite fast, this does not imply that convergence is as fast for other choices of k and p. The third line in Figure 6.2 demonstrates that the convergence for $k = 5, p = 10$ does need twice as many iteration steps as for $k = 8, p = 16$.

Example 6.2. Symplectic matrix pencils that appear in discrete-time linear-quadratic optimal control problems are typically of the form

$$L - \lambda N = \begin{bmatrix} F & 0 \\ C^T C & I \end{bmatrix} - \lambda \begin{bmatrix} I & -BB^T \\ 0 & F^T \end{bmatrix}, \quad F \in \mathbb{R}^{n\times n},\ C \in \mathbb{R}^{p\times n},\ B \in \mathbb{R}^{n\times m}.$$

(Note: For $F \neq I$, L and N are not symplectic, but $L - \lambda N$ is a symplectic matrix pencil.) Assuming that L and N are nonsingular (that is, F is nonsingular), solving this generalized eigenproblem is equivalent to solving the eigenproblem for the symplectic matrix

$$N^{-1}L = \begin{bmatrix} I & -BB^T \\ 0 & F^T \end{bmatrix}^{-1} \begin{bmatrix} F & 0 \\ C^T C & I \end{bmatrix}.$$

If one is interested in computing a few of the eigenvalues of $L - \lambda N$, one can use the restarted symplectic Lanczos algorithm on $M = N^{-1}L$. In each step of the symplectic Lanczos algorithm, one has to compute matrix-vector products of the form Mx and $M^T x$. Making use of the special form of L and N, this can be done without explicitly inverting N: Let us consider the computation of $y = Mx$. First compute

$$Lx = \begin{bmatrix} F & 0 \\ C^T C & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} Fx_1 \\ C^T C x_1 + x_2 \end{bmatrix} =: \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z,$$

where $x \in \mathbb{R}^{2n}$ is written as $x = [x_1\ x_2]^T, x_1, x_2 \in \mathbb{R}^n$. Next, one has to solve the linear system $Ny = z$. Partition $y \in \mathbb{R}^{2n}$ analogous to x and z, then from $Ny = z$ we obtain $y_2 = F^{-T} z_2$, and $y_1 = z_1 + BB^T y_2$. In order to solve $y_2 = F^{-T} z_2$ we compute the LU decomposition of F and solve the linear system $F^T y_2 = z_2$ using backward and forward substitution. Hence, the explicit inversion of N or F is avoided. In the

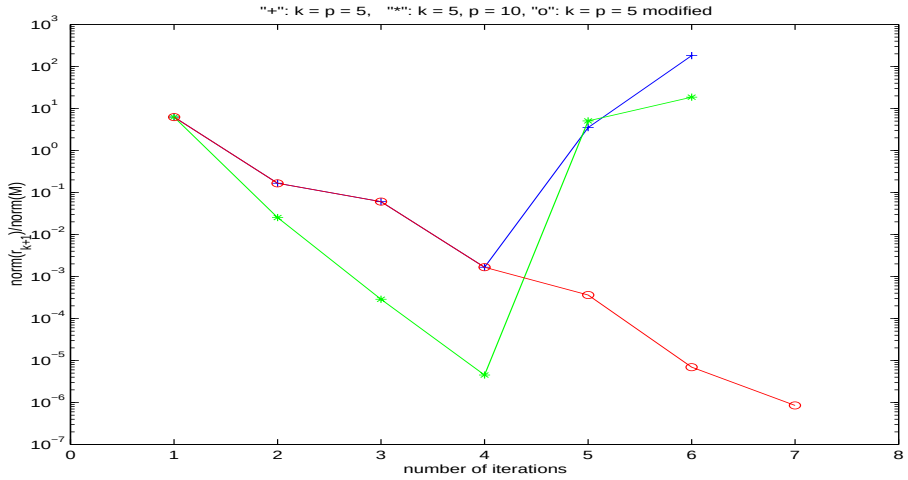


FIG. 6.3. *k-step restarted symplectic Lanczos method with different choices of the shifts.*

case in which F is a sparse matrix, sparse solvers can be employed. In particular, if the control system comes from some sort of discretization scheme, F is often banded, which can be used here by computing an initial band LU factorization of F in order to minimize the cost for the computation of y_2. Note that in most applications, $p, m \ll n$ such that the computational cost for $C^T C x_1$ and $BB^T y_2$ is significantly cheaper than a matrix-vector product with an $n \times n$ matrix. In case of single-input ($m = 1$) or single-output ($p = 1$) the corresponding operations come down to two dot products of length n each.

Using MATLAB's sparse matrix routine `sprandn`, sparse normally distributed random matrices F, B, C (here, $p = m = n$) of different dimensions and with different densities of the nonzero entries were generated. Here, an example of dimension $2n = 1000$ is presented, where the density of the different matrices was chosen to be

matrix	$\approx$ nonzero entries
F	$0.5n^2$
B	$0.2n^2$
C	$0.3n^2$.

MATLAB computed the norm of the corresponding matrix $M = N^{-1}L$ to be $\approx 5.3 \times 10^5$.

In the first set of tests k was chosen to be 5, and we tested $p = k$ and $p = 2k$. As can be seen in Figure 6.3, for the first three iterations, the norm of the residual decreases for both choice of p but then increases quite a bit. During the first step, the eigenvalues of $B^{10,10}$ are approximating the five eigenvalues of $L - \lambda N$ largest in modulus and their reciprocals. In step 4, a "wrong" choice of the shifts is done in both cases. The extended matrices $B^{20,20}$ and $B^{30,30}$ both still approximate the five eigenvalues of $L - \lambda N$ largest in modulus, but there is a new real eigenvalue coming in, which is not a good approximation to an eigenvalue of $L - \lambda N$. But, due to the way the shifts are chosen here, this new eigenvalue is kept, while an already good approximated eigenvalue—a little smaller in magnitude—is shifted away, resulting in a dramatic increase of $||r_{k+1}||$. Modifying the choice of the shifts such that the good approximation is kept, while the new real eigenvalue is shifted away, the problem is

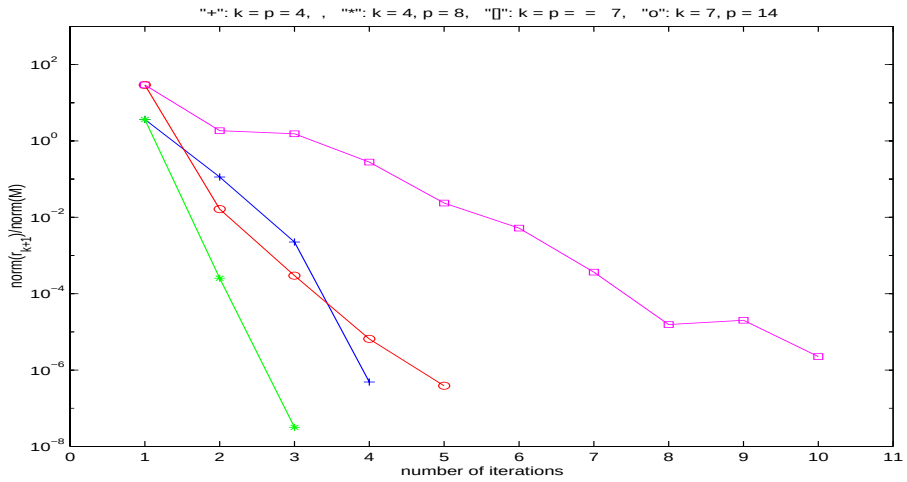


FIG. 6.4. *k-step restarted symplectic Lanczos method with different choices of k and p.*

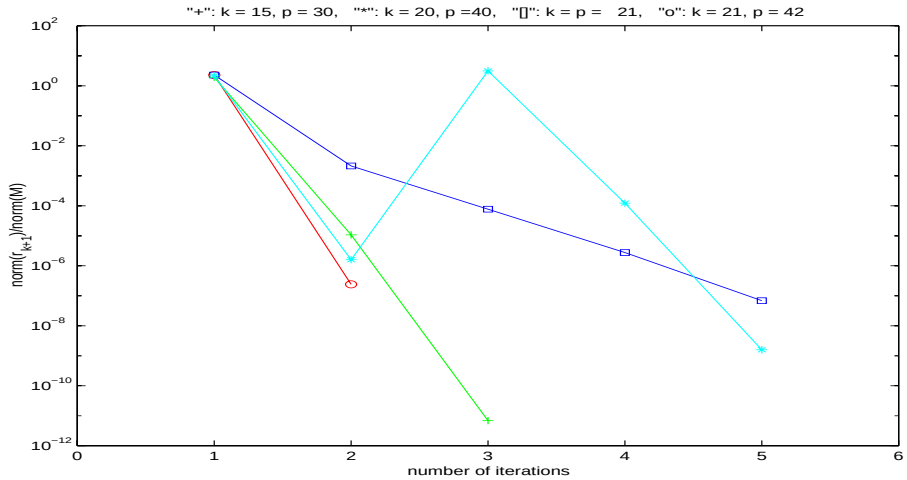


FIG. 6.5. *k-step restarted symplectic Lanczos method with different choices of k and p.*

resolved, the "good" eigenvalues are kept, and convergence occurs in a few steps (the o-line in Figure 6.3).

Using a slightly larger Lanczos factorization as a basis for the restart, e.g., a factorization of length $k+3$ instead of length k, and using a locking technique to decouple converged approximate eigenvalues and associated invariant subspaces from the active part of the iteration, this problem is avoided.

Figure 6.4 displays the behavior of the k-step implicitly restarted symplectic Lanczos method for different choices of k and p, where k is quite small. Convergence is achieved in any case.

So far, in the tests presented, k was always chosen such that there exists a deflating subspace of $L - \lambda N$ corresponding to the k eigenvalues largest in modulus and their reciprocals. For $k = 20$, there is no such deflating subspace. (There is one for $k = 19$ and one for $k = 21$.) See Figure 6.5 for a convergence plot. The eigenvalues of

$B^{2(k+p),2(k+p)}$ in the first iteration steps approximate the $k+j$ eigenvalues of largest modulus and their reciprocals (where $5 \leq j \leq p$) quite well. Our choice of shifts is to select the $2p$ eigenvalues with modulus closest to 1, but as λ_{k+1} is complex with $|\lambda_{k+1}| = |\lambda_k| \neq 1$, we can only choose $2(p-1)$ shifts that way. The last shift is chosen according to the strategy explained above. This eigenvalue keeps coming back before it is annihilated. A better idea to resolve the problem is to adapt k appropriately.

7. Concluding remarks. We have investigated a symplectic Lanczos method for symplectic matrices. Employing the technique of implicitly restarting the method using double or quadruple shifts as zeros of the driving Laurent polynomials, this results in an efficient method to compute a few extremal eigenvalues of symplectic matrices and the associated eigenvectors or invariant subspaces. The residual of the Lanczos recursion can be made to be zero by choosing proper shifts. It is an open problem how these shifts should be chosen in an optimal way. The preliminary numerical tests reported here show that for exact shifts, good performance is already achieved.

Before implementing the symplectic Lanczos process in a black-box algorithm, some details need consideration: in particular, techniques for locking of converged Ritz values as well as purging of converged, but unwanted, Ritz values need to be derived in a way as it has been done for the implicitly restarted Arnoldi method.

REFERENCES

[1] Z. Bai, *Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem*, Math. Comp., 62 (1994), pp. 209–226.
[2] G. Banse, *Symplektische Eigenwertverfahren zur Lösung zeitdiskreter optimaler Steuerungsprobleme*, Ph.D. thesis, Universität Bremen, Fachbereich 3–Mathematik und Informatik, Bremen, Germany, 1995.
[3] G. Banse and A. Bunse-Gerstner, *A condensed form for the solution of the symplectic eigenvalue problem*, in Systems and Networks: Mathematical Theory and Applications, U. Helmke, R. Menniken, and J. Sauer, eds., Akademie Verlag, Berlin, Germany, 1994, pp. 613–616.
[4] P. Benner and H. Fassbender, *An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem*, Linear Algebra Appl., 263 (1997), pp. 75–111.
[5] P. Benner and H. Fassbender, *The symplectic eigenvalue problem, the butterfly form, the SR algorithm and the Lanczos method*, Linear Algebra Appl., 275/276 (1998), pp. 19–47.
[6] P. Benner, H. Fassbender, and D. Watkins, *SR and SZ algorithms for the symplectic (butterfly) eigenproblem*, Linear Algebra Appl., 287 (1999), pp. 41–76.
[7] M. Bohner, *Linear Hamiltonian difference systems: Disconjugacy and Jacobi–type conditions*, J. Math. Anal. Appl., 199 (1996), pp. 804–826.
[8] A. Bunse-Gerstner, *Matrix factorization for symplectic QR-like methods*, Linear Algebra Appl., 83 (1986), pp. 49–77.
[9] A. Bunse-Gerstner and V. Mehrmann, *A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 1104–1113.
[10] D. Calvetti, L. Reichel, and D. C. Sorensen, *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal., 2 (1994), pp. 1–21.
[11] J. Della Dora, *Sur quelques Algorithmes de recherche de valeurs propres*, Thése, L'Université Scientifique es Medicale de Grenoble, Grenoble, France, 1973.
[12] J. Della Dora, *Numerical linear algorithms and group theory*, Linear Algebra Appl., 10 (1975), pp. 267–283.
[13] L. Elsner, *On some algebraic problems in connection with general eigenvalue algorithms*, Linear Algebra Appl., 26 (1979), pp. 123–138.
[14] H. Fassbender, *Error analysis of the symplectic Lanczos method for the symplectic eigenvalue problem*, BIT, 40 (2000), pp. 471–496.
[15] H. Fassbender, *Symplectic Methods for Symplectic Eigenproblems*, Habilitationsschrift, Universität Bremen, Fachbereich 3–Mathematik und Informatik, Bremen, Germany, 1998.

[16] U. Flaschka, V. Mehrmann, and D. Zywietz, *An analysis of structure preserving methods for symplectic eigenvalue problems*, RAIRO Automat.-Prod. Inform. Ind., 25 (1991), pp. 165–190.

[17] J. G. F. Francis, *The QR transformation Part* I *and Part* II, Comput. J., 4 (1961), pp. 265–271, 332–345.

[18] R. Freund, *A transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[19] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[20] E. Grimme, D. Sorensen, and P. Van Dooren, *Model reduction of state space systems via an implicitly restarted Lanczos method*, Numer. Algorithms, 12 (1996), pp. 1–31.

[21] W. Kahan, B. N. Parlett, and E. Jiang, *Residual bounds on approximate eigensystems of nonnormal matrices*, SIAM J. Numer. Anal., 19 (1982), pp. 470–484.

[22] V. N. Kublanoskaja, *On some algorithms for the solution of the complete eigenvalue problem*, U.S.S.R. Comput. Math. and Math. Phys., 3 (1961), pp. 637–657.

[23] P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*, Oxford University Press, Oxford, UK, 1995.

[24] A. Laub, *Invariant subspace methods for the numerical solution of Riccati equations*, in The Riccati Equation, S. Bittanti, A. Laub, and J. Willems, eds., Springer-Verlag, Berlin, Germany, 1991, pp. 163–196.

[25] R. Lehoucq and D. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[26] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK User's Guide. Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998.

[27] R. B. Lehoucq, *On the convergence of an implicitly restarted Arnoldi method*, preprint MS 1110, Sandia National Laboratory, Albuquerque, NM, 1999.

[28] W.-W. Lin, *A new method for computing the closed loop eigenvalues of a discrete-time algebraic Riccati equation*, Linear Algebra Appl., 6 (1987), pp. 157–180.

[29] W.-W. Lin, V. Mehrmann, and H. Xu, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 301–303 (1999), pp. 469–533.

[30] V. Mehrmann, *Der SR-Algorithmus zur Berechnung der Eigenwerte einer Matrix*, Diplomarbeit, Universität Bielefeld, Bielefeld, FRG, 1979.

[31] V. Mehrmann, *A symplectic orthogonal method for single input or single output discrete time optimal quadratic control problems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 221–247.

[32] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, Lecture Notes in Control and Information Sciences 163, Springer-Verlag, Heidelberg, Germany, 1991.

[33] V. Mehrmann and D. Watkins, *Structure preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, Tech. Report SFB393/00-02, Fak. f. Mathematik, TU Chemnitz, Chemnitz, FRG, 2000.

[34] R. B. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.

[35] C. Paige and C. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 14 (1981), pp. 11–32.

[36] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124.

[37] R. Patel, *Computation of the stable deflating subspace of a symplectic pencil using structure preserving orthogonal transformations*, in Proceedings of the 31st Annual Allerton Conference on Communication, Control and Computing, University of Illinois, Urbana, IL, 1993.

[38] R. Patel, *On computing the eigenvalues of a symplectic pencil*, Linear Algebra Appl., 188/189 (1993), pp. 591–611.

[39] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[40] D. C. Sorensen, *Deflation for implicitly restarted Arnoldi methods*, Tech. Report, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1998.

[41] D. R. Taylor, *Analysis of the look ahead Lanczos algorithm*, Ph.D. thesis, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1982.

[42] C. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 16 (1984), pp. 233–251.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 714–725

PARALLEL STRATEGIES FOR RANK-k UPDATING OF THE QR DECOMPOSITION*

ERRICOS J. KONTOGHIORGHES†

Abstract. Parallel strategies based on Givens rotations are proposed for updating the QR decomposition of an $n \times n$ matrix after a rank-k change ($k < n$). The complexity analyses of the Givens algorithms are based on the total number of Givens rotations applied to a 2-element vector. The algorithms, which are extensions of the rank-1 updating method, achieve the updating using approximately $2(k+n)$ compound disjoint Givens rotations (CDGRs) with elements annihilated by rotations in adjacent planes. Block generalization of the serial rank-1 algorithms are also presented. The algorithms are rich in level 3 BLAS operations, making them suitable for implementation on large scale parallel systems. The performance of some of the algorithms on a 2-D SIMD (single instruction stream–multiple instruction stream) array processor is discussed.

Key words. QR decomposition, Givens rotations, parallel algorithms

AMS subject classifications. 15A23, 65F05, 65F25, 65Y05

PII. S0895479896308585

1. Introduction.

Given the QR decomposition (QRD) of a nonsingular $n \times n$ matrix A

$$A = QR, \tag{1.1}$$

the problem of recomputing the QRD of

$$\tilde{A} = A + \sum_{i=1}^{k} x_i y_i^T = A + XY^T \tag{1.2}$$

is considered, where $x_i, y_i \in \Re^n$, $X = (x_1 \dots x_k)$, $Y = (y_1 \dots y_k)$, $R \in \Re^{n \times n}$ is upper triangular and $Q \in \Re^{n \times n}$ is orthogonal. The matrix XY^T has rank k and the problem is known as the rank-k updating of the QRD (hereafter, rank-k UQRD) problem. Observing that

$$\tilde{A} = Q(R + Q^T XY^T),$$

the rank-k UQRD problem requires the reduction of

$$\hat{A} = R + ZY^T \tag{1.3}$$

to upper triangular form using orthogonal transformations, where $Z = Q^T X$. An algorithm for updating the QRD after a rank-1 change (that is $k = 1$ in (1.2)) has been described in [5, 6].

The purpose of this work is to propose and analyze parallel strategies for rank-k UQRD, which is important in applications where repeated updating is required [23].

*Received by the editors August 29, 1996; accepted for publication (in revised form) by L. Elden April 12, 2000; published electronically October 25, 2000. This research was supported in part by Swiss National Foundation grants 21-54109.98, 1214-056900.99/1, and 2000-061875.00/1.
http://www.siam.org/journals/simax/22-3/30858.html
†Institut d'informatique, Université de Neuchâtel, Emile-Argand 11, Case postale 2, CH-2007 Neuchâtel, Switzerland (erricos.kontoghiorghes@info.unine.ch).

The algorithms are based on Givens rotations and Householder transformations. The first algorithm is an adaptation of a previously reported parallel Givens sequence for computing the QRD [22]. The second is a block-version of the rank-1 Givens method for UQRD [5, 6]. The performance of the algorithms on a single instruction stream–multiple data stream (SIMD) array processor is investigated.

Throughout the paper $G_{i,j}^{(k)}$ denotes a Givens rotation in plane (i, j) that reduces to zero the element w_{ik} when applied from the left of $W \in \Re^{n\times k}$. It is assumed that a maximum of $\lfloor n/2 \rfloor$ disjoint Givens rotations can be applied simultaneously. The product of these rotations is called compound disjoint Givens rotation (CDGR) [3, 13, 15, 19]. It is assumed that one time unit is required to construct and apply a single Givens rotation to a 2-element vector. For the complexity analyses of the Givens algorithms the time required to compute the rank-k updating $R + ZY^T$ in (1.3) is assumed to be negligible and will not be taken into account. For simplicity the construction of the orthogonal matrix in the UQRD will not be shown.

In section 2 a parallel realization of the algorithm in [5] for solving the rank-k UQRD problem is presented when $k > 1$. Block-versions of the rank-1 Givens algorithms are given in section 3 for solving the rank-k problem, where $k > 1$. In section 4 the performances of some of the algorithms on a SIMD computer are discussed. Conclusions and future work are presented and discussed in section 5.

2. Parallelization of the rank-1 UQRD algorithm. The parallelization of the rank-1 UQRD algorithm in [5] is considered for the rank-k case, where $1 \le k < n$. The first stage of the rank-1 algorithm in [5] is the application of the $n-1$ Givens rotations $V^T = G_{2,1}^{(1)} \cdots G_{n-1,n-2}^{(1)} G_{n,n-1}^{(1)}$ to the augmented matrix $(Z\ R)$ such that $V^T(Z\ R) = (\zeta e_1\ H)$, where $\zeta^2 = Z^T Z$, e_1 is the first column of the $n \times n$ identity matrix I_n and H is an upper Hessenberg matrix [6]. After computing $\tilde{H} = H + \zeta e_1 Y^T$, the second stage of the algorithm computes the $n-1$ Givens rotations $U^T = G_{n,n-1}^{(n-1)} \ldots G_{3,2}^{(2)} G_{2,1}^{(1)}$ to retriangularize the Hessenberg matrix $\tilde{H}$. Thus, the UQRD of $\tilde{A} = A + XY^T = QVUR_n = Q_n R_n$, where R_n is upper triangular and Q_n is orthogonal. In this algorithm a total of $2(n-1)$ rotations are applied in $3(n^2-1)$ time units.

For the solution of the rank-k UQRD problem when $k > 1$, this algorithm can be repeated k times using $2k(n-1)$ rotations and $3k(n^2-1)$ time units. However, computations on z_i, the ith column of Z, can commence immediately after the last two elements of z_{i-1} $(i = 2, \ldots, k)$ have been annihilated. Thus, the annihilation of the elements in z_i can start at the $(2i-1)$th step and will fill-in, successively, the ith subdiagonal of R. Once z_i has been reduced to the form $\zeta_i e_1$, the rank-1 updating $R + \zeta_i e_1 y_i^T$ is performed. This method is a variation of the Sameh and Kuck annihilation scheme (hereafter, the VSK algorithm) and requires $2k + n - 3$ CDGRs to perform the first stage of the rank-k UQRD [22]. Alternatively, the annihilation of the elements of z_i can cease once the ith subdiagonal of R has been filled-in, since, at this stage, the updating $R + z_i y_i^T$ may be performed without creating any further modification of the structure of R. This (SK algorithm) is equivalent to computing the QRD

$$\tilde{Q}^T Z = \begin{pmatrix} R_z \\ 0 \end{pmatrix} \begin{matrix} k \\ n-k \end{matrix} \tag{2.1}$$

using the Sameh and Kuck annihilation scheme and $H = \tilde{Q}^T R$. The matrix resulting from the rank-k updating

$$\tilde{H} = H + \begin{pmatrix} R_z \\ 0 \end{pmatrix} Y^T \tag{2.2}$$

has the same structure as H—that is, its last $n-k-1$ subdiagonals are zero. The first stage in Figure 2.1 illustrates the transformations on Z and the fill-in of R, with $n = 8$. The entries denote the annihilated elements of Z and the fill-in of R after applying a CDGR.

In the retriangularization of $\tilde{H}$ a total of $k+n-2$ CDGRs are used. At step $k+1-i$ the elements of the ith ($i = 1, \ldots, k$) subdiagonal begin to be annihilated successively by the $n-i$ Givens rotations $G^{(1)}_{i+1,i}, G^{(2)}_{i+2,i+1}, \ldots, G^{(n-i)}_{n,n-1}$. The VSK algorithm, which triangularizes $\tilde{H}$, can begin after the $(k+n-1)$th CDGR has been applied to Z. Thus, the VSK algorithm requires one CDGR more than the SK algorithm. The total number of CDGRs applied using the SK algorithm is given by $2(k+n-2)$. The computational details of the SK algorithm are shown by Algorithm 1, where steps 2–9 and 11–18 compute stage 1 and stage 2, respectively. The 2×2 Givens rotation $G^{(j)}$ annihilates the first element of the second row when it is applied from the left of a 2-row matrix, i.e., $G^{(j)} \equiv G^{(1)}_{2,1}$. The standard *colon* notation is used to denote submatrices [6].

```
 1: Let Z = Q^T X and R̃ = (R  Q^T).
 2: for i = 1, 2, ..., k + n − 2 do
 3:    for all j = 1, 2, ..., k do-in-parallel
 4:       p = n + 2j − i − 1
 5:       if (j < p ≤ n) then
 6:          (Z_{p−1:p,j:k}   R̃_{p−1:p,p−j:2n}) = G^(j) (Z_{p−1:p,j:k}   R̃_{p−1:p,p−j:2n})
 7:       end if
 8:    end for all
 9: end for
10: R_{1:k,1:k} = R_{1:k,1:n} + Z_{1:k,:} Y^T
11: for i = n − k, n − k + 1, ..., 2n − 3 do
12:    for all j = 1, 2, ..., n − 1 do-in-parallel
13:       p = n + 2j − i − 1
14:       if (j < p ≤ n) and (j ≤ i − n + k + 1) then
15:          R̃_{p−1:p,j:2n} = G^(j) R̃_{p−1:p,j:2n}
16:       end if
17:    end for all
18: end for
```

ALGORITHM 1. *SK algorithm.*

Alternatively, $\tilde{H}$ can be triangularized using a series of $n-1$ Householder transformations $P^{(1)}, P^{(2)}, \ldots, P^{(n-1)}$, where $P^{(i)}$ annihilates the elements $i+1$ to $\min(k+i, n)$ ($i = 1, \ldots, n-1$) of the ith column of $\tilde{H}$ using the ith row of $\tilde{H}$ as a pivot row. Observe that the last Householder reflection is equivalent to a single Givens rotation. Furthermore, the parallel algorithms are identical to the serial rank-1 Givens algorithm in [5] when $k = 1$. The second stage of Figure 2.1 shows the annihilation pattern for both

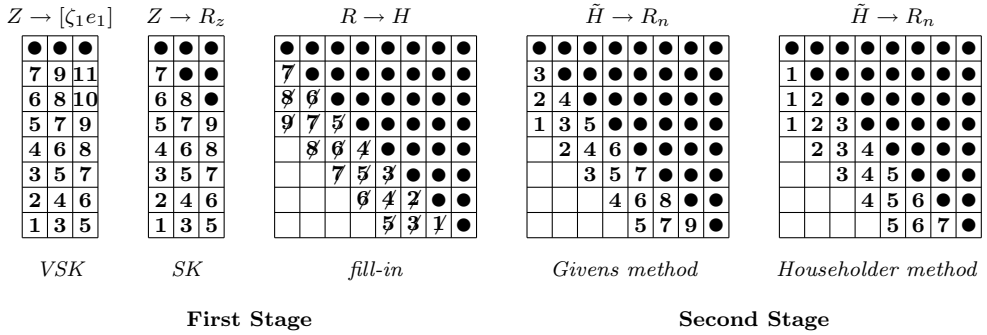


FIG. 2.1. *Rank-k UQRD by CDGRs and Householder transformations.*

methods. In the case of Householder reflections, an integer entry i $(i = 1, \ldots, n-1)$ denotes the elements annihilated by the ith Householder transformation.

The time complexities of the first and second stages of Algorithm 1 are given, respectively, by

$$T_{SK}^{(1)}(k,n) = \sum_{i=1}^{n-1}(k+n+1+i) + \sum_{i=1}^{k-1}(k+2n-i)$$

and

$$T_{SK}^{(2)}(k,n) = 2n(k-1) + \sum_{i=1}^{n-1}(2n+1-i).$$

Thus, the time complexity of the SK algorithm is computed by

$$\begin{aligned} T_{SK}(k,n) &= T_{SK}^{(1)}(k,n) + T_{SK}^{(2)}(k,n) \\ &= 3(n^2-1) + (k-1)(10n+k-2)/2. \end{aligned}$$

The QRD of an $n \times n$ matrix using the SK annihilation scheme requires $2n-3$ CDGRs. Thus, after explicitly computing the updating $R + ZY^T$, the QRD of $\hat{A}$ in (1.3) requires $(2k+1)$ fewer CDGRs than the SK rank-k algorithm, when $k > 1$. The time complexity of this method is given by

$$\begin{aligned} T_{QR}(k,n) &= \sum_{i=1}^{n-1} 2n + \sum_{i=1}^{n-2}(2n-i) \\ &\approx n(7n-9)/2. \end{aligned}$$

Comparing T_{SK} and T_{QR}, for fixed k, $\lim_{n\to\infty} T_{SK}(k,n)/T_{QR}(k,n) = 6/7$.

3. Block parallel strategies. Block generalizations of the rank-1 algorithms can be employed to solve the rank-k UQRD problem. The first block parallel algorithm is based on the serial rank-1 Givens algorithm. Although blocks are processed one at a time, the computations within the blocks can be performed in parallel, using

either Householder reflections or CDGRs [8, 13]. Partitioning the matrices Z and R according to

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_\nu \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix} \tag{3.1a}$$

and

$$R = \begin{array}{c} \begin{matrix} n_1 & n_2 & & n_\nu \end{matrix} \\ \begin{pmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,\nu} \\ & R_{2,2} & \dots & R_{2,\nu} \\ & & \ddots & \vdots \\ & & & R_{\nu,\nu} \end{pmatrix} \end{array} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix}, \tag{3.1b}$$

let the QRD of Z_ν be given by

$$Q_\nu^T Z_\nu = \begin{pmatrix} W_\nu \\ 0 \end{pmatrix} \begin{matrix} k \\ n_\nu - k \end{matrix} \tag{3.2a}$$

and

$$Q_\nu^T R_{\nu,\nu} = \begin{pmatrix} \tilde{R}_{\nu,\nu} \\ \hat{R}_{\nu,\nu} \end{pmatrix} \begin{matrix} k \\ n_\nu - k \end{matrix}, \tag{3.2b}$$

where $n = \sum_{j=1}^{\nu} n_j$ and $n_\nu \geq k$. Computing, for $i = \nu-1, \dots, 2, 1$, the block-updating QR factorization

$$Q_i^T \begin{array}{c} \begin{matrix} k & n_i & & n_\nu \end{matrix} \\ \begin{pmatrix} Z_i & R_{i,i} & \dots & R_{i,\nu} \\ W_{i+1} & 0 & \dots & \tilde{R}_{i+1,\nu} \end{pmatrix} \end{array} \begin{matrix} n_i \\ k \end{matrix} = \begin{array}{c} \begin{matrix} k & n_i & & n_\nu \end{matrix} \\ \begin{pmatrix} W_i & \tilde{R}_{i,i} & \dots & \tilde{R}_{i,\nu} \\ 0 & \hat{R}_{i,i} & \dots & \hat{R}_{i,\nu} \end{pmatrix} \end{array} \begin{matrix} k \\ n_i \end{matrix}, \tag{3.3}$$

the upper triangular matrix R_z in (2.1) is given by $R_z = W_1$ and the matrix $\tilde{H}$ in (2.2) has the block-Hessenberg structure

$$\tilde{H} = \begin{array}{c} \begin{matrix} n_1 & n_2 & & n_\nu \end{matrix} \\ \begin{pmatrix} \tilde{H}_{1,1} & \tilde{H}_{1,2} & \dots & \tilde{H}_{1,\nu} \\ \hat{R}_{1,1} & \hat{R}_{1,2} & \dots & \hat{R}_{1,\nu} \\ & \hat{R}_{2,2} & \dots & \hat{R}_{2,\nu} \\ & & \ddots & \vdots \\ & & & \hat{R}_{\nu,\nu} \end{pmatrix} \end{array} \begin{matrix} k \\ n_1 \\ n_2 \\ \\ n_\nu - k \end{matrix}, \tag{3.4}$$

where Q_i is orthogonal, W_i is a $k \times k$ upper triangular matrix, and

$$\begin{pmatrix} \tilde{H}_{1,1} & \tilde{H}_{1,2} & \dots & \tilde{H}_{1,\nu} \end{pmatrix} = \begin{pmatrix} \tilde{R}_{1,1} & \tilde{R}_{1,2} & \dots & \tilde{R}_{1,\nu} \end{pmatrix} + W_1 Y^T. \tag{3.5}$$

To triangularize $\tilde{H}$, the factorizations with $i = 1, \ldots, \nu - 1$,

$$\hat{Q}_i^T \begin{pmatrix} \tilde{H}_{i,i} & \tilde{H}_{i,i+1} & \ldots & \tilde{H}_{i,\nu} \\ \hat{R}_{i,i} & \hat{R}_{i,i+1} & \ldots & \hat{R}_{i,\nu} \end{pmatrix} \begin{matrix} k \\ n_i \end{matrix} = \begin{pmatrix} R^*_{i,i} & R^*_{i,i+1} & \ldots & R^*_{i\,\nu} \\ 0 & \tilde{H}_{i+1,i+1} & \ldots & \tilde{H}_{i+1,\nu} \end{pmatrix} \begin{matrix} n_i \\ k \end{matrix} \tag{3.6}$$

are first computed where $\hat{Q}_i$ is orthogonal and $R^*_{i\,i}$ is upper triangular. Then the QRD

$$\hat{Q}_\nu^T \begin{pmatrix} \tilde{H}_{\nu,\nu} \\ \hat{R}_{\nu,\nu} \end{pmatrix} = R^*_{\nu,\nu} \tag{3.7}$$

is computed so that the required upper triangular matrix is given by

$$R_n = \begin{matrix} n_1 & n_2 & & n_\nu \end{matrix} \begin{pmatrix} R^*_{1,1} & R^*_{1,2} & \ldots & R^*_{1,\nu} \\ & R^*_{2,2} & \ldots & R^*_{2,\nu} \\ & & \ddots & \vdots \\ & & & R^*_{\nu,\nu} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \\ \\ n_\nu \end{matrix} . \tag{3.8}$$

A summary of this block parallel strategy is shown in Algorithm 2.

1: Compute the factorization (3.2).
2: **for** $i = \nu - 1, \ldots, 2, 1$ **do**
3: Compute the updating QRD (3.3).
4: **end for**
5: Compute (3.5).
6: **for** $i = 1, 2, \ldots, \nu - 1$ **do**
7: Compute the factorization (3.6).
8: **end for**
9: Compute the QRD (3.7).

ALGORITHM 2. *Block-version of the serial rank-*1 *algorithm.*

The second block parallel algorithm operates on more than one block simultaneously. It is based on the recursive doubling approach and the block parallel algorithms in [2, 8]. Assume for simplicity that $n_i = k$ $(i = 1, \ldots, \nu)$ and $\nu = 2^g$. Using the partitioning of Z and R in (3.1), the QRDs

$$Q_{i,0}^T \begin{pmatrix} Z_i & R_{i,i} & \ldots & R_{i,\nu} \end{pmatrix} = \begin{pmatrix} \tilde{W}_i^{(0)} & R_{i,i}^{(0)} & \ldots & R_{i,\nu}^{(0)} \end{pmatrix} \tag{3.9}$$

are first computed simultaneously for $i = 1, \ldots, \nu$, where $\tilde{W}_i^{(0)}$ and $R_{i\,j}^{(0)}$ $(j = i, \ldots, \nu)$ are, respectively, upper triangular and full dense square matrices of order k. Then, in step $i = 1, \ldots, g$, the orthogonal factorizations

$$Q_{j,i}^T \begin{pmatrix} \tilde{W}_j^{(i-1)} & R_{j,j}^{(i-1)} & \ldots & R_{j,\nu}^{(i-1)} \\ \tilde{W}_p^{(i-1)} & 0 & \ldots & R_{p,\nu}^{(i-1)} \end{pmatrix} = \begin{pmatrix} \tilde{W}_j^{(i)} & R_{j,j}^{(i)} & \ldots & R_{j,\nu}^{(i)} \\ 0 & R_{p,j}^{(i)} & \ldots & R_{p,\nu}^{(i)} \end{pmatrix} ; p = j + 2^{(g-i)} \tag{3.10}$$

are computed simultaneously for $j = 1, \ldots, 2^{(g-i)}$, where $\tilde{W}_j^{(i)}$ is a $k \times k$ upper triangular matrix. After the gth step, $R_z = \tilde{W}_1^{(g)}$ is computed.

Using this method, the matrix $\tilde{H}$ in (2.2) is nonfull and dense. This structure can facilitate the development of efficient parallel strategies to triangularize $\tilde{H}$. One such parallel strategy is shown in Figure 3.1 for $g = 3$. The symbols ■ and ▲ denote square dense and upper triangular matrices of order k, respectively. A number in an annihilated block indicates the block-row used in the factorization. It is assumed that a single step is required to apply simultaneously a set of *disjoint* orthogonal factorizations. Note that, during a factorization step, some blocks which will be used in the subsequent factorization step are simultaneously triangularized also. This reduces the time complexity of applying the *disjoint* orthogonal factorizations, which reduces to simultaneously updating, rather than computing, a number of QRDs. In the first four steps both the Z and R matrices are shown, while in the remaining steps the operations are assumed to be performed on the $\tilde{H}$ matrix.

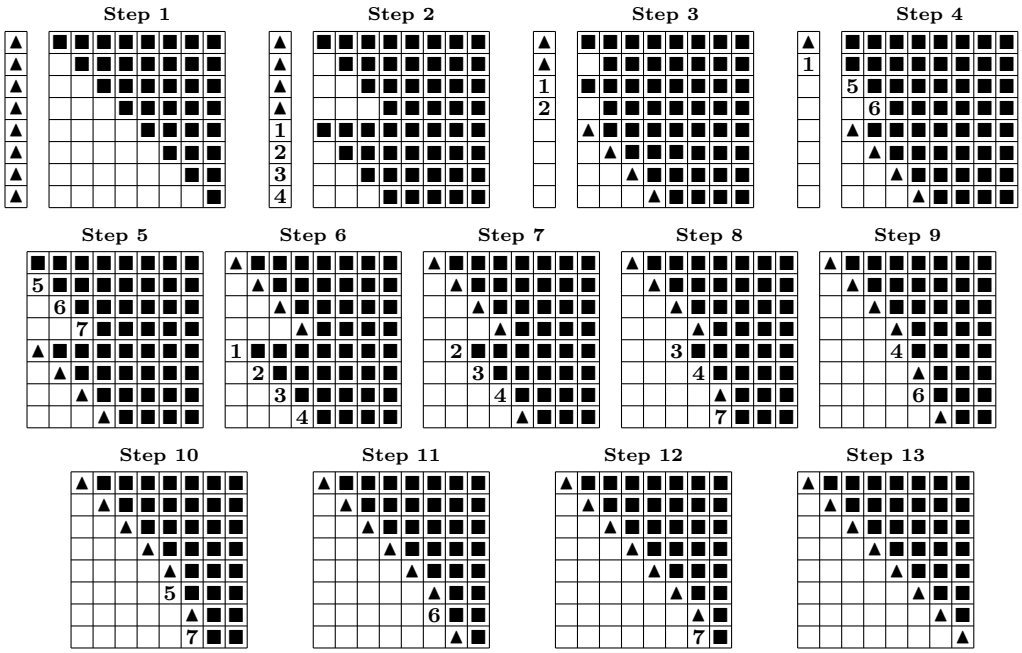


FIG. 3.1. *Solving the rank-k UQRD problem using the block-greedy rank-k algorithm.*

4. Numerical results. The main rank-k algorithms have been implemented on the 4096-processors MasPar MP-2 SIMD system. A SIMD computer comprises multiple processors which simultaneously execute an operation on parts of an array in a data parallel mode. The MasPar SIMD system is composed of a *front-end* (a DECstation 5000) and a data parallel unit (DPU). The parallel computations are executed by the processing element (PE) array in the DPU, while serial operations are performed on the front-end. The 4096 PEs of the MP-2 are arranged in an $\text{ES} \times \text{ES}$ array, where $\text{ES} = 64$. The default cyclic distribution has been used to map the matrices and vectors onto the DPU. In a cyclic distribution, an n element vector and an $m \times n$ element matrix are mapped onto $\lceil n/\text{ES}^2 \rceil$ and $\lceil m/\text{ES} \rceil \lceil n/\text{ES} \rceil$ layers of

TABLE 4.1
Execution time of the rank-k algorithms on the MasPar MP-2 SIMD system.

n	k	Househol. QRD	Givens QRD	Algor. 1 Givens	Algor. 1 Househol.	Algor. 2 Househol.	Rank-1 Algor. Givens
1408	32	130.69	768.44	156.50	73.54	41.88	967.94
1408	64	130.22	767.77	236.91	113.18	47.22	1923.39
1408	96	130.45	768.06	319.16	147.26	63.38	2885.09
1408	128	130.37	768.22	398.39	186.79	68.38	3853.06
1408	160	130.85	768.43	478.20	222.10	84.62	4804.64
1408	192	130.94	768.37	553.00	259.91	88.79	5774.98
1408	224	130.73	768.83	629.61	295.87	105.18	6731.87
1408	256	130.94	768.21	698.88	331.69	108.72	7699.97
1024	8	54.98	312.12	61.45	40.11	20.19	135.55
1024	16	55.04	311.85	64.09	40.43	21.04	270.32
1024	32	54.98	314.26	87.08	41.70	23.14	539.87
1024	64	54.85	314.11	131.40	63.84	26.46	1081.28
1024	96	55.14	314.17	175.29	82.59	35.52	1619.62
1024	128	55.10	313.95	217.43	104.40	38.26	2156.16
1024	160	55.17	313.70	259.35	123.93	47.46	2699.36
1024	192	55.16	313.84	296.65	143.80	49.56	3234.43
1024	224	55.28	313.51	334.37	163.16	58.67	3773.50
1024	256	55.77	313.77	366.53	181.61	60.42	4300.03
1024	288	55.37	314.17	398.90	200.34	69.63	4858.85
1024	320	55.42	313.83	423.61	216.21	71.04	5398.72
256	64	2.12	9.39	12.26	6.89	2.86	96.90
384	64	4.78	23.56	23.14	12.18	4.86	182.85
512	64	9.43	48.49	37.84	19.31	7.71	293.82
640	64	16.25	86.44	55.96	27.88	11.21	432.77
768	64	25.62	140.84	77.71	38.16	15.38	599.94
896	64	38.73	215.63	102.69	50.05	20.46	817.22
1152	64	75.46	434.20	162.82	78.56	32.67	1337.54
1280	64	100.35	584.85	198.24	95.02	39.63	1620.29

memory, respectively. The time to execute a single arithmetic operation on an array depends on the number of memory layers required to map the array on to the DPU [11]. Other processor mappings are available for efficiently mapping arrays onto the PE array when the default cyclic distribution is not the best choice [16]. The MasPar cannot be partitioned into smaller SIMD subarrays.

Table 4.1 shows the execution times in milliseconds (msec) for the various algorithms. The first two algorithms use Householder transformations and CDGRs to compute the QRD of $\tilde{A}$ in (1.2) without exploiting the previous QRD of A in (1.1). The SK algorithm (Algorithm 1) using CDGRs and Householder transformations to compute the second stage are shown in the following two columns, respectively. The final two columns show the execution times of Algorithm 2 and the rank-1 algorithm which is repeated k times. The block-parallel algorithm illustrated in Figure 3.1 has not been implemented since simultaneous factorization of matrices cannot be performed efficiently on the MasPar [11]. In the case of Algorithm 2, $n_\nu = k$ and $n_1 = \cdots = n_{\nu-1} = 64$, where 64 is the edge size of the 2-D SIMD array processor. The following observations can be made:

- Householder algorithms outperform equivalent Givens algorithms;
- With the exception of very small problems, the SK-algorithm employing only

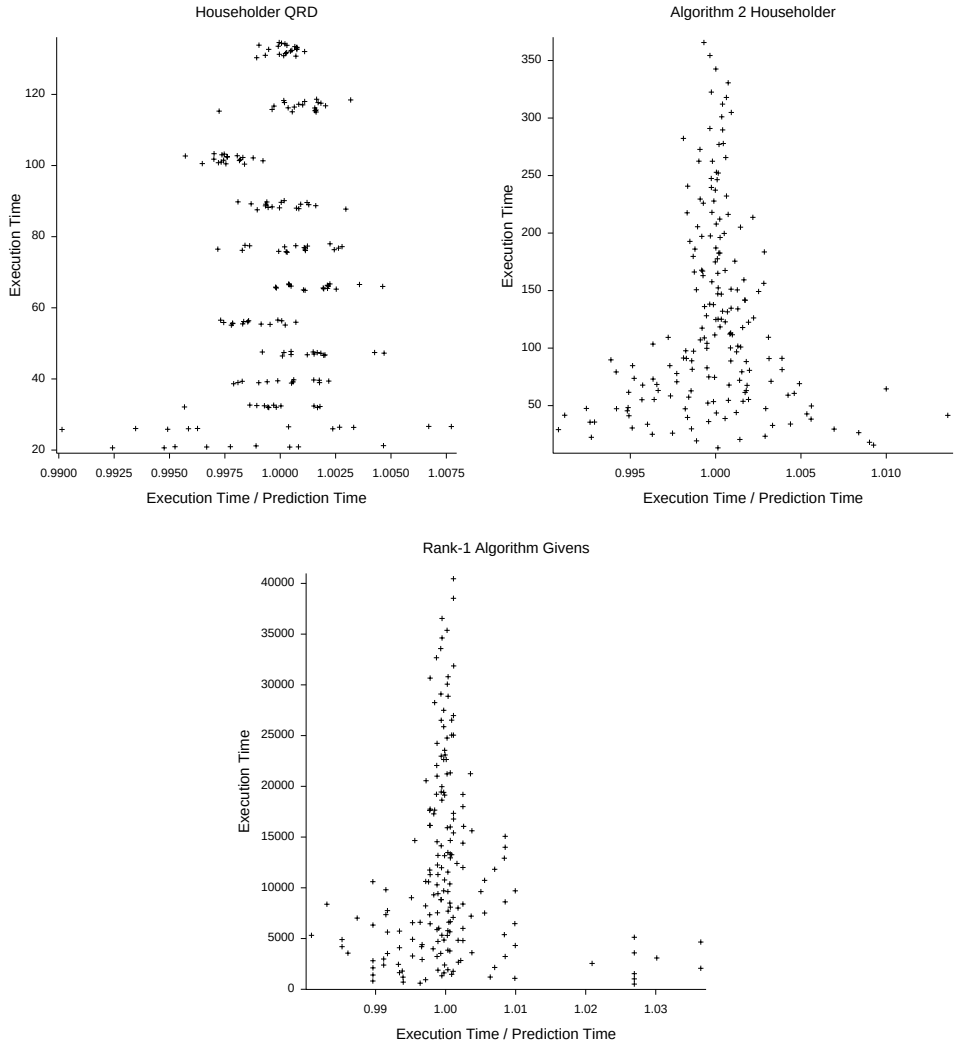


FIG. 4.1. *Ratio between actual and predicted execution times.*

CDGRs outperforms the Givens algorithm which solves the rank-k updating problem by computing the QRD afresh using the the SK-scheme;

- The performance of the SK-algorithm (Algorithm 1) improves when the second stage is computed using Householder transformations. However, this improvement in performance is sufficient to outperform the Householder algorithm in which the QRD is computed afresh only when k is very small and n is large;
- The block-parallel version of the rank-1 algorithm (Algorithm 2) performs better than the Householder QRD algorithm for fixed k and increasing n;
- The rank-1 algorithm when repeated k times results in the worst performance.

Performance models (msec $\times 10^{-3}$) for the Householder method that computes the QRD from afresh (T_{QR}), Algorithm 2 (T_{BH}), and the rank-1 algorithm (T_{R-1}) have been constructed using statistical regression methods [11, 16]. These are given

by

$$T_{QR}(N_{\mathrm{ES}}, K_{\mathrm{ES}}) = N(75.47 + 62.42N + 9.24N^2 + 0.43NK),$$

$$T_{BH}(N_{\mathrm{ES}}, K_{\mathrm{ES}}) = N(7.15 + 5.03N + 193.9K - 14.41K^2 + 36.6NK) + 1.94K^3,$$

and

$$T_{R-1}(N_{\mathrm{ES}}, K_{\mathrm{ES}}) = K(286100 + 42270N + 2646N^2).$$

It has been assumed that the dimensions of the matrices are multiples of the edge size of the array processor, that is, $n = N_{\mathrm{ES}}$ and $k = K_{\mathrm{ES}}$. These models predict accurately the execution time of the algorithms. Figure 4.1 shows the ratio of actual and predicted execution times for the three algorithms. An analysis of these performance models leads to the same conclusions as those drawn from the analysis of the execution times in Table 4.1. It should be noted that these observation are specific to SIMD systems and cannot be generalized for other parallel architectures. The performance capabilities of the algorithms on other kinds of parallel systems need to be investigated.

5. Conclusions and future research. Parallel strategies have been presented for retriangularizing an $n \times n$ upper triangular matrix R after a rank-k change. The first two algorithms are based on the SK annihilation scheme in [22] and solve the updating problem by applying $2(k + n - 2)$ CDGRs. However, these algorithms need more CDGRs than the $2n - 3$ CDGRs required to compute the QRD of $\hat{A}$ in (1.3), after forming the updating $R + ZY^T$. Two block parallel strategies based on the serial rank-1 algorithm and the *recursive doubling* method have also been described for solving the rank-k UQRD problem.

The theoretical measures of complexity of the Givens algorithms hold for shared memory machines where the number of processing elements is at least equal to the maximum number of disjoint rotations that comprise a CDGR—that is, for parallel systems of constant communication complexity that have enough processors to compute and apply simultaneously all the disjoint Givens rotations at any step of the algorithms. Generally, the performance of the algorithms will strongly depend on the hardware and software characteristics of the target parallel machine. It is expected that no single algorithm will be superior for all types of parallel architectures.

The block parallel algorithms will be suitable for execution on multiprocessor multiple instruction stream–multiple data stream (MIMD) systems because of their low communication overheads and heavy use of level 3 BLAS operations [2]. In conventional SIMD systems the use of Householder algorithms which do not take into consideration matrix structures are often found to outperform equivalent algorithms which are based on CDGRs and which exploit the nonfull dense structure of the matrices [9, 13, 15]. The advantage of employing a CDGR requiring less execution time than a single Householder transformation is offset by the large number of CDGRs applied to compute the factorizations.

The rank-k updating algorithms can be extended to solve the block downdating QRD problem and the general linear model (GLM) [4, 13, 15, 17, 20]. One of the methods for solving the block downdating QRD problem requires the QRD of the square matrix B after computing the orthogonal factorization

$$G^T \begin{pmatrix} Q & R \\ Z & 0 \end{pmatrix} = \begin{pmatrix} D & C \\ 0 & B \end{pmatrix} \begin{matrix} k \\ n \end{matrix},$$

where $\begin{pmatrix} Q^T & Z^T \end{pmatrix}$ has orthogonal rows, $Z \in \Re^{k\times k}$ and $R \in \Re^{n\times n}$ are upper triangular matrices, G is orthogonal, $|D| = I$, I is the identity matrix, $|C| = |\hat{A}|$, and $\hat{A}$ denotes the data deleted from the original data matrix. Within the context of the numerical solution of the GLM, a *generalized* QRD (GQRD) of the full column rank $Z \in \Re^{n\times k}$ $(n \geq k)$ and an $n \times n$ upper triangular matrix R is computed [1, 7]. The GQRD of Z and R is given by

$$Q^T Z = \begin{pmatrix} R_z \\ 0 \end{pmatrix} \begin{matrix} k \\ n-k \end{matrix}, \qquad (Q^T R)P = R_n,$$

where Q and P are $n \times n$ orthogonal matrices and R_z and R_n are upper triangular [18, 21]. In some econometric applications, Z, and consequently, R_z, are block-diagonals and R is defined by the Kronecker product $C \otimes I$, where C is upper triangular [10, 12, 14]. It may be observed that the common feature of the rank-k UQRD problem and the factorizations described here is the retriangularization of a triangular matrix after it has been premultiplied by the orthogonal matrix of a QRD. Block generalization of the parallel strategies reported in [7, 13] is currently being considered for solving the downdating and GLM problems by exploiting their special properties.

Acknowledgments. The author is grateful to Maurice Clint, Denis Parkinson, the anonymous referee, and the associate editor for their valuable comments and suggestions. The author would like also to thank the Institut für Informatik, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Germany for providing him access to the MasPar MP-2.

REFERENCES

[1] E. Anderson, Z. Bai, and J. J. Dongarra, *Generalized QR factorization and its applications*, Linear Algebra Appl., 162 (1992), pp. 243–271.
[2] M. W. Berry, J. J. Dongarra, and Y. Kim, *A parallel algorithm for the reduction of a nonsymmetric matrix to block upper-Hessenberg form*, Parallel Comput., 21 (1995), pp. 1189–1211.
[3] M. Cosnard, J.-M. Muller, and Y. Robert, *Parallel QR decomposition of a rectangular matrix*, Numer. Math., 48 (1986), pp. 239–249.
[4] L. Eldén and H. Park, *Block downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1018–1034.
[5] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.
[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[7] E. J. Kontoghiorghes, *Parallel Givens sequences for solving the general linear model on a EREW PRAM*, Parallel Algorithms Appl., 15 (2000), pp. 57–75.
[8] E. J. Kontoghiorghes, *New parallel strategies for block updating the QR decomposition*, Parallel Algorithms Appl., 5 (1995), pp. 229–239.
[9] E. J. Kontoghiorghes, *Ordinary linear model estimation on a massively parallel SIMD computer*, Concurrency: Practice and Experience, 11 (1999), pp. 323–341.
[10] E. J. Kontoghiorghes, *Parallel strategies for computing the orthogonal factorizations used in the estimation of econometric models*, Algorithmica, 25 (1999), pp. 58–74.
[11] E. J. Kontoghiorghes, *Parallel Algorithms for Linear Models: Numerical Methods and Estimation Problems*, Advances in Computational Economics 15, Kluwer Academic Publishers, Boston, MA, 2000.
[12] E. J. Kontoghiorghes, *Parallel strategies for solving SURE models with variance inequalities and positivity of correlations constraints*, Comput. Econ., 15 (2000), pp. 89–106.
[13] E. J. Kontoghiorghes and M. R. B. Clarke, *Solving the updated and downdated ordinary linear model on massively parallel SIMD systems*, Parallel Algorithms Appl., 1 (1993), pp. 243–252.

[14] E. J. Kontoghiorghes and M. R. B. Clarke, *An alternative approach for the numerical solution of seemingly unrelated regression equations models*, Comput. Statist. Data Anal., 19 (1995), pp. 369–377.

[15] E. J. Kontoghiorghes and M. R. B. Clarke, *Solving the general linear model on a SIMD array processor*, Comput. Artificial Intelligence, 14 (1995), pp. 353–370.

[16] E. J. Kontoghiorghes, M. Clint, and H.-H. Nägeli, *Recursive least–squares using hybrid Householder algorithms on massively parallel SIMD systems*, Parallel Comput., 25 (1999), pp. 1147–1159.

[17] S. Kourouklis and C. C. Paige, *A constrained least squares approach to the general Gauss–Markov linear model*, J. Amer. Statist. Assoc., 76 (1981), pp. 620–625.

[18] B. De Moor and P. Van Dooren, *Generalizations of the singular value and QR decompositions*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 993–1014.

[19] J. J. Modi and M. R. B. Clarke, *An alternative Givens ordering*, Numer. Math., 43 (1984), pp. 83–90.

[20] C. C. Paige, *Numerically stable computations for general univariate linear models*, Comm. Statist. Simulation Comput., 7 (1978), pp. 437–453.

[21] C. C. Paige, *Some Aspects of Generalized QR Factorizations*, in Reliable Numerical Computation, M. G. Cox and S. J. Hammarling, eds., Clarendon Press, Oxford, UK, 1990, pp. 71–91.

[22] A. H. Sameh and D. J. Kuck, *On stable parallel linear system solvers*, J. ACM, 25 (1978), pp. 81–91.

[23] G. M. Shroff and C. H. Bishof, *Adaptive condition estimation for rank-one updates of QR factorizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1264–1278.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 726–751

DIFFERENCES IN THE EFFECTS OF ROUNDING ERRORS IN KRYLOV SOLVERS FOR SYMMETRIC INDEFINITE LINEAR SYSTEMS*

GERARD L. G. SLEIJPEN†, HENK A. VAN DER VORST†, AND JAN MODERSITZKI‡

Abstract. The three-term Lanczos process for a symmetric matrix leads to bases for Krylov subspaces of increasing dimension. The Lanczos basis, together with the recurrence coefficients, can be used for the solution of symmetric indefinite linear systems, by solving a reduced system in one way or another. This leads to well-known methods: MINRES (minimal residual), GMRES (generalized minimal residual), and SYMMLQ (symmetric LQ). We will discuss in what way and to what extent these approaches differ in their sensitivity to rounding errors.

In our analysis we will assume that the Lanczos basis is generated in exactly the same way for the different methods, and we will not consider the errors in the Lanczos process itself. We will show that the method of solution may lead, under certain circumstances, to large additional errors, which are not corrected by continuing the iteration process.

Our findings are supported and illustrated by numerical examples.

Key words. linear systems, iterative methods, MINRES, GMRES, SYMMLQ, stability

AMS subject classifications. 65F10, 65N12

PII. S0895479897323087

1. Introduction.

We consider iterative methods for the construction of approximations to the solution of a linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is supposed to be a real symmetric n by n matrix. Without loss of generality, we assume $\mathbf{x}_0 = 0$. Let $\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}_k$ (in particular, $\mathbf{r}_0 = \mathbf{b}$) and

$$\mathcal{K}_k(\mathbf{A}; \mathbf{b}) \equiv \text{Span}\{\mathbf{b}, \mathbf{Ab}, \ldots, \mathbf{A}^{k-1}\mathbf{b}\},$$

the k-dimensional Krylov subspace. The methods to be analyzed build the iterates $\mathbf{x}_k$ such that

1. $\mathbf{x}_k \in \mathcal{K}_k(\mathbf{A}; \mathbf{b})$ and $\|\mathbf{b} - \mathbf{Ax}_k\|_2 = \min$ (GMRES, MINRES),
2. $\mathbf{x}_k \in \mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{b})$ and $\|\mathbf{A}^{-1}\mathbf{b} - \mathbf{x}_k\|_2 = \min$ (SYMMLQ).

With the standard three-term Lanczos process, we generate an orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_k$ for $\mathcal{K}_k(\mathbf{A}; \mathbf{b})$, with $\mathbf{v}_1 \equiv \mathbf{b}/\|\mathbf{b}\|_2$. The three-term Lanczos process can be recast in matrix formulation as

$$\mathbf{AV}_k = \mathbf{V}_{k+1}\underline{T}_k, \tag{1}$$

in which $\mathbf{V}_j$ is defined as the n by j matrix with columns $\mathbf{v}_1, \ldots, \mathbf{v}_j$, and $\underline{T}_k$ is a $k+1$ by k tridiagonal matrix.

Paige [9] has shown that in finite precision arithmetic, the Lanczos process can be implemented so that the *computed* $\mathbf{V}_{k+1}$ and $\underline{T}_k$ satisfy

$$\mathbf{AV}_k = \mathbf{V}_{k+1}\underline{T}_k + \mathbf{F}_k, \tag{2}$$

*Received by the editors June 17, 1997; accepted for publication (in revised form) by Z. Strakoš March 28, 2000; published electronically October 25, 2000.
http://www.siam.org/journals/simax/22-3/32308.html

†Mathematical Institute, Utrecht University, P.O. Box 80.010, 3508 TA Utrecht, The Netherlands (sleijpen@math.uu.nl, vorst@math.uu.nl).

‡Institute of Mathematics, Medical University of Lübeck, Wallstraße 40, 23560 Lübeck, Germany (modersitzki@math.mu-luebeck.de).

with, under mild conditions for k,

$$\|\mathbf{F}_k\|_2 \leq 2\sqrt{k}\,(7\|\mathbf{A}\|_2 + m_1\|\,|\mathbf{A}|\,\|_2)\,\mathbf{u}$$

($\mathbf{u}$ is the machine precision, and m_1 denotes the maximum number of nonzeros in any row of $\mathbf{A}$). Since $\|\,|\mathbf{A}|\,\|_2 \leq \sqrt{m_1}\,\|\mathbf{A}\|_2$ (see Lemma A.1), we obtain the convenient expression

$$\|\mathbf{F}_k\|_2 \leq 2\sqrt{k}\,(7 + m_1\sqrt{m_1}\,)\,\|\mathbf{A}\|_2\,\mathbf{u}. \tag{3}$$

Popular Krylov subspace methods for symmetric linear systems can be derived with formula (1) as a starting point: MINRES, GMRES (adapted to symmetric matrices; see below), and SYMMLQ. The matrix $\underline{T}_k$ can be interpreted as the restriction of $\mathbf{A}$ with respect to the Krylov subspace, and the main idea behind these Krylov solution methods is that the given system $\mathbf{Ax} = \mathbf{b}$ is replaced by a smaller system with $\underline{T}_k$ over the Krylov subspace. This reduced system is solved—implicitly or explicitly—in a convenient way and the solution is transformed with $\mathbf{V}_k$ to a solution in the original n-dimensional space. The main computational differences between the methods are due to a different way of solution of the reduced system and to differences in the back transformation to an approximate solution of the original system. We will describe these differences in relevant detail in coming sections.

Of course, these methods have been derived assuming exact arithmetic; for instance, the generating formulas are all based on an exact orthogonal basis for the Krylov subspace. In numerical reality, however, we have to compute this basis, as well as all other quantities in the methods, and then it is of importance to know how the generating formulas behave in finite precision arithmetic. The errors in the underlying Lanczos process have been analyzed by Paige [9, 10]. It has been proven by Greenbaum and Strakoš [7] that rounding errors in the Lanczos process may have a delaying effect on the convergence of iterative solvers but do not prevent eventual convergence in general. Usually, a rigorous error analysis is on a worst case scenario, and as a consequence, the error bounds cannot very well be used to explain differences between these methods, as observed in practical situations.

In this paper, we propose a different way of analyzing these methods, different in the way that we do not attempt to derive sharper upper bounds, but that we try to derive upper bounds for relevant differences between these processes in finite precision arithmetic. This will not help us to understand why any of these methods converges in finite precision, but it will give us some insight in answering practical questions such as the following.

• When and why is MINRES less accurate than SYMMLQ? This question was already posed in the original publication [11], but the answer in [11, p. 625] is largely speculative.

• Is MINRES suspect for ill-conditioned systems, because of the minimal residual approach (see [11, p. 619])? Hints are given for the explanation of the observation that MINRES may be more inaccurate than SYMMLQ [11, p. 625]. We will further substantiate this. In [2, p. 43] an explicit relation is suggested between MINRES and working with $\mathbf{A}^2$, and it is argued that its sensitivity to rounding errors of the solution depends on $\kappa_2(\mathbf{A})^2$. (It is even stated: 'the squared condition number of $\mathbf{A}^2$', implying $\kappa_2(\mathbf{A}^2)^2 = \kappa_2(\mathbf{A})^4$, which seems to be an unlucky formulation.)

• Why and when does SYMMLQ converge slower than, for instance, MINRES or GMRES?

```
Choose x_0
x = x_0,  r = b − Ax,  ρ = ||r||,  v = r/ρ
β = 0,  β̃ = 0,  c = −1,  s = 0
v_old = 0,  w = 0,  w̃̃ = v
while |ρ| > tol do
   ṽ ← A v − β v_old
   α ← v* ṽ,  ṽ ← ṽ − α v
   β ← ||ṽ||,  v_old ← v,  v ← ṽ/β
   ℓ_1 ← s α − c β̃,  ℓ_2 ← s β
   α̃ ← −s β̃ − c α,  β̃ ← c β
   ℓ_0 ← sqrt(α̃^2 + β^2),  c ← α̃/ℓ_0,  s ← β/ℓ_0
   w̃ ← w̃̃ − ℓ_1 w,  w̃̃ ← v − ℓ_2 w
   w ← w̃/ℓ_0
   x ← x + (ρ c) w,  ρ ← s ρ
end while
```

FIG. 1. *The* MINRES *algorithm.*

• Why does MINRES sometimes lead to rather large residuals, whereas the error in the approximation is significantly smaller? See, for instance, observations on this made in [11, p. 626]. Most important, understanding the differences between these methods will help us in making a choice.

We will now briefly characterize the different methods in our investigation.

1. **MINRES** (see [11]): Determine $\mathbf{x}_k = \mathbf{V}_k y_k$, $y_k \in \mathbb{R}^k$, such that $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2$ is minimal. This minimization leads to a small system with $\underline{T}_k$, and the tridiagonal structure of $\underline{T}_k$ is exploited to get a short recurrence relation for $\mathbf{x}_k$. The advantage of this is that only three vectors from the Krylov subspace have to be saved (in fact, MINRES works with transformed basis vectors; this will be explained in section 2.3). For the implementation of MINRES that we have used, see Figure 1.
2. **GMRES** (see [13]): This method also minimizes, for $y_k \in \mathbb{R}^k$, the residual $\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|_2$. GMRES was designed for unsymmetric matrices for which the orthogonalization of the Krylov basis is done with Arnoldi's method. This leads to a small upper Hessenberg system that has to be solved. However, when $\mathbf{A}$ is symmetric, then, in exact arithmetic, the Arnoldi method is equivalent to the Lanczos method (see also [6, p. 41]). Although GMRES is commonly presented with an Arnoldi basis, there are various implementations of it that differ in finite precision, for instance, with modified Gram–Schmidt, classical Gram–Schmidt, Householder, and other variants. We view Lanczos as one way to obtain an orthogonal basis, and therefore, we stick to the name GMRES. However, in order to stress the fact that our version of GMRES relies on Lanczos, we will use the notation GMRES*.

   Due to the way of solution in GMRES* (and in GMRES), all the basis

Choose $\mathbf{x}_0$

$\mathbf{x} = \mathbf{x}_0,\ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x},\ \rho = \|\mathbf{r}\|,\ \mathbf{v} = \mathbf{r}/\rho$

$\beta = 0,\ \widetilde{\beta} = 0,\ c = -1,\ s = 0$

$\mathbf{v}_{\text{old}} = \mathbf{0},\ \mathbf{V} = [\,],\ z = [\,],\ k = 0$

while $\rho > tol$ **do**

  $\mathbf{V} \leftarrow [\mathbf{V}, \mathbf{v}],\ k \leftarrow k + 1$

  $\widetilde{\mathbf{v}} \leftarrow \mathbf{A}\mathbf{v} - \beta\,\mathbf{v}_{\text{old}}$

  $\alpha \leftarrow \mathbf{v}^*\widetilde{\mathbf{v}},\ \widetilde{\mathbf{v}} \leftarrow \widetilde{\mathbf{v}} - \alpha\,\mathbf{v}$

  $\beta \leftarrow \|\widetilde{\mathbf{v}}\|,\ \mathbf{v}_{\text{old}} \leftarrow \mathbf{v},\ \mathbf{v} \leftarrow \widetilde{\mathbf{v}}/\beta$

  $\ell_1 \leftarrow s\,\alpha - c\,\widetilde{\beta},\ \ell_2 \leftarrow s\,\beta$

  $\widetilde{\alpha} \leftarrow -s\,\widetilde{\beta} - c\,\alpha,\ \widetilde{\beta} \leftarrow c\,\beta$

  $\ell_0 \leftarrow \sqrt{\widetilde{\alpha}^2 + \beta^2},\ c \leftarrow \widetilde{\alpha}/\ell_0,\ s \leftarrow \beta/\ell_0$

  if $k = 1$

    $\vec{\ell} = [\,],\ R = [\,\ell_0\,]$

  else

    $R \leftarrow \begin{bmatrix} R \\ \vec{0} \end{bmatrix},\ \vec{\ell} \leftarrow [\,\vec{\ell}, \ell_1, \ell_0\,]$

    $R \leftarrow [\,R, \vec{\ell}^{\,T}\,],\ \vec{\ell} \leftarrow [\,\vec{0}, \ell_2\,]$

  end if

  $z \leftarrow [\,z^{\,T}, c\rho\,]^{\,T},\ \rho \leftarrow s\,\rho$

end while

$\mathbf{x} = \mathbf{x} + \mathbf{V}(R^{-1}z)$

FIG. 2. *The* GMRES* *algorithm. The vector* $\vec{0}$ *for the expansion of the upper triangular matrix* R *is a row vector of zeros of appropriate size (different size at different occurrences).*

Choose $\mathbf{x}_0$

$\mathbf{x} = \mathbf{x}_0,\ \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x},\ \rho = \|\mathbf{r}\|,\ \mathbf{v} = \mathbf{r}/\rho$

$\beta = 0,\ \widetilde{\beta} = 0,\ c = -1,\ s = 0,\ \kappa = \rho$

$\mathbf{v}_{\text{old}} = \mathbf{0},\ \mathbf{w} = \mathbf{v},\ g = 0,\ \widetilde{\widetilde{g}} = \rho$

while $\kappa > tol$ **do**

  $\widetilde{\mathbf{v}} \leftarrow \mathbf{A}\mathbf{v} - \beta\,\mathbf{v}_{\text{old}}$

  $\alpha \leftarrow \mathbf{v}^*\widetilde{\mathbf{v}},\ \widetilde{\mathbf{v}} \leftarrow \widetilde{\mathbf{v}} - \alpha\,\mathbf{v}$

  $\beta \leftarrow \|\widetilde{\mathbf{v}}\|,\ \mathbf{v}_{\text{old}} \leftarrow \mathbf{v},\ \mathbf{v} \leftarrow \widetilde{\mathbf{v}}/\beta$

  $\ell_1 \leftarrow s\,\alpha - c\,\widetilde{\beta},\ \ell_2 \leftarrow s\,\beta$

  $\widetilde{\alpha} \leftarrow -s\,\widetilde{\beta} - c\,\alpha,\ \widetilde{\beta} \leftarrow c\,\beta$

  $\ell_0 \leftarrow \sqrt{\widetilde{\alpha}^2 + \beta^2},\ c \leftarrow \widetilde{\alpha}/\ell_0,\ s \leftarrow \beta/\ell_0$

  $\widetilde{g} \leftarrow \widetilde{\widetilde{g}} - \ell_1\,g,\ \widetilde{\widetilde{g}} \leftarrow -\ell_2\,g,\ g \leftarrow \widetilde{g}/\ell_0$

  $\mathbf{x} \leftarrow \mathbf{x} + (g\,c)\,\mathbf{w} + (g\,s)\,\mathbf{v}$

  $\mathbf{w} \leftarrow s\,\mathbf{w} - c\,\mathbf{v},\ \kappa \leftarrow \sqrt{\widetilde{g}^2 + \widetilde{\widetilde{g}}^2}$

end while

FIG. 3. *The* SYMMLQ *algorithm.*

vectors $\mathbf{v}_j$ have to be stored. For our implementation of GMRES*, see Figure 2.

3. **SYMMLQ** (see [11]): Determine $\mathbf{x}_k = \mathbf{A}\mathbf{V}_k y_k$, $y_k \in \mathbb{R}^k$, such that the error $\mathbf{x} - \mathbf{x}_k$ has minimal Euclidean length. It may come as a surprise that $\|\mathbf{x} - \mathbf{x}_k\|_2$ can be minimized without knowing $\mathbf{x}$, but this can be accomplished by restricting the choice of $\mathbf{x}_k$ to $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{b})$. Conjugate gradient approximations can, if they exist, be computed with little effort from the SYMMLQ information. In the SYMMLQ implementation suggested in [11] this is used to terminate iterations either at a SYMMLQ iterate or a conjugate gradient iterate, depending on which one is best. For the implementation of SYMMLQ that we have used, see Figure 3.

Note that these methods can be carried out with exactly the same basis vectors $\mathbf{v}_j$ and tridiagonal matrices $\underline{T}_j$.

Notations. Quantities associated with n dimensional spaces will be represented in boldface, like $\mathbf{A}$ and $\mathbf{v}_j$. Vectors and matrices on low dimensional subspaces are

denoted in normal mode: T, y. Constants will be denoted by lowercase Greek symbols, with the exception that we will use $\mathbf{u}$ to denote the relative machine precision. The absolute value of a matrix refers to elementwise absolute values, that is, $|A| = (|a_{ij}|)$ for $A = (a_{ij})$.

Most of our bounds on perturbations in the solutions at the kth iteration step will be expressed as bounds for corresponding perturbations to the residual in the kth step, relative to the norm of an initial residual. Since all these iteration methods construct their search spaces from residual vector information (that is, they all start with $\mathbf{r}_0 = \mathbf{b}$), and since we make at least errors in the order of $\mathbf{u}\,\|\mathbf{b}\|_2$ in the computation of the residuals, we may not expect perturbations of order less than $\mathbf{u}\,\kappa_2(\mathbf{A})\|\mathbf{b}\|_2$ in the iteratively computed solutions. So, our bounds can only be expected to show up in the computed residuals, if the errors are larger than the error induced by the computation of the residuals itself.

2. Differences in round-off errors for MINRES and GMRES*.

2.1. The basic formulas for GMRES* and MINRES in exact arithmetic.

We will first describe the generic formulas for the iterative methods MINRES and GMRES*, and we will assume *exact arithmetic* in the derivation of these formulas.

The aim is to minimize $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ over the Krylov subspace, and since

$$\begin{aligned} \|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|_2 &= \|\mathbf{b} - \mathbf{A}\mathbf{V}_k\, y_k\|_2 \\ &= \|\mathbf{b} - \mathbf{V}_{k+1}\,\underline{T}_k\, y_k\|_2 \\ &= \|\underline{T}_k\, y_k - \|\mathbf{b}\|_2\, e_1\|_2, \end{aligned} \tag{4}$$

we see that a minimizer y_k must be the linear least squares solution of the $k+1$ by k overdetermined system

$$\underline{T}_k\, y_k = \|\mathbf{b}\|_2 e_1.$$

This system is solved with Givens rotations, which leads to an upper triangular reduction of $\underline{T}_k$,

$$\underline{T}_k = \underline{Q}_k R_k, \tag{5}$$

in which R_k is k by k upper triangular with bandwidth 3 and $\underline{Q}_k$ is a $k+1$ by k matrix with orthonormal columns. Using (5), y_k can be solved from

$$R_k\, y_k = z_k \equiv \|\mathbf{b}\|_2\, \underline{Q}_k^T e_1,$$

and since $\mathbf{x}_k = \mathbf{V}_k y_k$, we obtain

$$\mathbf{x}_k = \mathbf{V}_k R_k^{-1} \underline{Q}_k^T \|\mathbf{b}\|_2 e_1 = \mathbf{V}_k R_k^{-1} z_k. \tag{6}$$

The GMRES method, proposed for unsymmetric $\mathbf{A}$ in [13], can be characterized by the specific order of computation in the above derivation, indicated by adding parentheses:

$$\mathbf{x}_k = \mathbf{V}_k (R_k^{-1} \underline{Q}_k^T \|\mathbf{b}\|_2 e_1) = \mathbf{V}_k (R_k^{-1} z_k). \tag{7}$$

When $\mathbf{A}$ is symmetric, then Arnoldi's method is equivalent to Lanczos's method, so that (7) describes GMRES for symmetric $\mathbf{A}$ (further referred to as GMRES*). The

well-known disadvantage of this approach is that we have to store all columns of $\mathbf{V}_k$ for the computation of $\mathbf{x}_k$.

MINRES follows essentially the same approach as GMRES for the minimization of the residual, but it exploits the banded structure of R_k in order to get short recurrences for $\mathbf{x}_k$ and in order to save on memory storage.

Indeed, the computations in the generating formula (6) can be grouped as

$$\mathbf{x}_k = \left(\mathbf{V}_k R_k^{-1}\right) z_k \equiv \mathbf{W}_k\, z_k. \tag{8}$$

For the computation of $\mathbf{W}_k = \mathbf{V}_k R_k^{-1}$, it is easy to see that the last column of $\mathbf{W}_k$ is obtained from the last two columns of $\mathbf{W}_{k-1}$ and $\mathbf{v}_k$. This makes it possible to update $\mathbf{x}_{k-1} = \mathbf{W}_{k-1}\, z_{k-1}$ to $\mathbf{x}_k$ with a short recurrence, since z_k follows from the kth Givens rotation applied to the vector $(z_{k-1}^T, 0)^T$. This interpretation leads to MINRES.

We see that MINRES and GMRES* both use $\mathbf{V}_k$, R_k, $\underline{T}_k$, $\underline{Q}_k$, and z_k for the computation of $\mathbf{x}_k$. Of course, we are not forced to compute these quantities in exactly the same way for the two methods, but there is no reason to compute them differently. Therefore, we will compare implementations of GMRES* and MINRES that are based on *exactly the same* quantities in floating point finite arithmetic.

From now on we will study in what way MINRES and GMRES* differ in finite precision arithmetic, given exactly the same set $\mathbf{V}_k$, R_k, $\underline{T}_k$, $\underline{Q}_k$, and z_k (all computed in finite precision, too) for the two different methods. Hence, the differences in finite precision between GMRES* and MINRES are only caused by a different order of computation of the formula $\mathbf{x}_k = \mathbf{V}_k R_k^{-1} z_k$, namely,

$$\text{for GMRES}^*\text{:} \quad \mathbf{x}_k = \mathbf{V}_k \left(R_k^{-1} z_k\right), \tag{9}$$

$$\text{for MINRES:} \quad \mathbf{x}_k = \left(\mathbf{V}_k R_k^{-1}\right) z_k. \tag{10}$$

In finite precision, the relation (5) will not be satisfied exactly. Instead, we have that [8, Theorem 18.4]

$$\underline{T}_k = \underline{Q}_k R_k + \underline{G}_k, \quad \text{where} \quad \|\underline{G}_k\|_F \le c\, k^2\, \mathbf{u}\, \|\underline{T}_k\|_F + \mathcal{O}(\mathbf{u}^2), \tag{11}$$

with c a modest constant. The matrix $\underline{Q}_k$ is orthogonal; it is the product of the exact Givens rotations involved in the elimination of subdiagonal elements in the actually computed reductions of $\underline{T}_k$.

2.2. Error analysis for GMRES*. In order to understand the difference between GMRES* and MINRES, we will study in this section the computational errors in $\mathbf{V}_k \left(R_k^{-1} z_k\right)$, with respect to the exactly evaluated $\mathbf{V}_k R_k^{-1} z_k$ (given the computed $\mathbf{V}_k$, R_k, and z_k). We will indicate actual computation in floating point finite precision arithmetic by *fl*, and the result will be denoted by a $\widehat{\ }$. Then, according to [4, p. 89], in floating point arithmetic the computed solution $\widehat{y}_k = fl(R_k^{-1} z_k)$ satisfies

$$(R_k + \Delta_R)\widehat{y}_k = z_k, \quad \text{with} \quad |\Delta_R| \le 3\,\mathbf{u}\,|R_k| + \mathcal{O}(\mathbf{u}^2). \tag{12}$$

This implies that $\widehat{y}_k = (I + R_k^{-1}\Delta_R)^{-1} R_k^{-1} z_k$, so that, apart from second order terms in $\mathbf{u}$, the error Δ_1 in the computation of y_k is

$$\Delta_1 \equiv \widehat{y}_k - y_k = -R_k^{-1}\Delta_R R_k^{-1} z_k.$$

Here $y_k = R_k^{-1} z_k$: y_k is the exact value based on the computed R_k and z_k. Then we also make errors in the computation of $\mathbf{x}_k$, that is, we compute $\widehat{\mathbf{x}}_k = fl(\mathbf{V}_k \widehat{y}_k)$. With the error bounds for the matrix vector product [8, p. 76], we obtain

$$\widehat{\mathbf{x}}_k = \mathbf{V}_k \widehat{y}_k + \Delta_2, \quad \text{with} \quad |\Delta_2| \le k\,\mathbf{u}\,|\mathbf{V}_k|\,|y_k| + \mathcal{O}(\mathbf{u}^2). \tag{13}$$

Hence, the error $\Delta\mathbf{x}_k = \widehat{\mathbf{x}}_k - \mathbf{x}_k$ (where $\mathbf{x}_k = \mathbf{V}_k R_k^{-1} z_k$), which can be attributed to the evaluation of the generating formula (9) for GMRES*, has two components:

$$\Delta\mathbf{x}_k = \mathbf{V}_k \Delta_1 + \Delta_2. \tag{14}$$

This error leads to a contribution $\Delta\mathbf{r}_k$ to the residual, that is, $\Delta\mathbf{r}_k$ is that part of $\mathbf{r}_k$ that can be attributed to errors in the evaluation of (9) (ignoring $\mathcal{O}(\mathbf{u}^2)$ terms):

$$\begin{aligned} \Delta\mathbf{r}_k \equiv \widehat{\mathbf{r}}_k - \mathbf{r}_k &= -\mathbf{A}\Delta\mathbf{x}_k \\ &= -\mathbf{A}\mathbf{V}_k\Delta_1 - \mathbf{A}\Delta_2 r \\ &= \mathbf{A}\mathbf{V}_k R_k^{-1}\Delta_R R_k^{-1} z_k - \mathbf{A}\Delta_2 \\ &= \mathbf{V}_{k+1}\underline{T}_k R_k^{-1}\Delta_R R_k^{-1} z_k - \mathbf{A}\Delta_2 \\ &= \mathbf{V}_{k+1}\underline{Q}_k \Delta_R R_k^{-1} z_k - \mathbf{A}\Delta_2. \end{aligned} \tag{15}$$

Note that in finite precision we have that $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\underline{T}_k + \mathbf{F}_k$, and that, because of (3), the term $\mathbf{F}_k$ leads to an additional contribution of $\mathcal{O}(\mathbf{u}^2)$ in $\Delta\mathbf{r}_k$. This is also the case in forthcoming situations where we replace $\mathbf{A}\mathbf{V}_k$ by $\mathbf{V}_{k+1}\underline{T}_k$ in the derivation of upper bounds for error contributions. In a similar way, the error term $\underline{G}_k R_k^{-1}$ in the formula for $\underline{T}_k R_k^{-1}$ (see (11)) leads to a $\mathcal{O}(\mathbf{u}^2)$ term.

Using the bound in (12) and the bound for Δ_2, we get (skipping higher order terms in $\mathbf{u}$)

$$\begin{aligned} \|\Delta\mathbf{r}_k\|_2 &\le \|\mathbf{V}_{k+1}\underline{Q}_k\|_2\, 3\,\mathbf{u}\, \| \,|R_k|\,\|_2\, \|R_k^{-1} z_k\|_2 + k\,\mathbf{u}\,\|\mathbf{A}\|_2\, \| \,|\mathbf{V}_k|\,\|_2 \|y_k\|_2 \\ &\le 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_{k+1}\|_2\,\|R_k\|_2\,\|R_k^{-1} z_k\|_2 + k\sqrt{k}\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|R_k^{-1} z_k\|_2 \\ &\le 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_{k+1}\|_2\,\kappa_2(R_k)\,\|\mathbf{b}\|_2 + k\sqrt{k}\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|R_k^{-1}\|_2\,\|\mathbf{b}\|_2. \end{aligned}$$

Here we have used that $\| \,|R_k|\,\|_2 \le \sqrt{3}\,\|R_k\|_2$ (which follows from [15, Theorem 4.2]; see Lemma A.1 for details) and $\| \,|\mathbf{V}_k|\,\|_2 \le \|\mathbf{V}_k\|_F \le \sqrt{k}$. The factor κ_2 denotes the condition number with respect to the Euclidean norm.

Note that we could bound $\|\mathbf{V}_{k+1}\|_2$ by

$$\|\mathbf{V}_{k+1}\|_2 \le \sqrt{k+1},$$

which is, because of the local orthogonality of the $\mathbf{v}_j$, a crude overestimate. According to [12, p. 267 (bottom)], it may be more realistic to replace this factor $\sqrt{k+1}$ by a factor $\sqrt{m}$, where m denotes the maximum number of times that a Ritz value of T_k has converged to any eigenvalue of $\mathbf{A}$. When solving a linear system, this value of m is usually small, e.g., 2 or 3.

We would like to replace R_k in the error bounds by something that can directly be related to $\mathbf{A}$. Therefore, we note that

$$R_k^T R_k = \underline{T}_k^T \underline{T}_k,$$

ignoring errors in the order of $\mathbf{u}$.

It has been shown in [5, 7] that the matrix $\underline{T}_k$ that has been obtained in finite precision arithmetic may be interpreted as the exact Lanczos matrix obtained from a matrix $\widetilde{\mathbf{A}}$ in which eigenvalues of $\mathbf{A}$ are replaced by multiplets. Each multiplet contains eigenvalues that differ by $\mathcal{O}(\mathbf{u}^{\frac{1}{4}})$ from an original eigenvalue of $\mathbf{A}$.[1] With

[1] This order of difference is pessimistic; factors proportional to $\mathbf{u}^{\frac{1}{2}}$, or even $\mathbf{u}$, are more likely but have not been proved [6, section 4.4.2].

$\widetilde{\mathbf{V}}_k$ we denote the orthogonal matrix that generates $\underline{T}_k$, in exact arithmetic, from $\widetilde{\mathbf{A}}$. Hence,

$$\underline{T}_k^T \underline{T}_k = \widetilde{\mathbf{V}}_k^T \widetilde{\mathbf{A}}^T \widetilde{\mathbf{A}} \widetilde{\mathbf{V}}_k,$$

so that

$$\sigma_{\min}(R_k^T R_k) \geq \sigma_{\min}(\widetilde{\mathbf{A}}^T \widetilde{\mathbf{A}}) \quad \text{and} \quad \sigma_{\max}(R_k^T R_k) \leq \sigma_{\max}(\widetilde{\mathbf{A}}^T \widetilde{\mathbf{A}}),$$

which implies (ignoring errors proportional to mild orders of $\mathbf{u}$)

$$\kappa_2(R_k) \leq \kappa_2(\widetilde{\mathbf{A}}) = \kappa_2(\mathbf{A}). \tag{16}$$

This finally results in an upper bound for the error in the residual for GMRES*, which can be attributed to the evaluation of the generating formula (9):

$$\frac{\|\Delta \mathbf{r}_k\|_2}{\|\mathbf{b}\|_2} \leq (3\sqrt{3}\,\|\mathbf{V}_{k+1}\|_2 + k\sqrt{k})\,\mathbf{u}\,\kappa_2(\mathbf{A}). \tag{17}$$

Note that, even if there were only rounding errors in the representation of $\mathbf{A}$ or $\mathbf{b}$, then we may expect a perturbation $\Delta\mathbf{x}$ to $\mathbf{A}^{-1}\mathbf{b}$ that is (in norm) up to the order of $\mathbf{u}\,\|\mathbf{A}^{-1}\|_2\,\|\mathbf{b}\|_2$. This corresponds to an error $-\mathbf{A}\Delta\mathbf{x}$ in the residual, for which the norm is up to the order of $\mathbf{u}\,\kappa_2(\mathbf{A})\|\mathbf{b}\|_2$. In this sense the stability of GMRES* is optimal.

Our analysis for GMRES* has been restricted to certain parts of the algorithm. For an analysis of all errors in the original GMRES, including those in the Arnoldi process and the Givens rotations, for unsymmetric $\mathbf{A}$, see [3].

2.3. Error analysis for MINRES. For MINRES we have to study the errors in the evaluation in finite precision of $\left(\mathbf{V}_k R_k^{-1}\right) z_k$.

We will first analyze the floating point errors introduced by the computation of the columns of $\mathbf{W}_k = \mathbf{V}_k R_k^{-1}$. The jth row $w_{j,:}$ of $\mathbf{W}_k$ satisfies

$$w_{j,:} R_k = v_{j,:},$$

which means that in floating point finite precision arithmetic we obtain the solution $\widehat{w}_{j,:}$ of a perturbed system:

$$\widehat{w}_{j,:}(R_k + \Delta_{R_j}) = v_{j,:}, \tag{18}$$

with

$$|\Delta_{R_j}| \leq 3\,\mathbf{u}\,|R_k| + \mathcal{O}(\mathbf{u}^2). \tag{19}$$

Note that the perturbation term Δ_{R_j} depends on j. This gives $\widehat{w}_{j,:} R_k = v_{j,:} - \widehat{w}_{j,:}\Delta_{R_j}$, and when we combine the relations for $j = 1, \ldots, k$, we obtain

$$\widehat{\mathbf{W}}_k = (\mathbf{V}_k + \Delta_W) R_k^{-1}, \tag{20}$$

with

$$|\Delta_W| \leq 3\,\mathbf{u}\,\left|\widehat{\mathbf{W}}_k\right|\,|R_k| + \mathcal{O}(\mathbf{u}^2). \tag{21}$$

We may replace $\widehat{\mathbf{W}}_k$ by $\mathbf{W}_k = \mathbf{V}_k R_k^{-1}$ in (21), because this leads only to $\mathcal{O}(\mathbf{u}^2)$ errors.

We also expect errors in the evaluation of $\widehat{\mathbf{x}_k} = fl((\mathbf{V}_k R_k^{-1}) z_k)$ because of finite precision errors in the multiplication of $\widehat{\mathbf{W}}_k$ with z_k:

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{W}}_k z_k + \Delta_3, \quad \text{with} \quad |\Delta_3| \leq k\,\mathbf{u}\,|\mathbf{W}_k|\,|z_k| + \mathcal{O}(\mathbf{u}^2). \tag{22}$$

The errors in $\widehat{\mathbf{W}}_k$ and the error term Δ_3 describe the errors that are due to the evaluation of the generating formula for MINRES. Added together, they lead to $\Delta\mathbf{x}_k \equiv \widehat{\mathbf{x}}_k - \mathbf{x}_k$ (with $\mathbf{x}_k = \mathbf{V}_k R_k^{-1} z_k$) related to MINRES

$$\Delta\mathbf{x}_k = \Delta_W R_k^{-1} z_k + \Delta_3, \tag{23}$$

and this leads to the following contribution to the MINRES residual:

$$\Delta\mathbf{r}_k \equiv \widehat{\mathbf{r}}_k - \mathbf{r}_k = -\mathbf{A}\Delta\mathbf{x}_k = -\mathbf{A}\Delta_W R_k^{-1} z_k - \mathbf{A}\Delta_3. \tag{24}$$

If we use the bound (21) for Δ_W, and use for other quantities bounds similar to those for GMRES, then we obtain (again, ignoring $\mathcal{O}(\mathbf{u}^2)$ terms)

$$\begin{aligned}
\|\Delta\mathbf{r}_k\|_2 &\leq 3\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|\,|\mathbf{V}_k R_k^{-1}|\,\|_2\,\|\,|R_k|\,\|_2\,\|R_k^{-1} z_k\|_2 + k\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|\,|\mathbf{V}_k R_k^{-1}|\,\|_2\,\|z_k\|_2 \\
&\leq 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|\mathbf{V}_k\|_F\,\|R_k^{-1}\|_2\,\|R_k\|_2\,\|R_k^{-1}\|_2\,\|\mathbf{b}\|_2 \\
&\qquad\qquad + k\,\mathbf{u}\,\|\mathbf{A}\|_2\,\|\mathbf{V}_k\|_F\,\|R_k^{-1}\|_2\,\|\mathbf{b}\|_2 \\
&\leq 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_k\|_F\,\kappa_2(\mathbf{A})^2\,\|\mathbf{b}\|_2 + k\,\mathbf{u}\,\kappa_2(\mathbf{A})\,\|\mathbf{V}_k\|_F\,\|\mathbf{b}\|_2.
\end{aligned}$$

Here we have also used the fact that

$$\|\,|\mathbf{V}_k R_k^{-1}|\,\|_2 \leq \|\mathbf{V}_k R_k^{-1}\|_F \leq \|\mathbf{V}_k\|_F\,\|R_k^{-1}\|_2, \tag{25}$$

and, with $\|\mathbf{V}_k\|_F \leq \sqrt{k}$, the expression can be further bounded.

This results in the following upper bound for the error contribution in the residual for MINRES, due to the computational errors in the generating formula (10):

$$\frac{\|\Delta\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2} \leq 3\sqrt{3k}\,\mathbf{u}\,\kappa_2(\mathbf{A})^2 + k\sqrt{k}\,\mathbf{u}\,\kappa_2(\mathbf{A}). \tag{26}$$

We see that the generating formula for MINRES leads to an upper bound for the norm of the relative error in the residual that is proportional to the squared condition number of $\mathbf{A}$, whereas for GMRES* this led to an upper bound for the relative error in norm proportional to the condition number only; see (17). This means that if we plot the norms of the residuals for MINRES and GMRES*, then the upper bounds suggest that we may expect to see differences.

More specifically, they suggest that the difference between the norms of the computed residuals for the two methods may be expected to be up to the order of the square of the condition number. As soon as the norm of the computed residual of GMRES* (involving all errors made in the process) gets below $\mathbf{u}\,\kappa_2(\mathbf{A})^2\,\|\mathbf{b}\|_2$, then this difference may be visible. Indeed, our experiments display a clear difference between the residual norms for MINRES and GMRES*, in the order of our upper bounds.

2.4. Discussion. In Figure 4, we have plotted the residuals obtained for GMRES* and MINRES. Our analysis suggests that there may be a difference between both up to the order of the square of the condition number times machine precision relative

to $\|\mathbf{b}\|_2$. Of course, the computed residuals reflect all errors made in both processes, and if all these errors together lead to perturbations in the same order for MINRES and GMRES*, then we will not see much difference in the norms of the residuals. However, as we see, all the errors in GMRES* lead to something proportional to the condition number, and now the effect of the square of the condition number is clearly visible in the error in the residual for MINRES.

Our analysis implies that one has to be careful with MINRES when solving linear systems with an ill-conditioned matrix $\mathbf{A}$, especially when eigenvector components in the solution, corresponding to small eigenvalues, are important.

The residual norm reduction $\|\mathbf{r}_k\|_2/\|\mathbf{b}\|_2$ for the exact (but unknown) MINRES residual can be expressed as the product $\rho_k \equiv |s_1 \cdot \ldots \cdot s_k|$ of the sines s_k of the Givens rotations; see [13, Proposition 1]. (See also (57) and its subsequent discussion). This is the last ($(k+1)$th) coordinate of the vector that is obtained by applying the k Givens rotations (used for the annihilation of the subdiagonal elements of $\underline{T}_k$) to the vector e_1 (of length $k+1$). In GMRES the computed value $\widehat{\rho}_k$, computed with the $\widehat{s}_k$, is often used for monitoring the reduction of the residual norm. In practical computations, a residual norm is not often computed explicitly at each iteration step as $\|\mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k\|_2$, with $\widehat{\mathbf{x}}_k$ the kth floating point approximate solution, because this would require an extra matrix-vector product.

In Figure 4, we have also plotted the computed residual reduction factors $\widehat{\rho}_k$ for MINRES and GMRES*, as dotted curves. We see that the $\widehat{\rho}_k$ are only close to the actual residual reductions (the drawn curves) until where these stagnate: for MINRES this happens at a level proportional to $\kappa_2(\mathbf{A})^2\mathbf{u}$, and for GMRES* this happens at a level proportional to $\kappa_2(\mathbf{A})\mathbf{u}$.

We do not know whether the $\widehat{\rho}_k$ are always close to the actual residual reduction factors before the latter ones stagnate because of errors due to the evaluation of the generating formulas; this might be not the case if there is a severe loss of orthogonality among the columns of $\mathbf{V}_k$ in an earlier phase of the iteration history.

We have not considered the question of how close to orthogonal $\mathbf{V}_{k+1}$ should be, but we have seen that the generating formula (10) for MINRES may lead to errors that are in norm proportional to $\kappa_2(\mathbf{A})^2\mathbf{u}$. Because the $\widehat{\rho}_k$ cannot reflect computational errors in the solution of the reduced system (in fact, the derivation of the ρ_k assumes exact solution of the reduced system), we should expect at least a deviation by that order of magnitude in $\widehat{\rho}_k$ with respect to $\|\mathbf{A}\widehat{\mathbf{x}}_k - \mathbf{b}\|_2/\|\mathbf{b}\|_2$. This suggests that the computed reduction factor may be very unreliable for ill-conditioned matrices $\mathbf{A}$.

The situation for GMRES* is much better: the errors introduced by the evaluation of the generating formula (9) have the same order of magnitude as the errors that we should expect from a small relative perturbation (of order $\mathcal{O}(\mathbf{u})$) of the given system.

2.5. Diagonal matrices. Numerical analysts often carry out experiments for (unpreconditioned) iterative solvers for symmetric systems with diagonal matrices, because, at least in exact arithmetic, the convergence behavior depends on the distribution of the eigenvalues and the structure of the matrix plays no role in Krylov solvers. However, the behavior of these methods for diagonal systems may be quite different in finite precision, as we will now show, and, in particular for MINRES, experiments with diagonal matrices may give a too optimistic view on the behavior of the method.

Rotating the matrix from diagonal to nondiagonal (i.e., $\mathbf{A} = \mathbf{Q}^T\mathbf{D}\mathbf{Q}$, with $\mathbf{D}$ diagonal and $\mathbf{Q}$ orthogonal, instead of $\mathbf{A} = \mathbf{D}$) has hardly any influence on the errors in the GMRES* residuals (no results shown here). This is not the case for MINRES:

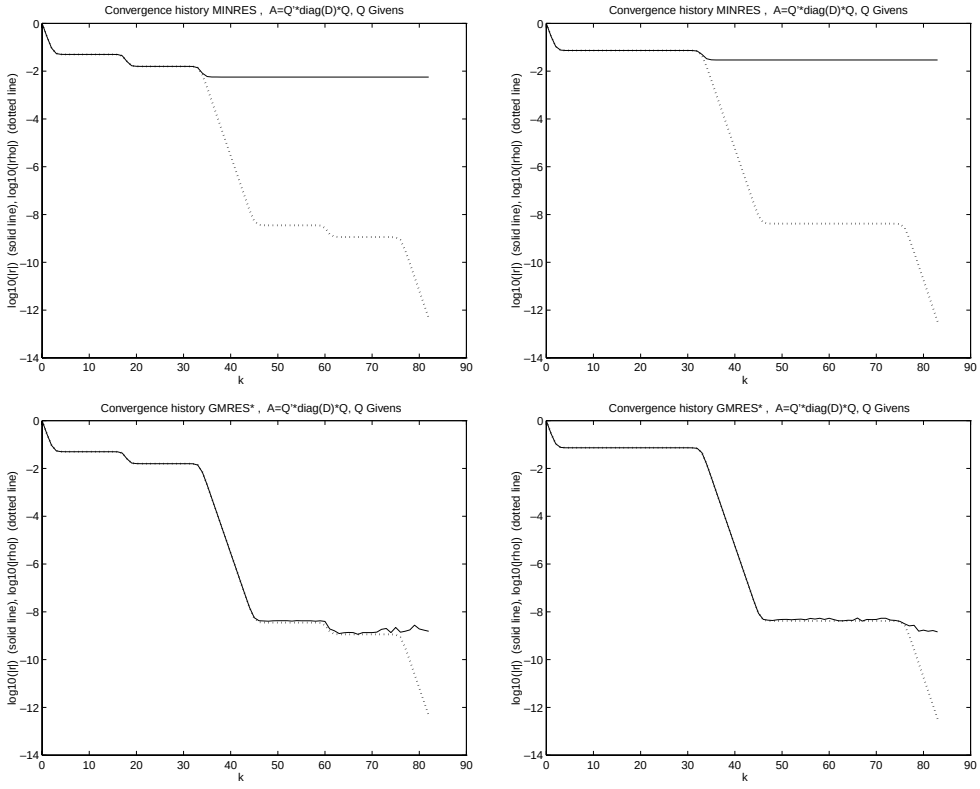


FIG. 4. MINRES (*top*) *and* GMRES* (*bottom*): *solid line* (—) $\log_{10}$ *of* $\|\mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k\|_2/\|\mathbf{b}\|_2$; *dotted line* $(\cdots)$ $\log_{10}$ *of the estimated residual norm reduction* ρ_k. *The pictures show the results for a positive definite system* (*the left pictures*) *and for an indefinite system* (*the right pictures*). *For both examples* $\kappa_2(\mathbf{A}) = 3 \cdot 10^8$. *To be more specific, at the left* $\mathbf{A} = \mathbf{GDG}'$ *with* $\mathbf{D}$ *diagonal,* $\mathbf{D} \equiv \mathrm{diag}(10^{-8}, 2\cdot 10^{-8}, 2 : h : 3)$, $h = 1/789$, *and* $\mathbf{G}$ *the Givens rotation in the* $(1, 30)$*-plane over an angle of* $45°$*; at the right* $\mathbf{A} = \mathbf{GDG}'$ *with* $\mathbf{D}$ *diagonal* $\mathbf{D} \equiv \mathrm{diag}(-10^{-8}, 10^{-8}, 2 : h : 3)$, $h = 1/389$, *and* $\mathbf{G}$ *the same Givens rotation as for the left example; in both examples* (*and others to come*) $\mathbf{b}$ *is the vector with all coordinates equal to 1,* $\mathbf{x}_0 = \mathbf{0}$, *and the relative machine precision* $\mathbf{u} = 1.1 \cdot 10^{-16}$.

experimental results (cf. Figure 5) indicate that the errors in the MINRES residuals for diagonal matrices are of order $\mathbf{u}\,\kappa_2(\mathbf{A})$, similar to GMRES*. This can be understood as follows.

If we neglect $\mathcal{O}(\mathbf{u}^2)$ terms, then, according to (18), the error, due to the inversion of R_k, in the jth coordinate of the MINRES-$\mathbf{x}_k$, due to the evaluation of the generating formula, is given by

$$(\Delta\mathbf{x}_k)_j = (\widehat{w}_{j,:} - w_{j,:})z_k + (\Delta_3)_j = -v_{j,:}\,R_k^{-1}\Delta_{R_j}R_k^{-1}z_k + (\Delta_3)_j,$$

where $(\Delta_3)_j$ is the jth coordinate of Δ_3 (see (22)).

When $\mathbf{A}$ is diagonal with (j,j)-entry λ_j, the error in the jth coordinate of the MINRES residual is equal to (use (1) and (5))

$$\begin{aligned}(\Delta\mathbf{r}_k)_j &= \lambda_j v_{j,:}\,R_k^{-1}\Delta_{R_j}R_k^{-1}z_k - \lambda_j(\Delta_3)_j \\ &= \mathbf{e}_j^T\mathbf{A}\mathbf{V}_k R_k^{-1}\Delta_{R_j}R_k^{-1}z_k - \lambda_j(\Delta_3)_j \\ &= \mathbf{e}_j^T\mathbf{V}_{k+1}\underline{Q}_k\Delta_{R_j}R_k^{-1}z_k - \lambda_j(\Delta_3)_j.\end{aligned} \tag{27}$$

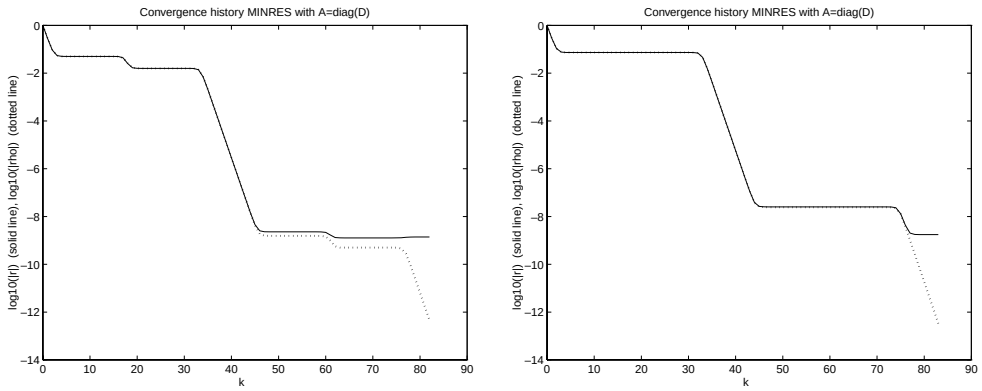


FIG. 5. MINRES: *solid line* (—) $\log_{10}$ *of* $\|\mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k\|_2/\|\mathbf{b}\|_2$; *dotted line* $(\cdots)$ $\log_{10}$ *of the estimated residual norm reduction* $\widehat{\rho}_k$. *The pictures show the results for a positive definite diagonal system* (*the left picture*) *and for an indefinite diagonal system* (*the right picture*). *Except for the Givens rotation, the matrices in these examples are equal to the matrices of the examples in Figure* 4: *here* $\mathbf{G} = \mathbf{I}$.

Therefore, in view of (19), and including the error term for the multiplication with $\widehat{\mathbf{W}}_k$ (cf. (22)), we have for MINRES applied to a diagonal matrix

$$\frac{\|\Delta\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2} \leq (3\sqrt{3}\|\mathbf{V}_{k+1}\|_2 + k\sqrt{k})\,\mathbf{u}\,\kappa_2(\mathbf{A}),$$

which is the same upper bound as for the errors in the GMRES* residuals in (17).

The perturbation matrix Δ_{R_j} depends on the row index j. Since, in general, Δ_{R_j} will be different for each coordinate j, (27) cannot be expected to be correct for non-diagonal matrices. In fact, if $\mathbf{A} = \mathbf{Q}^T\mathrm{diag}(\lambda_j)\mathbf{Q}$, with $\mathbf{Q}$ some orthogonal matrix, then errors of order $\mathbf{u}\,\|R_k^{-1}\|_2\,\kappa_2(R_k)$ in the jth coordinate of $\mathbf{x}_k$ can be transferred by $\mathbf{Q}$ to an mth coordinate and may not be damped by a small value $|\lambda_m|$. More precisely, if Γ is the maximum size of the off-diagonal elements of $\mathbf{A}$ that "couple" small diagonal elements of $\mathbf{A}$ to large ones, then the error in the MINRES residual will be of order $\Gamma\,\mathbf{u}\,\|R_k^{-1}\|_2\,\kappa_2(R_k^{-1}) \leq \Gamma\,\mathbf{u}\,\|\mathbf{A}^{-1}\|_2\,\kappa_2(\mathbf{A})$. If $\Gamma \approx \|\mathbf{A}\|_2$, we recover essentially the bound (26).

2.6. The errors in the approximations. In exact arithmetic we have that $\|\mathbf{x}_k\|_2 = \|\mathbf{V}_k R_k^{-1} z_k\|_2 = \|R_k^{-1} z_k\|_2$. We will in this section assume that, in finite precision, this also gives approximately the right order of magnitude for representations of the solution

$$\|\widehat{\mathbf{x}}_k\|_2 \approx \|\mathbf{x}_k\|_2 = \|y_k\|_2.$$

Then the errors (14) and (23), related to the evaluation of the generating formulas (9) and (10), respectively, can be bounded by essentially the same upper bound:

$$\frac{\|\Delta\mathbf{x}_k\|_2}{\|\widehat{\mathbf{x}}_k\|_2} \lesssim (3\sqrt{3} + k\sqrt{k})\mathbf{u}\,\|\mathbf{V}_k\|_2\,\kappa_2(R_k) \leq (3\sqrt{3k} + k\sqrt{k})\mathbf{u}\,\kappa_2(\mathbf{A}). \tag{28}$$

This may come as a surprise since the bound for the error contribution to the residual for MINRES is proportional to $\kappa_2(\mathbf{A})^2$.

Based upon our observations in numerical experiments, we think that this can be explained as follows. The error in the GMRES* approximation has its relatively largest components mainly in the direction of the 'small' eigenvectors of $\mathbf{A}$. These components are relatively reduced by the multiplication with $\mathbf{A}$, and then have less effect to the norm of the residual.

On the other hand, the errors in the MINRES approximation are more or less of the same magnitude over the spectrum of eigenvalues of $\mathbf{A}$. Multiplication with $\mathbf{A}$ will make error components associated with larger eigenvalues more effective in the residual.

We will support our viewpoint by a numerical example. The results in Figure 6 are obtained with a positive definite matrix with two tiny eigenvalues. For $\mathbf{b}$ we took a random perturbation of $\mathbf{Ay}$ in the order of 0.01: $\mathbf{b} = \mathbf{Ay} + \mathbf{p}$, $\|\mathbf{p}\|_2 \leq 10^{-2}$. This example mimics the situation where the right-hand-side vector is affected by errors from measurements. The solution $\mathbf{x}$ of the equation $\mathbf{Ax} = \mathbf{b}$ has huge components in the direction of the two eigenvectors with smallest eigenvalue. In the other directions $\mathbf{x}$ is equal to $\mathbf{y}$ plus a perturbation of less than one percent. The coordinates of the vector $\mathbf{y}$ in our example form a parabola, which makes the effects more easily visible.

The convergence histories of GMRES* and of MINRES (not shown here) for this example with $\mathbf{x}_0 = \mathbf{0}$ are comparable to the ones in the left pictures of Figure 4, but, because of a higher condition number, the final stagnation of the residual norm in the present example takes place on a higher level ($\approx 3 \cdot 10^{-8}$ for GMRES* and $\approx 10^0$ for MINRES).

Figure 6 shows the solution $\mathbf{x}_k$ as computed at the 80th step of GMRES (top pictures) and of MINRES (bottom pictures); the right pictures show the component of $\mathbf{x}_k$ orthogonal to the two eigenvectors with smallest eigenvalue, while the left pictures show the complete $\mathbf{x}_k$. Note that $\|\mathbf{x}_k\|_2 \approx 10^7$. The curve of the projected GMRES* solution (top right picture) is a slightly perturbed parabola indeed (the irregularities are due to the perturbation $\mathbf{p}$). The computational errors from the GMRES* process are not visible in this picture: these errors are mainly in the direction of the two 'small' eigenvectors.

In contrast, the irregularities in the MINRES curve (bottom right picture) are almost exclusively the effect of rounding errors in the MINRES process.

3. Error analysis for SYMMLQ. In SYMMLQ we minimize the norm of $\mathbf{x} - \mathbf{x}_k$, for $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{A}\mathbf{V}_k y_k$, which means that y_k is the solution of the normal equations

$$\mathbf{V}_k{}^T \mathbf{A}^T \mathbf{A} \mathbf{V}_k y_k = \mathbf{V}_k{}^T \mathbf{A}^T (\mathbf{x} - \mathbf{x}_0) = \mathbf{V}_k{}^T \mathbf{r}_0 = \|\mathbf{r}_0\|_2 \, e_1.$$

This system can be further simplified by exploiting the Lanczos relations (1):

$$\mathbf{V}_k{}^T \mathbf{A}^T \mathbf{A} \mathbf{V}_k = \underline{T}_k^T \mathbf{V}_{k+1}^T \mathbf{V}_{k+1} \underline{T}_k = \underline{T}_k^T \underline{T}_k.$$

A stable way of solving this set of normal equations is based on an LQ decomposition of $\underline{T}_k^T$, and this is equivalent to the transpose of the QR decomposition of $\underline{T}_k$ (see (5)), which is constructed for GMRES* and MINRES:

$$\underline{T}_k^T = R_k^T \underline{Q}_k^T.$$

This leads to

$$\underline{T}_k^T \underline{T}_k y_k = R_k^T R_k y_k = \|\mathbf{r}_0\|_2 \, e_1,$$

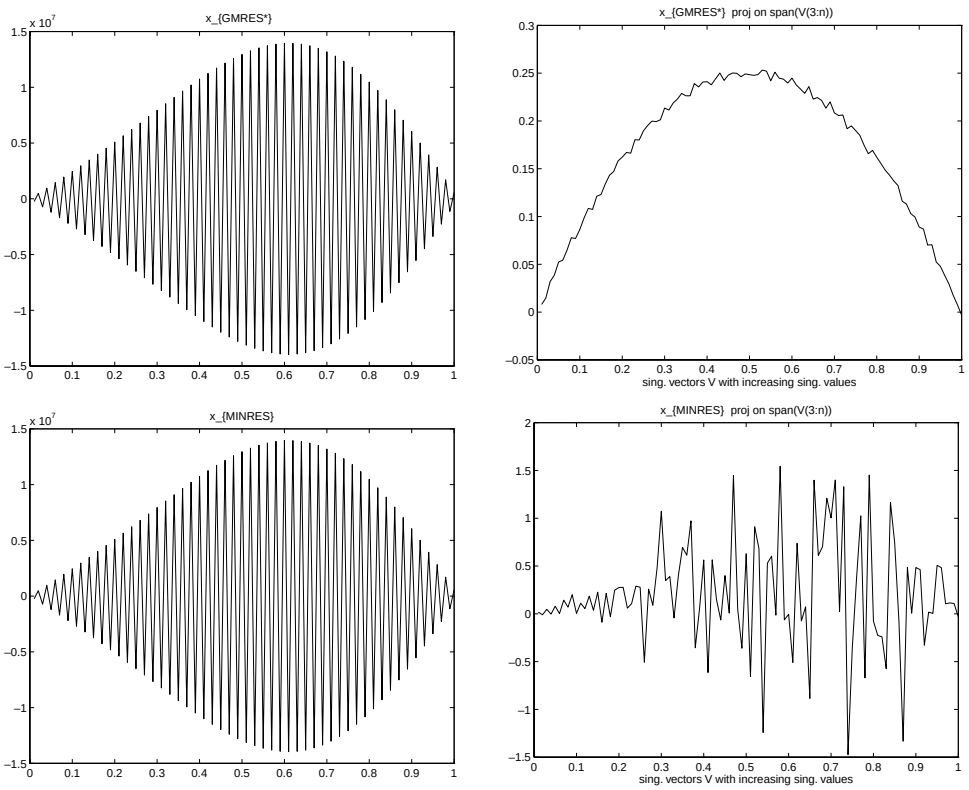


FIG. 6. *The pictures show the solution* $\mathbf{x}$ *of* $\mathbf{Ax} = \mathbf{b}$*, computed with* 80 *steps of* GMRES* (*top pictures*) *and of* MINRES (*bottom pictures*)*. The* i*th coordinate of* $\mathbf{x}_k$ (*along the vertical axis*) *is plotted against* $\frac{i}{n}$ (*along the horizontal axis*)*.* $\mathbf{A} = \mathbf{Q}^*\mathbf{DQ}$ *with* $\mathbf{D} = \mathrm{diag}(10^{-10}, 2 \cdot 10^{-10}, 2 : h : 3)$*,* $h = 1/97$ *and* $\mathbf{Q}$ *unitary,* $\mathbf{Q}_{ij} = \sqrt{\frac{2}{n+1}} \sin \frac{i(n+1-j)}{(n+1)\pi}$*,* $n = 100$*.* $\mathbf{b} = \mathbf{Ay} + \mathbf{p}$ *with* $y_i = \frac{i}{n}(1 - \frac{i}{n})$*, and* $\mathbf{p}$ *random,* $\|\mathbf{p}\|_2 \leq 0.01$*. The right pictures show the component of* $\mathbf{x}_k$ *orthogonal to the two eigenvectors with smallest eigenvalue, while the left pictures show the complete* $\mathbf{x}_k$*.*

from which the basic generating formula for SYMMLQ is obtained:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{x}_0 + \mathbf{A}\mathbf{V}_k R_k^{-1} R_k^{-T} \|\mathbf{r}_0\|_2 \, e_1 \\ &= \mathbf{x}_0 + \mathbf{V}_{k+1} \underline{T}_k R_k^{-1} R_k^{-T} \|\mathbf{r}_0\|_2 \, e_1 \\ &= \mathbf{x}_0 + (\mathbf{V}_{k+1} \underline{Q}_k) \, (L_k^{-1} \|\mathbf{r}_0\|_2 \, e_1), \end{aligned} \tag{29}$$

with $L_k \equiv R_k^T$. We will further assume that $\mathbf{x}_0 = \mathbf{0}$ and hence $\mathbf{r}_0 = \mathbf{b}$. This gives the following generating formula:

$$\mathbf{x}_k = (\mathbf{V}_{k+1} \underline{Q}_k) \, (L_k^{-1} \|\mathbf{b}\|_2 \, e_1). \tag{30}$$

The actual implementation of SYMMLQ [11] is based on an update procedure for $\mathbf{V}_{k+1}\underline{Q}_k$, and on a three-term recurrence relation for $g_k \equiv \|\mathbf{b}\|_2 \, L_k^{-1} e_1$.

The differences in finite precision between MINRES and GMRES* could be analyzed by studying the differences in the evaluation of the generating formula for these methods (see (6)):

$$\mathbf{x}_k = \mathbf{V}_k R_k^{-1} \underline{Q}_k^T \|\mathbf{b}\|_2 e_1. \tag{31}$$

Note that, because of $L_k = R_k^T$, the generating formulas for the three methods contain in principle the same computed ingredients $\mathbf{V}_{k+1}$, $\underline{Q}_k$, R_k, and $\mathbf{b}$. In fact, we see no good reason for using differently computed values for each of the algorithms.

The methods MINRES and GMRES* have been characterized by a different order of evaluation of essentially the same generating formula (see (9) and (10)). For SYMMLQ we have a completely different generating formula which even in exact arithmetic leads to completely different results. Observed differences in the results for SYMMLQ, compared to MINRES and GMRES*, can by no means be attributed to computational errors. However, we have tried to make plausible that eventually the norm of the residual for MINRES may be contaminated by a term proportional to $\|\mathbf{b}\|_2\kappa_2(\mathbf{A})^2\mathbf{u}$, which may lead to a stagnation of the residual norm at a significantly higher level than for GMRES*; see, for instance, Figure 4. Since SYMMLQ may be considered as an alternative for MINRES (one reason is that it avoids storage of the full $\mathbf{V}_{k+1}$), it may be of interest to see whether computational errors in the generating formula may have a similar polluting effect on the residual as for MINRES. Note that even if we can answer this question, then this does not reveal all differences due to rounding errors in MINRES and SYMMLQ. One reason could be that rounding errors in $\mathbf{V}_k$ manifest themselves differently (because of the right multiplication with $\underline{Q}_k$), although this does not seem very likely to us because of the (near) orthogonality of $\underline{Q}_k$.

We postulate that the main factor, for ill-conditioned systems, in the upper bound for the norm of the additional rounding errors in the residual for SYMMLQ, due to the evaluation of the generating formula, comes from solving $L_k g_k = \|\mathbf{b}\|_2 e_1$ for g_k. In order to simplify our rather complicated analysis for SYMMLQ, we have chosen to study only the effect of the errors introduced by this part of the formula.

The resulting error $\Delta\mathbf{x}_k$ is written as

$$\Delta\mathbf{x}_k = \mathbf{V}_{k+1}\,\underline{Q}_k(\widehat{g}_k - g_k) \quad \text{with} \quad L_k g_k = \|\mathbf{b}\|_2\, e_1, \tag{32}$$

where g_k represents the exact solution and $\widehat{g}_k$ is the value obtained in finite precision arithmetic. We write $g_k/\|\mathbf{b}\|_2 = (\gamma_1, \ldots, \gamma_k)^T$, and likewise the coordinates of $\widehat{g}_k/\|\mathbf{b}\|_2$ are denoted by $\widehat{\gamma}_j$. These coordinates can be written as

$$\gamma_k = e_k^T L_k^{-1} e_1, \quad \widehat{\gamma}_k = e_k^T (L_k + \Delta_L)^{-1} e_1, \quad \text{with} \quad |\Delta_L| \le 3\,\mathbf{u}\,|L_k| + \mathcal{O}(\mathbf{u}^2). \tag{33}$$

In order to simplify our formulas, we will omit the $\mathcal{O}(\mathbf{u}^2)$ terms in the further analysis.

For the analysis of the residual, we will be interested in the term $\mathbf{A}\mathbf{V}_{k+1}\,\underline{Q}_k$. Using the relation for the finite precision Lanczos process, we have (cf. (2))

$$\mathbf{A}\mathbf{V}_{k+1}\,\underline{Q}_k = \mathbf{V}_{k+2}\,\underline{T}_{k+1}\,\underline{Q}_k + \mathbf{F}_{k+1}\,\underline{Q}_k.$$

Since T_{k+3} is symmetric, we have for its submatrices that

$$\underline{T}_{k+1} = \underline{T}_{k+2}^T\,\underline{\underline{I}}_{k+1},$$

where $\underline{\underline{I}}_{k+1}$ is the $k+3$ by $k+1$ left block of the $k+3$-dimensional identity matrix. Moreover, for the LQ decomposition in finite precision, we have (cf. (11))

$$\underline{T}_{k+2}^T = L_{k+2}\,\underline{Q}_{k+2}^T + \underline{G}_{k+2}^T.$$

The matrix $\underline{Q}_{k+2}$ is upper Hessenberg. Hence, $\underline{\underline{I}}_{k+1}\,\underline{Q}_k$ consists of the first k columns of $\underline{Q}_{k+2}$ and orthogonality of $\underline{Q}_{k+2}$ implies that

$$\underline{Q}_{k+2}^T\,\underline{\underline{I}}_{k+1}\,\underline{Q}_k = \underline{\underline{I}}_k.$$

Hence, taking into account that $L_{k+2} = (\ell_{i,j})$ is lower tridiagonal ($\ell_{i,j} \neq 0$ only if $i \leq j \leq i+2$),

$$\begin{aligned}\mathbf{A}\mathbf{V}_{k+1}\underline{Q}_k &= \mathbf{V}_{k+2}\underline{T}_{k+1}\underline{Q}_k + \mathbf{F}_{k+1}\underline{Q}_k \\ &= \mathbf{V}_{k+2}\,L_{k+2}\underline{I}_k + \mathbf{V}_{k+2}\underline{G}_{k+2}^T\underline{I}_{k+1}\underline{Q}_k + \mathbf{F}_{k+1}\underline{Q}_k \\ &= \mathbf{V}_k L_k + [\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} + \mathbf{F}'_{k+1}, \end{aligned} \tag{34}$$

where M_k is the right 2 by 2 lower block of $L_{k+2}\underline{I}_k$,

$$M_k \equiv \begin{bmatrix} \ell_{k+1,k-1} & \ell_{k+1,k} \\ 0 & \ell_{k+2,k} \end{bmatrix},$$

and

$$\mathbf{F}'_{k+1} \equiv \mathbf{V}_{k+2}\underline{G}_{k+2}^T\underline{I}_{k+1}\underline{Q}_k + \mathbf{F}_{k+1}\underline{Q}_k.$$

Note that, on account of (3) and (11),

$$\|\mathbf{F}'_{k+1}\|_2 \leq c'\,k^2\sqrt{k}\,\mathbf{u}\,\|\mathbf{A}\|_2 \tag{35}$$

for some modest constant c'.

We will use that $(L_k + \Delta_L)^{-1} = L_k^{-1} - L_k^{-1}\Delta_L L_k^{-1}$ (neglecting $\mathcal{O}(\mathbf{u}^2)$ terms; cf. (33)). Then, from (34), we find for the residual $\widehat{\mathbf{r}}_k$ corresponding to the computed approximation $\widehat{\mathbf{x}}_k = \mathbf{x}_k + \Delta\mathbf{x}_k$ (see (32)),

$$\begin{aligned}\widehat{\mathbf{r}}_k \equiv \mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k &= \mathbf{b} - \mathbf{A}\mathbf{V}_{k+1}\underline{Q}_k\,(L_k + \Delta_L)^{-1}\|\mathbf{b}\|_2\,e_1 \\ &= \mathbf{b} - \mathbf{V}_k L_k L_k^{-1}\|\mathbf{b}\|_2 e_1 + \mathbf{V}_k L_k L_k^{-1}\Delta_L L_K^{-1}\|\mathbf{b}\|_2 e_1 \\ &\quad - \left([\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} + \mathbf{F}'_{k+1}\right)(L_k + \Delta_L)^{-1}\|\mathbf{b}\|_2\,e_1 \\ &= \mathbf{V}_k\Delta_L\|\mathbf{b}\|_2 L_k^{-1}e_1 - \|\mathbf{b}\|_2\,[\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]\,\widehat{t}_k - \mathbf{F}'_{k+1}\,(L_k + \Delta_L)^{-1}\|\mathbf{b}\|_2\,e_1, \end{aligned} \tag{36}$$

where

$$\widehat{t}_k \equiv M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix}(L_k + \Delta_L)^{-1}e_1. \tag{37}$$

For the process where the system $L_k g_k = \|\mathbf{b}\|_2 e_1$ is solved exactly ($\Delta_L = 0$), we have

$$\mathbf{r}_k \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_k = -\|\mathbf{b}\|_2\,[\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]\,t_k - \mathbf{F}'_{k+1}\,L_k^{-1}\|\mathbf{b}\|_2\,e_1, \tag{38}$$

where

$$t_k \equiv M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} L_k^{-1}e_1.$$

Neglecting order $\mathbf{u}^2$ terms (e.g., stemming from $\mathbf{F}'_{k+1}\Delta_L$), we conclude that the error in the SYMMLQ residual $\mathbf{r}_k$, due to the solution of $L_k g_k = \|\mathbf{b}\|_2\,e_1$ in finite precision, can be written as

$$\Delta\mathbf{r}_k \equiv \widehat{\mathbf{r}}_k - \mathbf{r}_k = \|\mathbf{b}\|_2\mathbf{V}_k\Delta_L L_k^{-1}e_1 - \|\mathbf{b}\|_2\,[\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]\,(\widehat{t}_k - t_k). \tag{39}$$

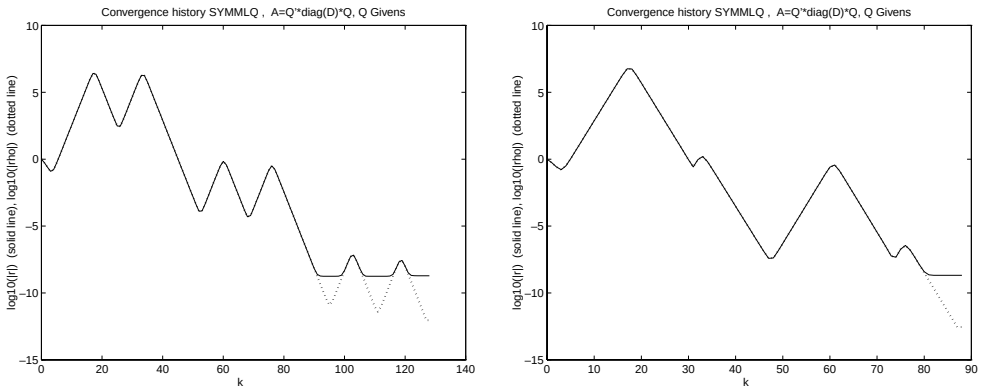


FIG. 7. SYMMLQ: *solid line* (—) $\log_{10}$ *of* $\|\mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_k\|_2/\|\mathbf{b}\|_2$; *dotted line* $(\cdots)$ $\log_{10}$ *of the estimated residual norm reduction* $\|\widehat{t}_k\|_2$. *The pictures show the results for the positive definite system* (*the left picture*) *and for the indefinite system* (*the right picture*) *of Figure* 4. *Both systems have condition number* $3 \cdot 10^8$.

To obtain a bound for norm of this error, note that (see (16))

$$\begin{aligned} \|\mathbf{V}_k \Delta_L L_k^{-1} e_1\|_2 &\leq 3\,\mathbf{u}\,\|\mathbf{V}_k\|_2\,\|\,|L_k|\,\|_2\,\|L_k\|_2 \leq 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_k\|_2\,\kappa_2(L_k) \\ &= 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_k\|_2\,\kappa_2(R_k) \leq 3\sqrt{3}\,\mathbf{u}\,\|\mathbf{V}_k\|_2\,\kappa_2(\mathbf{A}). \end{aligned} \tag{40}$$

Since $\mathbf{v}_{k+1}$ and $\mathbf{v}_{k+2}$ are orthonormal up to machine precision, this leads to

$$\frac{\|\Delta\mathbf{r}_k\|_2}{\|\mathbf{b}\|_2} \leq 3\sqrt{3}\,\|\mathbf{V}_k\|_2\,\mathbf{u}\,\kappa_2(\mathbf{A}) + (1 + c'\mathbf{u})\|\widehat{t}_k - t_k\|_2 \tag{41}$$

for some modest constant c'. A straightforward estimate is

$$\|\widehat{t}_k - t_k\|_2 = \left\| M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} L_k^{-1} \Delta_L L_k^{-1} e_1 \right\|_2 \leq 3\sqrt{3}\,\mathbf{u}\,\kappa_2(L_k)^2 \leq 3\sqrt{3}\,\mathbf{u}\,\kappa_2(\mathbf{A})^2, \tag{42}$$

which is much larger than the first term in (41). Experiments indicate that $\|\widehat{t}_k - t_k\|_2$ converges towards 0 (even below the value $\mathbf{u}\,\kappa_2(\mathbf{A})$). Below, we will explain why this is to be expected (cf. (60)). Figure 7 illustrates that the upper bound in (41), with $\|\widehat{t}_k - t_k\|_2 \approx 0$, is fairly sharp.

Accuracy. In exact arithmetic (where also $\mathbf{F}_{k+1} = \mathbf{0}$ and $\underline{G}_{k+2} = 0$), the norm $\|\mathbf{r}_k\|_2$ of the SYMMLQ residual is equal to $\|t_k\|_2$ (as can be seen from (38)). Therefore, the computed residual norm reduction $\|\widehat{t}_k\|_2$ is usually used for monitoring the convergence in a stopping criterion. In actual computations with SYMMLQ, no residual vectors are computed. To see how close $\|\widehat{t}_k\|_2$ is to the reduction $\|\widehat{\mathbf{r}}_k\|_2/\|\mathbf{b}\|_2$ of the norm of the actual residual, first note that rounding errors in the multiplication in (37) by M_k and in (36) by $[\mathbf{v}_{k+1}, \mathbf{v}_{k+2}]$ can be bounded by some modest multiple of $\mathbf{u}\,\kappa_2(L_k)$.[2] These bounds will be neglected in the estimates below: since $\kappa_2(L_k) \leq \kappa_2(\mathbf{A})$ (see (16)), they are much smaller than the bound on $\|\mathbf{F}'_{k+1} L_k^{-1} e_1\|_2$ arising from (35). The rounding errors in $\mathbf{v}_{k+1}$ and $\mathbf{v}_{k+2}$ have a similar effect: these vectors are orthonormal up to machine precision.

[2]Note the contrast in the effect of errors in the multiplication by M_k and in the solution of $L_k g_k = e_1$ (cf. (42)).

From (36), (35), and (40), neglecting relatively small terms, it follows that

$$\left| \|\widehat{t}_k\|_2 - \frac{\|\widehat{\mathbf{r}}_k\|_2}{\|\mathbf{b}\|_2} \right| \leq \|\mathbf{V}_k \Delta_L L_k^{-1} e_1\|_2 + \|\mathbf{F}'_{k+1} L_k^{-1} e_1\|_2 \leq c'\, k^{2\frac{1}{2}}\, \mathbf{u}\, \kappa_2(\mathbf{A}). \tag{43}$$

Apparently, SYMMLQ is rather accurate since, for any method, errors in the order $\mathbf{u}\,\kappa_2(\mathbf{A})$ should be expected anyway.

Convergence. It is not clear yet whether the convergence of SYMMLQ is insensitive to rounding errors in the assembly of $\mathbf{x}_k$ (cf. (31)). This would follow from (41) if both t_k and $\widehat{t}_k$ would approach 0. It is unlikely that $\|t_k\|_2$ will be (much) larger than $\|\widehat{t}_k\|_2$, that is, it is unlikely that the inexact process converges faster than the process in exact arithmetic. Therefore, when it is observed that $\|\widehat{t}_k\|_2$ is small (of order $\mathbf{u}\,\kappa_2(\mathbf{A})$), it may be concluded that the speed of convergence has not been affected seriously by rounding errors in the assembly of $\mathbf{x}_k$. In experiments, we see that $\widehat{t}_k$ approaches zero if k increases.

For practical applications, assuming that $\|t_k\|_2 \lesssim \|\widehat{t}_k\|_2$, it is useful to know that the computable value $\|\widehat{t}_k\|_2$ informs us on the accuracy of the computed approximate and on a possible loss of speed of convergence. However, it is of interest to know in advance whether the computed residual reduction will decrease to 0. Moreover, we would like to know whether $\|t_k\|_2 \lesssim \|\widehat{t}_k\|_2$. Of course, it is impossible to prove that SYMMLQ will converge for any symmetric problem: one can easily construct examples for which $\|\mathbf{r}_k\|_2$ will be of order 1 for any $k < n$. But, as we will analyze in the next subsection, the interesting quantities can be bounded in terms of the MINRES residual. That result will be used in order to show that the term $\|\widehat{t}_k - t_k\|_2$ will be relatively unimportant as soon as MINRES has converged to some degree.

3.1. A relation between SYMMLQ and MINRES residual norms. In this subsection we will assume exact arithmetic (in particular, the underlying Lanczos process is assumed to be exact, too). The residuals $\mathbf{r}_k^{\text{MR}}$ and $\mathbf{r}_k^{\text{ME}}$ denote the residuals of MINRES and SYMMLQ, respectively.

The norm of the residual $\mathbf{b} - \mathbf{A}\mathbf{x}^{\text{b}}$, with $\mathbf{x}^{\text{b}}$ the best approximate of $\mathbf{x}$ in $\mathcal{K}_k(\mathbf{A}; \mathbf{b})$, i.e., $\|\mathbf{x} - \mathbf{x}^{\text{b}}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{y} \in \mathcal{K}_k(\mathbf{A}; \mathbf{b})$, can be bounded in terms of the norm of the MINRES residual $\mathbf{r}_k^{\text{MR}}$:

$$\frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}^{\text{b}}\|_2}{\|\mathbf{r}_k^{\text{MR}}\|_2} \leq \kappa_2(\mathbf{A}). \tag{44}$$

This follows from the observation that $\mathbf{r}_k^{\text{MR}} = \mathbf{b} - \mathbf{A}\mathbf{x}_k^{\text{MR}}$, where $\mathbf{x}_k^{\text{MR}}$ is from the same subspace from which the best approximate $\mathbf{x}^{\text{b}}$ has been selected, and furthermore, that $\|\mathbf{b} - \mathbf{A}\mathbf{x}^{\text{b}}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x} - \mathbf{x}^{\text{b}}\|_2$ and $\|\mathbf{x} - \mathbf{x}_k^{\text{MR}}\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\mathbf{r}_k^{\text{MR}}\|_2$. Unfortunately, SYMMLQ selects its approximation $\mathbf{x}_k$ from a different subspace, namely $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{b})$. This makes a comparison less straightforward.

The following lemma will be used for bounding the SYMMLQ error in terms of the MINRES error. Its proof uses the fact that $\mathbf{r}_k^{\text{MR}}$ connects $\mathcal{K}_{k+1}(\mathbf{A}; \mathbf{b})$ and $\mathbf{A}\,\mathcal{K}_k(\mathbf{A}; \mathbf{b})$, that is, $\mathbf{r}_k^{\text{MR}} \in \mathcal{K}_{k+1}(\mathbf{A}; \mathbf{b})$, $\mathbf{r}_k^{\text{MR}} \perp \mathbf{A}\,\mathcal{K}_k(\mathbf{A}; \mathbf{b})$, and hence $\mathcal{K}_{k+1}(\mathbf{A}; \mathbf{b})$ is spanned by $\mathbf{r}_k^{\text{MR}}$ and $\mathbf{A}\,\mathcal{K}_k(\mathbf{A}; \mathbf{b})$.

LEMMA 3.1. *For each* $\mathbf{z} \in \mathcal{K}_{k+1}(\mathbf{A}; \mathbf{b})$, *we have*

$$\|\mathbf{x} - \mathbf{x}_k^{\text{ME}}\|_2^2 \leq \|\mathbf{x} - \mathbf{z}\|_2^2 + |\alpha_k|^2 \|\mathbf{r}_k^{\text{MR}}\|_2^2, \quad \textit{where} \quad \alpha_k \equiv \frac{(\mathbf{x}, \mathbf{r}_k^{\text{MR}})}{\|\mathbf{r}_k^{\text{MR}}\|_2^2}. \tag{45}$$

Proof. By construction $\mathbf{x}_k^{\mathrm{ME}}$ minimizes $\|\mathbf{x}-\mathbf{z}\|_2$ over all $\mathbf{z}$ in the space $\mathbf{A}\mathcal{K}_k(\mathbf{A};\mathbf{b})$. Hence $\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}} \perp \mathbf{A}\,\mathcal{K}_k(\mathbf{A};\mathbf{b})$. Since $\mathbf{r}_k^{\mathrm{MR}} \perp \mathbf{A}\mathcal{K}_k(\mathbf{A};\mathbf{b})$, it follows that $(\mathbf{x}_k^{\mathrm{ME}}, \mathbf{r}_k^{\mathrm{MR}}) = 0$, and therefore,

$$\alpha_k = (\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}, \mathbf{r}_k^{\mathrm{MR}})/\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2 \quad \text{and} \quad \mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}-\alpha_k\mathbf{r}_k^{\mathrm{MR}} \perp \mathbf{r}_k^{\mathrm{MR}}. \tag{46}$$

Since $\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}} \perp \mathbf{A}\mathcal{K}_k(\mathbf{A};\mathbf{b})$ and $\mathbf{r}_k^{\mathrm{MR}} \perp \mathbf{A}\mathcal{K}_k(\mathbf{A};\mathbf{b})$, (46) implies that

$$\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}-\alpha_k\mathbf{r}_k^{\mathrm{MR}} \perp \mathcal{K}_{k+1}(\mathbf{A};\mathbf{b}).$$

By construction we have that $\mathbf{x}_k^{\mathrm{ME}}-\alpha_k\mathbf{r}_k^{\mathrm{MR}} \in \mathcal{K}_{k+1}(\mathbf{A};\mathbf{b})$ and, as a consequence,

$$\|\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}-\alpha_k\mathbf{r}_k^{\mathrm{MR}}\|_2 \le \|\mathbf{x}-\mathbf{z}\|_2 \quad \text{for all} \quad \mathbf{z} \in \mathcal{K}_{k+1}(\mathbf{A};\mathbf{b}). \tag{47}$$

From Pythagoras's theorem, with (46), we conclude that

$$\|\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}\|_2^2 = \|\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}-\alpha_k\mathbf{r}_k^{\mathrm{MR}}\|_2^2 + |\alpha_k|^2\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2,$$

and (45) follows by combining this result with (47). ∎

Unfortunately, a combination of (45) with $\mathbf{z} = \mathbf{x}_k^{\mathrm{MR}}$ and the obvious estimate $|\alpha_k|\,\|\mathbf{r}_k^{\mathrm{MR}}\|_2 \le \|\mathbf{x}-\mathbf{x}_k^{\mathrm{ME}}\|_2$ from (46) does not lead to a useful result. An interesting result follows from an upper bound for $|\alpha_k|$ that can be obtained from a relation between two consecutive MINRES residuals and a Lanczos basis vector. This result is formulated in the next theorem.

THEOREM 3.2.

$$\|\mathbf{r}_k^{\mathrm{ME}}\|_2 \le \nu_{k+1}\,\kappa_2(\mathbf{A})\,\|\mathbf{r}_k^{\mathrm{MR}}\|_2 \quad \textit{with} \quad \nu_k \equiv k + \tfrac{1}{2}\ln(k). \tag{48}$$

Proof. We use the relation

$$\mathbf{r}_k^{\mathrm{MR}} = s^2\mathbf{r}_{k-1}^{\mathrm{MR}} + c^2\mathbf{r}_k^{\mathrm{CG}}, \tag{49}$$

where

$$s \equiv \frac{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2}, \tag{50}$$

and $\mathbf{r}_k^{\mathrm{CG}}$ is the kth conjugate gradient residual. The scalars s and c represent the Givens transformation used in the kth step of MINRES. This relation is a special case of the slightly more general relation between GMRES and FOM residuals, formulated in [1, 16]. For symmetric $\mathbf{A}$, GMRES is equivalent with MINRES, and FOM is equivalent with CG.

Since $\mathbf{r}_k^{\mathrm{CG}} = \|\mathbf{r}_k^{\mathrm{CG}}\|_2\mathbf{v}_{k+1} \perp \mathbf{r}_{k-1}^{\mathrm{MR}} \in \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$, it follows that

$$\mathbf{r}_k^{\mathrm{MR}} = s^2\mathbf{r}_{k-1}^{\mathrm{MR}} + \gamma\mathbf{v}_{k+1}, \tag{51}$$

where $\gamma = c^2\|\mathbf{r}_k^{\mathrm{CG}}\|_2$.

Since $\gamma\mathbf{v}_{k+1} \perp \mathbf{r}_{k-1}^{\mathrm{MR}} \in \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$, it follows that $\|\gamma\mathbf{v}_{k+1}\|_2 \le \|\mathbf{r}_k^{\mathrm{MR}}\|_2$. Moreover, since $\mathbf{r}_{k-1}^{\mathrm{MR}} \perp \mathbf{A}\,\mathcal{K}_{k-1}(\mathbf{A};\mathbf{r}_0)$ and $\gamma\mathbf{v}_{k+1} \perp \mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$, we have that $\mathbf{r}_{k-1}^{\mathrm{MR}} \perp \mathbf{x}_{k-1}^{\mathrm{ME}}$ and $\gamma\mathbf{v}_{k+1} \perp \mathbf{x}_k^{\mathrm{MR}}$. Therefore, with $\mathbf{e}_j^{\mathrm{ME}} \equiv \mathbf{x}-\mathbf{x}_j^{\mathrm{ME}}$, relation (51) implies

$$\begin{aligned}
|\alpha_k|\,\|\mathbf{r}_k^{\mathrm{MR}}\|_2 &= \left|\left(\mathbf{x}, \frac{\mathbf{r}_k^{\mathrm{MR}}}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}\right)\right| \le \frac{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2^2}\left|\left(\mathbf{x}, \frac{\mathbf{r}_{k-1}^{\mathrm{MR}}}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}\right)\right| + \left|\left(\mathbf{x}, \frac{\gamma\mathbf{v}_{k+1}}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}\right)\right| \\
&= \frac{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2^2}\left|\left(\mathbf{x}-\mathbf{x}_{k-1}^{\mathrm{ME}}, \frac{\mathbf{r}_{k-1}^{\mathrm{MR}}}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}\right)\right| + \left|\left(\mathbf{x}-\mathbf{x}_k^{\mathrm{MR}}, \frac{\gamma\mathbf{v}_{k+1}}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}\right)\right|,
\end{aligned}$$

and hence,

$$|\alpha_k| \le \frac{\|\mathbf{e}_k^{\mathrm{ME}}\|_2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2} + \frac{\|\mathbf{x} - \mathbf{x}_k^{\mathrm{MR}}\|_2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2}. \tag{52}$$

A combination of (52) and (45) with $\mathbf{z} = \mathbf{x}_{k+1}^{\mathrm{MR}}$ leads to

$$\frac{\|\mathbf{e}_k^{\mathrm{ME}}\|_2^2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2} \le \frac{\|\mathbf{x} - \mathbf{x}_{k+1}^{\mathrm{MR}}\|_2^2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2} + \left(\frac{\|\mathbf{e}_{k-1}^{\mathrm{ME}}\|_2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2} + \frac{\|\mathbf{x} - \mathbf{x}_k^{\mathrm{MR}}\|_2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2} \right)^2. \tag{53}$$

With

$$\beta_k \equiv \frac{\|\mathbf{e}_k^{\mathrm{ME}}\|_2}{\|\mathbf{A}^{-1}\|_2 \, \|\mathbf{r}_k^{\mathrm{MR}}\|_2},$$

and using the minimal residual property $\|\mathbf{r}_{k+1}^{\mathrm{MR}}\|_2 \le \|\mathbf{r}_k^{\mathrm{MR}}\|_2$, we obtain the following recursive upper bound from (53):

$$\beta_k^2 \le 1 + (\beta_{k-1} + 1)^2.$$

Now, a simple induction argument, using

$$\beta_0 = \frac{1}{\|\mathbf{A}^{-1}\|_2} \frac{\|\mathbf{e}_0^{\mathrm{ME}}\|_2}{\|\mathbf{r}_0^{\mathrm{MR}}\|_2} = \frac{1}{\|\mathbf{A}^{-1}\|_2} \frac{\|\mathbf{x}\|_2}{\|\mathbf{b}\|_2} \le 1,$$

shows that $\beta_k \le \nu_{k+1}$, and the definition of β_k implies

$$\frac{\|\mathbf{r}_k^{\mathrm{ME}}\|_2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2} \le \kappa_2(\mathbf{A}) \, \beta_k, \tag{54}$$

which completes the proof. ∎

For our analysis in section 3.2 of the additional errors in SYMMLQ, we also need a slightly more general result, formulated in the next theorem.

THEOREM 3.3. *Let* $\mathbf{c} = \mathbf{A}\mathbf{y}$ *for some* $\mathbf{y}$. *Consider the best approximation* $\mathbf{y}_k^{\mathrm{ME}}$ *of* $\mathbf{y}$ *in* $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{b})$ *and the* $\mathbf{y}_k^{\mathrm{MR}} \in \mathcal{K}_k(\mathbf{A}; \mathbf{b})$ *for which* $\mathbf{A}\mathbf{y}_k^{\mathrm{MR}}$ *is the best approximation of* $\mathbf{c}$ *in* $\mathbf{A}\mathcal{K}_k(\mathbf{A}; \mathbf{b})$.

Then, with ν_k *as in* (48), *we have*

$$\frac{\|\mathbf{c} - \mathbf{A}\,\mathbf{y}_k^{\mathrm{ME}}\|_2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2} \le \nu_{k+1} \, \kappa_2(\mathbf{A}) \, \mu_k, \quad \textit{where} \quad \mu_k \equiv \sup_{i \le k} \frac{\|\mathbf{c} - \mathbf{A}\,\mathbf{y}_i^{\mathrm{MR}}\|_2}{\|\mathbf{r}_i^{\mathrm{MR}}\|_2}. \tag{55}$$

Proof. The proof comes along the same lines as the proof of Theorem 3.2.

Replace the quantities $\mathbf{x}$ and $\mathbf{x}_k^{\mathrm{MR}}$ by $\mathbf{y}$ and $\mathbf{y}_k^{\mathrm{MR}}$. Since the $\mathbf{y}$ quantities fulfill the same orthogonality relations, (45) is valid also in the $\mathbf{y}$ quantities. This is also the case for the upper bound for $|\alpha_k| \, \|\mathbf{r}_k^{\mathrm{MR}}\|_2 = |(\mathbf{y}, \mathbf{r}_k^{\mathrm{MR}}/\|\mathbf{r}_k^{\mathrm{MR}}\|_2)|$. Hence, with $\mathbf{e}_j^{\mathrm{ME}} \equiv \mathbf{y} - \mathbf{y}_j^{\mathrm{ME}}$, we have

$$\frac{\|\mathbf{e}_k^{\mathrm{ME}}\|_2^2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2} \le \frac{\|\mathbf{y} - \mathbf{y}_{k+1}^{\mathrm{MR}}\|_2^2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2^2} + \left(\frac{\|\mathbf{e}_{k-1}^{\mathrm{ME}}\|_2}{\|\mathbf{r}_{k-1}^{\mathrm{MR}}\|_2} + \frac{\|\mathbf{y} - \mathbf{y}_k^{\mathrm{MR}}\|_2}{\|\mathbf{r}_k^{\mathrm{MR}}\|_2} \right)^2. \tag{56}$$

If we define $\widehat{\beta}_k \equiv \beta_k/\mu_k$, we find that

$$\widehat{\beta}_k^2 \le 1 + (\widehat{\beta}_{k-1} + 1)^2 \quad \text{and} \quad \widehat{\beta}_0 = \frac{1}{\mu_0 \|\mathbf{A}^{-1}\|_2} \frac{\|\mathbf{e}_0^{\mathrm{ME}}\|_2}{\|\mathbf{r}_0^{\mathrm{MR}}\|_2} \le 1.$$

Therefore, as in the proof of Theorem 3.2, $\widehat{\beta}_k \leq \nu_{k+1}$, which implies (55). ∎

For the relations between SYMMLQ and MINRES we have assumed exact arithmetic, that is, we have assumed an exact Lanczos process as well as an exact solve of the systems with L_k. However, we can exclude the influence of the Lanczos process by applying Theorem 3.2 right away to a system with a Lanczos matrix T_m and initial residual $\|\mathbf{r}_0\|_2 e_1$. In this setting, we have, for $k < m$, that [13, Proposition 1]

$$\|\mathbf{r}_k^{\mathrm{MR}}\|_2 = \|\mathbf{r}_0\|_2 \, \rho_k, \quad \text{where} \quad \rho_k \equiv |s_1 \cdot \ldots \cdot s_k|, \tag{57}$$

with s_j the sine in the jth Givens rotation for the QR decomposition of $\underline{T}_k$; ρ_k is the estimated reduction of the norms of the MINRES residuals. Note that (57) is also an immediate consequence of (50).

From relation (54) in combination with the fact that $\|\mathbf{r}_k^{\mathrm{ME}}\|_2 = \|\mathbf{r}_0\|_2 \, \|t_k\|_2$ (cf. (38), where, in this setting, $\mathbf{F}'_{k+1} = \mathbf{0}$), we conclude that

$$\|t_k\|_2 \leq \rho_k \, \kappa_2(T_m) \, \nu_{k+1} \quad \text{with} \quad \nu_k = k + \tfrac{1}{2}\ln(k), \tag{58}$$

for all $m > k$.

Note that inequality (58) is correct for any symmetric tridiagonal extension $\widetilde{T}_m$ of T_{k+1}: (58) holds with $\widetilde{T}_m$ instead of T_m. It has been shown in [5] that there is an extension $\widetilde{T}_m$ of which any eigenvalue is in a $\mathcal{O}(\mathbf{u}^{\frac{1}{4}})$-neighborhood of some eigenvalue of $\mathbf{A}$, and therefore, $\kappa_2(\widetilde{T}_m) \approx \kappa_2(\mathbf{A})$ in fairly good precision. This leads to our upper bound

$$\|t_k\|_2 \lesssim \rho_k \, \kappa_2(\mathbf{A}) \, \nu_{k+1} \quad \text{with} \quad \nu_k = k + \tfrac{1}{2}\ln(k). \tag{59}$$

In section 3.2, we will show that

$$\|\widehat{t}_k - t_k\|_2 \lesssim 5 \, \mathbf{u} \, \rho_k \, \kappa_2(\mathbf{A})^2 \, \left(\tfrac{1}{6}k^3 + \mathcal{O}(k^2 \ln k)\right). \tag{60}$$

The upper bound in (60) contains a square of the condition number. However, in the interesting situation where ρ_k decreases towards 0, the effect of the condition number squared will be annihilated eventually.

Remark 3.4. Except for the constants $k+\mathcal{O}(k)$ and $\frac{1}{6}k^3+\mathcal{O}(k^2 \ln k)$, the estimates (59) and (60), respectively, appear to be sharp (see Figure 8).

Although the maximal values of the ratio of $\|\widehat{t}_k - t_k\|_2/\rho_k$ in Figure 8 exhibit slowly growing behavior, the growth is not of order k^3. In the proof of (60) (cf. section 3.2), upper bounds as in (59) are used in a consecutive number of steps. In view of the irregular convergence of SYMMLQ, the upper bound (59) will be sharp for at most a few steps. By exploiting this observation, one can show that a growth of order k^2, or even less, will be more likely.

3.2. SYMMLQ recurrences. In this section we derive the upper bound (60).

Suppose that the jth recurrence for the γ_i's, with γ_i as defined in (33), is perturbed by a relatively small δ and all other recurrence relations are exact:

$$\delta = \ell_{jj}\widetilde{\gamma}_j + \ell_{jj-1}\gamma_{j-1} + \ell_{jj-2}\gamma_{j-2} \quad \text{with} \quad |\delta| \leq \mu \, \mathbf{u} \, |\ell_{jj}| \, |\gamma_j|. \tag{61}$$

The resulting perturbed quantities are labeled as $\widetilde{\ }$.

Then

$$\widetilde{t}_k - t_k = \delta \, M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} L_k^{-1} e_j. \tag{62}$$

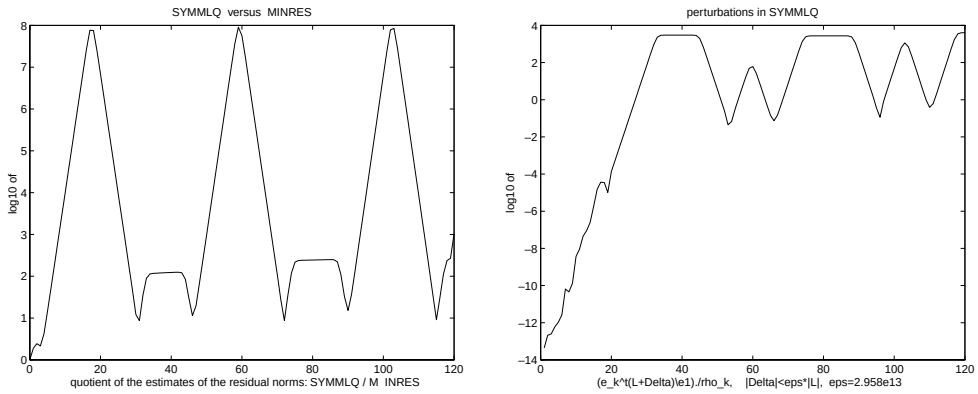


FIG. 8. *Results for the indefinite matrix with condition number* $3 \cdot 10^8$ *(as in the right pictures) of Figure* 4 *and Figure* 7. *The left picture shows* $\log_{10}$ *of the ratio* $\|\widehat{t}_k\|_2/\rho_k$ *of the estimated residual norm reduction* $\|\widehat{t}_k\|_2$ *of* SYMMLQ *and* ρ_k *for* MINRES (*cf.* (59)). *The right picture models* $\|\widehat{t}_k - t_k\|_2/\rho_k$ (*cf.* (60)) *with an artificial random pertubation* $\widetilde{\Delta}_L$, $|\widetilde{\Delta}_L| \gg |\Delta_L|$, *and* Δ_L *as in* (33)*: it shows the* $\log_{10}$ *of* $|e_k^T(L_k + \widetilde{\Delta}_L)^{-1}e_1/\rho_k - e_k^T(L_k + \Delta_L)^{-1}e_1/\rho_k|$, *where* $|\widetilde{\Delta}_L| \leq 3 \cdot 10^{-13}\,|L_k|$.

For $j = 1$, $\widetilde{t}_k - t_k$ is a multiple of the SYMMLQ residual for the T_m-system ($m > k$) and, as in the proof of inequality (59), Theorem 3.2 could be applied for estimating $\|\widetilde{t}_k - t_k\|_2$. For the situation where $j \neq 1$, Theorem 3.3 can be used.

To be more precise, we apply Theorem 3.3 with $\mathbf{v}_i = e_i$, $\mathbf{A} = T_m$, and $\mathbf{c} = e_j$. Then we have (in the notation as indicated in Theorem 3.3),

$$y_k^{\mathrm{ME}} = 0 \quad (k < j), \qquad \|e_j - T_m y_k^{\mathrm{ME}}\|_2 = \left\| M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} L_k^{-1} e_j \right\|_2 \quad (k \geq j), \tag{63}$$

and

$$y_k^{\mathrm{MR}} = 0 \quad (k+1 < j), \quad \|e_j - T_m\, y_k^{\mathrm{MR}}\|_2 = c_{j-1}\,\frac{\rho_k}{\rho_{j-1}} \leq \frac{\rho_k}{\rho_{j-1}} \quad (k+1 \geq j), \tag{64}$$

with c_{j-1} the cosine in the $(j-1)$th Givens rotation. Note that $\|e_j - T_m\, y_i^{\mathrm{MR}}\|_2/\rho_i \leq 1/\rho_{j-1}$ for all $i \leq k$. Therefore, by Theorem 3.3,

$$\left\| M_k \begin{bmatrix} e_{k-1}^T \\ e_k^T \end{bmatrix} L_k^{-1} e_j \right\|_2 \leq \kappa_2(T_m)\,\nu_{k+1}\,\frac{\rho_k}{\rho_{j-1}}. \tag{65}$$

For this specific situation, where $y_{j-1}^{\mathrm{ME}} = 0$, the estimate for β_k in the proof of Theorem 3.3 can be improved. If we take $\widehat{\beta}_k \equiv \rho_{j-1}\beta_k$, then we now have that $\widehat{\beta}_k^2 \leq 1 + (\widehat{\beta}_{k-1} + 1)^2$ and $\widehat{\beta}_{j-1} \leq 1$. This implies that $\rho_{j-1}\beta_k \leq \nu_{k-j+2}$. Therefore, the ν_{k+1} in (65) can be replaced by ν_{k-j+2}.

A combination of (62) with (65) gives (cf. (58) and following discussion)

$$\|\widetilde{t}_k - t_k\|_2 \leq \frac{|\delta|}{\rho_{j-1}}\,\rho_k\,\kappa_2(T_m)\,\nu_{k-j+2} \lesssim \frac{|\delta|}{\rho_{j-1}}\,\rho_k\,\kappa_2(\mathbf{A})\,\nu_{k-j+2}. \tag{66}$$

Using the definition of M_j and the recurrence relations for the γ_j, we can express t_{j-1} as

$$t_{j-1} = M_{j-1}\begin{bmatrix} \gamma_{j-2} \\ \gamma_{j-1} \end{bmatrix} = \begin{bmatrix} -\ell_{jj}\,\gamma_j \\ \ell_{j+1\,j-1}\,\gamma_{j-1} \end{bmatrix}.$$

Therefore, from (59), we have that

$$|\ell_{jj}|\,\frac{|\gamma_j|}{\rho_{j-1}} \le \frac{\|t_{j-1}\|_2}{\rho_{j-1}} \le \kappa_2(\mathbf{A})\,\nu_j. \tag{67}$$

Hence (cf. (61))

$$\frac{|\delta|}{\rho_{j-1}} \le \mu\,\mathbf{u}\,\kappa_2(\mathbf{A})\,\nu_j,$$

and, with (66), this gives

$$\|\widetilde{t}_k - t_k\|_2 \le \mu\,\mathbf{u}\,\rho_k\,\kappa_2(\mathbf{A})^2\,\nu_j\,\nu_{k-j+2}. \tag{68}$$

Because the recurrences are linear, the effect of a number of perturbations is the cumulation of the effects of single perturbations. If each recurrence relation is perturbed as in (61), then the estimate (60) appears as a cumulation of bounds as in (68). The vector $\widehat{t}_k$ in (60) represents the result of these successive perturbations due to finite precision arithmetic.

Finally, we will explain that the effect of rounding errors in solving $L^{-1}e_1$ can be described as the result of successively perturbed recurrence relations (61), with $\mu = 5$. First we note that the $\widetilde{\gamma}_k$'s resulting from the perturbation

$$\ell_{jj}\widetilde{\gamma}_j + \ell_{jj-1}\gamma_{j-1}(1+\mu\,\xi) + \ell_{jj-2}\gamma_{j-2} = 0 \quad \text{with} \quad |\xi| \le \mathbf{u}$$

are the same as those resulting from the perturbation

$$\ell_{j-1j-1}\widetilde{\gamma}_{j-1}(1+\mu\,\xi) + \ell_{j-1j-2}\gamma_{j-2} + \ell_{j-1j-3}\gamma_{j-3} = 0\,,$$

which means that a perturbation to the second term in the jth recurrence relation can also be interpreted as a similar perturbation to the first term in the $(j-1)$th recurrence relation.

Now we consider perturbations that are introduced in each recurrence relation due to finite precision arithmetic errors. Let $\widehat{\gamma}_j$ represent the actually computed γ_j, then

$$\widehat{\gamma}_j = -\frac{\ell_{jj-1}\widehat{\gamma}_{j-1}(1+\xi') + \ell_{jj-2}\widehat{\gamma}_{j-2}(1+\xi'')}{\ell_{jj}(1+2\,\xi)}, \quad \text{with} \quad |\xi|, |\xi'|, |\xi''| \le \mathbf{u},$$

and this can be rewritten, with different ξ and ξ', as

$$\ell_{jj}\widehat{\gamma}_j(1+3\,\xi) + \ell_{jj-1}\widehat{\gamma}_{j-1}(1+2\,\xi') + \ell_{jj-2}\widehat{\gamma}_{j-2} = 0, \quad \text{with} \quad |\xi|, |\xi'| \le \mathbf{u}.$$

Since the perturbation to the second term in this jth recurrence relation can be interpreted as a similar perturbation to the first term in the $(j-1)$th recurrence relation (which was already perturbed with a factor $(1+3\xi)$), we have that the computed $\widehat{\gamma}_j$ can be interpreted as the result of perturbing each leading term with a factor $(1+5\xi)$.

4. Discussion and conclusions. In Krylov subspace methods there are two main effects of floating point finite precision arithmetic errors. One effect is that the generated basis for the Krylov subspace deviates from the exact one. This may lead to a loss of orthogonality of the Lanczos basis vectors, but the main effect on the

iterative solution process is a delay in convergence rather than misconvergence. In fact, what happens is that we try to find an approximated solution in a subspace that is not as optimal, with respect to its dimension, as it could have been.

The other effect is that the determination of the approximation itself is perturbed with rounding errors, and this is, in our view, a serious point of concern; it has been the main theme of this study. In our study we have restricted ourselves to symmetric indefinite linear systems $\mathbf{Ax} = \mathbf{b}$. Before we review our main results, it should be noted that we should expect upper bounds for relative errors in approximations for $\mathbf{x}$ that contain at least the condition number of $\mathbf{A}$, simply because we can in general not compute $\mathbf{Ax}_k$ exactly. We have studied the effects of perturbations to the computed solution through their effect on the residual, because the residual (or its norm) is often the only information that we get from the process. This residual information is often obtained in a cheap way from some update procedure, and it is not uncommon that the updated residual may take values far smaller than machine precision (relative to the initial residual). Our analysis shows that there are limits on the reduction of the true residual because of errors in the approximated solution. For GMRES, this observation has also been made in [3].

In view of the fact that we may expect at least a linear factor $\kappa_2(\mathbf{A})$, when working with Euclidean norms, GMRES* (section 2.2) and SYMMLQ (section 3) lead to acceptable approximate solutions. When these methods converge, then the relative error in the approximate solution is, apart from modest factors, bounded by $\mathbf{u}\,\kappa_2(\mathbf{A})$. SYMMLQ is attractive since it minimizes the norm of the error, but it does so with respect to $\mathbf{A}$ times the Krylov subspace, which may lead to a delay in convergence with respect to GMRES* (or MINRES), by a number of iterations that is necessary to gain a reduction by $\kappa_2(\mathbf{A})$ in the residual; see Theorem 3.2 (also Figure 8). For ill-conditioned systems, this may be considerable.

As has been pointed out in [11], the conjugate gradient iterates can be constructed with little effort from SYMMLQ information if they exist. For indefinite systems the conjugate gradient iterates are well defined for at least every other iteration step, and they can be used to terminate the iteration if this is advantageous. However, the conjugate gradient process features no minimization property (in contrast to the positive definite case) when the matrix is indefinite, and so there is no guarantee that any of these iterates will be sufficiently close to the desired solution before SYMMLQ converges.

For indefinite symmetric systems we see that MINRES may lead to large perturbation errors: for MINRES the upper bound contains a factor $\kappa_2(\mathbf{A})^2$ (section 2.3). This means that if the condition number is large, then the methods of choice are GMRES or SYMMLQ. Note that for the symmetric case, GMRES can be based on the three-term recurrence relation, which means that the only drawback is the necessity to store all the Lanczos vectors. If storage is at a premium, then SYMMLQ is the method of choice.

If the given system is well conditioned, and if we are not interested in very accurate solutions, then MINRES may be an attractive choice.

Of course, one may combine any of the discussed methods with a variation on iterative refinement: after stopping the iteration at some approximation $\mathbf{x}_k$, we compute the residual $\mathbf{r}(\mathbf{x}_k) = \mathbf{b} - \mathbf{Ax}_k$, if possible in higher precision, and we continue to solve $\mathbf{Az} = \mathbf{r}(\mathbf{x}_k)$. The solution $\mathbf{z}_j$ of this system is used to correct $\mathbf{x}_k$: $\mathbf{x}_{\text{appr}} = \mathbf{x}_k + \mathbf{z}_j$. The procedure could be repeated, and eventually this leads to approximations for $\mathbf{x}$ so that the relative error in the residual is in the order of machine precision (for more details

on this, see [14]). However, if we would use MINRES, then, after restart, we have to carry out at least a number of iterations for the reduction by a factor equal to the condition number, in order to arrive at something of the same quality as GMRES*, which may make the method much less effective than GMRES*. For situations where $\kappa_2(\mathbf{A}) \geq 1/\sqrt{\mathbf{u}}$, MINRES may even be incapable of getting at a sufficient reduction for the iterative refinement procedure to converge.

It is common practice among numerical analysts to test the convergence behavior of Krylov subspace solvers for symmetric systems with well-chosen diagonal matrices. This often gives quite a good impression of what to expect for nondiagonal matrices with the same spectrum. However, as we have shown in our section 2.5, for MINRES this may lead to a too optimistic picture, since floating point error perturbations with MINRES for a diagonal matrix lead to errors in the residual (and the approximated solution) that are a factor $\kappa_2(\mathbf{A})$ smaller than for nondiagonal matrices.

Appendix.

LEMMA A.1. *If, for a matrix* $\mathbf{C}$, $n_C = \min(n_c, n_r)$ *with* n_c *the maximum number of nonzeros per column and* n_r *the maximum number of nonzeros per row, then*

$$\| \, |\mathbf{C}| \, \|_2 \leq \sqrt{n_C} \, \|\mathbf{C}\|_2. \tag{69}$$

Proof. We prove the lemma with respect to columns; the row variant follows from the fact that $\|\mathbf{B}^T\|_2 = \|\mathbf{B}\|_2$ for any matrix $\mathbf{B}$.

Since $\| \, |\mathbf{C}| \, \|_2^2 \leq n_C \max_j \left(\sum_i |c_{ij}|^2 \right)$ (see [15, Theorem 4.2]), we have

$$\| \, |\mathbf{C}| \, \|_2^2 \leq n_C \max_j \|\mathbf{C}e_j\|_2^2 \leq n_C \, \|\mathbf{C}\|_2^2. \qquad \square$$

Acknowledgments. The writing of this paper has been an exercise in modesty. We have to admit that it was only with extensive help of three anonymous referees, who invested embarrassing amounts of time, that the present version of this manuscript could be written. Somehow we seem to have developed a certain blindness for inaccuracies in the often complicated formulas, in the course of expressing our ideas. We are extremely thankful for the patience of the referees and for their detailed advice.

REFERENCES

[1] P. N. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.
[2] A. M. BRUASET, *A Survey of Preconditioned Iterative Methods*, Longman Scientific and Technical, Harlow, UK, 1995.
[3] J. DRKOŠOVÁ, A. GREENBAUM, M. ROZLOŽNÍK, AND Z. STRAKOŠ, *Numerical stability of GMRES*, BIT, 35 (1995), pp. 309–330.
[4] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The John Hopkins University Press, Baltimore, London, 1996.
[5] A. GREENBAUM, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
[6] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics 17, SIAM, Philadelphia, PA, 1997.
[7] A. GREENBAUM AND Z. STRAKOŠ, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.
[8] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
[9] C. C. PAIGE, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.
[10] C. C. PAIGE, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.

[11] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[12] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall Ser. Comput. Math., Prentice-Hall, Englewood Cliffs, NJ, 1980.
[13] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[14] K. Turner and H. F. Walker, *Efficient high accuracy solutions with GMRES(m)*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 815–825.
[15] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969/1970), pp. 14–23.
[16] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 752–771

MULTILEVEL ONE-WAY DISSECTION FACTORIZATION*

ALAN GEORGE†, WEI-PAI TANG†, AND YA DAN WU†

Abstract. Strategies for choosing an effective solver for a large sparse matrix equation are governed by the particular application. In this article, the context is the numerical solution of unsteady incompressible Navier–Stokes flow. When thousands of matrix equations differing only in their right-hand sides must be solved, a multilevel one-way dissection scheme is an attractive choice. This method has the property that large parts of the matrix factors are not stored; they are (implicitly) regenerated as needed during the solution process. The resulting storage requirement is competitive with those of preconditioned iterative methods. In addition, the efficiency at the solution stage is much superior to the iterative competitors.

Analysis of the storage and operation counts for the multilevel one-way dissection is presented along with numerical results for unsteady incompressible Navier–Stokes flow on a curvilinear grid. The improvements in performance of our new methods over other competitive methods are significant.

Key words. multilevel, one-way dissection, ordering, incompressible flow

AMS subject classifications. 65F05, 65N06, 76D05

PII. S0895479898332564

1. Introduction.

In solving the unsteady incompressible Navier–Stokes equations (INSE), the projection method and its numerous variants are very effective finite difference methods (see, e.g., [2, 19, 20, 22]). With this method, the most time-consuming task is the solution of a discretized Poisson equation for each time step. For a complex region Ω, a curvilinear grid is required; in this work, a half-staggered curvilinear grid [2, 14] is used. The purpose of the present study is to develop and validate an effective Poisson solver for unsteady viscous incompressible flow with irregular geometry.

Let the discretized Poisson equation on a half-staggered curvilinear grid be

$$Ax = f, \tag{1.1}$$

where the discretized Poisson operator uses a nine-point stencil. For our two-dimensional flow problems, the matrix size is between 40,000 and 330,000, where the corresponding grid is from 200×200 to 550×600. The right-hand sides are unrelated in time, and A is symmetric and semidefinite. In particular, A is singular and the zero eigenvalue has two independent eigenvectors, which implies that the right-hand side must meet two constraints[1] (see [8, 10]). Since (1.1) is to be solved at every time step for the unsteady INSE, an effective solver is crucial.

There are many methods which can be used for solving this problem. In general, those methods fall into two classes: iterative methods (preconditioner based, or multilevel based) and direct methods (factorization based). The choice is usually governed by a combination of two requirements: storage and computation.

*Received by the editors January 15, 1998; accepted for publication (in revised form) by S. Vavasis May 13, 2000; published electronically October 31, 2000. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/simax/22-3/33256.html

†Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (jageorge@sparse1.uwaterloo.ca, wptang@bryce1.uwaterloo.ca, y4wu@elora.uwaterloo.ca).

[1]For three-dimensional flow problems, the resulting Poisson operator is a 27-point stencil. There are many more independent eigenvectors which are associated with the zero eigenvalue which causes extra difficulties for many other iterative techniques. Our technique and analysis in this paper can be extended to many three-dimensional problems. See [11].

Multilevel-type methods are often particularly effective for Poisson operators. However, for two-dimensional Navier–Stokes flow problems, since the Poisson operator derived on a half-staggered grid has two independent eigenvectors associated with a zero eigenvalue, there are two constraints which are imposed on the right-hand side f. These constraints are difficult to satisfy on the coarse meshes. Consequently, the potential of the multilevel methods cannot be fully realized for this type of problem. In [13, 15] a comparison between a fast solver and a multigrid method on a nonuniform rectangular half-staggered grid showed that the former is about 6 times faster than the multigrid method. The solver presented in this study is an improvement over the fast solver of [13] (in the solution stage).

For three-dimensional problems, the situation is even more challenging; there are many more independent eigenvectors associated with the zero eigenvalue [11]. Moreover, the eigenvectors are difficult to compute.

For preconditioner-based iterative methods, the storage requirement is modest, but a relatively large number of iterations is required because the problem is difficult and a fine grid is required in order to obtain an acceptable resolution. As we will see in section 4, for a medium-sized grid (250×250), PCG-ILU(2) [6] needs 238 iterations to reduce the residual norm by a factor of 10^{-5}.

Direct methods, on the other hand, are more effective in terms of computation requirement in the solution stage, but the storage required for the matrix factors is prohibitively large. For a 250×250 grid, direct methods using the natural ordering can require up to 10 times the storage of methods proposed in this paper; even the optimal nested dissection ordering requires twice the storage of this method (see Table 3.1).

For a two-dimensional INSE, any effective Poisson solver for this particular application on a topologically rectangular $p \times q$ grid should meet the following requirements:

- Storage requirement $\approx O(p^\alpha q^\beta), \alpha \simeq 1, \beta = 1$.
- Solution cost for each time step $\approx O(p^\alpha q^\beta), \alpha \simeq 1, \beta = 1$.

That means the storage and solution cost are proportionally close to the number of unknowns, or the ratios rise only slightly with the number of unknowns. Since thousands of systems differing only in the right-hand side are to be solved, the cost for the factorization or the construction of the preconditioner is not crucial because it can be amortized over the large number of solution steps. This fact suggests that the use of a one-way dissection (1WD) [9, 18] method as a solver may be attractive.

The multilevel 1WD methods are direct methods that share with iterative methods the property of being economical with memory. Only part of the factor is saved during the factorization; the majority of the fill-ins in the off-diagonal blocks are "thrown away" and (effectively) recomputed as required during each solution process. This is the key to achieving good balance between storage requirement and solution time. It is a direct method, since only a finite number of operations are required for the solution. It also has the flavor of a domain decomposition iterative method, since solutions on subdomains are repeatedly computed in one solution process. For the details of using domain decomposition and preconditioners for solving compressible Navier–Stokes problems, see [1].

In [12, 18], the one-level and two-level 1WD schemes were studied. For a rectangular $p \times q$ grid problem, the storage requirement and the computational requirements for factorization and solution are given in Table 1.1.

In this paper, 1WD methods using levels greater than two are analyzed. Their utility and efficiency are demonstrated in the solving of some difficult flow problems

TABLE 1.1

The storage requirement and the computational requirements for factorization and solution for one-level and two-level 1WD on a $p \times q$ grid. The factorization cost for two-level 1WD is $O(p^{\frac{5}{3}}q^2)$ in [12, 18]; here it is improved to $O(p^{\frac{7}{3}}q)$.

	Storage requirement	Factorization cost	Solution cost
One-level	$O(p^{\frac{3}{2}}q)$	$O(p^{\frac{5}{2}}q)$	$O(p^{\frac{3}{2}}q)$
Two-level	$O(p^{\frac{4}{3}}q)$	$O(p^{\frac{7}{3}}q)$	$O(p^{\frac{4}{3}}q)$

using very large grids. For clarity, we consider a five-point stencil equation in the analysis. The generalization of our results to the nine-point stencil is straightforward.

In the next section, a brief description of the numerical method used for INSE is presented. In particular, the characteristic of the Poisson operator derived from this method is discussed. Section 3 describes the multilevel 1WD ordering method. Its analysis is presented in section 4, with the detailed proofs provided in the appendices. A comparison with PCG-ILU(2) and PCG-ILU(4) methods [6] is also presented to demonstrate the new solver's efficiency. Numerical results are listed in section 5.

2. Numerical method for INSEs. Consider the two-dimensional unsteady INSE

$$\frac{\partial \mathbf{w}}{\partial t} + u\frac{\partial \mathbf{w}}{\partial x} + v\frac{\partial \mathbf{w}}{\partial y} + \text{grad } p = \frac{1}{\text{Re}} \text{ div grad } \mathbf{w}, \tag{2.1}$$

$$\text{div } \mathbf{w} = 0. \tag{2.2}$$

Here $\mathbf{w} = (u, v)'$, with initial condition

$$\mathbf{w}(x, y, 0) = \mathbf{w}^0(x, y) \qquad \text{in } \Omega,$$

the constraint condition (2.2), and boundary condition

$$\mathbf{w}(x, y, t) = \mathbf{w}_b(x, y, t) \qquad \text{on } \partial\Omega.$$

The boundary condition on $\partial\Omega$ satisfies the consistency condition

$$\oint_{\partial\Omega} \mathbf{w}_n ds = 0. \tag{2.3}$$

As noted earlier, for a region Ω with complex geometry, a curvilinear grid is often required. We use a curvilinear half-staggered grid where the velocity is defined at the nodes, and the pressure is defined at the center of the cell of discretization [15], as depicted in Figure 2.1(a).

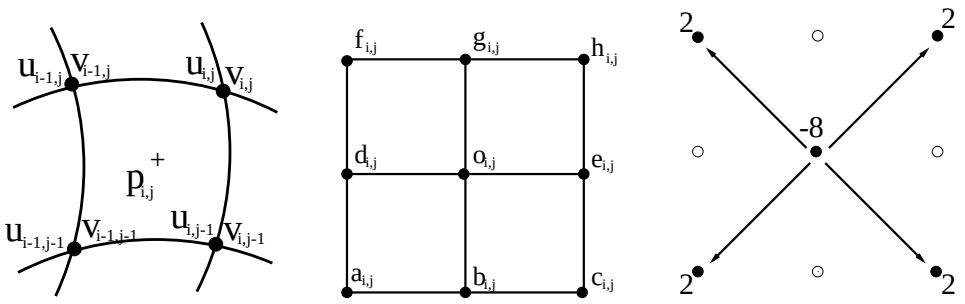
In the curvilinear coordinate system (ξ, η), the momentum equation (2.1) and continuity equation (2.2) can be obtained by a smooth coordinate transformation:

$$x = x(\xi, \eta), \qquad y = y(\xi, \eta).$$

After the coordinate transformation, the computational region, which may be a union of rectangles, is covered by a square mesh $\Delta\xi = \Delta\eta = 1$.

In the computational region (ξ, η), the partial derivatives of the transformation functions are approximated by central differencing. Thus for each node point (i, j)

$$x_\xi \approx \tfrac{x_{i+1,j} - x_{i-1,j}}{2\Delta\xi} \equiv x_c, \qquad y_\xi \approx \tfrac{y_{i+1,j} - y_{i-1,j}}{2\Delta\xi} \equiv y_c,$$
$$x_\eta \approx \tfrac{x_{i,j+1} - x_{i,j-1}}{2\Delta\eta} \equiv x_e, \qquad y_\eta \approx \tfrac{y_{i,j+1} - y_{i,j-1}}{2\Delta\eta} \equiv y_e.$$



(a) Half-staggered curvilinear grid.

(b) Nine-point stencil.

(c) Skewed five-point stencil.

FIG. 2.1. *Half-staggered curvilinear grid and the stencil of Poisson operator.*

The transformation Jacobian at (i, j) is given by

$$J = x_\xi y_\eta - x_\eta y_\xi \approx x_c y_e - x_e y_c \equiv s.$$

Now grad p is approximated at (i, j) by

$$Gp = \frac{1}{s} \begin{pmatrix} \frac{\bar{\delta} p}{\Delta \xi} y_e - \frac{\bar{\delta} p}{\Delta \eta} y_c \\ -\frac{\bar{\delta} p}{\Delta \xi} x_e + \frac{\bar{\delta} p}{\Delta \eta} x_c \end{pmatrix},$$

where

$$\frac{\bar{\delta} p}{\Delta \xi} = \frac{1}{2}(p_{i+1,j+1} + p_{i+1,j} - p_{i,j+1} - p_{i,j})$$

and

$$\frac{\bar{\delta} p}{\Delta \eta} = \frac{1}{2}(p_{i+1,j+1} + p_{i,j+1} - p_{i+1,j} - p_{i,j}).$$

If

$$R = \begin{pmatrix} y_e & -y_c \\ -x_e & x_c \end{pmatrix},$$

then $Gp = \frac{1}{s} R\ \bar{G}p$, where $\bar{G}p = (\frac{\bar{\delta} p}{\Delta \xi}, \frac{\bar{\delta} p}{\Delta \eta})^T$. Here div $\mathbf{w}$ is approximated by $\frac{1}{s} D\mathbf{w}$, and $D\mathbf{w} = \bar{D}\ R^T \mathbf{w} = -(R\ \bar{G})^T \mathbf{w}$.

Let the INSE, upon central differencing discretization in the (ξ, η) space, be represented by

$$\frac{d\mathbf{w}}{dt} + \mathbf{F}(\mathbf{w}, t) + Gp = 0 \tag{2.4}$$

and

$$D\mathbf{w} = \mathbf{0}, \tag{2.5}$$

where $\mathbf{F}$ is a nonlinear operator corresponding to the convection and diffusion, including the boundary condition and nonhomogeneous terms.

The Crank–Nicholson scheme for (2.4) is

$$\frac{\mathbf{w}^{n+1}-\mathbf{w}^n}{\Delta t}+\mathbf{H}(\mathbf{w}^n,\mathbf{w}^{n+1},t^{n+\frac{1}{2}})+Gp^{n+\frac{1}{2}}=0, \tag{2.6}$$

where $\mathbf{H}(\mathbf{w}^n,\mathbf{w}^{n+1},t)$ is a consistent and smooth approximation of $\mathbf{F}(\mathbf{w},t)$.

Using the component-consistent pressure correction projection method [16] on (2.6) and (2.5) yields

$$\frac{\tilde{\mathbf{w}}^{n+1}-\mathbf{w}^n}{\Delta t}+\mathbf{H}(\mathbf{w}^n,\tilde{\mathbf{w}}^{n+1},t^{n+\frac{1}{2}})+Gp^{n-\frac{1}{2}}=0, \tag{2.7}$$

$$\frac{\mathbf{w}^{n+1}-\tilde{\mathbf{w}}^{n+1}}{\Delta t}+G\phi=0, \tag{2.8}$$

and

$$D\mathbf{w}^{n+1}=0, \tag{2.9}$$

where $\tilde{\mathbf{w}}^{n+1}$ is the auxiliary velocity, and $\phi=p^{n+\frac{1}{2}}-p^{n-\frac{1}{2}}$. From (2.8) and (2.9), we obtain the discrete Poisson equation

$$DG\phi=\frac{1}{\Delta t}D\tilde{\mathbf{w}}^{n+1}, \tag{2.10}$$

where $DG=-(R\ \bar{G})^T(\frac{1}{s}R\ \bar{G})$.

The solution procedure per time step is as follows:

1. Solve for $\tilde{\mathbf{w}}^{n+1}$ from (2.7).
2. Solve for ϕ from (2.10).
3. Update $\mathbf{w}^{n+1}$.

Obviously, the matrix resulting from the operator DG is symmetric and nonpositive. For the interior nodes on the computational region, DG is the nine-point stencil operator as listed in Figure 2.1(b). The corresponding coefficients ($a_{i,j}, b_{i,j}, \ldots,$ $h_{i,j}$ and $o_{i,j}$) at node (i,j) are as follows:

$$\begin{aligned}
a_{i,j} &= \mathcal{A}_{i-1,j-1}+2\mathcal{B}_{i-1,j-1}+\mathcal{C}_{i-1,j-1},\\
b_{i,j} &= -\mathcal{A}_{i-1,j-1}+\mathcal{C}_{i-1,j-1}-\mathcal{A}_{i,j-1}+\mathcal{C}_{i,j-1},\\
c_{i,j} &= \mathcal{A}_{i,j-1}-2\mathcal{B}_{i,j-1}+\mathcal{C}_{i,j-1},\\
d_{i,j} &= \mathcal{A}_{i-1,j-1}-\mathcal{C}_{i-1,j-1}+\mathcal{A}_{i-1,j}-\mathcal{C}_{i-1,j},\\
e_{i,j} &= \mathcal{A}_{i,j-1}-\mathcal{C}_{i,j-1}+\mathcal{A}_{i,j}-\mathcal{C}_{i,j},\\
f_{i,j} &= \mathcal{A}_{i-1,j}-2\mathcal{B}_{i-1,j}+\mathcal{C}_{i-1,j},\\
g_{i,j} &= -\mathcal{A}_{i-1,j}+\mathcal{C}_{i-1,j}-\mathcal{A}_{i,j}+\mathcal{C}_{i,j},\\
h_{i,j} &= \mathcal{A}_{i,j}+2\mathcal{B}_{i,j}+\mathcal{C}_{i,j},\\
o_{i,j} &= -(a_{i,j}+b_{i,j}+c_{i,j}+d_{i,j}+e_{i,j}+f_{i,j}+g_{i,j}+h_{i,j}),
\end{aligned}$$

where

$$\mathcal{A}_{i,j}\equiv\frac{1}{s}(x_e^2+y_e^2)|_{i,j},\quad \mathcal{B}_{i,j}\equiv-\frac{1}{s}(x_cx_e+y_cy_e)|_{i,j},\quad \mathcal{C}_{i,j}\equiv\frac{1}{s}(x_c^2+y_c^2)|_{i,j}.$$

In particular, if the original curvilinear grid degenerates to an equally spaced square grid, the Poisson operator is the well-known skewed five-point stencil as shown in Figure 2.1(c). (See [10] for details.) In general, no pressure boundary condition is given; if the velocity boundary condition is a Dirichlet condition, we use only this velocity boundary condition to form (2.10). Thus, we get a Neumann condition for

ϕ in (2.10). In [13], we have shown that the resulting Poisson operator has two independent eigenvectors corresponding to zero eigenvalues. One is a constant vector and the other has a checkerboard pattern. The singular system imposes extra constraints on the right-hand side. When the multigrid method is used, the extra constraints are difficult to satisfy on the coarse grid, which causes the method to converge very slowly.

3. Multilevel 1WD ordering methods. The basic idea of one-level 1WD ordering on a $p \times q$ topologically rectangular mesh[2] is to choose k vertical grid lines, i.e., separators (k is an integer satisfying $(1 < k < q)$), dissecting the grid into $k+1$ roughly equal independent subdomains, each subdomain being approximately a $p \times \frac{q-k}{k+1}$ rectangular subgrid. The nodes in the subdomains are numbered row by row, followed by those in the separators, as depicted in Figure 3.1(a). After reordering the matrix A by one-level 1WD, the nonzero structure of A is as shown in Figure 3.1(b). The equation (1.1) can be written as

$$Ax = \begin{bmatrix} D_{11} & C_1^T \\ C_1 & D_{22} \end{bmatrix} \begin{bmatrix} x_I \\ x_S \end{bmatrix} = \begin{bmatrix} f_I \\ f_S \end{bmatrix}, \tag{3.1}$$

where D_{11} and D_{22} are symmetric diagonal blocks corresponding to the subdomains and separators, respectively. In particular, D_{11} is a block diagonal matrix where each of its diagonal blocks is a banded matrix with a very small bandwidth. This allows for effective storage scheme of the factorization. The matrix C_1 is a very sparse matrix corresponding to the coupling terms between the subdomains and separators.

Applying block asymmetric LU factorization to A [12, Chapter 6] yields

$$A = LU = \begin{bmatrix} D_{11} & 0 \\ C_1 & I \end{bmatrix} \begin{bmatrix} I & D_{11}^{-1}C_1^T \\ 0 & S_1 \end{bmatrix}, \tag{3.2}$$

where the Schur complement $S_1 = D_{22} - C_1 D_{11}^{-1} C_1^T$ is a symmetric matrix. Due to the ordering, S_1 is a block tridiagonal matrix (see Figure 3.1(c)). When the number of the separators is large, S_1 is a very sparse matrix.

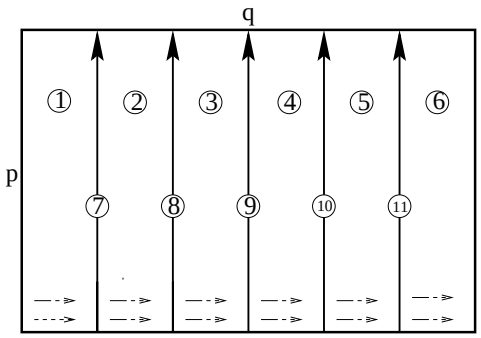
The solution process of the block 2×2 system involves the following:

1. Solve $D_{11}t_1 = f_I$.
2. Compute $t_2 = f_S - C_1 t_1$.
3. Solve $S_1 x_S = t_2$.
4. Solve $D_{11}(x_I - t_1) = -C_1^T x_S$.

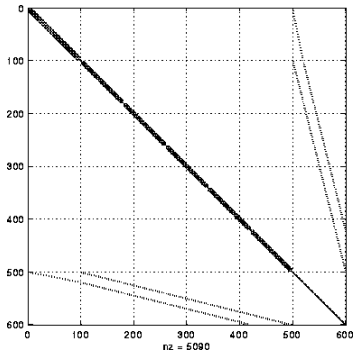
During the solution process, only D_{11} and S_1 are factored into $L_d L_d^T$ and $L_s L_s^T$ by Cholesky decomposition, and only the nonzeros in L_d, C_1, L_s are stored. That is, only the diagonal blocks of the whole matrix factor corresponding to the separators and subdomains are retained; the majority of the fill-ins in the off-diagonal blocks are "thrown away" during the factorization. The nonzero structure in the lower triangular factor is shown in Figure 3.1(c), where we only keep the black parts and throw away the gray parts. The key step in the 1WD method is to eliminate the storage for $D_{11}^{-1}C_1^T$ by the extra computation in step 4.

An ℓ-level ($\ell > 1$) 1WD is obtained by recursively applying the one-level 1WD on the original grid ℓ times, as depicted in Figures 3.2 and 3.3. These diagrams illustrate the two-level and three-level 1WDs applied to a rectangular computational region, the matrix structures induced by the 1WD ordering method, and the nonzero structure of the lower triangular factors.

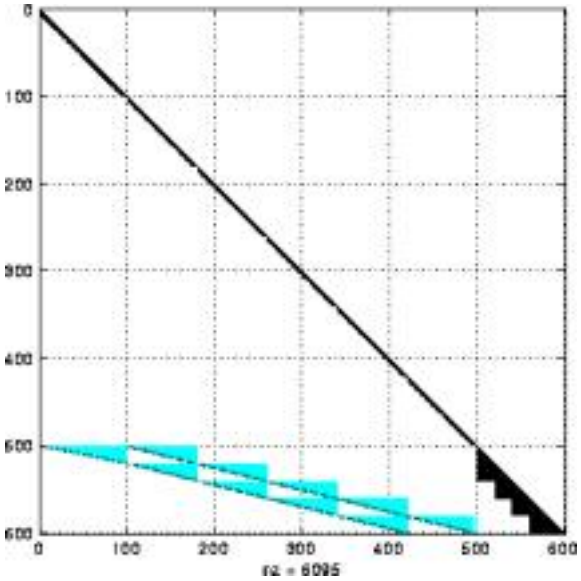
[2]It can be generalized to a union of rectangles.



(a) An example of applying one-level 1WD on a rectangular computational region. The numbers indicate the order in which the subdomains and separators are labeled.



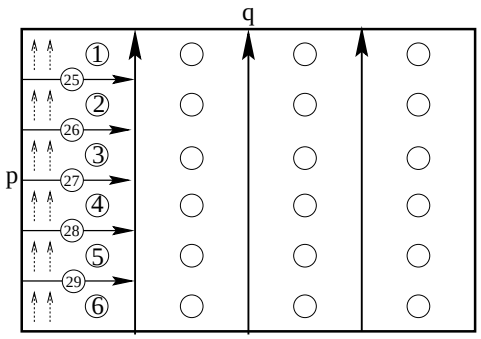
(b) The matrix structure induced by the one-level 1WD ordering of (a).



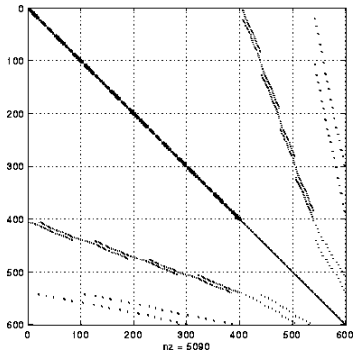
(c) The matrix structure of the lower triangular factor. The gray areas indicate the fill-ins in the off-diagonal blocks, which are thrown away. Only elements in the black areas are kept.

FIG. 3.1. *One-level* 1*WD on a* $p \times q$ *grid.*

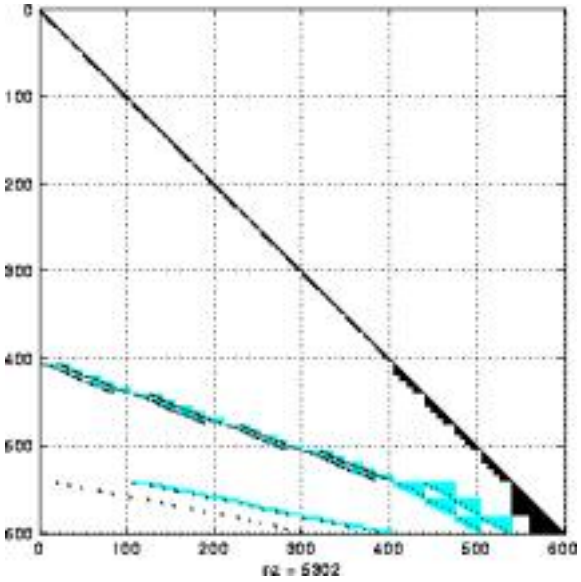
We compared the number of nonzeros in the lower triangular factor for two- and three-level 1WDs with the natural ordering and the nested dissection method. The results for 150×150 and 250×250 grids are listed in Table 3.1. Note, in particular, that the nested dissection ordering method needs twice as much storage as the three-level 1WD method. We normalized the storage for two-level 1WD to one. Obviously, the saving of 1WD in storage is significant compared with other direct methods. In



(a) An example of applying two-level 1WD on a rectangular computational region.



(b) The matrix structure induced by the two-level 1WD ordering of (a).

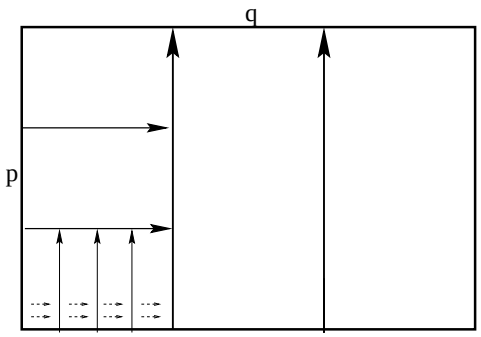


(c) The matrix structure of the lower triangular factor. The gray areas indicate the fill-ins in the off-diagonal blocks, which are thrown away. Only elements in the black areas are kept.

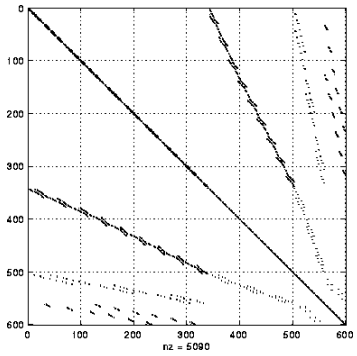
FIG. 3.2. *Two-level* 1*WD on a* $p \times q$ *grid.*

Table 4.1, we can also see that the storage requirement of two-level 1WD is close to that of the ILU(2) factorization.

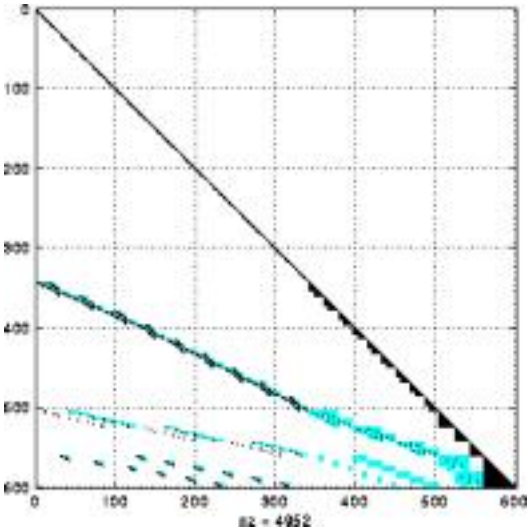
4. Storage requirement, operation counts analysis. We present the analysis of storage requirement and operation counts for the solution stage and the factorization stage in this section. First, consider the storage requirement for the ℓ-level 1WD method. Our computational scheme requires that only the diagonal blocks of the factor (i.e., L_d and L_s for one-level) and off-diagonal blocks of A (i.e., C_1 for



(a) An example of applying three-level 1WD on a rectangular computational region.



(b) The matrix structure induced by the three-level 1WD ordering of (a).



(c) The matrix structure of the lower triangular factor. The gray areas indicate the fill-ins in the off-diagonal blocks, which are thrown away. Only elements in the black areas are kept.

FIG. 3.3. *Three-level* 1*WD on a* $p \times q$ *grid.*

TABLE 3.1
A comparison of storage requirement for different orderings.

Grid size	Ordering algorithms				
	Natural ordering	1WD (1 level)	1WD (2 level)	1WD (3 level)	Nested dissection
150×150	8.33	1.55	1.	0.89	1.81
250×250	11.38	1.66	1.	0.84	1.73

one-level) to be stored. Since each diagonal block of the factor is dense near the diagonal (see Figure 3.3(c)), we can use the envelope storage scheme described in [12]; i.e., for each row in the matrix, all the entries from the first nonzero in each row to the diagonal are stored. These row portions are stored in contiguous locations in a one-dimensional array. An auxiliary index vector is used to point to the start of each row portion. This storage scheme requires less overhead storage compared with other sparse matrix storage formats. In particular, the elimination of the indexing for each individual nonzero reduces significant looping overhead in the solution stage. The off-diagonal blocks of A are very sparse, and their nonzeros are stored in a compressed sparse row format. The analysis of the storage requirement can be achieved in two steps. First, we show the optimal storage needed for one-level 1WD. Then the general case is a simple recursive result. The results are stated below, where the lower-order terms are omitted; the detailed proofs are presented in appendices.

THEOREM 4.1. *For a $p \times q$ grid, the storage requirement $S^{(l)}(p,q)$ for ℓ-level 1WD is approximately (omitting the lower-order terms)*

$$(\ell+1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q\, p^{\frac{\ell+2}{\ell+1}} + \frac{5p\, q}{4} - \frac{5p^2}{2} - (\ell-1)\left(\frac{3}{2}\right)^{\frac{1}{\ell+1}} q\, p^{\frac{\ell}{\ell+1}}, \tag{4.1}$$

and the optimal number of the separators on the top level (first level) is $k_{min} = (\frac{2}{3p^\ell})^{\frac{1}{\ell+1}} q - 1$.

Proof. See Appendix A. □

An estimate of the operation count required to solve $Ax = f$ by the solution process described in the last section, assuming the diagonal blocks of the factor have been given, is stated as Theorem 4.2 below.

THEOREM 4.2. *When the storage requirement for ℓ-level 1WD is minimized, the operation count ${\theta_s}^{(l)}(p,q)$ for the solution of a $p \times q$ grid problem is approximately (omitting the lower-order terms)*

$$2(2^{\ell+1}-1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q\, p^{\frac{\ell+2}{\ell+1}} + 5 \times 2^{\ell-2} p\, q - 5p^2. \tag{4.2}$$

Proof. See Appendix B. □

The storage requirements versus level for grid sizes from 200×200 to 1000×1000 are shown in Figure 4.1. For all of these grids, when the level goes up, the storage requirement goes down. Also, when the grid size increases, the saving of storage in using higher levels is more significant. Hence our interest in the use of higher levels for very large problems. Figure 4.2 shows the solution operation counts versus level for the corresponding grids. When the level goes up, the operation count increases sharply if the grid is small, but it increases more slowly if the grid is large. Thus, we can trade off storage requirements against computation time by choosing a suitable level for a particular grid.

We compare the performance of the two-level 1WD with a preconditioned conjugate gradient (PCG) method which uses ILU(2) or ILU(4) as preconditioner. The results from the simulation of driven polar cavity flows (Re=100) are listed in Table 4.1. For a 250×250 grid, the two-level 1WD is 91 times faster than PCG-ILU(2) and 67 times faster than PCG-ILU(4). The largest grid used in our flow computations is 550×600, and for the larger grids the speed-up over iterative methods is even greater. If the convergence criterion is less than 10^{-5}, we will also anticipate even greater advantage over iterative methods. The average storage requirement for each

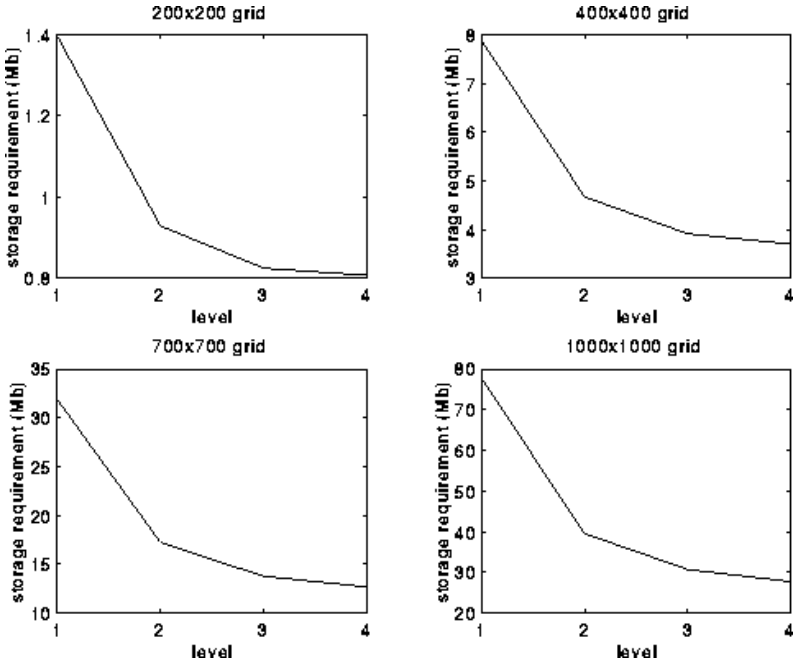


FIG. 4.1. *Storage requirements of* 1*WD against level.*

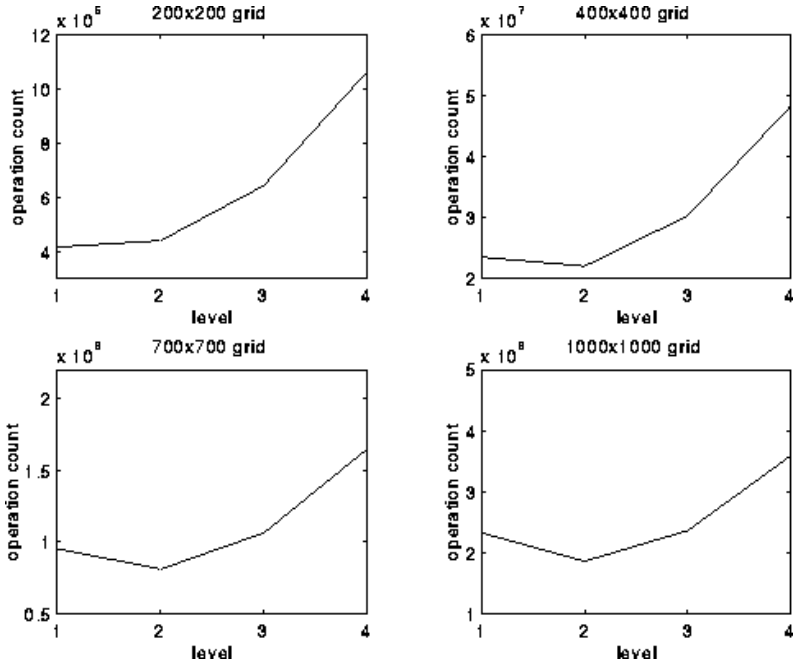


FIG. 4.2. *Solution operation counts of* 1*WD against level.*

grid node for the two-level 1WD is 7 percent more than that of PCG-ILU(2). It is noteworthy that the multilevel 1WD method is even faster than the fast solver for

TABLE 4.1
Comparison among two-level 1*WD, PCG-ILU*(2)*, and PCG-ILU*(4) *on average storage requirement for each grid node, iteration number, solution time required to reduce the residual norm by a factor of* 10^{-5} *at each time step for PCG-ILU*(2) *and ILU*(4)*, and factor time. The results are the average value over* 10 *time steps (using SUN*360 *computer and Fortran* 77 *compiler).*

Method	Grid size	Storage (per unknown)	Number of iterations	Avg. sol. time (seconds)	Fact. time (seconds)
1WD(1-level)	150×150	29.2	1	0.68	56.2
1WD(2-level)	150×150	19.6	1	0.84	121.6
PCG-ILU(2)	150×150	21.7	138.6	57.0	3.3
PCG-ILU(4)	150×150	27.5	90.9	42.9	5.4
1WD(1-level)	250×250	37.68	1	2.64	387.7
1WD(2-level)	250×250	23.4	1	3.03	840.6
PCG-ILU(2)	250×250	21.8	238.8	280.9	9.3
PCG-ILU(4)	250×250	27.7	155.4	207.8	15.0

nonuniform grids reported on in [13]. In addition, the latter cannot be applied in the curvilinear grid case.

The last column of Table 4.1 contains times for the factorization for both methods. The times for our method are large; however, the factorization is done only once, so this cost can be amortized over all the time steps. Since typically more than 1000 time steps are required, this cost is not a critical factor with respect to efficiency.

THEOREM 4.3. *When the storage requirement for* ℓ*-level* 1*WD is minimized, the factorization operation count* $\theta_f^{(l)}(p,q)$ *for a* $p \times q$ *grid is approximately (omitting the lower-order terms)*

one-level ($\ell = 1$):

$$\theta_f^{(1)}(p,q) = \frac{25}{9}\left(\frac{3}{2}\right)^{\frac{1}{2}} p^{\frac{5}{2}}\ q + \frac{3p^2\ q}{4} - \frac{13}{6}p^3 + O(p^{\frac{3}{2}}\ q);$$

multilevel ($\ell \geq 2$):

$$\theta_f^{(\ell)}(p,q) = \left(2^{\ell+2} - \frac{29}{9}\right)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} p^{\frac{2\ell+3}{\ell+1}}\ q + \alpha\ p^2\ q - \frac{13}{6}p^3 + O(p^{\frac{2\ell+1}{\ell+1}}\ q),$$

where $\alpha = \frac{55}{6}$ *when* $\ell = 2$*, and* $\alpha = 17 \times 2^{\ell-2} - \frac{29}{6}$ *when* $\ell > 2$.

Proof. See Appendix C. ∎

It is interesting to explore the asymptotic behavior of the storage estimate. Let $N = p = q$, which implies $l \leq 2(\log_2 N - 1)$. When $N \to \infty$, the optimal number of separators on the top level (first level) $k_{min} \to \sqrt{2} - 1$. Then we may get the following bound S_∞ for the minimum storage requirement $S^{(\ell)}(p,q)$:

$$S_\infty \approx 2\sqrt{2}N^2\log_2 N + O(N^2).$$

In practice, the number of the separators must be an integer. Assuming it is $k_{min} + \beta \quad (0 < \beta < 1\ )$ on the top level (first level), then the approximate minimum storage requirement $S^{(\ell)}(p,q)$ and its bound S_∞ are, respectively,

$$\begin{aligned} S^{(\ell)}(p,q) &= (\ell+1)\left(\tfrac{3}{2}\right)^{\frac{\ell}{\ell+1}} q\ p^{\frac{\ell+2}{\ell+1}} + \tfrac{5p\ q}{4} - \tfrac{5-3\beta}{2}p^2 \\ &\quad -\min\{1, \tfrac{5-3\beta}{4}\}(\ell-1)\left(\tfrac{3}{2}\right)^{\frac{1}{\ell+1}} q\ p^{\frac{\ell}{\ell+1}}, \\ S_\infty &\approx \max\{2, \tfrac{7+3\beta}{4}\}\sqrt{2}N^2\log_2 N + O(N^2). \end{aligned}$$

For solution and factorization operation counts, we need only to change the term $5p^2$ to $(5-3\beta)p^2$, and $\frac{13}{6}p^3$ to $\frac{13-7\beta}{6}p^3$, respectively. For simplicity in analysis, we assume the separator number k is real.

5. Numerical results. Numerical tests for the unsteady flows around a cylinder and an aerofoil are described in the following.

5.1. Unsteady flow over a circular cylinder. The fundamental fluid dynamics problem of a circular cylinder in uniform flow has been examined extensively in both computational and experimental studies and is considered a stringent test for flow solvers. The resulting flow field strongly depends on the Reynolds number.

Case I (low Reynolds numbers). For unsteady flow at Re = 100, the grid is nonorthogonal and 400×400. At a Reynolds number higher than 40, any perturbation excites an unsteady flow and eventually a periodic vortex shedding is established generating the well-known Von Kármán vortex street forced by vortices which are shed alternately with a distinct frequency from the top and bottom of the cylinder. This phenomenon has been addressed in several previous numerical and experimental works [4, 21]. In the present study, the formation of the vortex street is depicted clearly in spanwise vorticity contours (Figure 5.1).

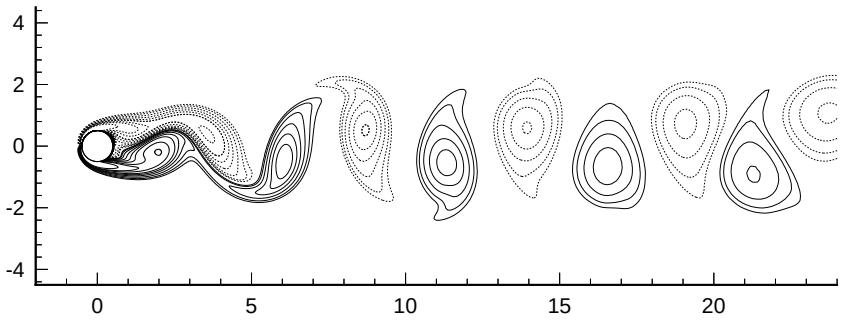
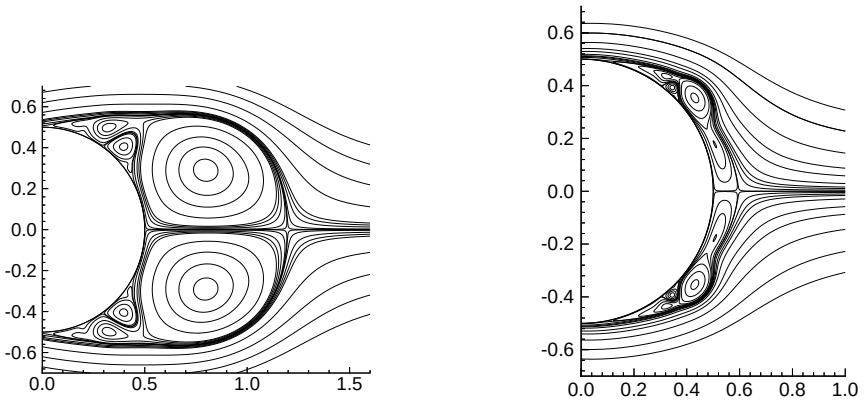


FIG. 5.1. *Spanwise vorticity of circular cylinder at* Re=100*; dotted and solid lines denote negative and positive levels, respectively.*

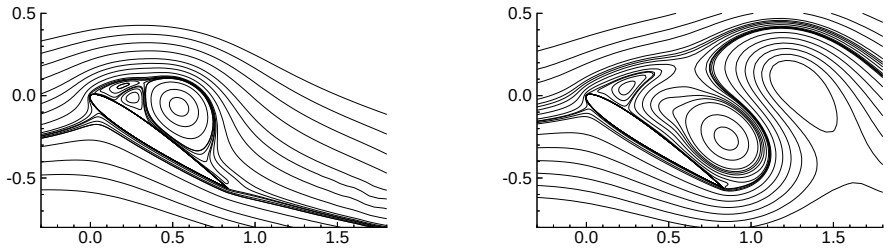
Case II (high Reynolds numbers). The initial development of an impulsively started flow at Re = 3000 and 9500 is simulated. At these Reynolds numbers, the flow exhibits a rich vortex structure which makes the computation difficult. In the following results, a 400×400 grid for Re = 3000 and 550×600 grid for Re = 9500 are used. According to experimental results [3], at early times, the α-phenomenon and the β-phenomenon are detected by visualization at Re = 3000 and 9500. The results of the computation in Figure 5.2 demonstrate the ability of the method to capture the α-phenomenon and β-phenomenon accurately. Its agreement with previous numerical results [17] is also quite good.

5.2. NACA 0012 aerofoil. The impulsively started flow around NACA 0012 is simulated at incidence 34° at Re = 1000. Figure 5.3 shows the streamlines of numerical flow at T = 1.6 and 4. From the figure, it is clear that a Von Kármán vortex street is created. The numerical results are in good agreement with experimental and previous numerical results [5, 7].



(a) Re = 3000, T = 2.50. (b) Re = 9500, T = 1.00.

FIG. 5.2. *Streamlines for impulsively started circular cylinder at* Re = 3000 *and* 9500.



(a) T = 1.6. (b) T = 4.0.

FIG. 5.3. *Streamlines for impulsively started NACA* 0012 *aerofoil at* Re = 1000*, incidence* 34°.

6. Conclusion. An effective multilevel 1WD ordering method was presented to solve a Poisson equation which resulted from the discrete unsteady INSEs on a two-dimensional half-staggered curvilinear grid. This discrete Poisson equation is difficult to solve iteratively, and no fast FFT-based direct methods are available [13]. The multilevel 1WD ordering method provides a good balance between storage requirement and solution time. Storage and operation counts have been derived, and the saving in storage is significant compared with other direct methods. The storage requirement of the 1WD is close to that of PCG-ILU(2), and the solution time is 1–2 orders of magnitude less than that of competitive methods. Although the factorization time is high, it can be amortized over the large number of solution steps, yielding a significant overall saving. Some difficult flows involving very large grids have been simulated, providing numerical results that are in good agreement with experimental and previous numerical results.

Appendix A. Storage requirement.

Proof of Theorem 4.1. First, we consider one-level 1WD method with k_1 separators on a $p \times q$ grid. The separators dissect the grid into $k_1 + 1$ roughly equal independent subdomains, each subdomain being approximately a $p \times \frac{q-k_1}{k_1+1}$ rectangular subgrid. After reordering A by the one-level 1WD, we partition A as in (3.1) and factor A as in (3.2), and keep only the nonzeros in the lower triangular factors L_d, L_s and the sparse matrix C_1.

D_{11} is a block diagonal matrix having $k_1 + 1$ diagonal blocks. Each diagonal block is a $\frac{p(q-k_1)}{k_1+1} \times \frac{p(q-k_1)}{k_1+1}$ band matrix with bandwidth $\frac{q+1}{k_1+1}$. After the Cholesky factorization, the number of nonzero elements in L_d is about

$$S(L_d) = (k_1 + 1) \times \frac{p(q - k_1)}{k_1 + 1} \frac{q + 1}{k_1 + 1}.$$

Approximately,

$$S(L_d) \leq \frac{p\ q^2}{k_1 + 1}.$$

C_1 is a $k_1 p \times p(q-k_1)$ sparse matrix, with each row having two nonzero elements. The storage requirement for C_1 is

$$S(C_1) = 2k_1 p.$$

The lower triangular factor L_s of matrix S_1 has $(k_1 - 1)$ full blocks and k_1 lower triangular blocks, all of which are $p \times p$ matrices. The number of nonzero elements in L_s is

$$S(L_s) = (k_1 - 1)p^2 + \frac{k_1 p(p + 1)}{2} = \frac{3k_1 p^2}{2} + \frac{k_1 p}{2} - p^2.$$

So, the total storage requirement $S(p, q, k_1)$ for one-level 1WD is

$$S(p, q, k_1) = S(L_d) + S(C_1) + S(L_s) \leq \frac{p\ q^2}{k_1 + 1} + \frac{3k_1 p^2}{2} + \frac{5k_1 p}{2} - p^2.$$

Since $k_1 \leq q/2$, we can replace the lower-order term $\frac{5k_1 p}{2}$ by its upper bound, and then

$$S(p, q, k_1) \leq \frac{p\ q^2}{k_1 + 1} + \frac{3k_1 p^2}{2} + \frac{5p\ q}{4} - p^2.$$

Obviously, when $k_1 = k_{min} = (\frac{2}{3p})^{\frac{1}{2}} q - 1$, the storage requirement $S(p, q, k_1)$ approaches the minimum $S^{(1)}(p, q)$,

$$S^{(1)}(p, q) = S(p, q, k_{min}) \leq 2 \left(\frac{3}{2}\right)^{\frac{1}{2}} q\ p^{\frac{3}{2}} + \frac{5p\ q}{4} - \frac{5p^2}{2},$$

which is just (4.1) when $\ell = 1$.

Assuming the storage requirement for ℓ-level ($\ell \geq 1$) 1WD satisfies the formula (4.1), we will show that the storage requirement of $(\ell + 1)$-level 1WD also satisfies (4.1).

For $(\ell+1)$-level 1WD, let the number of separators in the top level (first level) be $k_{\ell+1}$. They dissect the $p \times q$ grid into $k_{\ell+1}+1$ roughly equal independent subdomains. Each subdomain is a $p_\ell \times q_\ell$ rectangular subgrid, where $p_\ell = \frac{q-k_{\ell+1}}{k_{\ell+1}+1}, q_\ell = p$. These subgrids are ordered by ℓ-level 1WD. The minimum storage for each subgrid is approximately

$$(\ell+1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell+2}{\ell+1}} + \frac{5p_\ell\ q_\ell}{4} - \frac{5p_\ell^2}{2} - (\ell-1)\left(\frac{3}{2}\right)^{\frac{1}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell}{\ell+1}}.$$

The storage requirements for C_1 and L_s are $2k_{\ell+1}p$ and $\frac{3k_{\ell+1}p^2}{2} + \frac{k_{\ell+1}p}{2} - p^2$, respectively. Hence, the total storage for $(\ell+1)$-level 1WD is

$$\begin{aligned} S(p,q,k_{\ell+1}) \approx \quad & (k_{\ell+1}+1)\left[(\ell+1)\left(\tfrac{3}{2}\right)^{\frac{\ell}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell+2}{\ell+1}} + \tfrac{5p_\ell\ q_\ell}{4} - \tfrac{5p_\ell^2}{2} - (\ell-1)\left(\tfrac{3}{2}\right)^{\frac{1}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell}{\ell+1}}\right] \\ & + \tfrac{3k_{\ell+1}p^2}{2} + \tfrac{5k_{\ell+1}p}{2} - p^2. \end{aligned}$$

Substituting $\frac{q-k_{\ell+1}}{k_{\ell+1}+1}$ for p_ℓ, p for q_ℓ, we obtain approximately

$$\begin{aligned} S(p,q,k_{\ell+1}) \le \quad & (\ell+1)\left(\tfrac{3}{2}\right)^{\frac{\ell}{\ell+1}} p\ q\left(\frac{q}{k_{\ell+1}+1}\right)^{\frac{1}{\ell+1}} + \frac{3k_{\ell+1}p^2}{2} + \frac{5p\ q}{4} - p^2 \\ & - \frac{q^2}{k_{\ell+1}+1} - (\ell-1)\left(\tfrac{3}{2}\right)^{\frac{1}{\ell+1}} p\ q^{\frac{\ell}{\ell+1}}(k_{\ell+1}+1)^{\frac{1}{\ell+1}}. \end{aligned}$$

Consider the two highest-order terms,

$$(\ell+1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} p\ q\left(\frac{q}{k+1}\right)^{\frac{1}{\ell+1}} + \frac{3kp^2}{2}. \tag{A.1}$$

Obviously, (A.1) is minimum when $k = k_{min} = (\frac{2}{3p^{(\ell+1)}})^{\frac{1}{\ell+2}} q - 1$. Then letting $k_{\ell+1} = k_{min}$, the storage for $(\ell+1)$-level approaches the minimum $S^{(\ell+1)}(p,q)$,

$$S^{(\ell+1)}(p,q) = S(p,q,k_{min}) \le (\ell+2)\left(\frac{3}{2}\right)^{\frac{\ell+1}{\ell+2}} q\ p^{\frac{\ell+3}{\ell+2}} + \frac{5p\ q}{4} - \frac{5p^2}{2} - \ell\left(\frac{3}{2}\right)^{\frac{1}{\ell+2}} q\ p^{\frac{\ell+1}{\ell+2}}. \qquad \square$$

Appendix B. Operation count for solution.

Proof of Theorem 4.2. First, we consider one-level 1WD with k_1 separators. In the solution process, the matrix D_{11} is solved twice, which yields an operation count of approximately

$$2 \times 2 \times S(L_d) \le \frac{4p\ q^2}{k_1+1}.$$

The matrix C_1 is multiplied twice, which yields an operation count of approximately

$$2 \times S(C_1) = 4k_1p.$$

The matrix S_1 is solved once, which yields an operation count of approximately

$$2 \times S(L_s) = 3k_1p^2 + k_1p - 2p^2.$$

Therefore, the total operation count is approximately

$$\frac{4p\ q^2}{k_1+1} + 4k_1p + 3k_1p^2 + k_1p - 2p^2 \le \frac{4p\ q^2}{k_1+1} + 3k_1p^2 + \frac{5p\ q}{2} - 2p^2,$$

where we have replaced the lower-order term $5k_1p$ by its upper bound $\frac{5p\ q}{2}$. When the storage requirement is minimized, i.e., $k_1 = (\frac{2}{3p})^{\frac{1}{2}}q - 1$, the total operation count $\theta_s^{(1)}(p,q)$ for solution is approximately

$$\theta_s^{(1)}(p,q) = 6\left(\frac{3}{2}\right)^{\frac{1}{2}} q\ p^{\frac{3}{2}} + \frac{5p\ q}{2} - 5p^2.$$

Assuming that (4.2) is valid for ℓ-level 1WD, we will show that it is also valid for $(\ell+1)$-level 1WD.

For $(\ell+1)$-level 1WD, we regard it as applying one-level 1WD with $k_{\ell+1}$ separators on a $p \times q$ grid and then applying ℓ-level 1WD on the $(k_{\ell+1}+1)$ subgrids (each subgrid is a $p_\ell \times q_\ell$ rectangular grid, where $p_\ell = \frac{q-k_{\ell+1}}{k_{\ell+1}+1}, q_\ell = p$). When the storage is minimized for the subgrid, the operation count for solving each subgrid matrix is approximately

$$2(2^{\ell+1}-1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell+2}{\ell+1}} + 5\times 2^{\ell-2}p_\ell\ q_\ell - 5p_\ell^2.$$

There are $k_{\ell+1}+1$ subgrids and all are solved twice during the solution process. That yields an approximate operation count of

$$2\times(k_{\ell+1}+1)\times\left(2(2^{\ell+1}-1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell+2}{\ell+1}} + 5\times 2^{\ell-2}p_\ell\ q_\ell\right),$$

where the lower-order term p_ℓ^2 is ignored. Adding operation counts due to C_1 and S_1, i.e., $4k_{\ell+1}p$ and $3k_{\ell+1}p^2 + k_{\ell+1}p - 2p^2$, the total operation count for solution is roughly

$$2\times(k_{\ell+1}+1)\times\left(2(2^{\ell+1}-1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} q_\ell\ p_\ell^{\frac{\ell+2}{\ell+1}} + 5\times 2^{\ell-2}p_\ell\ q_\ell\right) + 3k_{\ell+1}p^2 + 5k_{\ell+1}p - 2p^2.$$

Substituting $\frac{q-k_{\ell+1}}{k_{\ell+1}+1}$ for p_ℓ, p for q_ℓ, we obtain approximately

$$4(2^{\ell+1}-1)\left(\frac{3}{2}\right)^{\frac{\ell}{\ell+1}} p\ q\left(\frac{q}{k_{\ell+1}+1}\right)^{\frac{1}{\ell+1}} + 5\times 2^{\ell-1}p\ q + 3k_{\ell+1}p^2 - 2p^2.$$

Therefore, when storage requirement is minimized, i.e., $k_{\ell+1} = (\frac{2}{3p^{\ell+1}})^{\frac{1}{\ell+2}}q - 1$, the total operation count for solution is approximately

$$\theta_s^{(\ell+1)}(p,q) = 2(2^{\ell+2}-1)\left(\frac{3}{2}\right)^{\frac{\ell+1}{\ell+2}} q\ p^{\frac{\ell+3}{\ell+2}} + 5\times 2^{\ell-1}p\ q - 5p^2. \qquad \square$$

Appendix C. Operation count for factorization.

First, we introduce a theorem from [12].

THEOREM C.1. *The number of operations required to compute the triangular factor L of the $N \times N$ matrix A is given by*

$$\frac{1}{2}\sum_{i=1}^{N-1}[\phi(i)-1][\phi(i)+2],$$

where $\phi(i)$ is the number of nonzeros in the ith column of L.

We now prove Theorem 4.3.

For one-level 1WD, the computation can be broken into three categories:

1. *The factorization of matrix* D_{11}. D_{11} is a block diagonal matrix having k_1+1 blocks, and each block is a $\frac{p(q-k_1)}{k_1+1} \times \frac{p(q-k_1)}{k_1+1}$ band matrix with bandwidth $\frac{q+1}{k_1+1}$. By Theorem C.1, the operation requirement for factoring matrix D_{11} is

$$\frac{1}{2}\left(\sum_{i=1}^{\frac{p(q-k_1)}{k_1+1}-1}\left(\frac{q+1}{k_1+1}-1\right)\left(\frac{q+1}{k_1+1}+2\right)\right)\times(k_1+1).$$

Assuming p, q are large enough, it is approximately

$$\frac{p\ q}{2}\left(\frac{q^2+2q}{(k_1+1)^2}+\frac{q}{k_1+1}\right).$$

2. *The factorization of matrix* S_1. This corresponds to factoring a $kp \times kp$ block tridiagonal matrix having blocks of size $p \times p$. By Theorem C.1, the operation requirement for factoring matrix S_1 is

$$\frac{1}{2}\left[(k_1-1)\sum_{i=1}^{p}(p+i-1)(p+i+2)+\sum_{i=1}^{p}(i-1)(i+2)\right]$$
$$=\frac{7k_1p^3}{6}-p^3+\frac{(3k_1-2)p^2}{2}-\frac{2k_1p}{3}.$$

3. *The computation of matrix* S_1.

$$S_1 = D_{22} - C_1 D_{11}^{-1} C_1^T.$$

D_{11} is a block diagonal matrix having k_1+1 blocks, and each block is a $\frac{p(q-k_1)}{k_1+1} \times \frac{p(q-k_1)}{k_1+1}$ band matrix with bandwidth $\frac{q+1}{k_1+1}$. C_1 is also a block matrix having $2k_1$ blocks, and each block is a $\frac{p(q-k_1)}{k_1+1} \times p$ matrix. Considering that C_1 is very sparse and $C_1(D_{11}^{-1}C_1^T)$ is symmetric, we can use the method of [12, 18] to compute $D_{11}^{-1}C_1^T = L_d^{-T}(L_d^{-1}C_1^T)$. When computing $W = L_d^{-1}C_1^T$, leading zeros in L_d should be exploited; when computing $\tilde{W} = L_d^{-T}W$, the computation should be stopped as soon as the last required element of $\tilde{W}$ has been computed.
Then the operation count for computing $D_{11}^{-1}C_1^T$ is

$$2k_1 \times (p-1) \times \frac{p(q-k_1)(q+1)}{(k_1+1)^2} \approx \frac{2p^2\ q^2}{k_1+1}.$$

In each block of C_1, there is only one nonzero entry in each row; hence the operation count for computing $C_1(D_{11}^{-1}C_1^T)$ is

$$2k_1 \times p \times 2p = 4k_1p^2.$$

Therefore, the total computation requirement for the factorization for one-level 1WD ordering is approximately

$$\frac{p\ q}{2}\left[\frac{q^2+2q}{(k_1+1)^2}+\frac{q}{k_1+1}\right]+\frac{7k_1p^3}{6}-p^3+\frac{(3k_1-2)p^2}{2}-\frac{2k_1p}{3}+\frac{2p^2\ q^2}{k_1+1}+4k_1p^2.$$

When $k_1 = (\frac{2}{3p})^{\frac{1}{2}} q - 1$, the expression above is approximately

$$\frac{25}{9}\left(\frac{3}{2}\right)^{\frac{1}{2}} p^{\frac{5}{2}}\ q + \frac{3p^2\ q}{4} - \frac{13}{6}p^3 + O(p^{\frac{3}{2}}q).$$

For $(\ell+1)$-level ($\ell \geq 1$) 1WD, we regard it as applying one-level 1WD with $k_{\ell+1}$ separators on $p \times q$ grid and applying ℓ-level 1WD on the $(k_{\ell+1}+1)$ subgrids (each subgrid is a $p_\ell \times q_\ell$ rectangular grid, in which $p_\ell = \frac{q-k_{\ell+1}}{k_{\ell+1}+1}, q_\ell = p$).

When the storage requirement is minimum, the operation count for factoring matrix D_{11} is approximately

$$(k_{\ell+1}+1) \times \theta_f^{(\ell)}(p_\ell, q_\ell).$$

D_{11} is a block diagonal matrix having $k_{\ell+1}+1$ blocks, and each block is a $p_\ell \times q_\ell$ grid problem which is ordered by ℓ-level 1WD. C_1 is also a block matrix having $2k_{\ell+1}$ blocks, and each block is a $\frac{p(q-k_{\ell+1})}{k_{\ell+1}+1}$ by p matrix.[3] Hence the operation count for computing $D_{11}^{-1}C_1^T$ is

$$2k_{\ell+1} \times p \times \theta_s^{(\ell)}(p_\ell, q_\ell),$$

and the operation count for computing $C_1(D_{11}^{-1}C_1^T)$ is

$$2k_{\ell+1} \times p \times 2p = 4k_{\ell+1}p^2.$$

Adding the operation count for factoring S_1, the total operation count for factorization is approximately

$$\begin{aligned}\theta_f(p,q,k_{\ell+1},p_\ell,q_\ell) = \quad & (k_{\ell+1}+1) \times \theta_f^{(\ell)}(p_\ell,q_\ell) + 2k_{\ell+1} \times p \times \theta_s^{(\ell)}(p_\ell,q_\ell) \\ & + 4k_{\ell+1}p^2 + \tfrac{7k_{\ell+1}p^3}{6} - p^3 + \tfrac{(3k_{\ell+1}-2)p^2}{2} - \tfrac{2k_{\ell+1}p}{3}.\end{aligned}$$

Let

$$p_\ell = \frac{q-k_{\ell+1}}{k_{\ell+1}+1}, \quad q_\ell = p, \quad k_{\ell+1} = \left(\frac{2}{3p^{\ell+1}}\right)^{\frac{1}{\ell+2}} q - 1.$$

Then,

$$\theta_f(p,q,k_{\ell+1},p_\ell,q_\ell) = \theta_f^{(\ell+1)}(p,q). \qquad \square$$

[3] To make the derivation clear, we order all the nodes in subdomains first, followed by their separators. In the real solution process we order the nodes as shown in Figures 3.2 and 3.3.

REFERENCES

[1] T. J. Barth, T. F. Chan, and W. P. Tang, *A parallel non-overlapping domain-decomposition algorithm for compressible fluid flow problems on triangulated domains*, Contemp. Math., 218 (1998), pp. 23–41.

[2] J. B. Bell, P. Colella, and H. M. Glaz, *A second-order projection method for the incompressible Navier–Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.

[3] R. Bouard and M. Coutanceau, *The early stages of development of the wake behind an impulsively started cylinder for* $40 < Re < 10^4$, J. Fluid Mech., 101 (1980), pp. 583–607.

[4] M. Coutanceau and J. R. Defaye, *Circular cylinder wake configurations: A flow visualization survey*, Appl. Mech. Rev., 44 (1991), pp. 255–306.

[5] O. Daube, Ta Phuoc Loc, P. Monnet, and M. Coutanceau, *Ecoulement instationnaire décollé d'un fluide incompressible autour d'un profil: une comparaison théorie-expérience*, AGARD Conference Proceedings 386, Neuilly-sur-Seine, France, 1985, Paper 3.

[6] E. F. D'Azevedo, P. A. Forsyth, and W. P. Tang, *Towards a cost-effective ILU preconditioner with high level fill*, BIT, 32 (1992), pp. 442–463.

[7] G. B. Deng, E. Guilmineau, J. Piquet, P. Queutey, and M. Visonneau, *Computation of unsteady laminar viscous flow past an aerofoil using the CPI method*, Internat. J. Numer. Methods Fluids, 19 (1994), pp. 765–794.

[8] M. Fortin, *An analysis of the convergence of mixed finite element methods*, RAIRO Anal. Numér., 11 (1977), pp. 341–354.

[9] A. George, *Numerical experiments using dissection methods to solve n by n grid problems*, SIAM J. Numer. Anal., 14 (1977), pp. 161–179.

[10] A. George, L. C. Huang, W. P. Tang, and Y. D. Wu, *Numerical simulation of unsteady incompressible flow* $Re \leq 9500$ *on the curvilinear half-staggered mesh*, SIAM J. Sci. Comput., 21 (2000), pp. 2331–2351.

[11] A. George, L. C. Huang, W. P. Tang, and Y. D. Wu, *Numerical solution for the time-dependent three-dimensional incompressible Navier-Stokes equations on a curvilinear half-staggered grid*, in the Sixth Annual Conference of the Computational Fluid Dynamics Society of Canada, University of Victoria, Victoria, British Columbia, Canada, 1998, pp. VIII-25–VIII-30.

[12] A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[13] G. Golub, L. C. Huang, H. Simon, and W. P. Tang, *A fast solver for incompressible Navier-Stokes equations with finite difference methods*, SIAM J. Sci. Comput., 19 (1998), pp. 1606–1624.

[14] L. C. Huang, *The numerical solution of the unsteady incompressible Navier-Stokes equations on the curvilinear half-staggered mesh*, J. Comput. Math, to appear.

[15] L. C. Huang, J. Oliger, W. P. Tang, and Y. D. Wu, *Toward efficient and robust finite difference schemes for unsteady incompressible Navier-Stokes equations—on the half staggered mesh*, in the Sixth International Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, 1995, pp. 467–472.

[16] L. C. Huang and Y. D. Wu, *The component-consistent pressure correction projection method for the incompressible Navier-Stokes Equations*, Comput. Math. Appl., 31 (1996), pp. 1–21.

[17] P. Koumoutsakos and A. Leonard, *High-resolution simulation of the flow around an impulsively started cylinder using vortex methods*, J. Fluid Mech., 296 (1995), pp. 1–38.

[18] E. Ng, *On One-Way Dissection Schemes*, Master's thesis, University of Waterloo, Waterloo, Ontario, Canada, 1979.

[19] E. G. Puckett, A. S. Almgren, J. B. Bell et al., *A high-order projection method for tracking fluid interfaces in variable density incompressible flows*, J. Comput. Phys., 130 (1997), pp. 269–282.

[20] M. Rosenfeld, D. Kwak, and M. Vinokur, *A fractional step solution for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems*, J. Comput. Phys., 94 (1991), pp. 102–137.

[21] E. M. Saiki and S. Biringen, *Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method*, J. Comput. Phys., 123 (1996), pp. 450–465.

[22] J. van Kan, *A second-order accurate pressure-correction scheme for viscous incompressible flow*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 870–891.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 772–792

SMOOTHNESS AND PERIODICITY OF SOME MATRIX DECOMPOSITIONS*

JANN-LONG CHERN† AND LUCA DIECI‡

Abstract. In this work we consider smooth orthonormal factorizations of smooth matrix-valued functions of constant rank. In particular, we look at Schur, singular value, and related decompositions. Furthermore, we consider the case in which the functions are periodic and study periodicity of the factors. We allow for eigenvalues and singular values to coalesce.

Key words. constant rank, orthonormal factorizations, periodic matrices

AMS subject classifications. 15A, 65F, 65L

PII. S0895479899353622

1. Introduction.

In the recent paper [6], Dieci and Eirola considered smooth orthonormal factorizations of smooth time-dependent matrix-valued functions. The purpose of the present work is to further the study of [6] in two distinct directions:

(i) extend some of the results of [6] to the case in which the function to be factored has constant rank;

(ii) consider the case in which the function is periodic and study periodicity of the factors.

Thus, we consider a k times differentiable matrix-valued function of real variable $t \to A(t)$ and write $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $k \geq 0$, where we have $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$. We can think of t as *time*, and in practice t may belong to an interval (open or close), or the half line, rather than the whole real line, but this has no bearing on our results. If all entries of A are periodic of period τ, then we will write $A \in \mathcal{C}^k_\tau(\mathbb{R}, \mathbb{F}^{m\times n})$. We call $Q \in \mathbb{F}^{m\times n}$ *orthonormal* if $Q^*Q = I$; in case $m = n$, we call Q *unitary* if $\mathbb{F} = \mathbb{C}$ and *orthogonal* if $\mathbb{F} = \mathbb{R}$. Also, in what follows, by $\Lambda(B)$ we will indicate the set of eigenvalues of the matrix B.

The study of functions with constant rank is important in applications related to differential algebraic systems; for example, see [16, 5, 15] and especially [2, sections 2.4–2.5]. Periodic matrix-valued functions arise quite often in the study of dynamical systems (e.g., see [20] and [11]), and it is clearly of interest being able to understand not just the smoothness of the factors relative to their factorizations, but also the periodicity of these factors.

Early study of both issues appear in the work of Sibuya; see [19]. Sibuya's study is about block diagonalization of matrix-valued functions with (two) disjoint groups of eigenvalues, and he studied both smoothness and periodicity of the diagonalizing transformation. Some of Sibuya's results were later somewhat improved by work of Eremenko (see [7]), but—as far as we could determine—Sibuya's periodicity results basically are still the best available. More recent study of factorization of analytic functions with constant rank is implicit in the work of Bunse-Gerstner et al. (see [3]),

*Received by the editors March 22, 1999; accepted for publication (in revised form) by A. Bunse-Gerstner June 6, 2000; published electronically October 31, 2000. This work was supported in part under NSF grants DMS-9625813 and DMS-9973266.

http://www.siam.org/journals/simax/22-3/35362.html

†Department of Mathematics, National Central University, Chung-Li 32054, Taiwan, Republic of China (chern@math.ncu.edu.tw).

‡School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (dieci@math.gatech.edu).

who considered analytic singular value decompositions (SVD) for analytic matrix functions. Some effort in the smooth, nonanalytic case for smooth SVDs of constant rank functions can be found in the Ph.D. thesis of Pütz [17], who also tackles computational issues.

Our chief contributions are twofold. On the one hand, we improve upon existing results, and give new results, about smoothness of constant rank functions, in particular for the SVD and related factorizations such as Takagi's factorization and generalized SVD. On the other hand, we give new results in the periodic case in the case of coalescing eigenvalues (singular values): we classify periodicity of the eigendecomposition in the Hermitian case and similarly for the SVD.

An outline of the paper is as follows. In the next section, we consider smooth factorizations for smooth matrix-valued functions of constant rank. Then, in section 3 we consider the periodic case. To maintain focus on these two separate issues, we found it convenient to split these two topics.

Remark 1.1. In [6], under some nondegenericity assumptions, differential equations were derived for the factors of the various decompositions examined. We have also derived differential equations for (some of) the decompositions of section 2 of the present work. However, we opted for purely algebraic proofs of the results of section 2, since the arguments used are more in tune with those adopted to prove the periodicity results of section 3 (and see also [19]), and it does not seem easy to use the differential equations to obtain the periodicity results.

2. Orthonormal decompositions in the constant rank case. The most important decomposition of this section is the SVD for functions of constant rank, Theorem 2.4, from which most other decompositions follow. In order to prove this result, we will make use of the following two lemmas, both of which are easy to prove, and the first is certainly well known (e.g., see [6]).

LEMMA 2.1 (QR decomposition). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$, $k \geq 0$, and $A(t)$ be full rank for all t. Then A admits a factorization $A(t) = Q(t)R(t)$ for all t, where $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$ is orthonormal and $R(t) \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ is upper triangular. The factorization can be made unique by requiring R to have positive diagonal entries (real even in the case $\mathbb{F} = \mathbb{C}$).*

Proof. This follows at once upon using the standard Gram–Schmidt's process on the columns of A. ∎

LEMMA 2.2 (invariant subspaces lemma). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ be Hermitian (symmetric if $\mathbb{F} = \mathbb{R}$). Let $Q_i \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n_i})$, $i = 1, \ldots, p$, be orthonormal representations of separated invariant subspaces of A; that is, for fixed p and dimensions n_i, each Q_i is orthonormal, $n_1 + \cdots + n_p \leq n$, and*

$$AQ_i = Q_i X_i, \quad \Lambda(X_i) \cap \Lambda(X_j) = \emptyset, \quad i \neq j, \ \text{for all } t. \tag{2.1}$$

Then the matrix valued function $Q := [Q_1, \ldots, Q_p]$ is orthonormal.

Proof. We need to verify that $Q_j^* Q_i = 0$ for $i \neq j$. Using (2.1) and $X_i^* = X_i$, we have

$$X_j(Q_j^* Q_i) = Q_j^* A^* Q_i = Q_j^* A Q_i = (Q_j^* Q_i) X_i.$$

From this we have the Lyapunov equation for $Q_j^* Q_i$,

$$(Q_j^* Q_i) X_i - X_j (Q_j^* Q_i) = 0,$$

which has the unique solution $Q_j^* Q_i = 0$ since $\Lambda(X_i) \cap \Lambda(X_j) = \emptyset$ for all t (see [10]). ∎

The next result was proven in [6] when $k \geq 1$; our technique here is different and is based on [19].

THEOREM 2.3. *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$, $k \geq 0$, and assume that (with p fixed) $\Lambda(A) = \Lambda_1 \cup \cdots \cup \Lambda_p$, where $\Lambda_i \cap \Lambda_j = \emptyset$ for all t and $i \neq j$, $i, j = 1, \ldots, p$. Further, in case $\mathbb{F} = \mathbb{R}$, we will assume that $\det(\Lambda_i) \in \mathbb{R}$ for all i. (This ensures that complex conjugate eigenvalues are grouped together.) Then there exists unitary (orthogonal if $\mathbb{F} = \mathbb{R}$) $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that*

$$Q^*(t)A(t)Q(t) = \begin{bmatrix} T_{11} & T_{12} & \ldots & T_{1p} \\ 0 & T_{22} & \ldots & T_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & T_{pp} \end{bmatrix}, \tag{2.2}$$

where $\Lambda(T_{ii}) = \Lambda_i$, $i = 1, \ldots, p$. Moreover, if A is normal,[1] then $T_{ij} \equiv 0$, $i \neq j$, and T_{ii}, $i = 1 \ldots, p$, are also normal.

Proof. Under the stated assumptions, [19, Theorem 3 and Remark 3] give the block diagonalization

$$S^{-1}(t)A(t)S(t) = \operatorname{diag}(E_{11}, \ldots, E_{pp}), \tag{2.3}$$

where $S \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ and the diagonal blocks E_{ii} correspond to the eigenvalues Λ_i, for $i = 1, \ldots, p$. By Lemma 2.1, we can choose $\mathcal{C}^k$ unitary Q and upper triangular R such that $S = QR$. It is now enough to block partition R according to the partitioning of (2.3) to obtain (2.2). Now consider the case of normal A. Let $T := Q^*AQ$; obviously T is normal. Now, partition T in (2.2) as $T = \left[\begin{smallmatrix} T_{11} & C \\ 0 & \hat{T} \end{smallmatrix}\right]$ and use $TT^* = T^*T$ to obtain

$$C^*T_{11} - \hat{T}C^* = 0 \quad \text{for all } t,$$

whose only solution is $C = 0$ (since $\Lambda(T_{11}) \cap \Lambda(\hat{T}) = \emptyset$ for all t). An obvious induction argument completes the proof. ∎

We are now ready for the SVD of a constant rank matrix-valued function.

THEOREM 2.4 (SVD). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$ and $k \geq 0$, have constant rank: $\operatorname{rank}(A(t)) = n - r$ for all t, r fixed: $0 \leq r \leq n - 1$. Then there exist unitary (orthogonal if $\mathbb{F} = \mathbb{R}$) $U \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$ and $V \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that*

$$U^*(t)A(t)V(t) = S = \begin{bmatrix} S_+ & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.4}$$

where $S_+ \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$ is Hermitian (symmetric if $\mathbb{F} = \mathbb{R}$) positive definite.

Further, suppose that the continuous eigenvalues of S_+ (i.e., singular values of A), $\lambda_1, \ldots, \lambda_{n-r}$, satisfy

$$\liminf_{\tau\to 0} \frac{|\lambda_i(t+\tau) - \lambda_j(t+\tau)|}{|\tau^e|} \in (0, \infty] \tag{2.5}$$

for some $e \leq k$ and for all t and $i \neq j$. Then there exists unitary (orthogonal if $\mathbb{F} = \mathbb{R}$) $Q \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$ such that $Q^*S_+Q = \operatorname{diag}(\lambda_1, \ldots, \lambda_{n-r})$. The singular values can be taken to be C^k functions.

Proof. Consider the following Hermitian function:

$$B(t) = \begin{bmatrix} 0 & A(t) \\ A^*(t) & 0 \end{bmatrix}. \tag{2.6}$$

[1] A is *normal* if $A^*A = AA^*$.

Then $B \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(m+n)\times(m+n)})$, and it is easily verified that

$$\text{if } \lambda(t) \in \Lambda(B(t)) \Rightarrow -\lambda(t) \in \Lambda(B(t)) \text{ for all } t. \tag{2.7}$$

We also have from (2.6)

$$\operatorname{rank} B(t) = 2 \operatorname{rank} A(t) = 2(n-r) \text{ for all } t. \tag{2.8}$$

Moreover, from Theorem 2.3, there exists unitary $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(m+n)\times(m+n)})$ such that

$$Q^*(t)B(t)Q(t) = \begin{bmatrix} S_+(t) & 0 & 0 \\ 0 & S_-(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.9}$$

where $S_+, S_- \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$ are Hermitian, and S_+/S_- comprise all the positive/negative eigenvalues, respectively (S_+/S_- are positive/negative definite). Column partition Q according to (2.9), $Q(t) = \begin{bmatrix} Q_1(t) & Q_2(t) & Q_3(t) \end{bmatrix}$, and let

$$W_1(t) = Q_1(t) = \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix}, \qquad W_2(t) = \begin{bmatrix} X(t) \\ -Y(t) \end{bmatrix},$$

where $X \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times(n-r)})$, $Y \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times(n-r)})$, and $W_1, W_2 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(m+n)\times(n-r)})$ are orthonormal. Let

$$W(t) = \begin{bmatrix} W_1(t) & W_2(t) & Q_3(t) \end{bmatrix}, \tag{2.10}$$

so that

$$BW_1 = W_1 S_+, \ BW_2 = W_2(-S_+), \ BQ_3 = Q_3 0.$$

Upon using Lemma 2.2, we see that $W(t)$ is unitary for all t and

$$W^*(t)B(t)W(t) = \begin{bmatrix} S_+(t) & 0 & 0 \\ 0 & -S_+(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.11}$$

Now, we set

$$U_1(t) = \sqrt{2}X(t), \qquad V_1(t) = \sqrt{2}Y(t), \tag{2.12}$$

so that $U_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times(n-r)})$ and $V_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times(n-r)})$ are orthonormal and

$$A(t)V_1(t) = U_1(t)S_+(t) \text{ and } A^*(t)U_1(t) = V_1(t)S_+(t) \text{ for all } t. \tag{2.13}$$

To complete the proof of (2.4), we need to get smooth orthonormal representations for the kernel of the row and column space of A. We proceed as follows. Since $A^*A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$, $AA^* \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$ are both of rank $(n-r)$, by Theorem 2.3 there exist unitary (orthogonal if $\mathbb{F} = \mathbb{R}$) $Q_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ and $Q_2 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$ such that

$$Q_1^*(t)A^*(t)A(t)Q_1(t) = \begin{bmatrix} M_1(t) & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_2^*(t)A(t)A^*(t)Q_2(t) = \begin{bmatrix} M_2(t) & 0 \\ 0 & 0 \end{bmatrix},$$

with $M_1,\ M_2 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$, Hermitian and nonsingular. Partition

$$Q_1(t) = \begin{bmatrix} Q_{11}(t) & V_2(t) \end{bmatrix}, \quad Q_2(t) = \begin{bmatrix} Q_{22}(t) & U_2(t) \end{bmatrix},$$

so that we have

$$(U_2^*(t)A(t))\ (U_2^*(t)A(t))^* = 0, \quad (A(t)V_2(t))^*\ (A(t)V_2(t)) = 0,$$

and thus

$$U_2^*(t)A(t) = 0, \quad A(t)V_2(t) = 0 \text{ for all } t. \tag{2.14}$$

From (2.13) and (2.14) we have

$$\begin{aligned} &(A(t)A^*(t))U_1(t) = U_1(t)S_+^2(t), \quad (A(t)A^*(t))U_2(t) = 0, \\ &(A^*(t)A(t))V_1(t) = V_1(t)S_+^2(t), \quad (A^*(t)A(t))V_2(t) = 0. \end{aligned} \tag{2.15}$$

Finally, let

$$U(t) = \begin{bmatrix} U_1(t) & U_2(t) \end{bmatrix}, \ V(t) = \begin{bmatrix} V_1(t) & V_2(t) \end{bmatrix}. \tag{2.16}$$

Since $0 \notin \Lambda(S_+(t))$, from (2.15) and Lemma 2.2, $U(t)$ and $V(t)$ are unitary (orthogonal), and we obtain the desired result (2.4).

The second part of the theorem is a direct application of [6, Theorems 3.3 and 3.5]. ∎

The next two factorizations are straightforward applications of Theorem 2.4.

COROLLARY 2.5 (complete QR). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$ and $k \geq 0$, be of constant rank:* $\operatorname{rank}(A) = n - r$ *for all t, and fixed r: $0 \leq r \leq n-1$. Then there exist unitary $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$ and $V \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that*

$$A(t)V(t) = Q(t)R(t) \quad \textit{for all } t, \tag{2.17}$$

and $R \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ is of the form

$$R(t) = \begin{bmatrix} R_1(t) & 0 \\ 0 & 0 \end{bmatrix} \tag{2.18}$$

with $R_1(t) \in \mathbb{F}^{(n-r)\times(n-r)}$ upper triangular and full rank for all t.

Proof. Use Theorem 2.4 to get $U^*AV = \left[\begin{smallmatrix} S_+ & 0 \\ 0 & 0 \end{smallmatrix}\right]$, and then use Lemma 2.1 to get $S_+(t) = Q_1(t)R_1(t)$. Finally, let

$$Q(t) = U(t) \begin{bmatrix} Q_1(t) & 0 \\ 0 & I_{r\times r} \end{bmatrix}, \ R(t) = \begin{bmatrix} R_1(t) & 0 \\ 0 & 0 \end{bmatrix}. \qquad \blacksquare$$

COROLLARY 2.6 (polar factorization). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$ and $r \geq 0$, be of constant rank:* $\operatorname{rank}(A) = n - r$ *for all t, and fixed r: $0 \leq r \leq n-1$. Then there exist orthonormal $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$ and Hermitian (symmetric) positive semidefinite $P \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that*

$$A(t) = Q(t)P(t) \quad \textit{for all } t. \tag{2.19}$$

Proof. From Theorem 2.4 we have $A(t) = U_1(t)S_1(t)V^*(t)$ with $S_1 = \left[\begin{smallmatrix} S_+ & 0 \\ 0 & 0 \end{smallmatrix}\right] \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ and S_+ positive definite for all t. Here, U_1 are the first n columns of

U and S_1 are the first n rows of S in Theorem 2.4. Thus, it is enough to rewrite $A(t) = (U_1(t)V^*(t))(V(t)S_1(t)V^*(t)) = Q(t)P(t)$. ∎

The next factorization is encountered in a number of applications; see [1, 12, 18] and see [4] for numerical study. The smoothness of its factors is proved similarly to how we proved Theorem 2.4.

THEOREM 2.7 (Takagi's factorization). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times n})$ be a complex symmetric matrix valued function (i.e., $A^T = A$) of constant rank:* $\operatorname{rank}(A(t)) \equiv n - r$ *for all t for fixed r: $0 \le r \le n-1$. Then there exists unitary $U \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times n})$ such that*

$$A(t) = U(t) \begin{bmatrix} S_+ & 0 \\ 0 & 0 \end{bmatrix} U^T(t) \quad \text{for all } t, \tag{2.20}$$

and $S_+ \in \mathcal{C}^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$ is symmetric positive definite.

Moreover, suppose that the continuous eigenvalues of S_+, $\lambda_1, \dots, \lambda_{n-r}$, satisfy (2.5) *for some $e \le k$ and for all t and $i \ne j$. Then there exists orthogonal $Q \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$ such that $Q^T S_+ Q = \operatorname{diag}(\lambda_1, \dots, \lambda_{n-r})$. The eigenvalues can be taken to be C^k functions.*

Proof. If A is complex symmetric, then $A(t) = B(t) + iC(t)$, where $B, C \in \mathcal{C}^k(\mathbb{R}, \mathbb{R}^{n\times n})$ are symmetric. Consider the symmetric function $M \in \mathcal{C}^k(\mathbb{R}, \mathbb{R}^{2n\times 2n})$,

$$M(t) = \begin{bmatrix} B(t) & C(t) \\ C(t) & -B(t) \end{bmatrix}, \tag{2.21}$$

and notice that we have $\operatorname{rank}(M) = 2(n-r)$ for all t; this fact follows from

$$(1/2) \left[\begin{smallmatrix} I_n & -iI_n \\ -iI_n & I_n \end{smallmatrix}\right] M^*(t) M(t) \left[\begin{smallmatrix} I_n & -iI_n \\ -iI_n & I_n \end{smallmatrix}\right]^* = \left[\begin{smallmatrix} A^*(t)A(t) & 0 \\ 0 & A(t)A^*(t) \end{smallmatrix}\right].$$

Thus, similarly to the proof of Theorem 2.4, we now obtain the $\mathcal{C}^k$ block Schur decomposition of M:

$$W^T(t) M(t) W(t) = \begin{bmatrix} S_+(t) & 0 & 0 \\ 0 & -S_+(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.22}$$

where $S_+ \in \mathbb{R}^{(n-r)\times(n-r)}$ is symmetric positive definite, and W is orthogonal of the form $W = \begin{bmatrix} W_1 & W_2 & Q_3 \end{bmatrix}$ with $W_1 = \left[\begin{smallmatrix} X \\ -Y \end{smallmatrix}\right]$ and $W_2 = \left[\begin{smallmatrix} Y \\ X \end{smallmatrix}\right]$. Now let

$$U_1(t) = X(t) - iY(t) \text{ for all } t, \tag{2.23}$$

so that $U_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times(n-r)})$ is orthonormal and

$$A(t) = U_1(t) S_+(t) U_1^T(t). \tag{2.24}$$

Next, let $P_1(t) = U_1(t)U_1^*(t)$. Then $\operatorname{rank}(P_1(t)) = n - r$ for all t, and from Theorem 2.3 there exists unitary $V \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times n})$ such that $V^*(t)P_1(t)V(t) = \left[\begin{smallmatrix} P_{11}(t) & 0 \\ 0 & 0 \end{smallmatrix}\right]$, with $P_{11} \in \mathbb{C}^{(n-r)\times(n-r)}$, Hermitian. Write $V(t) = \begin{bmatrix} V_1(t) & V_2(t) \end{bmatrix}$, and let

$$U(t) = \begin{bmatrix} U_1(t) & V_2(t) \end{bmatrix}. \tag{2.25}$$

Then, since $V_2^* P_1 V_2 = (U_1^* V_2)^*(U_1^* V_2) = 0$, we get $V_2^* U_1 = 0$, and so $U(t)$ is unitary. Moreover, trivially

$$A(t) = U(t) \begin{bmatrix} S_+(t) & 0 \\ 0 & 0 \end{bmatrix} U^T(t),$$

and (2.20) follows.

Finally, the statement about being able to eigendecompose S_+ under the assumption (2.5) is again a direct consequence of [6, Theorems 3.3 and 3.5]. ∎

We complete this section with a result on smoothness of the generalized SVD, which unfolds nicely as a consequence of things we proved earlier in this section. We first need the following elementary lemma.

LEMMA 2.8 (smooth Choleski factorization). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ and $A(t)$ be a Hermitian (symmetric) positive definite function for all t. Then there exists a unique lower triangular function $G \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$, with positive diagonal entries, such that*

$$A(t) = G(t)G^*(t) \quad \text{for all } t. \tag{2.26}$$

Proof. Write $A = \begin{bmatrix} a_{11} & b^* \\ b & \hat{A} \end{bmatrix}$. Let $G_1 = \begin{bmatrix} \sqrt{a_{11}} & 0 \\ b/\sqrt{a_{11}} & I_{n-1} \end{bmatrix}$. Clearly, $G_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$, and $G_1^{-1} A G_1^{-*} = \begin{bmatrix} 1 & 0 \\ 0 & A_1 \end{bmatrix}$, with $A_1 = \hat{A} - bb^*/a_{11}$. Obviously, $A_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-1)\times(n-1)})$ and is positive definite, so repeating this procedure gives the result. ∎

THEOREM 2.9 (generalized SVD). *Let $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$, be a constant rank function:* $\operatorname{rank}(A) = n - r$ *for all t, $r : \ 0 \leq r \leq n-1$, and let $B \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{p\times n})$, $p \geq n$, be full rank for all t. Then there exist unitary (orthogonal) $U_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$, orthonormal $U_2 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{p\times n})$, and invertible $X \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that, for all t,*

$$U_1^*(t)A(t)X(t) = \begin{bmatrix} S_A(t) & 0 \\ 0 & 0 \end{bmatrix}, \quad U_2^*(t)B(t)X(t) = \begin{bmatrix} S_B(t) & 0 \\ 0 & I_r \end{bmatrix}, \tag{2.27}$$

where $S_A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$, $S_B \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$, S_A is Hermitian positive definite, and

$$S_A^*(t)S_A(t) + S_B^*(t)S_B(t) = I \quad \text{for all } t.$$

Proof. Clearly, $\operatorname{rank}\begin{bmatrix} A(t) \\ B(t) \end{bmatrix} = n$ for all t, and hence from Lemma 2.1 there exist orthonormal $Q \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(m+p)\times n})$ and nonsingular upper triangular $R \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that $\begin{bmatrix} A(t) \\ B(t) \end{bmatrix} = Q(t)R(t) =: \begin{bmatrix} Q_1(t) \\ Q_2(t) \end{bmatrix} R(t)$. Here, $Q_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times n})$ has constant rank $n - r$, and Q_2 has full rank n for all t. Thus, from Theorem 2.4, there exist unitary (orthogonal) $U_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{m\times m})$ and $V \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{n\times n})$ such that

$$Q_1(t) = U_1(t) \begin{bmatrix} S_A(t) & 0 \\ 0 & 0 \end{bmatrix} V^*(t) \quad \text{for all } t,$$

and $S_A \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$ is Hermitian positive definite for all t.

Next, consider the following $\mathcal{C}^k$ orthonormal transformation

$$\begin{bmatrix} U_1^*(t) & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} Q_1(t) \\ Q_2(t) \end{bmatrix} V(t) = \begin{bmatrix} \begin{bmatrix} S_A(t) & 0 \\ 0 & 0 \end{bmatrix} \\ W(t) \end{bmatrix}, \tag{2.28}$$

where $W(t) = Q_2(t)V(t)$, and thus W has full rank for all t. Partition $W(t) =: (W_1(t) \ \ W_2(t))$, where $W_1 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{p\times(n-r)})$, $W_2 \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{p\times r})$. From (2.28),

$$S_A^*(t)S_A(t) + W_1^*(t)W_1(t) = I, \quad W_1^*(t)W_2(t) = 0, \quad \text{and} \ \ W_2^*(t)W_2(t) = I \ \ \text{for all } t.$$

Since W_1 is full rank for all t, then $W_1^*W_1$ is positive definite. So, from Lemma 2.8, we have $W_1^*W_1 = I - S_A^*S_A = GG^*$ for all t, where $G \in \mathcal{C}^k(\mathbb{R}, \mathbb{F}^{(n-r)\times(n-r)})$ is nonsingular and lower triangular. Now, we let

$$U_2(t) = W(t)\begin{bmatrix}(G^*(t))^{-1} & 0\\ 0 & I\end{bmatrix},\ S_B(t) = G^*(t) \quad \text{for all } t.$$

It is easy to check that $U_2(t)$ is orthonormal for all t and that

$$Q_2(t) = U_2(t)\begin{bmatrix}S_B(t) & 0\\ 0 & I\end{bmatrix}V^*(t).$$

So, letting $X(t) = (V^*(t)R(t))^{-1}$ for all t, the result is proved. ∎

3. Periodicity of the factors. In this section we consider the periodic case, $A \in \mathcal{C}_\tau^k(\mathbb{R}, \mathbb{F}^{m\times n})$. Without loss of generality, we will take $\tau = 1$ and assume that 1 is the minimal period of A. Also, we will henceforth assume that the function A does not have all constant eigenvalues. See Remarks 3.5 and 3.21 for what to expect in this case.

It is natural to inquire whether or not the (smooth) factors in section 2 (e.g., see Theorems 2.3 and 2.4) inherit some periodicity in case the function A is periodic of period 1. For example, can we say that Q in Theorem 2.3 has period 1? Besides being a question of theoretical interest, this inquiry has also practical implications, since it would indicate that it may be possible to compute a factorization of A over only one period. Guided by what we know from Floquet theory for differential equations and from the work of Sibuya in [19], we may expect that in the case $\mathbb{F} = \mathbb{R}$ the factors are periodic with twice the period they have in the case $\mathbb{F} = \mathbb{C}$.

We divide the results of this section in two parts. In the first part, we give somewhat "coarser" periodicity results: we look at periodicity of the factors for block eigendecompositions in the case in which the blocks correspond to disjoint groups of eigenvalues, and as a by-product we look at periodicity of the factorizations of section 2. In the second part, we look at the "finer" structure by allowing eigenvalues (singular values) to coalesce: for example, we ask ourselves about periodicity of the Schur factors in an eigendecomposition of a Hermitian matrix valued function.

3.1. Periodicity of block decompositions. The first group of results are lumped together in Theorem 3.3 below. The statements there are immediate consequences of similar results of section 2 and of Lemma 3.1 and Theorem 3.2, which extend Lemmas 2.1 and 2.8 and Theorem 2.3, to the periodic case.

LEMMA 3.1 (periodicity of QR and Choleski decompositions). *We have the following.*

- *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{m\times n})$, where $m \geq n$, $k \geq 0$, and let $A(t)$ be full rank for all t. Then A admits a unique factorization $A(t) = Q(t)R(t)$ for all t, where $Q \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{m\times n})$ is orthonormal and $R \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{n\times n})$ is upper triangular with positive diagonal entries (real even in the case $\mathbb{F} = \mathbb{C}$).*
- *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{n\times n})$ and A be Hermitian positive definite. Then there exists a unique lower triangular function G with positive diagonal entries, $G \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{n\times n})$, such that $A(t) = G(t)G^*(t)$ for all t.*

Proof. Both for $\mathbb{F} = \mathbb{C}$ and $\mathbb{F} = \mathbb{R}$, the stated results about periodicity follow immediately from: (i) the Gram–Schmidt's process on the columns of A, for the QR factorization, and (ii) the procedure of the proof of Lemma 2.8, for the Choleski factorization. ∎

THEOREM 3.2 (Sibuya's result and block Schur form). *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ with $k \geq 0$. Assume that $\Lambda(A) = \Lambda_1 \cup \cdots \cup \Lambda_l$, where $\Lambda_i \cap \Lambda_j = \emptyset$ for all t and $i \neq j$, $i, j = 1, \ldots, l$. Then there exist invertible $S \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ and unitary $Q \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ such that*

$$S^{-1}(t)A(t)S(t) = \begin{bmatrix} E_{11} & 0 & \ldots & 0 \\ 0 & E_{22} & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & E_{ll} \end{bmatrix} \equiv E(t), \tag{3.1}$$

$$Q^*(t)A(t)Q(t) = \begin{bmatrix} T_{11} & T_{12} & \ldots & T_{1l} \\ 0 & T_{22} & \ldots & T_{2l} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & T_{ll} \end{bmatrix} \equiv T(t)\,, \tag{3.2}$$

where each $T_{ij} \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n_i\times n_j})$ and $E_{ii} \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n_i\times n_i})$ with $\Lambda(E_{ii}) = \Lambda(T_{ii}) = \Lambda_i$, $i = 1, \ldots, l$, and $n_1 + \cdots + n_l = n$.

In case $\mathbb{F} = \mathbb{R}$, assume that $\det(\Lambda_i) \in \mathbb{R}$ for all i. Then the previous statements—in particular, (3.1) and (3.2)—are true for invertible $S \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$ and orthogonal $Q \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$. Now, each $T_{ij} \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n_i\times n_j})$ and $E_{ii} \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n_i\times n_i})$.

For either $\mathbb{F} = \mathbb{C}$ or $\mathbb{F} = \mathbb{R}$, if A is normal, then in (3.2) we have $T_{ij} = 0$, $i \neq j$, and T_{ii}, $i = 1, \ldots, l$, are also normal.

Proof. The smoothness results are in Theorem 2.3. As far as the periodicity results, we recall that in [19, Theorem 3, Remark 3], Sibuya proved the result (3.1) with S of period 1 for two disjoint blocks of eigenvalues, in the case $\mathbb{F} = \mathbb{C}$. It is immediate to apply his result over and over to obtain (3.1) with S of period 1 for p disjoint blocks of eigenvalues. Application of Lemma 3.1 yields $S = QR$ and thus (3.2). In the real case, $\mathbb{F} = \mathbb{R}$, Sibuya (see [19, Remark 1]) gives (3.1) with S of period 2 for two disjoint blocks of eigenvalues. One cannot simply apply this result over and over now, since this would increase the period of S. However, the arguments in Sibuya's proofs can be readily generalized to dealing with p blocks of eigenvalues simultaneously. So doing, one obtains $S \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$ and then trivially $Q \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$ in (3.2) applying Lemma 3.1 to get $S = QR$. The statement in the normal case is proved as in the last part of the proof of Theorem 2.3. ∎

Statements (i)–(v) in Theorem 3.3 below are extension to the periodic case of Theorem 2.4, Corollaries 2.5 and 2.6, and Theorems 2.7 and 2.9, respectively.

THEOREM 3.3. *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times n})$, respectively, $\mathbb{R}^{m\times n}$, $m \geq n$ and $k \geq 0$, be of constant rank: $\operatorname{rank}(A) = n - r$ for all t and fixed r: $0 \leq r \leq n - 1$.*

(i) *There exist unitary $U \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times m})$ and $V \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, respectively, orthogonal $U \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{m\times m})$ and $V \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$, such that (2.4) holds, where $S_+ \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{(n-r)\times(n-r)})$, respectively, $S_+ \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$, is Hermitian, respectively, symmetric, positive definite.*

(ii) *There exists unitary $Q \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times m})$ and $V \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, respectively, orthogonal $Q \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{m\times m})$ and $V \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$, such that (2.17) and (2.18) hold with $R_1 \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{(n-r)\times(n-r)})$, respectively, $R_1 \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$.*

(iii) *There exist unitary $Q \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times n})$, respectively, orthogonal $Q \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{m\times n})$, and Hermitian (symmetric) positive semidefinite $P \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, respectively, $P \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$, such that (2.19) holds.*

Now, let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ be complex symmetric with constant rank $n - r$, where $0 \leq r \leq n - 1$. Then,

(iv) *there exists unitary $U \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ and symmetric positive definite $S_+ \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$ such that (2.20) holds.*

Finally, let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{m\times n})$, $m \geq n$, *be of constant rank* $n - r$, *and let* $B \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{F}^{p\times n})$, $p \geq n$, *be full rank for all* t. *Then,*

(v) *there exist unitary* $U_1 \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times m})$ *and orthonormal* $U_2 \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{p\times n})$, *respectively, orthogonal* $U_1 \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{m\times m})$ *and* $U_2 \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{p\times n})$, *and invertible* $X \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, *respectively,* $X \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$, *such that* (2.27) *holds.*

We are now ready to provide sharper results in a number of cases. First, let us begin remarking that, as a consequence of Theorem 3.2, a function $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$ with distinct eigenvalues is diagonalizable with a $\mathcal{C}_1^k$ function of eigenvectors (similarly, it has a $\mathcal{C}_1^k$ Schur decomposition). This is because the diagonal blocks E_{ii} and T_{ii} in (3.1) and (3.2) are of period 1 when $\mathbb{F} = \mathbb{C}$. However, in the real case $\mathbb{F} = \mathbb{R}$, it appears that the blocks E_{ii} (and T_{ii}) are of period 2, since S and Q have period 2 in this case. Our next task is to show that, even in the real case, E_{ii} can be chosen of period 1, when A has only simple eigenvalues (real or complex conjugate), and that also T_{ii} can be chosen of period 1 in this case, when A is normal.

THEOREM 3.4. *Let* $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{n\times n})$, $k \geq 0$. *Suppose that* $A(t)$ *has only simple (real or complex conjugate) eigenvalues for all* t. *Then, in* (3.1), *we can choose* $S \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$ *such that* $E_{ii} \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{n_i\times n_i})$ *(here,* $n_i = 1$ *or* 2*). Further, if* A *is normal the conclusions remain true with* S *orthogonal, i.e., in* (3.2) *we have* $T_{ii} \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{n_i\times n_i})$.

Proof. Clearly, if A has only real distinct eigenvalues, then they have period 1, because they have period 1 seen as resulting from a complex similarity transformation. Next, consider the case of a complex conjugate pair, and write $E_{ii}(t) = \left[\begin{smallmatrix} b_{11}(t) & b_{12}(t) \\ b_{21}(t) & b_{22}(t) \end{smallmatrix}\right]$, where each b_{ij} is a $\mathcal{C}_2^k$ function. The eigenvalues of $E_{ii}(t)$ are λ and $\bar{\lambda}$, and they are both $\mathcal{C}_1^k$ functions (since they are simple). Since E_{ii} is real, and the eigenvalues of A are distinct, then $b_{12}(t) \neq 0$ for all t. Now, let $x(t) = \frac{b_{22}(t)-b_{11}(t)}{2b_{12}(t)}$, $a(t) = \frac{\mathrm{Im}(\lambda(t))}{b_{12}(t)}$ and $S_1(t) = \left[\begin{smallmatrix} 1 & 0 \\ x(t) & 1 \end{smallmatrix}\right]$, $S_2(t) = \left[\begin{smallmatrix} 1/a(t) & 0 \\ 0 & 1 \end{smallmatrix}\right]$, so that $S_i \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{2\times 2})$, $i = 1, 2$. We have

$$(S_1(t)S_2(t))^{-1} E_{ii}(t) (S_1(t)S_2(t)) = \left[\begin{smallmatrix} \mathrm{Re}(\lambda(t)) & \mathrm{Im}(\lambda(t)) \\ -\mathrm{Im}(\lambda(t)) & \mathrm{Re}(\lambda(t)) \end{smallmatrix}\right] \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{2\times 2}).$$

Therefore, we can choose $E_{ii} \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{2\times 2})$ even though $S \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times n})$. The statement in the case of A normal is an immediate consequence of the fact that a (2×2) nonsymmetric normal matrix has the form $\left[\begin{smallmatrix} \mathrm{Re}(\lambda(t)) & \mathrm{Im}(\lambda(t)) \\ -\mathrm{Im}(\lambda(t)) & \mathrm{Re}(\lambda(t)) \end{smallmatrix}\right]$. ∎

Remark 3.5. Of course, if $A \in \mathcal{C}_1^k$ with all eigenvalues constant, then Theorem 3.2 still holds. In this case, if all eigenvalues are distinct, E_{ii} and T_{ii} are constant.

3.2. Periodicity in coalescing case. Our next task is to characterize the periodicity of the eigenvalues (singular values), and of the corresponding orthonormal factors, in the case in which eigenvalues are allowed to coalesce. We will restrict to the Hermitian case. The first step will be to establish periodicity of the eigenvalues, and then we will determine the periodicity of the unitary transformation.

So, we have a function $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, $A = A^*$, or also A analytic: $A \in \mathcal{C}_1^\omega$, and we want to establish the periodicity of its eigenvalues. This seemingly simple problem is rather complex, since there is a very delicate interplay between smoothness of the eigenvalues and their periodicity. Let us begin with the following observation.

- The eigenvalues of A are roots of the characteristic polynomial of A, call it $\pi(\lambda, t)$, and clearly $\pi(\lambda, t+1) = \pi(\lambda, t)$. Therefore, we can certainly label the eigenvalues of A so that they are periodic functions of period 1. Since the eigenvalues are continuous functions, we thus immediately have that there

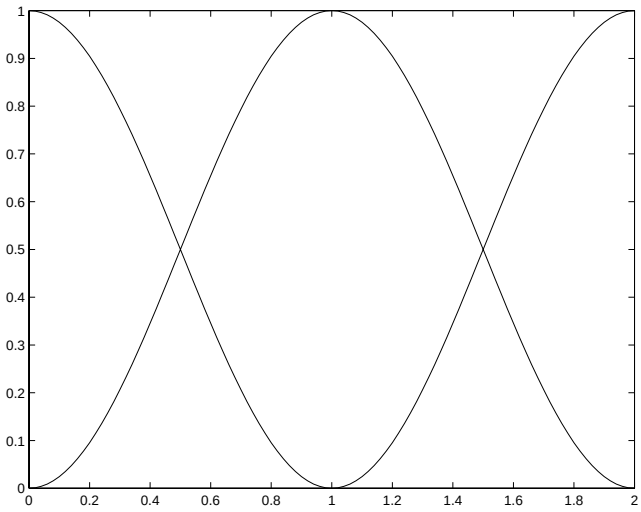


FIG. 1.

exists an ordering of the eigenvalues so that they are (at least) $\mathcal{C}_1^0$ functions. Now, if the eigenvalues are simple, then we know that they can be taken as C_1^k functions (this follows from Theorem 3.2). However, if two eigenvalues coalesce and we want them to be periodic functions of period 1, then in general we may have to settle for $\mathcal{C}^0$ eigenvalues. Alternatively, we must be willing to increase the period in order to retain smoothness. This fundamental conflict between having period 1 and/or maximal possible smoothness is already present in the analytic case. Recall that if $A = A^* \in \mathcal{C}^\omega$, then it has an analytic diagonalization $A = QDQ^*$ regardless of whether or not the eigenvalues coalesce; see [14].

Example 3.6. Consider the symmetric function $A(t) = \begin{bmatrix} 1 - \frac{1}{2}\sin^2 \pi t & -\frac{1}{4}\sin 2\pi t \\ -\frac{1}{4}\sin 2\pi t & \frac{1}{2}\sin^2 \pi t \end{bmatrix}$, $t \in \mathbb{R}$. Clearly, $A \in \mathcal{C}_1^\omega$, and thus we know that it has analytic eigenvalues (and eigenvectors). The analytic eigenvalues are $\lambda_1(t) = \frac{1+\cos \pi t}{2}$, and $\lambda_2(t) = \frac{1-\cos \pi t}{2}$, and we notice that $\lambda_1 = \lambda_2$ at $t = 1/2,\ 3/2, \ldots$. The associated orthogonal function of eigenvectors is $Q = \begin{bmatrix} \cos \frac{\pi}{2}t & \sin \frac{\pi}{2}t \\ -\sin \frac{\pi}{2}t & \cos \frac{\pi}{2}t \end{bmatrix}$, so that $Q^T(t)A(t)Q(t) = \begin{bmatrix} \lambda_1(t) & 0 \\ 0 & \lambda_2(t) \end{bmatrix}$. Notice that $\lambda_{1,2} \in \mathcal{C}_2^\omega$ and $Q \in \mathcal{C}_4^\omega$; in other words, we retained analyticity of the eigenvalues, but we have doubled their period (with respect to that of A) and then doubled again the period of Q. Of course, we could have chosen the eigenvalues to be merely continuous, and of period 1, as follows:

$$\tilde{\lambda}_1(t) = \begin{cases} \frac{1+\cos \pi t}{2} & \text{if } \ 0 \le t \le 1/2 \text{ or } 3/2 \le t \le 2, \\ \frac{1-\cos \pi t}{2} & \text{if } \ 1/2 \le t \le 3/2, \end{cases}$$

$$\tilde{\lambda}_2(t) = \begin{cases} \frac{1-\cos \pi t}{2} & \text{if } \ 0 \le t \le 1/2 \text{ or } 3/2 \le t \le 2, \\ \frac{1+\cos \pi t}{2} & \text{if } \ 1/2 \le t \le 3/2. \end{cases}$$

Figure 1 summarizes the situation: the curves on the top, bottom, of the value 1/2 give $\tilde{\lambda}_{1,2}$.

Example 3.7. For all t, take $A(t) = Q(t)D(t)Q^T(t)$, with $D(t) = \mathrm{diag}(\lambda_1(t), \lambda_2(t))$, $\lambda_1(t) = \cos^2 \pi t$, $\lambda_2(t) = \frac{1}{2}\sin^2 \pi t$, and $Q = \begin{bmatrix} \cos \pi t & \sin \pi t \\ -\sin \pi t & \cos \pi t \end{bmatrix}$. Easily, $A,\ D \in \mathcal{C}_1^\omega$, while Q is analytic of period 2. Figure 2 exemplifies the situation: even though the functions

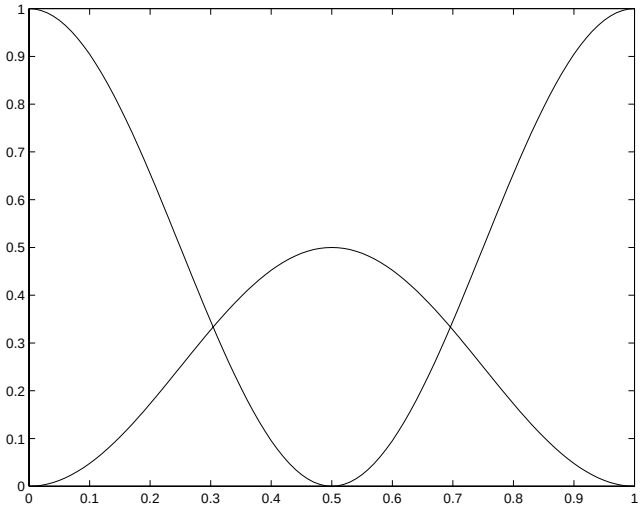


FIG. 2.

$\lambda_{1,2}$ intersect, after 1 unit of time they are back in their original position.

Based upon Examples 3.6 and 3.7, we conjecture that the periodicity of the eigenvalues depend on the relative positioning of coalescing eigenvalues after one period. This conjecture turns out to be essentially correct, as we set out to prove.

Consider the general case $A \in \mathcal{C}_1^k$ or $A \in \mathcal{C}_1^\omega$; for short, we will write this as $A \in \mathcal{C}_1^{k,\omega}$. Of course, we can (and will) think that the problem has been reduced to a block diagonal form as in (2.2), with the T_{ii} being associated to disjoint groups of eigenvalues. To be precise, we will assume that the T_{ii} satisfy Assumptions 3.8(i–ii) below.

ASSUMPTION 3.8. *Let $B = B^*$, $B \in \mathcal{C}_1^{k,\omega}$ be such that*

(i) *"B's eigenstructure is* as fine as possible*." By this we mean that there do not exist $\Lambda^{(1)}$ and $\Lambda^{(2)}$, both nonempty, such that $\Lambda^{(1)} \cap \Lambda^{(2)} = \emptyset$ and $\Lambda^{(1)} \cup \Lambda^{(2)} = \Lambda(B)$ for all t.*

(ii) *"If $B \notin \mathcal{C}^\omega$, its eigenvalues satisfy condition* (2.5)*."*

Assumption 3.8(ii) relative to each T_{ii} in (2.2) implies (see [6]) that each T_{ii} can be diagonalized with unitary (orthogonal) $U_{ii} \in \mathcal{C}^{k-e_i}(\mathbb{R}, \mathbb{F}^{n_i \times n_i})$. Therefore, if we let $e = \max e_i$, $i = 1, \ldots, l$, in the $\mathcal{C}^k$ case, there exists a unitary matrix $Q \in \mathcal{C}^{k-e,\omega}(\mathbb{R}, \mathbb{F}^{n \times n})$ which diagonalizes A:

$$Q^*AQ =: D = \operatorname{diag}(D_1, D_2, \ldots, D_l), \qquad D_i = \operatorname{diag}(\lambda_j^{(i)},\ j = 1, \ldots, n_i). \tag{3.3}$$

Our goal is to establish the periodicity of D, and the first step will be to establish the periodicity of the D_i's. We need the following definition.

DEFINITION 3.9. *Given an (integer) partition of n: $n_1, \ldots, n_l$ so that $n_1 + \cdots + n_l = n$, let $\mu(n) := \operatorname{lcm}(n_1, \ldots, n_l)$* (lcm *is the least common multiple*)*. For given value of n, let $\mu^*(n)$ be the maximum value of $\mu(n)$ over all partitions of n.*

THEOREM 3.10 (periods of eigenvalues). *Let $A \in \mathcal{C}_1^{k,\omega}(\mathbb{R}, \mathbb{F}^{n \times n})$, $A = A^*$, with eigenvalues not all constant. Let the function A be in the form* (2.2)*, where the diagonal blocks satisfy Assumption* 3.8(i)–(ii)*. Then, with the notation of* (3.3)*, the*

functions $D_i \in \mathcal{C}^{k,\omega}$ can be taken periodic of periods p_i, with $1 \le p_i \le n_i$.[2] *If $D_i = \lambda_i I$, λ_i constant, then we can take $p_i = 1$. The diagonal function of eigenvalues, D, can be taken periodic of period p, the least common multiple of the D_i's periods, $1 \le p \le \mu^*(n)$.*

Proof. The result follows from Lemma 3.14 below. ∎

Remark 3.11. It should be appreciated that the function $\mu^* : \mathbb{N} \to \mathbb{N}$ grows very rapidly. Although a closed formula for the exact values of $\mu^*(n)$ appears hard to obtain,[3] it is easy to obtain the upper bound

$$\log(\mu^*(n)) \le \frac{n}{e}. \tag{3.4}$$

The verification of (3.4) is simple, since

$$\operatorname{lcm}(n_1, n_2, \dots, n_l) \le \left(\frac{n_1 + n_2 + \cdots + n_l}{l}\right)^l = \left(\frac{n}{l}\right)^l \quad \text{for all } l,$$

and $f(x) = (\frac{n}{x})^x$ has a global maximum at $x_0 = \frac{n}{e}$. (In comparison, $\mu^*(10) = 30$ and $e^{\frac{10}{e}} \approx 40$, but already $\mu^*(16) = 140$ and $e^{\frac{16}{e}} \approx 360$.)

To prove Theorem 3.10, we need the following lemmas. We will use the concept of *irreducible* matrix (e.g., see [13]). Also, given a constant matrix B, we will call *period* of B the smallest integer $k \ge 1$, if it exists, such that $B^k = I$.

LEMMA 3.12. *If P is an (n, n) irreducible permutation matrix, then it has period n.*

Proof. From [13, p. 512], an irreducible permutation matrix P is similar, via a permutation similarity Π, to a cyclic matrix

$$\Pi P \Pi^T = \begin{bmatrix} 0 & 1 & & 0 \\ \vdots & 0 & \ddots & \\ 0 & \vdots & \ddots & 1 \\ 1 & 0 & \dots & 0 \end{bmatrix},$$

and thus $P^n = I$, but $P^k \ne I$ if $k < n$. ∎

LEMMA 3.13. *If P is an (n, n) permutation matrix, then it has the irreducible decomposition* $\operatorname{diag}(P_{ii}, i = 1, \dots, l)$*, with P_{ii} an (n_i, n_i) irreducible permutation, $i = 1, \dots, l$. The period of P is $\mu(n)$, where the partition of n is that associated to its irreducible decomposition.*

Proof. If P is irreducible, Lemma 3.12 gives the result. So, let P be reducible. Then there exists a permutation matrix Π such that $\Pi P \Pi^T = \left[\begin{smallmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{smallmatrix}\right]$. Since Π is a permutation, then P_{22} is necessarily also a permutation and hence $P_{12} = 0$. We continue this reduction process until all diagonal blocks are irreducible, and apply Lemma 3.12. ∎

LEMMA 3.14 (minimal integer period). *Under the assumptions of Theorem* 3.10*, there exists a smallest integer p such that $D(t+p) = D(t)$ for all t. We call this value p the* minimal integer period *of D, and we have $1 \le p \le \mu^*(n)$.*

Proof. First consider the $\mathcal{C}^k$ case. By assumption, the points where eigenvalues coalesce are isolated, and thus there is only a finite number of points in $[0, 1]$ where eigenvalues coalesce. Without loss of generality, let the eigenvalues be distinct at $t = 0$, and let $0 < t_1 < \cdots < t_{M+1} < 1$ be the points in $[0, 1]$ where some eigenvalues coalesce.

[2] The precise value of p_i depends on the eigenvalues' relative positions after one unit of time; see Lemma 3.14

[3] Our colleague Yang Wang communicated to us the asymptotic $\log(\mu^*(n)) = \sqrt{n \log n} + o(\sqrt{n \log n})$.

Let $I_j = (t_j, t_{j+1})$, $j = 1, \ldots, M$, and $I_0 = (t_{M+1} - 1, t_1)$, so that the eigenvalues are distinct on each of these subintervals. Let $D^{(j)}(t) = \mathrm{diag}(\lambda_{j_1}(t), \ldots, \lambda_{j_n}(t))$ be a labeling of the eigenvalues which reflects their relative ordering on each I_j: that is, for all $t \in I_j$, $\lambda_{j_1}(t) > \cdots > \lambda_{j_n}(t)$; let $D^{(0)}$ be called simply D. Let P_j be the permutation matrix defined so that

$$P_j D^{(j-1)}(t) P_j^T \;=\; D^{(j)}(t), \;\; j = 1, \ldots, M+1, \;\; t \in I_j.$$

In particular, if we take a point $\tau_0 \in I_0$, then we have

$$P_{M+1} D^{(M)}(1+\tau_0) P_{M+1}^T \;=\; D^{(M+1)}(1+\tau_0),$$

and thus also

$$\begin{aligned} D^{(M+1)}(1+\tau_0) \;&=\; P_{M+1} P_M D^{(M-1)}(1+\tau_0) P_M^T P_{M+1}^T = \cdots \\ &=\; P_{M+1} \cdots P_1 D(1+\tau_0) P_1^T \cdots P_{M+1}^T. \end{aligned}$$

Since the roots of the characteristic polynomial have period 1 (as locus of points), then $D^{(M+1)}(1+\tau_0) = D(\tau_0)$, and if we let $\Pi_1 := P_{M+1} \ldots P_1$, we then have

$$D(\tau_0) = \Pi_1 D(1+\tau_0) \Pi_1^T \qquad \text{and further} \qquad D(\tau_0) = \Pi_1^l D(l+\tau_0)(\Pi_1^T)^l,\ l = 1, 2, \ldots.$$

Now, let p_1 be the smallest integer such that $\Pi_1^{p_1} = I$. Then, we obtain that $D(t + p_1) = D(t)$ for all $t \in I_0$. In precisely the same way, we now take a point $\tau_j \in I_j$ and repeat the above reasoning to eventually obtain that there are permutation matrices $\Pi_2, \ldots, \Pi_{M+1}$ such that

$$D^{(j)}(\tau_j) = \Pi_{j+1} D^{(j)}(1+\tau_j) \Pi_{j+1}^T, \;\; j = 1, \ldots, M,$$

and hence if p_j are the periods of these Π_j, we would have that D itself has periods p_j for all $t \in I_j$. But it is a simple observation that all p_j are equal; e.g., it is immediate that $\Pi_2 = P_1 \Pi_1 P_1^T$, etc. We let p be this common value, and so we have obtained that $D(t+p) = D(t)$ for all $t \in \bigcup_j I_j$. Finally, because of continuity, the function D has period p everywhere. Applying Lemmas 3.12 and 3.13 we get the bound on $p:\ 1 \le p \le \mu^*(n)$.

In the $\mathcal{C}^\omega$ case, we have $T_{ii} \in \mathcal{C}_1^\omega$ and $D_i \in \mathcal{C}^\omega$, but we may now have that some eigenvalues are identical for all t. Let $\mu_1, \ldots, \mu_m$ ($m \le n$) be the eigenvalues of A, so that no two of them are identical for all t. Since the μ_i's are real analytic functions, then these functions must have a finite order of contact; that is, there exists an integer $e < \infty$ such that $\mu_i^{(e)}(t) \ne \mu_j^{(e)}(t)$ for all t and $i \ne j$. Thus, the points where the μ_i's coalesce are isolated. We now can repeat the reasoning of the $\mathcal{C}^k$ case relative to $M = \mathrm{diag}(\mu_1, \ldots, \mu_m)$. ∎

We also have the following lemma.

LEMMA 3.15. *Under the assumptions of Theorem* 3.10*, D cannot have irrational period.*

Proof. Suppose there exists irrational b such that $D(t+b) = D(t)$ for all t. Because of Lemma 3.14, there exists a minimal integer period p such that $D(t+p) = D(t)$. Then there exist integers k_1, k_2 such that $\lambda_j(t + k_1 b) = \lambda_j(t + k_2 p)$ for all t and $j = 1, \ldots, n$. But then all λ_j must be constant, a case which we have excluded. ∎

Up to this point, we know that the function D has minimal integer period p, $1 \le p \le \mu^*(n)$. However, we have left open the possibility for D to have *minimal*

period given by a rational number p/q, with $(p,q) = 1$ (relatively prime) and p the minimal integer period. Indeed, we now show—constructively—that there exist Hermitian functions B of period 1 with D having such values p/q as periods.

Let us begin by constructing functions $B = B^* \in \mathcal{C}_1^{k,\omega}(\mathbb{R}, \mathbb{C}^{m\times m})$ such that $Q^*(t)B(t)Q(t) = D(t)$ for all t with $D \in \mathcal{C}_{m/q}^{k,\omega}(\mathbb{R}, \mathbb{R}^{m\times m})$ diagonal, $(m,q) = 1$, and $Q \in \mathcal{C}_m^{\omega}(\mathbb{R}, \mathbb{C}^{m\times m})$ unitary.

LEMMA 3.16. *There exist $D \in \mathcal{C}_{m/q}^{k,\omega}(\mathbb{R}, \mathbb{R}^{m\times m})$ diagonal and unitary $Q \in \mathcal{C}_m^{\omega}(\mathbb{R}, \mathbb{C}^{m\times m})$ such that by letting $B(t) := Q(t)D(t)Q^*(t)$ for all t, then $B \in \mathcal{C}_1^{k,\omega}(\mathbb{R}, \mathbb{C}^{m\times m})$. Further, B satisfies Assumption* 3.8(i)–(ii).

Proof. We are going to take $D(t) = \mathrm{diag}(\lambda_1(t), \ldots, \lambda_m(t))$ of appropriate smoothness, and of period m/q, such that

$$\lambda_j(t+1) \;=\; \lambda_{j+1}(t), \;\; j = 1, \ldots, m \pmod m. \tag{3.5}$$

Specifically, for $j = 1, \ldots, m$ and for all t, we take

$$\lambda_j(t) = \begin{cases} \cos\left(\frac{2\pi}{m} q(t+j-1)\right) & \text{in the } \mathcal{C}^{\omega} \text{ case,} \\ \left|\cos^{k+1}\left(\frac{2\pi}{m} q(t+j-1)\right)\right| & \text{in the } \mathcal{C}^{k} \text{ case.} \end{cases} \tag{3.6}$$

In either case, for such D we have

$$D(t+1) \;=\; PD(t)P^T \;\text{ for all } t, \text{ where } P = \begin{bmatrix} 0 & 1 & & 0 \\ \vdots & 0 & \ddots & \\ 0 & \vdots & \ddots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{3.7}$$

Now we want to find Q of period m such that B has period 1.

CLAIM 3.17. *The following two properties hold.*

(i) *If $Q(t+1)P = Q(t)$ for all t, and P in* (3.7)*, then B has period* 1.

(ii) *If we let $\alpha = \frac{2\pi}{m}$ and $\alpha_k = k\alpha$, $k = 1, 2, \ldots, m-1$, then*

$$\sum_{j=1}^{m} e^{ij\alpha_k} \;=\; 0 \text{ for all } k = 1, \ldots, m-1.$$

Verification of Claim 3.17. To verify (i), observe that if $Q(t+1)P = Q(t)$, then $B(t+1) = Q(t+1)D(t+1)Q^*(t+1) = B(t)$ for all t. To verify (ii), we let $z = e^{i\alpha_k}$, so that $z \neq 1$, but $z^m = 1$. Thus, (ii) follows, since $0 = z^m - 1 = (z-1)(z^{m-1} + \cdots + z + 1)$. ∎

We are now ready to define Q. We take

$$Q(t) := \begin{bmatrix} q_1(t) & q_1(t+1) & \cdots & q_1(t+m-1) \end{bmatrix}, \; q_1(t) = \frac{1}{\sqrt{m}} \begin{bmatrix} e^{i\frac{2\pi}{m}t} \\ \vdots \\ e^{i\frac{(m-1)2\pi}{m}t} \\ e^{i2\pi t} \end{bmatrix}. \tag{3.8}$$

In (3.8), clearly $q_1 \in \mathcal{C}_m^{\omega}$ and so does Q, and m is the minimal period of Q. Moreover, by construction, $Q(t+1)P = Q(t)$. Direct verification, using Claim 3.17(ii), gives $Q^*(t)Q(t) = I$ for all t. Finally, using Claim 3.17(i), the lemma is proved. ∎

Remark 3.18. With the help of Lemma 3.16, we can build matrices $B \in \mathcal{C}_1^{k,\omega}(\mathbb{R}, \mathbb{C}^{m\times m})$ satisfying Assumption 3.8(i)–(ii), $B = QDQ^*$, $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, $D \in \mathcal{C}_{p/q}^{k,\omega}$, $p = 1, \ldots, m$, and $Q \in \mathcal{C}_p^{\omega}$ unitary. In fact, we may let $D = \left[\begin{smallmatrix} D_1 & 0 \\ 0 & D_2 \end{smallmatrix}\right]$ with

$D_1 = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ defined as D was in the proof of Lemma 3.16, and $D_2 = \alpha I_{m-p}$ for appropriate constant α chosen so that we satisfy Assumption 3.8(i)–(ii). Accordingly, let $Q = \left[\begin{smallmatrix} Q_1 & 0 \\ 0 & I \end{smallmatrix}\right]$ with Q_1 defined (relative to D_1) as Q was relative to D in the proof of Lemma 3.16.

Remark 3.19. As a consequence of Remark 3.18, we can justify the claim that there exist $B \in \mathcal{C}_1^{k,\omega}$, satisfying the assumptions of Theorem 3.10 with $D \in \mathcal{C}_{p/q}^{k,\omega}$ and $p:\ 1 \le p \le \mu^*(n)$. To achieve this in the $\mathcal{C}^\omega$ case, take $D = \mathrm{diag}(D_1, \ldots, D_l)$ with $D_j \in \mathcal{C}_{p_j/q_j}^{\omega}$ diagonal of dimension n_j, $1 \le p_j \le n_j$, according to Remark 3.18, and build Q in a similar block diagonal fashion. In the $\mathcal{C}^k$ case, in order to ensure that Assumption 3.8(ii) is satisfied, we may have to shift the spectra of some of the D_j appropriately.

Next, we examine the period of the eigenvectors Q in (3.3). Partition $Q = \begin{bmatrix} Q_1 & Q_2 & \ldots & Q_l \end{bmatrix}$ conformally to D's partitioning, so that we have

$$AQ_i = Q_i D_i, \ \ i = 1, \ldots, l, \qquad D_i = \mathrm{diag}(\lambda_j^{(i)}, j = 1, \ldots, n_i). \tag{3.9}$$

So far, we have established that each D_i has period p_i/q_i, where $1 \le p_i \le n_i$, $(p_i, q_i) = 1$.

The denominators q_i in the periods p_i/q_i of the functions D_i play no further role in what follows, and we will thus dispense with them, simply working with p_i-periodic D_i's. This said, it is worth stressing once more that the minimal period of D_i may be the rational number p_i/q_i; thus, also the minimal period of the (smooth) eigenvalues of A may well be a rational number p/q, $(p, q) = 1$, $1 \le p \le \mu^*(n)$. This same observation holds true also for the singular values of Theorems 3.24 and 3.27.

Remark 3.20. In [9], Gingold and Hsieh devised a Schur decomposition procedure for an analytic matrix-valued function A with real and analytic eigenvalues, which in particular is valid for a Hermitian analytic function. Then, in [9, Theorem 10.1], they noticed that if A and its analytic eigenvalues both have period 1, then their procedure will produce analytic unitary factors also of period 1. Clearly, in light of our results, one cannot generally assume that the (analytic) eigenvalues have period 1. However, we notice that it is enough to replace Gingold's and Hsieh's assumption of eigenvalues of period 1 with that of "eigenvalues' matrix D of minimal integer period p," and then the procedure of Gingold and Hsieh delivers a unitary, p-periodic, analytic, matrix-valued function of eigenvectors. The validity of our observation is immediately verified upon examining the procedure of [9].

Remark 3.21. If $A \in \mathcal{C}_1^k$ has all constant eigenvalues, then because of Assumption 3.8(ii) they must be distinct, and thus Theorem 3.2 applies. If instead $A \in \mathcal{C}_1^\omega$, then even if all eigenvalues of A are constant, and possibly many of them identical, the procedure of [9] still delivers a unitary and analytic Q of period 1.

Because of Remark 3.20, in the $\mathcal{C}^\omega$ case we can take the unitary eigenvectors $Q \in \mathcal{C}^\omega$ with period given by the $\mathrm{lcm}(p_1, \ldots, p_l)$. The next result is for the $\mathcal{C}^k$ case.

PROPOSITION 3.22. *Let $A = A^* \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$, or $A = A^T \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{n\times n})$, be such that its eigenvalues satisfy condition* (2.5). *Let $D_1, \ldots, D_l$ be diagonal matrix-valued functions, $D_i \cap D_j = \emptyset$, $i \ne j$, grouping the eigenvalues of A, as fine as possible according to Assumption* 3.8(i)*; thus, we can take $D_i \in \mathcal{C}_{p_i}^k(\mathbb{R}, \mathbb{R}^{n_i \times n_i})$, $1 \le p_i \le n_i$, $i = 1, \ldots, l$. Let unitary $Q \in \mathcal{C}^{k-e}$, $Q = \begin{bmatrix} Q_1 & Q_2 & \ldots & Q_l \end{bmatrix}$, be such that $AQ_i = Q_i D_i$, $i = 1, \ldots, l$. Then, we can take each Q_i of period p_i and hence Q of period given by* $\mathrm{lcm}(p_1, p_2, \ldots, p_l)$.

If the function A is symmetric real valued, and Q_i are real orthonormal, then each Q_i can be taken of period $2p_i$ and Q of period $2\,\mathrm{lcm}(p_1, p_2, \ldots, p_l)$.

The proof of Proposition 3.22 follows from Theorem 3.23 below and Lemma 2.2.

THEOREM 3.23. *Let $A = A^*$, $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times n})$. Suppose that $\Lambda(A) = \Lambda_1 \cup \Lambda_2$ and $\Lambda_1 \cap \Lambda_2 = \emptyset$ for all t. Let unitary $Q \in \mathcal{C}^k$ be such that $Q^*AQ = \left[\begin{smallmatrix} T_{11} & 0 \\ 0 & T_{22}\end{smallmatrix}\right]$, $\Lambda(T_{ii}) = \Lambda_i$, $i = 1, 2$, and suppose that T_{11} satisfies Assumption 3.8. With e_1 given in (2.5), let $Q_{11} \in \mathcal{C}^{k-e_1}(\mathbb{R}, \mathbb{C}^{n_1\times n_1})$ be such that $Q_{11}^* T_{11} Q_{11} = D_1 = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_1}) \in \mathcal{C}_{p_1}^k(\mathbb{R}, \mathbb{R}^{n_1\times n_1})$, $1 \le p_1 \le n_i$. Let $Q\left[\begin{smallmatrix} Q_{11} & 0 \\ 0 & I_{n-n_1}\end{smallmatrix}\right] =: \begin{bmatrix} Q_1 & Q_2\end{bmatrix}$, so that*

$$AQ_1 - Q_1 D_1 = 0 \quad \text{for all } t. \tag{3.10}$$

Then there exists orthonormal $\tilde{Q}_1 \in \mathcal{C}_{p_1}^{k-e_1}(\mathbb{R}, \mathbb{C}^{n\times m_1})$ such that (3.10) holds. If A and $\tilde{Q}_1$ are real valued, then $\tilde{Q}_1$ can be chosen of period $2p_1$.

Proof. The basic idea of the proof, motivated by [19, Theorem 6], consists of smoothly modifying the function Q_1 of (3.10) to bring it into a periodic orthonormal one still satisfying (3.10). Notice that our assumption implies that the points where eigenvalues coalesce are isolated; therefore without loss of generality we can assume that the eigenvalues are distinct at $t = 0$. Also, in what follows we let $l = k - e_1$.

First, observe that if $\pi(\lambda,\ t)$ is the characteristic polynomial of $A(t)$, then $\pi(\lambda,\ t) = \pi_1(\lambda,\ t)\pi_2(\lambda,\ t)$, where $\pi_1(\lambda,\ t) := (\lambda - \lambda_1(t)) \cdots (\lambda - \lambda_{n_1}(t))$ and $\pi_1(\cdot, t)$ and $\pi_2(\cdot, t)$ do not have common roots. Therefore,

$$\operatorname{rank} \pi_1(A(t), t) = n - n_1 \quad \text{and} \quad \operatorname{rank} \pi_2(A(t), t) = n_1, \quad \text{for all } t.$$

Next, we observe that

$$\pi_1(A(t), t)Q_1(t) = 0 \ \text{ for all } t, \quad \text{where } Q_1 \text{ satisfies (3.10).} \tag{3.11}$$

To show (3.11) is a simple computation:

$$\begin{aligned}
\pi_1(A,\ t)Q_1 &= (A - \lambda_1 I) \cdots (A(t)Q_1 - \lambda_{n_1} Q_1) \\
&= (A - \lambda_1 I) \cdots (A - \lambda_{n_1 - 1} I) Q_1 (D - \lambda_{n_1} I) = \cdots \\
&= Q_1 (D - \lambda_1 I)(D - \lambda_2 I) \cdots (D - \lambda_{n_1} I) \\
&= Q_1 \operatorname{diag}(0, \lambda_2, \dots, \lambda_{n_1}) \cdots \operatorname{diag}(\lambda_1, \dots, \lambda_{n_1 - 1}, 0) = 0 \quad \text{for all } t\ .
\end{aligned}$$

Therefore, $Q_1(\cdot + p_1)$ and $Q_1(\cdot)$ satisfy the same linear system (3.11), which has constant rank $n - n_1$, and their ranks are equal to n_1 for all t. In particular, this implies that there exists a sufficiently small $\rho > 0$ and a $C \in \mathcal{C}^l([-\rho, \rho], \mathbb{C}^{n_1\times n_1})$ such that

$$Q_1(t + p_1)\ C(t)\ =\ Q_1(t),\ |t| \le \rho. \tag{3.12}$$

Since both $Q_1(t + p_1)$ and $Q_1(t)$ are orthonormal, clearly $C^*C = I$. Since the eigenvalues are distinct at $t = 0$, then, for $|t| \le \rho$, $C(t) = \operatorname{diag}(e^{i\phi_j(t)},\ j = 1, \dots, n_1)$ with $\phi_j \in \mathcal{C}^l$. Next, let $\tilde{C}(t) = C^*(0)C(t)$ for all $t \in [-\rho, \rho]$, so that $\tilde{C}(0) = I$ and $[Q_1(t + p_1)C(0)]\tilde{C}(t) = Q_1(t)$. Further, for ρ sufficiently small, the following function is well defined:

$$R(t) = \frac{1}{2}(I + \tilde{C}(t))^{-1}(I - \tilde{C}(t)),\ \ |t| \le \rho,$$

and notice that $R^*(t) = -R(t)$ and $R(0) = 0$. Now take a function v which has continuous derivatives of all orders, $0 \le v(t) \le 1$ for all t, $v(t) = 1$ for $t \ge 0$ and

$v(t) = 0$ for $t \le -r_0$, where $r_0 > 0$ is sufficiently small, $r_0 \le \rho$ (such v is called a *mollifier* in [8]). Then define

$$\tilde{R}(t) = v(t)R(t), \quad -r_0 \le t \le 0,$$

and notice that $\tilde{R}(t) = 0$ for $-\infty < t \le -r_0$. Let

$$\hat{C}(t) = \begin{cases} C(t), & 0 \le t \le \rho, \\ C(0)(I - 2\tilde{R}(t))(I + 2\tilde{R}(t))^{-1}, & t \le 0, \end{cases}$$

notice that $\hat{C} \in \mathcal{C}^l$ is unitary (and diagonal), and set

$$\hat{Q}_1(t) = Q_1(t)\hat{C}(t - p_1) \text{ for all } t \le p_1 + \rho.$$

Thus, $\widehat{Q}_1$ is orthonormal, $\mathcal{C}^l$ on $[-\rho, p_1 + \rho]$, and satisfies (3.10). Using (3.12), we obtain

$$\hat{Q}_1(t) = \begin{cases} Q_1(t)C(0), & -\rho \le t \le p_1 - r_0, \\ Q_1(t - p_1), & p_1 \le t \le p_1 + \rho. \end{cases} \tag{3.13}$$

Now, take a $C^\infty(\mathbb{R})$ function w such that $0 \le w(t) \le 1$ for all t, $w(t) = 1$ for $t \ge p_1 - r_1$ and $w(t) = 0$ for $t \le r_1$, where $r_1 > 0$ is sufficiently small, $r_1 \le r_0$. Let

$$L = \log C(0), \quad N(t) = w(t)L, \quad H(t) = \exp(N(t)) \text{ for all } t,$$

so that $L^* = -L$, H is $\mathcal{C}^\infty$ and unitary for all t, and

$$H(t) = \begin{cases} I_{n-m_1}, & t \le r_1, \\ C(0), & t \ge p_1 - r_1. \end{cases}$$

Finally, let

$$\tilde{Q}_1(t) = \hat{Q}_1(t)H(t), \quad -r_1 \le t \le p_1 + r_1, \tag{3.14}$$

so that $\tilde{Q}_1 \in \mathcal{C}^l([-r_1, p_1 + r_1], \mathbb{C}^{n \times n_1})$, and $\tilde{Q}_1$ is orthonormal and satisfies (3.10). Moreover, $\tilde{Q}_1$ satisfies

$$\tilde{Q}_1(t) = \begin{cases} Q_1(t)C(0), & -r_1 \le t \le r_1, \\ Q_1(t - p_1)C(0), & p_1 \le t \le p_1 + r_1. \end{cases}$$

In particular, $\tilde{Q}_1(t + p_1) = \tilde{Q}_1(t)$, $0 \le t \le r_1$, and thus the proof of the theorem follows by periodically extending $\tilde{Q}_1([0, p_1], \mathbb{C}^{n \times n_1})$.

In the case where A is real valued, and Q_1 and thus C in (3.12) are real as well, the previous construction fails because $\log(C(0))$, and hence $H(t)$ are complex valued, in general. This is because now $C(0)$ is a diagonal matrix of ± 1. To remedy this, define the function $\bar{Q}_1$ by

$$\bar{Q}_1(t) = \begin{cases} \hat{Q}_1(t + p_1)C(0), & -p_1 - \rho \le t \le 0, \\ \widehat{Q}_1(t), & 0 \le t \le p_1 + \rho. \end{cases}$$

Then $\bar{Q}_1$ is orthonormal, $\bar{Q}_1 \in \mathcal{C}^l([-p_1-\rho, p_1+\rho], \mathbb{R}^{n\times m_1})$, and satisfies (3.10). Furthermore,

$$\bar{Q}_1(t) = \begin{cases} Q_1(t+p_1)(C(0))^2, & -p_1-\rho \le t \le -r_0, \\ Q_1(t-p_1), & p_1 \le t \le p_1+\rho. \end{cases}$$

Now let $L = \log(C(0))^2$, which we can (and do) take as real logarithm. As before, we now build an orthogonal $\mathcal{C}^\infty$ function H such that

$$H(t) = \begin{cases} I_{n-m_1}, & t \le -p_1+r_1, \\ (C(0))^2, & t \ge p_1-r_1. \end{cases}$$

Then we let $\tilde{Q}_1(t) = \bar{Q}_1(t)H(t)$ for $-p_1 - r_1 \le t \le p_1 + r_1$ and have $\tilde{Q}_1$ real orthonormal and C^l. Moreover, it satisfies

$$\tilde{Q}_1(t) = \begin{cases} Q_1(t+p_1)(C(0))^2, & -p_1-r_1 \le t \le -p_1+r_1, \\ Q_1(t-p_1)(C(0))^2, & p_1 \le t \le p_1+r_1, \end{cases}$$

so that $\tilde{Q}_1(t+2p_1) = \tilde{Q}_1(t)$, $-p_1 \le t \le -p_1+r_1$, and thus we can build a real $\mathcal{C}^l$ orthonormal function of period $2p_1$. ∎

We now give periodicity results for the SVD of a 1-periodic function allowing the singular values to coalesce. The situation is very close to what we have just proven for the Hermitian eigenproblem. We will consider the $\mathcal{C}^k$ case of a constant rank function. The $\mathcal{C}^\omega$ case is dealt with in a similar way (see Remark 3.26).

First, we consider the case of complex valued A. We have the following.

THEOREM 3.24. *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times n})$, let $\operatorname{rank}(A) = n-r$ for all t, and let there exist $e \le k$ such that for the nonzero singular values of A (2.5) holds:*

$$\liminf_{\tau\to 0} \frac{|\sigma_i(t+\tau) - \sigma_j(t+\tau)|}{|\tau^e|} \in (0, \infty] \tag{3.15}$$

for all t and $i \ne j$. Then, for the matrix-valued function of singular values of A, $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-r})$, we have $\Sigma \in \mathcal{C}_p^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$, $1 \le p \le \mu^*(n-r)$.

Proof. The result follows from Theorem 3.3(i) and Theorem 3.10. ∎

Under the assumptions of Theorem 3.24, we know that there exist orthonormal functions U, V of appropriate dimensions such that

$$U^*AV = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-r}).$$

Moreover, if we let $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$ and $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ partitioned so that $U_1^*AV_1 = \Sigma$, then from Theorem 3.3(i) we know that $U_2 \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{m\times(m-n+r)})$, $V_2 \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{C}^{n\times r})$. We also know that $U_1 \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{C}^{m\times(n-r)})$, $V_1 \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{C}^{n\times(n-r)})$. Further, we have the following.

THEOREM 3.25. *Under the assumptions of Theorem 3.24, and with above notation, there exist orthonormal $\tilde{U}_1 \in \mathcal{C}_p^{k-e}(\mathbb{R}, \mathbb{C}^{m\times(n-r)})$ and $\tilde{V}_1 \in \mathcal{C}_p^{k-e}(\mathbb{R}, \mathbb{C}^{n\times(n-r)})$, so that $\tilde{U} = \begin{bmatrix} \tilde{U}_1 & U_2 \end{bmatrix}$ and $\tilde{V} = \begin{bmatrix} \tilde{V}_1 & V_2 \end{bmatrix}$ are unitary and $U^*AV = \left[\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right]$.*

Proof. Since $AV_1 = U_1\Sigma$ and $A^*U_1 = V_1\Sigma$, then

$$\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}U_1 \\ \frac{1}{\sqrt{2}}V_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}}U_1 \\ \frac{1}{\sqrt{2}}V_1 \end{bmatrix} \Sigma. \tag{3.16}$$

Let $Q_1 := \left[\begin{smallmatrix} \frac{1}{\sqrt{2}}U_1 \\ \frac{1}{\sqrt{2}}V_1 \end{smallmatrix}\right]$. As previously noticed, $Q_1 \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{C}^{(m+n)\times(n-r)})$, and Q_1 is orthonormal. So, by Theorem 3.23, we can replace Q_1 by $\tilde{Q}_1 \in \mathcal{C}_p^{k-e}$, still satisfying (3.16): $\left[\begin{smallmatrix} 0 & A \\ A^* & 0 \end{smallmatrix}\right] \tilde{Q}_1 = \tilde{Q}_1 \Sigma$. Define $\tilde{U}_1$ and $\tilde{V}_1$, of the same dimensions as U_1, V_1, respectively, from the partition $\tilde{Q}_1 =: \left[\begin{smallmatrix} \frac{1}{\sqrt{2}}\tilde{U}_1 \\ \frac{1}{\sqrt{2}}\tilde{V}_1 \end{smallmatrix}\right]$. Since we have not only (3.16), but also

$$\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}U_1 \\ -\frac{1}{\sqrt{2}}V_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}}U_1 \\ -\frac{1}{\sqrt{2}}V_1 \end{bmatrix} (-\Sigma),$$

then we also get that

$$\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}\tilde{U}_1 \\ -\frac{1}{\sqrt{2}}\tilde{V}_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}}\tilde{U}_1 \\ -\frac{1}{\sqrt{2}}\tilde{V}_1 \end{bmatrix} (-\Sigma).$$

Recalling that $\sigma_i \neq 0$, $i = 1, \ldots, n-r$, and arguing as in the proof of Theorem 2.4, we obtain that $\tilde{U}_1$ and $\tilde{V}_1$ are orthonormal. Using Lemma 2.2, we finally obtain that $\tilde{U}$ and $\tilde{V}$ are unitary. ∎

Remark 3.26. For the SVD of $A \in \mathcal{C}^\omega$, we obtain much the same periodicity results as those of the $\mathcal{C}^k$ case, but the details differ somewhat. First, from [3] one obtains an analytic SVD of analytic A. Then, we can infer periodicity of the singular values as we did for the analytic eigenvalues of a Hermitian function. Finally, we can use the construction of [9] on the analytic Hermitian matrix $\left[\begin{smallmatrix} 0 & A \\ A^* & 0 \end{smallmatrix}\right]$, as already pointed out in Remark 3.20. The details of this construction are omitted.

To complete the periodicity results for the SVD of a matrix, we now turn to the case of a real-valued function A. We have the following.

THEOREM 3.27. *Let $A \in \mathcal{C}_1^k(\mathbb{R}, \mathbb{R}^{m\times n})$ and let* $\operatorname{rank}(A) = n - r$ *for all t. Let $e \leq k$ be such that for the nonzero singular values of A* (3.15) *holds, so that there exist orthogonal $U \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{R}^{m\times m})$ and $V \in \mathcal{C}^{k-e}(\mathbb{R}, \mathbb{R}^{n\times n})$ such that $U^*AV = \left[\begin{smallmatrix} \Sigma & 0 \\ 0 & 0 \end{smallmatrix}\right]$ with $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-r})$. Then we can take $\Sigma \in \mathcal{C}_p^k(\mathbb{R}, \mathbb{R}^{(n-r)\times(n-r)})$, $p \leq \mu^*(n-r)$. Moreover, we can choose $U = \begin{bmatrix} \tilde{U}_1 & U_2 \end{bmatrix}$ and $V = \begin{bmatrix} \tilde{V}_1 & V_2 \end{bmatrix}$, with $A\tilde{V}_1 = \tilde{U}_1\Sigma$, and $\tilde{U}_1 \in \mathcal{C}_{2p}^{k-e}(\mathbb{R}, \mathbb{R}^{m\times(n-r)})$, $\tilde{V}_1 \in \mathcal{C}_{2p}^{k-e}(\mathbb{R}, \mathbb{R}^{n\times(n-r)})$ and $U_2 \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{m\times(m-n+r)})$, $V_2 \in \mathcal{C}_2^k(\mathbb{R}, \mathbb{R}^{n\times r})$.*

Proof. The main ingredient is to show the stated periodicity of the singular values' function Σ. To this end, we can reason as follows. The σ_i's are the positive square roots of the nonzero eigenvalues of A^TA: $\sigma_i(t) = \sqrt{\lambda_i(t)}$ for all t for a given ordering of the nonzero $\mathcal{C}^k$ eigenvalues of A^TA. In particular, the σ_i's can be taken of the same period as the λ_i's. Because of the assumption (3.15), and of Theorem 3.2 (in particular, (3.2) in the real, normal case), we have that there exists orthogonal function $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ such that

$$V^T(A^TA)V = \begin{bmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_{n-r} & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix}.$$

Since A^TA has period 1, then reasoning as in the proof of Lemma 3.14 relatively to $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_{n-r})$, we obtain that D can be taken of period p, $1 \leq p \leq \mu^*(n-r)$.

At this point, we proceed similarly to the proof of Theorem 3.25, by using Theorem 3.3(i) and Theorem 3.23 in the real case, to obtain that U and V can be chosen as stated. ∎

Acknowledgment. J. L. Chern gratefully acknowledges the hospitality received from the School of Mathematics and Center for Dynamical Systems and Nonlinear Studies at Georgia Tech for the academic year 1997–1998.

REFERENCES

[1] V. M. Adamjan, D. Z. Arov, and M. G. Krein, *Analytic properties of Schmidted Schur-Takagi problem*, Math. USSR Sb., 15 (1971), pp. 31–73.
[2] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical solution of Initial Value Problems in Differential-Algebraic Equations*, North–Holland, New York, 1989.
[3] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, *Numerical computation of an analytic singular value decomposition by a matrix valued function*, Numer. Math., 60 (1991), pp. 1–40.
[4] A. Bunse-Gerstner and W. B. Gragg, *Singular value decompositions of complex symmetric matrices*, J. Comput. Appl. Math., 21 (1988), pp. 41–54.
[5] S. L. Campbell, *The numerical solution of higher index linear time varying singular systems of differential equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 334–348.
[6] L. Dieci and T. Eirola, *On smooth decomposition of matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 800–819.
[7] V. A. Eremenko, *Some properties of periodic matrices*, Ukrainian Math. J., 32 (1980), pp. 19–26.
[8] A. Friedman, *Foundations of Modern Analysis*, Holt, Rinehart and Winston, New York, 1970.
[9] H. Gingold and P. F. Hsieh, *Globally analytic triangularization of a matrix function*, Linear Algebra Appl., 169 (1992), pp. 75–101.
[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[11] J. K. Hale, *Ordinary Differential Equations*, Krieger Publishing Co., Malabar, 1980.
[12] M. Hayes, *Inhomogeneous plane waves*, Arch. Ration. Mech. Anal., 85 (1984), pp. 41–79.
[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[14] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed., Springer-Verlag, Berlin, 1976.
[15] P. Kunkel and V. Mehrmann, *Canonical forms for linear differential-algebraic equations with variable coefficients*, J. Comput. Appl. Math., 56 (1994), pp. 225–251.
[16] E. V. Mamontov, *Some properties of a system of first order ordinary differential nonlinear equations with a singular matrix of constant rank in front of the vector of the derivatives*, Differ. Uravn., 24 (1988), pp. 1055–1058.
[17] D. Pütz, *Strukturehaltende Interpolation glatter Singulärwertzerlegungen*, Ph.D. thesis, RWTH Aachen, Aachen, Germany, 1994.
[18] C. E. Reid and E. Brändas, *On a theorem for complex symmetric matrices and its relevance in the study of decay phenomena*, in Resonances, Lecture Notes in Phys., 325, Springer-Verlag, Berlin, 1989, pp. 475–483.
[19] Y. Sibuya, *Some global properties of matrices of functions of one variable*, Math. Ann., 161 (1965), pp. 67–77.
[20] V. A. Yakubovich and V. M. Starzhinskii, *Linear Differential Equations with Periodic Coefficients*, Vol. 1 and 2, John Wiley, New York, 1975.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 793–818

GENERALIZED AUGMENTED MATRIX PRECONDITIONING APPROACH AND ITS APPLICATION TO ITERATIVE SOLUTION OF ILL-CONDITIONED ALGEBRAIC SYSTEMS*

ALEXANDER PADIY†, OWE AXELSSON‡, AND BEN POLMAN‡

Abstract. The present work is devoted to a class of preconditioners based on the augmented matrix approach considered earlier by two of the present authors. It presents some generalizations of the subspace-correction schemes studied earlier and gives a brief comparison of the developed technique with a somewhat similar "deflation" algorithm.

The developed preconditioners are able to improve significantly an eigenvalue distribution of certain severely ill-conditioned algebraic systems by using properly chosen projection matrices, which correct the low-frequency components in the spectrum. One of the main advantages of the proposed approach is the possibility of using inexact solvers within the projectors. Another attractive feature of the developed method is that it can be easily combined with other preconditioners, for instance, those which correct the high-frequency eigenmodes.

Key words. iterative solvers, preconditioning, subspace correction

AMS subject classifications. 65F10, 65F15, 65N55

PII. S0895479899356754

1. Introduction.

In many problems the convergence of iterative schemes can be significantly slowed down by a presence of several very small eigenvalues in the spectrum of the algebraic system to be solved. This occurs, for example, when the conjugate gradient (CG) method is applied to algebraic problems arising from discretization of second-order elliptic problems, especially in the case of strongly discontinuous and/or anisotropic problem coefficients.

One of the ways to improve the convergence rate of the CG method is to "deflate" certain components of the residual by using the projector

$$B = I - V(V^T A V)^{-1} V^T A$$

as a (right) preconditioner; see, e.g., [12]. Here A is the original system matrix and V is a rectangular matrix constructed in such a way that the Rayleigh quotient $(\mathbf{x}^T\mathbf{x})/(\mathbf{x}^T A^{-1}\mathbf{x})$ does not take extremely small/large values on the subspace orthogonal to the image of V. Note that a projector of a similar structure appears also in the multigrid setting. If V is chosen to be a coarse-to-fine prolongation operator, then B is normally referred to as a *coarse-grid correction operator*. (An overview of the multigrid framework can be found in, e.g., [8, 16, 17].)

A nice feature of the algorithm is that the convergence rate of the "deflated" preconditioned conjugate (PCG) method depends on the ratio $\tilde{\kappa}$,

$$\tilde{\kappa} = \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}}, \qquad \tilde{\lambda}_{\max} = \sup_{\mathbf{x}\perp \operatorname{Im} V} \frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}}, \qquad \tilde{\lambda}_{\min} = \inf_{\mathbf{x}\perp \operatorname{Im} V} \frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}},$$

*Received by the editors June 8, 1999; accepted for publication (in revised form) by D. O'Leary May 15, 2000; published electronically October 31, 2000. This work was partly supported by NWO Dutch–Russian collaborative research program 047.003.017.

http://www.siam.org/journals/simax/22-3/35675.html

†Philips Research Laboratories Eindhoven, WAY-41, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands (alexander.padiy@philips.com).

‡Department of Mathematics, University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands (axelsson@sci.kun.nl, polman@sci.kun.nl).

rather than on the condition number κ,

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}, \qquad \lambda_{\max} = \sup \frac{\mathbf{x}^T A\,\mathbf{x}}{\mathbf{x}^T\mathbf{x}}, \qquad \lambda_{\min} = \inf \frac{\mathbf{x}^T A\,\mathbf{x}}{\mathbf{x}^T\mathbf{x}}.$$

Since $\tilde{\lambda}_{\max} \le \lambda_{\max}$ and $\tilde{\lambda}_{\min} \ge \lambda_{\min}$, the convergence rate of the "deflated" PCG method is always better than the convergence rate of the unpreconditioned one. As was shown in [12] for a class of second-order elliptic problems with smooth isotropic coefficients, the dimension of the low-frequency eigencluster is normally relatively small and, therefore, the number of columns in V can also be chosen small as compared to the dimension of A. This makes efficient implementations of the "deflation" procedure possible; see [10, 11, 12]. However, the algorithm requires the system with the matrix $A_V = V^T A\,V$ to be solved exactly on every iteration of the PCG method; if the action of A_V^{-1} is computed inaccurately, then the convergence of the iterative scheme can be slow or divergence can even occur. This is a drawback of the "deflation" procedure since the computation of A_V^{-1} can be costly if its dimension is not small; the system with A_V can be efficiently solved only if it has a simple sparsity structure (preferably block-diagonal) and, thus, the choice of V is severely restricted.

Following the idea suggested in [2], but strongly extending and improving that method, we consider in the present paper an alternative approach to tackle the low-frequency eigenmodes. Instead of "deflating" the small eigenvalues we propose to "move" them to the vicinity of the largest eigenvalue by using a preconditioner B in the form

$$B = I + \sigma V B_V^{-1} V^T,$$

where B_V is an easily invertible approximation of A_V; we also refer to [9], where a somewhat similar algorithm was studied.

One of the main advantages of the proposed algorithm is the possibility of avoiding exactly solving systems with A_V. This relaxes the restrictions posed on the choice of V and often leads to more efficient implementations of the solver. Moreover, the algorithm involves no extra multiplication with the system matrix A (as required in the "deflation" method) and can be easily combined with another preconditioner M which bounds the largest eigenvalues:

$$B = M^{-1} + \sigma V B_V^{-1} V^T.$$

The developed algorithm belongs to the additive Schwarz framework. When applied recursively with particular choices of M, V, and B_V it leads to a number of known methods such as I-cycle algebraic multilevel iterations (I-AMLI) [1, 3], Bramble–Pasciak–Xu (BPX) [4], or multiple level diagonal scaling (MDS) [21]. This issue is addressed in sections 3 and 4. The method has the same form as the auxiliary space or two-level method discussed in [18], which is presented in finite element matrix contexts. Our presentation is purely algebraic.

We will first introduce the method as a generalization of the augmented matrix preconditioning approach [2] and then discuss its application to the problems arising from finite element discretization of second-order elliptic equations with highly discontinuous and/or anisotropic coefficients.

2. The augmented matrix preconditioning approach. Let the matrices A and V be of order $n \times n$ and $n \times m$, respectively. Assume that $\operatorname{rank} V = m$. Consider

the augmented matrix

$$\widetilde{A} = \begin{bmatrix} A & -AV \\ -V^TA & V^TAV \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ -V^T & I_m \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_n & -V \\ 0 & I_m \end{bmatrix} \tag{1}$$

of order $(n+m) \times (n+m)$.

THEOREM 2.1 (see [2]). *The following relations between the eigenvalues of $\widetilde{A}$ and A hold.*

(a) *$\widetilde{A}$ has at least m zero eigenvalues. The rest of the spectrum of $\widetilde{A}$ coincides with the spectrum of $(I + VV^T)A$.*

(b) *If A is symmetric positive definite, then for every eigenvalue λ_i of A there exists an eigenvalue $\widetilde{\lambda}_i$ of $\widetilde{A}$ such that $\widetilde{\lambda}_i \geq \lambda_i$.*

(c) *If A is nonsingular and symmetric and V is constructed as*

$$V = [\alpha_1 \mathbf{v}_1, \ldots, \alpha_m \mathbf{v}_m],$$

where $\mathbf{v}_i$ are the normalized eigenvectors of A corresponding to λ_i, $i = 1, \ldots, m$, then the nonzero eigenvalues of $\widetilde{A}$ are the following:

$$\widetilde{\lambda}_i = \begin{cases} (1 + \alpha_i^2)\lambda_i, & i = 1, \ldots, m, \\ \lambda_i, & i = m+1, \ldots, n. \end{cases}$$

Proof. It follows from (1) that $\widetilde{A}$ is similar to

$$\begin{bmatrix} I_n & -V \\ 0 & I_m \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -V^T & I_m \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} (I_n + VV^T)A & 0 \\ -V^TA & 0 \end{bmatrix}.$$

This shows part (a) of the theorem. The eigenvalues of $(I_n + VV^T)A$ are equal to those of $A + A^{\frac{1}{2}}VV^TA^{\frac{1}{2}}$ so part (b) follows from $\mathbf{x}^TA\mathbf{x} + (V^TA^{\frac{1}{2}}\mathbf{x})^T(V^TA^{\frac{1}{2}}\mathbf{x}) \geq \mathbf{x}^TA\mathbf{x}$ for any x. Part (c) immediately follows from the orthonormality of the eigenvectors $\mathbf{v}_i$ of A, $i = 1, \ldots, n$. ∎

Assume that A is symmetric positive definite with an ordered set of eigenvalues $\{\lambda_i\}_{i=1}^n$, $\lambda_1 \leq \cdots \leq \lambda_n = \lambda_{\max}$. In this case the above theorem implies that in order to improve the condition number of $\widetilde{A}$ one can define the matrix V by using the eigenvectors $\mathbf{v}_i$, $i = 1, \ldots, m$ of A. If the scaling factors α_i are chosen as $\alpha_i = \sqrt{\lambda_n/\lambda_i - 1}$ or $\alpha_i = \sqrt{\lambda_n/\lambda_i}$, then the smallest eigenvalues λ_i of A are "moved" to $\tilde{\lambda}_i = \lambda_n = \lambda_{\max}$ or to $\tilde{\lambda}_i = \lambda_n + \lambda_i \leq 2\lambda_{\max}$, respectively. As was pointed out in [2], instead of using the matrix $\widetilde{A}$ in the iterative scheme, one can alternatively use the matrix $(I + VV^T)A$, i.e., one can use $I + VV^T$ as a preconditioner to A. The preconditioner can also be written in the form $I + VD^{-1}V^T$, where $D = \operatorname{diag}(\alpha_i^2)$ and $V = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$.

There are two problems associated with the practical implementation of this method, namely, the eigenvectors $\mathbf{v}_i$ are not generally known and likewise the scaling factors α_i are not known. To handle this we consider the case when $\mathbf{v}_i$ are only assumed to be linearly independent vectors spanning a proper subspace and introduce a more general scaling matrix. Moreover, as it turns out, it is the subspace spanned by $\{\mathbf{v}_i\}_1^m$ which matters and not the particular basis vectors used. Further we study a preconditioner in a more general form $I + VD^{-1}V^T$, where the matrix D is no longer assumed to be diagonal.

LEMMA 2.2. *Let A be symmetric positive definite. Then*

$$P = A^{\frac{1}{2}} V (V^T A V)^{-1} V^T A^{\frac{1}{2}}$$

is an orthogonal projector. Therefore, 0 and 1 are the only eigenvalues of P.

Proof. Clearly,

$$\begin{aligned} P^2 &= A^{\frac{1}{2}} V (V^T A V)^{-1} V^T A V (V^T A V)^{-1} V^T A^{\frac{1}{2}} \\ &= A^{\frac{1}{2}} V (V^T A V)^{-1} V^T A^{\frac{1}{2}} = P. \end{aligned}$$

Since $P^2 = P$, P is an orthogonal projector. ∎

LEMMA 2.3 (Monotonicity [13]). *Let A and $\hat{A}$ be symmetric positive definite matrices of order $n \times n$ and let V_k be rectangular matrices of order $n \times m_k$, $k = 1, 2$, such that* $\operatorname{rank} V_k = m_k$, *$k = 1, 2$. If* $\operatorname{Im} V_1 \subseteq \operatorname{Im} V_2$, *then for all i, $1 \le i \le n$, the following inequality holds*

$$\lambda_i\big((I + V_2 (V_2^T \hat{A} V_2)^{-1} V_2^T) A\big) \ge \lambda_i\big((I + V_1 (V_1^T \hat{A} V_1)^{-1} V_1^T) A\big).$$

Proof. It is readily seen that the proposition holds if

$$F = V_2 (V_2^T \hat{A} V_2)^{-1} V_2^T - V_1 (V_1^T \hat{A} V_1)^{-1} V_1^T$$

is nonnegative definite. But since $\operatorname{Im} V_1 \subseteq \operatorname{Im} V_2$, there exists some matrix Q of order $m_2 \times m_1$ such that $V_1 = V_2 Q$. Then with $D_k = V_k^T \hat{A} V_k$ we have

$$F = V_2 (D_2^{-1} - Q D_1^{-1} Q^T) V_2^T = V_2 D_2^{-\frac{1}{2}} (I - D_2^{\frac{1}{2}} Q D_1^{-1} Q^T D_2^{\frac{1}{2}}) D_2^{-\frac{1}{2}} V_2^T,$$

where

$$D_2^{\frac{1}{2}} Q D_1^{-1} Q^T D_2^{\frac{1}{2}} = D_2^{\frac{1}{2}} Q (Q^T D_2 Q)^{-1} Q^T D_2^{\frac{1}{2}}$$

is an orthogonal projector, whose only eigenvalues are 0 and 1. ∎

COROLLARY 2.4. *If* $\operatorname{Im} V_1 = \operatorname{Im} V_2$, *then $I + V_2 D_2^{-1} V_2^T = I + V_1 D_1^{-1} V_1^T$.*

Proof. In this case Q in $V_1 = V_2 Q$ is invertible, thus $D_2^{\frac{1}{2}} Q (Q^T D_2 Q)^{-1} Q^T D_2^{\frac{1}{2}} = I$. ∎

Remark 2.5. The above corollary shows that the individual eigenvectors of A are not needed when constructing the matrix V; we are rather interested in the subspace spanned by them.

Next we consider a specific version of the preconditioner $B = I + \sigma V A_V^{-1} V^T$ with the scaling matrix $A_V = V^T A V$. The following theorem is similar to a theorem from [13].

THEOREM 2.6. *Let A be an $n \times n$ symmetric positive semidefinite matrix and let a rectangular matrix V of order $n \times m$ be defined as $V = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$. Assume that* $\operatorname{rank} V = m$. *Further, define $\widetilde{A}$ as $\widetilde{A} = (I + \sigma V A_V^{-1} V^T) A$, where $A_V = V^T A V$. Then the following statements hold:*

(a) $\lambda_{\max}(\tilde{A}) \le \sigma + \lambda_{\max}(A)$;
(b) *if for some i, $1 \le i \le m$, $\mathbf{v}_i$ is an eigenvector of A with eigenvalue λ_i, then it is also an eigenvector of $\widetilde{A}$ with eigenvalue $\lambda_i + \sigma$;*
(c) *let $(\lambda_i, \mathbf{v}_i)$ be the eigenpairs of A and assume that $V = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$ contains m eigenvectors. Then*

$$\widetilde{A}\mathbf{v}_i = \begin{cases} (\lambda_i + \sigma)\mathbf{v}_i, & i = 1, \ldots, m, \\ \lambda_i \mathbf{v}_i, & i = m+1, \ldots, n. \end{cases}$$

Proof. Clearly,

$$\begin{aligned}\lambda_{\max}(\tilde{A}) &\leq \lambda_{\max}(A) + \sigma \sup \frac{\mathbf{x}^T V (V^T A V)^{-1} V^T \mathbf{x}}{\mathbf{x}^T A^{-1} \mathbf{x}} \\ &= \lambda_{\max}(A) + \sigma \sup \frac{\mathbf{x}^T A^{\frac{1}{2}} V (V^T A V)^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\ &= \lambda_{\max}(A) + \sigma,\end{aligned}$$

where the last equality follows from Lemma 2.2. This proves part (a).

Let $\mathbf{w}_i = (V^T V)^{-1} V^T \mathbf{v}_i$. Then for $i = 1, \ldots, m$

$$\begin{aligned}\widetilde{A}\mathbf{v}_i &= \lambda_i \mathbf{v}_i + \sigma V (V^T A V)^{-1} V^T A \mathbf{v}_i \\ &= \lambda_i \mathbf{v}_i + \sigma V (V^T A V)^{-1} V^T A V \mathbf{w}_i \\ &= \lambda_i \mathbf{v}_i + \sigma V \mathbf{w}_i = (\lambda_i + \sigma)\mathbf{v}_i,\end{aligned}$$

which shows part (b). To prove part (c) note that the eigenvectors are orthogonal so $V^T A \mathbf{v}_i = 0$, $i = m+1, \ldots, n$. ∎

COROLLARY 2.7. *Let V be such that $\operatorname{Im} V$ is spanned by the m eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ of A corresponding to the cluster of m smallest eigenvalues $\lambda_1, \ldots, \lambda_m$. Then the eigenvalues of $\widetilde{A} = (I + \sigma V (V^T A V)^{-1} V^T) A$ are $\tilde{\lambda}_i = \sigma + \lambda_i$ for $i = 1, \ldots, m$ and $\tilde{\lambda}_i = \lambda_i$ for $i = m+1, \ldots, n$, which implies that*

$$\min\{\sigma + \lambda_1, \lambda_{m+1}\} \leq \lambda_i(\widetilde{A}) \leq \max\{\sigma + \lambda_m, \lambda_{\max}(A)\}. \tag{2}$$

Proof. Use Corollary 2.4 and Theorem 2.6. ∎

COROLLARY 2.8. *Assume that*

$$\operatorname{Im} V \supseteq \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}. \tag{3}$$

Then the following estimate holds:

$$\min\{\sigma + \lambda_1, \lambda_{m+1}\} \leq \lambda_i(\widetilde{A}) \leq \sigma + \lambda_{\max}(A).$$

Proof. Use Corollary 2.7 and Lemma 2.2. ∎

As follows from the above corollaries, the preconditioner

$$B = I + \sigma V (V^T A V)^{-1} V^T, \quad \sigma = \lambda_{\max}(A), \quad \operatorname{Im} V \supseteq \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \tag{4}$$

scales the smallest eigenvalues λ_i of A, and they are "moved" to $\tilde{\lambda}_i = \lambda_{\max}(A) + \lambda_i$. Since $\tilde{\lambda}_i \leq 2\lambda_{\max}(A)$, this leads to the condition number estimate

$$\kappa(BA) \leq \frac{2\lambda_{\max}(A)}{\lambda_{m+1}}. \tag{5}$$

However, the preconditioner (4) is normally expensive to apply because of the need to invert the matrix A_V. In the following section we show that the action of A_V^{-1} can be replaced by the action of a preconditioner B_V^{-1} to A_V^{-1}. We also discuss there the possibility to relax condition (3).

3. Generalized version with inexact projectors.

THEOREM 3.1. *Define the preconditioner* $\hat{B}$ *as*

$$\hat{B} = I + \hat{\sigma} V B_V^{-1} V^T, \quad \hat{\sigma} = \frac{\lambda_{\max}(A)}{\lambda_{\max}(B_V^{-1} A_V)}, \quad \operatorname{Im} V \supseteq \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}, \tag{6}$$

where B_V *is an* $m \times m$ *symmetric positive definite approximation of* A_V*. The eigenvalues* $\lambda(\hat{B}A)$ *of* $\hat{B}A$ *are bounded as follows:*

$$\min\left\{\frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)} + \lambda_1, \lambda_{m+1}\right\} \le \lambda(\hat{B}A) \le 2\lambda_{\max}(A). \tag{7}$$

Proof. The minimal eigenvalue of $\hat{B}A$ can be estimated as

$$\begin{aligned}
\lambda_{\min}(\hat{B}A) &= \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T(I + \hat{\sigma} V B_V^{-1} V^T)\,\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}}\right\} \\
&= \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}} + \hat{\sigma}\cdot\frac{\mathbf{x}^T V B_V^{-1} V^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}}\right\} \\
&= \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}} + \hat{\sigma}\cdot\frac{\mathbf{x}^T V B_V^{-1} V^T\mathbf{x}}{\mathbf{x}^T V A_V^{-1} V^T\mathbf{x}}\cdot\frac{\mathbf{x}^T V A_V^{-1} V^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}}\right\} \\
&\ge \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}} + \hat{\sigma}\lambda_{\min}(B_V^{-1}A_V)\cdot\frac{\mathbf{x}^T V A_V^{-1} V^T\mathbf{x}}{\mathbf{x}^T A^{-1}\mathbf{x}}\right\} \\
&\ge \min\left\{\frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)} + \lambda_1, \lambda_{m+1}\right\},
\end{aligned}$$

where the last inequality follows from Corollary 2.7 with

$$\sigma = \hat{\sigma}\lambda_{\min}(B_V^{-1}A_V) = \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}.$$

Analogously, $\lambda_{\max}(\hat{B}A) \le 2\lambda_{\max}(A)$. ∎

Remark 3.2. The value of $\hat{\sigma}$ in (6) was chosen as $\hat{\sigma} = \lambda_{\max}(A)/\lambda_{\max}(B_V^{-1}A_V)$ for the ease of presentation. The optimal value of $\hat{\sigma}$ (the value of $\hat{\sigma}$ which minimizes the condition number of $\hat{B}A$) corresponds to the case when

$$\hat{\sigma}\lambda_{\min}(B_V^{-1}A_V) + \lambda_1 = \lambda_{m+1}.$$

Note that λ_{m+1} is not known in general.

Remark 3.3. As follows from (7), if $\kappa(B_V^{-1}A_V) \le \lambda_{\max}(A)/\lambda_{m+1}$, then the bounds for $\kappa(\hat{B}A)$ and $\kappa(BA)$ coincide.

Remark 3.4. As follows from the above remark, if $\kappa(A_V) \le \lambda_{\max}(A)/\lambda_{m+1}$, then one can define B_V simply as $B_V = I$ or $B_V = \operatorname{diag} A_V$.

As follows from Theorem 3.1, the preconditioner (6) is able to improve the spectrum of A even in the case when the action of A_V^{-1} is replaced by the action of a preconditioner B_V^{-1}. It should be noted, however, that the preconditioner (6) is still difficult to implement in practice since the condition $\operatorname{Im} V \supseteq \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is not easy to satisfy. Later, in Theorem 3.6, we show that this condition can be significantly relaxed.

In the following we use the notation $\cos(W_1, W_2)$ for $\cos(\varphi(W_1, W_2))$, where φ denotes the angle between the vector subspaces W_1 and W_2, namely,

$$\cos(W_1, W_2) = \cos(\varphi(W_1, W_2)) = \sup_{\substack{\mathbf{x} \in W_1 \\ \mathbf{y} \in W_2}} \frac{\mathbf{x}^T \mathbf{y}}{(\mathbf{x}^T \mathbf{x})^{\frac{1}{2}} (\mathbf{y}^T \mathbf{y})^{\frac{1}{2}}}.$$

LEMMA 3.5. *Consider two arbitrary matrices $V_1 \in \mathbb{R}_{n \times m_1}$, $\operatorname{rank} V_1 = m_1$, and $V_2 \in \mathbb{R}_{n \times m_2}$, $\operatorname{rank} V_2 = m_2$. If there exists $\gamma < 1$ such that*

$$\cos(\operatorname{Im} V_1, \operatorname{Im} V_2) = \sup_{\substack{\mathbf{x} \in \operatorname{Im} V_1 \\ \mathbf{y} \in \operatorname{Im} V_2}} \frac{\mathbf{x}^T \mathbf{y}}{(\mathbf{x}^T \mathbf{x})^{\frac{1}{2}} (\mathbf{y}^T \mathbf{y})^{\frac{1}{2}}} \leq \gamma,$$

then

$$\lambda_{\max} \left(\sum_{i=1}^{2} V_i (V_i^T V_i)^{-1} V_i^T \right) \leq 1 + \gamma.$$

Proof. Define an auxiliary $(m_1 + m_2) \times n$ matrix R as

$$R = \begin{bmatrix} (V_1^T V_1)^{-\frac{1}{2}} V_1^T \\ (V_2^T V_2)^{-\frac{1}{2}} V_2^T \end{bmatrix}.$$

The matrix R exists since V_i are full rank matrices and, thus, the matrices $V_i^T V_i$ are symmetric positive definite. Since $\lambda_{\max}(Q^T Q) = \lambda_{\max}(Q Q^T)$ for all Q,

$$\lambda_{\max} \left(\sum_{i=1}^{2} V_i (V_i^T V_i)^{-1} V_i^T \right) = \sup_{\mathbf{x}} \frac{\mathbf{x}^T R^T R \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \lambda_{\max}(R^T R) = \lambda_{\max}(R R^T).$$

Taking into account the explicit form of $R R^T$ we have

$$\lambda_{\max}(R R^T) = \lambda_{\max} \left(\begin{bmatrix} I & W_1^T W_2 \\ W_2^T W_1 & I \end{bmatrix} \right) = 1 + \cos(\operatorname{Im} W_1, \operatorname{Im} W_2),$$

where $W_i = V_i (V_i^T V_i)^{-\frac{1}{2}}$, $i = 1, 2$. Clearly, $\operatorname{Im} V_i = \operatorname{Im} W_i$. Thus,

$$\lambda_{\max} \left(\sum_{i=1}^{2} V_i (V_i^T V_i)^{-1} V_i^T \right) = \lambda_{\max}(R R^T) = 1 + \cos(\operatorname{Im} V_1, \operatorname{Im} V_2) \leq 1 + \gamma. \qquad \square$$

THEOREM 3.6. *Consider the preconditioner $\hat{B}$*

$$\hat{B} = I + \hat{\sigma} V B_V^{-1} V^T, \qquad \hat{\sigma} = \lambda_{\max}(A) / \lambda_{\max}(B_V^{-1} A_V). \tag{8}$$

Assume that $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^n$ is an ordered set of eigenpairs of A such that $\lambda_1 \leq \cdots \leq \lambda_n$. If V is such that the subspaces $\mathcal{W} = (\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$ and $\mathcal{V}_e = \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ satisfy the condition

$$\cos(\mathcal{W}, \mathcal{V}_e) = \sup_{\substack{\mathbf{x} \in \mathcal{W} \\ \mathbf{y} \in \mathcal{V}_e}} \frac{\mathbf{x}^T \mathbf{y}}{(\mathbf{x}^T \mathbf{x})^{\frac{1}{2}} (\mathbf{y}^T \mathbf{y})^{\frac{1}{2}}} \leq \gamma \tag{9}$$

for some $\gamma < 1$, then the minimal eigenvalue of $\hat{B}A$ is bounded as

$$\lambda_{\min}(\hat{B}A) \geq \max\left\{\lambda_1, (1-\gamma)\cdot\min\left\{\frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}, \lambda_{m+1}\right\}\right\} \tag{10}$$

while $\lambda_{\max}(\hat{B}A)$ is bounded as $\lambda_{\max}(\hat{B}A) \leq 2\lambda_{\max}(A)$ for any choice of V and B_V.

Proof. The maximal eigenvalue $\lambda_{\max}(\hat{B}A)$ can be estimated as

$$\begin{aligned}
\lambda_{\max}(\hat{B}A) &= \sup_{\mathbf{x}}\left\{\frac{\mathbf{x}^T(I+\hat{\sigma}VB_V^{-1}V^T)\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}}\right\} \\
&\leq \sup_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}} + \hat{\sigma}\lambda_{\max}(B_V^{-1}A_V)\cdot\frac{\mathbf{x}^TVA_V^{-1}V^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}}\right\} \\
&= \sup_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}} + \lambda_{\max}(A)\cdot\frac{\mathbf{x}^TVA_V^{-1}V^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}}\right\} \;\leq\; 2\lambda_{\max}(A).
\end{aligned}$$

If we take into account that the eigensubspaces $\mathcal{V}_e$ and $(\mathcal{V}_e)^\perp$ of A are A-orthogonal, then the minimal eigenvalue of $\hat{B}A$ can be estimated as

$$\begin{aligned}
\lambda_{\min}(\hat{B}A) &= \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T(I+\hat{\sigma}VB_V^{-1}V^T)\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}}\right\} \\
&\geq \inf_{\mathbf{x}}\left\{\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}} + \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\cdot\frac{\mathbf{x}^TVA_V^{-1}V^T\mathbf{x}}{\mathbf{x}^TA^{-1}\mathbf{x}}\right\} \\
&= \inf_{\mathbf{y}}\left\{\frac{\mathbf{y}^TA\mathbf{y}}{\mathbf{y}^T\mathbf{y}} + \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\cdot\frac{\mathbf{y}^TA^{\frac{1}{2}}VA_V^{-1}V^TA^{\frac{1}{2}}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \\
&= \inf_{\mathbf{y}}\left\{\frac{\mathbf{y}^TP_e^\perp AP_e^\perp\mathbf{y} + \mathbf{y}^TP_eAP_e\mathbf{y}}{\mathbf{y}^T\mathbf{y}} + \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\cdot\frac{\mathbf{y}^TP_w\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \\
&\geq \inf_{\mathbf{y}}\left\{\frac{\mathbf{y}^TP_e^\perp AP_e^\perp\mathbf{y}}{\mathbf{y}^T\mathbf{y}} + \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\cdot\frac{\mathbf{y}^TP_w\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \\
&\geq \inf_{\mathbf{y}}\left\{\lambda_{m+1}\cdot\frac{\mathbf{y}^TP_e^\perp\mathbf{y}}{\mathbf{y}^T\mathbf{y}} + \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\cdot\frac{\mathbf{y}^TP_w\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \\
&\geq \min\left\{\lambda_{m+1}, \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\right\}\cdot\inf_{\mathbf{y}}\left\{\frac{\mathbf{y}^TP_e^\perp\mathbf{y} + \mathbf{y}^TP_w\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\} \\
&= \min\left\{\lambda_{m+1}, \frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}\right\}\cdot\left(2 - \sup_{\mathbf{y}}\left\{\frac{\mathbf{y}^TP_e\mathbf{y} + \mathbf{y}^TP_w^\perp\mathbf{y}}{\mathbf{y}^T\mathbf{y}}\right\}\right),
\end{aligned}$$

where P_e, $P_e^\perp$, P_w, and $P_w^\perp$ are the orthogonal projectors onto $\mathcal{V}_e$, $\mathcal{V}_e^\perp$, $\operatorname{Im}A^{\frac{1}{2}}V$, and $(\operatorname{Im}A^{\frac{1}{2}}V)^\perp$, respectively. As follows from Lemma 3.5 with the matrices V_1 and V_2 chosen such that $\operatorname{Im}V_1 = \mathcal{V}_e$ and $\operatorname{Im}V_2 = (\operatorname{Im}A^{\frac{1}{2}}V)^\perp$,

$$\lambda_{\min}(\hat{B}A) \;\geq\; (1-\gamma)\cdot\min\left\{\frac{\lambda_{\max}(A)}{\kappa(B_V^{-1}A_V)}, \lambda_{m+1}\right\}.$$

Finally noting that $VB_V^{-1}V^T$ is positive semidefinite we conclude the proof of (10). ∎

Remark 3.7. As follows from (7), $\kappa(\hat{B}A) \leq 2\kappa(A)$ for all choices of V and B_V. Thus, for all V and B_V the convergence rate of the $\hat{B}$-preconditioned iterative scheme is not worse as of the same order as of the unpreconditioned one. In particular, no divergence of the iterative scheme can occur (if we assume that the round-off effects are neglected). This is a nice feature of the developed algorithm as compared to the "deflation" procedure [12], since the latter can be divergent if the matrix B_V is chosen inappropriately.

LEMMA 3.8. *If the eigensubspace* $\mathcal{V}_e = \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ *of* A *is known, then the value of* $\cos(\mathcal{W}, \mathcal{V}_e)$ *in* (9) *is readily computable. It can be evaluated as*

$$\cos(\mathcal{W}, \mathcal{V}_e) = \lambda_{\max}(ZA), \qquad Z = V_e(V_e^T A\, V_e)^{-1} V_e^T - V(V^T A\, V)^{-1} V^T,$$

where the matrix V_e *is chosen such that* $\operatorname{rank} V_e = m$, $\operatorname{Im} V_e = \mathcal{V}_e$.

Proof. Introduce two auxiliary matrices V_e and W such that $\operatorname{Im} V_e = \mathcal{V}_e$, $\operatorname{rank} V_e = \dim \mathcal{V}_e$, $V_e^T V_e = I$, $\operatorname{Im} W = \mathcal{W}$, $\operatorname{rank} W = \dim \mathcal{W}$, $\mathcal{W} = (\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$, $W^T W = I$. Similarly to the proof of Lemma 3.5, we have

$$\begin{aligned}
\cos(\mathcal{W}, \mathcal{V}_e) &= \lambda_{\max}\left(\begin{bmatrix} I & W^T V_e \\ V_e^T W & I \end{bmatrix}\right) - 1 \\
&= \lambda_{\max}\left(V_e V_e^T + W W^T\right) - 1 \\
&= \lambda_{\max}\left(V_e V_e^T + I - A^{\frac{1}{2}} V (V^T A\, V)^{-1} V^T A^{\frac{1}{2}}\right) - 1 \\
&= \sup_{\mathbf{x}} \frac{\mathbf{x}^T V_e (V_e^T V_e)^{-1} V_e^T \mathbf{x} - \mathbf{x}^T A^{\frac{1}{2}} V (V^T A\, V)^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\
&= \sup_{\mathbf{y}} \frac{\mathbf{y}^T V_e (V_e^T A V_e)^{-1} V_e^T \mathbf{y} - \mathbf{y}^T V (V^T A\, V)^{-1} V^T \mathbf{y}}{\mathbf{y}^T A^{-1} \mathbf{y}},
\end{aligned}$$

where the last equality follows from Corollary 2.4 by taking into account that the subspace $\mathcal{V}_e$ is $A^{\frac{1}{2}}$-invariant. Since

$$\lambda_{\max}(QA) = \sup_{\mathbf{x}} \frac{\mathbf{x}^T Q \mathbf{x}}{\mathbf{x}^T A^{-1} \mathbf{x}}, \qquad Q = Q^T \geq 0, \quad A = A^T > 0,$$

we conclude that

$$\cos(\mathcal{W}, \mathcal{V}_e) = \lambda_{\max}(ZA), Z = V_e(V_e^T A\, V_e)^{-1} V_e^T - V(V^T A\, V)^{-1} V^T. \qquad \square$$

Remark 3.9. Since the eigensubspace $\mathcal{V}_e = \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is $A^{\frac{1}{2}}$-invariant, it follows from $\operatorname{Im} V \supseteq \mathcal{V}_e$ that $\operatorname{Im} A^{\frac{1}{2}} V \supseteq \mathcal{V}_e$. This implies that $\gamma = 0$ if $\operatorname{Im} V \supseteq \mathcal{V}_e$.

As follows from Theorem 3.6, there is no need to approximate the subspace spanned by $\{\mathbf{v}_i\}_{i=1}^m$ with a very high accuracy. For the isotropic second-order elliptic problem with a smooth coefficient function we can let, for instance, $V = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$, where $\mathbf{w}_i$ are the pointwise nodal values of the coarse-mesh finite element basis functions φ_i^H. An iterative scheme based on such choice of V was constructed in [12]. Another choice of $\mathbf{w}_1, \ldots, \mathbf{w}_m$ could be the basis vectors of a known eigensubspace of a similar problem, such as a problem with a different coefficient function. This approach was taken in [3] and [14], where the low-frequency subspace of a strongly

anisotropic diffusion operator was approximated using the eigenvectors of the limit problem with a degenerate diffusion tensor.

It should also be noted that the algorithm (8) can be applied recursively, i.e., we can consider a nested sequence of preconditioners $\{\hat{B}_k\}_{k=0}^{J}$ defined as

$$\hat{B}_k = I + \hat{\sigma}_k V_{k,k-1} \hat{B}_{k-1} V_{k,k-1}^T. \tag{11}$$

According to the classification introduced in [17] the above recursive algorithm belongs to the class of *parallel subspace correction* methods.

Remark 3.10. In the case when $B_0 = I$ and $V_{k,k-1}$ are prolongation operators in the standard multigrid setting the developed algorithm corresponds to the BPX method [4].

Remark 3.11. The matrices $V_{k,k-1}$ in the multilevel preconditioner (11) can be constructed using the matrix-dependent prolongation operators developed in [6, 20] or [5, 15]. We also refer to [7], where preconditioners of a similar structure were studied.

If we additionally introduce a polynomial stabilization procedure to bound the condition number of $\kappa(B_k A_k)$ (see [1], for instance), then we arrive at the BPX-like preconditioner of the W-cycle type:

$$\bar{B}_k = \left(I - P_{\nu_k}(\hat{B}_k A_k)\right) A_k^{-1}, \qquad \hat{B}_k = I + \bar{\sigma}_k V_{k,k-1} \bar{B}_{k-1} V_{k,k-1}^T,$$

where

$$A_l = A, \qquad A_{k-1} = V_{k,k-1}^T A_k V_{k,k-1},$$

and P_{ν_k} denotes a Chebyshev polynomial of degree ν_k normalized at the origin.

4. Incorporation of an "external" smoother. The method presented above improves the condition number of the preconditioned system by "moving" the smallest eigenvalues to the upper part of the spectrum. Next we show how it can be combined with a smoother, which essentially improves the conditioning by making the largest eigenvalues smaller.

The preconditioner is constructed as

$$\tilde{B} = M^{-1} + \tilde{\sigma} V B_V^{-1} V^T, \qquad \tilde{\sigma} = \lambda_{\max}(M^{-1}A)/\lambda_{\max}(B_V^{-1} A_V), \tag{12}$$

where M and B_V are symmetric positive definite preconditioners for A and A_V, respectively.

Remark 4.1. Preconditioners in this form appear within the additive Schwarz framework (with application to domain decomposition methods). The term $\tilde{\sigma} V B_V^{-1} V^T$ then normally corresponds to the coarse-mesh correction operator while the smoother M^{-1} corresponds to a series of subdomain solves.

Remark 4.2. When applied recursively in the standard multigrid setting with $M_k = \operatorname{diag}(A_k)$, the algorithm (12) corresponds to the MDS method [21].

Remark 4.3. Consider the case when the matrices A_k are generated using the hierarchical basis of finite elements. If the smoother M_k is defined as

$$M_k^{-1} = I - V_{k,k-1}(V_{k,k-1}^T V_{k,k-1})^{-1} V_{k,k-1}^T,$$

then the algorithm corresponds to the multilevel method developed in [19]. If the smoother M_k is extended to the form

$$M_k^{-1} = (I - V_{k,k-1}(V_{k,k-1}^T V_{k,k-1})^{-1} V_{k,k-1}^T)\, M_{11}^{(k)}\, (I - V_{k,k-1}(V_{k,k-1}^T V_{k,k-1})^{-1} V_{k,k-1}^T)$$

and, additionally, the polynomial stabilization procedure is used to bound the condition number of $\kappa(\hat{B}_k A_k)$, then the method reduces to the additive version of the AMLI method [1, 3].

THEOREM 4.4. *Consider the preconditioner* (12). *Assume that* $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^n$ *is an ordered set of eigenpairs of* $M^{-1}A$ *such that* $\lambda_1 \le \cdots \le \lambda_n$. *Define the matrix* V_e *as* $V_e = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$. *If* V *is such that the subspaces* $\mathcal{W} = (\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$ *and* $\mathcal{V}_e = \operatorname{Im} A^{\frac{1}{2}} V_e$ *satisfy the condition* (9) *for some* $\gamma < 1$, *then the minimal eigenvalue of* $\tilde{B}A$ *is bounded as*

$$\lambda_{\min}(\tilde{B}A) \ge \max\left\{\lambda_1, (1-\gamma)\cdot \min\left\{\frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}, \lambda_{m+1}\right\}\right\}. \tag{13}$$

The maximal eigenvalue of $\tilde{B}A$ *is bounded as*

$$\lambda_{\max}(\tilde{B}A) \le 2\lambda_{\max}(M^{-1}A) \tag{14}$$

for any choice of V *and* B_V.

Proof. The maximal eigenvalue $\lambda_{\max}(\tilde{B}A)$ can be estimated as in the proof of Theorem 3.6. Taking into account that $\mathcal{V}_e$ is the eigensubspace of $A^{\frac{1}{2}} M^{-1} A^{\frac{1}{2}}$ the minimal eigenvalue can be estimated as

$$\begin{aligned}
\lambda_{\min}(\tilde{B}A) &= \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T A^{\frac{1}{2}}\, M^{-1} A^{\frac{1}{2}} \mathbf{x} + \tilde{\sigma} \mathbf{x}^T A^{\frac{1}{2}} V B_V^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\ge \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T A^{\frac{1}{2}}\, M^{-1} A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} + \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)} \cdot \frac{\mathbf{x}^T A^{\frac{1}{2}} V A_V^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\ge \min\left\{\lambda_{m+1}, \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}\right\} \cdot \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T P_e^{\perp} \mathbf{x} + \mathbf{x}^T P_* \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&= \min\left\{\lambda_{m+1}, \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}\right\} \cdot \left(2 - \sup_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T P_e \mathbf{x} + \mathbf{x}^T P_*^{\perp} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\}\right),
\end{aligned}$$

where P_e, $P_e^{\perp}$, P_*, and $P_*^{\perp}$ are the orthogonal projectors onto $\mathcal{V}_e = \operatorname{Im} A^{\frac{1}{2}} V_e$, $\mathcal{V}_e^{\perp} = (\operatorname{Im} A^{\frac{1}{2}} V_e)^{\perp}$, $\operatorname{Im} A^{\frac{1}{2}} V$, and $(\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$, respectively. As follows from Lemma 3.5 with the matrices V_1 and V_2 chosen such that $\operatorname{Im} V_1 = \mathcal{V}_e$ and $\operatorname{Im} V_2 = (\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$,

$$\lambda_{\min}(\tilde{B}A) \ge (1-\gamma)\cdot \min\left\{\frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}, \lambda_{m+1}\right\}.$$

Now (13) follows by taking into account that $V B_V^{-1} V^T$ is positive semidefinite. ∎

Remark 4.5. The value of $\tilde{\sigma}$ in (12) was chosen as

$$\tilde{\sigma} = \lambda_{\max}(M^{-1}A)/\lambda_{\max}(B_V^{-1}A_V)$$

for the ease of presentation. The optimal value of $\tilde{\sigma}$ (the value of $\tilde{\sigma}$ which minimizes the condition number of $\tilde{B}A$) corresponds to the case when

$$\tilde{\sigma}\lambda_{\min}(B_V^{-1}A_V) + \lambda_1 = \lambda_{m+1}.$$

LEMMA 4.6. *If the eigensubspace* $\mathcal{V}_e = \operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ *of* $M^{-1}A$ *is known, then the value of* $\cos(\mathcal{W}, \mathcal{V}_e) = \cos((\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}, \operatorname{Im} A^{\frac{1}{2}} V_e)$ *is readily computable. It can be evaluated as*

$$\cos(\mathcal{W}, \mathcal{V}_e) = \lambda_{\max}(ZA), \qquad Z = V_e (V_e^T A V_e)^{-1} V_e^T - V (V^T A V)^{-1} V^T,$$

where the matrix V_e *is chosen such that* $\operatorname{rank} V_e = m$, $\operatorname{Im} V_e = \mathcal{V}_e$.

Proof. The proof is similar to that of Lemma 3.8. ∎

In the following theorem the assumptions of Theorem 4.4 are slightly relaxed.

THEOREM 4.7. *Consider the preconditioner* (12). *Assume that there exist two matrices* $\hat{M}$ *and* $\hat{A}$ *such that* $\hat{M} = \hat{M}^T > 0$, $\hat{A} = \hat{A}^T \geq 0$, $\hat{M} \geq M$, *and* $\hat{A} \leq A$. *(All inequalities here are meant in positive definite sense.) Assume also that* $\{(\hat{\lambda}_i, \hat{\mathbf{v}}_i)\}_{i=1}^n$ *is an ordered set of eigenpairs of* $\hat{M}^{-1}\hat{A}$ *such that* $\hat{\lambda}_1 \leq \dots \leq \hat{\lambda}_n$. *Define the matrix* $\hat{V}_e$ *as* $\hat{V}_e = [\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_m]$. *If the subspaces* $\mathcal{W} = (\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$ *and* $\hat{\mathcal{V}}_e = \operatorname{Im} A^{\frac{1}{2}} \hat{V}_e$ *satisfy the condition* $\cos(\mathcal{W}, \hat{\mathcal{V}}_e) \leq \hat{\gamma}$ *for some* $\hat{\gamma} < 1$, *then the minimal eigenvalue of* $\tilde{B}A$ *is bounded as follows:*

$$\lambda(\tilde{B}A) \geq \max\left\{\lambda_{\min}(M^{-1}A), (1-\hat{\gamma}) \cdot \min\left\{\frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}, \hat{\lambda}_{m+1}\right\}\right\}. \tag{15}$$

Proof. Similarly to the proof of Theorem 4.4,

$$\begin{aligned}
\lambda_{\min}(\tilde{B}A) &= \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T A^{\frac{1}{2}} M^{-1} A^{\frac{1}{2}} \mathbf{x} + \tilde{\sigma} \mathbf{x}^T A^{\frac{1}{2}} V B_V^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\geq \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T A^{\frac{1}{2}} M^{-1} A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} + \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)} \cdot \frac{\mathbf{x}^T A^{\frac{1}{2}} V A_V^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\geq \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T \hat{A}^{\frac{1}{2}} \hat{M}^{-1} \hat{A}^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} + \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)} \cdot \frac{\mathbf{x}^T A^{\frac{1}{2}} V A_V^{-1} V^T A^{\frac{1}{2}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\geq \min\left\{ \hat{\lambda}_{m+1}, \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)} \right\} \cdot \inf_{\mathbf{x}} \left\{ \frac{\mathbf{x}^T \hat{P}_e^{\perp} \mathbf{x} + \mathbf{x}^T P_* \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right\} \\
&\geq (1-\hat{\gamma}) \cdot \min\left\{ \frac{\lambda_{\max}(M^{-1}A)}{\kappa(B_V^{-1}A_V)}, \hat{\lambda}_{m+1} \right\},
\end{aligned}$$

where $\hat{P}_e$, $\hat{P}_e^{\perp}$, P_*, and $P_*^{\perp}$ are the orthogonal projectors onto $\hat{\mathcal{V}}_e = \operatorname{Im} A^{\frac{1}{2}} \hat{V}_e$, $\hat{\mathcal{V}}_e^{\perp} = (\operatorname{Im} A^{\frac{1}{2}} \hat{V}_e)^{\perp}$, $\operatorname{Im} A^{\frac{1}{2}} V$, and $(\operatorname{Im} A^{\frac{1}{2}} V)^{\perp}$, respectively. Combining the above estimate with the result of Theorem 4.4 we obtain (15). ∎

The above theorem shows that there is no need to know the eigenvectors of the matrix $M^{-1}A$ in order to construct the matrix V in the preconditioner (12). It suffices to find the matrices $\hat{M}$ and $\hat{A}$ such that the low-frequency eigensubspace $\hat{\mathcal{V}}_e = \operatorname{span}\{\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_m\}$ of $\hat{M}^{-1}\hat{A}$ is known. Then the matrix V can be defined simply as $V = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_m]$. This approach is taken in the next section, where we construct a number of preconditioners for a class of singularly perturbed elliptic problems by using the known eigensubspace of a degenerate limit problem.

5. Application to second-order elliptic problems with strongly discontinuous and/or anisotropic coefficients. In this section we illustrate the application of the algorithm (12) on a number of severely ill-conditioned model problems involving a parameter.

Example 5.1. Consider the one-dimensional diffusion problem

$$\begin{aligned} (a(x)u')' &= f, \quad x \in [0,1], \\ u'(0) &= u_0', \\ u(1) &= u_1 \end{aligned} \tag{16}$$

discretized by means of conforming piecewise-linear finite elements on a uniform Cartesian mesh with stepsize h. Assume that $a(x) = 1$ for $x < 1/2$ and $a(x) = \varepsilon$ for $x \geq 1/2$. The stiffness matrix of the discrete problem has the following structure:

$$A = h^2 \begin{bmatrix} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & \cdots & \cdots & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 1+\varepsilon & -\varepsilon & & & \\ & & & -\varepsilon & 2\varepsilon & -\varepsilon & & \\ & & & & \cdots & \cdots & & \\ & & & & -\varepsilon & 2\varepsilon & -\varepsilon & \\ & & & & & -\varepsilon & 2\varepsilon & \end{bmatrix}.$$

For this problem the matrices $\hat{A}$ and $\hat{M}$ in Theorem 4.7 can be defined as follows:

$$\hat{A} = \hat{A}_1 = h^2 \begin{bmatrix} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & \cdots & \cdots & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 1 & & & & \\ & & & & \varepsilon & -\varepsilon & & \\ & & & & \cdots & \cdots & & \\ & & & & -\varepsilon & 2\varepsilon & -\varepsilon & \\ & & & & & -\varepsilon & 2\varepsilon & \end{bmatrix}, \tag{17}$$

$$\hat{M} = \hat{M}_1 = 2 \operatorname{diag} \hat{A}_1. \tag{18}$$

As can be easily verified, the following statements hold:

(a) $\hat{A}_1 \leq A$, $\hat{M}_1 \geq \operatorname{diag} A$.

(b) The null-space of $\hat{M}_1^{-1}\hat{A}_1$ consists of the single vector $\mathbf{v}_0 = (1, \ldots, 1, 0, \ldots, 0)$.

(c) On the subspace orthogonal to the null-space of $\hat{M}_1^{-1}\hat{A}_1$ the spectrum of $\hat{M}_1^{-1}\hat{A}_1$ is contained in an interval $[O(h^2), O(1)]$.

Thus, the preconditioner (12) for A can be constructed as

$$\tilde{B} = M^{-1} + \tilde{\sigma} V B_V^{-1} V^T, \qquad M = \operatorname{diag} A, \qquad V = [\mathbf{v}_0].$$

As follows from Theorem 4.7, the spectrum of $\tilde{B}A$ is contained in the interval $[O(h^2), O(1)]$, and the bounds for $\lambda(\tilde{B}A)$ are independent of ε.

Remark 5.2. The above algorithm for constructing the preconditioner (12) can be straightforwardly extended to the case when the dimension of the space is greater than 1.

Remark 5.3. The problems with multiple jumps in the coefficient function and with other types of boundary conditions can be treated analogously. In this case the

null-space of $\hat{M}_1^{-1}\hat{A}_1$ contains the vectors $\mathbf{v}_i$ which are constant along the "subdomains" Ω_i with constant value of $a(x)$, which are not adjacent to the Dirichlet part of the boundary. The matrix V in (12) is then constructed as

$$V = [\mathbf{v}_0, \ldots, \mathbf{v}_m]. \tag{19}$$

Remark 5.4. Alternatively to (17) and (18), we can define the matrices $\hat{A}$ and $\hat{M}$ as

$$\hat{A} = \hat{A}_2 = h^2 \begin{bmatrix} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & \cdots & \cdots & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 1 & & & & \\ & & & & \varepsilon & -\varepsilon & & \\ & & & & \cdots & \cdots & & \\ & & & & -\varepsilon & 2\varepsilon & -\varepsilon & \\ & & & & & -\varepsilon & \varepsilon & \end{bmatrix},$$

$$\hat{M} = \hat{M}_2 = 2\,\mathrm{diag}\,\hat{A}_2,$$

i.e., we can replace all the boundary conditions of Dirichlet type by boundary conditions of Neumann type. In this case the null-space of $\hat{M}_2^{-1}\hat{A}_2$ contains the vectors which are constant along all the "subdomains" Ω_i (even along the "subdomains" which are adjacent to the Dirichlet part of the boundary). If the matrix V in (12) is chosen such that $\mathrm{Im}\,V = \ker \hat{M}_2^{-1}\hat{A}_2$, then the preconditioner "shifts" the low-frequency eigencluster at least as well as (or even better than) the preconditioner with the matrix V defined by (19). The improvement is possible since the smallest nonzero eigenvalue $\lambda^+_{\min}(\hat{M}_2^{-1}\hat{A}_2)$ of $\hat{M}_2^{-1}\hat{A}_2$ is at least as large as the smallest nonzero eigenvalue $\lambda^+_{\min}(\hat{M}_1^{-1}\hat{A}_1)$ of $\hat{M}_1^{-1}\hat{A}_1$ (and in some cases it can be even significantly larger).

Example 5.5. Consider the two-dimensional diffusion problem

$$\begin{aligned} \nabla K(\mathbf{x})\nabla u(\mathbf{x}) &= f(\mathbf{x}) && \text{in} \quad \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= 0 && \text{on} \quad \mathbf{x} \in \Gamma_D \subseteq \partial\Omega, \\ \partial u(\mathbf{x})/\partial \mathbf{n} &= 0 && \text{on} \quad \mathbf{x} \in \Gamma_N = \partial\Gamma/\Gamma_D \end{aligned}$$

with highly anisotropic diffusion tensor $K(\mathbf{x})$

$$K(\mathbf{x}) = \begin{bmatrix} \varepsilon & 0 \\ 0 & 1 \end{bmatrix}, \qquad 0 < \varepsilon \ll 1,$$

discretized by means of conforming piecewise-linear finite elements on a uniform Cartesian grid. For this problem the auxiliary matrices $\hat{A}$ and $\hat{M}$ in Theorem 4.7 can be chosen as follows: the matrix $\hat{A}$ can be defined to be the stiffness matrix of the degenerate limit problem with $\varepsilon = 0$, and the matrix $\hat{M}$ can be defined as $\hat{M} = 2\,\mathrm{diag}\,\hat{A}$. Clearly, $\hat{A} \le A$, $\hat{M} \ge M = \mathrm{diag}\,A$. With this choice of $\hat{A}$ and $\hat{M}$ the null-space of $\hat{M}^{-1}\hat{A}$ is known: it consists of constant vectors aligned with the y-axis (except the constant vectors which are adjacent with the Dirichlet part of the boundary). The spectrum of $\hat{M}^{-1}\hat{A}$ on the subspace orthogonal to the kernel is contained in the interval $[O(h^2), O(1)]$. Thus, if we choose the matrix V such that $\mathrm{Im}\,V = \ker \hat{M}^{-1}\hat{A}$,

then the spectrum of $\tilde{B}A$ is contained in the interval $[O(h^2), O(1)]$ and is bounded independently on ε.

Remark 5.6. As can be easily verified, with the above choice of V ($\operatorname{Im} V = \ker \hat{M}^{-1}\hat{A}$) the condition number $\kappa(A_V)$ of $A_V = V^T A V$ is of order $O(h^{-2})$, i.e., it is of the same order as the effective condition number $\kappa^+(\hat{M}^{-1}\hat{A})$ of $\hat{M}^{-1}\hat{A}$, $\kappa^+(\hat{M}^{-1}\hat{A}) = \lambda_{\max}(\hat{M}^{-1}\hat{A})/\lambda^+_{\min}(\hat{M}^{-1}\hat{A})$. Thus, the preconditioner B_V for A_V can be constructed as $B_V = I$ or $B_V = \operatorname{diag} A_V$; see Remark 3.4. Such a choice of B_V allows an efficient parallel implementation of the preconditioner (12).

Remark 5.7. A similar approach for constructing the matrices V and B_V can also be applied in the three-dimensional case. If the diffusion tensor $K(\mathbf{x})$ has the form

$$K(\mathbf{x}) = \begin{bmatrix} \varepsilon^\beta & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad 0 < \varepsilon < 1, \qquad \beta \geq 1,$$

then the preconditioner can be constructed by applying the above-described algorithm recursively: first to the matrix A and then to the matrix A_V. As in the two-dimensional case, the resulting preconditioner can be efficiently parallelized.

Remark 5.8. We can define the matrix $\hat{A}$ in Theorem 4.7 to be the stiffness matrix which corresponds to the problem not only with $\varepsilon = 0$, but also with $\Gamma_N = \partial\Gamma$. In this case the subspace $\operatorname{Im} V$ contains all the constant vectors aligned with the y-axis.

Remark 5.9. The above algorithm for the anisotropic problems can be easily combined with the algorithm for the discontinuous problems. This allows us to treat the problems where the diffusion tensor is not only anisotropic but also discontinuous.

6. A purely algebraic algorithm for constructing the matrix V. For the class of diffusion-type problems considered in the previous section the matrix V can be constructed automatically using a heuristic technique developed in [3, 14] (see also [15] for a similar approach). For the sake of completeness a brief description of the algorithm follows.

Consider the diffusion problem as in Example 5.5:

$$\begin{array}{rcll} \nabla K(\mathbf{x}) \nabla u(\mathbf{x}) & = & f(\mathbf{x}) & \text{in} \quad \mathbf{x} \in \Omega, \\ u(\mathbf{x}) & = & 0 & \text{on} \quad \mathbf{x} \in \Gamma_D \subseteq \partial\Omega, \\ \partial u(\mathbf{x})/\partial \mathbf{n} & = & 0 & \text{on} \quad \mathbf{x} \in \Gamma_N = \partial\Gamma / \Gamma_D \end{array} \tag{20}$$

discretized on a regular finite element mesh. Assume that the diffusion tensor $K(\mathbf{x})$ is piecewise constant and uniformly symmetric positive definite.

Let $A = [a_{i,j}]_{i,j=1}^n$ be the stiffness matrix resulting from the discretization of (20). Define the matrix $Q = [q_{i,j}]_{i,j=1}^n$ which contains a pattern of "strong couplings" within A:

$$q_{i,j} = \begin{cases} 0 & \text{if } |a_{i,j}| < \omega \cdot \min \left\{ \max\limits_{\substack{k=1,n \\ k \neq i}} |a_{i,k}|, \max\limits_{\substack{k=1,n \\ k \neq j}} |a_{k,j}| \right\}, \qquad \omega \in (0,1), \\ 1 & \text{otherwise.} \end{cases} \tag{21}$$

Define a symmetric function $\chi(i,j)$ of two integer variables i and j: let the function $\chi(i,j)$ be equal to unity either if $q_{i,j} = 1$ or if there exists a k such that $\chi(i,k) \cdot \chi(k,j) = 1$; otherwise define the function $\chi(i,j)$ to be equal to zero. As can

be readily seen, the definition of $\chi(i,j)$ implies that $\chi(i,j)=1$ if and only if there is a "strong connectivity path" between the unknowns i and j; otherwise $\chi(i,j)=0$.

Define also a number of sets $G^{(p)}$ of size n_p:

$$G^{(p)} = \{i_1^{(p)}, \ldots, i_{n_p}^{(p)}\}, \qquad i_s^{(p)} \in \{1,2,\ldots,n\}, \qquad p = 1,\ldots,\tilde{m},$$

such that they satisfy the conditions

- $G^{(p_1)} \bigcap G^{(p_2)} = \{\emptyset\}$ for all $p_1 \neq p_2$ and
- for any i and j there exists p such that $i \in G^{(p)}$ and $j \in G^{(p)}$ if and only if $\chi(i,j)=1$.

As follows from the above definition, each set $G^{(p)}$ contains a list of "strongly connected unknowns" (with respect to A). The definition of $G^{(p)}$ also implies that if there is no "strong connectivity path" from i to j, then the unknowns i and j belong to different sets. If there is a "strong connectivity path" between the unknowns i and j, then they belong to the same set $G^{(p)}$. As can be readily shown, the sets $G^{(p)}$ can be computed with an arithmetic cost $O(n)$.

Define a set of $\tilde{m}$ sparse vectors $\hat{\mathbf{w}}^{(p)}$ of size n:

$$\hat{\mathbf{w}}_i^{(p)} = 1 \quad \text{if } i \in G^{(p)}; \quad \text{otherwise } \hat{\mathbf{w}}_i^{(p)} = 0. \tag{22}$$

Clearly, the vectors $\hat{\mathbf{w}}^{(p)}$ are L_2-orthogonal to each other. Next, define a vector $\mathbf{h}$ such that

$$\mathbf{h}_i = 1 \quad \text{if} \quad \left| \frac{\sum a_{i,j}}{a_{i,i}} \right| > \omega; \qquad \mathbf{h}_i = 0 \quad \text{otherwise.} \tag{23}$$

Define a set $\mathcal{X}$ of indices $p_i,\ i=1,\ldots,m,\ m \leq \tilde{m}$, such that

$$p_i \in \mathcal{X} \quad \text{if and only if} \quad \mathbf{h}^T \hat{\mathbf{w}}^{(p_i)} = 0. \tag{24}$$

Remark 6.1. The algorithm (23)–(24) selects only those sets $G^{(p_i)}$, which are "weakly connected" with the Dirichlet part of the boundary (see the previous section for the motivation).

Define the matrices V_1 and V_2 as follows:

$$\begin{aligned} V_1 &= \left[\tilde{\mathbf{w}}^{(p_1)}, \ldots, \tilde{\mathbf{w}}^{(p_m)}\right], \qquad p_i \in \mathcal{X},\ i=1,\ldots,m, \\ V_2 &= \left[\tilde{\mathbf{w}}^{(1)}, \ldots, \tilde{\mathbf{w}}^{(\tilde{m})}\right], \qquad i=1,\ldots,\tilde{m}. \end{aligned} \tag{25}$$

Finally, define the matrix V in the preconditioner (12) as either $V = V_1$ or $V = V_2$.

As was demonstrated in the previous section, both $\operatorname{Im} V_1$ and $\operatorname{Im} V_2$ approximate well the low-frequency eigensubspace of nearly degenerate diffusion-type operators. A nice feature of the choice $V = V_1$ is that it leads to a smaller condition number of $A_V = V^T A V$. In many practical cases this allows an easier construction of B_V. It should be noted, however, that in the case $V = V_2$ the smallest eigenvalues of A could be captured more efficiently than in the case $V = V_1$ since $\operatorname{Im} V_1 \subseteq \operatorname{Im} V_2$ (see Lemma 2.3).

As follows from the definition of $\{\tilde{\mathbf{w}}^{(i)}\}$, the matrix V is sparse and contains at most n nonzero entries. This means that if the action of B_V^{-1} requires $O(n)$ arithmetic operations, then the action of the whole preconditioner (12) also requires only $O(n)$ operations.

Another important feature of the developed preconditioner is that it can be efficiently parallelized since in many practical applications it suffices to use a (block-)

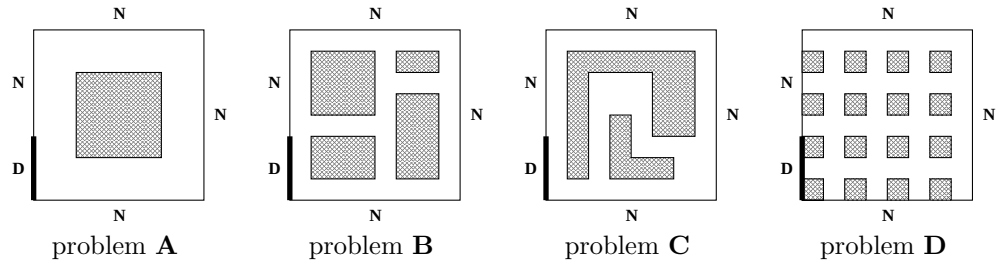


FIG. 1. *Test problems used in our numerical experiments.*

diagonal preconditioner B_V^{-1} for A_V^{-1}. As can be easily verified, if we distribute the algebraic system between the processors in the multiprocessor system such that the unknowns from the same group $G^{(p)}$ belong to the same processor and distribute the blocks of B_V accordingly, then no interprocessor communications are needed to perform the multiplication with $VB_V^{-1}V^T$ (if the matrix B_V is block-diagonal and the blocks are properly distributed).

7. Numerical experiments. In this section we illustrate the numerical performance of the developed technique on a number of singularly perturbed elliptic problems of the form given in Example 5.5. Namely, we consider piecewise-linear conforming finite element discretization of the diffusion equation (20) with $\Omega = [0,1]^2$, $\Gamma_D = \{\mathbf{x} = (x,y) : x = 0,\ 0 \le y < \bar{y} \le 1\}$, and $\Gamma_N = \partial\Omega/\Gamma_D$ on a uniform Cartesian grid. The diffusion tensor $K(\mathbf{x})$ is considered to be of the form

$$K(\mathbf{x}) = a(\mathbf{x}) \cdot \begin{bmatrix} \varepsilon & 0 \\ 0 & 1 \end{bmatrix}, \quad a(\mathbf{x}) > 0, \quad \varepsilon > 0.$$

The value of ε is chosen to be equal to 1, 10^3, or 10^6. The coefficient function $a(\mathbf{x})$ is assumed to be the following:

$$a(\mathbf{x}) = \begin{cases} a & \text{if } \mathbf{x} \text{ belongs to the shaded area (see Figure 1),} \\ 1 & \text{otherwise,} \end{cases}$$

where the value of a is chosen to be either 10^{-6}, 10^{-3}, 1, 10^3, or 10^6.

The main concern is to demonstrate insensitivity of the developed algorithm with respect to the problem parameters. The results of our numerical experiments are presented in Tables 1–12, where we study the performance of the method with respect to variations of a, ε, $\bar{y}$ and different modifications of the preconditioner (12). The performance of the diagonal (pointwise Jacobi) preconditioner is also presented for comparison. The stopping criterion within the PCG method is chosen to be $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-6}$, where $\mathbf{r}^{(0)}$ is the initial residual and $\mathbf{r}^{(k)}$ is the residual after the kth iteration. The right-hand side in the algebraic system is chosen to be random. The matrix V in (12) is constructed automatically using the heuristic algorithm (21)–(25) with $\omega = 0.1$.

Tables 1–4 show that for the problem with smooth isotropic coefficient function the convergence rate of the diagonally preconditioned PCG method depends mildly on the choice of the boundary conditions, whereas the situation is opposite in the case when the coefficient function is highly anisotropic. Tables 5, 7, 9, and 11 show that the jumps in the coefficient function have the effect of adding extra (internal) Neumann-type boundary conditions, which again leads to a slower convergence of

TABLE 1

Problems **A**, **B**, **C**, *and* **D**, $h = \frac{1}{64}$, $a = 1$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) for different values of* ε, $\bar{y}$ *and different choices of the preconditioner.*

	Preconditioner				
		$V = V_2$	$V = V_1$		
	$\hat{B} = (\operatorname{diag} A)^{-1}$	$B_V = A_V$	$B_V = A_V$	$B_V = I$	$B_V = \operatorname{diag} A_V$
$\bar{y}$	Anisotropy: $\varepsilon = 1$ (isotropic case)				
0.125	241 (-)	203 (1)	241 (0)	241 (0)	241 (0)
0.250	239 (-)	238 (1)	239 (0)	239 (0)	239 (0)
0.375	239 (-)	239 (1)	239 (0)	239 (0)	239 (0)
0.500	234 (-)	232 (1)	234 (0)	234 (0)	234 (0)
0.625	225 (-)	223 (1)	225 (0)	225 (0)	225 (0)
0.750	222 (-)	220 (1)	222 (0)	222 (0)	222 (0)
0.875	212 (-)	209 (1)	212 (0)	212 (0)	212 (0)
1.000	179 (-)	117 (1)	179 (0)	179 (0)	179 (0)
$\bar{y}$	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
0.125	1910 (-)	237 (65)	331 (57)	346 (57)	347 (57)
0.250	1781 (-)	243 (65)	423 (49)	417 (49)	421 (49)
0.375	1548 (-)	241 (65)	420 (41)	414 (41)	414 (41)
0.500	1290 (-)	235 (65)	405 (33)	407 (33)	410 (33)
0.625	1032 (-)	226 (65)	398 (25)	396 (25)	387 (25)
0.750	805 (-)	221 (65)	389 (17)	385 (17)	384 (17)
0.875	564 (-)	219 (65)	362 (9)	360 (9)	359 (9)
1.000	284 (-)	162 (65)	262 (1)	262 (1)	262 (1)

TABLE 2

Problem **A**, $h = \frac{1}{64}$, $a = 10^3$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) for different values of* ε, $\bar{y}$ *and different choices of the preconditioner.*

	Preconditioner				
		$V = V_2$	$V = V_1$		
	$\hat{B} = (\operatorname{diag} A)^{-1}$	$B_V = A_V$	$B_V = A_V$	$B_V = I$	$B_V = \operatorname{diag} A_V$
$\bar{y}$	Anisotropy: $\varepsilon = 1$ (isotropic case)				
0.125	310 (-)	195 (2)	216 (1)	216 (1)	216 (1)
0.250	307 (-)	194 (2)	214 (1)	214 (1)	214 (1)
0.375	308 (-)	193 (2)	213 (1)	213 (1)	213 (1)
0.500	303 (-)	205 (2)	219 (1)	219 (1)	219 (1)
0.625	298 (-)	205 (2)	217 (1)	217 (1)	217 (1)
0.750	290 (-)	194 (2)	207 (1)	207 (1)	207 (1)
0.875	280 (-)	182 (2)	197 (1)	197 (1)	197 (1)
1.000	260 (-)	160 (2)	171 (1)	171 (1)	171 (1)
$\bar{y}$	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
0.125	2114 (-)	275 (131)	415 (123)	905 (123)	477 (123)
0.250	2007 (-)	294 (131)	559 (115)	808 (115)	520 (115)
0.375	1917 (-)	252 (131)	480 (107)	755 (107)	422 (107)
0.500	1884 (-)	246 (131)	480 (99)	700 (99)	421 (99)
0.625	1877 (-)	247 (131)	474 (91)	723 (91)	422 (91)
0.750	1851 (-)	233 (131)	474 (83)	728 (83)	416 (83)
0.875	1648 (-)	251 (131)	471 (75)	595 (75)	429 (75)
1.000	1382 (-)	245 (131)	471 (67)	420 (67)	406 (67)

TABLE 3

Problem **C**, $h = \frac{1}{64}$, $a = 10^3$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) for different values of* ε, $\bar{y}$ *and different choices of the preconditioner.*

	Preconditioner				
		$V = V_2$	$V = V_1$		
	$\hat{B} = (\operatorname{diag} A)^{-1}$	$B_V = A_V$	$B_V = A_V$	$B_V = I$	$B_V = \operatorname{diag} A_V$
$\bar{y}$	Anisotropy: $\varepsilon = 1$ (isotropic case)				
0.125	400 (-)	267 (3)	288 (2)	289 (2)	292 (2)
0.250	397 (-)	263 (3)	284 (2)	281 (2)	283 (2)
0.375	394 (-)	262 (3)	279 (2)	276 (2)	279 (2)
0.500	393 (-)	262 (3)	275 (2)	272 (2)	273 (2)
0.625	389 (-)	260 (3)	271 (2)	268 (2)	272 (2)
0.750	388 (-)	262 (3)	273 (2)	266 (2)	271 (2)
0.875	382 (-)	262 (3)	271 (2)	268 (2)	272 (2)
1.000	380 (-)	260 (3)	268 (2)	265 (2)	268 (2)
$\bar{y}$	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
0.125	3410 (-)	270 (261)	423 (253)	611 (253)	445 (253)
0.250	3275 (-)	240 (261)	394 (245)	562 (245)	395 (245)
0.375	3111 (-)	233 (261)	388 (237)	542 (237)	386 (237)
0.500	3087 (-)	222 (261)	380 (229)	544 (229)	377 (229)
0.625	2870 (-)	220 (261)	379 (221)	535 (221)	369 (221)
0.750	2918 (-)	222 (261)	379 (213)	506 (213)	341 (213)
0.875	2766 (-)	225 (261)	380 (205)	506 (205)	341 (205)
1.000	2620 (-)	232 (261)	463 (197)	392 (197)	408 (197)

TABLE 4

Problem **D**, $h = \frac{1}{64}$, $a = 10^3$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) for different values of* ε, $\bar{y}$ *and different choices of the preconditioner.*

	Preconditioner				
		$V = V_2$	$V = V_1$		
	$\hat{B} = (\operatorname{diag} A)^{-1}$	$B_V = A_V$	$B_V = A_V$	$B_V = I$	$B_V = \operatorname{diag} A_V$
$\bar{y}$	Anisotropy: $\varepsilon = 1$ (isotropic case)				
0.125	606 (-)	148 (17)	153 (16)	172 (16)	174 (16)
0.250	606 (-)	179 (17)	191 (15)	186 (15)	165 (15)
0.375	603 (-)	178 (17)	183 (14)	181 (14)	183 (14)
0.500	603 (-)	178 (17)	183 (14)	177 (14)	184 (14)
0.625	563 (-)	173 (17)	180 (13)	176 (13)	179 (13)
0.750	560 (-)	170 (17)	172 (13)	168 (13)	173 (13)
0.875	515 (-)	159 (17)	158 (12)	158 (12)	168 (12)
1.000	512 (-)	154 (17)	156 (12)	156 (12)	156 (12)
$\bar{y}$	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
0.125	3709 (-)	199 (317)	207 (309)	965 (309)	508 (309)
0.250	4128 (-)	239 (317)	377 (301)	988 (301)	610 (301)
0.375	3964 (-)	229 (317)	335 (293)	913 (293)	546 (293)
0.500	3760 (-)	247 (317)	464 (285)	831 (285)	612 (285)
0.625	3461 (-)	211 (317)	339 (277)	691 (277)	470 (277)
0.750	2812 (-)	247 (317)	433 (269)	586 (269)	458 (269)
0.875	2111 (-)	205 (317)	334 (261)	455 (261)	326 (261)
1.000	1553 (-)	229 (317)	398 (253)	356 (253)	358 (253)

TABLE 5

Problem $\mathbf{A}$, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) as a function of the coefficient jump* a *and the anisotropy ratio* ε.

	Coefficient jump a				
	10^{-6}	10^{-3}	1	10^{3}	10^{6}
	Anisotropy: $\varepsilon = 1$ (isotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	282 (-)	281 (-)	239 (-)	304 (-)	382 (-)
$B_V = A_V$, V_2	275 (2)	276 (2)	239 (1)	193 (2)	228 (2)
$B_V = A_V$, V_1	285 (1)	282 (1)	239 (0)	224 (1)	232 (1)
$B_V = I$, V_1	271 (1)	288 (1)	239 (0)	214 (1)	240 (1)
$B_V = \operatorname{diag} A_V$, V_1	285 (1)	286 (1)	239 (0)	210 (1)	240 (1)
	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	2351 (-)	2335 (-)	1548 (-)	1911 (-)	2146 (-)
$B_V = A_V$, V_2	257 (127)	265 (127)	241 (65)	252 (131)	300 (131)
$B_V = A_V$, V_1	455 (103)	467 (103)	420 (41)	480 (107)	582 (107)
$B_V = I$, V_1	425 (103)	428 (103)	414 (41)	758 (107)	1171 (107)
$B_V = \operatorname{diag} A_V$, V_1	429 (103)	428 (103)	414 (41)	422 (107)	658 (107)
	Anisotropy: $\varepsilon = 10^6$ (strongly anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	9266 (-)	7916 (-)	4882 (-)	6173 (-)	5923 (-)
$B_V = A_V$, V_2	195 (127)	195 (127)	115 (65)	150 (131)	150 (131)
$B_V = A_V$, V_1	269 (103)	257 (103)	161 (41)	243 (107)	215 (107)
$B_V = I$, V_1	312 (103)	562 (103)	170 (41)	2086 (107)	2501 (107)
$B_V = \operatorname{diag} A_V$, V_1	288 (103)	436 (103)	170 (41)	451 (107)	254 (107)

TABLE 6

Problem $\mathbf{A}$, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *relative arithmetic cost needed to construct the preconditioner* (12) *(i.e., to compute* M, $\tilde{\sigma}$, V, B_V, *as well as to factorize* B_V*) and to multiply* (12) *by a vector as a function of the coefficient jump* a *and the anisotropy ratio* ε*; the arithmetic cost needed to multipy the stiffness matrix* A *by a vector is taken as a reference.*

	Coefficient jump a					
	$10^{-3}/10^{-6}$		1		$10^{3}/10^{6}$	
	Anisotropy: $\varepsilon = 1$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	3.43	0.75	3.42	0.68	3.43	0.75
$B_V = A_V$, V_1	2.30	0.33	1.92	0.30	2.35	0.34
$B_V = I$, V_1	2.30	0.33	1.92	0.30	2.35	0.34
$B_V = \operatorname{diag} A_V$, V_1	2.30	0.33	1.92	0.30	2.35	0.34
	Anisotropy: $\varepsilon = 10^3/10^6$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	5.92	0.81	3.54	0.77	6.16	0.81
$B_V = A_V$, V_1	4.73	0.67	2.86	0.60	4.92	0.68
$B_V = I$, V_1	2.92	0.63	2.74	0.59	2.94	0.63
$B_V = \operatorname{diag} A_V$, V_1	2.96	0.63	2.75	0.59	2.98	0.63

the diagonally preconditioned iterative scheme. To the contrary, the PCG method preconditioned by means of (12) exhibits robust performance in a wide range of a and ε and is insensitive to the choice of the boundary conditions; see Tables 1–4, 5, 7, 9, and 11.

Tables 6, 8, 10, and 12 show that the expense for construction and applying the preconditioner, including the construction of matrix V, is in general very small.

Numerical experiments demonstrate that the developed subspace-correction technique performs well even if the matrix A_V is replaced by a simple diagonal precondi-

TABLE 7

Problem **B**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) as a function of the coefficient jump* a *and the anisotropy ratio* ε.

	Coefficient jump a				
	10^{-6}	10^{-3}	1	10^{3}	10^{6}
	Anisotropy: $\varepsilon = 1$ (isotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	335 (-)	333 (-)	239 (-)	431 (-)	619 (-)
$B_V = A_V$, V_2	317 (5)	320 (5)	239 (1)	177 (5)	197 (5)
$B_V = A_V$, V_1	338 (4)	338 (4)	239 (0)	192 (4)	208 (4)
$B_V = I$, V_1	332 (4)	332 (4)	239 (0)	189 (4)	201 (4)
$B_V = \operatorname{diag} A_V$, V_1	337 (4)	347 (4)	239 (0)	192 (4)	208 (4)
	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	3064 (-)	2779 (-)	1548 (-)	3015 (-)	4637 (-)
$B_V = A_V$, V_2	239 (217)	242 (217)	241 (65)	232 (233)	278 (233)
$B_V = A_V$, V_1	400 (193)	414 (193)	420 (41)	384 (209)	422 (209)
$B_V = I$, V_1	380 (193)	392 (193)	414 (41)	571 (209)	1260 (209)
$B_V = \operatorname{diag} A_V$, V_1	387 (193)	384 (193)	414 (41)	395 (209)	902 (209)
	Anisotropy: $\varepsilon = 10^6$ (strongly anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	13314 (-)	12345 (-)	4882 (-)	10526 (-)	12225 (-)
$B_V = A_V$, V_2	219 (217)	234 (217)	115 (65)	164 (233)	197 (233)
$B_V = A_V$, V_1	307 (193)	303 (193)	161 (41)	249 (209)	279 (209)
$B_V = I$, V_1	318 (193)	801 (193)	170 (41)	2000 (209)	1689 (209)
$B_V = \operatorname{diag} A_V$, V_1	321 (193)	659 (193)	170 (41)	599 (209)	323 (209)

TABLE 8

Problem **B**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *relative arithmetic cost needed to construct the preconditioner* (12) *(i.e., to compute* M, $\tilde{\sigma}$, V, B_V, *as well as to factorize* B_V*) and to multiply* (12) *by a vector as a function of the coefficient jump* a *and the anisotropy ratio* ε*; the arithmetic cost needed to multipy the stiffness matrix* A *by a vector is taken as a reference.*

	Coefficient jump a					
	$10^{-3}/10^{-6}$		1		$10^{3}/10^{6}$	
	Anisotropy: $\varepsilon = 1$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	3.44	0.75	3.42	0.68	3.44	0.75
$B_V = A_V$, V_1	2.49	0.46	1.92	0.30	2.62	0.49
$B_V = I$, V_1	2.49	0.46	1.92	0.30	2.62	0.49
$B_V = \operatorname{diag} A_V$, V_1	2.49	0.46	1.92	0.30	2.62	0.49
	Anisotropy: $\varepsilon = 10^3/10^6$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	21.71	0.91	3.54	0.77	30.03	0.93
$B_V = A_V$, V_1	15.82	0.81	2.86	0.60	21.65	0.83
$B_V = I$, V_1	3.17	0.69	2.74	0.59	3.21	0.70
$B_V = \operatorname{diag} A_V$, V_1	3.26	0.69	2.75	0.59	3.31	0.70

tioner B_V; in many practical cases it suffices to take $B_V = \operatorname{diag} A_V$. However, if the matrix A_V is severely ill-conditioned, special care has to be taken when constructing the preconditioner B_V; one of the possible approaches was mentioned in Remark 5.7, alternatively one can use an incomplete factorization procedure to construct an approximation to A_V^{-1}. In a multilevel setting the matrix B_V can be constructed by using the algorithm (12) recursively; in this case we obtain a preconditioner of the form $B_k = M_k^{-1} + \sigma_k V_{k,k-1} B_{k-1} V_{k,k-1}^T$; see sections 3 and 4.

In Figures 2 and 3 we also illustrate the eigenvalue distribution of the preconditioned matrix $\hat{B}A$ for different a, ε, and $\hat{B}$. As one can see from the figures, the spectrum of the system preconditioned by (12) is contained in the interval $[O(h^2), O(1)]$,

TABLE 9

Problem **C**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) as a function of the coefficient jump* a *and the anisotropy ratio* ε.

	Coefficient jump a				
	10^{-6}	10^{-3}	1	10^{3}	10^{6}
	Anisotropy: $\varepsilon = 1$ (isotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	384 (-)	381 (-)	239 (-)	394 (-)	527 (-)
$B_V = A_V$, V_2	384 (3)	384 (3)	239 (1)	263 (3)	281 (3)
$B_V = A_V$, V_1	391 (2)	388 (2)	239 (0)	279 (2)	289 (2)
$B_V = I$, V_1	382 (2)	383 (2)	239 (0)	277 (2)	297 (2)
$B_V = \operatorname{diag} A_V$, V_1	390 (2)	385 (2)	239 (0)	279 (2)	294 (2)
	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	3777 (-)	3168 (-)	1548 (-)	3113 (-)	3771 (-)
$B_V = A_V$, V_2	247 (253)	251 (253)	241 (65)	233 (261)	257 (261)
$B_V = A_V$, V_1	405 (229)	413 (229)	420 (41)	387 (237)	409 (237)
$B_V = I$, V_1	444 (229)	402 (229)	414 (41)	553 (237)	934 (237)
$B_V = \operatorname{diag} A_V$, V_1	434 (229)	394 (229)	414 (41)	387 (237)	663 (237)
	Anisotropy: $\varepsilon = 10^6$ (strongly anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	15007 (-)	14853 (-)	4882 (-)	13727 (-)	13274 (-)
$B_V = A_V$, V_2	213 (253)	227 (253)	115 (65)	161 (261)	184 (261)
$B_V = A_V$, V_1	298 (229)	301 (229)	161 (41)	273 (237)	287 (237)
$B_V = I$, V_1	334 (229)	947 (229)	170 (41)	2218 (237)	1653 (237)
$B_V = \operatorname{diag} A_V$, V_1	322 (229)	693 (229)	170 (41)	726 (237)	324 (237)

TABLE 10

Problem **C**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *relative arithmetic cost needed to construct the preconditioner* (12) *(i.e., to compute* M, $\tilde{\sigma}$, V, B_V, *as well as to factorize* B_V*) and to multiply* (12) *by a vector as a function of the coefficient jump* a *and the anisotropy ratio* ε*; the arithmetic cost needed to multipy the stiffness matrix* A *by a vector is taken as a reference.*

	Coefficient jump a					
	$10^{-3}/10^{-6}$		1		$10^{3}/10^{6}$	
	Anisotropy: $\varepsilon = 1$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	3.43	0.75	3.42	0.68	3.43	0.75
$B_V = A_V$, V_1	2.41	0.43	1.92	0.30	2.54	0.47
$B_V = I$, V_1	2.41	0.43	1.92	0.30	2.54	0.47
$B_V = \operatorname{diag} A_V$, V_1	2.41	0.43	1.92	0.30	2.54	0.47
	Anisotropy: $\varepsilon = 10^3/10^6$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	44.56	0.98	3.54	0.77	46.60	0.99
$B_V = A_V$, V_1	34.43	0.88	2.86	0.60	35.08	0.89
$B_V = I$, V_1	3.18	0.69	2.74	0.59	3.22	0.70
$B_V = \operatorname{diag} A_V$, V_1	3.29	0.69	2.75	0.59	3.34	0.70

and the bounds are independent of ε and a, whereas in the case of Jacobi preconditioning the spectrum normally contains a number of extremely small eigenvalues, sometimes well separated from the remainder of the spectrum, which may cause slow convergence of the PCG algorithm.

The results of our numerical experiments are in strong agreement with the developed theory. Taking into account that the computational overhead associated with

TABLE 11

Problem **D**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *PCG iteration count, and the dimension of* $\operatorname{Im} V$ *(in parentheses) as a function of the coefficient jump* a *and the anisotropy ratio* ε.

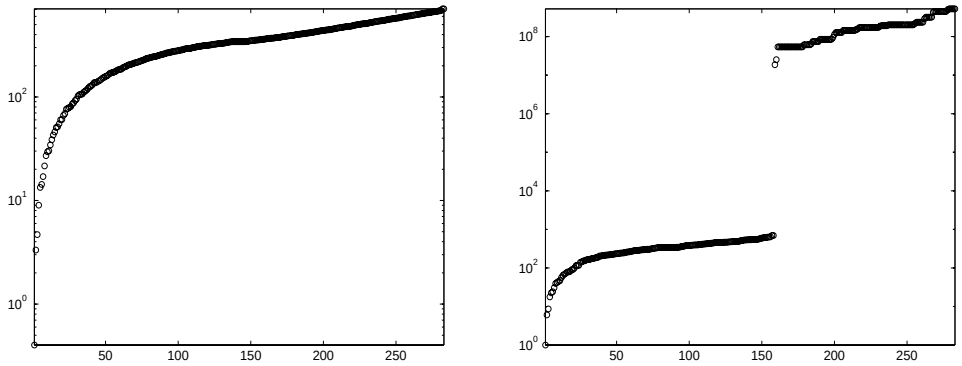
	Coefficient jump a				
	10^{-6}	10^{-3}	1	10^{3}	10^{6}
	Anisotropy: $\varepsilon = 1$ (isotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	292 (-)	259 (-)	239 (-)	602 (-)	984 (-)
$B_V = A_V$, V_2	240 (17)	244 (17)	239 (1)	178 (17)	205 (17)
$B_V = A_V$, V_1	299 (14)	299 (14)	239 (0)	183 (14)	203 (14)
$B_V = I$, V_1	304 (14)	303 (14)	239 (0)	179 (14)	183 (14)
$B_V = \operatorname{diag} A_V$, V_1	311 (14)	312 (14)	239 (0)	183 (14)	181 (14)
	Anisotropy: $\varepsilon = 10^3$ (anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	2495 (-)	2432 (-)	1548 (-)	3852 (-)	7422 (-)
$B_V = A_V$, V_2	238 (268)	246 (268)	241 (65)	229 (317)	273 (317)
$B_V = A_V$, V_1	394 (244)	418 (244)	420 (41)	334 (293)	381 (293)
$B_V = I$, V_1	391 (244)	383 (244)	414 (41)	914 (293)	3099 (293)
$B_V = \operatorname{diag} A_V$, V_1	386 (244)	390 (244)	414 (41)	540 (293)	2099 (293)
	Anisotropy: $\varepsilon = 10^6$ (strongly anisotropic case)				
$\hat{B} = (\operatorname{diag} A)^{-1}$	10032 (-)	13016 (-)	4882 (-)	18742 (-)	n/a
$B_V = A_V$, V_2	152 (268)	149 (268)	115 (65)	116 (317)	n/a
$B_V = A_V$, V_1	278 (244)	276 (244)	161 (41)	187 (293)	n/a
$B_V = I$, V_1	283 (244)	832 (244)	170 (41)	3559 (293)	n/a
$B_V = \operatorname{diag} A_V$, V_1	302 (244)	588 (244)	170 (41)	785 (293)	n/a

TABLE 12

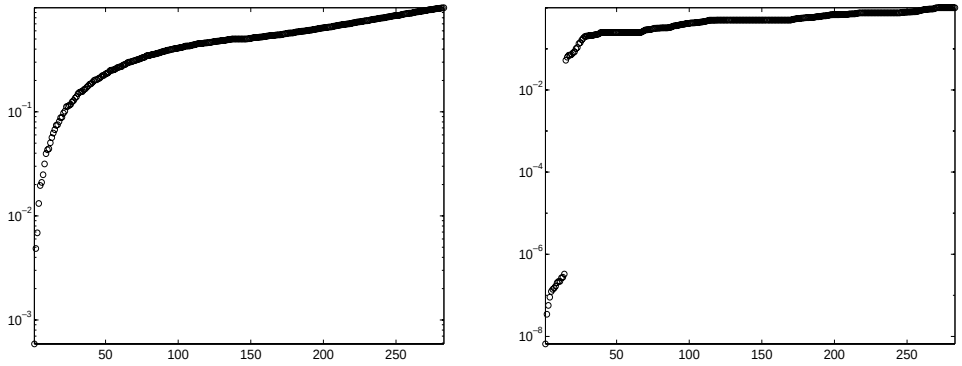
Problem **D**, $h = \frac{1}{64}$, $\bar{y} = \frac{3}{8}$, *relative arithmetic cost needed to construct the preconditioner* (12) *(i.e., to compute* M, $\tilde{\sigma}$, V, B_V, *as well as to factorize* B_V*) and to multiply* (12) *by a vector as a function of the coefficient jump* a *and the anisotropy ratio* ε*; the arithmetic cost needed to multipy the stiffness matrix* A *by a vector is taken as a reference.*

	Coefficient jump a					
	$10^{-3}/10^{-6}$		1		$10^{3}/10^{6}$	
	Anisotropy: $\varepsilon = 1$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	3.63	0.76	3.42	0.68	3.64	0.76
$B_V = A_V$, V_1	2.19	0.38	1.92	0.30	2.35	0.42
$B_V = I$, V_1	2.19	0.38	1.92	0.30	2.35	0.42
$B_V = \operatorname{diag} A_V$, V_1	2.19	0.38	1.92	0.30	2.35	0.42
	Anisotropy: $\varepsilon = 10^3/10^6$					
	Initialize	Apply	Initialize	Apply	Initialize	Apply
$B_V = A_V$, V_2	68.66	1.06	3.54	0.77	88.77	1.08
$B_V = A_V$, V_1	30.52	0.90	2.86	0.60	41.94	0.94
$B_V = I$, V_1	3.18	0.69	2.74	0.59	3.24	0.70
$B_V = \operatorname{diag} A_V$, V_1	3.30	0.69	2.75	0.59	3.38	0.70

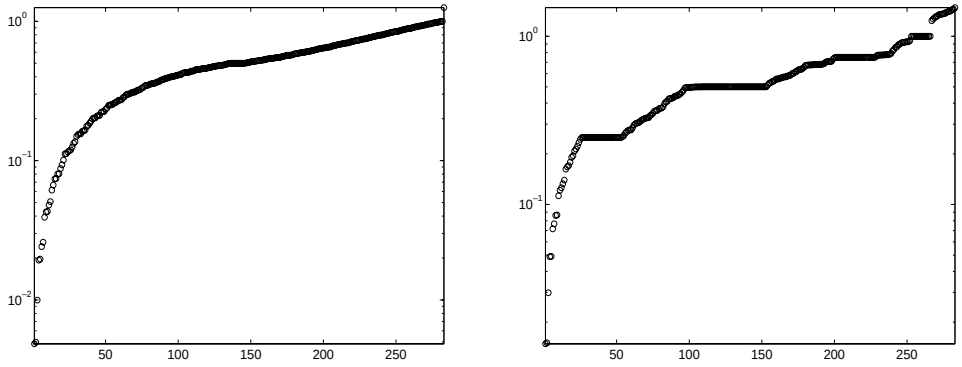
the preconditioner is very low (especially in the case when the matrix B_V is chosen to be diagonal) we conclude that the developed algorithm could be viewed as a viable option when constructing efficient solvers for the considered class of ill-conditioned elliptic problems. Note also that the method is even more attractive in a parallel environment, where it can be a serious competitor to more advanced methods (of multigrid/multilevel type, for instance) as it requires only a small amount of inter-processor communications.



No preconditioning: $\hat{B} = I$



Diagonal scaling: $\hat{B} = (\operatorname{diag} A)^{-1}$



Algorithm (12): $\hat{B} = (\operatorname{diag} A)^{-1} + \sigma V B_V^{-1} V^T,\ B_V = A_V,\ V = V_2$

FIG. 2. *Problem* **D**, $h = 1/16$, $\varepsilon = 1$, $\bar{y} = \frac{3}{8}$, *eigenvalue distribution of* $\hat{B}A$ *for different preconditioners* $\hat{B}$ *and different values of the coefficient jump* a*:* $a = 1$ *(left) and* $a = 10^6$ *(right).*

Acknowledgment. The time invested by Igor Kaporin (Computing Center of Russian Academy of Sciences) during a number of discussions about the developed algorithms is very much appreciated.

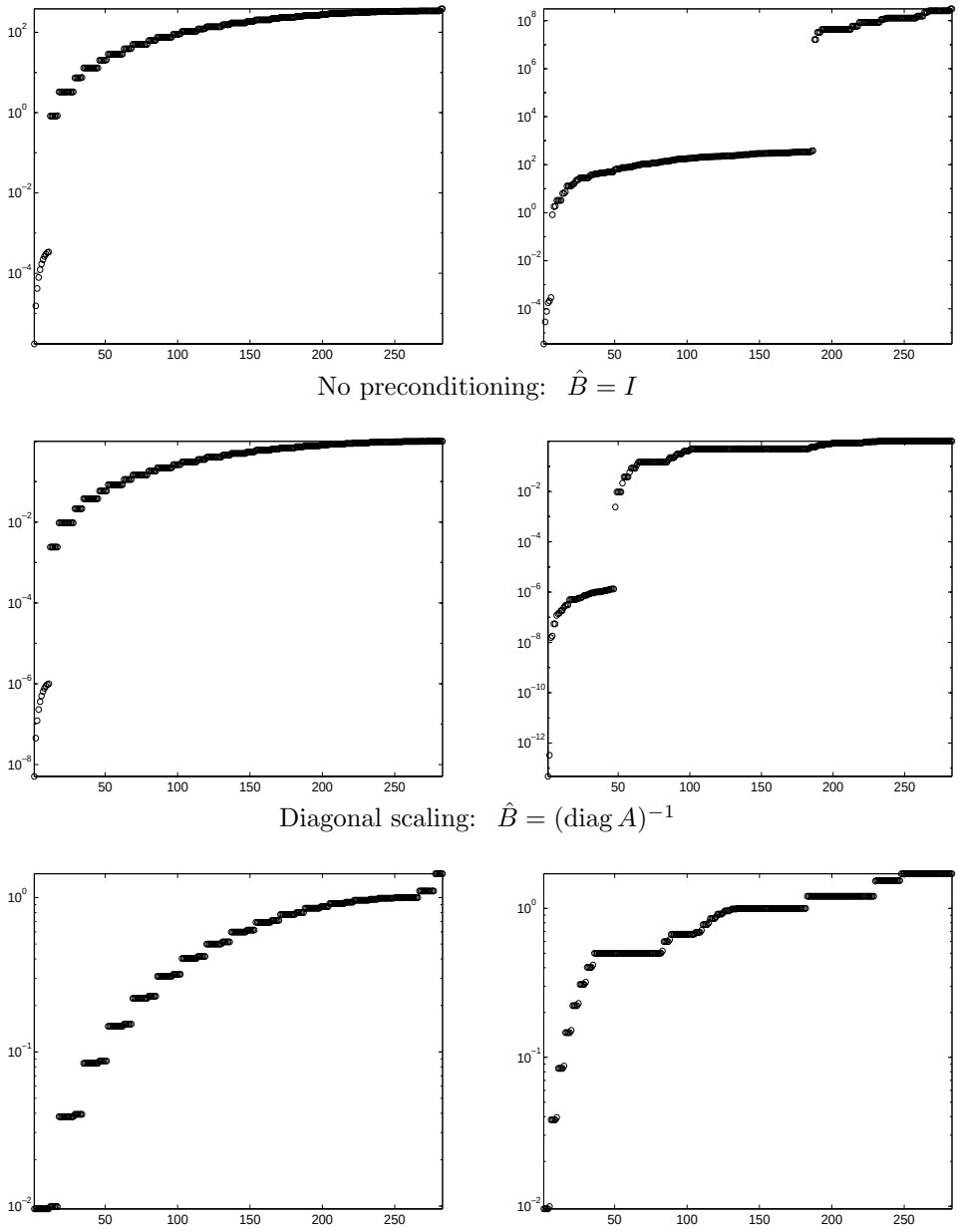


FIG. 3. *Problem* **D**, $h = \frac{1}{16}$, $\varepsilon = 10^6$, $\bar{y} = \frac{3}{8}$, *eigenvalue distribution of* $\hat{B}A$ *for different preconditioners* $\hat{B}$ *and different values of the coefficient jump* a*:* $a = 1$ *(left) and* $a = 10^6$ *(right).*

REFERENCES

[1] O. AXELSSON, *Stabilization of algebraic multilevel iteration methods; additive methods*, Numer. Algorithms, 21 (1999), pp. 23–47.

[2] O. AXELSSON, M. NEYTCHEVA, AND B. POLMAN, *An application of the bordering method to solve nearly singular systems*, Vestnik Moskov. Univ. Ser. XV, Vychisl. Mat. Kibernet. 1, Moscow, 1996, pp. 3–25.

[3] O. Axelsson and A. Padiy, *On the additive version of the algebraic multilevel iteration method for anisotropic elliptic problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1807–1830.
[4] J. Bramble, J. Pasciak, and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., 55 (1990), pp. 1–22.
[5] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
[6] J. Dendy, *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.
[7] T. Grauschopt, M. Griebel, and H. Regler, *Additive Multilevel Preconditioners Based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic Multigrid Coarsening for Second Order Elliptic PDEs*, internal report 342/02/96, University of München, Germany, 1996.
[8] W. Hackbush, *Multigrid Methods and Applications*, Springer-Verlag, Berlin, Heidelberg, New York, 1985.
[9] I. Kaporin, *Two-level explicit preconditioning for the conjugate gradient method*, Differential Equations, 28 (1992), pp. 280–289.
[10] L. Mansfield, *On the conjugate gradient solution of the Schur complement system obtained from domain decomposition*, SIAM J. Numer. Anal., 27 (1990), pp. 1612–1620.
[11] L. Mansfield, *Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1314–1323.
[12] R. A. Nicolaides, *Deflation of conjugate gradients with application to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.
[13] Y. Notay and A. van de Velde, *Coarse grid acceleration of parallel incomplete preconditioners*, in Iterative Methods in Linear Algebra II, S. Margenov and P. Vassilevski, eds., IMACS Series in Computational and Applied Mathematics 3, IMACS, New Brunswick, NJ, 1996, pp. 106–130.
[14] A. Padiy and M. Larin, *Model Analysis of a Subspace Correction Technique for Anisotropic Diffusion Problems*, internal report 9818, University of Nijmegen, Nijmegen, The Netherlands, 1998.
[15] J. Ruge and K. Stüben, *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, in Proceedings of the MG Conference, Bristol, England, 1983, Inst. Math. Appl. Conf. Ser. New Ser., 3 (1985), pp. 169–212.
[16] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Lecture Notes in Math. 960, Springer-Verlag, New York, 1982, pp. 1–176.
[17] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.
[18] J. Xu, *The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids*, Computing, 56 (1996), pp. 215–235.
[19] H. Yserentant, *On the multilevel splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.
[20] P. de Zeeuw, *Matrix-dependent prolongations and restrictions in a black-box multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.
[21] X. Zhang, *Multilevel Schwarz methods*, Numer. Math., 63 (1992), pp. 521–539.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 819–823

SINGULAR VALUES OF DIFFERENCES OF POSITIVE SEMIDEFINITE MATRICES*

XINGZHI ZHAN†

Abstract. Let M_n be the space of $n \times n$ complex matrices. For $A \in M_n$, let $s(A) \equiv (s_1(A), \ldots, s_n(A))$, where $s_1(A) \geq \cdots \geq s_n(A)$ are the singular values of A. We prove that if $A, B \in M_n$ are positive semidefinite, then (i) $s_j(A - B) \leq s_j(A \oplus B)$, $j = 1, 2, \ldots, n$, and (ii) the weak log-majorization relations $s(A-|z|B) \prec_{wlog} s(A+zB) \prec_{wlog} s(A+|z|B)$ hold for any complex number z. This sharpens some results due to R. Bhatia and F. Kittaneh.

Key words. singular values, positive semidefinite matrices, majorization, unitarily invariant norms

AMS subject classifications. 15A18, 15A42, 47A63

PII. S0895479800369840

1. Introduction.

Let M_n be the space of $n \times n$ complex matrices. For simplicity we treat matrices here, but all our results hold for compact operators on a Hilbert space. Suppose $A, B \in M_n$ are positive semidefinite. We shall study the relations between the singular values of

$$A - B \quad \text{and} \quad \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

and those of

$$A - |z|B, \quad A + zB, \quad \text{and} \quad A + |z|B,$$

where z is a complex number.

A norm $|||\cdot|||$ on M_n is called *unitarily invariant* if $|||UAV||| = |||A|||$ for all A and all unitary U, V. Every unitarily invariant norm is a symmetric gauge function of the singular values. See [3, 8]. We always denote the singular values of A by $s_1(A) \geq \cdots \geq s_n(A)$, and put $s(A) \equiv (s_1(A), \ldots, s_n(A))$. Familiar examples of unitarily invariant norms are the Ky Fan k-norms defined by $||A||_{(k)} = \sum_1^k s_j(A)$ and the Schatten p-norms: $||A||_p = (\sum_1^n s_j^p(A))^{1/p}$, $p \geq 1$. Note that $||\cdot||_\infty$ is just the operator (spectral) norm and $||\cdot||_2$ is the Frobenius norm.

A unitarily invariant norm may be considered as defined on M_n for all orders n by the rule

$$|||A||| = \left|\left|\left| \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \right|\right|\right|,$$

i.e., adding or deleting zero singular values does not affect the value of the corresponding symmetric gauge function.

*Received by the editors March 31, 2000; accepted for publication (in revised form) by R. Bhatia June 26, 2000; published electronically October 31, 2000. This work was done while the author was at the Graduate School of Information Sciences, Tohoku University, as a postdoctoral fellow of the Japan Society for the Promotion of Science.

http://www.siam.org/journals/simax/22-3/36984.html

†Institute of Mathematics, Peking University, Beijing 100871, People's Republic of China. Current address: Graduate School of Information Sciences, Tohoku University, Aoba-ku, Sendai 980-8579, Japan (zhan@math.is.tohoku.ac.jp).

Given a real vector $x = (x_i) \in \mathbb{R}^n$, rearrange its components as $x_{[1]} \geq \cdots \geq x_{[n]}$. For $x = (x_i)$, $y = (y_i) \in \mathbb{R}^n$, if

$$\sum_1^k x_{[i]} \leq \sum_1^k y_{[i]}, \qquad k = 1, 2, \ldots, n,$$

then we say x is *weakly majorized* by y, denoted $x \prec_w y$. If the components of x and y are nonnegative and

$$\prod_1^k x_{[i]} \leq \prod_1^k y_{[i]}, \qquad k = 1, 2, \ldots, n,$$

we say x is *weakly log-majorized* by y, denoted $x \prec_{wlog} y$. See [7] for a discussion of this topic.

Denote the block diagonal matrix $\left(\begin{smallmatrix} A & 0 \\ 0 & B \end{smallmatrix}\right)$ by $A \oplus B$.

Bhatia and Kittaneh [4, Remark 5] observed that if $A, B \in M_n$ are positive semidefinite, then

$$|||A - B||| \leq |||A \oplus B||| \tag{1.1}$$

for every unitarily invariant norm. By the Fan dominance principle [3, 8], (1.1) is equivalent to $s(A - B) \prec_w s(A \oplus B)$. We shall show that in fact each singular value of $A - B$ is not greater than the corresponding singular value of $A \oplus B$.

In another paper, Bhatia and Kittaneh [5, Theorem 2.1] proved that for positive semidefinite $A, B \in M_n$ and any complex number z

$$|||A - |z|B||| \leq |||A + zB||| \leq |||A + |z|B||| \tag{1.2}$$

for all unitarily invariant norms. Again (1.2) is equivalent to

$$s(A - |z|B) \prec_w s(A + zB) \prec_w s(A + |z|B).$$

We shall prove that the corresponding weak log-majorizations hold. Since weak log-majorization implies weak majorization [7, 8], our result strengthens (1.2).

2. Main results. Our first result sharpens (1.1).

THEOREM 2.1. *Let $A, B \in M_n$ be positive semidefinite. Then*

$$s_j(A - B) \leq s_j(A \oplus B), \qquad j = 1, 2, \ldots, n. \tag{2.1}$$

We shall use the fact [6, p. 29] that for $G \in M_n$ and $1 \leq j \leq n$

$$s_j(G) = \min\{||G - E|| : \mathrm{rank} E \leq j - 1,\ E \in M_n\}, \tag{2.2}$$

where $||\cdot||$ is the operator norm. We use the notation $H \leq K$ to mean that H, K are Hermitian and $K - H$ is positive semidefinite.

Proof of Theorem 2.1. Note that $s(A \oplus B) = s(A) \cup s(B)$. It is easily verified (say, by using the spectral decompositions of A, B) that for a fixed j with $1 \leq j \leq n$ there exist $H, F \in M_n$ satisfying $0 \leq H \leq A$, $0 \leq F \leq B$, $\mathrm{rank} H + \mathrm{rank} F \leq j - 1$ and

$$s_j(A \oplus B) = ||(A - H) \oplus (B - F)||.$$

Thus $s_j(A \oplus B) = \max\{||A - H||, ||B - F||\} \equiv \gamma$. Denote by I the identity matrix. Note that $A - H \geq 0$, $B - F \geq 0$, $\text{rank}(H - F) \leq \text{rank}H + \text{rank}F \leq j - 1$. By (2.2) we have

$$\begin{aligned} s_j(A - B) &\leq ||A - B - (H - F)|| \\ &= \left|\left|\left(A - H - \frac{\gamma}{2}I\right) - \left(B - F - \frac{\gamma}{2}I\right)\right|\right| \\ &\leq \left|\left|(A - H) - \frac{\gamma}{2}I\right|\right| + \left|\left|(B - F) - \frac{\gamma}{2}I\right|\right| \\ &\leq \frac{\gamma}{2} + \frac{\gamma}{2} = \gamma = s_j(A \oplus B). \end{aligned}$$

This proves (2.1). □

We can give a second proof of Theorem 2.1 by using the following result due to Bhatia and Kittaneh [4]: For any $X, Y \in M_n$

$$s_j(XY^*) \leq \frac{1}{2}s_j(X^*X + Y^*Y), \qquad j = 1, \ldots, n. \tag{2.3}$$

Just set

$$X = \begin{pmatrix} A^{1/2} & -B^{1/2} \\ 0 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} A^{1/2} & B^{1/2} \\ 0 & 0 \end{pmatrix}$$

in (2.3).

T. Ando has pointed out that (2.1) is equivalent to the statement that there exists an $n \times 2n$ contraction W (i.e., $WW^* \leq I$) such that $|A - B| = W(A \oplus B)W^*$. Such a W can be obtained explicitly by the Jordan decomposition of the Hermitian matrix $A - B$. This gives a third proof of Theorem 2.1.

The following result sharpens (1.2).

THEOREM 2.2. *Let $A, B \in M_n$ be positive semidefinite. Then for any complex number z*

$$s(A - |z|B) \prec_{wlog} s(A + zB) \prec_{wlog} s(A + |z|B). \tag{2.4}$$

Proof. We first prove the following determinant inequality for positive semidefinite matrices P, Q of the same order:

$$|\det(P - |z|Q)| \leq |\det(P + zQ)|. \tag{2.5}$$

Without loss of generality, suppose P is positive definite. The general case follows by a standard continuity argument. Let the eigenvalues of $P^{-1}Q$ be $\lambda_1 \geq \cdots \geq \lambda_k \geq 0$. Then

$$\begin{aligned} |\det(P - |z|Q)| &= |\det P \cdot \det(I - |z|P^{-1}Q)| = \det P \prod_i |1 - |z|\lambda_i| \\ &\leq \det P \prod_i |1 + z\lambda_i| = |\det P \cdot \det(I + zP^{-1}Q)| \\ &= |\det(P + zQ)|. \end{aligned}$$

This shows (2.5). Since $A - |z|B$ is Hermitian, for $1 \leq k \leq n$ there exists an $n \times k$ matrix U such that $U^*U = I$ and $\prod_{j=1}^k s_j(A - |z|B) = |\det[U^*(A - |z|B)U]|$. Using

(2.5) and the fact that for any $G \in M_n$, $s_j(U^*GU) \le s_j(G),\ j = 1, \ldots, k$, we have

$$\begin{aligned}\prod_{j=1}^{k} s_j(A - |z|B) &= |\det[U^*(A - |z|B)U]| = |\det(U^*AU - |z|U^*BU)| \\ &\le |\det(U^*AU + zU^*BU)| = \prod_{j=1}^{k} s_j[U^*(A + zB)U] \\ &\le \prod_{j=1}^{k} s_j(A + zB).\end{aligned}$$

In the third equality above we have used the fact that for any $F \in M_k$, $|\det F| = \prod_{j=1}^{k} s_j(F)$. This proves the first part of (2.4).

Recall [3, p. 268] that a continuous complex-valued function f on M_n is said to be a *Lieb function* if it satisfies the following two conditions:

(i) $f(B) \ge f(A) \ge 0$ if $B \ge A \ge 0$.

(ii) $|f(A^*B)|^2 \le f(A^*A)f(B^*B)$ for all A, B.

It is known [9, Theorem 6] that if $N, R \in M_n$ are normal, then for any Lieb function f on M_n

$$|f(N + R)| \le f(|N| + |R|). \tag{2.6}$$

It is easily verified (see [3, p. 269]) that $f(G) \equiv \prod_{j=1}^{k} s_j(G)$ is a Lieb function. Applying (2.6) to this f with $N = A$, $R = zB$ yields

$$s(A + zB) \prec_{wlog} s(A + |z|B).$$

This completes the proof. □

The special case $z = i = \sqrt{-1}$ of Theorem 2.2 says

$$s(A - B) \prec_{wlog} s(A + iB) \prec_{wlog} s(A + B). \tag{2.7}$$

It has been proved in [2] that for positive A, B and $p > 1$

$$s(A^p + B^p) \prec_w s((A + B)^p). \tag{2.8}$$

When $p \ge 2$, the above relation is refined as follows:

$$s(A^p + B^p) \prec_w s((A^2 + B^2)^{p/2}) \prec_w s(|A + iB|^p) \prec_{wlog} s((A + B)^p). \tag{2.9}$$

The first relation in (2.9) follows from (2.8) and the third relation follows from (2.7). To see the second relation let $T = A + iB$. This is the *Cartesian decomposition.* From $A^2 + B^2 = (T^*T + TT^*)/2$ we get

$$s(A^2 + B^2) \prec_w s(|A + iB|^2).$$

Note that $f(t) = t^{p/2}$ is convex and increasing on $[0, \infty)$. By a majorization principle [3, 8], applying this f to the preceding weak majorization yields the second relation in (2.9).

From (2.7) and the results in [1] and [2] it follows that for $0 < p \le 1$,

$$\begin{aligned}s(A^p - B^p) \prec_w s(|A - B|^p) \prec_{wlog} s(|A + iB|^p) \prec_{wlog} s((A + B)^p) \\ \prec_w \quad s(A^p + B^p).\end{aligned}$$

One might wonder whether the weak majorization (2.8) can be replaced by the stronger log-majorization. The answer is no, even for $p = 2$. Consider the example

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

We have $\det(A^2 + B^2) = 2 > 1 = \det[(A + B)^2]$.

Acknowledgments. The author thanks JSPS for the support and Professor F. Hiai for helpful discussions.

REFERENCES

[1] T. Ando, *Comparison of norms* $|||f(A) - f(B)|||$ *and* $|||f(|A - B|)|||$, Math. Z., 197 (1988), pp. 403–409.
[2] T. Ando and X. Zhan, *Norm inequalities related to operator monotone functions*, Math. Ann., 315 (1999), pp. 771–780.
[3] R. Bhatia, *Matrix Analysis*, Springer, Berlin, 1997.
[4] R. Bhatia and F. Kittaneh, *On the singular values of a product of operators*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 272–277.
[5] R. Bhatia and F. Kittaneh, *Norm inequalities for positive operators*, Lett. Math. Phys., 43 (1998), pp. 225–231.
[6] I. C. Gohberg and M. G. Krein, *Introduction to the Theory of Linear Nonselfadjoint Operators*, AMS, Providence, RI, 1969.
[7] F. Hiai, *Log-majorizations and norm inequalities for exponential operators*, in Linear Operators, Banach Center Publ. 38, 1997, pp. 119–181.
[8] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
[9] R. A. Horn and X. Zhan, *Inequalities for C-S seminorms and Lieb functions*, Linear Algebra Appl., 291 (1999), pp. 103–113.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 824–836

A REFINED POLAR DECOMPOSITION: $A = UPD^*$

TIMO EIROLA†

Abstract. A refinement of the polar decomposition of a nonsingular matrix A is considered. Here A is written as a product of unitary U, Hermitian and positive definite P which has unit diagonal, and diagonal positive D. It is shown that such a decomposition exists and is unique. Rectangular and singular cases are also considered. Then a simple fixed point iteration using SVD is given to compute this decomposition. Also, implementation of the Newton's method is discussed. The refined polar decomposition can be used to parameterize the orbit of a matrix with distinct eigenvalues.

Key words. polar decomposition, orbits of matrices

AMS subject classifications. 15A23, 65F35

PII. S0895479800369219

1. Introduction.

The polar decomposition of a matrix $A \in \mathbb{C}^{m\times n}$, $m \geq n$, is given as

$$A = QS,$$

where $Q \in \mathbb{C}^{m\times n}$ has orthonormal columns and $S \in \mathbb{C}^{n\times n}$ is Hermitian and positive semidefinite. S is unique and, in the case A has rank n, Q is also unique (see, e.g., [3], [5]). They are given by

$$S = (A^*A)^{1/2}, \qquad Q = A(A^*A)^{-1/2}, \tag{1.1}$$

where $C^{1/2}$ denotes the symmetric positive semidefinite square root of the symmetric positive semidefinite matrix C.

Here we want to further decompose $S = VPD$ so that this decomposition—with unitary V and nonnegative diagonal D—would have P Hermitian positive semidefinite and the diagonal elements of P would be ones. This leads to the decomposition

$$A = UPD, \tag{1.2}$$

where $U \in \mathbb{C}^{m\times n}$ has orthonormal columns, $P \in \mathbb{C}^{n\times n}$ is Hermitian and positive semidefinite with unit diagonal, and $D \in \mathbb{R}^{n\times n}$ is diagonal and nonnegative. The requirements on P mean that it is a *correlation matrix*. For the importance of such matrices, see [1] and references therein.

There is a result similar to (1.2). Olkin and Pratt show in [6] that every real symmetric positive definite matrix S can be uniquely decomposed as $S = \tilde{P}\tilde{D}^2\tilde{P}$, where $\tilde{P}$ is a correlation matrix and $\tilde{D}$ is positive diagonal. For full column rank A this leads to the unique decomposition $A = \tilde{U}\tilde{D}\tilde{P}$, i.e., the order of P and D parts of (1.2) reversed. A way to convert this result to (1.2) (or vice versa) was not found.

In section 2 it is shown that the decomposition (1.2) always exists. D is unique and if A does not have columns that are zero, then P is unique, too, and so is U if

*Received by the editors March 16, 2000; accepted for publication (in revised form) by N. Higham September 26, 2000; published electronically November 17, 2000.
http://www.siam.org/journals/simax/22-3/36921.html
†Institute of Mathematics, Helsinki University of Technology, FIN-02015 HUT, Espoo, Finland (Timo.Eirola@hut.fi).

A has full rank. As the usual polar factors these factors are also smooth functions of A in the set of full rank matrices.

Then, in section 3 some methods are considered for computing (1.2). First, a simple fixed point iteration of the D part is given and is shown to be locally convergent. Then, two variants of the Newton iteration are discussed.

Finally, an application is considered. This is to parameterize the orbit (set of similar matrices) of a matrix with distinct eigenvalues.

2. Existence and uniqueness. In this section the main theorem concerning (1.2) is given. For this, first note that the polar decomposition is real analytic as a function of full rank A. This is seen from (1.1) and

$$S = \tfrac{1}{2\pi i} \int_\gamma \sqrt{z}\, (zI - A^*A)^{-1}\, dz, \tag{2.1}$$

where γ is a positively oriented simple rectifiable closed curve in the right-hand side of $\mathbb{C}$ strictly enclosing the spectrum of A^*A, and $\sqrt{z}$ denotes the root with positive real part ([5], Thm. 6.2.28; see also [2]). Using the SVD it is then easy to show that formula (2.1) holds even in the case of singular A, now γ enclosing only the nonzero eigenvalues. It follows that S is smooth in each of the sets of constant rank matrices.

Notation. $|v|$ denotes the 2-norm of $v \in \mathbb{C}^n$, and $\| \ \|$ is the corresponding matrix norm. $\mathcal{D} \subset \mathbb{C}^{n\times n}$ is the set of diagonal matrices, $\mathcal{D}_+ \subset \mathcal{D}_{\mathbb{R}} \subset \mathcal{D}$ contain the ones with, respectively, nonnegative and real entries, and $\mathrm{Diag}(M) \in \mathcal{D}$ is the diagonal of $M \in \mathbb{C}^{n\times n}$ and $\mathrm{diag}(M) \in \mathbb{C}^n$ is the corresponding vector. For $x \in \mathbb{R}^n$ denote

$$\mathbf{D}(x) = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix} \in \mathcal{D}_{\mathbb{R}}$$

and let $\mathbf{X} : \mathcal{D}_{\mathbb{R}} \to \mathbb{R}^n$ be its inverse mapping. $A \circ B$ will denote the Hadamard (elementwise) product.

THEOREM 2.1 (refined polar decomposition). *Assume $A \in \mathbb{C}^{m\times n}$, $m \geq n$. Then there exists a decomposition*

$$A = UPD, \tag{2.2}$$

where $U^*U = I$, P is Hermitian, positive semidefinite, $\mathrm{Diag}(P) = I$, *and $D \in \mathcal{D}_+$. D is unique. If A has no zero columns, then P is unique. If* $\mathrm{rank}(A) = n$, *then U is also unique and P, D are positive definite.*

Proof. First assume that A does not have a zero column.

Existence. For given positive diagonal D let $U_D P_D = AD$ be a polar decomposition of AD. Then, P_D is unique. Set $F(D) = \mathrm{Diag}(P_D)$. By (2.1) F is continuous. Then, the existence of (2.2) amounts to solving $F(D^{-1}) = I$.

Fix a

$$D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix} \in \mathcal{D}_+.$$

Let $a_1, \ldots, a_n$ and $p_1, \ldots, p_n$, respectively, be the columns of A and P_D. These are nonzero. Set

$$m = \min_j |a_j|, \quad M = \max_j |a_j|.$$

Then, $p_{ii} \le |p_i| = |a_i| d_i \le M d_i$. Since P_D is Hermitian and positive semidefinite, for all i, j holds $p_{ii} p_{jj} \ge |p_{ij}|^2$. Thus, $M d_i \, p_{jj} \ge |p_{ij}|^2$ and

$$M \left(\sum_i d_i \right) p_{jj} \ge \sum_i |p_{ij}|^2 = |a_j|^2 d_j^2 \ge m^2 d_j^2.$$

Hence,

$$\frac{m^2}{M} \, \frac{d_j^2}{\sum_i d_i} \le F_{jj}(D) \le M d_j. \tag{2.3}$$

Fix $\alpha \in (0, 1/2]$. For $\delta \in \mathbb{R}^n$ set

$$g(\delta) = \delta + \alpha \mathbf{X}(\log(F(e^{-\mathbf{D}(\delta)}))) \in \mathbb{R}^n. \tag{2.4}$$

Here log is the principal logarithm (applied to positive diagonal matrices).

Assume $\delta_j \in [\lambda, \Lambda]$ for all j. Then, inequalities (2.3) imply

$$g_j(\delta) \le \delta_j + \alpha \log(M e^{-\delta_j}) = (1 - \alpha)\delta_j + \alpha \log(M) \le (1 - \alpha)\Lambda + \alpha \log(M)$$

and

$$\begin{aligned} g_j(\delta) &\ge \delta_j + \alpha \log(m^2 e^{-2\delta_j}) - \alpha \log\left(M \, \textstyle\sum_i e^{-\delta_i}\right) \\ &\ge (1 - 2\alpha)\delta_j + \alpha \log(m^2) - \alpha \log(Mn) + \alpha\lambda \\ &\ge (1 - \alpha)\lambda + \alpha \log(\tfrac{m^2}{Mn}). \end{aligned}$$

Choosing

$$\lambda = \log(\tfrac{m^2}{Mn}) \qquad \text{and} \qquad \Lambda = \log(M)$$

we get $g_j(\delta) \in [\lambda, \Lambda]$ for all j. Hence g maps the convex cube $[\lambda, \Lambda]^n \subset \mathbb{R}^n$ into itself. Further, g is continuous. Hence, by Brower's fixed point theorem (see, e.g., [7]), there exists $\delta \in \mathcal{D}_{\mathbb{R}}$ such that $g(\delta) = \delta$ and $D = e^{\mathbf{D}(\delta)}$ solves $F(D^{-1}) = I$.

Uniqueness. Define $f : \mathbb{R}^n \to \mathbb{R}^n$ as

$$f(d) = \mathbf{X}(F(e^{\mathbf{D}(d)})). \tag{2.5}$$

By Lemma 2.2 below, the derivative $f'(d) \in \mathbb{R}^{n \times n}$ is positive definite for all d. Hence, if $f(\hat{d}\,) = f(d)$, then

$$0 = (\hat{d} - d)^T (f(\hat{d}\,) - f(d)) = (\hat{d} - d)^T \int_0^1 f'(d + t(\hat{d} - d)) \, dt \, (\hat{d} - d)$$

so that $\hat{d} - d = 0$, since the integral is positive definite. Thus, f, and consequently also F, is an injection. This shows uniqueness of D. Then, uniqueness of P and U follow from the corresponding properties of the polar decomposition.

If A has $q \ge 1$ columns that are zero, then, necessarily, we set the corresponding elements of D to zero. Let $\hat{A} \in \mathbb{C}^{m \times (n-q)}$ consist of the nonzero columns of A. Take a refined decomposition $\hat{A} = \hat{U}\hat{P}\hat{D}$ and put elements of these in the corresponding places of U, P, and D. Fill the rest of P (except diagonal elements) with zeros, and U with orthonormal columns, orthogonal to $\hat{U}$. ∎

The following lemma was needed above and will be used again when considering the computation of the refined polar decomposition with Newton's method.

LEMMA 2.2. *Assume A has no zero columns. Let $UP = Ae^{\mathbf{D}(d)}$ be a polar decomposition, and let $P = V\mathbf{D}(\pi)V^*$ be an eigendecomposition of P. Then, the derivative of f of (2.5) is given by*

$$f'(d)\delta = \operatorname{diag}(V(\Pi \circ (V^*\mathbf{D}(\delta)V))V^*), \tag{2.6}$$

where $\Pi_{ij} = \frac{\pi_i^2+\pi_j^2}{\pi_i+\pi_j}$, $\Pi_{ij} = 0$ if $\pi_i + \pi_j = 0$. Further, $f'(d)$ is positive definite.

Proof. We have $P = (e^{\mathbf{D}(d)}A^*Ae^{\mathbf{D}(d)})^{\frac{1}{2}}$, and this is differentiable with respect to d. This is true even in the case of singular A since $e^{\mathbf{D}(d)}A^*Ae^{\mathbf{D}(d)}$ has constant rank and formula (2.1) can be applied.

For small $\delta \in \mathbb{R}^n$ we need a Hermitian Δ such that

$$(P+\Delta)^2 = e^{\mathbf{D}(d+\delta)}A^*Ae^{\mathbf{D}(d+\delta)} + O(\delta^2 + \Delta^2),$$

i.e.,

$$\begin{aligned} P\Delta + \Delta P &= \mathbf{D}(\delta)e^{\mathbf{D}(d)}A^*Ae^{\mathbf{D}(d)} + e^{\mathbf{D}(d)}A^*Ae^{\mathbf{D}(d)}\mathbf{D}(\delta) \\ &= \mathbf{D}(\delta)P^2 + P^2\mathbf{D}(\delta). \end{aligned}$$

Using $P = V\mathbf{D}(\pi)V^*$ we get

$$\mathbf{D}(\pi)\ V^*\Delta V + V^*\Delta V\ \mathbf{D}(\pi) = V^*\mathbf{D}(\delta)V\ \mathbf{D}(\pi)^2 + \mathbf{D}(\pi)^2\ V^*\mathbf{D}(\delta)V,$$

i.e.,

$$(\pi_i + \pi_j)(V^*\Delta V)_{ij} = (\pi_i^2 + \pi_j^2)(V^*\mathbf{D}(\delta)V)_{ij}.$$

If $\pi_i + \pi_j = 0$, we take $(V^*\Delta V)_{ij} = 0$. Hence $V^*\Delta V = \Pi \circ (V^*\mathbf{D}(\delta)V)$ and (2.6) follows.

Let $u = (1, \ldots, 1)$. Positive definiteness is shown by

$$\begin{aligned} \delta^T f'(d)\delta &= \operatorname{tr}(\mathbf{D}(\delta)V(\Pi \circ (V^*\mathbf{D}(\delta)V))V^*) \\ &= \operatorname{tr}(V^*\mathbf{D}(\delta)V(\Pi \circ (V^*\mathbf{D}(\delta)V))) \\ &= \textstyle\sum_{i,j} \frac{\pi_i^2+\pi_j^2}{\pi_i+\pi_j}\ |(V^*\mathbf{D}(\delta)V)_{ij}|^2 \\ &\geq \textstyle\sum_{i,j} \frac{\pi_i+\pi_j}{2}\ |(V^*\mathbf{D}(\delta)V)_{ij}|^2 \\ &= \tfrac{1}{2}\operatorname{tr}(V^*\mathbf{D}(\delta)V((\pi u^T + u\pi^T) \circ (V^*\mathbf{D}(\delta)V))) \\ &= \tfrac{1}{2}\operatorname{tr}(V^*\mathbf{D}(\delta)V\mathbf{D}(\pi)V^*\mathbf{D}(\delta)V) + \tfrac{1}{2}\operatorname{tr}(V^*\mathbf{D}(\delta)VV^*\mathbf{D}(\delta)V\mathbf{D}(\pi)) \\ &= \operatorname{tr}(\mathbf{D}(\delta)P\mathbf{D}(\delta)) = \textstyle\sum_j p_{jj}\delta_j^2. \end{aligned}$$

∎

REMARK 1. *In the proof of the theorem the g-function was defined by (2.4) for $\alpha \in (0, 1/2]$. Values $\alpha \in (1/2, 1)$ also work. Then, the lower bound becomes*

$$g_j(\delta) \geq (1-2\alpha)\Lambda + \alpha\lambda + \alpha\log(\tfrac{m^2}{Mn})$$

and the choice

$$\lambda = \tfrac{1}{1-\alpha}\ ((1-2\alpha)\Lambda + \alpha\log(\tfrac{m^2}{Mn}))$$

works. In the numerical computations we will mostly use $\alpha \approx 2/3$.

The following is kind of a dual result for the $S = \tilde{P}\tilde{D}^2\tilde{P}$-decomposition of Olkin and Pratt [6] mentioned in the introduction.

COROLLARY 2.3. *Every Hermitian positive definite matrix* S *can be uniquely decomposed as* $S = DP^2D$, *where* P *is a correlation matrix and* D *is positive diagonal.*

Proof. This is immediate after decomposing $S^{\frac{1}{2}} = UPD$. ∎

REMARK 2. *The* Π*-matrix above is quite interesting: it is positive and has only one positive eigenvalue. It is negative semidefinite in the subspace orthogonal to* $u = (1, \ldots, 1)$ *(i.e.,* $-\Pi$ *is conditionally positive definite; see* [5]*). This is seen as follows:*[1] *write*

$$\frac{\pi_i^2 + \pi_j^2}{\pi_i + \pi_j} = \pi_i + \pi_j - \frac{2\pi_i\pi_j}{\pi_i + \pi_j}.$$

Thus, the matrix $u\pi^T + \pi u^T - \Pi$ *has entries* $\frac{2\pi_i\pi_j}{\pi_i+\pi_j}$, *i.e., it is a diagonal scaling of a Cauchy matrix (see* [5, p. 348]*) and thus is positive semidefinite. This implies that* $f'(d)$ *is an* M*-matrix (see Remark* 5 *below).*

REMARK 3. *From the proof of Lemma* 2.2 *we see that* $\|f'(d)^{-1}\| \geq \frac{1}{\min_j P_{jj}}$. *Further, since* $\frac{\pi_i^2+\pi_j^2}{\pi_i+\pi_j} \leq \max(\pi_i, \pi_j)$ *we get* $\|f'(d)\| \leq \max_j \pi_j = \|P\|$. *In practice, we usually observe* $\|f'(d)\| \approx 2$.

Finally, for good matrices the decomposition is smooth (real analytic).

PROPOSITION 2.4. *In the set of full rank matrices* $A \in \mathbb{C}^{m\times n}$, $m \geq n$, *the factors* U, P, *and* D *are real analytic functions of the (real and imaginary parts of the) elements of* A.

Proof. By the implicit function theorem the matrix D that solves $F(D^{-1}) = I$ depends smoothly on A (F' is invertible). Then, $P = (D^{-1}A^*AD^{-1})^{\frac{1}{2}}$ and $U = AP^{-1}D^{-1}$ are also smooth. ∎

REMARK 4. *Consider diagonal scaling of a matrix to reduce its condition number. A result of van der Sluis* [8] *says that if* P *is Hermitian positive definite and has constant diagonal, then the* 2*-norm condition number satisfies*

$$\kappa(P) \leq n \inf_{E\in\mathcal{D}_+} \kappa(EPE).$$

Using this one gets from the refined polar decomposition $A = UPD$ *the following:*[2]

$$\begin{aligned}\kappa(AD^{-1}) = \kappa(UP) = \kappa(P) &\leq n \inf_{E\in\mathcal{D}_+} \kappa(EPE)\\ &= n \inf_{E\in\mathcal{D}_+} \frac{\lambda_{\max}(EPE)}{\lambda_{\min}(EPE)} = n \inf_{E\in\mathcal{D}_+} \frac{\lambda_{\max}(PE^2)}{\lambda_{\min}(PE^2)}\\ &\leq n \inf_{E\in\mathcal{D}_+} \frac{\sigma_{\max}(PE^2)}{\sigma_{\min}(PE^2)} = n \inf_{E\in\mathcal{D}_+} \kappa(PE) = n \inf_{E\in\mathcal{D}_+} \kappa(AE),\end{aligned}$$

i.e., a suboptimal column scaling. This is only a curiosity since a better result is obtained,[3] *just by scaling the columns of* $A\tilde{D}$ *to have equal norms, since then* $\tilde{D}A^*A\tilde{D}$ *has constant diagonal and the result of van der Sluis gives*

$$\kappa(A\tilde{D}) \leq \sqrt{n} \inf_{E\in\mathcal{D}_+} \kappa(AE).$$

[1] Thanks to the unknown referee.

[2] Here $\lambda_{\min\,/\,\max}$, $\sigma_{\min\,/\,\max}$ denote the maximal and minimal eigenvalues and singular values.

[3] Thanks to the editor and a referee.

3. Numerical computation. Here we consider the two obvious approaches to compute the refined polar decomposition. First, we consider fixed point iterations of the g-function (2.4). Then, we will apply (2.6) in a Newton scheme and also consider more economic approximate Newton steps.

3.1. Fixed point iteration. A simple numerical method is obtained by iteration of the map g of (2.4) with $\alpha \in (0, 1)$. We use the (economy version) SVD to compute values of f. From the SVD

$$Q\,\mathbf{D}(\pi)\,V^* = A\,e^{-\mathbf{D}(d)}$$

we get $P = V\,\mathbf{D}(\pi)\,V^*$ already eigendecomposed (see Lemma 2.2) and

$$f(-d) = \operatorname{diag}(P) = (V \circ \bar{V})\pi.$$

For MATLAB we can write

```
function [U,P,D]=UPD_F(A)

% This function computes the refined polar decomposition
% of A using the fixed point iteration.

sz=size(A); n=sz(2); alpha=2/3;
expd=ones(n,1); d=zeros(n,1);
err=1; k=0; tol=10^(-13);

while err > tol,
   [Q,E,V]=svd(A*diag(expd),0);
   f=log((V.*conj(V))*diag(E));
   d=d+alpha*f; expd=exp(-d);
   err=norm(f); k=k+1; end

U=Q*V';  P=V*E*V';  D=diag(1./expd);
```

Note that elementwise `log` is taken here from a positive vector. This is equivalent to the principal logarithm of the corresponding diagonal matrix.

For small α local convergence of this iteration is guaranteed.

PROPOSITION 3.1. *Given A, without zero columns, the iteration $d^{k+1} = g(d^k)$ converges from d_0 close to $d = g(d)$ provided $\alpha \in (0, \frac{2}{\|P\|})$, where P is the Hermitian part of $A = UPD$.*

Proof. This follows directly from $g'(d) = I - \alpha f'(-d)$ and the positive definiteness of f'. Note that by Remark 3 $\|f'(-d)\| \le \|P\| \le n$. □

In Figure 3.1 α varies from 0.2 to 1 and the iteration counts for different matrices are plotted. The matrices are as follows:

50×50 random matrix: `randn(50,50)` (—),
20×20 Hilbert matrix: `hilb(20)` ($\cdots$),
50×50 random matrix of rank 25: `rand(50,25)*rand(25,50)` (- - -),
50×25 random matrix: `randn(50,25)` ($-\cdot-\cdot-$).

A rule of thumb is that for S closer to a diagonal matrix, α closer to one gives fastest convergence.

3.2. Newton iteration. We want to solve $f(d) = u$, where $u = (1, \ldots, 1)$. To obtain the polar decomposition for computing $f(d)$ we use again the singular value decomposition $Q\,\mathbf{D}(\pi)\,V^* = A\,e^{\mathbf{D}(d)}$ and $f(d) = (V \circ \bar{V})\pi$.

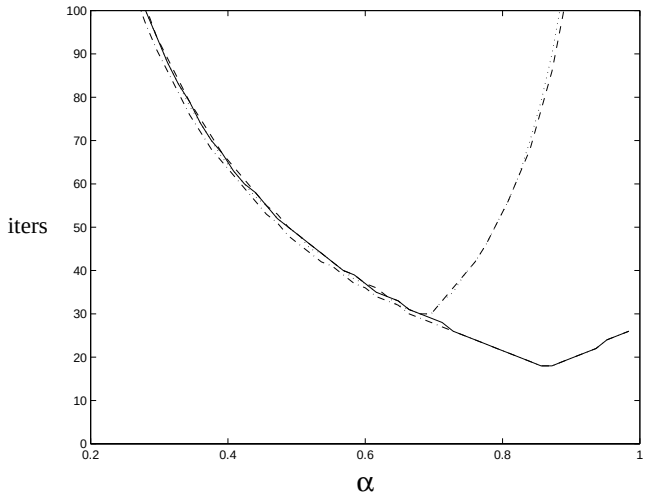


FIG. 3.1. *Iteration counts as functions of* α.

Using (2.6) we get the entries of f':

$$
\begin{aligned}
f'(d)_{ij} &= e_i^T\, f'(d)\, e_j \\
&= \operatorname{tr}\Big(\mathbf{D}(e_i)V\big(\Pi \circ (V^*\mathbf{D}(e_j)V)\big)V^*\Big) \\
&= \operatorname{tr}\Big(V^*\mathbf{D}(e_i)V\big(\Pi \circ (V^*\mathbf{D}(e_j)V)\big)\Big) \\
&= \operatorname{tr}\Big(v_i v_i^*\big(\Pi \circ (v_j v_j^*)\big)\Big) \\
&= (v_i \circ \bar{v}_j)^*\Pi\,(v_i \circ \bar{v}_j),
\end{aligned}
\tag{3.1}
$$

where v_i's are the columns of V^*.

REMARK 5. *By Remark* 2 Π *is conditionally negative semidefinite. Thus, for* $i \neq j$ *we get from* $(v_i \circ \bar{v}_j)^* u = v_i^* v_j = 0$ *that* $f'(d)_{ij} \leq 0$. *Hence,* $f'(d)$ *is an M-matrix.*

REMARK 6. By Remark 3

$$\operatorname{cond}(f'(d)) \leq \frac{\|P\|}{\min_j P_{jj}}.$$

At the solution this upper bound is $\leq n$. Hence, good conditioning of the Jacobian and local convergence is guaranteed.

Due to (3.1), the complexity (flop count) is $O(n^4)$ per iteration step. On the other hand, computing f' this way parallelizes easily.

In the experiments the initial guess $d = 0$ seems to work in most cases,[4] but for safety we take first one step of the fixed point iteration of g with $\alpha = 2/3$ to obtain $d_0 = -g(0)$.

With these remarks Newton's method for computing the refined polar decomposition can be written as follows:

[4] In the short series of test problems tried so far just a few cases required a better initial guess.

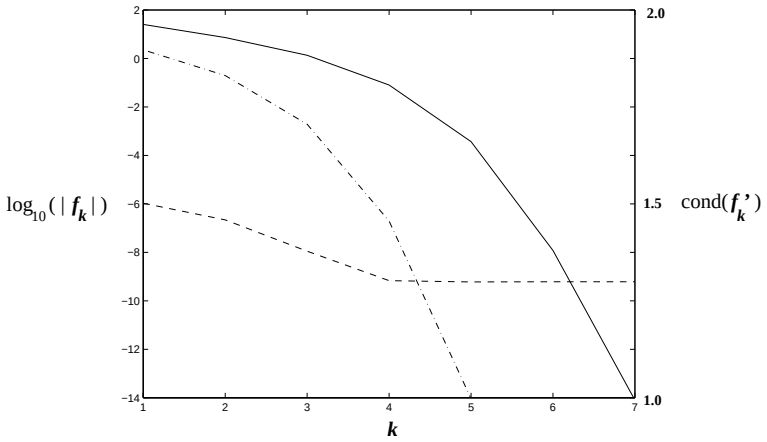


FIG. 3.2. *Error norm and condition number during Newton's method.*

```
function [U,P,D]=UPD_N(A)

% This function computes the refined polar decomposition
% of A using Newton's method.

sz=size(A); n=sz(2);
[U,E,V]=svd(A,0);
d=-2/3*log((V.*conj(V))*diag(E));
expd=exp(d); w=ones(n,1); df=zeros(n,n);
err=1; k=0; tol=10^(-13);
[U,E,V]=svd(A*diag(expd),0);
Vc=conj(V); p=diag(E);
f=(V.*Vc)*p-w;

while err > tol,
   Pi=(p.^2*w'+w*p'.^2)./(p*w'+w*p');
   for i=1:n , for j=i:n ,                    % these
      v=(V(i,:).*Vc(j,:))';                   % take
      df(i,j)=v'*Pi*v;                        % O(n^4)
      df(j,i)=df(i,j); end, end,              % flops
   d=d-df\f; expd=exp(d);
   [Q,E,V]=svd(A*diag(expd),0);
   Vc=conj(V);   p=diag(E);
   f=real(V.*Vc)*p-w;
   err=norm(f); k=k+1; end

U=Q*V';  P=V*E*V';  D=diag(1./expd);
```

In Figure 3.2 a typical convergence graph is drawn. The solid line is 10-base logarithm of the norm of f when started from the trivial guess $d = 0$, and $(-\cdot-\cdot-)$ corresponds to the better starting value. The dashed line plots $\mathrm{cond}(f')$. Here A is a random 100×100 matrix (`randn(100)`).

3.3. Approximate Newton. In Newton's method above the computation of f' is the most flops consuming part $O(n^4)$. In each Newton step we solve $f'(d)\delta =$

$u - f(d)$, i.e. (see (2.6)),

$$\operatorname{diag}(V(\Pi \circ (V^*\mathbf{D}(\delta)V))V^*) = u - f(d). \tag{3.2}$$

Let us eigendecompose $\Pi = W\Lambda W^T$. Π has one positive and many small negative eigenvalues. We take an approximation

$$\Pi \approx \tilde{\Pi} = \sum_{|\lambda_j|>\varepsilon} \lambda_j \; w_j w_j^T,$$

where w_j's are the columns of W. Using $\tilde{\Pi}$ in (3.2) we get simplification:

$$\begin{aligned}
&\operatorname{diag}\Big(V\big(\tilde{\Pi} \circ (V^*\mathbf{D}(\delta)V)\big)V^*\Big) \\
&= \sum_{|\lambda_j|>\varepsilon} \lambda_j \; \operatorname{diag}\Big(V\big((w_j w_j^T) \circ (V^*\mathbf{D}(\delta)V)\big)V^*\Big) \\
&= \sum_{|\lambda_j|>\varepsilon} \lambda_j \; \operatorname{diag}\Big((V\mathbf{D}(w_j)V^*)\; \mathbf{D}(\delta)\; (V\mathbf{D}(w_j)V^*)\Big) \\
&= \sum_{|\lambda_j|>\varepsilon} \lambda_j \; (G_j \circ \bar{G}_j)\; \delta,
\end{aligned}$$

where $G_j = V\mathbf{D}(w_j)V^*$. Hence,

$$f'(d) \approx \sum_{|\lambda_j|>\varepsilon} \lambda_j \; (G_j \circ \bar{G}_j).$$

For $\varepsilon = 0.001$ we typically get four to six terms in the sum while the iteration count stays practically the same as for the genuine Newton method. This takes the Newton step back to $O(n^3)$ as can be seen from the tests of the next section.

The code for this approximate Newton is obtained by replacing the four lines ("`these take O(n^4) flops`") by lines

```
[W,lambda]=eig(Pi); df=zeros(n,n);
for j=1:n ,
   if abs(lambda(j,j)) > rtol ,
      G=V*diag(W(:,j))*V';
      df=df+lambda(j,j)*(G.*conj(G)); end,end
```

REMARK 7. *Here we just use the* `eig` *routine to get the eigendecomposition of* Π. *Since we want only a couple of largest (in modulus) eigenvalues and the corresponding eigenvectors, the Lanczos iteration should give extra savings.*

3.4. Comparison. In Figure 3.3 we have plotted the flop counts divided by n^3 of the three methods: F: the fixed point iteration with $\alpha = 2/3$ (—), N: the genuine Newton's method (- - -), and A: the approximate Newton's method (–·–·–).

We took 4 series of test problems. In each of these the (column) dimension (horizontal axis) grows from 10 to 100.

1. Complex random matrices (`A=randn(n)+sqrt(-1)*randn(n)`).
2. Hilbert matrices (`A=hilb(n)`).
3. Singular real matrices with rank $= \frac{n}{2}$ (`A=randn(n,n/2)*randn(n/2,n)`).
4. Random $2n \times n$ full rank matrices (`A=randn(2*n,n)`).

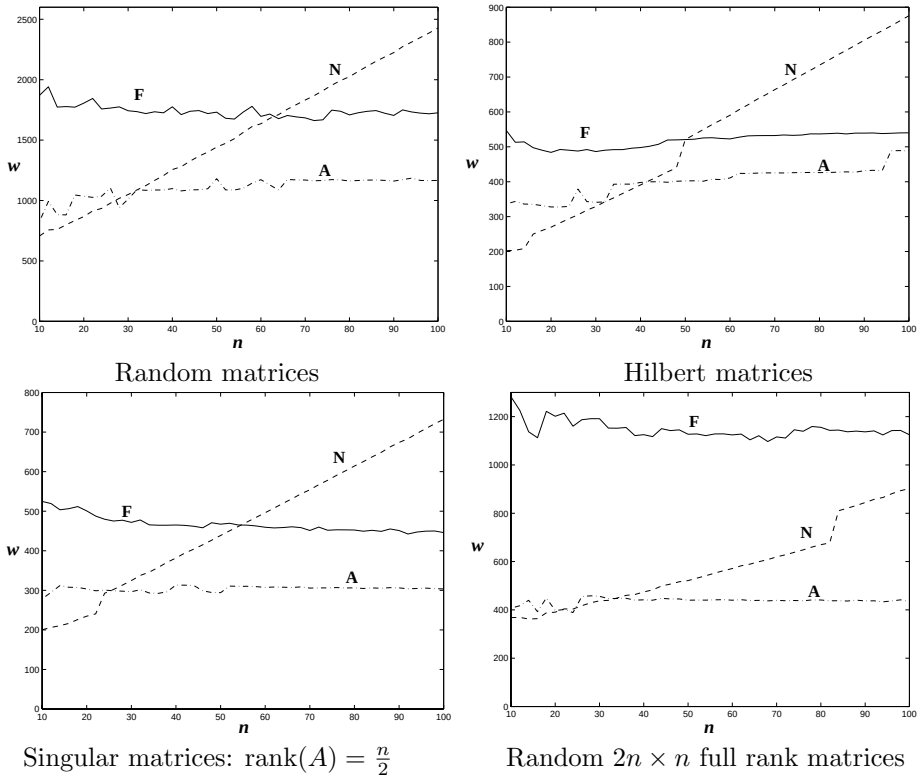


FIG. 3.3. *Scaled work* $w = \texttt{flops}/n^3$ *versus* n *for different methods.*

Methods F and A seem to have $O(n^3)$ complexity, while N is clearly $O(n^4)$.

REMARK 8. *The methods above are just simple first approaches. It will be interesting to study how the iterations for the polar decomposition (see, e.g.,* [4]*) can be adapted to this case.*

4. An application.

4.1. Parameterizing the orbit of a diagonalizable matrix. The *orbit* of a matrix is the set of matrices similar to it.

4.1.1. Complex case. Let the eigenvalues of $A \in \mathbb{C}^{n\times n}$ be distinct. Then, A is diagonalizable:

$$A = T\Lambda T^{-1},\ \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \in \mathcal{D}, \lambda_i \neq \lambda_j \text{ for } i \neq j.$$

The orbit of A is

$$\mathcal{S}(A) = \left\{ X\Lambda X^{-1} \,\middle|\, X \in \mathbb{C}^{n\times n},\ \det(X) \neq 0 \right\}.$$

If X and Y are nonsingular such that $X\Lambda X^{-1} = Y\Lambda Y^{-1}$, then $Y^{-1}X\Lambda = \Lambda Y^{-1}X$ and for $i \neq j$, $(\lambda_i - \lambda_j)(Y^{-1}X)_{ij} = 0$ holds. Hence, $\hat{D} = Y^{-1}X$ is diagonal. Write any diagonal nonsingular matrix as $\hat{D} = ED$, where $|E_{ii}| = 1$ and $D_{ii} > 0$ for all i.

Let $X = UP$, i.e., U is unitary and $P \in \mathcal{P}_I$, the set of Hermitian positive definite matrices with unit diagonal. Then, all matrices that transform Λ to the same matrix

as X does are of the form

$$Y = UPED.$$

Since $E^*PE \in \mathcal{P}_I$, too, and $\hat{X} = UEE^*PE$ also gives $\hat{X}\Lambda\hat{X}^{-1} = X\Lambda X^{-1}$ we still have to choose E. That means we have to choose coordinates in the set of unitary matrices modulo unitary diagonal scaling. We do this by requiring that the first nonzero entry in each column of U is real and positive. Let $\mathcal{U}$ denote the set of such unitary matrices. Then,

$$\mathcal{S}(A) = \left\{ UP\Lambda P^{-1}U^* \,\middle|\, U \in \mathcal{U},\ P \in \mathcal{P}_I \right\},$$

and for each $B \in \mathcal{S}(A)$ the factors U and P are uniquely defined.

To separate the *unitary orbit* $\mathcal{S}_{\mathcal{U}}(A)$ and the transversal part $\mathcal{S}_{\mathcal{P}}(A)$, take the refined polar decomposition $T = U_0 P_0 D$ with $U_0 \in \mathcal{U}$. Then,

$$\mathcal{S}_{\mathcal{U}}(A) = \left\{ UP_0\Lambda P_0^{-1}U^* \,\middle|\, U^*U = I \right\},$$
$$\mathcal{S}_{\mathcal{P}}(A) = \left\{ U_0 P\Lambda P^{-1}U_0^* \,\middle|\, P \in \mathcal{P}_I \right\}.$$

4.1.2. Real case. For $A \in \mathbb{R}^{n\times n}$ with distinct eigenvalues one might want to consider only the real orbit. If the eigenvalues are real, then we can proceed exactly as in the complex case, now restricting only to real matrices. This way we obtain unique coordinates for any real matrix in the orbit.

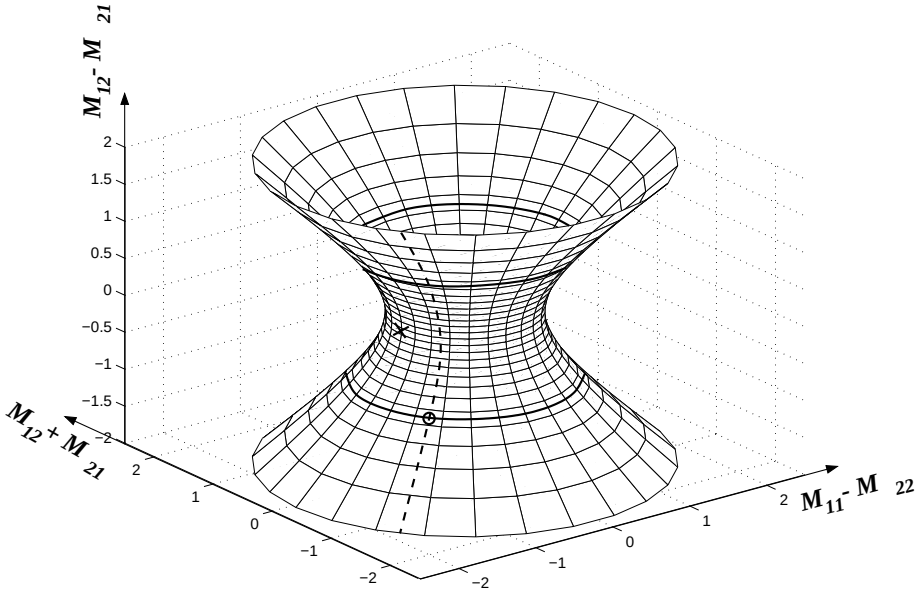


FIG. 4.1. *Orbit of a matrix with real eigenvalues.*

In Figure 4.1 the orbit of $A = \left[\begin{smallmatrix} 1.0 & -0.8 \\ 0.1 & 2.0 \end{smallmatrix}\right]$ is drawn. The displayed coordinates of $M \in \mathbb{R}^{2\times 2}$ are $M_{11} - M_{22}$, $M_{12} + M_{21}$, and $M_{12} - M_{21}$. The fourth coordinate $\mathrm{trace}(M)$ is not shown, since it is constant on orbits.

A is drawn as a small circle, and $\Lambda = \left[\begin{smallmatrix} 1.0877 & 0 \\ 0 & 1.9123 \end{smallmatrix}\right]$ is drawn as a cross. The two darker circles on the surface form the *orthogonal orbit* of A:

$$\mathcal{S}_{\mathcal{O}}(A) = \left\{ UAU^T \,\middle|\, U \in \mathbb{R}^{n\times n},\ U^TU = I \right\}.$$

The two parts correspond to orthogonal matrices with determinant ± 1, respectively. The dashed curve on the surface is the transversal part

$$\mathcal{S}_{\mathcal{P}}(A) = \left\{ U_0 P \Lambda P^{-1} U_0^T \,\middle|\, P \in \mathcal{P}_I \cap \mathbb{R}^{n\times n} \right\}.$$

If $A \in \mathbb{R}^{n\times n}$ has distinct eigenvalues, but some of them are complex, then it admits a real similarity transformation $A = T\Lambda T^{-1}$ to real block diagonal Λ, where the blocks are either real numbers or 2×2 blocks of the form

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}.$$

Let $\mathcal{D}_\Lambda$ denote the set of block diagonal matrices with the same block structure as Λ has. Now the diagonal matrices do not commute with Λ but those in $\mathcal{D}_\Lambda$ do. Hence, we want to consider refined polar decompositions

$$X = UP\tilde{D}$$

with orthogonal U, symmetric positive definite P having unit diagonal, and $\tilde{D} \in \mathcal{D}_\Lambda$ having nonnegative diagonal. Existence and uniqueness results can be obtained using similar techniques as in the proof of Theorem 2.1. The idea is to write $\tilde{D} = CD$, where $C, D \in \mathcal{D}_\Lambda$ and C is orthogonal and D is diagonal. This is to first transform a symmetric positive definite S to $\tilde{S} = C^T S C$, so that the diagonal pairs of $\tilde{S}$ corresponding to the 2×2 blocks of Λ match: $\tilde{S}_{j,j} = \tilde{S}_{j+1,j+1}$. Then, combine this with diagonal scaling. This combination can then be used[5] in F.

An algorithm for computing this is obtained by modifying the fixed point iteration. In the following code, vector `z` contains the starting indices of the 2×2 blocks, i.e., it defines $\mathcal{D}_\Lambda$.

```
function [U,P,D]=C_UPD(A,z)

% This function computes the refined polar decomposition of A
% A=UPD with D "real C-diagonal" determined by z.
% Here the fixed point iteration is used.

sz=size(A); n=sz(2); alpha=2/3;
expd=ones(n,1); d=zeros(n,1); C=eye(n);
err=1; k=0; tol=10^(-13);


while err > tol,
   [U,E,V]=svd(A*diag(expd),0);
   for j=z , jj=j:j+1 ;
      W=V(jj,:)*E*V(jj,:)';
      fi=atan((W(1,1)-W(2,2))/(2*W(1,2)))/2;
      c=cos(fi); s=sin(fi); C(jj,jj)=[c,-s;s,c]; end
   V=C*V;
   f=log((V.*conj(V))*diag(E));
   d=d+alpha*f; expd=exp(-d);
   err=norm(f); k=k+1; end

U=U*V';  P=V*E*V';  D=C*diag(1./expd);
```

[5]The details are not written here, since more general cases are under investigation.

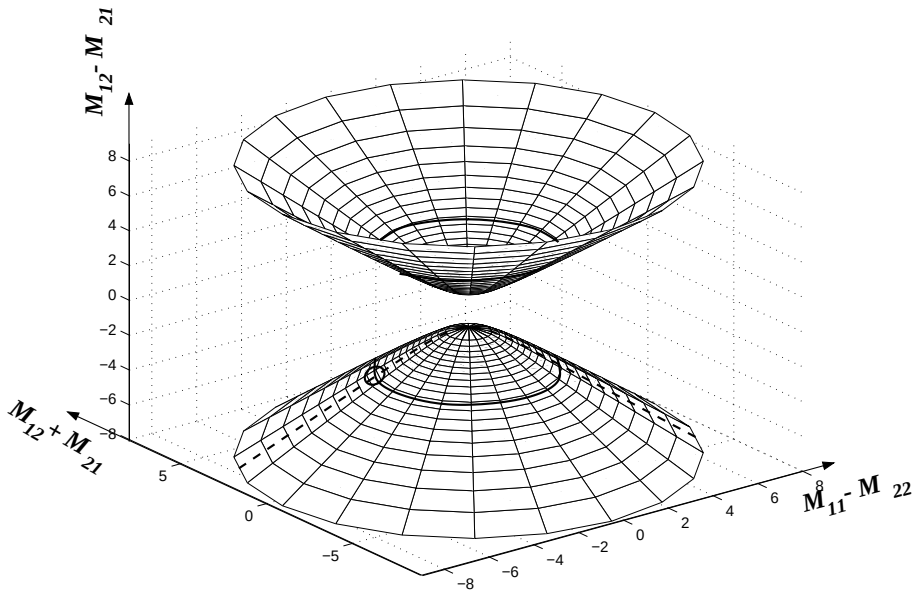


FIG. 4.2. *Orbit of a matrix with complex eigenvalues.*

In Figure 4.2, the orbit of $A = \left[\begin{smallmatrix} -0.7 & -1.0 \\ 2.5 & 2.3 \end{smallmatrix}\right]$ is drawn. A is shown as a small circle and $\Lambda = \left[\begin{smallmatrix} 0.8 & -0.5 \\ 0.5 & 0.8 \end{smallmatrix}\right]$ is on the top of the lower part. The two darker circles on the surface form again the orthogonal orbit, and the dashed curve is the transversal part.

REMARK 9. *The orthogonal orbits and transversal parts seem to intersect orthogonally (w.r.t.* $\langle A, B\rangle = \operatorname{tr}(AB^*)$*). This is true for* 2×2 *matrices, but not generally.*

5. Acknowledgments. The author is very grateful to the editor and the referees for their excellent work, many points to improve the paper, and for pointing out [6].

REFERENCES

[1] P. I. DAVIES AND N. J. HIGHAM, *Numerically stable generation of correlation matrices and their factors*, BIT, 2000, to appear.
[2] L. DIECI AND T. EIROLA, *On smooth decompositions of matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 800–819.
[3] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[4] N. J. HIGHAM AND R. S. SCHREIBER, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 648–655.
[5] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
[6] I. OLKIN AND J. W. PRATT, *A multivariate Tchebycheff inequality*, Ann. Math. Statist., 29 (1958), pp. 226–234.
[7] D. R. SMART, *Fixed Point Theorems*, Cambridge University Press, London, New York, 1974.
[8] A. VAN DER SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 23 (1969/1970), pp. 14–23.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 837–852

COMPUTING PROBABILISTIC BOUNDS FOR EXTREME EIGENVALUES OF SYMMETRIC MATRICES WITH THE LANCZOS METHOD*

JOS L. M. VAN DORSSELAER†, MICHIEL E. HOCHSTENBACH†, AND HENK A. VAN DER VORST†

Abstract. We study the Lanczos method for computing extreme eigenvalues of a symmetric or Hermitian matrix. It is not guaranteed that the extreme Ritz values are close to the extreme eigenvalues—even when the norms of the corresponding residual vectors are small. Assuming that the starting vector has been chosen randomly, we compute probabilistic bounds for the extreme eigenvalues from data available during the execution of the Lanczos process. Four different types of bounds are obtained using Lanczos, Ritz, and Chebyshev polynomials. These bounds are compared theoretically and numerically. Furthermore we show how one can determine, after each Lanczos step, a probabilistic upper bound for the number of steps still needed (without performing these steps) to obtain an approximation to the largest or smallest eigenvalue within a prescribed tolerance.

Key words. symmetric and Hermitian matrices, eigenvalues, Lanczos method, Ritz values, computation of probabilistic eigenvalue bounds, misconvergence, Lanczos polynomials, Ritz polynomials

AMS subject classification. 65F15

PII. S0895479800366859

1. Introduction.

Knowledge about the extreme eigenvalues of symmetric or Hermitian matrices is important in many applications. For example, the stability of processes involving such matrices is often governed by the location of their eigenvalues. The extreme eigenvalues can also be used to determine condition numbers, the field of values, and ε-pseudospectra of arbitrary matrices (see, e.g., [1, 12]). For small-sized matrices the eigenvalues can be computed by the QR-method (see, e.g., [2]), but this is not feasible for large matrices. A method which is often used in practice to compute a few extreme eigenvalues of large sparse symmetric or Hermitian matrices is the Lanczos method (see, e.g., [2, 7, 14]). The approximations of the eigenvalues obtained with the Lanczos method (the Ritz values) lie between the smallest and largest eigenvalue of the original matrix and one would like to know whether the largest (or smallest) Ritz value is sufficiently close to the largest (or smallest) eigenvalue of that matrix.

The classical a priori error estimates for the Lanczos method, established by Kaniel, Paige, and Saad (see, e.g., [2, 3, 6, 7, 10]) are not applicable in practice to obtain bounds on the spectrum of Hermitian matrices, because they involve knowledge about the eigenvalues and angles between the eigenvectors and the starting vector. Furthermore one should note that small residuals for the Ritz values *only* imply that these Ritz values are close to an eigenvalue, but it is not guaranteed that this eigenvalue is indeed the one we are looking for (cf., e.g., [8]). In fact, it is not possible to derive rigorous bounds on the spectrum from *any* possible starting vector: if the

*Received by the editors February 4, 2000; accepted for publication (in revised form) by L. Reichel July 14, 2000; published electronically December 7, 2000.
http://www.siam.org/journals/simax/22-3/36685.html
†Mathematical Institute, Utrecht University, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands (dorssela@math.uu.nl, hochsten@math.uu.nl, vorst@math.uu.nl). Part of the research of the first author was carried out at CWI (Amsterdam, The Netherlands).

starting vector is perpendicular to the eigenvector (or eigenspace in case of multiple eigenvalues) corresponding to the largest or smallest eigenvalue, it is impossible to obtain any information regarding this eigenvalue from the Lanczos process.

In this paper we derive various a posteriori bounds for the spectrum of real symmetric matrices using a probabilistic approach. Assuming that the starting vector of the Lanczos process is chosen randomly from the uniform distribution over the unit sphere, we derive, using data available while executing the Lanczos process, for every $\varepsilon \in (0,1)$ bounds for the spectrum with probability at least $1-\varepsilon$. No intrinsic properties of the matrix (apart from being symmetric) are required to compute our bounds. Polynomials related to the Lanczos process, namely the Lanczos polynomials and Ritz polynomials, are used to derive two types of such bounds. For symmetric positive definite matrices Kuczyński and Woźniakowski [5, Theorem 3] give, for arbitrary $t > 1$, an a priori upper bound for the probability that the largest eigenvalue is greater than t times the largest Ritz value; Chebyshev polynomials of the second kind are used to obtain these bounds. This result can be used to compute a posteriori probabilistic bounds for the spectrum while executing the Lanczos process, and bounds based on [5, Theorem 3] can be used for symmetric indefinite matrices as well. The fourth kind of bounds for the spectrum is obtained with Chebyshev polynomials of the first kind. The sharpness of the different bounds is analyzed theoretically and compared numerically. It turns out that the bounds based on Lanczos polynomials are the sharpest ones in most cases; however, the Ritz polynomials sometimes provide better bounds when the Lanczos method suffers from a misconvergence (i.e., the largest (or smallest) Ritz values in consecutive Lanczos steps seem to converge, but not to an extreme eigenvalue).

Apart from the bounds on the spectrum, we also study probabilistic bounds for the number of Lanczos steps needed to get an error (or relative error) in the largest or smallest eigenvalue that is smaller than a given tolerance. In [4, Theorem 4.2] the authors present a probabilistic upper bound for the number of Lanczos steps needed to yield a relative error in the largest eigenvalue of a symmetric positive definite matrix that is smaller than a given tolerance. For this special case numerical experiments demonstrate that our bound and the one from [4, Theorem 4.2] are almost the same. Furthermore, we provide upper bounds for the number of Lanczos steps needed to guarantee with probability at least $1-\varepsilon$ that either the spectrum lies between certain prescribed bounds, or that a misconvergence has occurred.

The results in this paper deal with the Lanczos process applied to real symmetric matrices and real starting vectors. This includes the case of Hermitian matrices, because the Lanczos method applied to a complex Hermitian matrix (with a complex starting vector) can be written as the application of the Lanczos method to a related real symmetric matrix of double size with a real starting vector (see Remark 2.1 for details).

All bounds discussed in this paper are easily implemented and can be computed with little effort while executing the Lanczos process.

The paper has been organized as follows. In section 2 some notations and definitions are introduced. Bounds based on Lanczos polynomials are presented in section 3, and bounds obtained with Ritz polynomials can be found in section 4. In section 5 we derive bounds from Chebyshev polynomials. The estimates for the number of Lanczos steps still to be done for sufficiently accurate approximations can be found in section 6.1, and the estimates for the number of Lanczos steps needed to obtain prescribed bounds for the spectrum or to detect misconvergence are given in section 6.2.

Numerical experiments are presented in section 7, and the conclusions can be found in section 8.

2. Preliminaries and notation. In this section we introduce some notations and present relevant properties of the Lanczos method. For an introduction to the Lanczos method and more details, as well as implementation issues, the reader may consult, e.g., [2, 7]. Throughout this paper we do not consider the effect of rounding errors.

The standard inner product on $\mathbb{R}^n$ will be denoted by $(\cdot,\cdot)$, and $\|\cdot\|$ stands for the Euclidean norm, and I is the $n \times n$ identity matrix.

Let A be a real symmetric $n \times n$ matrix with eigenvalues

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n. \tag{2.1}$$

The corresponding normalized eigenvectors x_j form an orthonormal basis of $\mathbb{R}^n$. We use the Lanczos method to approximate one or a few extreme eigenvalues of A. The unit starting vector is denoted by v_1 and can be written as

$$v_1 = \sum_{j=1}^{n} \gamma_j x_j. \tag{2.2}$$

If v_1 is chosen randomly from the uniform distribution with respect to the unit sphere, the dimension of the Krylov subspace

$$K_k(A, v_1) = \operatorname{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\}$$

is equal to k with probability one for k less than the number of distinct eigenvalues of A.

In the Lanczos process vectors v_k are generated by the three-term recurrence

$$\delta_k v_{k+1} = Av_k - \alpha_k v_k - \beta_{k-1} v_{k-1} \qquad \text{for} \quad k = 1, 2, 3, \ldots, \tag{2.3}$$

where $v_0 = 0$, $\beta_0 = 1$, $\alpha_k = (Av_k, v_k)$, $\beta_{k-1} = (Av_k, v_{k-1})$, and $\delta_k > 0$ is chosen such that $\|v_{k+1}\| = 1$. With this choice one has $\delta_k = \beta_k$ for $k \ge 1$. The vectors $v_1, v_2, \ldots, v_k$ form an orthonormal basis of the Krylov subspace $K_k(A, v_1)$. Let V_k be the $n \times k$ matrix of which v_j is the jth column. The *Ritz values* occurring in step k of the Lanczos process are the eigenvalues of the tridiagonal $k \times k$ matrix $T_k = V_k^{\mathrm{T}} A V_k$, and are denoted by

$$\theta_1^{(k)} < \theta_2^{(k)} < \cdots < \theta_k^{(k)};$$

the Ritz values satisfy $\theta_j^{(k)} > \lambda_j$ and $\theta_{k+1-j}^{(k)} < \lambda_{n+1-j}$ $(1 \le j \le k)$. We denote the eigenvectors of T_k by $s_j^{(k)}$: $T_k s_j^{(k)} = \theta_j^{(k)} s_j^{(k)}$ and the *Ritz vectors* by $y_j^{(k)} = V_k s_j^{(k)}$, where we assume that these Ritz vectors are normalized. We also introduce the residuals

$$r_j^{(k)} = Ay_j^{(k)} - \theta_j^{(k)} y_j^{(k)}.$$

Related to the three-term recursion (2.3) are the polynomials p_k of degree k defined by $p_{-1}(t) = 0$, $p_0(t) = 1$, and

$$\beta_k p_k(t) = (t - \alpha_k) p_{k-1}(t) - \beta_{k-1} p_{k-2}(t) \qquad \text{for} \quad k = 1, 2, 3, \ldots. \tag{2.4}$$

From (2.3) with $\delta_k = \beta_k$ and (2.4) it follows that

$$v_{k+1} = p_k(A)v_1 \qquad \text{for} \quad k = 1, 2, 3, \ldots.$$

The polynomials p_k are called the *Lanczos polynomials* with respect to A and v_1. Other polynomials related to the Lanczos method are the *Ritz polynomials* $q_j^{(k)}$ of degree $k-1$, which are characterized by the fact that

$$y_j^{(k)} = q_j^{(k)}(A)v_1 \qquad \text{for} \quad j = 1, 2, \ldots, k. \tag{2.5}$$

In the following sections estimates for the eigenvalues of A, based on Lanczos and Ritz polynomials, will be studied and compared. Therefore it is important to understand the relation between these polynomials. The polynomial p_k is a scalar multiple of the characteristic polynomial of the matrix T_k (cf., e.g., [7, section 7.3]), which implies that $\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_k^{(k)}$ are the zeros of p_k. From [7, section 12.3] it follows that these Ritz values without $\theta_j^{(k)}$ are the zeros of $q_j^{(k)}$. Hence $p_k(t) = c_j^{(k)}\,(t-\theta_j^{(k)})q_j^{(k)}(t)$ for a certain constant $c_j^{(k)}$.[1] Because $v_{k+1} = p_k(A)v_1 = c_j^{(k)}\,(A-\theta_j^{(k)}I)q_j^{(k)}(A)v_1 = c_j^{(k)}r_j^{(k)}$, we have $c_j^{(k)} = 1/\|r_j^{(k)}\|$, which yields the following relation between the Lanczos and Ritz polynomials:

$$p_k(t) = (t-\theta_j^{(k)})q_j^{(k)}(t) \,/\, \|r_j^{(k)}\| \qquad \text{for} \quad j = 1, 2, \ldots, k. \tag{2.6}$$

REMARK 2.1. *The Lanczos method described above can also be used to determine a few extreme eigenvalues of a complex Hermitian matrix A. The results in this paper are only valid for real symmetric matrices, but the Lanczos method for Hermitian matrices can be formulated in terms of real matrices and vectors. Let* $\operatorname{Re} A$ *and* $\operatorname{Im} A$ *be the real and imaginary part of A, respectively. The Lanczos method applied to the $2n \times 2n$ real symmetric matrix*

$$B = \begin{pmatrix} \operatorname{Re} A & -\operatorname{Im} A \\ \operatorname{Im} A & \operatorname{Re} A \end{pmatrix}$$

with starting vector $\binom{\operatorname{Re} v_1}{\operatorname{Im} v_1}$ *yields the same tridiagonal matrices T_k as the Lanczos method applied to A with starting vector v_1; this can be seen from taking the real and imaginary part of the three-term recurrence* (2.3). *The numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of B, but with multiplicity twice as large as for the matrix A. Therefore (probabilistic) bounds for the spectrum of B are (probabilistic) bounds for the spectrum of A as well.*

3. Spectral bounds using the Lanczos polynomial. In this section we will give probabilistic upper and lower bounds for the spectrum of A, based on Lanczos polynomials. For each step of the Lanczos process we obtain these bounds based on the information computed so far. No assumptions on the location or separation of the eigenvalues are required.

The Lanczos polynomials p_k are a byproduct of the process. They are usually small between $\theta_1^{(k)}$ and $\theta_k^{(k)}$ and increase rapidly outside this interval. We can exploit

[1] From this relation it follows that $q_j^{(k)}$ is a scalar multiple of $\prod_{i \neq j}(t-\theta_i^{(k)})$ and that polynomial is called a reduced Ritz polynomial in [11]. The relation between these polynomials and (2.5) also follows from [11, Formula (5.14)].

this fact: assuming that the starting vector has components in the direction of x_1 and x_n, we can provide upper and lower bounds for the spectrum of A.

From

$$1 = \|v_{k+1}\|^2 = \|p_k(A)v_1\|^2 = \sum_{j=1}^{n} \gamma_j^2 \, p_k(\lambda_j)^2$$

and $p_k(\lambda_n) > 0$ it follows that

$$1 \;\geq\; |\gamma_n| \, p_k(\lambda_n).$$

If γ_n is known, this estimate provides an upper bound λ^{up} for λ_n: let λ^{up} be the largest real zero of

$$f_L(t) = p_k(t) - 1/|\gamma_n|. \tag{3.1}$$

This number λ^{up} exists and satisfies $\lambda^{\mathrm{up}} > \theta_k^{(k)}$ because p_k is strictly increasing on $(\theta_k^{(k)}, \infty)$. The number λ^{up} can be determined by Newton's method or bisection. As a starting point for the Newton process one can take $\|A\|_\infty$ (the maximal row sum of the absolute values of the entries of A) or a previously computed upper bound for λ_n.

In practice we do not know γ_n, but we can determine the probability that $|\gamma_n|$ is smaller than a given (small) constant. Let S^{n-1} denote the $(n-1)$-dimensional unit sphere in $\mathbb{R}^n$. We assume that v_1 is chosen randomly with respect to the uniform distribution over S^{n-1}. Then, as a result, $(\gamma_1, \gamma_2, \ldots, \gamma_n)$ is also random with respect to the uniform distribution over S^{n-1} (cf., e.g., [4, p. 1116]). In the following lemma we compute the probability that $|\gamma_n|$ is smaller than δ.

LEMMA 3.1. *Assume that the starting vector v_1 has been chosen randomly with respect to the uniform distribution over the unit sphere S^{n-1} and let $\delta \in [0,1]$. Then*

$$P(|\gamma_n| \leq \delta) \;=\; 2\, B(\tfrac{n-1}{2}, \tfrac{1}{2})^{-1} \cdot \int_0^{\arcsin \delta} \cos^{n-2} t \, \mathrm{d}t,$$

where B denotes Euler's Beta function: $B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\mathrm{d}t$.

Proof. Define $S_\delta = \{\gamma \in S^{n-1} : |\gamma_n| < \delta\}$; we want to determine the ratio of the areas of the sets S_δ and S^{n-1}. The image of the map

$$\varphi \;:\; (-\pi, \pi) \;\times\; (-\tfrac{\pi}{2}, \tfrac{\pi}{2})^{n-2} \;\to\; S^{n-1}$$

defined by

$$\varphi : \begin{pmatrix} \alpha \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{n-2} \end{pmatrix} \mapsto \begin{pmatrix} \cos\alpha \cos\psi_1 \cos\psi_2 \cdots \cos\psi_{n-3} \cos\psi_{n-2} \\ \sin\alpha \cos\psi_1 \cos\psi_2 \cdots \cos\psi_{n-3} \cos\psi_{n-2} \\ \sin\psi_1 \cos\psi_2 \cdots \cos\psi_{n-3} \cos\psi_{n-2} \\ \vdots \\ \sin\psi_{n-3} \cos\psi_{n-2} \\ \sin\psi_{n-2} \end{pmatrix}$$

equals the sphere up to a negligible set. One can check that the associated Euclidean density is given by

$$\omega(\alpha, \psi_1, \psi_2, \ldots, \psi_{n-2}) \;=\; \cos\psi_1 \cdot \cos^2\psi_2 \cdots \cos^{n-2}\psi_{n-2}.$$

Therefore we can compute the areas of S_δ and S^{n-1} by integrating this density over the respective domains. Taking the ratio of the two results, we get

$$\begin{aligned} P(|\gamma_n| \le \delta) &= P(|\psi_{n-2}| \le \arcsin\delta) \\ &= 2\int_0^{\arcsin\delta} \cos^{n-2} t \,\mathrm{d}t \,\Big/ \int_{-\pi/2}^{\pi/2} \cos^{n-2} t \,\mathrm{d}t \\ &= 2\int_0^{\arcsin\delta} \cos^{n-2} t \,\mathrm{d}t \,\Big/ B(\tfrac{n-1}{2}, \tfrac{1}{2}), \end{aligned}$$

which proves the lemma. ∎

Now suppose we would like to have an upper bound for the spectrum of A that is correct with probability at least $1-\varepsilon$. Then we determine the value of δ for which

$$\int_0^{\arcsin\delta} \cos^{n-2} t \,\mathrm{d}t \;=\; \tfrac{\varepsilon}{2} B(\tfrac{n-1}{2}, \tfrac{1}{2}) \quad \left(= \varepsilon \int_0^{\pi/2} \cos^{n-2} t \,\mathrm{d}t \right) \tag{3.2}$$

holds, e.g., by using Newton's method. The integrals in (3.2) can be computed using an appropriate quadrature formula. We replace $|\gamma_n|$ in (3.1) by the value δ computed from (3.2) and determine the zero $\lambda^{\mathrm{up}} > \theta_k^{(k)}$. This λ^{up} is an upper bound for the spectrum of A with probability at least $1-\varepsilon$, and we call λ^{up} a probabilistic upper bound.

A lower bound λ^{low} for the spectrum of A with probability at least $1-\varepsilon$ can be obtained in a similar way. (Note that Lemma 3.1 remains valid if $|\gamma_n|$ is replaced by $|\gamma_1|$.) The only difference is that we have to separate the cases where k, the degree of p_k, is even ($p_k(t) \to +\infty$ for $t \to -\infty$) or odd ($p_k(t) \to -\infty$ for $t \to -\infty$). Hence we have proved the following theorem.

THEOREM 3.2. *Assume that the starting vector v_1 has been chosen randomly with respect to the uniform distribution over S^{n-1} and let $\varepsilon \in (0,1)$. Then λ^{up}, the largest zero of the polynomial*

$$f_L(t) = p_k(t) - 1/\delta \tag{3.3}$$

with δ given by (3.2)*, is an upper bound for the spectrum of A with probability at least $1-\varepsilon$, and λ^{low}, the smallest zero of*

$$f_L(t) = (-1)^k p_k(t) - 1/\delta, \tag{3.4}$$

is a lower bound for the spectrum of A with probability at least $1-\varepsilon$.

Note that if we are unlucky in choosing v_1, so that $|\gamma_n| < \delta$, then the computed bounds may or may not be correct; see section 7 for an illustration.

The determination of the lower and upper bounds from Theorem 3.2 is rather cheap in general (compared with a matrix-vector multiplication with A); the computation of $f_L(t)$ (using (2.4)) costs approximately $6k$ floating point operations. Note that the Ritz values and vectors are not needed to obtain these bounds of the spectrum. For very small k one cannot expect to obtain tight bounds, so it only makes sense to compute the zeros of (3.3) and (3.4) for k of moderate size. In practice one could, e.g., compute these zeros only every second or third Lanczos step until the bounds become sufficiently sharp.

4. Spectral bounds using Ritz polynomials. We can also try to obtain probabilistic upper and lower bounds for the spectrum of A using some Ritz polynomials $q_j^{(k)}$. The degree of these polynomials is one less than the degree of p_k, but while $p_k(\theta_k^{(k)}) = 0$, the polynomial $q_k^{(k)}$ has its last zero in $\theta_{k-1}^{(k)}$ and could be a competitor of p_k to give a possibly tighter upper bound. Similarly, $q_1^{(k)}$ may be used to obtain another lower bound.

We write $\theta_j^{(k)}$ as a Rayleigh quotient:

$$\theta_j^{(k)} = (Ay_j^{(k)}, y_j^{(k)}) = \sum_{i=1}^{n} \lambda_i \gamma_i^2 \, q_j^{(k)}(\lambda_i)^2. \tag{4.1}$$

First suppose that A is positive semidefinite. Then set $j = k$ to derive the inequality $\theta_k^{(k)} \geq \lambda_n \gamma_n^2 \, q_k^{(k)}(\lambda_n)^2$. Hence the zero $\lambda^{\text{up}} > \theta_k^{(k)}$ of

$$f_R(t) = t q_k^{(k)}(t)^2 \; - \; \theta_k^{(k)}/\gamma_n^2 \tag{4.2}$$

is an upper bound for λ_n. If γ_n is not known, one can obtain a probabilistic upper bound λ^{up} of λ_n with probability at least $1 - \varepsilon$, as in the previous section. (Replace γ_n in (4.2) by δ where δ satisfies (3.2).)

As in the previous section, if we happen to choose a v_1 so that $|\gamma_n| < \delta$, then we are not certain that the computed upper bound is correct. It can even happen that the largest zero λ^{up} of f_R with γ_n replaced by δ satisfies $\lambda^{\text{up}} < \theta_k^{(k)}$! See section 7 for an illustration.

When it is not known whether A is positive definite, we can obtain a probabilistic upper bound in the following way. Let $-\sigma < 0$ be a known lower bound for the spectrum of A: then the matrix $A + \sigma I$ is positive semidefinite. We get

$$\theta_k^{(k)} + \sigma = \sum_{i=1}^{n} (\lambda_i + \sigma) \, \gamma_i^2 \, q_k^{(k)}(\lambda_i)^2$$

with $\lambda_i + \sigma \geq 0$ for all i. The rightmost zero of

$$f_R(t) = (t + \sigma) q_k^{(k)}(t)^2 \; - \; (\theta_k^{(k)} + \sigma)/\gamma_n^2 \tag{4.3}$$

is an upper bound for the spectrum of A. Again, we can replace γ_n by the δ that satisfies (3.2) to compute a probabilistic upper bound.

For a lower bound, we use the polynomial $q_1^{(k)}$. If A is negative semidefinite, it follows from $\theta_1^{(k)} \leq \lambda_1 \gamma_1^2 \, q_1^{(k)}(\lambda_1)^2$ (cf. (4.1)) that the unique zero $\lambda^{\text{low}} < \theta_1^{(k)}$ of

$$f_R(t) = t q_1^{(k)}(t)^2 \; - \; \theta_1^{(k)}/\gamma_1^2 \tag{4.4}$$

is a lower bound for λ_1. Otherwise one has to use a shift $\tau > 0$ such that $A - \tau I$ becomes negative semidefinite and modify f_R in (4.4) accordingly. Of course the shifts σ and τ should be chosen as small as possible to get the best results.

The bounds discussed in this section can be determined for example by Newton's method or bisection. In order to compute $f_R(t)$ one has to know the largest or smallest Ritz value and the corresponding eigenvector of the tridiagonal matrix T_k. Apart from that, the computation of $f_R(t)$ is cheap. The determination of the bounds based on Ritz polynomials will be more expensive in general than the determination

of the bounds based on the Lanczos polynomials. (The Ritz values and vectors are not needed in the latter case.)

It is interesting to compare the sharpness of the bounds based on Ritz polynomials and those based on Lanczos polynomials. For simplicity we assume that A is positive semidefinite and compare the largest zero of (4.2) with the largest zero of (3.1). (The other cases, including those where shifts are used, can be analyzed in a similar way.) Consider the function

$$g(t) = \sqrt{t/\theta_k^{(k)}}\, q_k^{(k)}(t) \;-\; 1/|\gamma_n|; \tag{4.5}$$

the largest zero of g is the largest zero of f_R from (4.2). After some straightforward calculations, using (2.6) with $j = k$, one obtains that (with f_L as in (3.1) and g as in (4.5))

$$f_L(t) \;<\; g(t) \qquad \text{for} \quad \theta_k^{(k)} \;\leq\; t \;\leq\; (1+c)\,\theta_k^{(k)}$$

and

$$f_L(t) \;>\; g(t) \qquad \text{for} \quad t \;\geq\; (1+c+c^2)\,\theta_k^{(k)},$$

where $c = \|r_k^{(k)}\|/\theta_k^{(k)}$. The quantity c can be interpreted as an approximation of the relative error for the largest eigenvalue, and c will be small after sufficiently many Lanczos steps. For small c the Ritz polynomial provides a smaller upper bound for λ_n *only* when this upper bound is very close to $\theta_k^{(k)}$—but in that case the Lanczos polynomial yields a very tight upper bound as well. Hence, it is not likely that the bounds based on Ritz polynomials are sharper than the bounds obtained with the Lanczos polynomials—unless c is large. Numerical experiments illustrating these observations can be found in section 7.

5. Spectral bounds using Chebyshev polynomials. Chebyshev polynomials are often used to obtain error bounds for the Lanczos method; cf., e.g., [2, 5, 7]. In this section we explain how these polynomials can be used to obtain probabilistic upper and lower bounds for the spectrum of A, based on computations with the Lanczos method. One type of bounds follows easily from a result by Kuczyński and Woźniakowski [5, Theorem 3].

Let $c_j(t) = \cos(j \arccos t)$ be the *Chebyshev polynomial (of the first kind)* of degree j, with the usual extension outside the interval $[-1, 1]$. The polynomial

$$u_{j-1}(t) = \tfrac{1}{j}\, c_j'(t)$$

of degree $j-1$ is a *Chebyshev polynomial of the second kind* (cf. [9, p. 7]).

In [5, Theorem 3], the following result has been derived for symmetric positive definite matrices. Let $t > 1$ and v_1 be chosen randomly from the uniform distribution over S^{n-1}. Then

$$P(\lambda_n \leq t\,\theta_k^{(k)}) \;\geq\; 1 \;-\; 2\left(B(\tfrac{n-1}{2}, \tfrac{1}{2})\,\sqrt{t-1}\,u_{2(k-1)}(\sqrt{t}\,)\right)^{-1}. \tag{5.1}$$

(B is the Euler Beta function.) The estimate (5.1) can be generalized for symmetric indefinite matrices by using a shift σ such that $A + \sigma I$ is positive definite. Probability estimates for lower bounds of λ_1 can be obtained similarly. Along these lines we can derive bounds for the spectrum of A with probability at least $1 - \varepsilon$, and these results are presented in the following theorem.

THEOREM 5.1. *Let $\varepsilon \in (0,1)$ and $\sigma, \tau \in \mathbb{R}$ be such that $A + \sigma I$ is positive semidefinite, and $A - \tau I$ is negative semidefinite. Consider for $t \geq 1$ the function*

$$f(t) \;=\; \tfrac{\varepsilon}{2}\, B(\tfrac{n-1}{2}, \tfrac{1}{2})\, \sqrt{t-1}\, u_{2(k-1)}(\sqrt{t}\,) \;-\; 1 \tag{5.2}$$

(B is the Euler Beta function) and let $t_k > 1$ be the (unique) zero of f. Furthermore, let v_1 be chosen randomly from the uniform distribution over S^{n-1}. Then

$$\lambda^{\mathrm{up}} \;=\; t_k\, \theta_k^{(k)} \;+\; (t_k - 1)\sigma \tag{5.3}$$

is an upper bound for the spectrum of A with probability at least $1 - \varepsilon$, and

$$\lambda^{\mathrm{low}} \;=\; t_k\, \theta_1^{(k)} \;-\; (t_k - 1)\tau \tag{5.4}$$

is a lower bound for the spectrum of A with probability at least $1 - \varepsilon$.

The quantity t_k can be determined numerically. The numbers $u_j(t)$ can be computed from the three-term recurrence $u_j(t) = 2tu_{j-1}(t) - u_{j-2}(t)$ for $j \geq 2$, $u_0(t) = 1$, $u_1(t) = 2t$ (see, e.g., [9, p. 40]). From (5.3) and (5.4) it is clear that the shifts σ and τ should be chosen as small as possible (cf. section 4).

Other bounds for the spectrum of A can be obtained as follows, using Chebyshev polynomials of the first kind. Let $a < b$ and $c_j(t; a, b) = c_j(1 + 2(t-b)/(b-a))$ be the Chebyshev polynomial of degree j with respect to the interval $[a, b]$. With σ such that $A + \sigma I$ is positive semidefinite, we define the polynomial $h(t) = c_{k-1}(t; -\sigma, \theta_k^{(k)})$ and the vector $x = h(A)v_1 \in K_k(A, v_1)$. From $\theta_k^{(k)}(x,x) \geq (Ax, x)$ it follows that[2] the largest zero of

$$f_C(t) \;=\; (t - \theta_k^{(k)}) c_{k-1}(t; -\sigma, \theta_k^{(k)})^2 \;-\; (\theta_k^{(k)} + \sigma)/\gamma_n^2 \tag{5.5}$$

is an upper bound for λ_n. With γ_n replaced by the δ computed from (3.2), as in the previous sections, one obtains an upper bound λ^{up} for the spectrum of A with probability at least $1 - \varepsilon$. A lower bound for the spectrum of A can be obtained in a similar way, using $\theta_1^{(k)}(x,x) \leq (Ax,x)$ with $x = c_{k-1}(A; \theta_1^{(k)}, \tau)v_1$, where τ is such that $A - \tau I$ is negative semidefinite.

In order to compare the bounds derived along these lines with those obtained from Theorem 5.1, we first replace γ_n in (5.5) by δ and scale the interval $[-\sigma, \theta_k^{(k)}]$ to $[0, 1]$. The largest zero λ^{up} of (5.5) satisfies the equality $\lambda^{\mathrm{up}} = \widehat{t}\,\theta_k^{(k)} + (\widehat{t} - 1)\sigma$, where $\widehat{t} > 1$ is the unique zero of

$$g(t) \;=\; \delta\, \sqrt{t-1}\, c_{k-1}(t; 0, 1) \;-\; 1.$$

One can show that $c_{k-1}(t; 0, 1) = c_{2(k-1)}(\sqrt{t}\,; -1, 1)$ $(= c_{2(k-1)}(\sqrt{t}\,))$ for $t > 0$. This means that we have to compare the zeros of (5.2) and those of

$$g(t) \;=\; \delta\, \sqrt{t-1}\, c_{2(k-1)}(\sqrt{t}\,) \;-\; 1. \tag{5.6}$$

The relation between δ and $\frac{\varepsilon}{2}B(\frac{n-1}{2}, \frac{1}{2})$ is given by (3.2). One has $\delta > \frac{\varepsilon}{2}B(\frac{n-1}{2}, \frac{1}{2})$ for all $\varepsilon \in (0,1)$ and $n > 3$, but $\delta \approx \frac{\varepsilon}{2}B(\frac{n-1}{2}, \frac{1}{2})$ for ε and n of practical interest. For

[2] Invoke (2.2): use $\sum \gamma_j^2 \leq 1$ where the summation is with respect to those j satisfying $\lambda_j \leq \theta_k^{(k)}$ and $h(\lambda_j)^2 \leq 1$ for $\lambda_j \leq \theta_k^{(k)}$.

instance, $(\delta - \frac{\varepsilon}{2}B(\frac{n-1}{2}, \frac{1}{2}))/\delta \approx 2.6 \cdot 10^{-5}$ for $\varepsilon = 1.0 \cdot 10^{-2}$ and $n = 10^3, 10^4, 10^5, 10^6$. On the other hand one has the relation

$$u_{2(k-1)}(\sqrt{t}\,) \;=\; 2\,c_{2(k-1)}(\sqrt{t}\,) \;+\; u_{2(k-2)}(\sqrt{t}\,) \qquad \text{for} \quad t > 0$$

(cf., e.g., [9, p. 9]) so that $u_{2(k-1)}(\sqrt{t}\,) > 2c_{2(k-1)}(\sqrt{t}\,)$ for $t \geq 1$ and this implies, together with $\delta \approx \frac{\varepsilon}{2}B(\frac{n-1}{2}, \frac{1}{2})$, that the zero of (5.6) is larger than the zero of (5.2) in most applications. Hence, the upper bound λ^{up} from (5.3) is in general smaller than the upper bound obtained from (5.5), so Theorem 5.1 will produce sharper bounds than the construction described above. These observations are supported by numerical experiments in section 7.

6. Upper bounds for the number of Lanczos steps.

6.1. Bounds based on Theorem 5.1. Theorem 5.1 can also be used to compute a probabilistic upper bound for the number of Lanczos steps necessary to obtain a Ritz value close enough to λ_n in a relative or absolute sense. These estimates can be obtained while executing the Lanczos process. First we investigate how many Lanczos steps are needed to obtain a relative error that is smaller than a prescribed tolerance `tol` with probability at least $1 - \varepsilon$.

Suppose k steps of the Lanczos method have been performed and $\theta_k^{(k)} > 0$; if $\theta_k^{(k)} \leq 0$ the eigenvalue λ_n can be arbitrarily close to zero and the relative error $(\lambda_n - \theta_m^{(m)})/\lambda_n$ cannot be estimated properly. Let $m \geq k$ and let t_m be the zero of the function f in (5.2) with k replaced by m. It follows from (5.3) that

$$\frac{\lambda_n - \theta_m^{(m)}}{\lambda_n} \;\leq\; \frac{(t_m - 1)(\theta_m^{(m)} + \sigma)}{\lambda_n} \;\leq\; \frac{(t_m - 1)(\lambda_n + \sigma)}{\lambda_n} \;\leq\; \frac{(t_m - 1)(\mu + \sigma)}{\mu} \tag{6.1}$$

holds with probability at least $1 - \varepsilon$; here $\mu = \theta_k^{(k)}$ if $\sigma \geq 0$, and $\mu \geq \lambda_n$ (e.g., $\mu = \|A\|_\infty$; one should not take a probabilistic upper bound for λ_n) whenever $\sigma < 0$; σ is as in Theorem 5.1. The requirement $(t_m - 1)(\mu + \sigma)/\mu \leq \texttt{tol}$ is equivalent to $t_m \leq 1 + \texttt{tol} \cdot \mu/(\mu + \sigma)$, and the smallest integer m, for which the quantity t_m from (5.2) satisfies

$$t_m \;\leq\; 1 + \texttt{tol} \cdot \mu/(\mu + \sigma), \tag{6.2}$$

is an upper bound for the number of Lanczos steps necessary to provide an approximation $\theta_m^{(m)}$ to λ_n that satisfies $(\lambda_n - \theta_m^{(m)})/\lambda_n \leq \texttt{tol}$ with probability at least $1 - \varepsilon$. Note that in case $\sigma > 0$ the right-hand side of (6.2) increases with k, so that the smallest number m satisfying (6.2) may decrease during the execution of the Lanczos process.

For symmetric positive definite matrices an upper bound m for the number of Lanczos steps which yields an approximation to the largest eigenvalue, such that the relative error is bounded by `tol` with probability at least $1 - \varepsilon$, has been given in [4, Theorem 4.2]: the number m should satisfy

$$1.648\,\sqrt{n}\,\mathrm{e}^{-(2m-1)\sqrt{\texttt{tol}}} \;\leq\; \varepsilon. \tag{6.3}$$

Numerical experiments show that (6.3) yields almost the same upper bound as (6.2) with $\sigma = 0$ (in most cases the bounds were exactly the same, while the difference was at most two steps); this is not surprising in view of the discussion in [5, p. 679]. However,

(6.2) can be used for indefinite matrices as well, as long as $\theta_k^{(k)} > 0$. Furthermore, for symmetric positive definite matrices smaller numbers m may be obtained when (6.2) is applied with $\sigma < 0$.

To estimate the number of steps, still necessary to have the absolute error $\lambda_n - \theta_m^{(m)} \leq \texttt{tol}$ with probability at least $1 - \varepsilon$, we proceed as follows. If m satisfies the requirement (cf. (6.1))

$$(t_m - 1)(\mu + \sigma) \;\leq\; \texttt{tol}, \tag{6.4}$$

with $\mu \geq \lambda_n$ (μ should not be a probabilistic upper bound), the equality $\lambda_n - \theta_m^{(m)} \leq \texttt{tol}$ holds with probability at least $1 - \varepsilon$. The smallest integer m satisfying (6.4) can be computed. Note that (6.4) is also valid when $\theta_k^{(k)} \leq 0$ and we do not have to distinguish between the cases $\sigma \geq 0$ and $\sigma < 0$.

Estimates for the number of Lanczos steps, to be done so that the (relative) error in the smallest eigenvalue is less than tol with probability at least $1 - \varepsilon$, can be derived in a similar way.

6.2. Upper bounds for the number of Lanczos steps in case of misconvergence. Suppose that after sufficiently many Lanczos steps the largest Ritz value seems to have converged to an eigenvalue: $\theta_k^{(k)} \approx \theta_{k-1}^{(k-1)}$ for several consecutive k and $\|r_k^{(k)}\|$ is small. It is known that $|\theta_k^{(k)} - \lambda_j| \leq \|r_k^{(k)}\|$ for a certain eigenvalue λ_j (see, e.g., [7, section 4.5]), and in most cases the largest Ritz value will have converged to the largest eigenvalue λ_n, but it may also happen that $\theta_k^{(k)}$ is not close to λ_n (misconvergence); this can happen, e.g., if $|\gamma_n|$ is very small. Below we show how one can determine a probabilistic upper bound for the number of Lanczos steps needed after which one can conclude that either $\lambda_n < \lambda$ holds for a given constant λ, or a misconvergence has been detected, i.e., $\lambda_n > \theta_k^{(k)} + \|r_k^{(k)}\|$.

Let $m > k$ and g be a polynomial of degree $m - 1$, and $x = g(A)v_1 \in K_m(A, v_1)$. If $\lambda_n > \theta_k^{(k)} + \|r_k^{(k)}\|$, the inequality

$$(Ag(A)v_1, g(A)v_1) \;>\; \left(\theta_k^{(k)} + \|r_k^{(k)}\|\right) (g(A)v_1, g(A)v_1) \tag{6.5}$$

is satisfied for a certain m and a suitable polynomial g: the Ritz polynomial $q_m^{(m)}$ maximizes the Rayleigh quotient $(Ag(A)v_1, g(A)v_1)/(g(A)v_1, g(A)v_1)$ but $q_m^{(m)}$ is not available after k steps of the Lanczos process, so we will consider another polynomial of degree $m - 1$. Rewriting (6.5) using (2.2) gives

$$\begin{aligned} \left(\lambda_n - (\theta_k^{(k)} + \|r_k^{(k)}\|)\right) \gamma_n^2 \, g(\lambda_n)^2 \quad &> \quad \left(\theta_k^{(k)} + \|r_k^{(k)}\| - \lambda_{n-1}\right) \gamma_{n-1}^2 \, g(\lambda_{n-1})^2 \\ &+ \quad \sum_{j=1}^{n-2} \left(\theta_k^{(k)} + \|r_k^{(k)}\| - \lambda_j\right) \gamma_j^2 \, g(\lambda_j)^2. \end{aligned} \tag{6.6}$$

In order to satisfy (6.6) with m as small as possible we search for a polynomial g that resembles the Ritz polynomial $q_m^{(m)}$. We have $q_k^{(k)}$ to our disposal, and therefore we take $g(t) = q_k^{(k)}(t)\, h(t)$ with h a suitable polynomial of degree $m - k$. We assume that $|\theta_k^{(k)} - \lambda_{n-1}| \leq \|r_k^{(k)}\|$ (with $\|r_k^{(k)}\|$ small); this assumption is likely to be realistic in case of a misconvergence. In order to amplify the effect of $q_k^{(k)}$ in (6.6) we choose h such that h is large in λ_n and small in $\lambda_1, \ldots, \lambda_{n-2}$. Hence $h(t) = c_{m-k}(t; \lambda_1, \lambda_{n-2})$ would

be a proper choice, but λ_1 and λ_{n-2} are not known, so we replace both quantities. Again let $-\sigma \leq \lambda_1$, and assume that $\lambda_{n-2} \leq \theta_{k-1}^{(k)} + \|r_{k-1}^{(k)}\|$; we now define

$$g(t) \;=\; q_k^{(k)}(t)\, c_{m-k}\big(t; -\sigma, \theta_{k-1}^{(k)} + \|r_{k-1}^{(k)}\|\big).$$

If we replace in the right-hand side of (6.6) the quantity $\theta_k^{(k)} + \|r_k^{(k)}\| - \lambda_{n-1}$ by $2\,\|r_k^{(k)}\|$, γ_{n-1}^2 by 1, $g(\lambda_{n-1})$ by $g(\theta_k^{(k)} + \|r_k^{(k)}\|)$, and $g(\lambda_j)$ by M, where

$$M \;=\; \max\{\, |q_k^{(k)}(t)| \;:\; -\sigma \leq t \leq \theta_{k-1}^{(k)} + \|r_{k-1}^{(k)}\| \,\},$$

then the inequality

$$\begin{aligned} \big(\lambda_n - (\theta_k^{(k)} + \|r_k^{(k)}\|)\big)\, g(\lambda_n)^2 \quad &> \quad 2\,\|r_k^{(k)}\|\, g\big(\theta_k^{(k)} + \|r_k^{(k)}\|\big)^2 / \gamma_n^2 \\ &+ \quad M^2\big(\theta_k^{(k)} + \|r_k^{(k)}\| + \sigma\big) / \gamma_n^2 \end{aligned} \tag{6.7}$$

implies (6.6) (cf. the derivation of (5.5), which is based on the same ideas). We now replace λ_n in (6.7) by the given constant λ and γ_n by δ, where $|\gamma_n| \geq \delta$ holds with probability $1 - \varepsilon$. We determine the smallest integer $m > k$ such that

$$\begin{aligned} \big(\lambda - (\theta_k^{(k)} + \|r_k^{(k)}\|)\big)\, g(\lambda)^2 \quad &> \quad 2\,\|r_k^{(k)}\|\, g\big(\theta_k^{(k)} + \|r_k^{(k)}\|\big)^2 / \delta^2 \\ &+ \quad M^2\big(\theta_k^{(k)} + \|r_k^{(k)}\| + \sigma\big) / \delta^2 \end{aligned} \tag{6.8}$$

is satisfied and perform $m - k$ Lanczos steps to obtain $\theta_m^{(m)}$. If $\theta_m^{(m)} < \theta_k^{(k)} + \|r_k^{(k)}\|$, then (6.5) and (6.6) are violated. This implies that (6.7) does not hold if, e.g., $\lambda_{n-1} \leq \theta_{k-1}^{(k)} + \|r_{k-1}^{(k)}\|$. (This will be satisfied in most cases.) From the fact that (6.7) is violated and (6.8) holds we conclude that $\lambda_n < \lambda$ holds with probability at least $1 - \varepsilon$.

If $\theta_m^{(m)} > \theta_k^{(k)} + \|r_k^{(k)}\|$, we know that a misconvergence has occurred and we do not know whether $\lambda_n < \lambda$ is satisfied or not. In the latter case one may repeat the above construction with k replaced by m.

These ideas can also be used to investigate whether or not the smallest Ritz value has converged to λ_1.

7. Numerical experiments. In this section we compare the different bounds derived in the previous sections. All experiments are carried out with Matlab on a SUN workstation. Without loss of generality we can restrict ourselves to diagonal matrices A (cf. [4, section 6]): this will reduce the influence of rounding errors on our computations. For analysis it is also convenient to know the eigenvalues and eigenvectors of A. The vector v_1 is chosen randomly from the uniform distribution over the unit sphere S^{n-1}; in [4, p. 1116] it is explained how this can be done.

In our first example we take

$$n \;=\; 1000, \quad A \;=\; \mathrm{diag}(1, 2, \ldots, 1000). \tag{7.1}$$

Let $\varepsilon = 0.01$, i.e., we are looking for bounds of the spectrum that are 99% reliable. From (3.2) one obtains $\delta = 3.97 \cdot 10^{-4}$. We checked that our randomly chosen starting vector v_1 satisfied $|\gamma_1| > \delta$ and $|\gamma_n| > \delta$, so the computed probabilistic bounds are true bounds for the spectrum of A. We have performed 100 Lanczos steps. The shifts (see sections 4 and 5) used in our computations are $\sigma = 0$ and $\tau = \lambda_n = 1000$. The results are displayed in Figure 7.1.

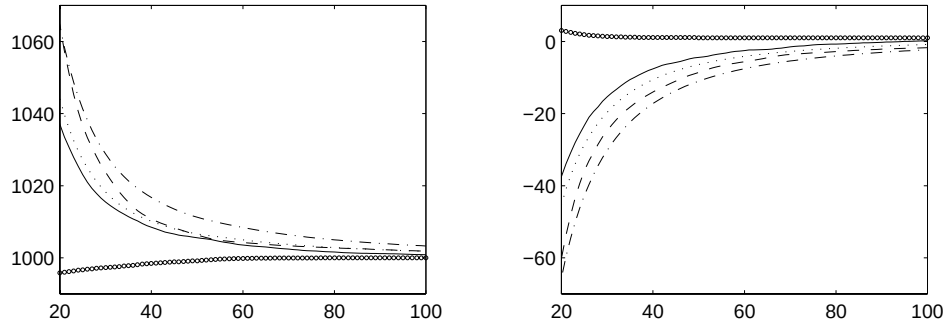


FIG. 7.1. *Probabilistic bounds for the spectrum of A. Solid curves correspond to the bounds based on Lanczos polynomials, the dashed curves correspond to bounds based on Ritz polynomials, the dotted curves correspond to bounds obtained from Theorem* 5.1*, and the dash-dotted curves correspond to* (5.5)*. The left figure shows the upper bounds and the right figure the lower bounds. The largest Ritz values (left picture) and smallest Ritz values (right picture) are indicated by small circles.*

We see that the Lanczos polynomials provide the sharpest bounds and (5.5) yields the worst bounds. In section 4 it has already been explained why the Lanczos polynomials may provide better bounds than the Ritz polynomials. Furthermore, it may not be a surprise that the Lanczos polynomials produce better bounds than the Chebyshev polynomials, because more information regarding the actual Lanczos process is used in the construction of the Lanczos polynomials. The relationship between the different bounds based on Chebyshev polynomials is in agreement with the discussion on this topic in section 5. We repeated the same experiment with other random starting vectors v_1, and the bounds behaved similarly as those displayed in Figure 7.1.

We also investigated how many Lanczos steps are needed to obtain an approximation to λ_n with a relative error less than a prescribed tolerance `tol`. Again we set $\sigma = 0$, so that (6.2) reduces to $t_m \leq 1 + \texttt{tol}$; the upper bound m for the number of Lanczos steps does not depend on the matrix A or the starting vector v_1 and can be computed in advance. The results are displayed in Table 7.1. We see that the upper bound m from (6.2) is much larger than k_1, the actual number of steps needed to obtain a relative error smaller than `tol`; this has already been observed in other examples for the upper bound obtained with (6.3) [4, 5]. We also observe that $m > k_2$, the number of steps needed to obtain $(\lambda^{\mathrm{up}} - \theta_k^{(k)})/\lambda^{\mathrm{up}} \leq \texttt{tol}$ with λ^{up} the upper bound obtained from the Lanczos polynomial of degree k. This is not surprising in view of the results from Figure 7.1, because m is related to the upper bound determined with Theorem 5.1, and these bounds are not as sharp as those based on Lanczos polynomials. Instead of performing m Lanczos steps, it may be useful in practice to compute $(\lambda^{\mathrm{up}} - \theta_k^{(k)})/\lambda^{\mathrm{up}}$ while executing the Lanczos method and check whether this quantity is smaller than `tol` or not.

We have repeated the experiments described above with $\varepsilon = 0.001$ (instead of $\varepsilon = 0.01$). The behavior of the bounds is the same as for $\varepsilon = 0.01$, but of course the bounds are further away from the spectrum of A. In order to compare the different bounds, let λ^{up} be an upper bound corresponding to $\varepsilon = 0.01$ (determined with one of the four techniques discussed here), and let $\widetilde{\lambda}^{\mathrm{up}}$ be the upper bound determined with the same technique but with $\varepsilon = 0.001$. For all four techniques we observed that $1 < (\widetilde{\lambda}^{\mathrm{up}} - \lambda_n)/(\lambda^{\mathrm{up}} - \lambda_n) < 2.2$ for $20 \leq k \leq 100$ (k denotes the number

TABLE 7.1

The second column displays the smallest integer m satisfying (6.2) with $\sigma = 0$. The smallest integer k_1 for which $(\lambda_n - \theta_k^{(k)})/\lambda_n \leq$ `tol` is shown in the third column, and the smallest integer k_2 with $(\lambda^{\mathrm{up}} - \theta_k^{(k)})/\lambda^{\mathrm{up}} \leq$ `tol`, where λ^{up} is the upper bound for λ_n obtained with the Lanczos polynomial of degree k, is listed in the fourth column of the table.

`tol`	m	k_1	k_2
$5.0 \cdot 10^{-2}$	20	5	18
$1.0 \cdot 10^{-2}$	44	11	40
$5.0 \cdot 10^{-3}$	61	17	55
$1.0 \cdot 10^{-3}$	136	48	97

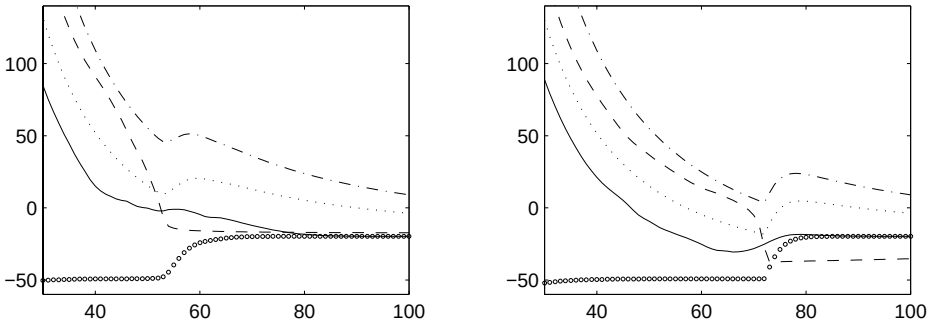


FIG. 7.2. *"Upper bounds" for the spectrum of A, obtained with two different starting vectors; the starting vector for the left picture satisfies $|\gamma_n| > \delta$, while $|\gamma_n| < \delta$ for the starting vector used to produce the right picture. Solid curves correspond to the bounds based on Lanczos polynomials, the dashed curves correspond to bounds based on Ritz polynomials, the dotted curves correspond to bounds obtained from Theorem 5.1, and the dash-dotted curves correspond to (5.5). The largest Ritz values are indicated by small circles.*

of Lanczos steps) and the same holds for $(\lambda_1 - \widetilde{\lambda}^{\mathrm{low}})/(\lambda_1 - \lambda^{\mathrm{low}})$, where the lower bounds λ^{low} and $\widetilde{\lambda}^{\mathrm{low}}$ are defined analogously. Hence the behavior of the bounds for the spectrum of A does not change much when ε is decreased from 0.01 to 0.001, which is reasonable because the polynomials used to derive the bounds grow fast outside the spectrum of A.

The second example comes from the discretization of the Laplace operator on the unit square with homogeneous Dirichlet boundary conditions. When the standard second order finite difference scheme with uniform meshwidth equal to 1/33 (in both directions) is used, one obtains a symmetric matrix of order $n = 32^2 = 1024$ with eigenvalues

$$33^2(-4 + 2\cos(\tfrac{i\pi}{33}) + 2\cos(\tfrac{j\pi}{33})), \qquad i, j = 1, 2, \ldots, 32 \tag{7.2}$$

(see, e.g., [13, section 6.5]). Let A be the diagonal matrix of order 1024 with these eigenvalues on its diagonal in increasing order. Note that A is negative definite.

We have computed bounds for the spectrum of A with $\varepsilon = 0.01$ (which yields $\delta = 3.92 \cdot 10^{-4}$ by (3.2)), $\sigma = -\lambda_1$ and $\tau = 0$, using different randomly chosen starting vectors. For most starting vectors the bounds behave similarly as in the first example and we will not consider this further. Instead we deal with two different starting vectors that provide a different behavior for the upper bounds (similar results can be obtained for lower bounds as well), and the results can be found in Figure 7.2.

In the left picture we see what can happen if $|\gamma_n|$ is small ($|\gamma_n| = 5.46 \cdot 10^{-4}$ for this example), but still greater than δ. The Ritz polynomials provide the sharpest bounds at a certain stage of the Lanczos process. At that stage the misconvergence behavior of the Lanczos process (cf., e.g., [8]) is discovered: for $37 \leq k \leq 49$ one has $|\lambda_{n-1} - \theta_k^{(k)}| \leq 0.15$ ($\lambda_{n-1} = -49.22\cdots$), and the largest Ritz values seem to converge to a number close to the (double) eigenvalue λ_{n-1}. For larger values k the Lanczos process notices the existence of a larger eigenvalue ($\lambda_n = -19.72\cdots$) and starts to converge to this eigenvalue. At the stage of the Lanczos process where the misconvergence behavior is discovered, the norm of the residual usually increases strongly (for example, $\|r_{42}^{(42)}\| = 5.65$ and $\|r_{55}^{(55)}\| = 102$) and a large residual norm may explain why the Ritz polynomials provide sharper bounds than the Lanczos polynomials (see the discussion at the end of section 4). However, for larger k the bounds based on Lanczos polynomials are again the sharpest ones. The misconvergence of the Lanczos process also causes a hump in the upper bounds obtained with the Chebyshev polynomials. Finally we note that the upper bounds obtained with the Lanczos polynomials are much sharper than those obtained with the Chebyshev polynomials.

In the right figure the behavior is shown for a starting vector for which, in contrary to our assumption, $|\gamma_n| < \delta$ ($|\gamma_n| = 3.13 \cdot 10^{-5}$). This means that the probabilistic upper bounds for λ_n need not to be true bounds, and the right picture in Figure 7.2 shows that at certain stages of the Lanczos process the Lanczos and Ritz polynomials provide bounds that are actually smaller than λ_n. The Chebyshev bounds follow the jump of the Ritz values at the discovering of the misconvergence, as in the left picture. At that stage the Lanczos bound corrects its value to give a tight bound, but the Ritz bound fails completely: the upper bound stays far below the largest Ritz value.

In the third example we illustrate the theory of section 6.2. We take

$$n = 1000, \quad A = \operatorname{diag}(1, 2, \ldots, 999, 1020). \tag{7.3}$$

We set $\sigma = -\lambda_1$ and the starting vector v_1 is chosen as follows: $\gamma_1 = \gamma_2 = \gamma_{n-2} = \gamma_{n-1} = c$, $\gamma_j = 10^{-3}c$ ($3 \leq j \leq n-3$), $\gamma_n = 10^{-6}c$, and the constant c is such that $\sum \gamma_j^2 = 1$. For $k = 34$ we have $\theta_k^{(k)} = \lambda_{n-1} - 3.20 \cdot 10^{-5}$, $\|r_k^{(k)}\| = 7.3 \cdot 10^{-2}$ so that $\lambda_n > \theta_k^{(k)} + \|r_k^{(k)}\|$. We now determine the smallest integer m for which (6.8) holds. We take $k = 34$, $\lambda = \lambda_n$, $\delta = \gamma_n = 5.0 \cdot 10^{-7}$ and $M = 2.11$. The smallest m satisfying (6.8) is $m = 69$. The Lanczos process finds the largest eigenvalue λ_n earlier: one has, e.g., $\theta_{50}^{(50)} = \lambda_n - 2.4 \cdot 10^{-2}$, $\theta_{60}^{(60)} = \lambda_n - 5.5 \cdot 10^{-5}$ and $\theta_{69}^{(69)} = \lambda_n - 2.4 \cdot 10^{-7}$. This behavior is not surprising: the Ritz polynomial $q_m^{(m)}$ maximizes the Rayleigh quotient $(Ag(A)v_1, g(A)v_1)/(g(A)v_1, g(A)v_1)$ and several other estimates used in the derivation of (6.8) may not be sharp as well.

8. Conclusion. Using the fact that the Lanczos, Ritz, and Chebyshev polynomials increase rapidly outside the smallest interval containing the Ritz values, we have derived probabilistic bounds for the spectrum of a symmetric matrix. These bounds can be computed while executing the Lanczos process. From theoretical arguments supported by experiments, we conclude that the bounds obtained with the Lanczos polynomials are generally sharper than those derived from Chebyshev polynomials. In most cases the bounds based on Lanczos polynomials are also sharper than the bounds found with Ritz polynomials—unless the norm of the corresponding residual is relatively large (which occurs if the Lanczos method suffers from a misconvergence).

The bounds corresponding to the Lanczos polynomials are cheap to compute, because the Ritz values are not required. When the Ritz values are available, it is

useful to compute the bounds based on these polynomials as well, because they might be sharper; in that case it can indicate a misconvergence of the Lanczos method. The bounds based on Theorem 5.1, using Chebyshev polynomials of the second kind, may be determined as well because they can be computed cheaply when the Ritz values are known. The bounds obtained from Theorem 5.1 are sharper than those derived from (5.5), which are based on Chebyshev polynomials of the first kind, in all cases of practical interest; hence it seems not useful to determine the latter ones.

Chebyshev polynomials may also be used to determine probabilistic bounds for the number of Lanczos steps still to be done to get bounds for the (relative) error which are smaller than the desired tolerance. However, our experiments suggest that these bounds are much larger than the actual number of Lanczos steps still necessary to get an approximation which is sufficiently accurate. From their derivation (6.1) it is clear that one cannot expect a proper estimation of the number of steps required if the bounds from Theorem 5.1 are far from sharp.

A combination of Ritz and Chebyshev polynomials can be used to obtain probabilistic bounds for the number of Lanczos steps needed such that one can decide that either the spectrum lies between certain prescribed bounds or a misconvergence has occurred.

Acknowledgments. The authors wish to thank Joop Kolk for discussions regarding Lemma 3.1 and Gerard Sleijpen for pointing out reference [5].

REFERENCES

[1] T. Braconnier and N. J. Higham, *Computing the field of values and pseudospectra using the Lanczos method with continuation*, BIT, 36 (1996), pp. 422–440.
[2] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[3] S. Kaniel, *Estimates for some computational techniques in linear algebra*, Math. Comp., 20 (1966), pp. 369–378.
[4] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.
[5] J. Kuczyński and H. Woźniakowski, *Probabilistic bounds on the extremal eigenvalues and condition number by the Lanczos algorithm*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 672–691.
[6] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, University of London, London, 1971.
[7] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980.
[8] B. N. Parlett, H. Simon, and L. M. Stringer, *On estimating the largest eigenvalue with the Lanczos algorithm*, Math. Comp., 38 (1982), pp. 153–165.
[9] T. J. Rivlin, *Chebyshev Polynomials*, 2nd ed., John Wiley, New York, 1990.
[10] Y. Saad, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.
[11] A. van der Sluis and H. A. van der Vorst, *The convergence behavior of Ritz values in the presence of close eigenvalues*, Linear Algebra Appl., 88/89 (1987), pp. 651–694.
[12] L. N. Trefethen, *Computation of pseudospectra*, Acta Numer., 8 (1999), pp. 247–295.
[13] R. S. Varga, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.
[14] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 853–873

ON ACCURATE QUOTIENT SINGULAR VALUE COMPUTATION IN FLOATING-POINT ARITHMETIC*

ZLATKO DRMAČ† AND ELIZABETH R. JESSUP‡

Abstract. This paper presents a new algorithm for floating-point computation of the quotient singular value decomposition of an arbitrary matrix pair $(A, B) \in \mathbf{R}^{m\times n} \times \mathbf{R}^{p\times n}$. In the case of full column rank A, the new algorithm computes all finite quotient singular values with high relative accuracy if $\min\{\kappa_2(AD),\ D \text{ diagonal}\}$ is moderate and if an accurate rank revealing LU factorization of B is possible. Numerical experiments show that in such a case the new algorithm computes the quotient singular values of all pairs (AD, D_1BD_2) with nearly the same accuracy, where D, D_1, D_2 are arbitrary diagonal nonsingular matrices.

Key words. generalized singular value decomposition, generalized eigenvalue problem, Jacobi method, regularization, relative accuracy, singular value decomposition

AMS subject classifications. 65F15, 65F25, 65G05

PII. S0895479896310548

1. Introduction.

In [36], Van Loan introduces a new matrix decomposition of a general matrix pair $(A, B) \in \mathbf{C}^{m\times n} \times \mathbf{C}^{p\times n}$ $(m \geq n)$. He proves that there always exist unitary matrices U, V and a nonsingular matrix X such that U^*AX and V^*BX are diagonal matrices, and he defines the B-singular values of A as the elements of the set $\{\sigma \geq 0 \ : \ \det(A^*A - \sigma^2 B^*B) = 0\}$. Paige and Saunders [27] remove the minor constraint $m \geq n$ and reformulate the original decomposition to avoid the nonunitary matrix X. They show that there exist unitary matrices U, V, Q, diagonal matrices Σ_A, Σ_B, and a nonsingular triangular matrix R such that $U^*AQ = \Sigma_A[\mathbf{O}, R]$, $V^*BQ = \Sigma_B[\mathbf{O}, R]$. This form is equivalent to Van Loan's with $X = Q(I \oplus R^{-1})$. In the nomenclature of various generalizations of the singular value decomposition (SVD) proposed by De Moor and Golub [6], this decomposition is called the *quotient singular value decomposition* (QSVD) of (A, B), and the B-singular values of A are the *quotient singular values* of (A, B). If B is square and nonsingular, then the QSVD of (A, B) is equivalent to the SVD of AB^{-1}.

The QSVD is a powerful tool in both theoretical analysis and the numerical solution of problems like regularization and various types of constrained least squares [4], [20], [37], [38], [25]. It also arises in the symmetric definite generalized eigenvalue problem $Kx = \lambda Mx$, where the positive definite matrices K and M are factored as $K = A^*A$ and $M = B^*B$, respectively. The quotient singular values of (A, B) are then the square roots of the eigenvalues of $K - \lambda M$. An important advantage of using (A, B) instead of the pencil $K - \lambda M$ is that $\kappa_2(A) = \sqrt{\kappa_2(K)}$, $\kappa_2(B) = \sqrt{\kappa_2(M)}$. (Here $\kappa_2(A) = \|A\|_2\|A^\dagger\|_2$ is the spectral condition number, where $A^\dagger$ is the Moore–Penrose generalized inverse and $\|\cdot\|_2$ is the matrix norm induced by the Euclidean vector norm.)

*Received by the editors October 14, 1996; accepted for publication (in revised form) by F. Luk July 14, 1999; published electronically December 7, 2000. This research was supported by National Science Foundation grants ACS-9357812 and ASC-9625912, Department of Energy grant DE-FG03-94ER25215, the Intel Corporation, and Croatian Ministry of Science and Technology grant 037012.
http://www.siam.org/journals/simax/22-3/31054.html

†Department of Mathematics, University of Zagreb, Zagreb, Croatia (drmac@math.hr).

‡Department of Computer Science, University of Colorado, Boulder, CO 80309-0430 (jessup@cs.colorado.edu).

In this paper, we propose a new efficient and numerically stable algorithm for computation of the quotient singular values of a real pair $(A, B) \in \mathbf{R}^{m\times n} \times \mathbf{R}^{p\times n}$ in floating-point arithmetic. To explain the main ideas of the new approach, we first briefly review some known QSVD algorithms. The first algorithm is based on the connection between the QSVD and the cosine-sine decomposition (CSD) of a partitioned orthonormal matrix [31], [32], [39]. This algorithm first computes the QR factorization $\mathcal{G} \equiv \left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right] = \mathcal{Q}R$ and then it computes the CSD of $\mathcal{Q} = \left[\begin{smallmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \end{smallmatrix}\right]$, where $\mathcal{Q}$ is partitioned so that $\mathcal{Q}_1 \in \mathbf{R}^{m\times n}$, $\mathcal{Q}_2 \in \mathbf{R}^{p\times n}$. The main shortcoming of the CSD approach is that computation with the $(m + p) \times n$ matrix $\mathcal{G}$ is not efficient and precludes backward stability.

The second algorithm avoids the use of the matrix $\mathcal{G}$ and transforms A and B separately. It has two phases: (i) using an algorithm of Bai and Zha [3], a general pair (A, B) is reduced to an equivalent pair of upper triangular matrices $(A_\triangleright, B_\triangleright)$ with nonsingular $B_\triangleright$; (ii) using an algorithm of Paige [26], implemented carefully by Bai and Demmel [2], the procedure completes with the QSVD computation of $(A_\triangleright, B_\triangleright)$. It is shown in [2], [3] that both phases of the algorithm are backward stable in the Frobenius matrix norm $\|\cdot\|_F$. That is, floating-point computation is equivalent to exact computation with $(A + \Delta A, B + \Delta B)$, where $\|\Delta A\|_F/\|A\|_F$ and $\|\Delta B\|_F/\|B\|_F$ are of order machine precision $\mathbf{u}$ times a moderate function of matrix dimensions. This algorithm is superior to the CSD approach, and it is implemented as the LAPACK 2.0 [1] procedure `SGGSVD()`.

Both the CSD algorithm and the LAPACK algorithm are designed to use only orthogonal transformations. This restriction is unnecessary because the quotient singular values are in fact invariant under the more general transformation $(A, B) \mapsto (A', B') = (U^T AX, V^T BX)$, where U, V are arbitrary orthogonal matrices and X is an arbitrary nonsingular matrix.

The first method for quotient singular value computation using nonorthogonal transformations is proposed by Deichmöller and Veselić [8]. It is an implicit variant of the Falk–Langemeyer method [18] for the diagonalization of matrix pencils. An error analysis for the full column rank case is given in [7], and the method is further analyzed and modified in [14].

The second method that is not entirely based on orthogonal transformations is the *tangent algorithm* proposed by Drmač [14], [16]. In this method, a pair (A, B) of full column rank matrices is replaced by an equivalent pair (A', B'), and the SVD of the explicitly computed matrix $A'B'^{-1}$ is computed using the Jacobi SVD method [21], [12], [15]. An important novelty in the tangent algorithm is that the iterative part is performed on a single matrix. The computed quotient singular values of (A, B) approximate the exact values up to an error of (cf. [16])

$$\max_{1\le i\le n} \frac{|\delta\sigma_i|}{\sigma_i} \le g(m, n, p) \cdot \mathbf{u} \cdot K_c(A, B), \quad K_c(A, B) = \kappa_2(AD_A^{-1}) + \kappa_2(BD_B^{-1}),$$

where $g(\cdot,\cdot,\cdot)$ is a modestly growing function of matrix dimensions, and D_A, D_B are diagonal matrices of Euclidean column norms of A and B, respectively. Furthermore, the computed quotient singular values correspond to the exact quotient singular values of a pair $(A + \delta A, B + \delta B)$, where, for all i, the values of $\|\delta Ae_i\|_2/\|Ae_i\|_2$ and $\|\delta Be_i\|_2/\|Be_i\|_2$ are small. (Here e_i denotes the ith column of the identity matrix I.) It is a remarkable fact that this method has the same accuracy within the family of all pairs (AD_1, BD_2), where D_1 and D_2 are arbitrary diagonal nonsingular matrices. This accuracy property is shared neither by the CSD algorithm nor the LAPACK procedure.

Our new algorithm is an improvement and a generalization of the tangent algorithm and it is designed to reduce a general pair (A, B) to a (regular) pair with finite quotient singular values. We use both orthogonal and nonorthogonal transformations.

The nonorthogonal transformations are introduced in section 2.1, and the new algorithm is described in detail in section 2.2. In the new algorithm, computation involving both matrices (A and B or their submatrices) is on the BLAS level 3 [13] (at most two calls of `xTRSM()` and one call of `xGEMM()`). The remaining nontrivial operations include the LU factorization with complete pivoting, the QR factorization with column pivoting (which can be implemented using BLAS 3 operations; cf. [30]) and the ordinary SVD of a full rank matrix. This modular structure provides a solid basis for a high performance QSVD computation on both serial and parallel computers.

The analysis in section 2.3 shows that the computed regular pair has small backward error $(\Delta A, \Delta B)$, where, for all i, $\|\Delta A e_i\|_2/\|A e_i\|_2$ is small and ΔB is the backward error from the LU or the QR factorization (with complete pivoting). It is also shown that the computed quotient singular values have similar small backward error. That is, the combination of the new reduction algorithm and the tangent algorithm for regular pairs is backward stable.

In section 2.4, we show that in the case of full column rank A and full rank B, the relative accuracy of the computed quotient singular values of (A, B) is determined by the accuracy of the floating-point QR factorization of A and the LU factorization with complete pivoting of B.

Finally, in section 3 we present the results of rigorous numerical testing that demonstrate the numerical robustness of our software. The numerical results confirm the analysis from section 2.4, and they also indicate that, in the case of full column rank A and full (column or row) rank B, the algorithm computes the quotient singular values of all pairs $\{(AD, D_1 B D_2),\ D, D_1, D_2 \ \text{ diagonal matrices}\}$ with nearly the same relative accuracy. We recommend our algorithm as the method of choice for quotient singular value computation in floating-point arithmetic.

2. Reduction algorithm based on the LU factorization. In the LAPACK 2.0 library [1], the procedure `SGGSVD()` for the QSVD computation has two stages: (i) reduction of a general pair (A, B) to a regular pair (A', B') of upper triangular matrices; (ii) QSVD computation of the regular pair (A', B'). (The pair (A', B') is called *regular* if B' is a full column rank matrix.) Stage (i) of `SGGSVD` uses an algorithm of Bai and Zha [3], while stage (ii) is an implementation of the algorithm of Paige (see [26], [2]). Working with a regular pair has several advantages in Paige's algorithm because its implementation in the case of an irregular triangular pair is quite complicated [3], [2]. Bai and Zha's reduction algorithm is based on the QR factorization and the URV decomposition

$$A = U \begin{bmatrix} \mathbf{O} & R \\ \mathbf{O} & \mathbf{O} \end{bmatrix} V^T, \quad U, V \ \text{ orthogonal}, \quad R \ \text{ triangular nonsingular}.$$

Since Paige's algorithm is based on plane rotations, the whole process is therefore completed using solely orthogonal transformations.

In our new algorithm, we adopt a similar two-stage strategy but with different realizations of the two main stages. The main differences are that (i) we use an initial column scaling to preserve the numerical stability in subsequent steps; (ii) we replace the URV factorization with another simple factorization based on certain nonorthogonal but well-conditioned matrices; (iii) we use an algorithm from [16] to

compute the QSVD of the regular pair using the SVD of a single matrix instead of using simultaneous transformations of a pair of matrices.

2.1. QRT and LUT factorizations. The key feature of the new algorithm is a new simple factorization of a general matrix. It is based on pivoted QR or LU factorization and, in the case of a full column rank matrix, it reduces to QR or LU, respectively. In the general case, it provides a simple way to cancel out columns that are identified in pivoted QR or LU as linearly dependent on the remaining ones.

THEOREM 2.1. *Let $B \in \mathbf{R}^{p\times n}$ and $r_B = \mathrm{rank}(B)$. Then there exist an orthogonal $p \times p$ matrix Q, permutation matrices Π_1, Π_2, an $r_B \times (n - r_B)$ matrix X, and an $r_B \times r_B$ upper triangular nonsingular matrix R such that*

$$\Pi_1 B \Pi_2 = Q \begin{bmatrix} R & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} I & X \\ \mathbf{O} & I \end{bmatrix}. \tag{2.1}$$

Furthermore, there exist permutation matrices P_1, P_2, a unit lower trapezoidal $p \times r_B$ matrix L, a unit upper triangular $r_B \times r_B$ matrix U, a diagonal nonsingular $r_B \times r_B$ matrix Δ, and an $r_B \times (n - r_B)$ matrix Y such that

$$P_1 B P_2 = L \Delta [U, \mathbf{O}] \begin{bmatrix} I & Y \\ \mathbf{O} & I \end{bmatrix}. \tag{2.2}$$

The factorizations (2.1) *and* (2.2) *define the QRT and the LUT factorizations of B, respectively.*

Proof. Let

$$\Pi_1 B \Pi_2 = Q \begin{bmatrix} R & \hat{R} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \quad Q^T Q = Q Q^T = I$$

be a rank revealing QR factorization of B, where the row permutation matrix Π_1 is optional (to enhance numerical stability) and R is an $r_B \times r_B$ upper triangular nonsingular matrix. In this work, we use the column pivoting of Golub [19]. Define $X = R^{-1}\hat{R}$. Then

$$\begin{bmatrix} R & \hat{R} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} I & -X \\ \mathbf{O} & I \end{bmatrix} = \begin{bmatrix} R & \hat{R} - RX \\ \mathbf{O} & \mathbf{O} \end{bmatrix} = \begin{bmatrix} R & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}.$$

Similarly, let $P_1 B P_2 = L\Delta[U, \hat{U}]$ be a rank revealing LU factorization. Define $Y = U^{-1}\hat{U}$ and note that

$$\begin{bmatrix} U & \hat{U} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} I & -Y \\ \mathbf{O} & I \end{bmatrix} = \begin{bmatrix} U & \hat{U} - UY \\ \mathbf{O} & \mathbf{O} \end{bmatrix} = \begin{bmatrix} U & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}. \qquad \square$$

The triangular transformations used in Theorem 2.1 are well-conditioned and easy to invert.

PROPOSITION 2.1. *Let*

$$T \equiv T(X) = \begin{bmatrix} I & X \\ \mathbf{O} & I \end{bmatrix}. \tag{2.3}$$

Then $T(X)^{-1} = T(-X)$ and $\max\{\|T(X)\|_2,\ \|T(X)^{-1}\|_2\} \le 1 + \|X\|_2$. Further, let R, U, X, Y be as in Theorem 2.1*, where the QR factorization is computed with Golub's column pivoting and the LU factorization is computed with complete pivoting.*

If D_R and D_U are diagonal matrices that satisfy $|D_R| \geq |\mathbf{diag}(R)|$, $|D_U| \geq |\mathbf{diag}(U)|$, where $\mathbf{diag}(R) = \mathrm{diag}(R_{11}, \ldots, R_{r_B,r_B})$, then

$$\|X\|_2 \leq \sqrt{r_B(n-r_B)}\|R^{-1}D_R\|_2, \quad \|Y\|_2 \leq \sqrt{r_B(n-r_B)}\|U^{-1}D_U\|_2.$$

The values of $\|R^{-1}D_R\|_2$ and $\|U^{-1}D_U\|_2$ are bounded by moderate functions of the dimensions (cf. Stewart [34]).

2.2. The algorithm. The new algorithm has two stages. We first reduce the general pair (A, B) to a regular pair that has the same finite quotient singular values as (A, B). In the second stage, the regular pair is reduced to a single matrix and the only iterative part in the computation is the ordinary SVD.

ALGORITHM 2.1 (LU-based QSVD computation).

Input. $(A, B) \in \mathbf{R}^{m\times n} \times \mathbf{R}^{p\times n}$, $\mathrm{rank}(A) = r_A$, $\mathrm{rank}(B) = r_B$.

Stage A. Reduction.

Step 0. Scaling. Define $D_A = \mathrm{diag}(\|Ae_i\|_2)$. If some column of A is zero, then replace the corresponding diagonal entry in the definition of D_A by a small nonzero scalar. Compute $A^{(0)} = AD_A^{-1}$, $B^{(0)} = BD_A^{-1}$. The new pair $(A^{(0)}, B^{(0)})$ is equivalent to (A, B).

Step 1. Compute the LU factorization with complete pivoting of $B^{(0)}$:

$$\Pi_1 B^{(0)} \Pi_2 = LU = \begin{bmatrix} L^{(1,1)} \\ L^{(2,1)} \end{bmatrix} [U^{(1,1)}, U^{(1,2)}], \quad L^{(1,1)}, U^{(1,1)} \in \mathbf{R}^{r_B \times r_B},$$

where $L^{(1,1)}$ is unit lower triangular and $U^{(1,1)}$ is upper triangular and nonsingular. Partition $A^{(0)}\Pi_2$ accordingly:

$$A^{(0)}\Pi_2 = [A_{11}^{(0)}, A_{12}^{(0)}], \quad A_{11}^{(0)} \in \mathbf{R}^{m\times r_B}.$$

Step 2. Set $U^{(1,1)}$ to I_{r_B} and $U^{(1,2)}$ to zero. Compute $X = A_{11}^{(0)}(U^{(1,1)})^{-1}$ and

$$A^{(1)} \equiv [A_{11}^{(1)}, A_{12}^{(1)}] = [X, A_{12}^{(0)} - XU^{(1,2)}].$$

Step 3. If $A_{12}^{(1)}$ is not void ($r_B < n$), compute a rank revealing QR factorization of $A_{12}^{(1)}$,

$$A_{12}^{(1)}\Pi_3 = Q_{12}^{(2)} \begin{bmatrix} A_{12}^{(2)} & \hat{A}_{12}^{(2)} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \quad A_{12}^{(2)} \in \mathbf{R}^{r_A^{(1)} \times r_A^{(1)}}, \quad r_A^{(1)} = \mathrm{rank}(A_{12}^{(1)}),$$

where $A_{12}^{(2)}$ is upper triangular and nonsingular. Set $\hat{A}_{12}^{(2)}$ to zero. Update $A_{11}^{(1)}$ by $A_{11}^{(1)} \mapsto A_{11}^{(2)} = (Q_{12}^{(2)})^T A_{11}^{(1)}$. With suitable row and column partition the new pair is

$$(A^{(2)}, B^{(2)}) = \left(\begin{bmatrix} A_{11,1}^{(2)} & A_{12}^{(2)} & \mathbf{O} \\ A_{11,2}^{(2)} & \mathbf{O} & \mathbf{O} \end{bmatrix}, \begin{bmatrix} L^{(1,1)} \\ L^{(2,1)} \end{bmatrix} [I_{r_B}, \mathbf{O}, \mathbf{O}] \right). \tag{2.4}$$

Set $A_{12}^{(2)}$ to I and $A_{11,1}^{(2)}$ to $\mathbf{O}$. If $r_B = n$, we have $A^{(2)} = A_{11,2}^{(2)} = A^{(1)}$.

Return. At the end of Step 3 the reduced pair of full column rank matrices is

$$(A_{11,2}^{(2)}, L) \in \mathbf{R}^{(m-r_A^{(1)})\times r_B} \times \mathbf{R}^{p\times r_B}. \tag{2.5}$$

Stage B. QSVD of the regular pair (2.5). Use the tangent algorithm from [16].

Step 0. Compute $D_{11,2} = \mathrm{diag}(\|A^{(2)}_{11,2}e_i\|_2)$ and $(A^{(2)}_{11,2})_c = A^{(2)}_{11,2}D^{-1}_{11,2}$, $L_1 = LD^{-1}_{11,2}$.

Step 1. Compute the QR factorization with column pivoting, $L_1\Pi_4 = Q_L\left[\begin{smallmatrix} R_L \\ \mathbf{O}\end{smallmatrix}\right]$.

Step 2. Compute $F = (A^{(2)}_{11,2})_c\Pi_4 R_L^{-1}$ by solving the equation $FR_L = (A^{(2)}_{11,2})_c\Pi_4$.

Step 3. Compute the SVD of F using the Jacobi SVD algorithm, $\Sigma = VFJ^T$. (V, J orthogonal.)

Step 4. Compute the matrices $Z = D^{-1}_{11,2}\Pi_4 R_L^{-1}J^T$ and $W = Q_L(J^T \oplus I_{p-r_B})$.

Return. The QSVD of $(A^{(2)}_{11,2}, L)$ is $VA^{(2)}_{11,2}Z = \Sigma$, $W^TLZ = [I, \mathbf{O}]^T$. The Paige–Saunders form is obtained using the RQ factorization $Z^{-1} = R_ZQ_Z$. Then $VA^{(2)}_{11,2}Q_Z^T = \Sigma R_Z$, $W^TLQ_Z^T = \left[\begin{smallmatrix} I \\ \mathbf{O}\end{smallmatrix}\right]R_Z$. Note that $Z^{-1} = UR_L\Pi_4^TD_{11,2}$ is as easy to compute as Z and that $\kappa_2(Z) = \kappa_2(L)$ is moderate.

Output. Assembling all transformations from Stages A and B gives the QSVD of (A, B). For instance, in the case $m \geq n$, $r_B = p < n$, the pair (A, B) is equivalent to the pair (Σ_A, Σ_B) with

$$\Sigma_A = \begin{bmatrix} I_p & \mathbf{O} \\ \mathbf{O} & \Sigma \end{bmatrix}, \quad \Sigma_B = [\mathbf{O}_{p\times(n-p)}, I_p].$$

The orthogonal form of the QSVD can be obtained from the nonorthogonal form using the RQ factorization as on Return from Stage B.

Remark 2.1. The LU factorization of $B^{(0)}$ can be replaced with the QR factorization with complete pivoting (see Powell and Reid [29], Björck [4, section 4.4.3], Cox and Higham [5]). In that case, Stage B is simpler because L is replaced with an orthogonal matrix and $F = A^{(2)}_{11,2}$. (Generally, the tangent algorithm can be modified to include row pivoting in the QR factorization in Step 1. A similar comment holds for the QR factorization in Step 3 of Stage A.) From the numerical point of view, the difference between the two variants is precisely the difference in accuracy between the LU and the QR factorization with complete pivoting. For the sake of brevity, in this paper we analyze only the LU variant of the algorithm. We also note that in Stage A we can write $LU = LD_uU_d$, where $D_u = \mathrm{diag}(U_{11}, \ldots, U_{r_B,r_B})$, and use LD_u and U_d instead L and U, respectively.

Remark 2.2. Computation and stability analysis of the matrices that transform the pair (A, B) into (Σ_A, Σ_B) are quite complicated, and in this work we consider only the quotient singular values.

2.3. Backward stability analysis. We now consider the backward stability of Algorithm 2.1. In a three-step scheme, we analyze Stage A, then Stage B, and, finally, we show how the backward errors from Stage B propagate backward into the initial data.

For ease of notation, we assume that the matrices are permuted so that no row or column interchanges are necessary in the LU factorization in Step 1. Let $\tilde{D}_A \approx D_A$ be the computed diagonal matrix of the Euclidean norms of A's columns, and let $(A^{(0)}, B^{(0)})$ denote the computed scaled matrices. Then there exist small elementwise backward error matrices δA, δB such that $|\delta A| \leq \mathbf{u}|A|$, $|\delta B| \leq \mathbf{u}|B|$ and

$$A^{(0)} = (A + \delta A)\tilde{D}_A^{-1}, \quad B^{(0)} = (B + \delta B)\tilde{D}_A^{-1}. \tag{2.6}$$

The columns of $A^{(0)}$ are nearly of unit Euclidean length, that is, there exists a small constant η, $0 \leq \eta \leq O(m\mathbf{u})$, such that for all i, $1 - \eta \leq \|A^{(0)}e_i\|_2 \leq 1 + \eta$. (Using double precision accumulation reduces the bound for η to the order of $\mathbf{u} + O(m\mathbf{u}^2)$.)

We also note that the transformation (2.6) is an exact equivalence transformation between the pairs $(A+\delta A, B+\delta B)$ and $(A^{(0)}, B^{(0)})$ so that the perturbation is merely an elementwise rounding independent of the accuracy of the matrix $\tilde{D}_A$.

If $\tilde{L}\tilde{U}$ is the computed LU factorization of $B^{(0)}$, then there exists a backward error $\delta B^{(0)}$ such that (cf. [23, Chapter 9])

$$\tilde{L}\tilde{U} = B^{(0)}+\delta B^{(0)}, \quad |\delta B^{(0)}| \le \varepsilon_{LU}|\tilde{L}|\cdot|\tilde{U}|, \quad \text{where} \quad \varepsilon_{LU} \le \frac{\min\{p,n\}\mathbf{u}}{1-\min\{p,n\}\mathbf{u}}. \tag{2.7}$$

(The matrix absolute value and the matrix inequality are understood elementwise.) Thus, using the matrix $\tilde{L}\tilde{U}$ corresponds to an exact computation with the backward perturbed matrix

$$\begin{aligned} &\tilde{L}(\tilde{U}\tilde{D}_A) = B + \Delta B, \\ &\text{where} \quad \Delta B = \delta B + \delta B^{(0)}\tilde{D}_A, \quad |\Delta B| \le \mathbf{u}|B| + \varepsilon_{LU}|\tilde{L}|\cdot|\tilde{U}\tilde{D}_A|. \end{aligned} \tag{2.8}$$

Relation (2.8) suggests that the backward error in B is similar to the backward error one would have obtained by computing the LU factorization of B. If we consider $(A^{(0)}, B^{(0)})$ as the initial pair, then the backward error $\delta B^{(0)}$ is precisely the backward error in the LU factorization. The computed rank of B is denoted r_B and, in the backward error analysis, we do not analyze whether or not it has been determined correctly.

The computation in Step 2 of the algorithm involves standard BLAS 3 operations for triangular systems (`xTRSM()`) and matrix-matrix multiplication (`xGEMM()`). The following two propositions analyze the backward stability of Step 2.

PROPOSITION 2.2. *Let $\tilde{A}_{11}^{(1)}$ be the solution of the matrix equation $X\tilde{U}^{(1,1)} = A_{11}^{(0)}$ computed by substitutions in any ordering. Then there exist a backward error $\delta A_{11}^{(0)}$ and a small constant ε_{TS}, $0 \le \varepsilon_{TS} \le r_B\mathbf{u}/(1-r_B\mathbf{u})$, such that $\tilde{A}_{11}^{(1)} = (A_{11}^{(0)} + \delta A_{11}^{(0)})(\tilde{U}^{(1,1)})^{-1}$, and*

$$|\delta A_{11}^{(0)}| \le \varepsilon_{TS}|A_{11}^{(0)}|\cdot|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,1)}|(I-\varepsilon_{TS}|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,1)}|)^{-1}. \tag{2.9}$$

The value of ε_{TS}, which is assumed to be less than one, can be as small as $\mathbf{u} + O(r_B\mathbf{u}^2)$ if one uses double precision accumulation of the dot product in a row-oriented triangular system solver.

Proof. There exist matrices $\delta\tilde{U}_k^{(1,1)}$, $1 \le k \le m$, such that (cf. [23, Theorem 8.5])

$$e_k^T\tilde{A}_{11}^{(1)}(\tilde{U}^{(1,1)}+\delta\tilde{U}_k^{(1,1)}) = e_k^T A_{11}^{(0)}, \quad |\delta\tilde{U}_k^{(1,1)}|_{ij} \le \varepsilon_{TS}|\tilde{U}^{(1,1)}|_{ij}, \quad 1 \le i \le j \le r_B. \tag{2.10}$$

The matrix $\delta A_{11}^{(0)}$ is defined by the relation $\tilde{A}_{11}^{(1)}\tilde{U}^{(1,1)} - A_{11}^{(0)} = \delta A_{11}^{(0)}$. Relation (2.9) is obtained using (2.10) and the fact that $I - \varepsilon_{TS}|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,1)}|$ is an M-matrix for $\varepsilon_{TS} < 1$. ∎

The bound (2.9) is invariant under row scaling of $\tilde{U}^{(1,1)}$. Furthermore, since (due to pivoting) $|\tilde{U}^{(1,1)}|_{ii} \ge |\tilde{U}^{(1,1)}|_{ij}$, $j > i$, it holds for all $j \ge i$ that $(|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,1)}|)_{ij} \le 2^{j-i}$; see [22, Lemma 3.1]. In practice, the matrix $|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,1)}|$ is of moderate size with norm bounded by a moderate polynomial of its dimension. A similar behavior has the matrix $|(\tilde{U}^{(1,1)})^{-1}|\cdot|\tilde{U}^{(1,2)}|$ which determines the size of the backward error in the computation of the matrix $A_{12}^{(1)}$.

PROPOSITION 2.3. *Let $\tilde{A}_{12}^{(1)} = \boldsymbol{fl}(A_{12}^{(0)} - \tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)})$ be the matrix computed in floating-point arithmetic. Then there exists a backward error $\delta A_{12}^{(0)}$ and a small*

constant $\boldsymbol{\varepsilon}_{MP}$ *such that* $0 \le \boldsymbol{\varepsilon}_{MP} \le r_B\mathbf{u}/(1-r_B\mathbf{u})$, $\tilde{A}_{12}^{(1)} = A_{12}^{(0)} + \delta A_{12}^{(0)} - \tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)}$ *and*

$$|\delta A_{12}^{(0)}| \le \mathbf{u}|A_{12}^{(0)}| + (\mathbf{u} + \boldsymbol{\varepsilon}_{MP} + \mathbf{u}\boldsymbol{\varepsilon}_{MP})|A_{11}^{(0)}| \cdot |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,2)}| + (\mathbf{u} + \boldsymbol{\varepsilon}_{MP} + \mathbf{u}\boldsymbol{\varepsilon}_{MP})|\delta A_{11}^{(0)}| \cdot |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,2)}|.$$

The value of $\boldsymbol{\varepsilon}_{MP}$ *is reduced to the order of* $\mathbf{u} + O(r_B\mathbf{u}^2)$ *if one uses double precision accumulation of the dot product.*

Proof. Since $\boldsymbol{fl}(\tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)}) = \tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)} + E_1$, $|E_1| \le \boldsymbol{\varepsilon}_{MP}|\tilde{A}_{11}^{(1)}| \cdot |\tilde{U}^{(1,2)}|$, we have

$$\boldsymbol{fl}(A_{12}^{(0)} - \boldsymbol{fl}(\tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)})) = A_{12}^{(0)} - \tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)} - E_1 + E_2,$$

where $|E_2| \le \mathbf{u}|A_{12}^{(0)} - \tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)} - E_1|$. Now define $\delta A_{12}^{(0)} = E_2 - E_1$ and apply Proposition 2.2 and the triangle inequality to estimate $|\tilde{A}_{11}^{(1)}| \cdot |\tilde{U}^{(1,2)}|$. ∎

From Propositions 2.2 and 2.3, it follows that the computed matrix $\tilde{A}^{(1)} = [\tilde{A}_{11}^{(1)}, \tilde{A}_{12}^{(1)}]$ can be obtained in exact computation using the matrices $[A_{11}^{(0)} + \delta A_{11}^{(0)}, A_{12}^{(0)} + \delta A_{12}^{(0)}]$ and $[\tilde{U}^{(1,1)}, \tilde{U}^{(1,2)}]$. Next, we show that a similar result holds for the computed matrices in relations (2.4) and (2.5).

THEOREM 2.2. *Let* $(\tilde{A}_{11,2}^{(2)}, \tilde{L})$ *be the computed approximation of the matrix pair* (2.5) *in Algorithm* 2.1 *and let* ΔB *be as in relation* (2.8). *Then there exist a backward error* ΔA *and moderate polynomials* $\wp_1(m,n,p)$, $\wp_2(m,n,p)$ *such that* $(\tilde{A}_{11,2}^{(2)}, \tilde{L})$ *can be computed in an exact computation (similar to Algorithm* 2.1*) with the input pair* $(A + \Delta A, B + \Delta B)$ *and such that, for all* i,

$$\frac{\|\Delta A e_i\|_2}{\|Ae_i\|_2} \le \eta_i, \quad \textit{where} \tag{2.11}$$

$$\eta_i \le \wp_1\mathbf{u}\| \, |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,1)}|e_i \, \|_1 + \wp_2\mathbf{u}\| \, |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,2)}|e_i \, \|_1. \tag{2.12}$$

The matrix pair $(\tilde{A}_{11,2}^{(2)}, \tilde{L})$ *can also be computed in exact computation using* $(A^{(0)} + \Delta A^{(0)}, B^{(0)} + \delta B^{(0)})$, *where* $\delta B^{(0)}$ *is as in relation* (2.7) *and* $\|\Delta A^{(0)}e_i\|_2/\|A^{(0)}e_i\|_2 \le \eta_i$, $1 \le i \le n$.

Proof. It remains to analyze the QR factorization of $\tilde{A}_{12}^{(1)}$. For the computed upper trapezoidal matrix $[\tilde{A}_{12}^{(2)}, \tilde{\tilde{A}}_{12}^{(2)}]$, there exists a backward error $\delta\tilde{A}_{12}^{(1)}$ such that (cf. [23, Chapter 18])

$$\hat{Q}_{12}^{(2)} \begin{bmatrix} \tilde{A}_{12}^{(2)} & \tilde{\tilde{A}}_{12}^{(2)} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} = (\tilde{A}_{12}^{(1)} + \delta\tilde{A}_{12}^{(1)})\Pi_3, \qquad |\delta\tilde{A}_{12}^{(1)}| \le \boldsymbol{\varepsilon}_{QR}G|\tilde{A}_{12}^{(1)}|,$$

$$G = \frac{1}{m}ee^T \quad (e = (1,\ldots,1)^T), \tag{2.13}$$

where $\hat{Q}_{12}^{(2)}$ is a certain orthogonal matrix, close to the computed nearly orthogonal matrix $\tilde{Q}_{12}^{(2)}$. The quantity $\boldsymbol{\varepsilon}_{QR}$ is bounded by $\mathbf{u}$ times a modestly growing function of matrix dimensions. The matrix $\tilde{A}_{11}^{(2)}$ can be computed simultaneously with $\tilde{A}_{12}^{(2)}$ (using the same sequence of elementary orthogonal matrices) and there exists a backward perturbation $\delta\tilde{A}_{11}^{(1)}$ such that

$$\hat{Q}_{12}^{(2)}\tilde{A}_{11}^{(2)} = \tilde{A}_{11}^{(1)} + \delta\tilde{A}_{11}^{(1)}, \quad |\delta\tilde{A}_{11}^{(1)}| \le \boldsymbol{\varepsilon}_{QR}G|\tilde{A}_{11}^{(1)}|. \tag{2.14}$$

(A similar relation holds if we use the matrix-matrix product $(\tilde{Q}_{12}^{(2)})^T \tilde{A}_{11}^{(1)}$.) Partitioning $\tilde{A}_{11}^{(2)}$ yields

$$\begin{bmatrix} \tilde{A}_{11,1}^{(2)} & \tilde{A}_{12}^{(2)} & \tilde{\hat{A}}_{12}^{(2)} \\ \tilde{A}_{11,2}^{(2)} & \mathbf{O} & \mathbf{O} \end{bmatrix} = (\hat{Q}_{12}^{(2)})^T [\tilde{A}_{11}^{(1)} + \delta\tilde{A}_{11}^{(1)}, \tilde{A}_{12}^{(1)} + \delta\tilde{A}_{12}^{(1)}](I \oplus \Pi_3). \tag{2.15}$$

If we let the perturbations $\delta\tilde{A}_{11}^{(1)}$ and $\delta\tilde{A}_{12}^{(1)}$ propagate backward into $A_{11}^{(0)}$ and $A_{12}^{(0)}$, respectively, we obtain

$$\tilde{A}_{11}^{(1)} + \delta\tilde{A}_{11}^{(1)} = (A_{11}^{(0)} + \delta A_{11}^{(0)} + \delta\tilde{A}_{11}^{(1)}\tilde{U}^{(1,1)})(\tilde{U}^{(1,1)})^{-1}, \tag{2.16}$$

$$\tilde{A}_{12}^{(1)} + \delta\tilde{A}_{12}^{(1)} = A_{12}^{(0)} + \delta A_{12}^{(0)} + \delta\tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)} + \delta\tilde{A}_{12}^{(1)} - (\tilde{A}_{11}^{(1)} + \delta\tilde{A}_{11}^{(1)})\tilde{U}^{(1,2)}. \tag{2.17}$$

Using $\Delta A_{11}^{(0)} = \delta A_{11}^{(0)} + \delta\tilde{A}_{11}^{(1)}\tilde{U}^{(1,1)}$ and $\Delta A_{12}^{(0)} = \delta A_{12}^{(0)} + \delta\tilde{A}_{11}^{(1)}\tilde{U}^{(1,2)} + \delta\tilde{A}_{12}^{(1)}$, we obtain the backward error $\Delta A^{(0)} = [\Delta A_{11}^{(0)}, \Delta A_{12}^{(0)}]$ such that

$$\begin{bmatrix} \tilde{A}_{11,1}^{(2)} & \tilde{A}_{12}^{(2)} & \tilde{\hat{A}}_{12}^{(2)} \\ \tilde{A}_{11,2}^{(2)} & \mathbf{O} & \mathbf{O} \end{bmatrix} = (\hat{Q}_{12}^{(2)})^T (A^{(0)} + \Delta A^{(0)}) \begin{bmatrix} (\tilde{U}^{(1,1)})^{-1} & -(\tilde{U}^{(1,1)})^{-1}\tilde{U}^{(1,2)} \\ \mathbf{O} & I \end{bmatrix} (I \oplus \Pi_3).$$

(2.18)

The matrix $\Delta A^{(0)}$ can be estimated columnwise using Propositions 2.2 and 2.3, relations (2.13) and (2.14), and the fact that the columns of $A^{(0)}$ are, up to the order of $\mathbf{u}$, of unit norm. It holds that

$$\|\Delta A_{11}^{(0)} e_i\|_2 \le (\varepsilon_{TS} + \varepsilon_{QR}) \| \, |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,1)}| e_i \, \|_1 + O(\mathbf{u}^2), \tag{2.19}$$

$$\|\Delta A_{12}^{(0)} e_i\|_2 \le \mathbf{u} + \varepsilon_{QR} + (\varepsilon_{MP} + \varepsilon_{QR}) \| \, |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,2)}| e_i \, \|_1 \tag{2.20}$$

$$+ \varepsilon_{QR} \| \, |(\tilde{U}^{(1,1)})^{-1}| \cdot |\tilde{U}^{(1,1)}| e_i \, \|_1 + O(\mathbf{u}^2). \tag{2.21}$$

(A variant of Algorithm 2.1 (cf. Remark 2.1) can be used to also obtain a rowwise bound for $\Delta A^{(0)}$. We omit the details for the sake of brevity.) Finally, note that relation (2.18) and

$$\tilde{L}[I_{r_B}, \mathbf{O}, \mathbf{O}] = \tilde{L}[\tilde{U}^{(1,1)}, \tilde{U}^{(1,2)}] \begin{bmatrix} (\tilde{U}^{(1,1)})^{-1} & -(\tilde{U}^{(1,1)})^{-1}\tilde{U}^{(1,2)} \\ \mathbf{O} & I \end{bmatrix} (I \oplus \Pi_3) \tag{2.22}$$

imply that the computed approximation $(\tilde{A}_{11,2}^{(2)}, \tilde{L})$ of the pair (2.5) corresponds to an exact computation with the initial pair $(A^{(0)} + \Delta A^{(0)}, \tilde{L}\tilde{U})$. The latter is obtained using column scaling of the pair $(A + \Delta A, B + \Delta B)$, where $\Delta A = \delta A + \Delta A^{(0)}\tilde{D}_A$ and ΔB is as in relation (2.8). ∎

Hence, the computation of the pair $(\tilde{A}_{11,2}^{(2)}, \tilde{L})$ is backward stable, provided that the matrices $(\tilde{U}^{(1,1)})^{-1}\tilde{U}^{(1,1)}$ and $(\tilde{U}^{(1,1)})^{-1}\tilde{U}^{(1,2)}$ are moderate. The upper theoretical bounds for these matrices are functions of the dimensions, independent of the initial matrices. Although the theoretical bounds are exponential, these matrices are usually moderate in practice and their spectral norms are typically of the order of the dimensions (cf. Stewart [34], [33]).

Since the matrix $\tilde{L}$ is well-conditioned ($\tilde{L}$ is computed in Gaussian elimination with complete pivoting), the QR factorization with column pivoting of $\tilde{L}$ is accurate and the computed matrix $\tilde{R}_L \approx R_L$ is well-conditioned. In particular, the matrix

$|\tilde{R}_L^{-1}| \cdot |\tilde{R}_L|$ is moderate. This observation is important in the following backward stability analysis of Stage B of Algorithm 2.1.

PROPOSITION 2.4. *Let $\tilde{Q}_L$, $\tilde{J}$, $\tilde{V}$, $\tilde{\Sigma}$ be the computed matrices in Stage B of Algorithm* 2.1. *Then there exist backward errors $\Delta\tilde{A}_{11,2}^{(2)}$, $\Delta\tilde{L}$, a nonsingular matrix $\hat{Z}$, and exactly orthogonal matrices $\hat{V} \approx \tilde{V}$, $\hat{Q}_L \approx \tilde{Q}_L$ such that*

$$\hat{V}(\tilde{A}_{11,2}^{(2)} + \Delta\tilde{A}_{11,2}^{(2)})\hat{Z} = \tilde{\Sigma}, \quad \begin{bmatrix} \tilde{J} & \mathbf{O} \\ \mathbf{O} & I \end{bmatrix} \hat{Q}_L^T(\tilde{L} + \Delta\tilde{L})\hat{Z} = \begin{bmatrix} I \\ \mathbf{O} \end{bmatrix}. \tag{2.23}$$

It holds that $|\Delta\tilde{L}| \leq \mathbf{u}|\tilde{L}| + (1+\mathbf{u})\boldsymbol{\varepsilon}_{QR}G|\tilde{L}|$, where $\boldsymbol{\varepsilon}_{QR}$, G are defined analogously as in (2.13). *Further, there exists a moderate polynomial $\wp_3$ of the dimensions such that it holds, for all i, that $\|\Delta\tilde{A}_{11,2}^{(2)}e_i\|_2 \leq \wp_3\mathbf{u}\|\,|\tilde{R}_L^{-1}| \cdot |\tilde{R}_L|\,\|_1 \|\,\tilde{A}_{11,2}^{(2)}e_i\|_2 + O(\mathbf{u}^2)$. (Here $\|\cdot\|_1$ denotes the matrix norm induced by the vector ℓ_1 norm.)*

Proof. To prove this backward stability result, we follow [16]. For the sake of simplicity, we consider only the first-order bounds. Let $D_{11,2}$ denote the computed scaling matrix and let $(\tilde{A}_{11,2}^{(2)})_c$, $\tilde{L}_1$ be the computed scaled matrices in Step 0 of Stage B. This computation introduces only small elementwise rounding errors. For instance, it holds that $(\tilde{A}_{11,2}^{(2)})_c = (\tilde{A}_{11,2}^{(2)} + \delta\tilde{A}_{11,2}^{(2)})D_{11,2}^{-1}$, where $|\delta\tilde{A}_{11,2}^{(2)}| \leq \mathbf{u}|\tilde{A}_{11,2}^{(2)}|$, and that $\tilde{L}_1 = (\tilde{L} + \delta\tilde{L})D_{11,2}^{-1}$ with $|\delta\tilde{L}| \leq \mathbf{u}|\tilde{L}|$.

If $\tilde{R}_L$ is the computed triangular matrix in Step 1, then there exist a backward error $\delta\tilde{L}_1$ and an orthogonal matrix $\hat{Q}_L$ such that

$$\hat{Q}_L^T(\tilde{L}_1 + \delta\tilde{L}_1)\Pi_4 = \begin{bmatrix} \tilde{R}_L \\ \mathbf{O} \end{bmatrix}, \quad |\delta\tilde{L}_1| \leq \boldsymbol{\varepsilon}_{QR}G|\tilde{L}_1|.$$

The computed matrix $\tilde{F}$ satisfies (cf. Proposition 2.2)

$$\begin{aligned} \tilde{F} &= ((\tilde{A}_{11,2}^{(2)})_c + \delta(\tilde{A}_{11,2}^{(2)})_c)\Pi_4\tilde{R}_L^{-1}, \\ |\delta(\tilde{A}_{11,2}^{(2)})_c| &\leq \boldsymbol{\varepsilon}_{TS}|(\tilde{A}_{11,2}^{(2)})_c| \cdot \Pi_4 \cdot |\tilde{R}_L^{-1}| \cdot |\tilde{R}_L| \cdot \Pi_4^T + O(\mathbf{u}^2). \end{aligned}$$

Let $\tilde{J}$, $\tilde{V}$, $\tilde{\Sigma}$ be the matrices computed by the Jacobi SVD algorithm. Then there exist a backward error $\delta\tilde{F}$ and an orthogonal matrix $\hat{V} \approx \tilde{V}$ such that (cf. [16], [17])

$$\hat{V}(\tilde{F} + \delta\tilde{F}) = \tilde{\Sigma}\tilde{J}, \quad |\delta\tilde{F}| \leq \wp_J\mathbf{u}(G + \Gamma)|\tilde{F}| + O(\mathbf{u}^2)|\tilde{F}|,$$

where $\wp_J$ is a modest polynomial of the dimensions, G is as in relation (2.13) and $\max_{i,j}|\Gamma_{ij}| \leq 1$. Thus, if we define $\Delta\tilde{A}_{11,2}^{(2)} = \delta\tilde{A}_{11,2}^{(2)} + (\delta(\tilde{A}_{11,2}^{(2)})_c + \delta\tilde{F}\tilde{R}_L\Pi_4^T)D_{11,2}$ and $\Delta\tilde{L} = \delta\tilde{L} + \delta\tilde{L}_1D_{11,2}$, then relation (2.23) holds with $\hat{Z} = D_{11,2}^{-1}\Pi_4\tilde{R}_L^{-1}\tilde{J}^{-1}$. The bounds for $\Delta\tilde{A}_{11,2}^{(2)}$ and $\Delta\tilde{L}$ are straightforward. We finally note that relation (2.23) is not the QSVD because the matrix $\tilde{J}$ is only nearly orthogonal (up to $O(r_B\mathbf{u})$). The QSVD can be obtained by computing the QL (or the QR) factorization

$$\begin{bmatrix} \tilde{J} & \mathbf{O} \\ \mathbf{O} & I \end{bmatrix} \hat{Q}_L^T = Q'_L(I + E), \quad \|E\|_2 \leq O(r_B\mathbf{u})$$

and replacing $\tilde{L} + \Delta\tilde{L}$ with $(I+E)(\tilde{L} + \Delta\tilde{L}) = \tilde{L} + \Delta\tilde{L} + E\tilde{L} + E\Delta\tilde{L}$. ∎

It remains to analyze how the perturbations $\Delta\tilde{A}_{11,2}^{(2)}$ and $\Delta\tilde{L}$ propagate backward into the initial pair $(A^{(0)}, B^{(0)})$ (or (A, B)). It is easily seen that $\Delta\tilde{L}$ is obtained by

replacing $\tilde{L}\tilde{U}$ with $(I+\Delta\tilde{L}\tilde{L}^{\dagger})\tilde{L}\tilde{U} = (I+\Delta\tilde{L}\tilde{L}^{\dagger})(B^{(0)}+\delta B^{(0)})$. This is a multiplicative backward error in $\tilde{L}\tilde{U}$ (committed as additive error in $\tilde{L}$) and it introduces a relative uncertainty of order $\|\Delta\tilde{L}\tilde{L}^{\dagger}\|_2$ in the quotient singular values of $(A^{(0)}, B^{(0)})$ (see section 2.4).

Creating a backward history of $\Delta\tilde{A}^{(2)}_{11,2}$ is more technical. One can easily check that it suffices to update $\Delta A^{(0)}_{11}$ and $\Delta A^{(0)}_{12}$ by adding $\hat{Q}^{(2)}_{12}[\,{}^{\mathbf{O}}_{\Delta\tilde{A}^{(2)}_{11,2}}\,]\tilde{U}^{(1,1)}$ and $\hat{Q}^{(2)}_{12}[\,{}^{\mathbf{O}}_{\Delta\tilde{A}^{(2)}_{11,2}}\,]\tilde{U}^{(1,2)}$, respectively. Further, for a columnwise bounds for these new errors, one needs only to analyze $\Delta\tilde{A}^{(2)}_{11,2}\tilde{U}^{(1,1)}$ and $\Delta\tilde{A}^{(2)}_{11,2}\tilde{U}^{(1,2)}$. Let $\alpha = \max_i \|\Delta\tilde{A}^{(2)}_{11,2}e_i\|_2/\|\tilde{A}^{(2)}_{11,2}e_i\|_2$ and let $\tilde{U}^{(1,1)} = D_u\tilde{U}^{(1,1)}_d$, where D_u is diagonal scaling with the diagonal of $\tilde{U}^{(1,1)}$. From Proposition 2.4, it follows that α is a small multiple of $\mathbf{u}$. The matrix $\tilde{U}^{(1,1)}_d$ is well-conditioned, that is, its inverse is small in norm (cf. [34]). From relation (2.15), it follows that, for all i, $\|\Delta\tilde{A}^{(2)}_{11,2}e_i\|_2 \le \alpha\|\tilde{A}^{(1)}_{11} + \delta\tilde{A}^{(1)}_{11}\|_2$ and we obtain

$$\begin{aligned}
\|\Delta\tilde{A}^{(2)}_{11,2}\tilde{U}^{(1,1)}e_i\|_2 &\le \alpha(1+\varepsilon_{QR})\sum_{k=1}^{i}\|\tilde{A}^{(1)}_{11}e_k\|_2|\tilde{U}^{(1,1)}|_{ki}\\
&\le \alpha(1+\varepsilon_{QR})\sum_{k=1}^{i}\|(A^{(0)}_{11}+\delta A^{(0)}_{11})(\tilde{U}^{(1,1)}_d)^{-1}e_k\|_2|\tilde{U}^{(1,1)}_d|_{ki}\\
&\le \alpha\sum_{k=1}^{i}|\tilde{U}^{(1,1)}_d|_{ki}\|(\tilde{U}^{(1,1)}_d)^{-1}e_k\|_1 + O(\mathbf{u}^2) \quad (\text{with } |\tilde{U}^{(1,1)}_d|_{ki}\le 1).
\end{aligned}$$

A similar bound holds for the matrix $\Delta\tilde{A}^{(2)}_{11,2}\tilde{U}^{(1,2)}$, and we conclude that Stage A and Stage B in Algorithm 2.1 can be glued together while preserving backward stability. The upper bounds for the backward errors depend on the sizes of the matrices $(\tilde{U}^{(1,1)}_d)^{-1}$ and $|\tilde{R}_L^{-1}|\cdot|\tilde{R}_L|$. Due to pivoting, the norms of these matrices are always bounded by a function of the dimensions. In practice, the bounds are modest polynomials.

Remark 2.3. If $m \gg n$, then using the QR factorization $A = Q_A R_A$ and replacing A with the $n\times n$ matrix R_A increases the efficiency of Algorithm 2.1. It can be shown that such a modification does not essentially change the columnwise structure of the backward error in the matrix A.

Remark 2.4. A similar analysis can be done for the variant of Algorithm 2.1 with the LU factorization replaced with the QR factorization with complete pivoting (cf. Remark 2.1).

2.4. The accuracy of the computed singular values. In this section, we analyze the accuracy of the computed singular values. Since we are interested in cases where all finite quotient singular values can be computed with relative accuracy, we restrict our analysis in this section to the full rank case $r_A = n$, $r_B = \min\{p, n\}$.

For simplicity, we consider only the case $r_B = p$ and assume that the matrices are previously pre- and postmultiplied by suitable permutation matrices so that no row or column interchanges are necessary in the LU factorization of $B^{(0)}$. We also assume that r_A and r_B are well-determined in the presence of backward errors described in section 2.3.

It suffices to estimate the difference between the singular values of the pairs $(A^{(0)}, B^{(0)})$ and $(A^{(0)}+\Delta A^{(0)}, B^{(0)}+\delta B^{(0)})$. The desired bound is derived in a two-step scheme. First, we compare the quotient singular values $\sigma'_1 \ge \cdots \ge \sigma'_p$ of

$(A^{(0)}, B^{(0)})$ and the corresponding values $\sigma_1'' \geq \cdots \geq \sigma_p''$ of $(A^{(0)}, B^{(0)} + \delta B^{(0)})$. Then we compare the σ_i''s with the corresponding quotient singular values $\sigma_1''' \geq \cdots \geq \sigma_p'''$ of the pair $(A^{(0)} + \Delta A^{(0)}, B^{(0)} + \delta B^{(0)})$.

To derive a forward error bound for the quotient singular values of $(A^{(0)}, B^{(0)} + \delta B^{(0)})$, we use the following two observations: (i) If $A^{(0)} = QR$ is the QR factorization of A, then the nonzero quotient singular values of $(A^{(0)}, B^{(0)})$ are the inverses of the nonzero singular values of the matrix $S = BR^{-1}$; (ii) If $\tilde{S} = D_1^* S D_2$, where D_1, D_2 are nonsingular matrices, then the ordered singular values $\xi_1 \geq \xi_2 \geq \cdots$ and $\tilde{\xi}_1 \geq \tilde{\xi}_2 \geq \cdots$ of S and $\tilde{S}$, respectively, satisfy for all i (cf. [24, Theorem 5.2])

$$d(\tilde{\xi}_i, \xi_i) \equiv \frac{|\tilde{\xi}_i - \xi_i|}{\sqrt{\tilde{\xi}_i \xi_i}} \leq \mathcal{E}(D_1, D_2) \equiv \frac{1}{2} \frac{\|D_1^* - D_1^{-1}\|_2 + \|D_2^* - D_2^{-1}\|_2}{1 - (1/32)\|D_1^* - D_1^{-1}\|_2 \|D_2^* - D_2^{-1}\|_2}, \tag{2.24}$$

provided that $\|D_1^* - D_1^{-1}\|_2 \|D_2^* - D_2^{-1}\|_2 < 32$. (Note that $d(\tilde{\xi}_i, \xi_i) = d(\xi_i, \tilde{\xi}_i) = d(1/\tilde{\xi}_i, 1/\xi_i)$.)

PROPOSITION 2.5. *Let $B^{(0)} = LU$ and $B^{(0)} + \delta B^{(0)} = \tilde{L}\tilde{U}$ be the exact and the computed LU factorization of $B^{(0)}$, respectively, and let $A^{(0)} = QR$ be the QR factorization of $A^{(0)}$. Let E_L, E_U be such that $\tilde{L} = (I + E_L)L$, $\tilde{U} = U(I + E_U)$. Then, for $1 \leq i \leq p$,*

$$\begin{aligned} d(\sigma_i'', \sigma_i') &\leq \mathcal{E}(I + E_L^T, I + R E_U R^{-1}) \\ &\leq \|E_L\|_2 + \|R E_U R^{-1}\|_2 + O(\|E_L\|_2^2) + O(\|R E_U R^{-1}\|_2^2). \end{aligned} \tag{2.25}$$

Proof. Compare the singular values of BR^{-1} and $(I + E_L)BR^{-1}(I + RE_U R^{-1})$, using relation (2.24). ∎

Thus, the bound for $d(\sigma_i'', \sigma_i')$ depends on the accuracy of the LU factorization (with complete pivoting) of $B^{(0)}$ and on the value of $\kappa_2(A^{(0)})$. (Recall that the columns of $A^{(0)}$ are, up to a small $O(\mathbf{u})$ error, of unit norm.) To complete the bound in Proposition 2.5, we need an estimate of E_L and E_U.

THEOREM 2.3. *Let $B^{(0)} = LU$ be of full row rank and let $B^{(0)} = [B^{(0)}_{(1,1)}, B^{(0)}_{(1,2)}]$, $U = [U^{(1,1)}, U^{(1,2)}]$, where $B^{(0)}_{(1,1)} = LU^{(1,1)}$ is the leading $p \times p$ submatrix of $B^{(0)}$. Let $\delta B^{(0)} = [\delta B^{(0)}_{(1,1)}, \delta B^{(0)}_{(1,2)}]$, and let $B^{(0)} + \delta B^{(0)} = \tilde{L}\tilde{U}$ be the perturbed LU factorization. If the spectral radius of $E_B = |\tilde{L}^{-1} \delta B^{(0)}_{(1,1)} (\tilde{U}^{(1,1)})^{-1}|$ is less than 1, there exist a strictly lower triangular E_L and an upper triangular E_U such that*

$$\tilde{L} = (I + E_L)L, \quad \tilde{U} = U(I + E_U), \quad E_U = \begin{bmatrix} E_U^{(1,1)} & E_U^{(1,2)} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}. \tag{2.26}$$

Further, it holds that

$$|E_L| \leq |\tilde{L}| \cdot \mathbf{tril}\,((I - E_B)^{-1} E_B) \cdot |L^{-1}|, \tag{2.27}$$

$$|E_U^{(1,1)}| \leq |(U^{(1,1)})^{-1}| \cdot \overline{\mathbf{triu}}\,(E_B (I - E_B)^{-1}) \cdot |\tilde{U}^{(1,1)}|, \tag{2.28}$$

$$|E_U^{(1,2)}| \leq |(U^{(1,1)})^{-1} \tilde{L}^{-1} \delta B^{(1,2)}| + |(U^{(1,1)})^{-1} \tilde{L}^{-1} \delta L\, U^{(1,2)}|, \tag{2.29}$$

where $\mathbf{tril}\,(\cdot)$ and $\overline{\mathbf{triu}}\,(\cdot)$ denote, respectively, the strictly lower triangular part and the upper triangular part of a matrix.

Proof. The proof is based on the following two equations.

$$B^{(1,1)} + \delta B^{(1,1)} = \tilde{L}\tilde{U}^{(1,1)}, \;\; \delta U^{(1,2)} = \tilde{L}^{-1}(\delta B^{(1,2)} - \delta L\, U^{(1,2)}). \tag{2.30}$$

An application of [35, Theorem 5.1] to the first equation in (2.30) yields the estimates (2.27), (2.28) with $E_L \equiv \delta L\, L^{-1}$ and $E_U^{(1,1)} \equiv (U^{(1,1)})^{-1}\delta U^{(1,1)}$. To estimate E_U, we use the relation

$$[\tilde{U}^{(1,1)}, \tilde{U}^{(1,2)}] = [U^{(1,1)}, U^{(1,2)}]\begin{bmatrix} I + E_U^{(1,1)} & E_U^{(1,2)} \\ \mathbf{O} & I \end{bmatrix}, \quad E_U^{(1,2)} \equiv (U^{(1,1)})^{-1}\delta U^{(1,2)},$$

and the second equation in relation (2.30). ∎

In our algorithm, the error matrix $\delta B^{(0)}$ satisfies (cf. relation (2.7)) $B^{(0)} + \delta B^{(0)} = \tilde{L}\tilde{U}$, $|\delta B^{(0)}| \leq \varepsilon_{LU}|\tilde{L}| \cdot |\tilde{U}|$. This structure allows an improvement of the result of Theorem 2.3. Let $B^{(0)} = D_1 C D_2$, where D_1, D_2 are diagonal nonsingular matrices, and let δC be defined by the relation $B^{(0)} + \delta B^{(0)} = D_1(C + \delta C)D_2$. Then the LU factorizations $C = L_C U_C$ and $C + \delta C = \tilde{L}_C \tilde{U}_C$ are, respectively,

$$C = (D_1^{-1} L D_1)(D_1^{-1} U D_2^{-1}), \quad C + \delta C = (D_1^{-1}\tilde{L} D_1)(D_1^{-1}\tilde{U} D_2^{-1}),$$

and, as in relation (2.7), it holds that

$$\tilde{L}_C \tilde{U}_C = C + \delta C, \quad |\delta C| \leq \varepsilon_{LU}|\tilde{L}_C| \cdot |\tilde{U}_C|.$$

Hence, the analysis from Theorem 2.3 can be applied to C and $C + \delta C$, with E_C defined analogously as E_B. If $\tilde{L}_C = (I + E_{L_C})L_C$, $\tilde{U}_C = U_C(I + E_{U_C})$, then using, e.g., relation (2.27) (with L_C, $\tilde{L}_C$, E_C) and the Neumann expansion for $(I - E_C)^{-1}$, we obtain

$$|E_{L_C}| \leq \varepsilon_{LU}|\tilde{L}_C| \cdot \mathbf{tril}\,(|\tilde{L}_C^{-1}| \cdot |\tilde{L}_C| \cdot |\tilde{U}_C^{(1,1)}| \cdot |(\tilde{U}_C^{(1,1)})^{-1}|)|L_C^{-1}| + O(\mathbf{u}^2).$$

(Analogously for E_{U_C}.) Suppose that in the factorization $B^{(0)} = D_1 C D_2$ the diagonals of D_1 and D_2 are nonincreasingly ordered and that the matrices L_C and U_C are well-conditioned, so that E_{L_C} and E_{U_C} are small. From the relation

$$\tilde{L} = (I + D_1 E_{L_C} D_1^{-1})L, \quad \tilde{U} = U(I + D_2^{-1} E_{U_C} D_2) \tag{2.31}$$

it follows that $|E_L| \leq |E_{L_C}|$, $|E_U| \leq |E_{U_C}|$, and, thus, that E_L and E_U are also small. The role of pivoting is then to ensure that $B^{(0)}$ is of the form $D_1 C D_2$, where D_1 and D_2 have diagonals graded from large to small (or nearly graded) and C admits stable LU factorization with well-conditioned L_C and U_C.

Hence, the forward error in the quotient singular values depends on the accuracy of the computed LU factorization with complete pivoting of $B^{(0)}$. If the matrix $B^{(0)}$ has some additional properties (special zero or sign patterns, for example), then it is possible to construct algorithms for forward stable computation of L and U. In that case, much sharper error bounds hold, and the LU approach is numerically more attractive than the QR approach (cf. Remark 2.1). For more details, see [10], [9], [23, Chapter 9], [28].

PROPOSITION 2.6. *Let $A^{(0)} + \Delta A^{(0)} = \tilde{Q}\tilde{R}$ be the QR factorization of $A^{(0)} + \Delta A^{(0)}$ and let $R = (I + E_R)\tilde{R}$, where R is the triangular QR factor of $A^{(0)}$. Then, for $1 \leq i \leq p$,*

$$d(\sigma_i''', \sigma_i'') \leq \mathcal{E}(I, (I + E_R)^{-1}) \leq \|E_R\|_2 + O(\|E_R\|_2^2), \tag{2.32}$$

where for $\|\Delta A^{(0)} R^{-1}\|_F + \|\Delta A^{(0)}\tilde{R}^{-1}\|_F < 1$ and $\|\Delta A^{(0)}\|_F \|(A^{(0)})^\dagger\|_2 < 1$ it holds that

$$\|E_R\|_2 \leq \frac{2\|\Delta A^{(0)}\|_F \|(A^{(0)})^\dagger\|_2}{1 - \|\Delta A^{(0)}\|_F \|(A^{(0)})^\dagger\|_2}.$$

Proof. We consider the singular values of $(B^{(0)}+\delta B^{(0)})\tilde{R}^{-1}(I+E_R)^{-1}$ and apply relation (2.24). To estimate E_R, we use [35, Theorem 4.1]. ∎

Finally, combining Propositions 2.5 and 2.6, we obtain, for $1 \le i \le p$,

$$d(\sigma_i''', \sigma_i') \le d(\sigma_i'', \sigma_i')(1 + d(\sigma_i''', \sigma_i'')) + d(\sigma_i''', \sigma_i'')(1 + d(\sigma_i'', \sigma_i')) \tag{2.33}$$

$$\le \|E_L\|_2 + \kappa_2(A^{(0)})\|E_U\|_2 + \|E_R\|_2 + O(\mathbf{u}^2). \tag{2.34}$$

Thus, the accuracy of the values σ_i''' is determined by the sensitivity of the triangular factor of $A^{(0)}$ and by the accuracy of the LU factorization with complete pivoting of $B^{(0)}$. The errors from stage B are easily included into the forward error analysis by updating $\Delta A^{(0)}$ and replacing $\tilde{L}\tilde{U}$ with $(I + \Delta\tilde{L}\tilde{L}^{\dagger})\tilde{L}\tilde{U}$ (cf. Proposition 2.4 and the discussion after its proof).

3. Numerical experiments. Software implementation of Algorithm 2.1 efficiently uses the benefits of an optimized BLAS 3 [13] library. The QR factorization with column pivoting is a BLAS 3 version from [30] and the matrix operations in Stage A are implemented using the `XTRSM()` and `XGEMM()` subroutines. Currently, the LU factorization with complete pivoting and the Jacobi SVD are less optimized and this will be the subject of our future work. In the QR variant of Algorithm 2.1, the QR factorization with complete pivoting (used instead of the LU factorization) can be implemented as a combination of an initial row sorting (cf. Remark 2.1) and the BLAS 3 version of the QR factorization with column pivoting. Recent results [17] indicate that we can expect development of a high-performance implementation of the Jacobi SVD algorithm. Thus, the efficiency of our approach will improve with better implementations of the BLAS 3 and the Jacobi SVD algorithm. We also expect that the modular structure of our algorithm will make it possible to develop efficient and stable QSVD subroutines for modern parallel architectures.

The output of our algorithm is the QSVD of (A, B) in Van Loan's form. That is, we compute orthogonal matrices V and W and a nonsingular matrix X such that V^TAX and W^TBX are in diagonal form. An interesting feature of our algorithm is an option to return V and W in factored forms, using products of the Householder reflections. In that case, the information necessary to retrieve the reflections that define V and W is overwritten on the arrays A and B, respectively. Hence, we can compute and use V and W without additional square arrays. This saves $m^2 + p^2$ memory locations which is attractive if $m \gg n$ or $p \gg n$. However, in this paper we do not analyze the accuracy of the computed matrices V, W, X.

3.1. The results. We use several different types of test pairs. The first type is taken from [16] and contains full column rank matrices with controlled spectral condition number $\kappa_2((\cdot)_c)$ of the column equilibrated matrix. For the reader's convenience, we give a detailed description of the test pair generation.

Example 3.1. We generate random full column rank matrices A_c and B_c with nearly unit columns and with given $\kappa_2(A_c)$ and $\kappa_2(B_c)$ and apply scalings $A = A_c\Delta_A$, $B = B_c\Delta_B$, where Δ_A, Δ_B are random diagonal, nonsingular matrices with given spectral condition numbers.

Each 4-tuple $(\kappa_2(A_c), \kappa_2(\Delta_A), \kappa_2(B_c), \kappa_2(\Delta_B))$ is chosen from a four-dimensional mesh of condition numbers,

$$\mathcal{C} = \left\{ \kappa_{ijkl} = (10^i, 10^j, 10^k, 10^l) \ : \ (i,j,k,l) \in \mathcal{I} \times \mathcal{J} \times \mathcal{K} \times \mathcal{L} \subset \mathbf{N}^4 \right\},$$

where $\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}$ are determined at the very beginning of the test and kept fixed. For each fixed κ_{ijkl}, we generate A_c, Δ_A, B_c, Δ_B using different distributions of their

singular values. We use all admissible values of the parameter `MODE` in LAPACK's `DLATM1()` test procedure [11]. Hence, for each 4-tuple $(A_c, \Delta_A, B_c, \Delta_B)$ we can choose the singular value distribution modes from the set

$$\mathcal{M} = \{\mu_{i'j'k'l'} = (\mu_{i'}, \mu_{j'}, \mu_{k'}, \mu_{l'})\} \subseteq \{\pm 1, \ldots, \pm 6\}^4.$$

For each fixed $(\kappa_{ijkl}, \mu_{i'j'k'l'})$ we generate random pairs using different random number generators as specified by the parameter `IDIST` in `DLATM1()` procedure. Thus, our set of random number distributions is $\mathcal{D} \subseteq \{\mathcal{U}(-1,1), \mathcal{U}(0,1), \mathcal{N}(0,1)\}$, where $\mathcal{U}(-1,1)$, $\mathcal{U}(0,1)$ are uniform distributions on $(-1,1)$ and $(0,1)$, respectively, and $\mathcal{N}(0,1)$ is the normal distribution. For each fixed distribution $\chi \in \mathcal{D}$ we generate a set $\mathcal{E}^{\chi}_{\kappa_{ijkl}, \mu_{i'j'k'l'}}$ of different pairs, where the cardinality of $\mathcal{E}^{\chi}_{\kappa_{ijkl}, \mu_{i'j'k'l'}}$ is fixed at the very beginning of the test. This process makes a total of

$$\tau \equiv |\mathcal{I}|\,|\mathcal{J}|\,|\mathcal{K}|\,|\mathcal{L}|\,|\mathcal{M}|$$

different classes and $\tau \prod_{\chi \in \mathcal{D}} |\mathcal{E}^{\chi}_{\kappa_{ijkl}, \mu_{i'j'k'l'}}|$ different matrix pairs, generated in a sequence of nested loops controlled by the indices from $\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}, \mathcal{M}$.

Each test pair is generated in double precision and its quotient singular values are computed using a double precision procedure. The quotient singular values computed by the double precision procedure are then taken as reference for the single precision procedure run on the original pair rounded to single precision.

For a test pair (A, B) with well-conditioned A_c and B_c, the value of

$$\zeta_1(A,B) = \mathbf{u} \max\{\kappa_2(A_c), \kappa_2(B_c)\}$$

gives a good estimate of relative errors in the quotient singular values computed by the tangent algorithm from [16]. Our numerical experiments show that ζ_1 is equally good in Algorithm 2.1 as long $\kappa_2(B_c)$ is moderate.

Following the analysis in section 2.4, we compute another a priori relative error estimate in the following way. For computed triangular factors $\tilde{L}$, $\tilde{U}$ of $\Pi_1 B \Pi_2$, we compute the residual $\delta B = \tilde{L}\tilde{U} - \Pi_1 B \Pi_2$ and, in the case of square nonsingular $\tilde{L}$, $\tilde{U}$, the values

$$\begin{aligned}
\tilde{\boldsymbol{\varepsilon}}_L(B) &= \| \ |\tilde{L}|\, \mathbf{tril}(|\tilde{L}^{-1}\delta B \tilde{U}^{-1}|)\, |\tilde{L}^{-1}| \ \|_1, \\
\tilde{\boldsymbol{\varepsilon}}_U(B) &= \| \ |\tilde{U}^{-1}|\, \overline{\mathbf{triu}}(|\tilde{L}^{-1}\delta B \tilde{U}^{-1}|)\, |\tilde{U}| \ \|_1,
\end{aligned}$$

(cf. (2.27) and (2.28) in Theorem 2.3) and

$$\zeta_2(A,B) = \max\{\ \mathbf{u}\kappa_2(A_c), \tilde{\boldsymbol{\varepsilon}}_L(B), \tilde{\boldsymbol{\varepsilon}}_U(B)\ \}.$$

(If $\tilde{L}$ and $\tilde{U}$ are not square, the quantities $\tilde{\boldsymbol{\varepsilon}}_L(B)$ and $\tilde{\boldsymbol{\varepsilon}}_U(B)$ are defined using the leading square submatrices $\tilde{L}^{(1,1)}$, $\tilde{U}^{(1,1)}$. We use the matrix ℓ_1 norm $\|\cdot\|_1$ instead of $\|\cdot\|_2$ because $\|\cdot\|_1$ is easier to compute. Note that computation of ζ_1, ζ_2 is not error-free.) If $\zeta_1(A,B)$ and $\zeta_2(A,B)$ are realistic and sharp enough to be used in the practice, then the values of

$$\theta_1(A,B) = \frac{\max_{\sigma \in \sigma(A,B)} \frac{|\delta\sigma|}{\sigma}}{\zeta_1(A,B)}, \quad \theta_2(A,B) = \frac{\max_{\sigma \in \sigma(A,B)} \frac{|\delta\sigma|}{\sigma}}{\zeta_2(A,B)}$$

should be bounded by a moderate function of m, n, p and should not be much less than 1. (A value of $\theta_i(A,B)$ that is below 1 means that $\zeta_i(A,B)$ overestimates the actual relative error.)

We also use the following measure for the accuracy of our algorithm:

$$\varepsilon(i,k) = \max_{\kappa_2(A_c)=10^i,\kappa_2(B_c)=10^k} \max_{\sigma\in\sigma(A,B)} \frac{|\delta\sigma|}{\sigma}, \quad (i,k)\in\mathcal{I}\times\mathcal{K},$$

that is, we compute the maximal relative error over all quotient singular values of all matrix pairs with fixed $\kappa_2(A_c) = 10^i$, $\kappa_2(B_c) = 10^k$. Note that $-\log_{10}\varepsilon(i,k)$ gives an approximate minimal number of correct digits in the computed approximations of the quotient singular values of the test pairs with fixed "coordinates" $(i,k) \in \mathcal{I}\times\mathcal{K}$. According to the theory from [16], we can expect $-\log_{10}\varepsilon(i,k)$ to be roughly $7-\max\{i,k\}$, independent of the column scalings Δ_A, Δ_B.

To inspect the values of some relevant condition numbers, we also compute

$$\theta_3(B) = \frac{\kappa_2(B_c)}{\max\{\tilde{\varepsilon}_L(B),\tilde{\varepsilon}_U(B)\}/\mathbf{u}}, \quad \theta_4(B) = \| \, |U^{-1}|\cdot|U| \, \|_1.$$

The value of $\theta_3(B)$ is a comparison of $\kappa_2(B_c)$ and the condition number that determines the accuracy of the LU factorization of B. The quantity $\theta_4(B)$ is an important factor in the bound for the backward error in the columns of the matrix A and it should be of the order of the matrix dimensions.

In this example, the input data are $m = n = p = 100$ and

$$\begin{aligned}
&\mathcal{I} = \{2,\ldots,7\}, \;\; \mathcal{K} = \mathcal{I},\\
&\mathcal{J} = \{4,8,10,12,14,16\}, \;\; \mathcal{L} = \mathcal{J},\\
&\mathcal{M} = \{(5,4,-5,3),(3,-4,5,-3),(4,5,3,-4)\}, \;\; \mathcal{D} = \{\mathcal{U}(-1,1)\}.
\end{aligned}$$

For each node of $\mathcal{C}\times\mathcal{M}\times\mathcal{R}$ we performed one test on a randomly generated pair. As a reference, we use the double precision tangent algorithm from [16], because it computes the singular values to approximately $15-\max\{i,k\}$ correct decimal places. The measured values of θ_1 and θ_2 are bounded by 1.24 and 113, respectively, which means that the accumulated round-off enters the error linearly in matrix dimensions. Both ζ_1 and ζ_2 provide good relative error estimates. The number of correct digits shown in Figure 1 corresponds to the predicted theoretical behavior. The values of θ_3 are between 0.0042 and 539, which shows that, in this example, the condition numbers $\kappa_2(B_c)$ and $\max\{\tilde{\varepsilon}_L(B),\tilde{\varepsilon}_U(B)\}/\mathbf{u}$ differ by a factor on the order of the dimensions of the problem. This means that in this example both quantities are a good estimate of the condition number related to perturbations in the matrix B. Finally, the values of θ_4 are, as expected, at most of order of the dimension (below 99), which confirms that the relative backward error in the columns of the matrix A is small.

In the next example we test the stability of Algorithm 2.1 in the presence of heavy row and column weighting of the matrix B.

Example 3.2. In this example, the test matrix generator follows the scheme described in Example 3.1 and additionally scales the rows of each generated B by a diagonal ill-conditioned matrix D. That is, we first generate a random matrix B_S in the same way as we generated the matrix B_c in Example 3.1. Then we generate diagonal matrices Δ_B, $\Delta_B^{(1)}$ with the same spectral condition number and compute $B = \Delta_B^{(1)} B_S \Delta_B$. In this way we obtain a matrix B such that the matrix B_c, obtained by equilibrating the columns of B, has high $\kappa_2(B_c)$. However, since B_S is well-scaled, the value of $\kappa_2(B_S)$ controls the accuracy of the LU factorization of B. (If $B_S = L_S U_S$ is the LU factorization of B_S, then from the relations $B_S^\dagger = U_S^\dagger L_S^{-1}$, $L =$

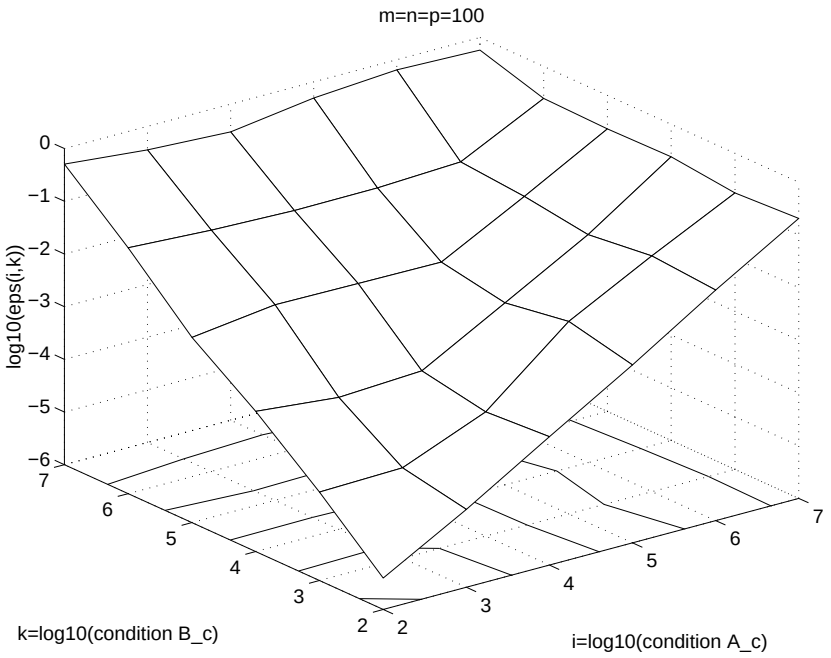


FIG. 1. *The values of* $\log_{10}\varepsilon(i,k)$, $(i,k)\in\mathcal{I}\times\mathcal{K}$ *in Example* 3.1. *Observe that* $-\log_{10}\varepsilon(i,k)$ *behaves like* $7-\max\{i,k\}$ *(roughly).*

$B_S U_S^\dagger$, $U_S^\dagger = B_S^\dagger L_S$, it follows that $\|B_S^\dagger\|_2 \le \|L_S^{-1}\|_2\|U_S^\dagger\|_2$, $\|L_S^{-1}\|_2 \le \|B_S^\dagger\|_2\|U_S\|_2$, $\|U_S^\dagger\|_2 \le \|B_S^\dagger\|_2\|L_S\|_2$.) For this reason, we use $\kappa_2(B_S)$ instead of $\kappa_2(B_c)$ in the definition of $\varepsilon(i,k)$.

Tests with LAPACK's `SGGSVD()` and with the tangent algorithm show that neither of those procedures is capable of achieving high relative accuracy with such a matrix. Since we use $\kappa_2(\Delta_B^{(1)})$ up to 10^{16}, we cannot use the double precision tangent algorithm as a reference. We use a double precision implementation of Algorithm 2.1 instead. (Numerical experiments show that the tangent algorithm improves if the QR factorization is computed with complete pivoting (cf. Remark 2.1).)

The input data in this example are the same as in Example 3.1. The test results are given in Figures 2 and 3. The values of θ_2 are at most of the order of 100, while the values of θ_1 are much smaller. This indicates a rather pessimistic estimate if we use ζ_1, and together with large values of θ_3, it shows that $\kappa_2(B_c)$ is not a good estimate of the condition number related to the perturbations of the matrix B. The number of correct digits, shown in Figure 3, confirms that $\kappa_2(B_S)$ performs much better.

Example 3.3. In this example we first generate an $m\times n$ matrix A and an $n\times p$ matrix C in the same way as A and B, respectively, in Example 3.1. Then we define $B = C^T$. In this way we control the size of $\kappa_2(B_r)$, where B_r is obtained from B by scaling its rows to have unit Euclidean norm. Note that in this example $\kappa_2(B_r)$ is used instead of $\kappa_2(B_c)$ in the definitions of $\zeta_1(A,B)$ and $\theta_3(B)$. Also note that the algorithm from [16] is not applicable because B does not have full column rank. We choose $m=300$, $n=150$, $p=50$, and $\mathcal{I}=\mathcal{K}=\{2,\dots,6\}$, $\mathcal{J}=\mathcal{L}=\{8,12,14,16\}$. For simplicity, we display only the values of $\varepsilon(i,k)$ (Figure 4), where in the definition of $\varepsilon(i,k)$ we replace $\kappa_2(B_c)$ with $\kappa_2(B_r)$ (cf. comments on $\kappa_2(B_S)$ in Example 3.2).

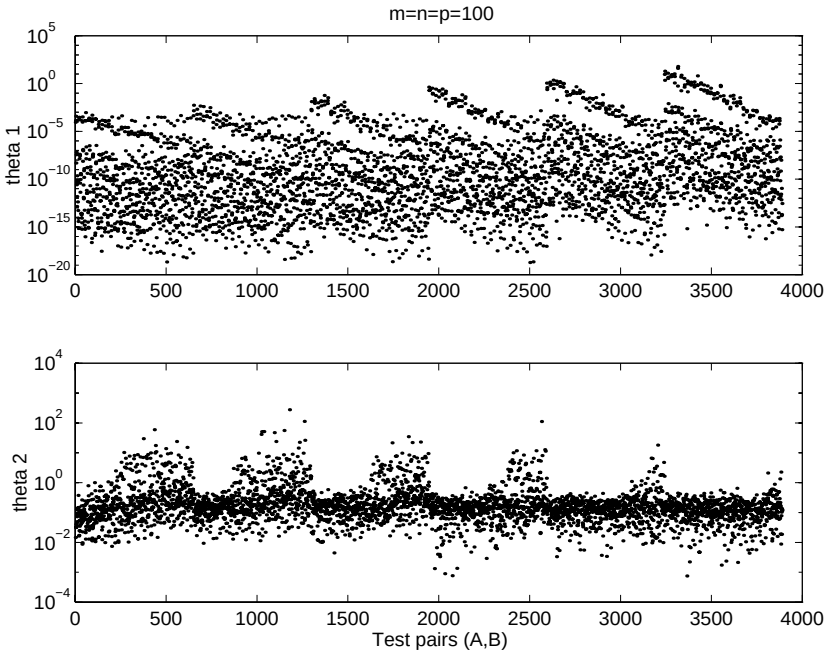


FIG. 2. *The values of* $\theta_1(\cdot,\cdot)$ *and* $\theta_2(\cdot,\cdot)$ *in Example* 3.2.

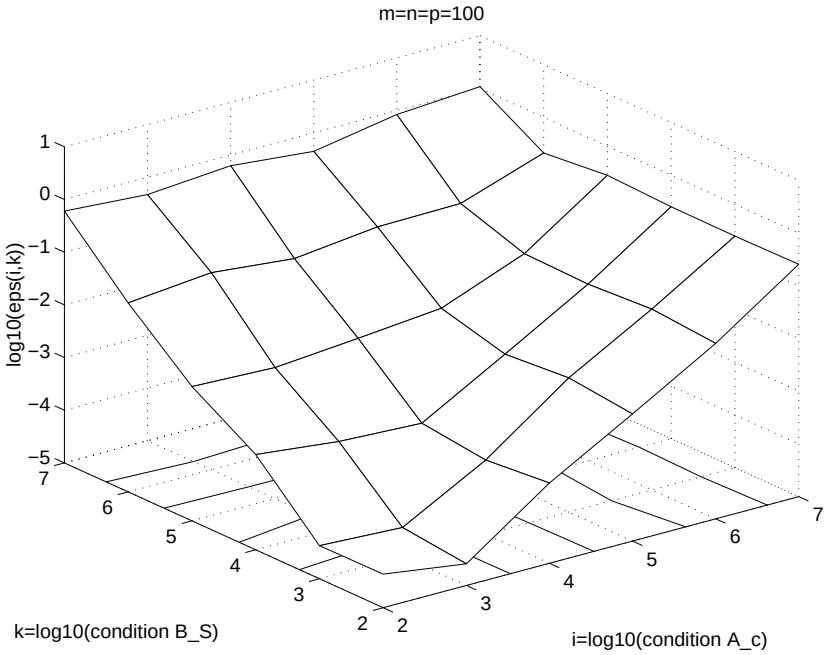


FIG. 3. *The values of* $\log_{10}\varepsilon(i,k)$, $(i,k)\in\mathcal{I}\times\mathcal{K}$ *in Example* 3.2.

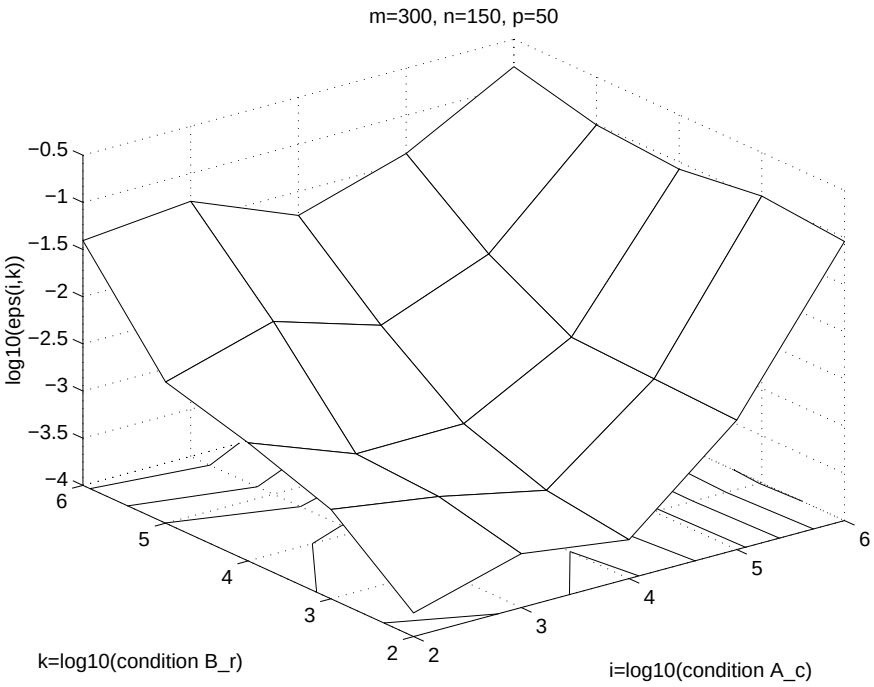


FIG. 4. *The values of* $\log_{10}\varepsilon(i,k)$, $(i,k)\in\mathcal{I}\times\mathcal{K}$ *in Example* 3.3.

We can see the minimal number of correct digits shows the same behavior as in the previous examples.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide,* 2nd ed., SIAM, Philadelphia, PA, 1995.

[2] Z. BAI AND J. DEMMEL, *Computing the Generalized Singular Value Decomposition*, LAPACK Working Note 46, Department of Computer Science, University of Tennessee, Knoxville, TN, 1992.

[3] Z. J. BAI AND H. Y. ZHA, *A new preprocessing algorithm for the computation of the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1007–1012.

[4] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[5] A. J. COX AND N. J. HIGHAM, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., Pitman Res. Notes Math. Ser. 380, Longman, Harlow, Essex, UK, 1998, pp. 57–73.

[6] B. L. R. DE MOOR AND G. H. GOLUB, *Generalized Singular Value Decompositions: A Proposal for a Standardized Nomenclature*, Technical report NA–89–04, Computer Science Department, Stanford University, Stanford, CA, 1989.

[7] A. DEICHMÖLLER, *Über die Berechnung verallgemeinerter singulärer Werte mittels Jacobi–änlicher Verfahren*, Ph.D. thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Hagen, Germany, 1991.

[8] A. DEICHMÖLLER AND K. VESELIĆ, *Two Algorithms for Computing the Symmetric Positive Definite Generalized Eigenvalue Problem and the Generalized Singular Values of Full Column Rank Matrices*, preprint, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Hagen, Germany, 1991.

[9] J. DEMMEL, *Accurate SVDs of Structured Matrices*, LAPACK Working Note 130, Technical re-

port UT–CS–97–375, Department of Computer Science, University of Tennessee, Knoxville, TN, 1997.

[10] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.

[11] J. Demmel and A. McKenney, *A Test Matrix Generation Suite*, LAPACK Working Note 9, Courant Institute, New York University, New York, 1989.

[12] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[13] J. J. Dongarra, J. J. D. Croz, I. Duff, and S. Hammarling, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[14] Z. Drmač, *Computing the Singular and the Generalized Singular Values*, Ph.D. thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Hagen, Germany, 1994.

[15] Z. Drmač, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comput., 18 (1997), pp. 1200–1222.

[16] Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal., 35 (1998), pp. 1804–1832.

[17] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[18] S. Falk and P. Langemeyer, *Das Jacobische Rotationsverfahren für reellsymmetrische Matrizenpaare* I, II, Elektronische Datenverarbeitung, 1960, pp. 30–43.

[19] G. H. Golub, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[21] M. R. Hestenes, *Inversion of matrices by biorthogonalization and related results*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 51–90.

[22] N. J. Higham, *The accuracy of solutions to triangular systems*, SIAM J. Numer. Anal., 26 (1989), pp. 1252–1265.

[23] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.

[24] R.-C. Li, *Relative Perturbation Theory:* (I) *Eigenvalue and Singular Value Variations*, Technical report, Mathematical Science Section, Oak Ridge National Laboratory, Oak Ridge, TN, 1996.

[25] C. C. Paige, *The general linear model and the generalized singular value decomposition*, Linear Algebra Appl., 70 (1985), pp. 269–284.

[26] C. C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

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

[28] J. M. Peña, *Backward stability of a pivoting strategy for sign–regular linear systems*, BIT, 37 (1997), pp. 910–924.

[29] M. J. D. Powell and J. K. Reid, *On applying Householder transformations to linear least squares problems*, in Information Processing 68, Proceedings of the International Federation of Information Processing Congress, Edinburgh, 1968, North–Holland, Amsterdam, 1969, pp. 122–126.

[30] G. Quintana-Ortí, X. Sun, and C. H. Bischof, *A BLAS-*3 *version of the QR factorization with column pivoting*, SIAM J. Sci. Comput., 19 (1998), pp. 1486–1494.

[31] G. W. Stewart, *Computing the CS decomposition of a partitioned orthonormal matrix*, Numer. Math., 40 (1982), pp. 297–306.

[32] G. W. Stewart, *A method for computing the generalized singular value decomposition*, in Matrix Pencils, Lecture Notes in Math. 973, Springer-Verlag, New York, 1983, pp. 207–220.

[33] G. W. Stewart, *On the perturbation of LU and Cholesky factors*, Technical report TR–3535, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1995.

[34] G. W. Stewart, *The Triangular Matrices of Gaussian Elimination and Related Decompositions*, Technical report TR–3533, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1995.

[35] J.-G. Sun, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.

[36] C. F. Van Loan, *Generalized Singular Values with Algorithms and Applications*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1973.

[37] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.
[38] C. F. Van Loan, *A generalized SVD analysis of some weighting methods for equality constrained least squares*, in Matrix Pencils, Lecture Notes in Math. 973, Springer-Verlag, New York, 1983, pp. 245–262.
[39] C. F. Van Loan, *Computing the CS and the generalized singular value decomposition*, Numer. Math., 46 (1985), pp. 479–491.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 874–894

POLYNOMIAL INSTANCES OF THE POSITIVE SEMIDEFINITE AND EUCLIDEAN DISTANCE MATRIX COMPLETION PROBLEMS*

MONIQUE LAURENT†

Abstract. Given an undirected graph $G = (V, E)$ with node set $V = [1, n]$, a subset $S \subseteq V$, and a rational vector $a \in \mathbf{Q}^{S \cup E}$, the positive semidefinite matrix completion problem consists of determining whether there exists a real symmetric $n \times n$ positive semidefinite matrix $X = (x_{ij})$ satisfying $x_{ii} = a_i$ $(i \in S)$ and $x_{ij} = a_{ij}$ $(ij \in E)$. Similarly, the Euclidean distance matrix completion problem asks for the existence of a Euclidean distance matrix completing a partially defined given matrix. It is not known whether these problems belong to NP. We show here that they can be solved in polynomial time when restricted to the graphs having a fixed minimum fill-in, the minimum fill-in of graph G being the minimum number of edges needed to be added to G in order to obtain a chordal graph. A simple combinatorial algorithm permits us to construct a completion in polynomial time in the chordal case. We also show that the completion problem is polynomially solvable for a class of graphs including wheels of fixed length (assuming all diagonal entries are specified). The running time of our algorithms is polynomially bounded in terms of n and the bitlength of the input a. We also observe that the matrix completion problem can be solved in polynomial time in the real number model for the class of graphs containing no homeomorph of K_4.

Key words. positive semidefinite matrix, Euclidean distance matrix, matrix completion, chordal graph, minimum fill-in, order of a graph, polynomial algorithm, bit model, real number model

AMS subject classifications. 05C50, 15A48, 15A57, 90C25

PII. S0895479899352689

1. Introduction.

1.1. The matrix completion problem. This paper is concerned with the completion problem for positive semidefinite and Euclidean distance matrices. The *positive semidefinite matrix completion problem* (P) is defined as follows:

Given a graph $G = (V, E)$, a subset $S \subseteq V$, and a rational vector $a \in \mathbf{Q}^{S \cup E}$, determine whether there exists a real matrix $X = (x_{ij})_{i,j \in V}$ satisfying

$$X \succeq 0 \ \text{ and } x_{ii} = a_i \ (i \in S),\ x_{ij} = a_{ij} \ (ij \in E). \tag{1.1}$$

(The notation $X \succeq 0$ means that X is a symmetric positive semidefinite matrix or, for short, a psd matrix.) In other words, problem (P) asks whether a partially specified matrix can be completed to a psd matrix, the terminology of graphs being used as a convenient tool for encoding the positions of the specified entries. When problem (P) has a positive answer, one says that a is *completable to a psd matrix*; a matrix X satisfying (1.1) is called a *psd completion* of a and a *positive definite (pd) completion* when X is positive definite. We let (P_s) denote problem (P) when $S = V$, i.e., when all diagonal entries are specified. If one looks for a pd completion, then one can assume without loss of generality that all diagonal entries are specified (cf. Lemma 2.5); this is, however, not obviously so if one looks for a psd completion (although this can be shown to be true when restricting the problem to the class of chordal graphs; cf. the proof of Theorem 3.5).

*Received by the editors February 26, 1999; accepted for publication (in revised form) by L. El Ghaoui June 5, 2000; published electronically December 20, 2000.
http://www.siam.org/journals/simax/22-3/35268.html
†CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands (monique@cwi.nl).

A matrix $Y = (y_{ij})_{i,j=1}^n$ is called a *Euclidean distance matrix* (a *distance matrix*, for short) if there exist vectors $u_1, \ldots, u_n \in \mathbf{R}^k$ (for some $k \geq 1$) such that

$$y_{ij} = \|u_i - u_j\|^2 \quad \text{for } i, j = 1, \ldots, n. \tag{1.2}$$

(Here, $\|u\|$ denotes the Euclidean norm of vector $u \in \mathbf{R}^k$.) A set of vectors u_i satisfying (1.2) is called a *realization* of Y. Note that all diagonal entries of a distance matrix are equal to zero. The *Euclidean distance matrix completion problem* (D) is defined as follows:

Given a graph $G = (V, E)$ and a rational vector $d \in \mathbf{Q}^E$, determine whether there exists a real matrix $Y = (y_{ij})_{i,j \in V}$ satisfying

$$Y \text{ is a distance matrix and } y_{ij} = d_{ij} (ij \in E). \tag{1.3}$$

Hence problem (D) asks whether a partially specified matrix can be completed to a distance matrix.

As will be recalled in section 2.3, psd matrices and distance matrices are closely related and, thus, their associated completion problems can often be treated in an analogous manner. These matrix completion problems have many applications, e.g., to multidimensional scaling problems in statistics (cf. [29]), to the molecule conformation problem in chemistry (cf. [11], [18]), and to moment problems in analysis (cf. [5]).

1.2. An excursion to semidefinite programming. The psd matrix completion problem is obviously an instance of the general *semidefinite programming feasibility problem* (F):

Given integral $n \times n$ symmetric matrices $Q_0, Q_1, \ldots, Q_m$, determine whether there exist real numbers $z_1, \ldots, z_m$ satisfying

$$Q_0 + z_1 Q_1 + \cdots + z_m Q_m \succeq 0. \tag{1.4}$$

The complexity status of problem (F) is a fundamental open question in the theory of semidefinite programming; this is true for both the Turing machine model and the real number model, the two most popular models of computation used in complexity theory. In particular, it is not known whether there exists an algorithm solving (F) whose running time is polynomial in the size L of the data, that is, the total space needed to store the entries of the matrices $Q_0, \ldots, Q_m$.

The *Turing machine model* (also called rational number model, or *bit model*; cf. [13]) works on rational numbers and, more precisely, on their binary representations; in particular, the running time of an elementary operation $(+, -, \times, \div)$ depends on the length of the binary representations of the rational numbers involved. Hence, the size L of the data of problem (F) in this model can be defined as $mn^2 L_0$, where L_0 is the maximum number of bits needed to encode an entry of a matrix Q_i. On the other hand, the *real number model* (introduced in [10]) works with real numbers and it assumes that exact real arithmetic can be performed; in particular, an elementary operation $(+, -, \times, \div)$ between any two real numbers takes unit time. Hence, the size L of the data of (F) in this model is equal to mn^2.

Semidefinite programming (SDP) deals with the decision problem (F) and its optimization version:

$$\begin{array}{ll} \max & c^T z \\ \text{subject to} & Q_0 + z_1 Q_1 + \cdots + z_m Q_m \succeq 0, \end{array} \tag{1.5}$$

where $c \in \mathbf{Q}^m$. SDP can be seen as a generalization of linear programming (LP), obtained by replacing the nonnegativity constraints of the vector variable in LP by the semidefiniteness of the matrix variable in SDP. Information about SDP can be found in the handbook [42]; cf. also the survey [40] and [3], [17] with an emphasis on applications to discrete optimization.

A standard result in LP is that every feasible linear system $Ax \leq b$ with rational coefficients has a solution whose size is polynomially bounded in terms of the size of A and b (cf. [38], Corollary 3.2b). This implies that the problem of testing the feasibility of an LP program belongs to NP in the bit model. (This fact is obvious for the real number model.) Moreover, any LP optimization problem can be solved in polynomial time in the bit model using the ellipsoid algorithm of Khachiyan [23] or the interior-point method of Karmarkar [22]; it is an open question whether LP can be solved in polynomial time in the real number model (cf. [43, p. 60]).

The feasibility problem (F) belongs to NP in the real number model (since one can test in polynomial time whether a matrix is psd, for instance, using Gaussian elimination; in fact, for a rational matrix the running time is polynomial in its bitlength (cf. [16, p. 295])). However, it is not known whether problem (F) belongs to NP in the bit model. Indeed, in contrast with LP, it is not true that if a solution exists then one exists which is rational and has a polynomially bounded size. Consider, for instance, the following matrix:

$$X := \begin{pmatrix} 2x & 2 & 0 & 0 \\ 2 & x & 0 & 0 \\ 0 & 0 & 2 & x \\ 0 & 0 & x & 1 \end{pmatrix}. \tag{1.6}$$

Then, $x = \sqrt{2}$ is the unique real for which $X \succeq 0$; hence, this is an instance where there is a real solution but no rational solution. Consider now the following matrix (taken from [35]):

$$X = \left(\begin{array}{c|cc|c|cc|c|cc} x_1 - 2 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \hline 0 & 1 & x_1 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & x_1 & x_2 & \dots & 0 & 0 & \dots & 0 & 0 \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ \hline 0 & 0 & 0 & \dots & 1 & x_i & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & x_i & x_{i+1} & \dots & 0 & 0 \\ \hline \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & x_{n-1} \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & x_{n-1} & x_n \end{array}\right).$$

Then, $X \succeq 0$ if and only if $x_1 \geq 2$ and $x_{i+1} \geq x_i^2$ for $i = 1, \dots, n-1$; hence, $x_n \geq 2^{2^{n-1}}$ and thus any rational solution has exponential bitlength. More examples of "ill-conditioned" semidefinite problems can be found in [35].

However, Ramana [35] has developed an exact duality theory for SDP which enables him to show the following results: Problem (F) belongs to NP $\cap$ co-NP in the real number model. In the bit model, (F) belongs to NP if and only if it belongs to co-NP; hence, (F) is not NP-complete nor co-NP complete unless NP = co-NP.

Algorithms have been found that permit us to solve the optimization problem (1.5) *approximatively* in polynomial time; they are based on the ellipsoid method (cf. [16]) and interior-point methods (cf. [32], [3]).

More precisely, set $K := \{z \in \mathbf{R}^m \mid Q_0 + \sum_{i=1}^m z_i Q_i \succeq 0\}$ and, given $\epsilon > 0$, set $S(K,\epsilon) := \{y \mid \exists z \in K \text{ with } \|z-y\| < \epsilon\}$ ("the points that are in the ϵ-neighborhood of K") and $S(K,-\epsilon) := \mathbf{R}^m \setminus S(\mathbf{R}^m \setminus K, \epsilon)$ ("the points that are at distance at least ϵ from the border of K"). Let L denote the maximum bit size of the entries of the matrices $Q_0, \ldots, Q_m$. Assume that we know a constant $R > 0$ such that either $K = \emptyset$ or $\exists\ z \in K$ with $\|z\| \le R$. Then, the ellipsoid based algorithm, given rational $\epsilon > 0$, either finds $y \in S(K,\epsilon)$ for which $c^T z \le c^T y + \epsilon$ for all $z \in S(K,-\epsilon)$, or asserts that $S(K,-\epsilon) = \emptyset$. Its running time is polynomial in n, m, L, and $\log \epsilon$ and this algorithm is polynomial in the bit model.

Assume that we know a constant $R > 0$ such that $\|z\| \le R$ for all $z \in K$ and a point $z^* \in K$ for which $Q_0 + \sum_{i=1}^m z_i^* Q_i \succ 0$ (z^* is "strictly feasible"). There is an interior-point algorithm which finds $y \in K$ strictly feasible such that $c^T y \ge \max_{z \in K} c^T z - \epsilon$ in time polynomial in n, m, L, $\log \epsilon$, $\log R$, and in the bitlength of z^*. Note, however, that no polynomial bound has been established for the bitlengths of the intermediate numbers occurring in the algorithm.

Khachiyan and Porkolab have shown that problem (F) and its analogue in rational numbers can be solved in polynomial time in the bit model for a fixed number m of variables.

THEOREM 1.1.

(i) [33] *Problem* (F) *can be solved in polynomial time for any fixed* m.

(ii) [24] *The following problem can be solved in polynomial time for any fixed* m: *Given* $n \times n$ *integral symmetric matrices* $Q_0, Q_1, \ldots, Q_m$, *find rational numbers* $z_1, \ldots, z_m$ *satisfying* (1.4) *or determine that no such numbers exist.* □

The result from Theorem 1.1 (ii) extends to the context of semidefinite programming the result of Lenstra [30] on the polynomial solvability of integer LP in fixed dimension.

1.3. Back to the matrix completion problem. Since the matrix completion problem is a special instance of SDP, it can be solved *approximatively* in polynomial time; specific interior-point algorithms for finding approximate psd and distance matrix completions have been developed, e.g., in [20], [11], [2], [31]. However, such algorithms are not guaranteed to find *exact* completions in polynomial time. This motivates our study in this paper of some classes of matrix completion problems that can be solved exactly in polynomial time.

As mentioned earlier, one of the difficulties in the complexity analysis of SDP arises from the fact that a rational SDP problem might have no rational solution. (Recall the example from (1.6).) This raises the following question in the context of matrix completion: *If a rational partial matrix has a psd completion, does a rational completion always exist?*

We do not know of a counterexample to this statement. On the other hand, we will show that the answer is positive, e.g., when the graph of specified entries is chordal or has minimum fill-in 1 (cf. Lemma 4.2). (Note that the answer is obviously positive if a pd completion exists.)

Motivated by the above discussion, let us define for each of the problems (P) and (D) its rational analogue (P^{Q}) and (D^{Q}). Problem (P^{Q}) is defined as follows:

Given a graph $G = (V,E)$, *a subset* $S \subseteq V$, *and a rational vector* $a \in \mathbf{Q}^{S \cup E}$, *find a rational matrix* X *satisfying* (1.1) *or determine that no such matrix exists.*

When $S = V$ (i.e., all diagonal entries are specified), we denote the problem as ($\mathrm{P}_s^{\mathrm{Q}}$). Problem ($\mathrm{D}^{\mathrm{Q}}$) is defined as follows:

Given a graph $G = (V, E)$ and a rational vector $d \in \mathbf{Q}^E$, find a rational matrix Y satisfying (1.3) *or determine that no such matrix exists.*

The complexity of the problems (P), (D), (P^Q), and (D^Q) is not known; in particular, it is not known whether they belong to NP in the bit model (they do trivially in the real number model). In this paper, we present some instances of graphs for which the completion problems can be solved in polynomial time. All our complexity results apply for the bit model (unless otherwise specified, as in section 5.3).

Recall that a graph is said to be *chordal* if it does not contain a circuit of length ≥ 4 as an induced subgraph. Then, the *minimum fill-in* of graph G is defined as the minimum number of edges needed to be added to G in order to obtain a chordal graph. Note that computing the minimum fill-in of a graph is an NP-hard problem [44]. The following is the main result of sections 3 and 4.

THEOREM 1.2. *For any integer $m \geq 0$, problems* (P), (P^Q), (D), *and* (D^Q) *can be solved in polynomial time (in the bit model) when restricted to the class of graphs whose minimum fill-in is equal to m.*

The essential ingredients in the proof of Theorem 1.2 are the subcase $m = 0$ (chordal case), Theorem 1.1, and the link (exposed in section 2.3) between psd matrices and distance matrices. In the chordal case, a simple combinatorial algorithm permits to solve the completion problem in polynomial time.

The psd matrix completion problem for chordal graphs has been extensively studied in the literature (cf. the survey of Johnson [19] for detailed references). In some sense, this problem has been solved by Grone et al. [15] who, building upon a result of Dym and Gohberg [12], have characterized when a vector a indexed by the nodes and edges of a chordal graph admits a psd completion; cf. Theorem 3.1. From this follows the polynomial time solvability of problem (P_s) for chordal graphs. In fact, the result from Theorem 3.1 is proved in [15] in a constructive manner and, thus, yields an algorithm permitting to solve problem (P_s^Q) for chordal graphs. This algorithm has a polynomial running time in the real number model; however, it has to be modified in order to achieve a polynomial running time in the bit model.

To summarize, the result from Theorem 1.2 also holds in the real number model for chordal graphs ($m = 0$); it would hold for all graphs having fixed minimum fill-in $m \geq 1$ if the result from Theorem 1.1 would remain valid in the real number model.[1]

We present in section 5.1 another class of graphs for which the matrix completion problem (P_s) can be solved in polynomial time (in the bit model). This class contains (generalized) circuits and wheels having a fixed length (and fatness); these graphs arise naturally when considering the polar approach to the psd matrix completion problem. Then, section 5.2 contains a brief description of this polar approach, together with some open questions and remarks. In the final section 5.3, we consider the matrix completion problem for the class of graphs containing no homeomorph of K_4. (It contains circuits.) Then a condition characterizing existence of a psd or distance matrix completion exists which permits us to obtain a simple combinatorial algorithm solving the existence and construction problems in polynomial time in the real number model.

2. Preliminaries. We recall here some basic facts about Schur complements and Euclidean distance matrices that will be needed in the paper, and we make a few observations about psd completions.

[1] L. Porkolab [34] claims to have a proof of this fact.

2.1. Schur complements. For a symmetric matrix M, set $\mathrm{In}(M) := (p, q, r)$, where p (resp., q, r) denotes the number of positive (resp., negative, zero) eigenvalues of M. When $M \succeq 0$, a maximal nonsingular principal submatrix of M is a nonsingular principal submatrix of M of largest possible order, thus equal to the rank of M.

LEMMA 2.1. *Let $M = \left(\begin{smallmatrix} A & B \\ B^T & C \end{smallmatrix}\right)$ be a symmetric matrix, where A is nonsingular. Then,*

$$\mathrm{In}(M) = \mathrm{In}(A) + \mathrm{In}(C - B^T A^{-1} B);$$

the matrix $C - B^T A^{-1} B$ is known as the Schur complement *of A in M. In particular, $M \succeq 0$ if and only if $A \succeq 0$ and $C - B^T A^{-1} B \succeq 0$. Moreover, if $M \succeq 0$ and if A is a maximal nonsingular principal submatrix of M, then $C = B^T A^{-1} B$.* □

As a direct application, we have the following results which will be used at several occasions in the paper.

LEMMA 2.2. *Let X be a symmetric matrix having the block decomposition*

$$X = \begin{array}{c} \\ \ell \\ n \\ m \end{array} \begin{array}{c} \begin{array}{ccc} \ell & n & m \end{array} \\ \begin{pmatrix} T & R^T & Z^T \\ R & A & S \\ Z & S^T & D \end{pmatrix} \end{array}, \tag{2.1}$$

where T, R, Z, A, S, D are rational matrices of suitable orders; all entries of X being specified except those of Z that have to be determined in order to obtain $X \succeq 0$. Assume that

$$X_1 := \begin{pmatrix} T & R^T \\ R & A \end{pmatrix} \succeq 0, \; X_2 := \begin{pmatrix} A & S \\ S^T & D \end{pmatrix} \succeq 0.$$

In the case when $n \geq 1$ and $A \neq 0$, let A_0 be a maximal nonsingular principal submatrix of A, and let

$$A = \begin{pmatrix} A_0 & B \\ B^T & C \end{pmatrix}, \; X = \begin{pmatrix} T & R_0^T & R_1^T & Z^T \\ R_0 & A_0 & B & S_0 \\ R_1 & B^T & C & S_1 \\ Z & S_0^T & S_1^T & D \end{pmatrix}$$

denote the corresponding block decompositions of A and X. Then, $X \succeq 0$ if we set

$$Z := S_0^T A_0^{-1} R_0 \tag{2.2}$$

when $n \geq 1$ and $A \neq 0$, and $Z := 0$ otherwise.

Proof. The result follows using Lemma 2.1 after noting that the Schur complement of A_0 in X is given by

$$\begin{pmatrix} T & R_1^T & Z^T \\ R_1 & C & S_1 \\ Z & S_1^T & D \end{pmatrix} - \begin{pmatrix} R_0^T \\ B^T \\ S_0^T \end{pmatrix} A_0^{-1} \begin{pmatrix} R_0 & B & S_0 \end{pmatrix}$$

$$= \begin{pmatrix} T - R_0^T A_0^{-1} R_0 & R_1^T - R_0^T A_0^{-1} B & Z^T - R_0^T A_0^{-1} S_0 \\ R_1 - B^T A_0^{-1} R_0 & C - B^T A_0^{-1} B & S_1 - B^T A_0^{-1} S_0 \\ Z - S_0^T A_0^{-1} R_0 & S_1^T - S_0^T A_0^{-1} B & D - S_0^T A_0^{-1} S_0 \end{pmatrix}$$

$$= \begin{pmatrix} T - R_0^T A_0^{-1} R_0 & 0 & Z^T - R_0^T A_0^{-1} S_0 \\ 0 & 0 & 0 \\ Z - S_0^T A_0^{-1} R_0 & 0 & D - S_0^T A_0^{-1} S_0 \end{pmatrix}.$$

Indeed, the Schur complement $C - B^T A_0^{-1} B$ of A_0 in A is equal to 0 since $A \succeq 0$ and A_0 is a maximal nonsingular principal submatrix of A; as $X_1, X_2 \succeq 0$ this implies that $R_1 - B^T A_0^{-1} R_0 = S_1 - B^T A_0^{-1} S_0 = 0$. □

LEMMA 2.3. *Let X be a symmetric matrix of the form*

$$X = \begin{pmatrix} T & R^T \\ R & A \end{pmatrix},$$

where $A \succeq 0$ and T is a symmetric matrix of order ℓ whose diagonal entries are all equal to some scalar N. Let A_0 be a maximal nonsingular principal submatrix of A and let

$$A = \begin{pmatrix} A_0 & B \\ B^T & B^T A_0^{-1} B \end{pmatrix}, \; X = \begin{pmatrix} T & R_0^T & R_1^T \\ R_0 & A_0 & B \\ R_1 & B^T & B^T A_0^{-1} B \end{pmatrix}$$

denote the corresponding block decompositions of A and X. Then, $X \succeq 0$ if and only if (i) $R_1 = B^T A_0^{-1} R_0$ *and* (ii) $T - R_0^T A_0^{-1} R_0 \succeq 0$. *In particular, X is pd if and only if A and $T - R^T A^{-1} R$ are pd. Moreover, $T - R_0^T A_0^{-1} R_0$ is psd for N large enough (namely, for N greater or equal to the largest eigenvalue of $R_0^T A_0^{-1} R_0 - T_0$, where T_0 has zero diagonal entries and as off-diagonal entries those of T).*

2.2. Some observations about psd completions. Given a graph $G = (V, E)$, a subset $S \subseteq V$, a vector $a \in \mathbf{Q}^{S \cup E}$, and a scalar $N > 0$, let $a^N \in \mathbf{Q}^{V \cup E}$ denote the extension of a obtained by setting $a_i := N$ for all $i \in V \setminus S$.

LEMMA 2.4. *a is completable to a psd matrix if and only if a^N is completable to a psd matrix for some $N > 0$ (and then for all $N' \geq N$).* □

Therefore, if one can "guess" a value N to assign to the unspecified diagonal entries, then one can reduce the problem to the case when all diagonal entries are specified. This can be done when the graph G of specified off-diagonal entries is chordal as we see later or if we look for a pd completion as the next result shows.

LEMMA 2.5. *Given $a \in \mathbf{Q}^{S \cup E}$, let $b := (a_i (i \in S), a_{ij} (ij \in E, \; i, j \in S))$ denote its restriction to the subgraph induced by S. Then, a has a pd completion if and only if b has a pd completion.*

Proof. Apply Lemma 2.3. □

This result does not extend to psd completions (which contradicts a claim from [15] (psd case in Proposition 1)). Indeed, the partial matrix

$$\begin{pmatrix} ? & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

has no psd completion while its lower principal 2×2 submatrix is psd.

A final observation is that if a partial matrix contains a fully specified row, then the completion problem can be reduced to considering a matrix of smaller order. Indeed, suppose that $A = (a_{ij})$ is a partial symmetric matrix whose first row is fully specified. If $a_{11} < 0$, then A is not completable. If $a_{11} = 0$, then A is completable if and only if its first row is identically zero and its lower principal submatrix of order $n - 1$ is completable. If $a_{11} > 0$ then one can reduce to a problem of order $n - 1$ by considering the Schur complement of a_{11} in A.

2.3. Euclidean distance matrices. The following connection (2.4) between psd and distance matrices has been established by Schoenberg [37]. Let $Y = (y_{ij})_{i,j \in V}$ be a square symmetric matrix with zeros on its main diagonal and whose rows and columns are indexed by a set V, and let i_0 be a given element of V. Then, $\varphi_{i_0}(Y)$ denotes the square symmetric matrix $X = (x_{ij})_{i,j \in V \setminus \{i_0\}}$ whose rows and columns are indexed by set $V \setminus \{i_0\}$ and whose entries are given by

$$x_{ij} = \frac{1}{2}(y_{i_0 i} + y_{i_0 j} - y_{ij}) \quad \text{for } i, j \in V \setminus \{i_0\}. \tag{2.3}$$

Then,

$$Y \text{ is a distance matrix } \iff \varphi_{i_0}(Y) \succeq 0. \tag{2.4}$$

(Indeed, a set of vectors u_i $(i \in V)$ forms a realization of the matrix Y if and only if $\varphi_{i_0}(Y)$ is the Gram matrix of the vectors $u_i - u_{i_0}$ $(i \in V \setminus \{i_0\})$, which means that its (i,j)th entry is equal to $(u_i - u_{i_0})^T(u_j - u_{i_0})$.) Thus, φ_{i_0} establishes a linear bijection between the set of distance matrices of order $|V|$ and the set of psd matrices of order $|V|-1$. Relation (2.4) has a direct consequence for the corresponding matrix completion problems. Let $G = (V, E)$ be a graph and assume that $i_0 \in V$ is a universal node, i.e., that i_0 is adjacent to all other nodes of G. Then, an algorithm permitting to solve the psd matrix completion problem for graph $G \backslash i_0$ can be used for solving the distance matrix completion problem for graph G and vice versa. Indeed,

$$\begin{aligned} &Y \text{ is a distance matrix completion of } d \in \mathbf{R}^E \\ &\iff \varphi_{i_0}(Y) \text{ is a psd completion of } \varphi_{i_0}(d). \end{aligned} \tag{2.5}$$

(For the definition of $\varphi_{i_0}(d)$, use (2.3) restricted to the pairs ij with $i, j \in V \setminus \{i_0\}$, $i = j$, or $i \neq j$ with ij edge of G.) For more information about connections between the two problems, see [21], [27].

3. The matrix completion problem for chordal graphs. We consider here the matrix completion problems for chordal graphs. First, we recall results from [15] and [4] yielding a good characterization for the existence of a completion; then, we see how they can be used for constructing a completion in polynomial time.

3.1. Characterizing existence of a completion. Let $G = (V, E)$ be a graph and let $a \in \mathbf{Q}^{V \cup E}$ be a vector; in the distance matrix case, the entries of a indexed by V (corresponding to the diagonal entries of a matrix completion) are assumed to be equal to zero. If $K \subseteq V$ is a clique in G (i.e., any two distinct nodes in K are joined by an edge in G), the entries a_{ij} of vector a are well-defined for all nodes $i, j \in K$; then, we let $a(K)$ denote the $|K| \times |K|$ symmetric matrix whose rows and columns are indexed by K and with ijth entry a_{ij} for $i, j \in K$. Obviously, if a is completable to a psd matrix, then a satisfies

$$a(K) \succeq 0 \quad \text{for every maximal clique } K \text{ in } G. \tag{3.1}$$

Similarly, if a is completable to a distance matrix, then a satisfies

$$a(K) \text{ is a distance matrix for every maximal clique } K \text{ in } G. \tag{3.2}$$

The conditions (3.1) and (3.2) are not sufficient in general for ensuring the existence of a completion. For instance, if $G = (V, E)$ is a circuit and $a \in \mathbf{Q}^{V \cup E}$ has all its

entries equal to 1 except one entry on an edge equal to -1, then a satisfies (3.1) but a is not completable to a psd matrix. However, if G is a chordal graph, then (3.1) and (3.2) suffice for ensuring the existence of a completion.

THEOREM 3.1. *Let $G = (V, E)$ be a chordal graph and let $a \in \mathbf{R}^{V \cup E}$. If a satisfies* (3.1)*, then a is completable to a psd matrix* [15]*; if a satisfies* (3.2)*, then a is completable to a distance matrix* [4]*; moreover, if a is rational valued, then a admits a rational completion.* □

As the maximal cliques in a chordal graph can be enumerated in polynomial time [39] (cf. below) and as one can check positive semidefiniteness of a rational matrix in polynomial time (cf. [16, p. 295]), one can verify whether (3.1) holds in polynomial time when G is chordal; in view of (2.4), one can also verify whether (3.2) holds in polynomial time when G is chordal. This implies the next theorem.

THEOREM 3.2. *Problems* (P_s) *and* (D) *can be solved in polynomial time for chordal graphs.* □

The proof given in [15], [4] for Theorem 3.1 is constructive; thus, it provides an algorithm for constructing a completion and, as we see below, a variant of it can be shown to have a polynomial running time. The proof is based on the following properties of chordal graphs. Let $G = (V, E)$ be a graph.

Then, G is chordal if and only if it has a perfect elimination ordering; moreover, such an ordering can be found in polynomial time [36]. An ordering $v_1, \ldots, v_n$ of the nodes of a graph $G = (V, E)$ is called a *perfect elimination ordering* if, for every $j = 1, \ldots, n-1$, the set of nodes v_k with $k > j$ that are adjacent to v_j induces a clique in G. For $j = 1, \ldots, n-1$, let K_j denote the clique consisting of node v_j together with the nodes v_k ($k > j$) that are adjacent to v_j; then the cliques $K_1, \ldots, K_{n-1}$ comprise all maximal cliques of a chordal graph G.

If G is chordal and not a clique, then one can find (in polynomial time) an edge $e \notin E$ for which the graph $H := G+e$ (obtained by adding e to G) is chordal. (Indeed, let i be the largest index in $[1, n]$ for which there exists $j > i$ such that v_i and v_j are not adjacent in G; then we can choose for e the pair ij as $v_1, \ldots, v_n$ remains a perfect elimination ordering for H.)

If G is chordal then, for any $e \notin E$, there exists a unique maximal clique in $G + e$ containing edge e [15] (easy to check).

Therefore, if G is complete and not a clique, we can order the missing edges in G as $e_1, \ldots, e_p$ in such a way that the graph $G_q := (V, E \cup \{e_1, \ldots, e_q\})$ is chordal for every $q = 1, \ldots, p$. For $q = 1, \ldots, p$, let K_q be the unique maximal clique in G_q containing edge e_q. Given $a \in \mathbf{Q}^{V \cup E}$ satisfying (3.1), set $G_0 := G$ and $x_0 := a$. We execute the following step for $q = 1, \ldots, p$.

Find $z_q \in \mathbf{Q}$ for which the vector $x_q := (x_{q-1}, z_q)$ of $\mathbf{Q}^{V \cup E(G_q)}$ satisfies

$$x_q(K_q) \succeq 0. \tag{3.3}$$

This can be done in view of Lemma 2.2 (case $\ell = m = 1$) applied to the matrix $X := x_q(K_q)$ and one can choose for z_q the rational value given by (2.2). Then, the final vector $x_p = (a, z_1, \ldots, z_p)$ provides a rational psd completion of a. This shows Theorem 3.1 in the psd case (the Euclidean distance matrix case being similar).

As mentioned earlier, the preprocessing step (find the suitable ordering $e_1, \ldots, e_p$ of the missing edges and the cliques K_q) can be done in polynomial time. Then, one can construct the values $z_1, \ldots, z_p$ yielding a psd completion of a in $p \leq n^2$ steps. Therefore, the algorithm is polynomial in the real number model. In order to show polynomiality in the bit model, one has to verify that the encoding sizes

of $z_1, \ldots, z_p$ remain polynomially bounded in terms of n and the encoding size of a. This is, however, not clear. Indeed, both R_0 and S_0 in the definition of z_q via (2.2) may involve some previously defined z_h for $h < q$ (the same may hold for A_0); then, we have a quadratic dependence between z_q and the previously defined $z_1, \ldots, z_{q-1}$ which may cause a problem when trying to prove that the encoding size of z_q remains polynomially bounded. However, as we see below, the above algorithm can be modified to obtain a polynomial running time. The basic idea is that, instead of adding the missing edges one at a time, one adds them by "packets" consisting of edges sharing a common end node. Then, in view of Lemma 2.2, one can specify simultaneously all the entries on these edges, which permits to achieve a linear dependency among the z_q's.

3.2. Constructing a psd completion in polynomial time. Let $G = (V, E)$ be a chordal graph and let $1, \ldots, n$ denote a perfect elimination ordering of its nodes. For $i \in [1, n]$, set

$$J(i) := \{j \in [1, n] : \ j > i \ \text{ and } \ ij \notin E\}$$

and let $i_1 > \cdots > i_L$ denote the elements $i \in [1, n]$ for which $J(i) \neq \emptyset$. For $\ell = 1, \ldots, L$, set $F_\ell := \{i_\ell j \mid j \in J(i_\ell)\}$ and let G_ℓ denote the graph with node set V and edge set $E \cup F_1 \cup \cdots \cup F_\ell$. Hence, we have a sequence of graphs

$$G_0 := G \subseteq G_1 \subseteq \cdots \subseteq G_\ell \subseteq \cdots \subseteq G_L, \tag{3.4}$$

where each G_ℓ is chordal (since $1 \ldots n$ remains a perfect elimination ordering of its nodes) and G_L is the complete graph. We now show that G_ℓ has only one maximal clique which is not a clique in $G_{\ell-1}$.

LEMMA 3.3. *For $\ell = 1, \ldots, L$, there is a unique maximal clique K_ℓ in G_ℓ which is not a clique in $G_{\ell-1}$. Moreover, $J(i_\ell) \cup \{i_\ell\} \subseteq K_\ell$, the set $K_\ell \setminus \{i_\ell\}$ is a clique in $G_{\ell-1}$, and the set $K_\ell \setminus J(i_\ell)$ is a clique in G.*

Proof. Let K be a maximal clique in G_ℓ which is not a clique in $G_{\ell-1}$; then, $i_\ell \in K$ and $K \cap J(i_\ell) \neq \emptyset$; we first show that $J(i_\ell) \subseteq K$. For this, assume that $j, j' \in J(i_\ell)$ with $j \in K$ and $j' \notin K$. By maximality of K, there exists an element $i \in K$ such that i and j' are not adjacent in G_ℓ. Then, $i < i_\ell$ since the set $[i_\ell, n]$ is a clique in G_ℓ. Therefore, the pairs ij and ii_ℓ are edges of G_ℓ and, thus, of G. Since the ordering of the nodes is a perfect elimination ordering for G, this implies that i_ℓ and j must be adjacent in G, yielding a contradiction.

Suppose now that K, K' are two distinct maximal cliques in G_ℓ such that $i_\ell \in K \cap K'$ and $J(i_\ell) \subseteq K \cap K'$. Then, there exist nodes $i \in K \setminus K'$, $i' \in K' \setminus K$ that are not adjacent in G_ℓ. Given a node $j \in J(i_\ell)$, one can easily verify that (i, i_ℓ, i', j) is an induced circuit in $G_{\ell-1}$, which contradicts the fact that $G_{\ell-1}$ is chordal and, thus, shows unicity of the clique K_ℓ. It is obvious that $K_\ell \setminus \{i_\ell\}$ is a clique in $G_{\ell-1}$. We now verify that $K_\ell \setminus J(i_\ell)$ is a clique in G. For this, note first that i_ℓ is adjacent to every node of $K_\ell \setminus (J(i_\ell) \cup \{i_\ell\})$ in G_ℓ and, thus, in G. Suppose now that $x \neq y$ are two nodes in $K_\ell \setminus (J(i_\ell) \cup \{i_\ell\})$ that are not adjacent in G. Then, as xy is an edge of $G_{\ell-1}$, we have $x = i_h$, $y \in J(i_h)$ for some $h \leq \ell - 1$ and, thus, $i_\ell < x, y$. As i_ℓ is adjacent to both x and y in G this implies that x and y must be adjacent in G, yielding a contradiction. □

We now describe the modified algorithm. Let $G = (V, E)$ be a chordal graph and let $a \in \mathbf{Q}^{V \cup E}$ satisfying (3.1). Setting $x_0 := a$, we execute the following step for $\ell = 1, \ldots, L$.

Find $z_\ell \in \mathbf{Q}^{F_\ell}$ for which the vector $x_\ell := (x_{\ell-1}, z_\ell) \in \mathbf{Q}^{V \cup E(G_\ell)}$ satisfies

$$x_\ell(K_\ell) \succeq 0. \tag{3.5}$$

Then, the final vector $x_L = (a, z_1, \ldots, z_L)$ provides a rational psd completion of a. For instance, we can choose for z_ℓ the value given by relation (2.2), applying Lemma 2.2 to the matrix $X := x_\ell(K_\ell)$. (Indeed, in view of Lemma 3.3, $X_1 = a(K_\ell \setminus J(i_\ell)) \succeq 0$ and $X_2 = x_{\ell-1}(K_\ell \setminus \{i_\ell\})$; thus, $X_2 \succeq 0$ can be verified by induction.)

We verify that the encoding sizes of $z_1, \ldots, z_L$ are polynomially bounded in terms of n and the encoding size of a. For this, we note that $z_1, \ldots, z_L$ are determined by a recurrence of the form

$$z_\ell = S_\ell^T A_\ell^{-1} R_\ell \text{ for } \ell = 1, \ldots, L, \tag{3.6}$$

where R_ℓ, A_ℓ, S_ℓ are matrices of (appropriate) orders $\leq n$. A crucial observation is that all entries of R_ℓ and A_ℓ belong to the set, denoted as $\mathcal{A}$, of entries of a (as $K_\ell \setminus J(i_\ell)$ is a clique in G, by Lemma 3.3), while the entries of S_ℓ belong to the set $\mathcal{A} \cup \mathcal{Z}_{\ell-1}$, where $\mathcal{Z}_{\ell-1}$ denotes the set of entries of $(z_1, \ldots, z_{\ell-1})$.

For $r \in \mathbf{Q}$, let $\langle r \rangle$ denote the encoding size of r, i.e., the number of bits needed to encode r in binary notation and, for a vector $x = (x_1, \ldots, x_p) \in \mathbf{Q}^p$, set $s(x) := \max(\langle x_1 \rangle, \ldots, \langle x_p \rangle)$. One can verify that, for two vectors $x, y \in \mathbf{Q}^p$, $\langle x^T y \rangle \leq \langle n \rangle + s(x) + s(y)$. Let S_a denote the maximum encoding length of the entries of vector a and, for $\ell = 1, \ldots, L$, set $S_\ell := \max(\langle z \rangle \mid z \in \mathcal{Z}_\ell)$. We derive from (3.6) that

$$S_\ell \leq \langle n \rangle + s(A_\ell^{-1} R_\ell) + S_a + S_{\ell-1}$$

for all ℓ (setting $S_0 := 0$). This implies that

$$S_L \leq L(S_a + \langle n \rangle) + \sum_{\ell=1}^{L} s(A_\ell^{-1} R_\ell).$$

As $L \leq n$, we obtain that all encoding sizes of $z_1, \ldots, z_L$ are polynomially bounded in terms of n and the encoding size of a. (We also use here the fact that the entries of A_ℓ^{-1} are polynomially bounded in the input size; cf. [16, Chapter 1.3].) Thus, we have shown the following theorem.

THEOREM 3.4. *Problem* ($\mathrm{P}_s^{\mathrm{Q}}$) *can be solved in polynomial time for chordal graphs.* □

We finally indicate how to solve the general problem when some diagonal entries are unspecified.

THEOREM 3.5. *Problems* (P) *and* (P^{Q}) *can be solved in polynomial time for chordal graphs.*

Proof. Let $G = (V, E)$ be a chordal graph, let $S \subseteq V$, and let $a \in \mathbf{Q}^{S \cup E}$ satisfying $a(K) \succeq 0$ for each maximal clique $K \subseteq S$. (Else, we can conclude that a is not completable.) Following Lemma 2.4, we search for a scalar $N > 0$ such that a is completable if and only if its extension $a^N \in \mathbf{Q}^{V \cup E}$ (assigning value N to the unspecified diagonal entries) is completable or, equivalently, $a^N(K) \succeq 0$ for all maximal cliques K in G. Note that each matrix $a^N(K)$ has the same form as matrix X from Lemma 2.3. Therefore, such N exists if and only if the linear condition (i) from Lemma 2.3 holds for each clique K and an explicit value for N can be constructed as indicated in Lemma 2.3. Once N has been determined, we proceed with completing a^N by applying the algorithm presented above. □

To conclude note that the algorithm presented in this section outputs a pd completion if one exists.

3.3. Constructing a distance matrix completion. The distance matrix completion problem for chordal graphs can be solved in an analogous manner. Namely, let $G = (V, E)$ be a chordal graph, let

$$G_0 := G \subseteq \cdots \subseteq G_\ell \subseteq \cdots \subseteq G_L$$

be the sequence of chordal graphs from (3.4), let K_ℓ ($\ell = 1, \ldots, L$) be the cliques constructed in Lemma 3.3, and let $a \in \mathbf{Q}^E$ satisfying (3.2). Setting $a_0 := a$, we execute the following step for $\ell = 1, \ldots, L$:

Find $z_\ell \in \mathbf{Q}^{F_\ell}$ for which the vector $x_\ell := (a_{\ell-1}, z_\ell) \in \mathbf{Q}^{E(G_\ell)}$ satisfies

$$x_\ell(K_\ell) \text{ is a distance matrix.} \tag{3.7}$$

Then, the final vector $x_L = (a, z_1, \ldots, z_L)$ provides a distance matrix completion of a. The above step can be performed as follows. If $K_\ell = J(i_\ell) \cup \{i_\ell\}$, then we let z_ℓ be defined by $z_\ell(j) := x_{\ell-1}(j_0, j)$ for $j \in J(i_\ell)$, where j_0 is a given element of $J(i_\ell)$. Otherwise, let $j_0 \in K_\ell \setminus (J(i_\ell) \cup \{i_\ell\})$; then j_0 is a universal node in $G[K_\ell]$, the subgraph of G induced by K_ℓ. Therefore, in view of relation (2.5), we can find z_ℓ satisfying (3.7) by applying Lemma 2.2. The polynomial running time of the above algorithm follows from the polynomial running time of the corresponding algorithm in the psd case. Thus, we have shown the following theorem.

THEOREM 3.6. *Problem* ($\mathrm{D}^{\mathbf{Q}}$) *can be solved in polynomial time for chordal graphs.*

4. The matrix completion problem for graphs with fixed minimum fill-in. In this section we describe an algorithm permitting us to solve problems (P), ($\mathrm{P}^{\mathbf{Q}}$), (D), and ($\mathrm{D}^{\mathbf{Q}}$) in polynomial time for the graphs having minimum fill-in m, where $m \geq 1$ is a given integer. This algorithm is based on Theorems 1.1, 3.1, 3.2, 3.4, and 3.6.

Let $G = (V, E)$ be a graph with minimum fill-in m, let $S \subseteq V$ and let $a \in \mathbf{Q}^{S \cup E}$ be given. (Again we assume that $a_i = 0$ for $i \in V$ in the distance matrix case.) We first execute the following step.

Step 0. Find edges $e_1, \ldots, e_m \notin E$ for which the graph $H := (V, E \cup \{e_1, \ldots, e_m\})$ is chordal and find the maximal cliques $K_1, \ldots, K_p$ in H. (Such edges exist since G has minimum fill-in m and they can be found in polynomial time, simply by enumeration as m is fixed. The maximal cliques in H can also be enumerated in polynomial time since H is chordal and, moreover, $p \leq n$.)

Then, we perform step x in order to solve problem (x) for $x = \mathrm{P}$, $\mathrm{P}^{\mathbf{Q}}$, D, $\mathrm{D}^{\mathbf{Q}}$.

Step P. Determine whether there exist real numbers $z_1, \ldots, z_m, z_{m+1}$ for which the vector $x \in \mathbf{Q}^{V \cup E(H)}$ defined by $x_i := a_i$ ($i \in S$), $x_i := z_{m+1}$ ($i \in V \setminus S$), $x_{ij} = a_{ij}$ ($ij \in E$), and $x_{e_h} := z_h$ ($h = 1, \ldots, m$) satisfies

$$x(K_1) \succeq 0, \ldots, x(K_p) \succeq 0. \tag{4.1}$$

Step D. Determine whether there exist real numbers $z_1, \ldots, z_m$ for which the vector $x \in \mathbf{Q}^{E(H)}$ defined by $x_{ij} = a_{ij}$ ($ij \in E$), and $x_{e_h} := z_h$ ($h = 1, \ldots, m$) satisfies

$$x(K_1), \ldots, x(K_p) \text{ are distance matrices.} \tag{4.2}$$

Then, a has a completion if and only if the answer in Step P or D is positive.

Step $P^{\mathbf{Q}}$. Find rational numbers $z_1, \ldots, z_m, z_{m+1}$ for which (4.1) holds or determine that no such numbers exist; if they exist, find a rational psd completion of x.

Step $D^{\mathbf{Q}}$. Find rational numbers $z_1, \ldots, z_m$ for which (4.2) holds or determine that no such numbers exist; if they exist, find a rational distance matrix completion of x.

Steps P and $P^{\mathbf{Q}}$ can be executed in the following manner. Let M denote the block diagonal matrix with the p matrices $x(K_1), \ldots, x(K_p)$ as diagonal blocks (and zeros elsewhere). Hence, M has order $|K_1| + \cdots + |K_p| \leq n^2$ and (4.1) holds if and only if $M \succeq 0$. Clearly, the matrix M can be written under the form

$$M = Q_0 + z_1 Q_1 + \cdots + z_{m+1} Q_{m+1},$$

where $Q_1, \ldots, Q_{m+1}$ are symmetric matrices with (0,1)-entries and Q_0 is a symmetric matrix whose nonzero entries belong to the set of entries of a. Therefore, in view of Theorem 1.1, one can determine the existence of $z_1, \ldots, z_{m+1}$ satisfying (4.1) in polynomial time. Then, finding a rational psd completion of x in Step $P^{\mathbf{Q}}$ can be done in polynomial time in view of Theorem 3.4.

In the distance matrix case, we use the following construction for distance matrices. For $a = 1, \ldots, p$, let D_a be a square symmetric matrix whose rows and columns are indexed by set V_a and let i_a be a given element of V_a. We construct a new matrix D, denoted as $D_1 \oplus \cdots \oplus D_p$, whose rows and columns are indexed by set $V_1 \cup \cdots \cup V_p$ and whose entries are given by

$$D(i,j) = \begin{cases} D_a(i,j) & \text{if } i,j \in V_a,\ a \in [1,p], \\ D_a(i,i_a) + D_b(j,i_b) & \text{if } i \in V_a,\ j \in V_b,\ a \neq b \in [1,p]. \end{cases} \tag{4.3}$$

LEMMA 4.1. *$D_1 \oplus \cdots \oplus D_p$ is a distance matrix if and only if $D_1, \ldots, D_p$ are distance matrices.*

Proof. The "only if" part is obvious. Conversely, assume that $D_1, \ldots, D_p$ are distance matrices; we show that $D := D_1 \oplus \cdots \oplus D_p$ is a distance matrix. For $a \in [1,p]$, let $u_i^a \in \mathbf{R}^{n_a}$ $(i \in V_a)$ be vectors providing a realization of D_a; we can assume without loss of generality that $u_{i_a}^a = 0$. Then, we construct a sequence of vectors $w_i \in \mathbf{R}^{n_1 + \cdots + n_p}$ $(i \in \bigcup_{a=1}^p V_a)$ by setting $w_i := (0_{n_1}, \ldots, 0_{n_{a-1}}, u_i^a, 0_{n_{a+1}}, \ldots, 0_{n_p})$ for $i \in V_a$. (0_n denotes the zero vector in $\mathbf{R}^n$.) One can easily verify that the vectors w_i provide a realization of D. □

Steps D and $D^{\mathbf{Q}}$ can be performed as follows. Let $M := x(K_1) \oplus \cdots \oplus x(K_p)$ denote the matrix indexed by $K_1 \cup \cdots \cup K_p$ constructed as indicated in relation (4.3). Clearly, M can be written under the form

$$M = Q_0 + z_1 Q_1 + \cdots + z_m Q_m,$$

where $Q_1, \ldots, Q_m$ are symmetric matrices with entries in $\{0,1\}$ and Q_0 is a symmetric matrix whose nonzero entries are sums of at most two entries of a. Let i_0 be a given element of $K_1 \cup \cdots \cup K_p$. Then,

$$\varphi_{i_0}(M) = \varphi_{i_0}(Q_0) + z_1 \varphi_{i_0}(Q_1) + \cdots + z_m \varphi_{i_0}(Q_m).$$

Hence, (4.2) holds if and only if matrix M is a distance matrix (by Lemma 4.1) or, equivalently, if and only if $\varphi_{i_0}(M)$ is positive semidefinite (by relation (2.4)). Therefore, in view of Theorems 3.2 and 3.6, Steps D and $D^{\mathbf{Q}}$ can be executed in polynomial time. This completes the proof of Theorem 1.2.

LEMMA 4.2. *When the minimum fill-in m is equal to 1, existence of a completion implies existence of a rational one.*

Proof. To see it, suppose first that all diagonal entries are specified; then, Steps P and $P^{\mathbf{Q}}$ can be executed in an elementary manner. Indeed, each matrix $x(K_i)$ $(i = 1, \ldots, p)$ has at most one unspecified entry z_1. Hence, the set of scalars z_1 for which $x(K_i) \succeq 0$ is an interval of the form $I_i = [\beta_i - \sqrt{\alpha_i}, \beta_i + \sqrt{\alpha_i}]$ where $\alpha_i, \beta_i \in \mathbf{Q}$ (easy to see from Lemma 2.2). Therefore, (4.1) holds if and only if $z_1 \in \bigcap_{i=1}^{p} I_i = [u, v]$, where $u := \max_i(\beta_i - \sqrt{\alpha_i})$ and $v := \min_i(\beta_i + \sqrt{\alpha_i})$. Moreover, if there is a completion (i.e., if $u \leq v$), then one can find one with z_1 rational. This is obvious if $u < v$ and, if $u = v$, this follows from the fact (easy to verify) that

$$\beta - \sqrt{\alpha} = \beta' + \sqrt{\alpha'},\ \alpha, \alpha', \beta, \beta' \in \mathbf{Q} \Longrightarrow \sqrt{\alpha}, \sqrt{\alpha'} \in \mathbf{Q}.$$

Suppose now some diagonal entries are unspecified. If there is a completion with value z_2 at the unspecified diagonal entries, then we can assume that z_2 is rational (replacing if necessary z_2 by a larger rational number). Then, by the above discussion, the off-diagonal unspecified entry z_1 can also be chosen to be rational. □

5. Further results and open questions. We present in section 5.1 another class of graphs for which the completion problem can be solved in polynomial time (in the bit model). Then, we discuss in section 5.2 some open questions arising when considering a polar approach to the psd completion problem. Finally, we describe in section 5.3 a simple combinatorial algorithm permitting us to solve the completion problem in polynomial time (in the real number model) for the class of graphs containing no homeomorph of K_4.

5.1. Another class of polynomial instances. We present here another class of graphs for which the psd matrix completion problem (P_s) can be solved in polynomial time. Given two integers $p, q \geq 1$, let $\mathcal{G}_{p,q}$ be the class consisting of the graphs $G = (V, E)$ satisfying the following properties. There exist two disjoint subsets V_1, V_2 of V such that $\min(|V_1|, |V_2|) = p$, the set $F := \{ij \mid i \in V_1, j \in V_2\}$ is disjoint from E, the graph

$$H := (V, E \cup F)$$

is chordal, and H has q maximal cliques that are not cliques in G.

THEOREM 5.1. *Given integers $p, q \geq 1$, the psd completion problem (P_s) can be solved in polynomial time (in the bit model) over the class $\mathcal{G}_{p,q}$.*

Examples of graphs belonging to class $\mathcal{G}_{p,q}$ arise from circuits, wheels, and some generalizations. A *generalized circuit* of length n is defined in the following manner: its node set is $U_1 \cup \cdots \cup U_n$ with two nodes $u \in U_i$, $v \in U_j$ being adjacent if and only if $i = j$ or $j = i + 1$ (modulo n); a *generalized wheel* of length n is obtained by adding a set U_0 (the *center* of the wheel) of pairwise adjacent nodes to a generalized circuit of length n and making each node in U_0 adjacent to each node in $U_1 \cup \cdots \cup U_n$. Call a generalized circuit or wheel *p-fat* if $\min(|U_i| : i = 1, \ldots, n) = p$. Cf. Figure 5.1 for an example. Then, any p-fat generalized circuit or wheel of length $q + 2$ belongs to $\mathcal{G}_{p,q}$. We will see in section 5.2 that generalized circuits and wheels arise as basic objects when studying the matrix completion problem on graphs of small order.

The proof of Theorem 5.1 is based on the following result of Barvinok [8], which shows that one can test feasibility of a system of quadratic equations in polynomial time for any fixed number of equations.[2]

[2]In [8] Barvinok considers the homogeneous case, where each equation is of the form $f_i(x) = x^T A_i x = 0$ for some symmetric matrix A_i. However, the general nonhomogeneous case can be derived from it [9].

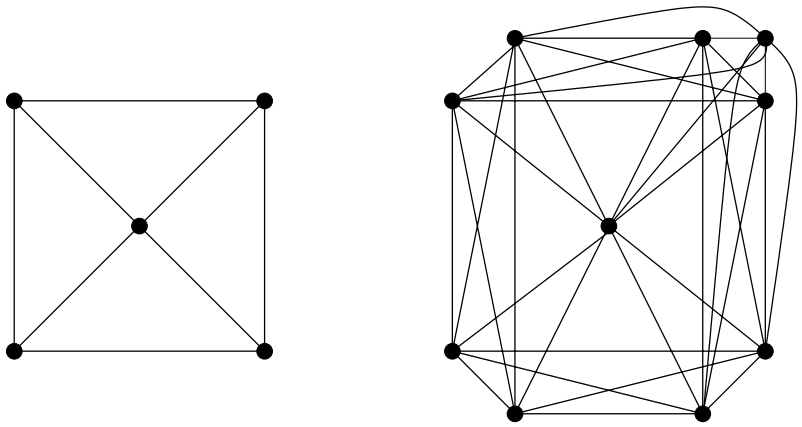


FIG. 5.1. (a) *The wheel of length* 4; (b) *a* 2*-fat generalized wheel of length* 4.

THEOREM 5.2. *For* $i = 1, \ldots, m$, *let* $f_i(x) = x^T A_i x + b_i^T x + c_i$ *be a quadratic polynomial in* $x \in \mathbf{R}^n$, *where* A_i *is an* $n \times n$ *symmetric matrix,* $b_i \in \mathbf{R}^n$, *and* $c_i \in \mathbf{R}$. *One can test feasibility of the system* $f_i(x) = 0$ *for* $i = 1, \ldots, m$ *in polynomial time (in the bit model) for any given* m. □

Proof of Theorem 5.1. Let $G = (V, E)$ be a graph in class $\mathcal{G}_{p,q}$ and let $a \in \mathbf{R}^{V \cup E}$ be given. We are also given the sets V_1 and V_2 for which, say, $p = |V_1| \leq |V_2|$ and adding to G all edges in $F := \{ij \mid i \in V_1, j \in V_2\}$ creates a chordal graph H. We show that deciding whether a can be completed to a psd matrix amounts to testing the feasibility of a system of m quadratic polynomials where m depends only on p and q. As H is chordal, a is completable to a psd matrix if and only if there exists a matrix Z of order $V_2 \times V_1$ for which $x := (a, Z) \in \mathbf{R}^{V \cup E \cup F}$ satisfies $x(K) \succeq 0$ for each maximal clique K in H. We assume that $x(K) = a(K) \succeq 0$ for each maximal clique K of H contained in G. (Else, we can conclude that a is not completable.) Consider now a maximal clique K of H which is not contained in G. Then, $x(K)$ has the form

$$x(K) = \begin{array}{c} \\ V_1 \cap K \\ V_0 \cap K \\ V_2 \cap K \end{array} \begin{array}{c} \begin{array}{ccc} V_1 \cap K & V_0 \cap K & V_2 \cap K \end{array} \\ \begin{pmatrix} T & R^T & Z_K^T \\ R & A & S \\ Z_K & S^T & D \end{pmatrix} \end{array},$$

setting $V_0 := V \setminus (V_1 \cup V_2)$ and $Z_K := Z[V_2 \cap K, V_1 \cap K]$, the submatrix of Z with row indices in $V_2 \cap K$ and column indices in $V_1 \cap K$. With the notation of Lemma 2.2, we obtain that $x(K) \succeq 0$ if and only if the following matrix

$$M_K := \begin{pmatrix} T - R_0^T A_0^{-1} R_0 & Z_K^T - R_0^T A_0^{-1} S_0 \\ Z_K - S_0^T A_0^{-1} R_0 & D - S_0 A_0^{-1} S_0 \end{pmatrix}$$

is psd. (We have assumed that $A \succeq 0$.) We can apply again a Schur decomposition to matrix M_K in order to reformulate the condition on Z. Setting $T_K := T - R_0^T A_0^{-1} R_0$, $Z' := Z_K - S_0^T A_0^{-1} R_0$, and $D' := D - S_0 A_0^{-1} S_0$, we have that $M_K = \left(\begin{smallmatrix} T_K & Z'^T \\ Z' & D' \end{smallmatrix}\right)$. Let

D'_0 be a largest nonsingular submatrix of D' and let

$$D' = \begin{pmatrix} D'_0 & E \\ E^T & F \end{pmatrix}, \; M_K = \begin{pmatrix} T_K & Z_0'^T & Z_1'^T \\ Z'_0 & D'_0 & E \\ Z'_1 & E^T & F \end{pmatrix}$$

denote the corresponding block decompositions of D' and M_K. Taking the Schur complement of D'_0 in M_K, we obtain that $M_K \succeq 0$ if and only if

$$D' \succeq 0, \; T_K - Z_0'^T D_0'^{-1} Z'_0 \succeq 0, \;\; \text{and } Z'_1 - E^T D_0'^{-1} Z'_0 = 0.$$

Let $Y_K := Z[V_2, V_1 \cap K]$ denote the column submatrix of Z with column indices in $V_1 \cap K$ and set

$$V_K := \begin{matrix} \\ V_2 \cap K \\ V_2 \setminus K \end{matrix} \begin{matrix} V_1 \cap K \\ \begin{pmatrix} S_0^T A_0^{-1} R_0 \\ 0 \end{pmatrix} \end{matrix}, \; Q_K := \begin{pmatrix} D_0^{-1} & 0 \\ 0 & 0 \end{pmatrix}, \; G_K := \begin{pmatrix} -E^T D_0'^{-1} & I & 0 \end{pmatrix}.$$

Then,

$$T_K - Z_0'^T D_0'^{-1} Z'_0 = T_K - (Y_K - V_K)^T Q_K (Y_K - V_K),$$

$$Z'_1 - E^T D_0'^{-1} Z'_0 = G_K (Y_K - V_K).$$

Therefore, the condition $x(K) \succeq 0$ can be rewritten as the system

$$\begin{cases} \text{(1K)} \; T_K - (Y_K - V_K)^T Q_K (Y_K - V_K) \succeq 0, \\ \text{(2K)} \; G_K (Y_K - V_K) = 0, \end{cases}$$

where T_K, V_K, Q_K are matrices depending on input data a. We can reformulate condition (1K) as an equation by introducing a new square matrix S_K of order $V_1 \cap K$ as "slack variable"; namely, rewrite (1K) as

$$\text{(1'K)} \; T_K - (Y_K - V_K)^T Q_K (Y_K - V_K) - S_K^T S_K = 0.$$

Now, let $z_1, \ldots, z_p \in \mathbf{R}^{V_2}$ denote the columns of matrix Z, and let s_i^K (for $i \in V_1 \cap K$) denote the columns of matrix S_K for each clique K. Then, condition (1'K) can be expressed as a system of $\binom{|V_1 \cap K|+1}{2}$ equations of the form

$$f(z_1, \ldots, z_p, s_i^K \; (i \in V_1 \cap K)) = 0,$$

where f is a quadratic polynomial, similarly for condition (2K). The total number of quadratic equations obtained in this manner depends only on p and q. Therefore, in view of Theorem 5.2, one can check feasibility of this system in polynomial time when p and q are fixed. □

Let $\mathcal{G}'_{p,q}$ denote the subclass of $\mathcal{G}_{p,q}$ consisting of the graphs G for which every maximal clique of H (the chordal extension of G) which is not a clique of G is not contained in $V_1 \cup V_2$. Then, the Euclidean distance matrix completion problem can be solved in polynomial time over the class $\mathcal{G}'_{p,q}$ for any fixed p and q. The proof is similar to that of Theorem 5.1, since we can get back to the psd case using relation (2.4) (a matrix and its image under φ_{i_0} having the same pattern of unknown entries if i_0 belongs to $V \setminus (V_1 \cup V_2)$). In particular, the Euclidean distance matrix completion

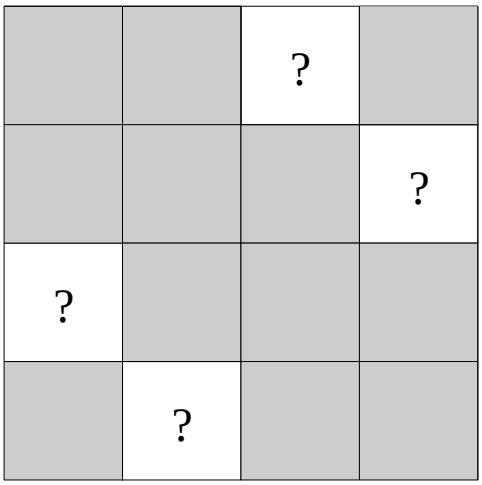


FIG. 5.2. *The matrix completion problem for generalized circuits of length* 4.

problem can be solved in polynomial time for generalized circuits of length 4 and fixed fatness, or for generalized wheels (with a nonempty center) of fixed length and fatness.

The complexity of the psd completion problem for generalized wheels and circuits is not known; in fact, in view of the remark made at the end of section 2.2, it suffices to consider circuits. In view of Theorem 5.1, the problem is polynomial if we fix the length and the fatness of the circuit. It would be particularly interesting to determine the complexity of the completion problem for generalized circuits of length 4 and unrestricted fatness. This problem can be reformulated as follows: Determine whether and how one can fill the unspecified entries in the blocks marked "?" of the matrix X shown in Figure 5.2, so as to obtain $X \succeq 0$. (All entries are assumed to be specified in the grey blocks.) Indeed, as will be seen in section 5.2, these graphs constitute in some sense the next case to consider after chordal graphs.

5.2. A polar approach to the completion problem. Given a graph $G = (V, E)$, consider the cone C_G consisting of the matrices $X = (x_{ij})_{i,j \in V}$ satisfying $X \succeq 0$ and $x_{ij} = 0$ for all $i \neq j$ such that $ij \notin E$. Call $X \in C_G$ *extremal* if X lies on an extremal ray of the cone C_G (i.e., $X = Y + Z$ with $Y, Z \in C_G$ implies that $Y = \alpha X$ for some $\alpha \geq 0$) and define the *order* of G as the maximum rank of an extremal matrix $X \in C_G$. It is shown in [1] that $a \in \mathbf{R}^{V \cup E}$ is completable to a psd matrix if and only if a satisfies

$$\sum_{ij \in E} a_{ij} x_{ij} + \sum_{i \in V} a_i x_{ii} \geq 0 \tag{5.1}$$

for every extremal matrix $X = (x_{ij}) \in C_G$. One might suspect that the psd matrix completion problem is somewhat easier to solve for graphs having a small order since the extremal matrices in C_G have then a small rank. Indeed, the graphs of order 1 are precisely the chordal graphs for which the problem is polynomially solvable. On the other hand, a circuit of length n has order $n - 2$ which is the highest possible order for a graph on n nodes. Moreover, if i_0 is a universal node in a graph G, then both graphs G and $G \setminus i_0$ have the same order, which corroborates the observation made

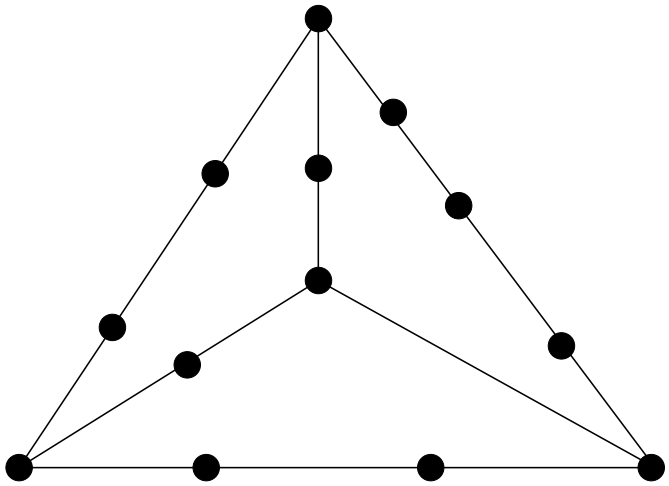


FIG. 5.3. *A homeomorph of* K_4.

at the end of section 2.2. A natural question concerns the complexity of the problem for graphs of order 2.

The graphs of order 2 have been characterized in [28]. It is shown there that, up to a simple graph operation (clique-sum), they belong to two basic classes $\mathcal{G}_1$ and $\mathcal{G}_2$. All the graphs in $\mathcal{G}_1$ have minimum fill-in at most 3; hence, the problem is polynomially solvable for them (by Theorem 1.2). The graphs in class $\mathcal{G}_2$ are the generalized wheels of length 4 (and unrestricted fatness). Hence, if the psd matrix completion problem is polynomially solvable for generalized wheels of length 4, then the same holds for all graphs of order 2.

5.3. The matrix completion problem for graphs with no homeomorph of K_4. We now discuss the matrix completion problem for the class $\mathcal{H}$ consisting of the graphs containing no homeomorph of K_4 as a subgraph; a *homeomorph* of K_4 being obtained from K_4 by replacing its edges with paths; cf. Figure 5.3 for an example. (Graphs in $\mathcal{H}$ are also known as *series parallel graphs*.) Clearly, $\mathcal{H}$ contains all circuits. The case of circuits is certainly interesting to understand since circuits are the most simple nonchordal graphs.

Similarly to the chordal case, a condition characterizing existence of a psd completion is known for the graphs in $\mathcal{H}$. Namely, the following is shown in [25] (using a result of [7]). Given a graph $G = (V, E)$ in $\mathcal{H}$ and $a \in \mathbf{R}^{V \cup E}$ satisfying $a_i = 1$ for all $i \in V$, then a has a psd completion if and only if the scalars $x_e := \frac{1}{\pi} \arccos a_e$ $(e \in E)$ satisfy the inequalities

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \leq |F| - 1 \quad \text{for all } F \subseteq C \text{ with } C \text{ circuit in } G, |F| \text{ odd.} \tag{5.2}$$

PROPOSITION 5.3 (see [6]). *Given $x \in [0,1]^E$, one can test in polynomial time whether x satisfies the linear system* (5.2).

Proof. Consider the graph $\tilde{G} := (V \cup V', \tilde{E})$ where $V' := \{i' \mid i \in V\}$ and $\tilde{E}$ consists of the pairs ij, $i'j'$, ij', $i'j$ for $ij \in E$. Define $z \in \mathbf{R}^{\tilde{E}}$ by $z_{ij} = z_{i'j'} = x_{ij}$ and $z_{i'j} = z_{ij'} = 1 - x_{ij}$ for $ij \in E$. Then, it is easy to see that x satisfies (5.2) if and only if $z(P) \geq 1$ for every path P from i to i' in $\tilde{G}$ and every $i \in V$. The result now follows as one can compute shortest paths in polynomial time. ∎

Therefore, problem (P_s) is polynomial time solvable in the real number model for graphs in $\mathcal{H}$. It is not clear how to extend this result to the bit model since the scalars $x_e := \frac{1}{\pi} \arccos a_e$ are in general irrational and, thus, one encounters problems of numerical stability when trying to check whether (5.2) holds.

Moreover, there is a simple combinatorial algorithm (already briefly mentioned in [26]) permitting us to construct a psd completion in polynomial time in the real number model. Let $G = (V, E)$ be a graph in $\mathcal{H}$ and let $a \in \mathbf{R}^{V \cup E}$ be given satisfying $a_i = 1$ for all $i \in V$. The algorithm performs the following steps.

1. Set $x_e := \frac{1}{\pi} \arccos a_e$ for $e \in E$ and test whether x satisfies (5.2). If not, one can conclude that a has no psd completion. Otherwise, go to step 2.

2. Find a set F of edges disjoint from E for which the graph $H := (V, E \cup F)$ is chordal and contains no homeomorph of K_4.

3. Find an extension $y \in [0,1]^{E \cup F}$ of x satisfying the linear system (5.2) with respect to graph H.

4. Set $b_e := \cos(\pi y_e)$ for $e \in E \cup F$ and $b_i := 1$ for $i \in V$. Then, b is completable to a psd matrix (since y satisfies (5.2) and H has no homeomorph of K_4) and one can compute a psd completion X of b with the algorithm of section 3.2 (since H is chordal). Then, X is a completion of a.

All steps can be executed in polynomial time. This follows from earlier results for steps 1 and 4; for step 2 use a result of [41] and, for step 3, one can use an argument similar to the proof of Proposition 5.3. Namely, given $x \in [0,1]^E$ satisfying (5.2), in order to extend x to $[0,1]^{E \cup \{e\}}$ in such a way that (5.2) remains valid with respect to $G + e$, one has to find a scalar $\alpha \in [0,1]$ satisfying $L_1 \leq \alpha \leq L_2$, where

$$L_1 := \max_{C, F \mid e \in C \setminus F} (x(F) - x(C \setminus (F \cup \{e\})) - |F| + 1),$$

$$L_2 := \min_{C, F \mid e \in F} (x(C \setminus F) - x(F \setminus e) + |F| - 1).$$

We have $L_1 \leq L_2$ (since x satisfies (5.2)) and $L_1 \leq 1$, $L_2 \geq 0$ (since $x \in [0,1]^E$); thus, $[L_1, L_2] \cap [0,1] \neq \emptyset$. With the notation of the proof of Proposition 5.3, one finds that

$$L_1 = 1 - \min(z(P) \mid P \text{ is an } ab'\text{-path in } \tilde{G}),$$

$$L_2 = \min(z(P) \mid P \text{ is an } ab\text{-path in } \tilde{G}).$$

Hence one can compute α in polytime. One can then determine the extension y of x to H by iteratively applying this procedure.

The distance matrix completion problem for graphs in $\mathcal{H}$ can be treated in a similar manner. Indeed, given $G = (V, E)$ in $\mathcal{H}$ and $a \in \mathbf{R}^E_+$, set $x_e := \sqrt{a_e}$ for $e \in E$. Then, a is completable to a distance matrix if and only if x satisfies the linear inequalities

$$x_e - \sum_{f \in C \setminus e} x_f \leq 0 \quad \text{for all circuits } C \text{ in } G \text{ and all } e \in C \tag{5.3}$$

(cf. [27]). Again one can test in polynomial time whether $x \geq 0$ satisfies (5.3). (Simply, test for each edge $e = ab \in E$ whether $x_e \leq \min(x(P) \mid P \text{ is an } ab\text{-path in } G)$.) An algorithm analogous to the one exposed in the psd case permits us to construct a distance matrix completion. Therefore, we have shown the following theorem.

THEOREM 5.4. *One can construct a real psd (distance matrix) completion or decide that none exists in polynomial time in the real number model for the graphs containing no homeomorph of* K_4. □

It is an open question whether the above result extends to the bit model of computation, even for the simplest case of circuits.

Acknowledgments. We are grateful to A. Barvinok for providing us insight about Theorem 5.2, to L. Porkolab for bringing [24] to our attention, and to A. Schrijver for discussions about section 3. We also thank the referees for their careful reading and for their suggestions which helped us improve the presentation of the paper.

REFERENCES

[1] J. AGLER, J. W. HELTON, S. MCCULLOUGH, AND L. RODMAN, *Positive semidefinite matrices with a given sparsity pattern*, Linear Algebra Appl., 107 (1988), pp. 101–149.
[2] A. Y. ALFAKIH, A. KHANDANI, AND H. WOLKOWICZ, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1998), pp. 13–30.
[3] F. ALIZADEH, *Interior point methods in semidefinite programming with applications in combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.
[4] M. BAKONYI AND C. R. JOHNSON, *The Euclidean distance matrix completion problem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 646–654.
[5] M. BAKONYI AND G. NAEVDAL, *On the matrix completion method for multidimensional moment problems*, Acta Sci. Math. (Szeged), 64 (1998), pp. 547–558.
[6] F. BARAHONA AND A. R. MAHJOUB, *On the cut polytope*, Math. Programming, 36 (1986), pp. 157–173.
[7] W. BARRETT, C. R. JOHNSON, AND P. TARAZAGA, *The real positive definite completion problem for a simple cycle*, Linear Algebra Appl., 192 (1993), pp. 3–31.
[8] A. I. BARVINOK, *Feasibility testing for systems of real quadratic equations*, Discrete Comput. Geom., 10 (1993), pp. 1–13.
[9] A. I. BARVINOK, *personal communication*, 1998.
[10] L. BLUM, M. SHUB, AND S. SMALE, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc., 21 (1989), pp. 1–46.
[11] G. M. CRIPPEN AND T. F. HAVEL, *Distance Geometry and Molecular Conformation*, Research Studies Press, Taunton, Somerset, England, 1988.
[12] H. DYM AND I. GOHBERG, *Extensions of band matrices with band inverses*, Linear Algebra Appl., 36 (1981), pp. 1–24.
[13] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.
[14] M. C. GOLUMBIC, *Algorithmic Theory and Perfect Graphs*, Academic Press, New York, 1980.
[15] R. GRONE, C. R. JOHNSON, E. M. SÁ, AND H. WOLKOWICZ, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.
[16] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.
[17] C. HELMBERG, F. RENDL, R. J. VANDERBEI, AND H. WOLKOWICZ, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.
[18] B. HENDRICKSON, *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5 (1995), pp. 835–857.
[19] C. R. JOHNSON, *Matrix completion problems: A survey*, in Matrix Theory and Applications 40, Proc. Sympos. Appl. Math., C. R. Johnson, ed., AMS, Providence, RI, 1990, pp. 171–198.
[20] C. R. JOHNSON, B. KROSCHEL, AND H. WOLKOWICZ, *An interior-point method for approximate positive semidefinite completions*, Comput. Optim. Appl., 9 (1998), pp. 175–190.
[21] C. R. JOHNSON AND P. TARAZAGA, *Connections between the real positive semidefinite and distance matrix completion problems*, Linear Algebra Appl., 223/224 (1995), pp. 375–391.
[22] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[23] L. KHACHIYAN, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady, 20 (1979), pp. 191–194.

[24] L. Khachiyan and L. Porkolab, *Computing integral points in convex semi-algebraic sets*, in 38th Annual IEEE Symposium on Foundations of Computer Science, Miami, FL, 1997, pp. 162–171.
[25] M. Laurent, *The real positive semidefinite completion problem for series-parallel graphs*, Linear Algebra Appl., 252 (1997), pp. 347–366.
[26] M. Laurent, *Cuts, matrix completions, and graph rigidity*, Math. Programming, 79 (1997), pp. 255–283.
[27] M. Laurent, *A connection between positive semidefinite and Euclidean distance matrix completion problems*, Linear Algebra Appl., 273 (1998), pp. 9–22.
[28] M. Laurent, *On the order of a graph and its deficiency in chordality*, Combinatorica, to appear.
[29] J. de Leeuw and W. Heiser, *Theory of multidimensional scaling*, in Handbook of Statistics, Vol. 2, P. R. Krishnaiah and L. N. Kanal, eds., North Holland, 1982, pp. 285–316.
[30] H. W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.
[31] J. J. Moré and Z. Wu, *Distance geometry optimization for protein structures*, J. Global Optim., 15 (1999), pp. 219–234.
[32] Y. E. Nesterov and A. S. Nemirovsky, *Interior Point Polynomial Algorithms in Convex Programming: Theory and Algorithms*, SIAM, Philadelphia, 1994.
[33] L. Porkolab and L. Khachiyan, *On the complexity of semidefinite programs*, J. Global Optim., 10 (1997), pp. 351–365.
[34] L. Porkolab, *private communication*, 2000.
[35] M. V. Ramana, *An exact duality theory for semidefinite programming and its complexity implications*, Math. Programming, 77 (1997), pp. 129–162.
[36] 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.
[37] I. J. Schoenberg, *Remarks to M. Fréchet's article "Sur la définition axiomatique d'une classe d'espaces vectoriels distanciés applicables vectoriellement sur l'espace de Hilbert*," Ann. of Math., 36 (1935), pp. 724–732.
[38] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons, New York, 1986.
[39] R. E. Tarjan, *Decomposition by clique separators*, Discrete Math., 55 (1985), pp. 221–232.
[40] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.
[41] J. A. Wald and C. J. Colbourn, *Steiner trees, partial 2-trees and minimum IFI networks*, Networks, 13 (1983), pp. 159–167.
[42] H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
[43] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1996.
[44] 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. 22, No. 3, pp. 895–911

INCOMPLETE MULTILEVEL CHOLESKY FACTORIZATIONS*

J. C. DIAZ† AND K. KOMARA†

Abstract. Adaptive in-time local grid refinement techniques use multilevel local discretizations designed to achieve local accuracy. The changing nature of the matrix structure of the linear systems arising from the multilevel local discretizations requires flexible approximate factorizations that focus on local components and coordinate their interaction. The solution of these composite grid systems with Krylov solvers is considered. The selection of an adequate preconditioner is crucial. The incomplete Cholesky (IC) factorization of the composite matrix and the inexact BEPS preconditioner are two such potential preconditioners. The inexact BEPS preconditioner can be constructed and applied with significantly more flexibility than the IC factorizations of the composite multilevel grid matrix.

An extension of the IC factorizations for matrices arising from discretizations of self-adjoint PDEs on multilevel composite grids is proposed. The resulting factorization is referred to as the incomplete multilevel Cholesky (IMC) factorization. The IMC factorization is spectrally equivalent to the IC factorization of the matrix of the composite grid system.

IMC factorization can be constructed with the same flexibility as the inexact BEPS preconditioner. The application of IMC factorization is achieved via a multilevel LL^T-cycle consisting of a forward elimination pass proceeding downward on the grids from fine to coarse followed by an reverse-order upward back substitution pass.

The application of multilevel factorization as a preconditioner requires roughly one-half as many operations as the inexact BEPS preconditioner. The numerical results illustrate the potential of the method.

Key words. Krylov methods, multilevel methods, sparse linear systems, incomplete factorizations, preconditioning

AMS subject classifications. Primary, 86A60, 65N55; Secondary, 65C20

PII. S0895479896311128

1. Introduction.

Adaptive local grid refinement techniques are used to solve time-dependent PDE problems whose solutions exhibit changing localized phenomena [6, 10, 15, 16, 18, 19]. Multilevel local discretizations are designed to achieve more accuracy in the regions where the solution's local behavior may have a significant impact on the entire domain. A composite grid can be thought of as a collection of nested grids which may change from time to time. It contains one coarse and some local nested finer grids for regions of the computational domain selected for refinement [6, 15, 16, 18, 19].

The resulting multilevel systems of linear equations are solved using Krylov methods with preconditioning. One important issue is the changing nature of the structure of the matrices of the multilevel systems of linear equations as the grid is adapted in time. Hence, the data structure used to store the individual matrices has to be flexible enough to work with the constantly changing shape of the matrix. The composite grid matrix should not be assembled. The factorization should focus on the components and coordinate their interaction. This paper discusses issues relating to efficient factorizations that can be used as preconditionings for the multilevel systems.

*Received by the editors October 25, 1996; accepted for publication (in revised form) by J. Liu March 28, 1998; published electronically December 20, 2000. This research was partly supported by Oklahoma Center for Advancement of Science and Technology grants RB9-008 (3748) and ARO-36 (3910).

http://www.siam.org/journals/simax/22-3/31112.html

†Center for Parallel and Scientific Computing, University of Tulsa, 600 S. College Ave., Tulsa, OK 74104-3189 (diaz@utulsa.edu).

Incomplete LL^T factorizations are extended to matrices arising from discretizations of self-adjoint PDEs on composite grids. The proposed incomplete multilevel LL^T factorization is derived from the incomplete Cholesky (IC) factorization matrices of the discretizations on the coarse and local grids. The existence of the incomplete multilevel Cholesky (IMC) factorization depends on the existence of IC factorizations of the grid matrices which have been extensively studied [2, 3, 4, 7, 13, 20]. The flexible construction of the IMC factorization allows more parallelism than would be allowed by the IC factorization of the assembled composite grid matrix.

This paper considers multilevel linear systems of equations with symmetric positive definite M-matrices [5]. The main result of this paper is to prove that the IMC factorization is spectrally equivalent to the IC factorization of the assembled composite grid matrix. Further, the condition number of the composite grid matrix preconditioned by the IMC factorization is shown to be bounded by a constant times the condition number of the composite grid matrix preconditioned by the IC factorization. This implies that the IMC factorization is as robust as the IC factorization for preconditioning the composite grid matrix. Flexibility has been achieved without sacrificing stability.

There is a similarity between the IMC factorization and the inexact BEPS preconditioner [15, 18, 16]. The application of either on a vector requires a downward pass followed by a upward pass on the nested levels. During each cycle, downward and upward, the inexact BEPS preconditioner performs a forward and a backward solve at each level. However, the IMC preconditioner requires only a forward elimination at each level for the downward pass and a backward substitution in the upward pass. Therefore, the action of IMC as a preconditioning on a vector requires nearly half the operations of the inexact BEPS preconditioner. The incomplete multilevel IC has the familiar feel of an LL^T factorization consisting of a forward elimination pass followed by a backward substitution pass.

The factorizations are introduced in section 2, which includes a brief review of the IC factorizations considered herein. That section considers approximate factorizations for the composite grid matrix for a two-level grid system including the IC factorizations, the inexact BEPS preconditioner, and the construction of IMC. The cost complexity of the application of the inexact BEPS preconditioner and IMC factorization are compared for a multilevel grid system. The stability analysis of the IMC factorization is investigated in section 3. Comparative experiments that illustrate the potential of IMC factorizations used as preconditionings are presented in section 4. Finally, section 5 includes a summary of the main results presented in this paper.

2. The factorizations. The three factorizations for the composite grid matrix are introduced. Section 2.1 presents a brief overview of the IC factorizations considered herein. A two-level grid is presented in section 2.2, which also includes a presentation of the composite and coarse grid systems. Section 2.3 presents the composite grid matrix and its IC factorization for a two level-system. The inexact BEPS preconditioner is reviewed in section 2.4. The IMC factorization is presented in section 2.5. Like the inexact BEPS preconditioner, the IMC factorization can be defined for multiple levels. A simple comparison of the complexity of the application of the inexact BEPS preconditioner and the IMC factorization for multiple levels is presented in section 2.6.

2.1. IC factorizations. This section provides a brief overview of IC factorizations. The matrices obtained from finite difference discretizations of two-dimensional self-adjoint PDEs on rectangular regions are symmetric positive definite M-matrices.

Let $A = L + D_A + L^T$ be an M-matrix of order n, where D_A is the diagonal (or block diagonal) of A. The IC factorization of the matrix A considered herein has the form

$$Q = (L + D)D^{-1}(D + L^T), \tag{1}$$

where L is the strictly lower (block) triangular part of A, and D is a (block) diagonal matrix. The diagonal or block diagonal matrix D is usually different from D_A. When both are diagonals, the factorization is a *point* factorization of A. Otherwise, it is a *block* factorization. Point and block IC factorizations are well defined for symmetric positive definite M-matrices [1, 2, 3, 4, 7, 9, 13, 20, 21].

The point factorization of the matrix A is determined by the condition $diagonal(A - Q) = 0$. Let $A = (a_{i,j})$ and $D = diagonal(d_i)$. Then the diagonal elements of D are computed as follows:

$$\begin{aligned} d_{1,1} &= a_{1,1}, \\ d_{i,i} &= a_{i,i} - \sum_{s=1}^{i-1} a_{i,s} d_{s,s}^{-1} a_{s,i} \quad \text{for } i = 2, \ldots, n. \end{aligned}$$

The IC factorization defined in this manner is usually referred to as DKR (Dupont–Kendall–Rachford) factorization [13], the IC(0) [14], or the standard IC factorization (with no fill-in) [12].

The block factorization assumes the matrix $A = blocktrid(A_{i-1,i}, A_{i,i}, A_{i,i+1})$, $i = 1, \ldots, m$, to have a block tridiagonal form where the square diagonal blocks $A_{i,i}$ are tridiagonal, and $A_{i-1,i}$ and $A_{i,i+1}$ are diagonal. Let $D = blockdiag(D_{i,i})$; then the diagonal elements of D are given by

$$\begin{aligned} D_{1,1} &= A_{1,1}, \\ D_{i,i} &= A_{i,i} - A_{i,i-1} X_{i-1,i-1} A_{i-1,i} \quad \text{for } i = 2, \ldots, m, \end{aligned}$$

where $X_{i-1,i-1}$ is an approximate inverse of $D_{i-1,i-1}$. Different choices for the selection of $X_{i,i}$ exist. In this paper, the consideration is limited to $X_{i,i} = \left[D_{i,i}^{-1}\right]^{(p)}$, where $\left[D_{i,i}^{-1}\right]^{(p)}$ is a matrix with half bandwidth p whose elements inside the band coincide with those of $D_{i,i}^{-1}$. This IC factorization Q is referred to as INV(1) factorization when $p = 1$ and INV(2) factorization when $p = 2$ [7, 9].

2.2. Two-level grid. Consider a composite grid with two grids, one coarse and one fine. Let the computational domain Ω be a rectangular subregion of $\mathcal{R}^2$. Introduce the coarse grid $\widetilde{\omega}$ in Ω with spacing h_c. The subregion selected for refinement is $\Omega^{(f)}$. Introduce the fine grid ω_f in $\Omega^{(f)}$ with spacing h_f. $\Omega \setminus \Omega^{(f)}$ is the unrefined subregion of Ω. Let ω_u denote the set of coarse grid points in the unrefined subregion. Similarly let ω_r denote the set of coarse grid points in the refined region $\Omega^{(f)}$. Thus, $\widetilde{\omega} = \omega_r \cup \omega_u$. The composite grid is the set of grid points in $\omega_f \cup \omega_u$ and is denoted by ω. Figure 1 illustrates a cell-centered composite grid.

Let A be the composite grid matrix on ω, and let $\widetilde{A}$ be the coarse grid matrix on $\widetilde{\omega}$. Assume a natural ordering. Partition a vector in ω by ordering first the components in ω_f and then the components in ω_u. Then partition the composite grid system $Ax = b$ in a 2×2 block form:

$$Ax = \begin{bmatrix} A_{f,f} & A_{u,f}^T \\ A_{u,f} & A_{u,u} \end{bmatrix} \begin{bmatrix} x_f \\ x_u \end{bmatrix} = \begin{bmatrix} b_f \\ b_u \end{bmatrix} = b. \tag{2}$$

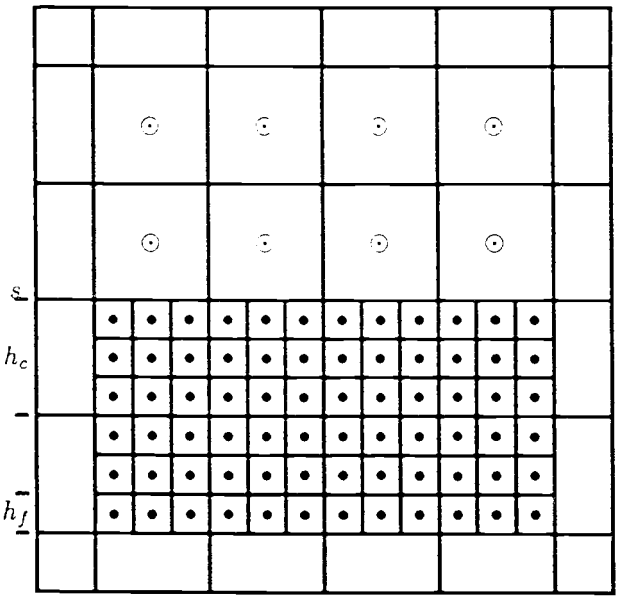


FIG. 1. *Two-level composite grid. Grid points are labeled* $\odot$ *for coarse and* $\bullet$ *for fine.*

Similarly, the coarse system $\widetilde{A}\widetilde{x} = \widetilde{b}$ can also be partitioned in a 2×2 block form:

$$\widetilde{A}\widetilde{x} = \begin{bmatrix} \widetilde{A}_{r,r} & \widetilde{A}_{u,r}^T \\ \widetilde{A}_{u,r} & \widetilde{A}_{u,u} \end{bmatrix} \begin{bmatrix} \widetilde{x_r} \\ \widetilde{x_u} \end{bmatrix} = \begin{bmatrix} \widetilde{b}_r \\ \widetilde{b}_u \end{bmatrix} = \widetilde{b}. \tag{3}$$

Consider the splittings $A_{i,i} = L_{i,i} + D_{Ai,i} + L_{i,i}^T$ for $i = f, u$, where $L_{i,i}$ is the strictly (block) lower part of $A_{i,i}$ and $D_{Ai,i}$ is the (block) diagonal part of $A_{i,i}$. Also consider similar splittings for $\widetilde{A_{i,i}}$ for $i = r, u$. Let I_i be the identity matrix on the grid ω_i for $i = f, r, u$.

The submatrices $A_{u,u}$ and $\widetilde{A}_{u,u}$ act on the same grid points, the unrefined portion of the coarse grid. The off-diagonal entries account for discrete relations among grid points in the unrefined grid. Hence, $\widetilde{L}_{u,u} = L_{u,u}$. However, the diagonals of $\widetilde{A}_{u,u}$ and $A_{u,u}$ differ because $\widetilde{A}_{u,u}$ also accounts for discrete relations with grid points in ω_r, whereas $A_{u,u}$ must also account for discrete relations with grid points in ω_f.

2.3. IC factorizations of the composite matrix. The incomplete factorizations of $\widetilde{A}$ and A, and therefore that of $A_{f,f}$, exist because they are assumed to be symmetric positive definite M-matrices [2, 3, 4, 7, 13, 20, 21].

The IC factorization of the composite grid matrix A is given by

$$Q_{\text{IC}} = \begin{bmatrix} L_{f,f} + D_{f,f} & 0 \\ A_{u,f} & L_{u,u} + D_{u,u} \end{bmatrix} \begin{bmatrix} D_{f,f}^{-1} & 0 \\ 0 & D_{u,u}^{-1} \end{bmatrix} \begin{bmatrix} D_{f,f} + L_{f,f}^T & A_{u,f}^T \\ 0 & D_{u,u} + L_{u,u}^T \end{bmatrix}, \tag{4}$$

where $Q_{f,f} = (L_{f,f} + D_{f,f})D_{f,f}^{-1}(D_{f,f} + L_{f,f}^T)$ is the IC factorization of $A_{f,f}$ and $\left[\begin{smallmatrix} D_{f,f} & 0 \\ 0 & D_{u,u} \end{smallmatrix}\right]$ is a (block) diagonal matrix. $D_{f,f}$ and $D_{u,u}$ are diagonal matrices with positive diagonal elements for point incomplete factorizations [13, 20, 21]. Note that $(L_{u,u} + D_{u,u})D_{u,u}^{-1}(D_{u,u}^{-1} + L_{u,u}^T)$ is an IC factorization of $A_{u,u} - \chi$, where χ is the

diagonal or block diagonal of $A_{u,f}Q_{f,f}^{-1}A_{u,f}^T$. For block incomplete factorizations $D_{f,f}$ and $D_{u,u}$ are symmetric positive definite block diagonal M-matrices, [4, 3, 2, 7].

Similarly, the IC factorization of the coarse grid matrix $\widetilde{A}$ is given by

$$\widetilde{Q} = \begin{bmatrix} \widetilde{L}_{r,r} + \widetilde{D}_{r,r} & 0 \\ \widetilde{A}_{u,r} & L_{u,u} + \widetilde{D}_{u,u} \end{bmatrix} \begin{bmatrix} \widetilde{D}_{r,r}^{-1} & 0 \\ 0 & \widetilde{D}_{u,u}^{-1} \end{bmatrix} \begin{bmatrix} \widetilde{D}_{r,r} + \widetilde{L}_{r,r}^T & \widetilde{A}_{u,r}^T \\ 0 & \widetilde{D}_{u,u} + L_{u,u}^T \end{bmatrix}, \tag{5}$$

where $\begin{bmatrix} \widetilde{D}_{r,r} & 0 \\ 0 & \widetilde{D}_{u,u} \end{bmatrix}$ is a (block) diagonal matrix. Again, $\widetilde{D}_{r,r}$ and $\widetilde{D}_{u,u}$ are diagonal matrices with positive diagonal elements for point incomplete factorizations and are symmetric positive definite block diagonal M-matrices for block incomplete factorizations. Let $\widetilde{Q}_{r,r} = (\widetilde{L}_{r,r} + \widetilde{D}_{r,r})\widetilde{D}_{r,r}^{-1}(\widetilde{D}_{r,r} + \widetilde{L}_{r,r}^T)$. Note that $\widetilde{Q}_{u,u} = (L_{u,u} + \widetilde{D}_{u,u})\widetilde{D}_{u,u}^{-1}(\widetilde{D}_{u,u}^{-1} + L_{u,u}^T)$ is an IC factorization of $\widetilde{A}_{u,u} - \chi$, where χ is the diagonal or block diagonal of $\widetilde{A}_{u,r}\widetilde{Q}_{r,r}^{-1}\widetilde{A}_{u,r}^T$.

2.4. Inexact BEPS preconditioners. This section reviews the BEPS preconditioner [15, 16, 18]. Let $\widetilde{A}$, $A_{f,f}$, and A be the partitioned matrices given in (2) and (3). The BEPS preconditioner is given by

$$Q_{\text{BEPS}} = \begin{bmatrix} A_{f,f} & 0 \\ A_{u,f} & S_{u,u} \end{bmatrix} \begin{bmatrix} I_f & A_{f,f}^{-1}A_{u,f}^T \\ 0 & I_u \end{bmatrix}, \tag{6}$$

where $S_{u,u} = \widetilde{A}_{u,u} - \widetilde{A}_{u,r}\widetilde{A}_{r,r}^{-1}\widetilde{A}_{r,u}$ is the Schur complement of $\widetilde{A}_{r,r}$ in $\widetilde{A}$. The action of the inverse of $S_{u,u}$ on a vector b_u on ω_u is obtained by solving a problem of the form

$$\widetilde{A}\widetilde{x} = \widetilde{A}\begin{bmatrix} * \\ x_u \end{bmatrix} = \begin{bmatrix} 0 \\ b_u \end{bmatrix} = \widetilde{b}.$$

The inexact BEPS considered here consists of replacing $A_{f,f}$ and $\widetilde{A}$ with respective IC factorizations. Using the IC factorizations of $A_{f,f}$ and $\widetilde{A}$ given in section 2.5, the inexact BEPS preconditioner is defined by

$$Q_{\text{IBEPS}} = \begin{bmatrix} Q_{f,f} & 0 \\ A_{u,f} & \widetilde{Q}_{u,u} \end{bmatrix} \begin{bmatrix} I_f & Q_{f,f}^{-1}A_{u,f}^T \\ 0 & I_u \end{bmatrix}, \tag{7}$$

where $Q_{f,f}$ is the IC factorization of $A_{f,f}$ and $\widetilde{Q}_{u,u}$ is determined from the IC factorization $\widetilde{Q}$ of $\widetilde{A}$ such that the solution of $\widetilde{Q}_{u,u}x_u = b_u$ is obtained by solving the problem

$$\widetilde{Q}\widetilde{x} = \widetilde{Q}\begin{bmatrix} * \\ x_u \end{bmatrix} = \begin{bmatrix} 0 \\ b_u \end{bmatrix} = \widetilde{b}. \tag{8}$$

Together, (7) and (8) define the inexact BEPS preconditioner.

Note that the ordering used to compute the factorization $\widetilde{Q}$ for (8) does not need to coincide with the ordering implicit in the partition of $\widetilde{A}$ in (3). Therefore, $\widetilde{Q}$ can be computed for the coarse grid $\widetilde{\omega}$ independently of where the region of refinement ω_r is to be located. Hence, inexact BEPS provides more flexibility than IC of the composite grid matrix.

The construction of the inexact BEPS requires the IC factorizations of the fine grid matrix $A_{f,f}$ and the whole coarse grid matrix $\widetilde{A}$. The action of the inexact

BEPS preconditioner on a composite grid vector requires more work than the IC preconditioner. Inexact BEPS requires a forward and a backward solve for each level of the downward and upward passes. Inexact BEPS is applied first on a downward pass on the grids from fine to coarse followed by a reverse-order upward pass. The action of the IC preconditioner is obtained by performing a downward forward elimination pass followed by an upward back substitution pass.

2.5. IMC factorizations. This section introduces the IMC factorization. Constructing Q_{IMC} requires the IC factorizations of the fine grid matrix $A_{f,f}$ and the whole coarse grid matrix $\widetilde{A}$. As with the inexact BEPS method, the factorizations of $A_{f,f}$ and $\widetilde{A}$ can be done independently.

The IMC factorization of A is obtained by replacing the matrix $D_{u,u}$ with the matrix $\widetilde{D}_{u,u}$ in the IC factorization of $\widetilde{A}$. Thus, it is explicitly given by

$$Q_{\text{IMC}} = \begin{bmatrix} L_{f,f} + D_{f,f} & 0 \\ A_{u,f} & L_{u,u} + \widetilde{D}_{u,u} \end{bmatrix} \begin{bmatrix} D_{f,f}^{-1} & 0 \\ 0 & \widetilde{D}_{u,u}^{-1} \end{bmatrix} \begin{bmatrix} D_{f,f} + L_{f,f}^T & A_{u,f} \\ 0 & \widetilde{D}_{u,u} + L_{u,u}^T \end{bmatrix}. \tag{9}$$

The action of Q_{IMC} on a composite grid vector requires the action of the Schur complement in $\widetilde{Q}$ obtained after eliminating the submatrix associated with the grid points in ω_r. This Schur complement is $\widetilde{Q}_{u,u} = (L_{u,u} + \widetilde{D}_{u,u})\widetilde{D}_{u,u}^{-1}(\widetilde{D}_{u,u} + L_{u,u}^T)$. As was the case for the inexact BEPS, the solution of $\widetilde{Q}_{u,u}x_u = b_u$ is obtained by solving (8). Together, (9) and (8) define the IMC factorization Q_{IMC}.

As was the case for the inexact BEPS, the ordering used to compute the factorization $\widetilde{Q}$ in (8) does not need to coincide with the ordering implicit in the partition of $\widetilde{A}$ in (3). $\widetilde{Q}$ can be computed for the coarse grid $\widetilde{\omega}$ independently of where the region of refinement ω_r is to be located.

Using inexact BEPS, Q_{IMC}, and Q_{IC} as preconditionings in conjunction with Krylov solvers requires the action of the preconditioners on a vector. A central feature of the IMC factorization is that its action on a vector can be carried out by a downward forward elimination pass followed by an upward back substitution pass. In the downward pass, fine to coarse, forward eliminations are performed on the fine and coarse grids. In the upward pass, coarse to fine, back substitutions are performed in the reverse order on the coarse and fine grids.

The construction of the inexact BEPS and the IMC preconditioners requires the same amount of work. They both need the IC factorizations of the fine grid matrix $A_{f,f}$ and the coarse grid matrix $\widetilde{A}$. The action of the inexact BEPS preconditioner on a composite grid vector, however, requires more work than the IMC preconditioner.

It must be noted here that other forms of inexact BEPS preconditioner exist where $Q_{f,f}$ and $\widetilde{Q}$ in (7) and (8) are not IC factorizations. Whereas in some cases corresponding IMC factorizations exist, these are not discussed herein.

2.6. Multilevel complexity. Like the inexact BEPS preconditioner, the IMC factorization can be defined for multilevels and for non-self-adjoint problems. Their construction requires only the construction of approximate Cholesky factors of the matrices at each level [11, 19], under the assumption that the refined subregions are nested. The construction for the two-level case is the recursive step for the multilevel case. The factorization is constructed proceeding through the nested levels of refinement starting from the finest and ending at the coarsest level. In order to extend the factorization from two levels to a third level, the current coarse level acts as

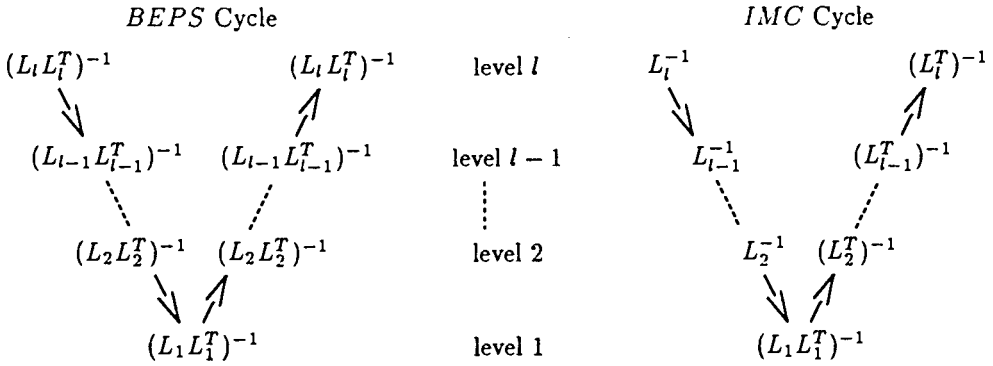


FIG. 2. *Computational flow of the action of BEPS and IMC on a composite grid vector.*

the fine level and the third level acts as the coarse level. In fact, all the subdomain factorizations are computed independently for each level.

The actions of the inexact BEPS preconditioner and the IMC preconditioner on a composite grid vector are visualized in Figure 2, where the lower and upper (block) triangular matrices, L_k and L_k^T, are the approximate Cholesky factors of the grid matrix on level k. The arrows in the figure signify movement of information from fine to coarse grid and vice versa.

During the downward and upward passes, the inexact BEPS preconditioner solves an LL^T system for each level. The IMC solves only one triangular linear system on each level during the downward pass (forward elimination) and only one triangular linear system during the upward pass (back substitution). The complexity of IMC preconditioners is nearly half of that of the inexact BEPS preconditioner.

The outline of the cycle given in Figure 2 can be used to visualize the extension of application of IMC. The presentation in section 2.5 assumes that the factorizations can be put into the form given by (1). There are indeed many such approximate factorizations where the L of (1) does not coincide with the strictly lower part of A. These factorizations can be applied to each subdomain resulting on an inexact BEPS which still has the same cycle as the one given in Figure 2. As long as the preconditioning applied at each grid level has the form of an LL^T factorization, IMC is equally applicable as depicted in Figure 2. IMC factorizations can easily be incorporated with only minor changes in existing preconditioned iterative methods using inexact BEPS preconditioners.

3. Stability analysis. This section considers the properties of the incomplete multilevel factorization and compares them with those of the IC factorization. The main results of this section are the spectral equivalence of Q_{IC} and Q_{IMC} and a relation between the condition numbers of $Q_{\text{IC}}^{-1}A$ and $Q_{\text{IMC}}^{-1}A$. These are summarized in two theorems in the following section. Their proofs are presented in section 3.2.

3.1. Equivalence of Q_{IMC} and Q_{IC}. The following theorem establishes the spectral equivalence of the factorizations Q_{IMC} and Q_{IC}. The proof of this theorem is presented in section 3.2.

THEOREM 1. *The matrices Q_{IMC} and Q_{IC} are spectrally equivalent; therefore, for any nonzero vector x of the composite grid $\omega = \omega_f \cup \omega_u$, it follows that*

$$\gamma_1 x^T Q_{IMC} x \le x^T Q_{IC} x \le \gamma_2 x^T Q_{IMC} x, \tag{10}$$

where γ_1 and γ_2 are constants such that $0 < \gamma_1 \leq 1 \leq \gamma_2$.

The following theorem derives a relation between $\kappa(Q_{\text{IMC}}^{-1}A)$ and $\kappa(Q_{\text{IC}}^{-1}A)$. This relation provides an insight on the performances of Q_{IMC} and Q_{IC} when used as preconditionings in conjunction with Krylov solvers.

THEOREM 2. *For the IMC preconditioner Q_{IMC} and the IC preconditioner Q_{IC} of the composite grid matrix A, there exists a constant $C_\gamma \geq 1$ such that*

$$\kappa(Q_{IMC}^{-1}A) \leq C_\gamma \kappa(Q_{IC}^{-1}A) \quad \textit{and} \quad C_\gamma = \frac{\gamma_2}{\gamma_1}, \tag{11}$$

where γ_1 and γ_2 are constants given in Theorem 1.

Proof. Rewrite $Q_{\text{IMC}}^{-1}A$ as

$$Q_{\text{IMC}}^{-1}A = Q_{\text{IMC}}^{-1}Q_{\text{IC}}Q_{\text{IC}}^{-1}A.$$

It then follows that

$$\kappa(Q_{\text{IMC}}^{-1}A) \leq \kappa(Q_{\text{IMC}}^{-1}Q_{\text{IC}}) \cdot \kappa(Q_{\text{IC}}^{-1}A).$$

From Theorem 1 and Lemma 4 (see section 3.2), it follows that $\kappa(Q_{\text{IMC}}^{-1}Q_{\text{IC}}) \leq \gamma_2/\gamma_1$. Because $0 < \gamma_1 \leq \gamma_2$, the $\gamma_2/\gamma_1 \geq 1$. Let $C_\gamma = \gamma_2/\gamma_1$ and use the last two inequalities to complete the proof of the theorem.

If A is a symmetric positive definite matrix and the linear system $Ax = b$ is solved using the preconditioned conjugate gradient (PCG) [8] method with a symmetric positive definite preconditioner M, then the number of PCG iterations needed to reduce A^{-1}-norm of the initial residual by a factor ϵ is $\mathcal{O}(\kappa(M^{-1}A)^{1/2}\ln(\frac{1}{\epsilon}))$ [1]. From Theorem 2 and the convergence properties of the PCG method, it follows that the preconditioner Q_{IMC} will be a good substitute for the preconditioner Q_{IC} if the constant C_γ of Theorem 2 is independent of the average discretization mesh size and is not too large.

3.2. Spectral equivalence. This is a very technical section intended to prove Theorem 1. On a first reading the reader may want to skip to section 4 for the numerical results. The main result of this section is the proof of the spectral equivalence of Q_{IC} and Q_{IMC}.

The Schur complements of $Q_{f,f} = (L_{f,f} + D_{f,f})(I_f + D_f^{-1}L_{f,f}^T)$ in Q_{IC} and Q_{IMC} are given by

$$S_{\text{IC}} = (L_{u,u} + D_{u,u})D_{u,u}^{-1}(D_{u,u} + L_{u,u}^T)$$

and

$$S_{\text{IMC}} = (L_{u,u} + \widetilde{D}_{u,u})\widetilde{D}_{u,u}^{-1}(\widetilde{D}_{u,u} + L_{u,u}^T),$$

where $D_{u,u}$ and $\widetilde{D}_{u,u}$ are symmetric positive definite M-matrices.

The spectral equivalence of S_{IC} and S_{IMC} will be presented first. From this the spectral equivalence of Q_{IC} and Q_{IMC} is established.

First recall that if G is a symmetric positive definite block diagonal matrix and L a nonsingular block strictly lower triangular matrix, then the matrix LGL^T is symmetric positive definite. This property guarantees that matrices of the form $(G + L)G^{-1}(G + L^T)$, where G is symmetric positive definite and L is lower triangular, are symmetric positive definite. Therefore, the matrices S_{IC}, S_{IMC}, Q_{IC}, and Q_{IMC} are

symmetric positive definite. The following technical result will be used to establish the spectral equivalence of the Schur complements S_{IC} and S_{IMC}.

LEMMA 1. *Let G_1 and G_2 be two symmetric positive definite block diagonal matrices of order n, and let L be a strictly block lower triangular matrix of order n such that $L \leq 0$. If there exist constants $0 < \alpha \leq \beta$ such that for any n-vector x the inequalities*

$$\alpha x^T G_2 x \leq x^T G_1 x \leq \beta x^T G_2 x$$

hold, then the matrices

$$F_1 = (G_1 + L)G_1^{-1}(G_1 + L^T) \quad \text{and} \quad F_2 = (G_2 + L)G_2^{-1}(G_2 + L^T)$$

are spectrally equivalent; that is, there exist constants $0 < \gamma_1 \leq 1 \leq \gamma_2$ such that for any n-vector x

$$\gamma_1 x^T F_2 x \leq x^T F_1 x \leq \gamma_2 x^T F_2 x. \tag{12}$$

Proof. Let $\sigma = \max(\beta, 1/\alpha)$. Define

$$\begin{aligned}
V_i &= G_i^{-1/2}(I + LG_i^{-1})^{-1}(-L - L^T)(I + G_i^{-1}L^T)^{-1}G_i^{-1/2}, && i = 1,\ 2,\\
\rho_i &= \max_{x \neq 0} \frac{x^T V_i x}{x^T x}, && i = 1,\ 2,\\
\gamma_1 &= (\sigma + (\sigma - 1)\rho_1)^{-1},\\
\gamma_2 &= \sigma + (\sigma - 1)\rho_2.
\end{aligned}$$

Let $x \neq 0$ be an n-vector. The quotients $x^T F_1 x / x^T F_2 x$ and $x^T F_2 x / x^T F_1 x$ have to be bounded above by constants. Note first that $0 < \alpha \leq \beta$ and therefore $\sigma \geq 1$. By using similarity transformations, it can be easily shown that for all n-vector z the inequalities $\alpha z^T G_2 z \leq z^T G_1 z \leq \beta z^T G_2 z$ implies the inequalities $\frac{1}{\beta} z^T G_2^{-1} z \leq z^T G_1^{-1} z \leq \frac{1}{\alpha} z^T G_2^{-1} z$. From the inequalities $z^T G_1 z \leq \beta z^T G_2 z$ and $z^T G_1^{-1} z \leq \frac{1}{\alpha} z^T G_2^{-1} z$ for all n-vector z, an upper bound for $x^T F_1 x / x^T F_2 x$ is obtained from

$$\begin{aligned}
\frac{x^T F_1 x}{x^T F_2 x} &= \frac{x^T (G_1 + L)G_1^{-1}(G_1 + L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&= \frac{x^T G_1 x + (L^T x)^T G_1^{-1}(L^T x) + x^T (L + L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&\leq \frac{\beta x^T G_2 x + \frac{1}{\alpha} x^T L G_2^{-1} L^T x + x^T (L + L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&\leq \frac{\sigma x^T G_2 x + \sigma x^T L G_2^{-1} L^T x + x^T (L + L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&= \frac{\sigma x^T (G_2 + L G_2^{-1} L^T + L + L^T)x + (1 - \sigma)x^T (L + L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&= \sigma + (\sigma - 1)\frac{x^T (-L - L^T)x}{x^T (G_2 + L)G_2^{-1}(G_2 + L^T)x}\\
&\leq \sigma + (\sigma - 1)\max_{y \neq 0} \left\{ \frac{y^T (-L - L^T)y}{y^T (G_2 + L)G_2^{-1}(G_2 + L^T)y} \right\}.
\end{aligned}$$

Let $z = G_2^{-\frac{1}{2}}(G_2 + L^T)y$. Then,

$$\frac{x^T F_1 x}{x^T F_2 x} \leq \sigma + (\sigma - 1)\max_{z \neq 0} \frac{z^T V_2 z}{z^T z} = \sigma + (\sigma - 1)\rho_2,$$

which gives the desired upper bound. It remains to find an upper bound for $x^T F_2 x / x^T F_1 x$. Similarly, from the inequalities $z^T G_2 z \leq \frac{1}{\alpha} z^T G_1 z$ and $z^T G_2^{-1} z \leq \beta z^T G_1^{-1} z$ for all n-vector z, an upper bound for $x^T F_2 x / x^T F_1 x$ can be derived. It is given by

$$\frac{x^T F_2 x}{x^T F_1 x} \leq \sigma + (\sigma - 1)\max_{z \neq 0} \frac{z^T V_1 z}{z^T z} = \sigma + (\sigma - 1)\rho_1.$$

Combining the last two inequalities completes the proof of the lemma. ∎

The spectral equivalence of Lemma 1 can also be achieved with upper bounds of the largest eigenvalues ρ_1 and ρ_2 of V_1 and V_2.

Let $L = L_{u,u} \leq 0$, $G_1 = D_{u,u}$, and $G_2 = \widetilde{D}_{u,u}$ and let α and β be the smallest and largest eigenvalues of $\widetilde{D}_{u,u}^{-1} D_{u,u}$, respectively. From Lemma 1, it follows that there exist constants $0 < \gamma_1 \leq 1 \leq \gamma_2$ such that for any vector x_u on the grid ω_u

$$\gamma_1 x_u^T S_{\text{IMC}} x_u \leq x_u^T S_{\text{IC}} x_u \leq \gamma_2 x_u^T S_{\text{IMC}} x_u. \tag{13}$$

Hence the Schur complements S_{IMC} and S_{IC} are spectrally equivalent.

LEMMA 2. *Let G_1, G_2, L, F_1, and F_2 be as in Lemma 1 and V_1 and V_2 be as defined above. If $W_i = G_i^{-1/2} V_i G_i^{+1/2}$ is irreducible, G_i is an M-matrix, $\| G_i^{-1} L \|_\infty < 1$, and $\| G_i^{-1} L^T \|_\infty < 1$ for $i = 1, 2$, then the matrices F_1 and F_2 are spectrally equivalent. Then there exist constants $0 < \gamma_1 \leq 1 \leq \gamma_2$ such that for any n-vector x*

$$\gamma_1 x^T F_2 x \leq x^T F_1 x \leq \gamma_2 x^T F_2 x. \tag{14}$$

Proof. First show that W_i is a nonnegative matrix, and then use the theory of nonnegative matrices to conclude the proof. Note that ρ_1 and ρ_2 are the largest eigenvalues of V_1 and V_2, respectively. Consequently, they are also the largest eigenvalues of W_1 and W_2.

Recall that $-L$ is a nonnegative matrix by assumption and G_i^{-1} is a nonnegative because G_i is a positive definite M-matrix by assumption. From the inequalities $G_i^{-1} \geq 0$, $-L \geq 0$, and $\| G_i^{-1} L \|_\infty < 1$, it follows that the Neumann expansion of $(I + G_i^{-1} L)^{-1}$ exists and is nonnegative. Therefore $(I + G_i^{-1} L)^{-1}$ is also nonnegative. Similarly it can also be shown that $(I + G_i^{-1} L^T)^{-1}$ is nonnegative. Note that W_i is the product of nonnegative matrices and is therefore a nonnegative matrix.

From the theory of nonnegative matrices it follows that ρ_i, the largest eigenvalue of W_i, coincides with the spectral radius of W_i which is smaller than or equal to in magnitude $\| W_i \|_\infty = \max_{1 \leq j \leq n} (W_i \mathbf{e})_j$, where $\mathbf{e}$ is the n-vector whose components are all 1's. Since $\| G_i^{-1} L \|_\infty < 1$ and $\| G_i^{-1} L^T \|_\infty < 1$, it follows from the Neumann expansions of $(I + G_i^{-1} L)^{-1}$ and $(I + G_i^{-1} L^T)^{-1}$ that

$$\| (I + G_i^{-1} L)^{-1} \|_\infty \leq \frac{1}{1 - \| G_i^{-1} L \|_\infty}$$

and

$$\| (I + G_i^{-1} L^T)^{-1} \|_\infty \leq \frac{1}{1 - \| G_i^{-1} L^T \|_\infty}.$$

Therefore

$$\begin{aligned}\rho_i &\le \| W_i \|_\infty \\ &\le \| (I + G_i^{-1}L)^{-1} \|_\infty \| G_i^{-1}(L + L^T) \|_\infty \| (I + G_i^{-1}L^T)^{-1} \|_\infty \\ &\le \frac{\| G_i^{-1}L \|_\infty + \| G_i^{-1}L^T \|_\infty}{(1 - \| G_i^{-1}L \|_\infty)(1 - \| G_i^{-1}L^T \|_\infty)}.\end{aligned}$$

From Lemma 1 and the above inequalities, it follows that for a n-vector $x \neq 0$

$$\begin{aligned}\frac{x^T F_2 x}{x^T F_1 x} &\le \sigma + (\sigma - 1)\rho_1 \le \sigma + (\sigma - 1) \| W_1 \|_\infty = \gamma_1, \\ \frac{x^T F_1 x}{x^T F_2 x} &\le \sigma + (\sigma - 1)\rho_2 \le \sigma + (\sigma - 1) \| W_2 \|_\infty = \gamma_2.\end{aligned}$$

Similarly, it follows that for a n-vector $x \neq 0$

$$\begin{aligned}\frac{x^T F_2 x}{x^T F_1 x} &\le \sigma + (\sigma - 1)\rho_1 \le \sigma + (\sigma - 1)\frac{\| G_1^{-1}L \|_\infty + \| G_1^{-1}L^T \|_\infty}{(1 - \| G_1^{-1}L \|_\infty)(1 - \| G_1^{-1}L^T \|_\infty)} = \gamma_1, \\ \frac{x^T F_1 x}{x^T F_2 x} &\le \sigma + (\sigma - 1)\rho_2 \le \sigma + (\sigma - 1)\frac{\| G_2^{-1}L \|_\infty + \| G_2^{-1}L^T \|_\infty}{(1 - \| G_2^{-1}L \|_\infty)(1 - \| G_2^{-1}L^T \|_\infty)} = \gamma_2.\end{aligned}$$

Combining the above inequalities completes the proof. ∎

The conditions required to bound the largest eigenvalues of W_1 and W_2 in Lemma 2 are not very severe. These conditions can easily be satisfied for matrices arising from the IC factorizations of M-matrices.

In practice, the diagonal matrices $D_{u,u}$ and $\widetilde{D}_{u,u}$ can be constructed such that the inequalities $-D_{u,u}^{-1}L_{u,u}^T\mathbf{e} < \mathbf{e}$, $-D_{u,u}^{-1}L_{u,u}\mathbf{e} < \mathbf{e}$, $-\widetilde{D}_{u,u}^{-1}L_{u,u}^T\mathbf{e} < \mathbf{e}$, and $-\widetilde{D}_{u,u}^{-1}L_{u,u}\mathbf{e} < \mathbf{e}$ hold [2, 3, 4, 7, 13, 20, 21]. Therefore Lemma 2 can be applied to S_{IMC} and S_{IC}.

The proof of Theorem 1 can now proceed. It establishes the spectral equivalence of the preconditioners Q_{IMC} and Q_{IC}.

Proof of Theorem 1. The proof of this theorem is similar to [18]. Let x be a vector on the composite grid $\omega = \omega_f \cup \omega_u$. Partition $x = \left[\begin{smallmatrix} x_f \\ x_u \end{smallmatrix}\right]$, where x_f is a vector in ω_f and x_u is a vector in ω_u. Hence

$$\begin{aligned}x^T Q_{\text{IC}} x &= x^T Q_{\text{IMC}} x + x^T (Q_{\text{IC}} - Q_{\text{IMC}}) x \\ &= x^T Q_{\text{IMC}} x + x_u^T (S_{\text{IC}} - S_{\text{IMC}}) x_u.\end{aligned}$$

From (13) it follows that $x_u^T S_{\text{IC}} x_u \le \gamma_2 x_u^T S_{\text{IMC}} x_u$. Hence

$$x^T Q_{\text{IC}} x \le x^T Q_{\text{IMC}} x + (\gamma_2 - 1) x_u^T S_{\text{IMC}} x_u.$$

Using Lemma 3, it follows that

$$x_u^T S_{\text{IMC}} x_u = \inf_{x_f} x^T Q_{\text{IMC}} x \le x^T Q_{\text{IMC}} x.$$

Finally,

$$x^T Q_{\text{IC}} x \le \gamma_2 x^T Q_{\text{IMC}} x.$$

Similarly,

$$x^T Q_{\text{IMC}} x = x^T Q_{\text{IC}} x + x_u^T (S_{\text{IMC}} - S_{\text{IC}}) x_u.$$

Again from (13), it follows that $x_u^T S_{\text{IMC}} x_u \leq \frac{1}{\gamma_1} x_u^T S_{\text{IC}} x_u$ yielding

$$x^T Q_{\text{IMC}} x \leq x^T Q_{\text{IC}} x + \left(\frac{1}{\gamma_1} - 1\right) x_u^T S_{IC} x_u.$$

Since Lemma 3 implies

$$x_u^T S_{\text{IC}} x_u = \inf_{x_f} x^T Q_{\text{IC}} x \leq x^T Q_{\text{IC}} x,$$

it can be concluded that

$$x^T Q_{\text{IMC}} x \leq \frac{1}{\gamma_1} x^T Q_{\text{IC}} x \text{ or } \gamma_1 x^T Q_{\text{IMC}} x \leq x^T Q_{\text{IC}} x. \qquad \square$$

The next two lemmas summarize properties of symmetric positive definite matrices that are used in the proof of Theorem 1; see, for instance, [18].

LEMMA 3. *Let* $C = \left[\begin{smallmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{smallmatrix}\right]$ *be a symmetric positive definite matrix and* $S_C = C_{2,2} - C_{2,1} C_{1,1}^{-1} C_{1,2}$ *be the Schur complement of* $C_{1,1}$ *in* C. *Then for any vector* $x = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$ *consistently partitioned with* C, *it follows that*

$$x_2^T S_C x_2 = \inf_{x_1} x^T C x.$$

LEMMA 4. *Let* E *and* F *be symmetric positive definite matrices of order* n. *Assume that there exist constants* $0 < \alpha \leq \beta$ *such that for any* n*-vector* $x \neq 0$

$$\alpha x^T E x \leq x^T F x \leq \beta x^T E x;$$

then the condition number of $E^{-1}F$ *satisfies*

$$\kappa(E^{-1}F) \leq \frac{\beta}{\alpha}.$$

4. Numerical experiments. The illustration of the potential of the method is done through the solution of some sample problems. First a two-level self-adjoint case is presented. Then a multilevel non-self-adjoint case is presented.

4.1. Two-level self-adjoint problem. The system of linear equations is generated by a 5-point cell-centered finite difference discretization of Poisson's equation:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(x,y) \quad \text{in} \quad \Omega,$$
$$u = 0 \quad \text{in} \quad \partial\Omega,$$

where $\Omega = [0,1] \times [0,1]$ is the unit square. The function $f(x,y)$ is chosen such that $u(x,y) = x(1-x)y(1-y)$ is the exact solution.

The system of linear equations is obtained using a cell-centered discretization of the PDE on the composite grid $\omega = \omega_f \cup \omega_u$. For the test, the region Ω is divided into two rectangular subregions $\Omega^{(f)} = [0,1] \times [0,s]$ and $\Omega \setminus \Omega^{(f)}$, with six different values of s in $(0,1)$. See Figure 1 for an illustration of the composite grid.

For each fixed value of s, a cell-centered coarse grid is introduced with uniform spacing h_c in Ω, and a cell-centered fine grid with uniform spacing $h_f = h_c/3$ in $\Omega^{(f)}$ is introduced for values 1/32 and 1/64 of h_c. The composite grid points are ordered using the natural ordering. The symmetric cell-centered approximation of [16, 17] is

TABLE 1
Comparing IC and IMC with point factorization.

$h_c = 1/32$				$h_c = 1/64$			
N	IC	IMC	C_γ	N	IC	IMC	C_γ
2945	48	47	5.35	12033	93	92	5.35
3937	56	56	5.35	16065	107	107	5.35
4929	62	62	5.34	20097	123	123	5.35
5921	70	70	5.28	24129	135	135	5.35
6913	73	73	4.93	28161	143	143	5.34
7905	73	73	3.29	32193	143	143	4.93

used because it yields a symmetric composite grid matrix A. The iterative solver is the PCG method [8].

The number of iterations and the CPU time spent in the PCG is reported. The numerical calculations were carried out in double precision arithmetic on a Sun workstation. Three preconditioners are tested—the IMC preconditioner Q_{IMC}, the IC preconditioner Q_{IC}, and the inexact BEPS preconditioner Q_{IBEPS}. The stopping criteria is $r_i^T r_i \leq 10^{-12} r_0^T r_0$, where $r_i = b - Ax_i$ is the ith residual vector and x_i is the ith approximation to the solution x. The initial guess is $x_0 = Q^{-1}b$, where Q is the preconditioning matrix for A. The preconditioners are the IC(0) [12, 13, 21] for point IC factorizations and the INV(1) and INV(2) [7, 9, 22] for block IC factorizations. Two different experiments were carried out.

The first experiment compares the performance of the IMC preconditioner against that of the IC preconditioner. The IC preconditioner used for this test is the IC(0) factorization of the composite grid matrix A. Further, the constant $C_\gamma = \gamma_2/\gamma_1$ of Theorem 2 is computed. The numerical results for the first experiment are presented in Table 1, which gives the number of iterations taken by the *PCG* with the IMC and IC preconditioner. The results show that both preconditioners require similar numbers of iterations. Further, it illustrates that the constant C_γ has a reasonable size and is independent of the grid size.

The second experiment compares the performance of the IMC preconditioner against that of the inexact BEPS preconditioner. Both point and block factorizations of the coarse and fine grid matrices are considered for the IMC and inexact BEPS preconditioners. Table 2 presents the results of the second set of experiments. In Table 2, h_c is the coarse grid spacing, N represents the number of unknowns in the linear system of equations, *itr* is the number of PCG iterations using the given preconditioning, and *time* gives the CPU time in seconds required by PCG with the given preconditioning.

The second experiment shows that for IC(0) factorizations, the PCG with the IMC and the PCG with the inexact BEPS preconditioners take similar numbers of iterations (see Table 2). The IMC preconditioner, however, requires less CPU time. The same observation is true for the inexact BEPS and the IMC preconditioners derived from INV(1) and INV(2) factorizations (see Table 2).

4.2. Non-self-adjoint multilevel systems. An extension of the IMC factorizations to non-self-adjoint multilevel problems can be achieved simply by replacing L^T with U in the appropriate places throughout section 2. The solution of more difficult sample problems further illustrates the potential of the method.

The system of linear equations is generated by a 5-point cell-centered finite dif-

TABLE 2
Comparing IMC and inexact BEPS with point and block factorizations.

	IC(0) factorization				INV(1) factorization				INV(2) factorization			
	IMC		IBEPS		IMC		IBEPS		IMC		IBEPS	
N	itr	time	itr	time	itr	time	itr	time	itr	time	itr	time
$h_c = 1/32$												
2945	47	3.7	47	4.9	18	1.2	18	1.5	14	0.9	14	1.1
3937	56	5.9	55	7.9	22	2.1	22	2.7	17	1.6	17	2.0
4929	62	8.4	62	11.3	26	3.2	26	4.0	19	2.2	19	2.8
5921	70	11.1	69	15.2	27	4.0	27	5.0	20	2.9	20	3.6
6913	73	13.5	73	18.8	31	5.6	31	7.0	22	3.8	22	4.9
7905	73	15.5	73	21.6	31	6.4	31	8.0	23	4.6	23	5.8
$h_c = 1/64$												
12033	92	30.1	92	39.0	37	11.4	37	13.6	27	8.1	27	9.7
16065	107	46.0	108	62.5	41	17.0	41	20.9	31	12.5	31	15.3
20097	123	66.7	123	90.1	50	26.4	50	32.5	38	19.5	38	24.2
24129	135	87.7	134	119.0	54	34.6	54	42.9	40	25.0	40	31.1
28161	143	107.2	143	153.7	61	46.9	61	58.6	45	33.9	45	42.2
32193	143	122.5	143	171.2	63	54.7	63	67.9	46	39.4	46	48.8

ference discretization of the non-self-adjoint PDE:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial y} - 100u\right) = f(x,y) \text{ in } \Omega,$$
$$u = g(x,y) \text{ in } \partial\Omega,$$

where $\Omega = [0,1]\times[0,1]$ is the unit square. The functions $f(x,y)$ and $g(x,y)$ are chosen such that $u(x,y) = x(1-x)(\exp(10(1-y))-1)$ is the exact solution.

The system of linear equations is obtained using a cell-centered discretization of the PDE on the composite grid. Up to four levels are used. The coarsest grid, $\omega^{(1)}$, is a 5×5 grid. The other grids are constructed recursively by refining a few blocks of the last fine grid. The grid at level k is obtained by refining the blocks of $\omega^{(k-1)}$ in $[1, N_{k-1}]\times[1, 2^{(k-1)}]$, where N_{k-1} is number of grid blocks in the x-direction in grid $\omega^{(k-1)}$.

A cell-centered coarse grid is introduced with uniform spacing $h_1 = 1/6$ in Ω. For the grid at level k, a cell-centered grid with uniform grid spacing $h_k = h_{k-1}/5$ is introduced. Hence, each coarse grid block is subdivided by 5 to obtain the fine grid blocks.

The symmetric cell-centered approximation of [17, 16] is used to approximate the symmetric part of the PDE. The first derivatives are approximated using upwinded difference approximations. The unknowns on each local grid were ordered using natural ordering which results in a block tridiagonal local grid matrix. The size of the submatrices of the grid matrix is equal to the number of grid blocks in the x-direction in the corresponding grid. The resulting composite grid matrix A is a nonsingular M-matrix for each problem. The linear system of equations resulting from the discretization of the model problem is solved using the preconditioned GMRES [23] and Bi-CGSTAB [24] methods. Two different preconditioners are considered—the IMC and the inexact BEPS preconditioners. Each grid matrix is a block tridiagonal matrix with block diagonal entries being tridiagonal matrices and block off-diagonal entries being square diagonal matrices. The block ILU factorization for the local grid matrices was used. The entries of the block diagonal matrix constructed from the block ILU factorization are tridiagonal.

TABLE 3
Non-self-adjoint multilevel problems—inexact BEPS and IMC.

		Inexact BEPS				IMC			
		GMRES(20)		Bi-CGSTAB		GMRES(20)		Bi-CGSTAB	
N	level	itr	time	itr	time	itr	time	itr	time
265	1	4	0.046	2	0.040	4	0.036	2	0.028
2665	2	10	1.165	5	0.89	10	1.08	5	0.76
26665	3	34	52.04	18	34.79	34	47.45	18	26.05
266665	4	131	2285.47	67	1398.28	131	2008.27	70	1125.61

The stopping criterion is $r_i^T r_i \leq 10^{-12} r_0^T r_0$ where $r_i = b - Ax_i$ is the ith residual vector and x_i is the ith approximation to the solution x. The initial guess is $x_0 = Q^{-1}b$, where Q is the preconditioning matrix for A.

The number of iterations and the CPU time spent in the preconditioned *GMRES* and Bi-CGSTAB are reported. The results gathered are presented in Table 3. The column labels of the tables are N for the number of unknowns in the composite grid problem, *level* for the number of levels, *itr* for the number of iterations taken by the iterative solver, and *time* for CPU the time spent in the solver. The numerical calculations were carried out in double precision arithmetic on a Sun workstation.

The experiments show that GMRES(20) took the same number of iterations with the IMC and the inexact BEPS. The preconditioned Bi-CGSTAB also took the same number of iterations with these preconditioners except for the largest problem, where the solver with the IMC needed a few extra iterations. The inexact BEPS needed more CPU time because it solves two local grid problems for each grid at each evaluation, except for the coarsest grid. The sparsity of the systems presented herein is very low, a five point star. When denser discretizations are used, the relative performance of IMC over BEPS should be even more accentuated.

The analysis of section 3 does not extend so simply to the non-self-adjoint case. The IMLU, a refinement of the IMC for non-self-adjoint multilevel problems, is introduced in [11, 19]. The IMLU is analyzed there for non-self-adjoint M-matrices. Extensive numerical results are also contained therein.

5. Conclusion. The construction of the IMC factorization requires the IC factorization for each grid level. The IMC is spectrally equivalent to the IC factorization of the assembled composite grid system. The IMC factorization has the familiar feel of an LL^T factorization consisting of a forward elimination pass followed by a backward substitution pass. Therefore, the IMC factorization is a natural extension of the IC factorization for systems arising from the discretization of PDEs on multilevel composite grids. The sparsity of the systems considered in section 4 is a five point star. The relative performance of IMC over BEPS should be even more sharp when denser discretizations are considered.

Other forms of inexact BEPS preconditioner exist where the local domain preconditioners are not IC factorizations. Whereas in some cases corresponding IMC factorizations exist, these are not discussed here. The extent of application of IMC goes beyond incomplete factorizations where the L of (1) coincides with the strictly lower part of A. These factorizations can be applied to each grid level resulting on an inexact BEPS which follows the same downward-upward cycle. So long as the preconditioning applied at each grid level has the form of an LL^T factorization, IMC is equally applicable. IMC can easily be incorporated with only minor changes in existing preconditioned iterative methods using inexact BEPS preconditioners with

approximate LL^T factorizations at each grid level.

The experiments illustrate that the IMC performs as well as the inexact BEPS preconditioner in terms of the number iterations required by the preconditioned iterative solvers. Its construction allows as much parallelism as the inexact BEPS. However, the application of the IMC factorization requires less computing time than the application of the inexact BEPS preconditioner. This leads to a better performance in CPU time for the overall computing time required by the preconditioned iterative solvers.

Acknowledgments. The authors would like to express their appreciation to the referees for their very helpful comments. Their very thorough reading and insightful comments helped the authors significantly improve the presentation. The authors would also like to thank J. L. Hensley, who patiently read several versions of this manuscript.

REFERENCES

[1] O. Axelsson, *A class of iterative methods for finite element equations*, Comput. Methods Appl. Mech. Engrg., 9 (1976), pp. 123–137.

[2] O. Axelsson, *Incomplete block matrix factorization preconditioning methods: The ultimate answer?*, J. Comput. Appl. Math., 12/13 (1985), pp. 3–18.

[3] O. Axelsson and S. Brinkkemper, *On some versions of incomplete block-matrix factorization iterative methods*, Linear Algebra Appl., 58 (1984), pp. 3–15.

[4] O. Axelsson and B. Polman, *On approximate factorization methods for block-matrices suitable for vector and parallel processors*, Linear Algebra Appl., 77 (1986), pp. 3–26.

[5] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[6] L. Chu, J. C. Díaz, M. Komara, and A. C. Reynolds, *Local Grid Refinement for Reservoir Simulation Applications*, Tech. report 98-3SC, University of Tulsa, Tulsa, OK, 1998.

[7] P. Concus, G. Golub, and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 220–252.

[8] P. Concus, G. H. Golub, and P. D. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309–332.

[9] P. Concus and G. Meurant, *On computing INV block preconditionings for the conjugate gradient method*, BIT, 26 (1986), pp. 493–504.

[10] J. C. Díaz, J. L. Hensley, and M. Komara, *Multilevel LU factorization for modeling multiphase contaminant transport*, in Proc. Second Internat. Petroleum Environmental Conference, K. L. Sublette, ed., New Orleans, LA, October 25–27, 1995, pp. 689–699.

[11] J. C. Díaz and M. Komara, *Incomplete Multilevel LU Factorizations*, Tech. report 98-5SC, University of Tulsa, Tulsa, OK, 1998.

[12] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers,* SIAM, Philadelphia, 1991.

[13] T. Dupont, R. Kendall, and H. Rachford, *An approximate procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.

[14] D. J. Evans, *Preconditioning Methods: Analysis and Applications,* Gordon and Breach Science, New York, 1983.

[15] R. E. Ewing, *Domain decomposition techniques for efficient adaptive local grid refinement*, in Domain Decomposition Methods, T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1989, pp. 192–206.

[16] R. E. Ewing and R. D. Lazarov, *Adaptive local grid refinement*, SPE 17806, in Proc. Rocky Mountain Regional Meeting, Casper, WY, May 11–13, 1988.

[17] R. E. Ewing, R. D. Lazarov, and P. S. Vassilevski, *Finite difference schemes on grids with local refinement in time and in space for parabolic problems,* I. *Derivation, stability and error analysis*, Computing, 45 (1990), pp. 193–215.

[18] R. E. Ewing, R. D. Lazarov, J. E. Pasciak, and P. S. Vassilevski, *Domain decomposition type iterative techniques for parabolic problems on locally refined grids*, SIAM J. Numer. Anal., 31 (1993), pp. 1537–1557.

[19] M. Komara, *Incomplete Multilevel LU Factorizations*, Ph.D. dissertation, Center for Parallel and Scientific Computing, University of Tulsa, Tulsa, OK, 1996.

[20] J. A. Meijerink and H. A. van der Vorst, *An iterative method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[21] J. A. Meijerink and H. A. van der Vorst, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*, J. Comput. Phys., 44 (1981), pp. 134–155.

[22] G. Meurant, *The block preconditioned conjugate gradient method on vector computers*, BIT, 24 (1984), pp. 623–633.

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

[24] H. A. van der Vorst, *Bi–CGSTAB: A fast and smoothly converging variant of Bi–CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 912–924

A STEPWISE APPROACH FOR THE GENERALIZED EIGENSTRUCTURE ASSIGNMENT PROBLEM*

GEORGE MIMINIS†

Abstract. The *DEscriptor Multi-input Eigenstructure Assignment* problem using *State* feedback (DEMESAS) is considered. It is pointed out, by referring to the relevant literature, that in many situations the *final step* of the DEMESAS is (or can be) the solution of the matrix equation

$$(A - BF)\,X = EXL \Longleftrightarrow BFX = AX - EXL \tag{0.1}$$

with respect to F. Little attention has been paid, up to now, on deriving a numerically sound algorithm for the computation of F. Consequently, a straightforward approach has often been used. We show that this approach has *numerical problems*, and we introduce a new approach for the solution of (0.1). We illustrate the reasons that make the new approach numerically better than the straightforward approach, and we present two numerical examples that support our claim.

Key words. eigenstructure assignment, descriptor systems, state feedback, complete controllability, robust control, singular pencils

AMS subject classifications. 93C05, 93C45, 93B40, 93B55, 93B52, 93B55, 65F05, 15A24, 15A22

PII. S0895479899298484

1. Preliminaries. Real and complex numbers will be represented by $\mathbb{R}$ and $\mathbb{C}$, respectively. Scalars will be represented by Greek letters, matrices by upper case Roman, while vectors and indices by lower case Roman. A superscript T will represent transposition, $\mathcal{R}(A)$, $\mathcal{N}(A)$ will denote the column and the null spaces of A, respectively, and $\lambda(A)$ will denote the set of eigenvalues of A. The zero and the identity matrices will be represented by O and I, respectively, the zero vector will be denoted by o, and e_i will denote the ith column of I. Finally $\chi(A) = \|A\| \left\|A^{-1}\right\|$ is the condition number of A with respect to inversion.

The paper is organized as follows. In the introduction the problem is defined and its significance with respect to other control problems is pointed out. In sections 3 and 4 we introduce algorithms for the solution of (0.1), based on a new stepwise approach and in section 5 we point out why the algorithms in sections 3 and 4 are numerically better than a straightforward algorithm that is commonly used for the solution of (0.1). In the same section we suggest some heuristics that improve the numerical properties of the algorithms in sections 3, 4 and we present a numerical example that supports our claims regarding the numerical properties of the new approach. We finish with our conclusion in section 6. The paper is a complete version of [19].

2. Introduction and literature discussion. Consider the continuous time-invariant *descriptor* system

$$E\dot{x}(t) = Ax(t) + Bu(t), \tag{2.1}$$

where $E \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$ is the *open-loop* system matrix, $B \in \mathbb{R}^{n\times m}$ is the *control influence* matrix, $x(t) \in \mathbb{R}^n$ is the *state* of the system at time t, and $u(t) \in \mathbb{R}^m$ is the

*Received by the editors July 20, 1999; accepted for publication (in revised form) by V. Mehrmann July 5, 2000; published electronically December 28, 2000. This work was supported by NSERC grant OGP0944.

http://www.siam.org/journals/simax/22-3/29848.html

†Department of Computer Science, Memorial University of Newfoundland, St. John's, NF, Canada, A1B 3X5 (miminis@cs.mun.ca, http://www.cs.mun.ca/~george)

input or *control* of the system. System (2.1) is said to be *completely controllable* if and only if

$$\{\forall\lambda \in \mathbb{C} \Longrightarrow \operatorname{rank}(B, A - \lambda E) = n\} \wedge \{\operatorname{rank}(B, E) = n\}.$$

For definitions on the controllability of descriptor systems, see, for example, [4]. An important problem in control theory is to guarantee the stability of (2.1) by choosing $u(t)$. One way to accomplish this is by using the *state feedback* $u(t) = -Fx(t)$, with $F \in \mathbb{R}^{m\times n}$, which gives the *closed-loop* system

$$E\dot{x}(t) = (A - BF)\, x(t). \tag{2.2}$$

It can be proven that when E is not singular, (2.2) is stable if all the eigenvalues of the pencil $[(A - BF), E]$ have negative real parts. The above discussion also applies to discrete time systems. The only difference is that all the eigenvalues of the corresponding pencil should be less than 1 in absolute value. A method that is concerned with placing the eigenvalues at the right points is that of *eigenvalue assignment*. According to this method, we are given a completely controllable system (E, A, B) and a self-conjugate set Λ of at most n scalars, an F may then be computed such that $\lambda[(A - BF), E] - \{-\infty, \infty\} \subseteq \Lambda$. This definition takes into consideration the possibility of a singular E and it guarantees that the resulting pencil $[(A - BF), E]$ is regular ($\lambda[(A - BF), E] \neq \mathbb{C}$). It may be shown that when $m > 1$ (multi-input case) there is no unique F that accomplishes eigenvalue assignment (actually F is unique if and only if $m = 1$ and E nonsingular). It appears that the freedom in the choice of F was first identified, for the case $E = I$, in [20], and it was associated with freedom in the selection of the corresponding eigenvectors (closed-loop eigenvectors) that were *implicitly* assigned along with the given eigenvalues. More elaborate discussions on the subject may be found in [11] for the case $E = I$ and in [12] for the case $E \neq I$, where it is shown that an eigenvector x_i is *feasible* (assignable) if and only if $x_i \in \mathcal{N}\left[P^T (A - \lambda_i E)\right]$, with P being orthonormal and $\mathcal{R}(P) = \mathcal{N}(B^T)$. Given now that normally the *plant* ("machinery" that is mathematically modeled by (2.1) and designed for a specific task) is known with some uncertainty, it is natural to ask how to use the freedom in the choice of F in order to design a control system that satisfies various stability and performance specifications in the face of plant uncertainty. This gives rise to the eigenstructure assignment problem. Actually the computation of a feedback that will satisfy various robustness criteria (stability and/or performance) is the central subject of *robust control*, where along with eigenstructure assignment, other methods have also been developed, like *linear quadratic regulator* (LQR) and *linear quadratic gaussian* (LQG) optimal control, H_∞ optimal control, *adaptive control*, etc. Among these methods, eigenstructure assignment is probably the simplest, and since it also appears to be fairly successful, it has naturally become the subject of extensive research, as well as the method of choice for a good number of applications. On the other hand, eigenstructure assignment has had its share of criticism (see, for example, [10], [14], [15]). The main criticism of this method is that it becomes ill conditioned as n increases and m decreases. The remedy for this, suggested, for example, in [10], is to solve a linear quadratic problem (LQ) instead and/or to place the eigenvalues in *regions* instead of simply *points*. Solid evidence, however, that the remedy will cure the problem has not been demonstrated to this point. Furthermore, research suggests that being able to choose eigenvalue locations and eigenvector shapes (easily accomplished by eigenstructure assignment) is necessary in certain applications. See, for example, [16], where eigenstructure assignment is used along with LQ to form a

hybrid method for the design of aircraft stability augmentation systems. Improving, therefore, algorithms related to eigenstructure assignment is clearly useful.

Next we give a brief account of a sample from the respective literature, including various robustness criteria that researchers have attempted to satisfy via eigenstructure assignment. From this discussion our motivation follows.

In some references, for example, [6], [7], [8], the eigenvalue assignment problem is solved via the eigenstructure assignment problem. This, however, is not advisable from a numerical point of view since eigenvalue assignment may be accomplished without the *explicit* assignment of a specific set of closed-loop eigenvectors. See [2], [17], [18], [21], [22] and for a counterexample, see [17]. In [20] the *output* $y(t) = Cx(t)$ (with $C \in \mathbb{R}^{p\times n}$) of the system is also considered and the closed-loop eigenvectors are chosen so that a desired distribution of the *modes* ($e^{\lambda_i t}$) among the components of the output $y(t)$, is achieved. To see this, let (λ_i, x_i) be a closed-loop eigenpair and z_i the corresponding left closed-loop eigenvector, then the output vector may be given by

$$y(t) = \sum_{i=1}^{n} Cx_i \left(z_i^T x(0)\right) e^{\lambda_i t}.$$

If now x_i is chosen such that, for example, $Cx_i = (2, 1, 0, \ldots, 0)^T$, the ith mode will appear in the first two components of $y(t)$ and it will be twice as large in the first component than in the second. In [13] the assignment of principal closed-loop eigenvectors is considered. [20], [13] are mathematical treatments of the subject and along with [25] include a considerable number of interesting results that were actually rediscovered more recently. In [5] a parametric approach to the eigenstructure assignment problem is proposed and good numerical properties are claimed for the algorithms within; however, no evidence of the latter is given. In [23] a set of desired, but not necessarily feasible, closed-loop eigenvectors are given along with the closed-loop eigenvalues. Since, however, the desired eigenvectors may not be feasible, a number of least squares problems are solved so that the "closest" feasible eigenvectors to the corresponding desired eigenvectors may be found. In [24] the freedom in the choice of F is used to minimize the index

$$J = \sum_{i=1}^{n} \omega_i \left\| Fe_i \right\|_2^2,$$

with ω_i being desired "weights." It is interesting to observe that the weights may be chosen so that the state feedback will effectively result into a specific output feedback ($u = -Ky$ for some matrix K). To see this, consider the example given in [24], where $n = 19$, and choose $\omega_i = 1$ for $i \in \{1, 2, 7, 12, 14\}$ and $\omega_i = 100$ for the remaining i, so that only columns 1, 2, 7, 12, 14 of F are significant. Then with $y(t) = Cx(t)$, where $C = (e_1, e_2, e_7, e_{12}, e_{14})^T$ and $K = FC^T$, we have

$$u = -Fx \approx -Ky.$$

In [1] the freedom in the choice of the closed-loop eigenvectors is used for the optimization of the following problem:

$$\left\{ \begin{array}{ll} \min\limits_{u=-Fx} \displaystyle\int_{t_0}^{\infty} \left(x^T Q x + u^T R u\right) dt \\ \text{subject to} \qquad \lambda(A - BF) = \Lambda \end{array} \right\},$$

where Q is symmetric positive semidefinite and R is symmetric positive definite. By far, however, the most extensively studied robustness criterion is that of optimizing the condition number of the eigenproblem of the pencil $[(A - BF), E]$. See, for example, [3], [11], [12], [26], [27].

In general, when the freedom in the choice of the eigenvectors is to be used in order to solve a control problem beyond simply eigenvalue assignment, the following "two-step process" may be considered:

(i) A suitable set of eigenvectors is computed so that the control problem beyond eigenvalue assignability is solved.
(ii) These eigenvectors along with the corresponding given eigenvalues are assigned.

Although at first glance it may not be obvious that this process may be used in all of the problems described in the above references, often the algorithms within can be customized to suit the "two-step process." In this paper we will *only* be interested in the *second* step of the "two-step process"; the reason is twofold. Initially the second step may be common to a large number of applications despite which robustness criterion the first step attempts to satisfy. Therefore a numerically sound solution of this step may be welcome. The lack of such a numerical solution provides the second reason. The efficient numerical solution of (0.1) has not received much attention due to the fact that at first glance, it does not appear to hide any surprises. Thus far the methods employed for its solution are variations of the following three-step straightforward approach.

(i) Compute $G = AX - EXL$.
(ii) Solve the system $BY = G$ with respect to Y.
(iii) Solve $FX = Y$ with respect to F.

Clearly, the above process makes the accuracy of F depend on the condition numbers of B and X. Also, if E is singular, the above approach proceeds in such a way that the resulting pencil $[(A - BF), E]$ is always singular, even if the system is completely controllable, where theory dictates that singularity of the closed-loop pencil can be avoided. This point will become obvious in section 3 from (3.8), (3.10). To overcome this problem, the following two-step "fix" has been recommended in [8].

(i) Compute an orthonormal matrix N that spans the null space of E. Then compute a matrix D such that $E + ANN^T + BDN^T$ is nonsingular.
(ii) Solve $BF(X, N) = (AX - EXL, D)$ with respect to F.

In the next section we will present an algorithm that computes F in a stepwise approach. The algorithm assigns one real eigenpair at a time in a single step or one complex conjugate eigenpair at a time in a double step. The stepwise approach is advantageous over the above three-step approach accompanied by the two-step fix, in two points. First, when it is possible, the stepwise approach will make the task of producing a regular pencil $[(A - BF), E]$ straightforward. Second, the accuracy of the computed F will *not* depend on the condition number of the entire X.

3. Initial reduction and a stepwise algorithm for the DEMESAS. Our algorithm computes F such that

$$(A - BF)X = EXL, \tag{3.1}$$

is satisfied, where L is a diagonal matrix with the desired eigenvalues on its diagonal and X an $n \times n$ matrix with the corresponding feasible eigenvectors on its columns; $\operatorname{rank}(B) = m$ will be assumed throughout, if this does not initially hold it can easily be arranged. The algorithm begins by separating the uncontrollable part of the system

(E, A, B) from the completely controllable part. It then assigns a desirable set of eigenpairs to the resulting completely controllable system. The decoupling of the uncontrollable part is accomplished by an algorithm presented in [18]. According to this algorithm, orthogonal matrices U, V, and W are computed such that

$$U^T A V = \left(\begin{array}{c|c} A_1 & \widehat{A} \\ \hline O & \widetilde{A} \end{array}\right), U^T E V = \left(\begin{array}{c|c} E_1 & \widehat{E} \\ \hline O & \widetilde{E} \end{array}\right), U^T B W = \left(\begin{array}{c} B_1 \\ \hline O \end{array}\right),$$

where all the uncontrollable eigenvalues have been accumulated in $(\widetilde{A}, \widetilde{E})$. It is rather straightforward to show that, if X consists of a set of feasible eigenvectors, then

$$V^T X = \left(\begin{array}{c|c} X_1 & \widehat{X} \\ \hline O & \widetilde{X} \end{array}\right),$$

$$\text{with } \widetilde{X} \begin{cases} = O \text{ if } \lambda(\widetilde{A}, \widetilde{E}) \text{ does not include desired eigenvalue,} \\ \neq O \text{ if } \lambda(\widetilde{A}, \widetilde{E}) \text{ includes desired eigenvalue(s).} \end{cases}$$

If we take now $(F_1, \widetilde{F}) = W^T F V$ and $L = \left(\begin{array}{c|c} L_1 & O \\ \hline O & \widetilde{L} \end{array}\right)$, (3.1) is equivalent to

$$\left(\begin{array}{c|c} (A_1 - B_1 F_1) X_1 & (A_1 - B_1 F_1)\widehat{X} + \left(\widehat{A} - B_1 \widetilde{F}\right)\widetilde{X} \\ \hline O & \widetilde{A}\widetilde{X} \end{array}\right)$$

$$= \left(\begin{array}{c|c} E_1 X_1 L_1 & E_1 \widehat{X} \widetilde{L} + \widehat{E}\widetilde{X}\widetilde{L} \\ \hline O & \widetilde{E}\widetilde{X}\widetilde{L} \end{array}\right). \tag{3.2}$$

If $\widetilde{X} = O$, we need only to solve

$$(A_1 - B_1 F_1) X_1 = E_1 X_1 L_1 \tag{3.3}$$

for F_1, and since the desired set of eigenvectors is feasible, (3.2) will be compatible. In this case $\widetilde{F}$ may assume any convenient value. If, however, $\widetilde{X} \neq O$, meaning that some of the uncontrollable eigenvalues are also desired, we first solve (3.3) and then

$$(A_1 - B_1 F_1)\widehat{X} + \left(\widehat{A} - B_1 \widetilde{F}\right)\widetilde{X} = E_1 \widehat{X}\widetilde{L} + \widehat{E}\widetilde{X}\widetilde{L} \tag{3.4}$$

with respect to $\widetilde{F}$. Since the desired eigenvectors are feasible, (3.2) will be compatible. The latter means that we may change (assign) the eigenvector of an uncontrollable eigenvalue so long as the desired eigenvector is feasible. This is of course a well-known result (see, for example, [20], [25]); the difference here is that it appears in an algorithmic way which facilitates its computer solution. We will consider the solution of (3.4) when an algorithm for (3.3) has been developed.

We should now observe that if the pencil $(\widetilde{A}, \widetilde{E})$ is singular, so is $[(A - BF), E]$ for every F_1. If, however, $(\widetilde{A}, \widetilde{E})$ is regular but (A_1, E_1) is singular, (3.3) suggests that there may be a way to change this property in $[(A_1 - B_1 F_1), E_1]$ by choosing the right F_1, and eventually produce a regular pencil $[(A - BF), E]$. In the subsequent discussion we will see that this is possible, and we will show how it can be accomplished.

The problem of assigning eigenpairs to a completely controllable system (E_1, A_1, B_1) will occupy us immediately, hence its formal definition follows.

PROBLEM 3.1. *Given a completely controllable system* $(E_1, A_1, B_1) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$, *with* $\operatorname{rank}(B_1) = m$, *and a desired self-conjugate set of eigenvalues* $\{\lambda_1, \ldots, \lambda_r\}$, *with* $r = \operatorname{rank}(E_1)$ *and also given a feasible set of corresponding eigenvectors* $\{x_1, \ldots, x_r\}$, *that is,* $x_i \in \mathcal{N}\left[P^T (A - \lambda_i E)\right]$, *with* P *being orthonormal and* $\mathcal{R}(P) = \mathcal{N}(B^T)$, *an* $m \times n$ *matrix* F_1 *must be computed, so that*

$$(A_1 - B_1 F_1) X_1 = E_1 X_1 L_1,$$

where $L_1 = \operatorname{diag}(\lambda_1, \ldots, \lambda_r, \underbrace{\infty, \ldots, \infty}_{n-r})$, $X_1 = (x_1, \ldots, x_r, X_\infty)$ *with* $X_\infty \in \mathbb{R}^{n\times n-r}$ *and such that* X_1 *is nonsingular. Furthermore* $[(A_1 - B_1 F_1), E_1]$ *should form a regular pencil.*

The following two observations will be implemented in the algorithm. First, we see that if complex eigenpairs are to be assigned, (3.3) may produce a complex F_1. Since, however, such an F_1 may not be useful in practice, we can slightly change the problem in order to address this important issue. To accomplish this, we will first assume that complex conjugate eigenpairs appear successively in L_1 and X_1. Furthermore, if $\mu_1 \pm i\nu_1$ are two desired complex conjugate eigenvalues, they should appear on the diagonal of L_1 as a 2×2 block $\begin{pmatrix} \mu_1 & \nu_1 \\ -\nu_1 & \mu_1 \end{pmatrix}$. If also $x_1 \pm ix_2$ are the corresponding desired eigenvectors, the two vectors x_1, x_2 should appear in the respective columns of X_1 instead. The latter is justifiable since $\mathcal{R}(x_1, x_2) = \mathcal{R}(x_1 + ix_2, x_1 - ix_2)$. The second observation is that assigning eigenvalues of multiplicity greater than m to $[(A_1 - B_1 F_1), E_1]$ will make these eigenvalues become defective. See [11], [25] for the $E = I$ case and [12] for the $E \neq I$ case. It is well known, however, that defective eigenvalues are sensitive to perturbations in the data, therefore unless it is absolutely necessary, we should avoid multiplicities greater than m. Next we present a stepwise algorithm for the solution of (3.3) that takes into consideration the two restrictions we just described. For simplicity B_1 will be assumed to be $n \times m$ and E_1, A_1, X_1 $n \times n$ each, E_1 will be allowed to be singular.

Let $\widetilde{V}$ and $\widetilde{U}$ be the orthogonal matrices of the QR decompositions of X_1 and $E_1\widetilde{V}$, respectively. Consider also the following partitioning:

$$X_1 \equiv \widetilde{V}^T X_1 = \left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline O & X_3 \end{array}\right), \ \widetilde{U}^T \left(E_1 \widetilde{V}\right) = \left(\begin{array}{c|c} E_{11} & E_{12} \\ \hline O & E_3 \end{array}\right), \tag{3.5}$$

$$\widetilde{U}^T A_1 \widetilde{V} = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline & A_3 \end{array}\right), \ B_1 \equiv \widetilde{U}^T B_1 = \left(\begin{array}{c} B_{11} \\ \hline B_3 \end{array}\right), \ L_1 = \left(\begin{array}{c|c} L_{11} & O \\ \hline O & L_3 \end{array}\right), \tag{3.6}$$

with A_{11} and B_{11} being $n \times 2$ and $2 \times m$, respectively, and

$$X_{11} = \begin{pmatrix} \xi_{11} & \xi_{12} \\ 0 & \xi_{22} \end{pmatrix}, \ E_{11} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} \\ 0 & \varepsilon_{22} \end{pmatrix}.$$

The algorithm may then attempt to assign the eigenpairs $(\mu_1 \pm i\nu_1, x_1 \pm ix_2)$ by computing the $m \times 2$ matrix F_{11} in $F_1\widetilde{V} = (F_{11}, F_3)$ as follows:

$$\begin{aligned} &(A_1 - B_1 F_1) X_1 = E_1 X_1 L_1 \\ \iff &\widetilde{U}^T (A_1 - B_1 F_1) \widetilde{V}\widetilde{V}^T X_1 = \widetilde{U}^T E_1 \widetilde{V}\widetilde{V}^T X_1 L_1 \\ \iff &\left(\begin{array}{c|c} (A_{11} - B_1 F_{11}) X_{11} & \times \\ \hline & (A_3 - B_3 F_3) X_3 \end{array}\right) = \left(\begin{array}{c|c} E_{11} X_{11} L_{11} & \times \\ \hline O & E_3 X_3 L_3 \end{array}\right). \end{aligned} \tag{3.7}$$

From the first two columns of (3.7) we derive the equation

$$B_1 F_{11} X_{11} = A_{11} X_{11} - \left(\frac{H}{O} \right), \tag{3.8}$$

where if E_{11} is nonsingular, we take $H = E_{11} X_{11} L_{11}$ and by solving (3.8) with respect to F_{11} the complex eigenpair $(\mu_1 \pm i\nu_1, x_1 \pm i x_2)$ is assigned. If, however, E_{11} is singular, the assignment cannot proceed. In this case we make certain that the computed F_{11} will not cause singularity of the pencil $[(A_1 - B_1 F_1), E_1]$. This may be accomplished by taking H in (3.8) to be any nonsingular 2×2 matrix. To avoid unnecessary numerical problems we may choose $\|H\|$ comparable to the magnitude of our data. If, alternatively, a real eigenpair (λ_1, x_1) is to be assigned,

$$B_1 f_{11} \xi_{11} = a_{11} \xi_{11} - \begin{pmatrix} \eta \\ o \end{pmatrix} \tag{3.9}$$

needs to be solved instead of (3.8), where f_{11}, a_{11} are $m \times 1$ and $n \times 1$, respectively. If now $\varepsilon_{11} \neq 0$, we take $\eta = \varepsilon_{11} \xi_{11} \lambda_1$ in (3.9) and solve

$$B_1 f_{11} = a_{11} - \begin{pmatrix} \varepsilon_{11} \lambda_1 \\ o \end{pmatrix} \tag{3.10}$$

with respect to f_{11}. If, however, $\varepsilon_{11} = 0$, then (λ_1, x_1) cannot be assigned and as with the complex case f_{11} should be chosen so as to avoid singularity in $[(A_1 - B_1 F_1), E_1]$. This may be achieved by taking $\eta \neq 0$ and solving

$$B_1 f_{11} = a_{11} - \begin{pmatrix} \eta \\ o \end{pmatrix}. \tag{3.11}$$

In this case too, $|\eta|$ must be comparable to the magnitude of our data. Once (3.8) or (3.9) is solved, (3.7) takes the form

$$\left(\begin{array}{c|c} H & \times \\ \hline O & (A_3 - B_3 F_3) X_3 \end{array} \right) = \left(\begin{array}{c|c} E_{11} X_{11} L_{11} & \times \\ \hline O & E_3 X_3 L_3 \end{array} \right). \tag{3.12}$$

It is now clear why a nonsingular H (or a nonzero η) will have the desired effect regarding the regularity of the closed-loop pencil $[(A_1 - B_1 F_1), E_1]$. Note here that the straightforward approach would not have been capable of choosing a nonzero η or a nonsingular H, with obvious results. The algorithm continues in a similar manner with the next assignment by considering the equation $(A_3 - B_3 F_3) X_3 = E_3 X_3 L_3$.

We may now observe that after each assignment, the number of states, say n, of the next system (for example, (E_3, A_3, B_3) above) becomes one or two less than the number of states of its predecessor system (for example, (E_1, A_1, B_1) above), depending on whether a real or a complex conjugate eigenpair was assigned, respectively. Since at the same time the number of inputs m remains the same, the algorithm reaches one of the following two stages.

- $n = m$,
- $n = m + 1$ and only complex conjugate eigenpairs to be assigned.

At this point (3.8) or (3.9) are not adequate to determine all the elements of F_1, therefore a modification of the above process needs to be introduced which is described next.

Assume that $(\mu_1 \pm \nu_1, x_1 \pm ix_2)$ is to be assigned, with $n = m$. The same partitioning as in (3.5), (3.6) will be considered, only that the following partitioning of a QR-like decomposition of $\widetilde{U}^T B_1$ will be employed:

$$\widetilde{U}^T B_1 \widetilde{W} = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline O & B_3 \end{array}\right) \text{ as well as } \widetilde{W}^T F_1 \widetilde{V} = \left(\begin{array}{c|c} F_{11} & F_{12} \\ \hline & F_3 \end{array}\right).$$

In this case B_{11} is a 2×2 upper triangular matrix, and F_{12} is $2 \times (n-2)$. Once F_{11} is computed by solving (3.8) the following equation may be formed:

$$\begin{aligned} &\left(\begin{array}{c|c} H & HX_{11}^{-1}X_{12} + (A_{12} - B_{11}F_{12} - B_{12}F_3)\, X_3 \\ \hline O & (A_3 - B_3F_3)\, X_3 \end{array}\right) \\ &\qquad = \left(\begin{array}{c|c} E_{11}X_{11}L_{11} & (E_{11}X_{12} + E_{12}X_3)\, L_3 \\ \hline O & E_3X_3L_3 \end{array}\right), \end{aligned} \tag{3.13}$$

where F_{12} and F_3 still need to be computed. It is apparent from (3.13) that F_3 should be computed first and then F_{12} may be computed from the equation

$$B_{11}F_{12} = \left[HX_{11}^{-1}X_{12} - (E_{11}X_{12} + E_{12}X_3)\, L_3\right] X_3^{-1} + A_{12} - B_{12}F_3. \tag{3.14}$$

The computation of F_3 will also involve steps like the above. Therefore, in this case the algorithm has in fact two parts; the forward part, where for all i, F_{ii} or f_{ii} is computed by solving equations of the kind (3.8) or (3.9) and the backward part, where $F_{i,i+1}$ or $f_{i,i+1}$ is computed by solving equations of the kind (3.14) (F_{12} in our presentation).

Once F_1 has been computed, F may be obtained by applying the history of orthogonal transformations. To see this, assume for the sake of presentation that the process begins with $n > m$, and after a specific number of steps we reach F_r for some r, with F_r being square. Assume also that throughout the process only double steps were applied. We may now compute $\widetilde{F}_r = \widetilde{W}(F_{rr}, \ldots, F_{n-1,n-1})$ and F is finally obtained by

$$F = W\left[\left(F_{11}, F_{33}, \ldots, F_{r-2,r-2}, \widetilde{F}_r\right)\widetilde{V}^T, \widetilde{F}\right] V^T.$$

Recall that $\widetilde{F}$ originates at (3.2) where the separation of the uncontrollable part of the system took place. In the case $\widetilde{X} = O$, $\widetilde{F}$ can assume any convenient value, it may, for example, become zero. If, however, $\widetilde{X} \neq O$, (3.4) must be solved with respect to $\widetilde{F}$. This may be accomplished in a way similar to that of solving (3.3) and we briefly describe it next. A useful observation is that $\widetilde{X}$ may have zero columns which correspond to uncontrollable eigenvalues that are not desired. In view of this, let $\widetilde{X} = P\left(\begin{array}{c|c} O & \widetilde{X}_1 \\ \hline O & O \end{array}\right)$ be the QR decomposition of $\widetilde{X}$ and consider the partitioning $\widetilde{F}P = (\widetilde{F}_1, \widetilde{F}_2)$ as well as $\widetilde{L} = \left(\begin{array}{c|c} \times & O \\ \hline O & \widetilde{L}_1 \end{array}\right)$. Equation (3.4) may then be written as

$$\begin{aligned} B_1\widetilde{F}\widetilde{X} = \widehat{A}\widetilde{X} - \widehat{E}\widetilde{X}\widetilde{L} + \underbrace{\left[(A_1 - B_1F_1)\,\widehat{X} - E_1\widehat{X}\widetilde{L}\right]}_{D} \\ \Longleftrightarrow B_1\left(\widetilde{F}P\right)\left(P^T\widetilde{X}\right) = \left(\widehat{A}P\right)\left(P^T\widetilde{X}\right) - \left(\widehat{E}P\right)\left(P^T\widetilde{X}\right)\widetilde{L} + D \\ \Longleftrightarrow B_1\widetilde{F}_1\widetilde{X}_1 = \widehat{A}_1\widetilde{X}_1 - \widehat{E}_1\widetilde{X}_1\widetilde{L}_1 + D_1, \end{aligned} \tag{3.15}$$

where $\widehat{A}_1$, $\widehat{E}_1$ are the relevant parts of $\widehat{A}P$, $\widehat{E}P$, respectively, and D_1 is the relevant part of D. Once $\widetilde{F}_1$ is computed from (3.15) and $\widetilde{F}_2$ is given any desired value, $\widetilde{F}$ may be computed as $\widetilde{F} = (\widetilde{F}_1, \widetilde{F}_2)P^T$. It remains now to solve (3.15) with respect to $\widetilde{F}_1$. The fundamental difference between (3.15) and (3.3) is the factor D_1. Therefore a similar method may be employed, with the relevant columns of D_1 being included as additional terms in the corresponding equations.

4. The bottom-up algorithm. The development of the stepwise approach in the last section facilitates the presentation of a more efficient algorithm, which we term *bottom-up*. Consider the general case $n > m+1$ and assume for simplicity that only double steps have been performed (assume, for the sake of presentation, $\lceil \frac{n-m}{2} \rceil$ such steps), until $\tilde{n}$ and $\tilde{m}$, with $[\tilde{n}, \tilde{m}] = \text{size}\,(B_k)$, satisfy $\tilde{n} = \tilde{m}$ or $\tilde{n} = \tilde{m}+1$. Equation $(A_1 - B_1F_1)X_1 = E_1X_1L_1$ may then be partitioned as follows:

$$\left(\begin{array}{c|c|c|c|c} (A_{11} - B_1F_{11})\,X_{11} & \multicolumn{4}{c}{\times} \\ \hline & (A_{33} - B_3F_{33})\,X_{33} & \multicolumn{3}{c}{\times} \\ \hline & & \cdots & \multicolumn{2}{c}{\vdots} \\ \hline & & & \left(A_{k-2,k-2} - B_{k-2}F_{k-2,k-2}\right)X_{k-2,k-2} & \times \\ \hline & & & & \left(A_k - B_kF_k\right)X_k \end{array}\right)$$

$$= \left(\begin{array}{c|c|c|c|c} E_{11}X_{11}L_{11} & \multicolumn{4}{c}{\times} \\ \hline & E_{33}X_{33}L_{33} & \multicolumn{3}{c}{\times} \\ \hline & & \ddots & \multicolumn{2}{c}{\vdots} \\ \hline & & & E_{k-2,k-2}X_{k-2,k-2}L_{k-2,k-2} & \times \\ \hline & & & & E_kX_kL_k \end{array}\right), \tag{4.1}$$

where $k = \begin{cases} n-m+1 \text{ if } n-m \text{ is even, in this case } \tilde{n} = \tilde{m}, \\ n-m \qquad \text{if } n-m \text{ is odd, in this case } \tilde{n} = \tilde{m}+1. \end{cases}$

From (4.1) we see that equations of the type (3.8) and (3.9) can be solved independently and thus in any order (even in parallel). Similarly, if the case $n = m$ or $n = m+1$ is considered, $\lceil \frac{n}{2} \rceil$ double steps will take place and equation $(A_1 - B_1F_1)\,X_1 = E_1X_1L_1$ may be partitioned as follows:

$$\left(\begin{array}{c|c|c|c} (A_{11} - B_1F_{11})\,X_{11} & \multicolumn{3}{c}{HX_{11}^{-1}X_{12} + (A_{12} - B_{11}F_{12} - B_{12}F_3)\,X_3} \\ \hline & \multicolumn{3}{c}{HX_{33}^{-1}X_{34} + (A_{34} - B_{33}F_{34} - B_{34}F_5)\,X_5} \\ \hline & (A_{33} - B_3F_{33})\,X_{33} & \multicolumn{2}{c}{\vdots} \\ \hline & & \cdots & (A_q - B_qF_q)\,X_q \end{array}\right)$$

$$= \left(\begin{array}{c|c|c|c} E_{11}X_{11}L_{11} & \multicolumn{3}{c}{(E_{11}X_{12} + E_{12}X_3)\,L_3} \\ \hline & \multicolumn{3}{c}{E_{33}X_{33}L_{33} \quad (E_{33}X_{34} + E_{34}X_5)\,L_5} \\ \hline & & \ddots & \vdots \\ \hline & & & E_qX_qL_q \end{array}\right), \tag{4.2}$$

with $q = \begin{cases} n \qquad \text{if } n \text{ is odd}, \\ n-1 \text{ if } n \text{ is even}. \end{cases}$

The inherent flexibility of the algorithm regarding the order with which equations of the kind (3.8) and (3.9) can be solved (even in parallel) can be efficiently exploited by adopting a *bottom-up* approach. According to this approach $(A_q - B_qF_q)\,X_q = E_qX_qL_q$ is solved first, moving towards $(A_{11} - B_1F_{11})\,X_{11} = E_{11}X_{11}L_{11}$. In this way the QR decomposition of B_1, which is useful for the solution of (3.8) and (3.9), needs

to be computed only once and gradually. For example, suppose B_1 is a 7×3 matrix. Initially the QR decomposition of the 3×3 bottom part of B_1 is computed as

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \hline \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \equiv \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \hline B_k \end{pmatrix}.$$

The triangular form of B_k may then be used in (4.1) or (4.2) for the solution of equations of the type (3.8), (3.9), and (3.14) in order to eventually compute F_k in $(A_k - B_k F_k) X_k = E_k X_k L_k$. Assuming now that the algorithm continues with a double step, reduce to

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \hline \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \longrightarrow \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \hline \times & \times & \times \\ & \times & \times \\ & & \times \\ \hline O \end{pmatrix} \equiv \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \hline B_{k-2} \\ \hline O \end{pmatrix},$$

compute $F_{k-2,k-2}$, and proceed in a similar way until the computation of F_{11} takes place. Finally the computation of F is performed in a way similar to that described in section 3.

5. Discussion of the algorithm, heuristics, and numerical examples. The advantage of the algorithms presented in sections 3 and 4, over the three-step approach accompanied by the two-step fix, is twofold. First, when it is possible, the stepwise approach makes the task of producing a regular pencil $[(A_1 - B_1 F_1), E_1]$ straightforward. See (3.8), (3.11), (3.12). Second, it is apparent from (3.8), (3.10), (3.11), and (3.14) that the computed F depends on $\chi(B_1)$, the condition numbers of certain 2×2 diagonal blocks of X_1, as well as the condition number of the $m \times m$ south-east block of X_1, whereas the straightforward approach depends on $\chi(B_1)$ and the condition number of the *entire* X_1. This may be visualized by the following example where we take $n = 9$, $m = 3$, and a desired set of eigenvalues denoted by $\{c, \bar{c}, r, r, c, \bar{c}, \times, \times, \times\}$. The notation $(c, \bar{c})$ stands for a complex conjugate pair of eigenvalues, r for a real eigenvalue, and $\times$ any eigenvalue. With X_1 given as

$$X_1 = \begin{pmatrix} \bullet & \bullet & \times & \times & \times & \times & \times & \times & \times \\ & \bullet & \times & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times & \times & \times \\ & & & & \bullet & \bullet & \times & \times & \times \\ & & & & & \bullet & \times & \times & \times \\ & & & & & & \bullet & \bullet & \bullet \\ & & & & & & & \bullet & \bullet \\ & & & & & & & & \bullet \end{pmatrix},$$

the computed F_1 will depend on the condition numbers of only those blocks with $\bullet$ elements, as well as $\chi(B_1)$.

A few heuristics will now be presented that will attempt to make the relevant blocks of X_1 well conditioned, even if X_1 is not. In view of this, when computing the QR decomposition of X_1, an attempt should be made to hide any possible closeness of X_1 to rank deficiency. If this is impossible, an attempt should be made to associate any relatively small diagonal elements of X_1 with real eigenpairs, so that they are cancelled out (see (3.10)). To this end, the desired QR decomposition of X_1 may be computed in two-stages as follows.

In the first stage the QR decomposition of X_1 with *minimum* norm column pivoting is computed. Unlike the maximum norm column pivoting which can be used in revealing the rank of a matrix (see, for example, [9, p. 248]), the minimum norm column pivoting tends to have the opposite effect. Consider, for example, the well-known $n \times n$ matrix

$$X_1 = \begin{pmatrix} 1 & -\gamma & -\gamma & \cdots & -\gamma \\ & \sigma & -\gamma & \cdots & -\gamma \\ & & \sigma^2 & \cdots & -\gamma \\ & & & \ddots & \vdots \\ & & & & \sigma^{n-1} \end{pmatrix}, \tag{5.1}$$

where $\gamma^2 + \sigma^2 = 1$. If we take $n = 10$ and $\gamma = 0.7$, then $\chi(X_1) \approx 10^7$. QR with minimum norm column pivoting, however, does not change X_1 and since, for the given case, the smallest diagonal element of X_1 is $\sigma^{n-1} \approx 0.048$, this example appears to bring our point forward. Note that maximum norm column pivoting produces a triangular matrix with smallest diagonal element $\approx 10^{-7}$. Since column pivoting is required here, it is worth pointing out that those pairs of columns in X_1 that span 2-dimensional eigenspaces corresponding to complex conjugate eigenvalues should remain in consecutive positions so that our algorithms work properly. Therefore, if such a column is to be relocated, its "companion" column should follow as well.

In the second stage we are looking for relatively small elements on the diagonal of X_1. If such an element is found outside the $m \times m$ south-east diagonal block and it is associated with a real eigenvalue, it will not affect the accuracy of the algorithm; therefore nothing needs to be done in this case. If, however, it is associated with a complex eigenvalue, we find the columns with which the current column is almost linearly dependent, within a given tolerance. If one of these columns is associated with a real eigenvalue, we make an exchange similar to the exchange in the first stage and we update the slightly distorted upper triangular form of X_1. As a result the small diagonal element is now associated with a real eigenvalue, and it will be cancelled out. Finally if a small diagonal element is found within the $m \times m$ south-east diagonal block, we proceed as above in an attempt to first bring it outside this block and then associate it with a real eigenvalue.

Next we give two numerical examples to demonstrate the performance of the *bottom-up* algorithm equipped with the heuristics we just described. The accuracy of the computed solution will be compared against that of the three-step straightforward approach.

For the first example consider the case $n = 10$, $m = 2$ and take X_1 as in (5.1) with $\gamma = 0.7$. Compute random B_1, E_1 (make sure E_1 has full rank) and L_1 of appropriate forms. Also compute A_1 so that the equation $A_1 X_1 = E_1 X_1 L_1$ is satisfied. Then from $(A_1 - B_1 F_1) X_1 = E_1 X_1 L_1$, obviously $F_1 = O$, since E_1 has full rank. The choice of X_1 suggests that the three-step approach should demonstrate a loss of accuracy, up to seven significant digits (recall that $\chi(X_1) \approx 10^7$). Using MATLAB

(which uses an accuracy of approximately 16 significant digits) on a Pentium processor which is equipped with the IEEE floating point standard of arithmetic, the *bottom-up* algorithm produced an F_1 with $\|F_1\|_2 \approx 10^{-15}$. The three-step approach produced an F_1 with $\|F_1\|_2 \approx 10^{-10}$. The *bottom-up* algorithm, performed as well as was expected given MATLAB's accuracy, whereas the three-step approach clearly indicated loss of accuracy.

For the second example take

$$E_1 = \begin{pmatrix} 5 & 3 & 7 \\ & 0 & 4 \\ & & 1 \end{pmatrix},\ A_1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},\ B_1 = \begin{pmatrix} 3 \\ 6 \\ 1 \end{pmatrix},\ L_1 = \operatorname{diag}(5, 1, 3),$$

then the matrix of the feasible eigenvectors is given by

$$X = \begin{pmatrix} -0.6756 & -0.3353 & 0.5833 \\ -0.0528 & -0.5340 & 0.2419 \\ 0.7354 & 0.7762 & -0.7754 \end{pmatrix}.$$

The *bottom-up* algorithm produced $F_1 = \begin{pmatrix} 20.9532 & 12.5300 & 17.1445 \end{pmatrix}$, which gave $\lambda(A_1 - B_1F_1, E_1) = \{\infty, 5, 1\}$. The three-step approach produced $F_1 = \begin{pmatrix} -12 & -13.5 & -15 \end{pmatrix}$; using MATLAB's function eig for the computation of the eigenvalues of pencils we get $\operatorname{eig}(A_1 - B_1F_1, E_1) = \{-2.0265, 1.7765, \infty\}$, which suggests that apart from ∞ the other eigenvalues could be any complex numbers, that is, the resulting pencil is singular.

The MATLAB programs implementing the *bottom-up* algorithm are part of the MATLAB toolbox PolePack developed by the author and it can be found at the author's home page.

6. Conclusion. We have pointed out, by referring to the relevant literature, that the solution of the matrix equation $(A_1 - B_1F_1)X_1 = E_1X_1L_1$, with respect to F_1, is (or can be) a key point in eigenstructure assignment. We presented a new stepwise approach for its solution and we demonstrated that the new approach is numerically better than a three-step straightforward method that has been used up to now.

Acknowledgment. I wish to express my sincere thanks to the referees for their thorough reviews.

REFERENCES

[1] A. T. Alexandridis and G. D. Galanos, *Optimal pole-placement for linear multi-input controllable systems,* IEEE Trans. Circuits and Systems, 34 (1987), pp. 1602–1604.

[2] M. Arnold and B. N. Datta, *An algorithm for the multi-input eigenvalue problem*, IEEE Trans. Automat. Control, 35 (1990), pp. 1149–1152.

[3] 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.

[4] E. K.-W. Chu, *Controllability of descriptor systems*, Internat. J. Control., 46 (1987), pp. 1761–1770.

[5] G. R. Duan, *Solutions of the equation $AV + BW = VF$ and their application to eigenstructure assignment in linear systems*, IEEE Trans. Automat. Control, 38 (1993), pp. 276–280.

[6] M. M. Fahmy and J. O'Reilly, *On eigenstructure assignment in linear multivariable systems*, IEEE Trans. Automat. Control, 27 (1982), pp. 690–693.

[7] M. M. Fahmy and H. S. Tantawy, *Eigenstructure assignment via linear state-feedback control*, Internat. J. Control, 40 (1984), pp. 161–178.

[8] L. R. Fletcher, J. Kautsky, and N. K. Nichols, *Eigenstructure assignment in descriptor systems*, IEEE Trans. Automat. Control, 31 (1986), pp. 1138–1141.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[10] C. He, A. Laub, and V. Mehrmann, *Placing Plenty of Poles is Pretty Preposterous*, preprint SPC 95-17, Forschergruppe Scientific Parallel Computing, Fakultät für Mathematik, Technische Universität, Chemnitz-Zwickau, Germany.
[11] J. Kautsky, N. K. Nichols, and P. VanDooren, *Robust pole assignment in linear state feedback*, Internat. J. Control. 41 (1985), pp. 1129–1155.
[12] J. Kautsky, N. K. Nichols, and E. K.-W. Chu, *Robust pole assignment in singular control systems*, Linear Algebra Appl., 121 (1989), pp. 9–37.
[13] G. Klein and B. C. Moore, *Eigenvalue generalized eigenvector assignment with state feedback*, IEEE Trans. Automat. Control, 22 (1977), pp. 140–141.
[14] V. Mehrmann and H. Xu, *An analysis of the pole placement problem.* I. *The single-input case*, Electron. Trans. Numer. Anal., 4 (1996), pp. 138–157.
[15] V. Mehrmann and H. Xu, *An analysis of the pole placement problem.* II. *The multi-input case*, Electronic Transactions on Numerical Analysis, 5 (1997), pp. 77–97.
[16] G. Mengali, *Mixed linear-quadratic/eigenstructure strategy for the design of stability augmentation systems*, AIAA Journal of Guidance, Control and Dynamics, 19 (1996), pp. 1231–1238.
[17] G. S. Miminis and C. C. Paige, *A direct algorithm for pole assignment of time-invariant multi-input linear systems using state feedback*, Automatica, 24 (1988), pp. 343–356.
[18] G. S. Miminis, *Deflation in eigenvalue assignment of descriptor systems using state feedback*, IEEE Trans. Automat. Control, 38 (1993), pp. 1322–1336.
[19] G. S. Miminis, *Improving the performance of certain algorithms in eigenstructure assignment*, Proceedings of the 3rd IEEE Mediterranean Symposium on New Directions in Control and Automation, Limassol Cyprus I, 1995, IEEE, Piscataway, NJ, 1995, pp. 171–178.
[20] B. C. Moore, *On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment*, IEEE Trans. Automat. Control, 21 (1976), pp. 689–692.
[21] R. V. Patel and P. Misra, *Numerical algorithms for eigenvalue assignment by state feedback*, Proc. IEEE, 72 (1984), pp. 1755–1764.
[22] P. Petkov, N. Christov, and M. Konstantinov, *A computational algorithm for pole assignment of linear multi-input systems*, IEEE Trans. Automat. Control, 31 (1986), pp. 1044–1047.
[23] S. Pradhan, V. J. Modi, M. S. Bhat, and A. K. Misra, *Matrix method for eigenstructure assignment: The multi-input case with application*, AIAA Journal of Guidance, Control and Dynamics, 17 (1994), pp. 983–989.
[24] G. Roppenecker, *On parametric state feedback design*, Internat. J. Control, 43 (1986), pp. 793–804.
[25] V. Sinswat and F. Fallside, *Eigenvalue eigenvector assignment by state-feedback*, Internat. J. Control, 26 (1977), pp. 389–403.
[26] V. L. Syrmos and F. L. Lewis, *Robust eigenvalue assignment for generalized systems*, Automatica, 28 (1992), pp. 1223–1228.
[27] C.-C. Tsui, *On the solution to matrix equation $TA - FT = LC$ and its applications*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 33–44.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 925–947

A MULTILEVEL DUAL REORDERING STRATEGY FOR ROBUST INCOMPLETE LU FACTORIZATION OF INDEFINITE MATRICES*

JUN ZHANG†

Abstract. A dual reordering strategy based on both threshold and graph reorderings is introduced to construct robust incomplete LU (ILU) factorization of indefinite matrices. The ILU matrix is constructed as a preconditioner for the original matrix to be used in a preconditioned iterative scheme. The matrix is first divided into two parts according to a threshold parameter to control diagonal dominance. The first part with large diagonal dominance is reordered using a graph-based strategy, followed by an ILU factorization. A partial ILU factorization is applied to the second part to yield an approximate Schur complement matrix. The whole process is repeated on the Schur complement matrix and continues for a few times to yield a multilevel ILU factorization. Analyses are conducted to show how the Schur complement approach removes small diagonal elements of indefinite matrices and how the stability of the LU factor affects the quality of the preconditioner. Numerical results are used to compare the new preconditioning strategy with two popular ILU preconditioning techniques and a multilevel block ILU threshold preconditioner.

Key words. reordering strategies, sparse matrices, incomplete LU factorization, multilevel incomplete LU preconditioner

AMS subject classifications. 65F10, 65N06

PII. S0895479899354251

1. Introduction.

This paper is concerned with reordering strategies used in developing robust preconditioners based on incomplete LU (ILU) factorization of the coefficient matrix of sparse linear system of the form

$$Au = b, \tag{1.1}$$

where A is an unstructured matrix of order n. In particular, we are interested in ILU preconditioning techniques for which A is an indefinite matrix, i.e., a matrix with an indefinite symmetric part. Indefinite matrices arise frequently from finite element discretizations of coupled partial differential equations in computational fluid dynamics and from other applications.

ILU preconditioning techniques have been successful for solving many nonsymmetric and indefinite matrices, despite the fact that their existence in these applications is not guaranteed. However, their failure rates are still too high for them to be used as blackbox library software for solving general sparse matrices of practical interests [9, 25]. In fact, the lack of robustness of preconditioned iterative methods is currently the major impediment for them to gain acceptance in industrial applications, in spite of their intrinsic advantage for large scale problems.

For indefinite matrices, there are at least two reasons that make ILU factorization approaches problematic [9]. The first problem is due to small or zero pivots [26].

*Received by the editors April 14, 1999; accepted for publication (in revised form) by E. Ng September 6, 2000; published electronically January 5, 2001. The work of this author was supported in part by the U.S. National Science Foundation under grants CCR-9902022, CCR-9988165, and CCR-0043861, and in part by the University of Kentucky Center for Computational Sciences and by the University of Kentucky College of Engineering.

http://www.siam.org/journals/simax/22-3/35425.html

†Laboratory for High Performance Scientific Computing and Numerical Simulation, Department of Computer Science, University of Kentucky, 773 Anderson Hall, Lexington, KY 40506-0046 (jzhang@cs.uky.edu, http://www.cs.uky.edu/∼jzhang).

Pivots in an indefinite matrix can be arbitrarily small. This may lead to unstable and inaccurate factorizations. In such cases, the size of the elements in the LU factors may be very large, and these large size elements lead to inaccurate factorization. The second problem is due to unstable triangular solutions [18]. The incomplete factors of an indefinite matrix are usually not diagonally dominant. An indication of unstable triangular solutions is when $\|L^{-1}\|$ and $\|U^{-1}\|$ are extremely large while the offdiagonal elements of L and U are reasonably bounded. Such problems are usually caused by very small pivots. They may sometimes happen without a small pivot. A statistic, *condest*, was introduced by Chow and Saad [9] to measure the stability of triangular solutions. It is defined to be $\|(LU)^{-1}e\|_\infty$, where e is a vector of all ones. This statistic is useful when its value is very large, e.g., on the order of 10^{15}.

Small pivots are usually related to small or zero diagonal elements. It can be argued that by restricting the magnitude of the diagonal elements, we may be able to alleviate, if not eliminate, these two problems of ILU factorizations to a certain degree. Such restrictions can be seen in the form of full or partial pivoting strategies in Gaussian elimination. In ILU factorization, column pivoting strategy has been implemented with Saad's ILU threshold (ILUT), resulting in an ILU threshold pivoting (ILUTP) variant [35]. However, ILUTP has not always been helpful in dealing with nonsymmetric matrices [3, 9]. As Chow and Saad pointed out [9], a poor pivoting sequence can occasionally trap a factorization into a zero pivot, even if the factorization would have succeeded without pivoting. In addition, existing pivoting strategies for incomplete factorization cannot guarantee that a nonzero pivot will always be found, unlike the case with Gaussian elimination [9].

Another obvious strategy for dealing with small pivots is to replace them by a larger value. The ILU factorization can continue, and the resulting preconditioner may be well conditioned. In such a way, the ILU factorization is said to be stabilized. However, this strategy alters the values of the matrix, and the quality of the resulting preconditioner may be deteriorated. Thus, the choice of the replacing value for the small pivots is critical for good performance, and a good choice is usually problem dependent [26]. Too large a value will result in a stable but less accurate factorization; too small a value will result in an unstable factorization. A similar strategy is to factor a shifted matrix $A+\alpha I$, where α is a positive scalar so that $A+\alpha I$ is well conditioned [30, 47]. Such a strategy too obviously has a tradeoff between stable and accurate factorization. For more studies on the stability of ILU factorizations, we refer to [19, 32, 45, 13, 48].

It is also possible to reorder the rows of the matrix so that their diagonal dominance in a certain sense is in decreasing order. In this way, small pivots are in the last rows of the matrix and may not be used in an ILU factorization. This strategy also has some problems since the values of the pivots are modified in an unpredictable way and small pivots may still affect the ILU factorization. In addition, the effect of standard reordering schemes applied to general nonsymmetric sparse matrices is still an unsettled issue [17, 27, 46].

This paper follows the above idea of moving the rows with small diagonal elements to the last few rows. However, these small diagonal elements will never be used in the ILU factorization. Instead, these rows form the rows of a Schur complement matrix, and the values of the diagonal elements are modified in a systematic way. This process is continued for a few times until all small diagonal elements are removed, or until the last Schur complement matrix is small enough that a full pivoting strategy can be implemented inexpensively. With this reordering strategy, we can expect to obtain a

stable and accurate ILU factorization. We also implement a graph-based reordering strategy (nondecreasing degree algorithm) to reduce the fill-in amount during the stable ILU factorization.

This paper is organized as follows. The next section introduces a dual reordering strategy based on both the values and the graph of the matrix. Section 3 discusses a partial ILU factorization technique to construct the Schur complement matrix implicitly. Section 4 gives analyses on the values of the diagonal elements of the Schur complement matrix and shows how the stability of the LU factor affects the quality of a preconditioner. Section 5 outlines the multilevel dual reordering algorithm. Section 6 contains numerical experiments. Concluding remarks are included in section 7.

2. A dual reordering strategy. Most reordering strategies are originally developed for the direct solution of sparse matrices based on Gaussian elimination. They are mainly used to reduce fill-in elements in the Gaussian elimination process or to extract parallelism from LU factorizations [15, 24]. They have also been used in ILU preconditioning techniques for almost the same reasons [16, 20, 33]. Various reordering strategies were first studied for preconditioned conjugate gradient methods, i.e., for the cases where the matrix is symmetric positive definite [1, 4, 5, 10, 11, 29, 34]. They were then extended for treating nonsymmetric problems [2, 7, 12, 14]. Most of these strategies are based on the adjacency graph but not on the values of the matrices. They are robust for general sparse matrices only if used with suitable pivoting strategies, which are based on the values of the matrices, to prevent unstable factorizations. Hence, reordering strategies based on matrix values are needed to yield robust stable ILU factorizations [11]. Such an observation has largely been overlooked in ILU techniques for some time, partly because the early ILU techniques were mainly developed to solve sparse matrices arising from finite difference discretizations of partial differential equations [31]. In such cases, the diagonal elements of the matrices usually have nonzero values.

In this paper, we introduce a dual reordering strategy for robust ILU factorization for solving general sparse indefinite matrices. To this end, we first introduce a strategy to determine the row diagonal dominance of a matrix.[1] We actually compute a certain measure to determine the relative strength of the diagonal element with respect to a certain norm of the row in question. Algorithm 2.1 is an example of computing a diagonal dominance measure for each row of the matrix and was originally introduced in [43] as a diagonal threshold strategy in a multilevel ILU factorization.

ALGORITHM 2.1. Computing a measure for each row of a matrix.

1. *For* $i = 1, \ldots, n$, *do*
2. $\quad r_i = \sum_{j \in \mathrm{Nz}(A_i)} |a_{ij}|$
3. $\quad$ *If* $r_i \neq 0$, *then*
4. $\quad\quad \tilde{t}_i = |a_{ii}|/r_i$
5. $\quad$ *End if*
6. *End do*
7. $T = \max_i\{\tilde{t}_i\}$
8. *For* $i = 1, \ldots, n$, *do*
9. $\quad t_i = \tilde{t}_i/T$
10. *End do*

[1] The reference to row diagonal dominance is due to the assumption that our matrix is stored in a row-oriented format, such as in the compressed sparse row format [37]. The proposed strategy works equally well if the matrix is stored in a column-oriented format with the reference to column diagonal dominance.

In line 2 of the Algorithm 2.1 the set $\mathrm{Nz}(A_i)$ is defined as $\mathrm{Nz}(A_i) = \{j : a_{ij} \neq 0, 1 \leq i, j \leq n\}$, i.e., the nonzero row pattern for the row i. A row with a small absolute diagonal value will have a small t_i measure. A row with a zero diagonal value will have an exact zero t_i measure.

Let $G = (V, K)$ denote the adjacency (directed) graph of the matrix A, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and K is the set of edges. Let (v_j, v_k) denote an edge from vertex v_j to vertex v_k. Since a node in the adjacency graph of a matrix corresponds to a row of the matrix, we will use the term node and row of a matrix interchangeably. Given a diagonal threshold tolerance $\epsilon > 0$, we divide the nodes of A into two parts, V_1 and V_2, such that $t_i \geq \epsilon \rightarrow v_i \in V_1$, otherwise $v_i \in V_2$. It is obvious that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. (For ensuring fast reduction of matrix size, it is important that the size of V_1 be large enough, e.g., larger than $n/2$. Hence, a very large value of ϵ is not suitable. However, there is no restriction on the size of V_1 explicitly implemented in this paper.)

For convenience, we assume that a symmetric permutation is performed so that the nodes in V_1 are listed first, followed by the nodes in V_2. Since the nodes in V_1 are "good" for ILU factorization in terms of stability, we may further improve the quality of the ILU factorization by implementing a graph-based reordering strategy. The following nondecreasing degree reordering algorithm is just one example of such graph-based reordering strategies.

We denote by $\deg(v_i)$ the degree of the node v_i, which equals the number of nonzero elements of the ith row minus one, i.e., $\deg(v_i) = \mathrm{Nz}(A_i) - 1 = \mathrm{Nz}(v_i) - 1$. The set of the degrees of the rows of the matrix A can be conveniently computed when Algorithm 2.1 is run to compute the diagonal dominance measure of A. For example, in line 2 of Algorithm 2.1, the number of nonzero elements of the ith row will be counted.

After the first reordering based on the threshold tolerance ϵ, we perform a second reordering based on the degrees of the nodes. But the second reordering is only performed with respect to the nodes in V_1. To be more precise, we reorder the nodes in V_1 in a nondecreasing degree fashion; i.e., the nodes with smaller degrees are listed first and those with larger degrees are listed last. After the two steps of reorderings, we have

$$A \sim P_g P_t A P_t^T P_g^T = \begin{pmatrix} D & F \\ E & C \end{pmatrix}, \tag{2.1}$$

where P_t and P_g are the permutation matrices corresponding to the threshold tolerance reordering and to the nondecreasing degree reordering, respectively. We use P_g here to emphasize that it is just a graph-based reordering strategy and is not necessarily restricted to the nondecreasing degree reordering. Other graph-based reordering strategies such as the Cuthill-McKee or reverse Cuthill-McKee algorithms [28] may be used to replace the nondecreasing degree strategy. However, their meaning may be slightly changed since not all neighboring nodes of a node in V_1 belong to V_1; some of them may be in V_2. Also, these graph reordering algorithms are implemented in a prereordered fashion, not in a dynamic fashion. Thus, the ordering is not updated during the factorization process. For simplicity, we use A to denote both the original and the permuted matrices in what follows so that the permutation matrices will no longer appear explicitly. We also refer to the two reordering strategies as threshold reordering and graph reordering for short.

3. Partial ILU factorization. An ILU factorization process with a double dropping strategy (ILUT) is first applied to the upper part $(D\ F)$ of the reordered matrix A in (2.1). The ILUT algorithm uses two parameters p and τ to control the amount of fill-in elements caused by the Gaussian elimination process and is described in detail in [35]. ILUT builds the preconditioning matrix row by row. For each row of the LU factors, ILUT first drops all computed elements whose absolute values are smaller than τ times the average nonzero absolute values of the current row. After an (incomplete) row is computed, ILUT performs a search with respect to the computed current row such that the largest p elements in absolute values are kept and the rest of the nonzero elements are dropped again. Thus, the resulting ILUT factorization has at most p elements in each row of the L and U parts. The use of a double dropping strategy ensures that the memory requirement be met. It is easy to see that the total storage cost for ILUT is bounded by $2pn$ for a matrix of order n.

The ILUT process is continued to the second part of the matrix A in (2.1) with respect to the $(E\ C)$ submatrix. However, the elimination process is only performed with respect to the columns in E, and linear combinations for columns in C are performed accordingly. In other words, the elements corresponding to the C submatrix are not eliminated. (This can be done by modifying the ILUT algorithm of Saad [37] and restricting the elimination process to the columns corresponding to V_1, when the row index is greater than the size of V_1.) Such a process is called a partial Gaussian elimination or a partial LU factorization in [41]. Note that, due to the partial Gaussian elimination, all rows in the $(E\ C)$ submatrix can be processed independently (in parallel). This is because all nodes in the E submatrix that are to be eliminated use only the computed (I)LU factorization of the $(D\ F)$ part. Note also that the diagonal values of the rows of the C submatrix are never used as pivots. It can be shown [41] that such a partial Gaussian elimination process modifies C into the (incomplete) Schur complement of A. In exact arithmetic, C would be changed into

$$A_1 = C - ED^{-1}F = C - EU^{-1}L^{-1}F, \tag{3.1}$$

where LU is the standard LU factorization of the D submatrix. We point out that A_1 is constructed by updating C row by row with drop tolerance applied even on updates. Hence, this method constructs the Schur complement indirectly, in contrast to some alternative methods, e.g., the multilevel block ILU (BILUM) preconditioner in [40], in which the Schur complement is constructed explicitly by matrix-matrix multiplications. Sparsity and computation costs are kept low by adapting the dual dropping strategy of ILUT with respect to the computation of each row of A_1. In particular, small size fill-in with respect to τ is dropped as soon as it is computed in each update. The maximum number of nonzeros kept in a row of A_1 is limited to p after the row is computed.

The partial ILU factorization process just described yields a block LU factorization of the matrix A of the form

$$\begin{pmatrix} D & F \\ E & C \end{pmatrix} = \begin{pmatrix} L & 0 \\ EU^{-1} & I \end{pmatrix} \begin{pmatrix} U & L^{-1}F \\ 0 & A_1 \end{pmatrix}, \tag{3.2}$$

where I and 0 are generic identity and zero matrices, respectively. If the factorization is exact and if we can solve the Schur complement matrix A_1, the solution of the original linear system (1.1) can be found by a backward substitution.

The partial ILU factorization process is the backbone of a domain-based multilevel ILU preconditioning technique (BILUTM) described in [41]. Such an ILU

factorization with a suitable block independent set ordering yields a preconditioner (BILUTM) that is highly robust and possesses a high degree of parallelism. However, in this paper, the parallelism due to block independent set ordering is not our concern, so we restrict our attention to the robustness of multilevel ILU factorization resulting from removing small pivots.

We can heuristically argue that the ILU factorization resulting from applying the above partial ILU factorization to the reordered matrix is likely to be more stable than that which would be generated by applying ILUT directly to the original matrix. This is because the factorization is essentially performed with respect to the nodes in V_1 that have a relatively good diagonal dominance. The partial ILU factorization with respect to the nodes in V_2 never needs to divide any pivot elements. So there is no reason that large size elements should be produced.

As remarked previously, if we can solve the Schur complement matrix A_1 in (3.1) to a certain degree of accuracy, we can develop a two level preconditioner for the matrix A. An alternative is based on the observation that A_1 is another sparse matrix and we can apply the same procedures to A_1 that have been applied to A to yield an even smaller Schur complement A_2. This is the philosophy of multilevel ILU preconditioning techniques developed in [36, 40, 41]. However, for the moment, we only discuss the possible construction of a two level preconditioner.

A two level preconditioner. The easiest way to construct a two level preconditioner is to apply the ILUT factorization technique to the matrix A_1. One question will be naturally asked. Is the ILUT factorization more stable when applied to A_1 than when applied to A?

Notice that since the nodes with good (large) diagonal dominance have all been factored out, we tend to think that the nodes of A_1 are not good for a stable ILUT factorization. This may not always be true, since the measure of diagonal dominance computed in Algorithm 2.1 is relative to a certain norm of the row in question. We need to examine relative changes in size of the diagonal value when a node is considered as a node in A and when it is considered as a node in A_1.

4. Analyses.

Diagonal submatrix D. For the ease of analysis, unless otherwise indicated explicitly, we assume that the partial LU factorization described above is exact; i.e., no dropping strategy is enforced. We also assume that, in the reordered matrix, the D submatrix is diagonal. Such a reordering can be achieved by an independent set search as in a multielimination strategy of Saad [36, 40]. Thus, the factorization (3.2) is reduced to

$$\begin{pmatrix} D & F \\ E & C \end{pmatrix} = \begin{pmatrix} I & 0 \\ ED^{-1} & I \end{pmatrix} \begin{pmatrix} D & F \\ 0 & A_1 \end{pmatrix}. \tag{4.1}$$

We now assume that all indices are local to individual submatrices. In other words, when we say the ith row of the matrix F, we mean the ith row of the submatrix F, not the ith row of the matrix A, original or permuted. For convenience we assume that D is of dimension m and A_1 is of dimension $l = n - m$. We also use the following notations:

$$D = \mathrm{diag}[d_1, \ldots, d_m], \quad F = (f_{ij})_{m\times l}, \quad E = (e_{ij})_{l\times m}, \quad C = (c_{ij})_{l\times l}, \quad A_1 = (s_{ij})_{l\times l}.$$

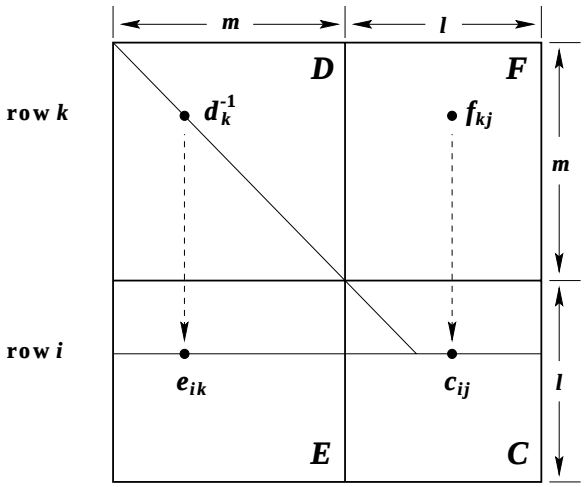


FIG. 4.1. *An illustration of the partial LU factorization to eliminate e_{ik} in the E submatrix.*

It can be shown [24, 41] that, with the partial LU factorization without dropping, an arbitrary element of the Schur complement matrix A_1 is

$$s_{ij} = c_{ij} - \sum_{k=1}^{m} e_{ik} f_{kj} / d_k. \tag{4.2}$$

Since we assume that the nodes with large diagonal dominance measure are in V_1 and the nodes in V_2 have small or zero diagonal dominance measure, we are interested in knowing how the diagonal value of a node of A may change when it becomes a node in A_1.

The following proposition is obvious from (4.2) and from Figure 4.1.

PROPOSITION 4.1. *If either the jth column of the submatrix F or the ith row of the submatrix E is a zero vector, then $s_{ij} = c_{ij}$.*

DEFINITION 4.2. *A node v_i of the vertex set V is said to be independent from a subset V_I of V if and only if*

$$a_{ij} = 0 \quad and \quad a_{ji} = 0 \quad for\ all \quad v_j \in V_I.$$

An immediate consequence of the independence is the following corollary that is first proved in [42].

COROLLARY 4.3. *If a node v_i in V_2 is independent from all the nodes in V_1, then $s_{ii} = c_{ii}$; i.e., the values of the ith row of C will not be modified in the partial LU factorization.*

We now modify our threshold tolerance reordering strategy slightly to a diagonal threshold strategy, similar to that discussed in [42]. We assume that the node v_i is in V_1 if $|a_{ii}| \geq \epsilon$ and D is still a diagonal matrix. With such a modification, we have $|d_i| \geq \epsilon$ for $1 \leq i \leq m$. Denote by $M = \max_{1 \leq i,j \leq n}\{|a_{ij}|\}$ the size of the largest elements in absolute value of A.

PROPOSITION 4.4. *The size of the elements of the Schur complement matrix A_1 is bounded by $M(1 + mM/\epsilon)$.*

Proof. Starting from (4.2)

$$|s_{ij}| \le |c_{ij}| + \sum_{k=1}^{m} |e_{ik}||f_{kj}|/|d_k| \le M + \sum_{k=1}^{m} MM/\epsilon = M(1 + mM/\epsilon). \qquad \square$$

Proposition 4.4 shows that the size of the elements of the Schur complement matrix cannot grow uncontrollably if ϵ is large enough. This result indicates that our first level (I)LU factorization is stable.

As we hinted previously, we will be interested in recursively applying our strategy to the successive Schur complement matrices. We may assume that the matrix A is presparsified so that small nondiagonal elements are removed. To be more specific, for the parameter τ used in the ILUT factorization, we assume $\min_{1\le i,j\le n}\{|a_{ij}|\} \ge \tau$ for all nonzero elements of A, except for possibly the diagonal elements. With some additional assumptions, we can have a lower bound on the variation of the diagonal values of the Schur complement matrix A_1.

PROPOSITION 4.5. *Suppose $|a_{ij}| \ge \tau$ for all nonzero offdiagonal elements of the matrix A, and suppose that either $e_{ik}f_{ki}/d_k \ge 0$ or $e_{ik}f_{ki}/d_k \le 0$ holds for all $1 \le k \le m$. Then*

$$|s_{ii} - c_{ii}| \ge \frac{\operatorname{card}\left(\operatorname{Nz}(E_i) \cap \operatorname{Nz}(F_i)\right)\tau^2}{M},$$

where $\operatorname{Nz}(E_i)$ and $\operatorname{Nz}(F_i)$ are the index sets of the nonzero elements of the ith row of the E submatrix and the ith column of the F submatrix, respectively. $\operatorname{card}(V)$ *denotes the cardinality of a set V.*

Proof. If either $e_{ik}f_{ki}/d_k \ge 0$ or $e_{ik}f_{ki}/d_k \le 0$ holds for all $1 \le k \le m$, we have

$$\left|\sum_{k=1}^{m} e_{ik}f_{ki}/d_k\right| = \sum_{k=1}^{m} |e_{ik}f_{ki}/d_k|. \tag{4.3}$$

The kth term in the right-hand side sum of (4.3) is nonzero if and only if both e_{ik} and f_{ki} are nonzero. This happens if and only if $k \in \operatorname{Nz}(E_i) \cap \operatorname{Nz}(F_i)$.

Note that $|e_{ik}| \ge \tau, |f_{ki}| \ge \tau$ and $|d_k| \le M$ for all $1 \le k \le m$. It follows that

$$|s_{ii} - c_{ii}| = \sum_{k=1}^{m} |e_{ik}f_{ki}/d_k| \ge \frac{\operatorname{card}\left(\operatorname{Nz}(E_i) \cap \operatorname{Nz}(F_i)\right)\tau^2}{M}. \qquad \square$$

It is implicitly assumed that $\epsilon < M$. In practice, ϵ is small so that the set V_1 may be large enough to avoid constructing a large Schur complement matrix. (We consider a reduction of matrix size large if $\operatorname{card}(V_1) \ge \operatorname{card}(A)/2$.) Denote

$$\Delta_i = \frac{\operatorname{card}\left(\operatorname{Nz}(E_i) \cap \operatorname{Nz}(F_i)\right)\tau^2}{M}.$$

By the motivation of the diagonal threshold strategy, the value of $|c_{ii}|$ is zero or very small. Thus, the size of $|s_{ii}|$ can be considered as being close to Δ_i.

COROLLARY 4.6. *Under the conditions of the Proposition* 4.5, *if $c_{ii} = 0$, then $|s_{ii}| \ge \Delta_i$.*

Corollary 4.6 shows that if the ith diagonal element of A_1 is zero in A and if the set $\operatorname{Nz}(E_i) \cap \operatorname{Nz}(F_i)$ is nonempty, then the size of the ith diagonal element is nonzero in the Schur complement. Thus, under these conditions, a zero pivot is removed. In

fact, the cardinality of $\mathrm{Nz}(E_i) \cap \mathrm{Nz}(F_i)$ seems to be the key factor to remove zero diagonal elements.

It is difficult to derive more useful bounds for general sparse matrices. If certain conditions are given to restrict the class of matrices under consideration, it is possible to obtain more realistic bounds to characterize the size of the elements of the Schur complement matrix, especially the size of its diagonal elements.

General submatrix D. For general submatrix D corresponding to the factorization (3.2), it is easy to see that, if the jth column of the submatrix F is zero, the jth column of the submatrix $L^{-1}F$ is zero. Hence, Proposition 4.1 carries over to the general case.

At this moment, we are unable to show results analogous to Propositions 4.4 and 4.5 for general submatrix D. However, it can be argued heuristically that, if D is not a diagonal matrix, the cardinality of the set $\mathrm{Nz}(E_i) \cap \mathrm{Nz}((L^{-1}F)_i)$ is likely to be larger than that of $\mathrm{Nz}(E_i) \cap \mathrm{Nz}(F_i)$.

Size of $\|(LU)^{-1}\|$. Most authors discuss the quality of preconditioning techniques with respect to condition number of the preconditioned matrix $(LU)^{-1}A$, which is very difficult to obtain for general sparse matrices. We choose to consider the quality of preconditioning in a nonstandard way [26, 42]. Denote by

$$R = A - LU \tag{4.4}$$

the error (residual) matrix of the ILU factorization. At each iteration, the preconditioning step solves for $\bar{w}$ the system

$$LU\bar{w} = r, \tag{4.5}$$

where r is the residual of the current iterate. In a certain sense, we can consider $\bar{w}$ as an approximate to the correction term of the current iterate. The quality of the preconditioning step (4.5) can be judged by comparing (4.5) with the exact or perfect preconditioning step

$$Aw = r. \tag{4.6}$$

If (4.6) could be solved to yield the exact correction term w, the preconditioned iterative method would converge in one step. Of course, solving (4.6) is as hard as solving the original system (1.1). However, we can measure the relative difference in the correction term when approximating (4.6) by (4.5). This difference may tell us how good the preconditioning step (4.5) approximates the exact preconditioning step (4.6). The following proposition is motivated by the work of Kershaw [26].

PROPOSITION 4.7. *Suppose the matrix A and the factor LU from the incomplete LU factorization are nonsingular; then the following inequality holds:*

$$\frac{\|w - \bar{w}\|}{\|w\|} \le \|(LU)^{-1}\| \, \|R\| \tag{4.7}$$

for any consistent norm $\|\cdot\|$.

Proof. It is obvious that $r \neq 0$, otherwise the iteration would have converged. The nonsingularity of A implies that $w \neq 0$. Note that $\bar{w} = (LU)^{-1}r$. From (4.6), we have

$$w - \bar{w} = w - (LU)^{-1}r = w - (LU)^{-1}Aw = \left(I - (LU)^{-1}A\right) w = -(LU)^{-1}Rw.$$

It follows that, for any consistent norm,

$$\|w - \bar{w}\| = \|(LU)^{-1}Rw\| \le \|(LU)^{-1}\| \, \|R\| \, \|w\|.$$

The desired result (4.7) follows immediately by dividing $\|w\|$ on both sides. ∎

It is well known that the size of the error matrix R directly affects the convergence rate of the preconditioned iterative methods [16]. Proposition 4.7 shows that the quality of a preconditioning step is directly related to the size of both $(LU)^{-1}$ and R. A high quality preconditioner must be accurate; i.e., it must have an error matrix that is small in size. A high quality preconditioner must also have a stable factorization and stable triangular solutions; i.e., the size of $(LU)^{-1}$ must be small. Since the condition estimate, condest $= \|(LU)^{-1}e\|_\infty$, is a lower bound for $\|(LU)^{-1}\|_\infty$, it should provide some information about the quality of the preconditioner and may be used to measure the stability of the LU factorization and of the triangular solutions.

If D is diagonal, the Schur complement approach is similar to the red-black ordering applied to certain matrices, followed by one step cyclic reduction. The reduced system A_1 may be better conditioned than the original matrix A and thus may be a better starting point for constructing a better preconditioner [21, 22].

When D is diagonal, both BILUM [40] and BILUTM [41] of Saad and Zhang may encounter stability problems if the diagonal values are not restricted. This shows the advantage of our first threshold-based reordering strategy. Of course, the preconditioners proposed in this paper rarely result in a submatrix D that is diagonal. The assumption of the diagonal property of D is made for convenience in the analysis.

5. Multilevel dual reordering and ILU factorization. Based on our previous analyses, the size of a diagonal element of the matrix A_1 is likely to be larger than that of the same element in A.[2] We can apply Algorithm 2.1 to A_1 and repeat on A_1 the procedures that were applied to A. (The measure of diagonal dominance has to be recomputed for the rows in A_1, but A_1 is not recomputed.) This process may be repeated for a few times until all small diagonal elements are modified to large values, or until the last Schur complement matrix is small enough that an ILU factorization with a full pivoting strategy can be implemented inexpensively. Since the number of small or zero pivots in the last Schur complement matrix is small, a third strategy is to replace them by a larger value. This will not introduce too much error to the overall factorization. Given a maximum level $\mathcal{L}$ and denoting $A_0 = A$, the multilevel dual reordering strategy and ILU factorization can be formulated as Algorithm 5.1.

ALGORITHM 5.1. Multilevel dual reordering and ILU factorization.

Given the parameters $\tau, p, \epsilon, \mathcal{L}$
For $j = 0, 1, \ldots, \mathcal{L} - 1$, *do:*
  Run Algorithm 2.1 *with* ϵ *to find permutation matrices* P_{j_t} *and* P_{j_g}
  Perform matrix permutation $A_j = P_{j_g}^T P_{j_t}^T A_j P_{j_t} P_{j_g}$
  If no small pivot has been found, then
    Apply ILUT(p, τ) *to* A_j *and exit*
  Else
    Apply a partial ILU factorization to A_j
      to yield a Schur complement matrix A_{j+1}
  End if
End do
Apply ILUTP or a stabilized ILUT to $A_{\mathcal{L}}$ *if* $A_{\mathcal{L}}$ *exists*

The ILU preconditioner constructed by Algorithm 5.1 is structurally similar to the BILUTM preconditioner in [41]. The difference is that we do not construct a block

[2] This is obviously false for an M-matrix. However, there will be no Schur complement matrix at all if A is an M-matrix, since $V_1 = V$ for some $\epsilon > 0$.

independent set for the D_j submatrix. Instead, we set up a diagonal measure constraint and employ a graph reordering scheme to increase preconditioning robustness. The emphasis of this paper is on solving indefinite matrices by removing small pivots. The emphasis of BILUM [40] and BILUTM [41] is to extract potential parallelism from ILU factorizations, although both BILUM and BILUTM have been shown to be much more robust than standard ILU preconditioners. We have departed from a fundamental multilevel concept of treating different error components on different levels, and we have considered preconditioning strategies in a sense closer to constructing approximate direct solvers.

It can be seen, if $\mathcal{L}$ levels of reduction are performed, that the resulting ILU preconditioner has the following structure:

$$\begin{pmatrix} L_0U_0 & & & L_0^{-1}F_0 \\ & \begin{pmatrix} L_1U_1 & & & L_1^{-1}F_1 \\ & \begin{pmatrix} \cdots & & \cdots \\ & \begin{pmatrix} L_{\mathcal{L}-1}U_{\mathcal{L}-1} & L_{\mathcal{L}-1}^{-1}F_{\mathcal{L}-1} \\ E_{\mathcal{L}-1}U_{\mathcal{L}-1}^{-1} & L_{\mathcal{L}}U_{\mathcal{L}} \end{pmatrix} \end{pmatrix} \end{pmatrix} \\ E_0U_0^{-1} & E_1U_1^{-1} & \cdots \end{pmatrix}.$$

The application of the preconditioner can be done by a level-by-level forward elimination, followed by a level-by-level backward substitution. There are also permutations and inverse permutations to be performed; specific procedures depend on implementations. For detailed descriptions, we refer the reader to [40, 41].

6. Numerical experiments. Standard implementations of multilevel preconditioning methods have been described in detail in [36, 40, 41]. We used full GMRES as the accelerator [38]. We tested four preconditioners: standard ILUT of [35], a column pivoting variant ILUTP [35], a domain-based multilevel block ILUT preconditioner (BILUTM) [41], and the multilevel dual reordering preconditioner designed in this paper, abbreviated as MDRILU (multilevel dual reordering ILU factorization). All preconditioners used a safeguard (stabilization) procedure by replacing a zero pivot with $(0.0001 + \tau)r_i$, where r_i was computed as the average nonzero values of the row in question. They were used as right preconditioners for GMRES [37]. The main parameters used in all four preconditioners are the pair (p, τ) in the double dropping strategy. ILUTP needs another parameter $0 \le \sigma \le 1$ to control the actual pivoting. A nondiagonal element a_{ij} is a candidate for a permutation only when $\sigma|a_{ij}| > |a_{ii}|$. It is suggested that reasonable values of σ are between 0.5 and 0.01, with 0.5 being the best in many cases [37, p. 295]. MDRILU also needs another parameter ϵ to enforce the diagonal threshold reordering as in Algorithm 5.1. The block size of BILUTM was chosen equal to p. The maximum possible level number in MDRILU and BILUTM was $\mathcal{L} = 10$. If after 10 levels of dual reorderings the Schur complement A_{10} is not empty, a stabilized ILUT factorization was employed to factor A_{10}.[3]

For all linear systems, the right-hand side was generated by assuming that the solution is a vector of all ones. The initial guess was a vector of some random numbers. The iteration was terminated when the 2-norm of the residual was reduced by a factor of 10^7. We also set an upper bound of 100 for the full GMRES iteration. A symbol "–" indicates lack of convergence.

In all tables with numerical results, "iter" shows the number of preconditioned GMRES iterations; "spar" shows the sparsity ratio which is the ratio between the

[3]We found stabilized ILUT was better than ILUTP for solving the last system. We did not implement an ILUT factorization with a full pivoting strategy.

number of nonzero elements of the preconditioner to that of the original matrix; "prec" shows the CPU time in seconds spent in constructing the preconditioners; "totl" is the total CPU time in seconds, including the preconditioner construction time and the solution (iteration) time; "cond" $= \|(LU)^{-1}e\|_\infty$ is the condition estimate of the preconditioners as introduced in section 1. Since these ILU preconditioners approach direct solvers as $p \to n$ and $\tau \to 0$, we compare their robustness with respect to the memory cost (sparsity ratio). We remark that our codes were not optimized and they computed and printed information such as the number of zero diagonals, smallest pivots, etc. Consequently, the CPU times reported in this paper have only relative meaning. Note that the solution time at each iteration is mainly the cost of the matrix vector products (both A and the preconditioner) and is thus proportional to the product of the iteration count and the sparsity ratio, i.e., solution time $\sim$ iter $*$ (1 + spar).

The numerical experiments were conducted on a Power-Challenge XL Silicon Graphics workstation equipped with 512 MB of main memory, one 190 MHz R10000 processor, and 1 MB secondary cache. We used Fortran 77 programming language in 64 bit arithmetic computation.

Test matrices. Three test matrices were selected from different applications. Table 6.1 contains simple descriptions of the first three test matrices. They have been used in several other papers [6, 9, 42, 50]. None of the three matrices has a zero diagonal.

TABLE 6.1
Simple descriptions of the test matrices.

Matrix	Order	Nonzeros	Description
RAEFSKY4	19 779	1 328 611	Buckling problem for container model
UTM5940	5 940	83 842	Nuclear fusion plasma simulation
WIGTO966	3 864	238 252	Euler equation model

WIGTO966 *matrix.* The WIGTO966 matrix[4] was supplied by L. Wigton from Boeing Company. It is solvable by ILUT with large values of p [6]. This matrix was also used to compare BILUM with ILUT in [39] and BILUTM with ILUT in [41], and to test point and block preconditioning techniques in [8, 9]. Since ILUT requires a very large amount of fill-in to converge, the WIGTO966 matrix is ideal to test alternative preconditioners and to show the least memory that is required for convergence. For example, BILUM (with GMRES(10)) was shown to be six times faster than ILUT with only one-third of the memory required by ILUT [39]. BILUTM (with GMRES(50)) converged almost five times faster and used just about one-fifth of the memory required by ILUT [41]. Table 6.2 lists results from several runs to compare MDRILU and ILUT. It shows that MDRILU could converge with low sparsity ratios, as low as 0.94. The threshold parameter ϵ was in a fixed range when the other parameters p and τ changed. For all the values of p and τ tested in Table 6.2, ILUT did not converge. We found that there was no very small pivot; the size of the smallest pivot in all tests in Table 6.2 was 1.19e-5. But the condition estimates for ILUT were very large and the smallest condest value is 1.1e+82, indicating unstable triangular solutions had resulted during the factorization and solution processes.

We further compared ILUTP and ILUT using large values of p, and we list the results in Table 6.3. We see that ILUTP is more robust than ILUT for solving the

[4]The WIGTO966 matrix is available from the author.

TABLE 6.2
Comparison of MDRILU and ILUT for solving the WIGTO966 *matrix.*

		MDRILU						ILUT				
p	τ	ϵ	iter	prec	totl	spar	cond	iter	prec	totl	spar	cond
25	2.0e-2	0.40	94	1.62	8.05	0.94	9.2e+5	–	–	–	–	4.9e+149
30	1.0e-3	0.37	64	3.32	8.46	1.48	1.4e+6	–	–	–	–	1.7e+99
40	1.0e-3	0.38	31	4.10	6.82	1.82	1.5e+5	–	–	–	–	1.1e+82
40	1.0e-4	0.38	33	5.99	9.14	2.05	2.9e+4	–	–	–	–	3.1e+97
50	1.0e-3	0.38	27	4.92	7.58	2.17	4.4e+4	–	–	–	–	9.9e+116
50	1.0e-4	0.38	25	7.48	10.21	2.55	2.7e+4	–	–	–	–	2.7e+91

TABLE 6.3
Comparison of ILUTP and ILUT using large values of p for solving the WIGTO966 *matrix.*

		ILUTP (original ordering)						ILUT (original ordering)				
p	τ	σ	iter	totl	prec	spar	cond	iter	prec	totl	spar	cond
50	1.0e-3	0.50	–	–	–	–	7.7e+6	–	—	–	–	9.9e+116
50	1.0e-4	0.50	–	–	–	–	3.7e+8	–	–	–	–	2.7e+91
100	1.0e-3	0.50	49	18.17	24.88	3.06	1.3e+3	–	–	–	–	9.1e+101
100	1.0e-4	0.50	34	22.90	27.48	3.08	2.3e+5	–	–	–	–	3.0e+69
300	1.0e-3	0.10	9	44.98	47.31	7.39	1.2e+4	74	51.52	71.49	7.91	1.3e+8
340	1.0e-4	0.10	7	72.3	74.99	8.90	9.4e+3	45	69.93	81.92	9.14	3.4e+7

WIGTO966 matrix. ILUT required high sparsity ratios to converge. For those converged cases, ILUTP was able to converge with fewer iterations. When we chose $p = 200, \tau =$ 1.0e-4, ILUT failed to converge, but ILUTP converged in 49 iterations with a sparsity ratio 3.06. Notice that both ILUTP and ILUT did not converge with $p = 50, \tau =$ 1.0e-3, while MDRILU could converge with these parameters (see Table 6.2). In addition, both ILUT and ILUTP are much more expensive than MDRILU to construct. We point out that the condition estimates of ILUTP are much smaller than those of ILUT. This implies that ILUTP did stabilize the ILU factorization process with a column pivoting strategy, although there was no very small pivot in the factorization. The results of Table 6.3 also show that the additional cost of implementing ILUTP (relative to ILUT) is not high in this test. However, as far as solving the WIGTO966 matrix is concerned, computing an MDRILU preconditioner is much cheaper than computing either an ILUT or an ILUTP preconditioner.

The test statistics in Tables 6.2 and 6.3 clearly indicate that MDRILU is more efficient and robust than both ILUT and ILUTP in terms of preconditioner quality and memory cost for solving the WIGTO966 matrix. Since MDRILU combines a dual reordering strategy with multilevel recursive factorization, it would be interesting to see how the dual reordering strategy of MDRILU affects the (single level) ILUT and ILUTP. To this end, we prereordered the WIGTO966 matrix using the first level dual ordering of MDRILU corresponding to the parameters used in Table 6.2. We then applied ILUT and ILUTP on the reordered matrix. The test results are given in Table 6.4. It is found that the dual reordering strategy did improve the quality (condition) of both ILUT and ILUTP (and reduced memory cost), although ILUT did not converge in any case. Convergence was obtained for ILUTP for several sets of parameters. However, the performance of ILUTP is still poorer than that of MDRILU as shown in Table 6.2. We also see that ILUTP took much more time than MDRILU to construct for solving the WIGTO966 matrix.

Table 6.5 shows the performance statistics of BILUTM of Saad and Zhang [41] for

TABLE 6.4
Performance of ILUTP and ILUT for solving the WIGTO966 *matrix, using the first level dual ordering strategy of MDRILU in Table* 6.2.

			ILUTP (MDRILU ordering)					ILUT (MDRILU ordering)				
p	τ	ϵ	iter	prec	totl	spar	cond	iter	prec	totl	spar	cond
25	2.0e-2	0.40	–	–	–	–	1.7e+8	–	–	–	–	1.4e+88
30	1.0e-3	0.37	86	14.57	20.39	0.89	4.5e+5	–	–	–	–	2.0e+49
40	1.0e-3	0.38	–	–	–	–	8.2e+5	–	–	–	–	1.1e+53
40	1.0e-4	0.38	98	32.64	40.29	1.19	8.9e+4	–	–	–	–	1.3e+62
50	1.0e-3	0.38	78	17.92	24.38	1.44	4.8e+5	–	–	–	–	6.3e+52
50	1.0e-4	0.38	81	28.07	34.86	1.47	2.4e+8	–	–	–	–	3.6e+66

TABLE 6.5
Test results of BILUTM for solving the WIGTO966 *matrix using parameters corresponding to those in Table* 6.2.

BILUTM						
p	τ	iter	prec	totl	spar	cond
25	2.0e-2	–	–	–	–	3.1e+10
30	1.0e-3	–	–	–	–	3.4e+7
40	1.0e-3	–	–	–	–	2.7e+8
40	1.0e-4	94	3.54	12.91	2.07	2.2e+6
50	1.0e-3	81	4.35	13.00	2.40	1.0e+8
50	1.0e-4	94	4.64	14.96	2.47	7.7e+6

solving the WIGTO966 matrix with the parameters used by MDRILU in Table 6.2. Although BILUTM converged for the last three sets of parameters, it did not converge for the first three sets of parameters. MDRILU is more robust than BILUTM for solving the WIGTO966 matrix. BILUTM is also more expensive than MDRILU to construct. Thus the dual reordering strategy does have advantages in multilevel factorizations.

RAEFSKY4 *matrix.* The RAEFSKY4 matrix[5] was supplied by H. Simon from Lawrence Berkeley National Laboratory (originally created by A. Raefsky from Centric Engineering). This is probably the hardest one in the total of 6 RAEFSKY matrices. Figure 6.1 shows the convergence history of the 4 preconditioners with $p = 50$ and $\tau =$ 1.0e-4. The other parameters were $\epsilon = 0.4$ for MDRILU and $\sigma = 0.03$ for ILUTP. We see that both ILUT and ILUTP did not have much convergence in 100 iterations. BILUTM converged in 94 iterations. MDRILU converged in only 13 iterations and is clearly faster than the other three preconditioners.

In Figure 6.2 we plotted the iteration counts (left part) and the values of condition estimate (right part) of the MDRILU preconditioner with different values of the threshold parameter ϵ, keeping $p = 50, \tau =$ 1.0e-4 fixed. We found that the iteration count and the condition estimate were linked to each other. A large value of condition estimate is usually accompanied by a large iteration count of MDRILU. We also see that the convergence rates of MDRILU are not very sensitive to the choice of the value of ϵ. For $0.38 \leq \epsilon \leq 0.78$, MDRILU gave very similar performance.

[5]The RAEFSKY4 matrix is available online from the University of Florida Sparse Matrix Collection at http://www.cise.ufl.edu/∼davis/sparse.

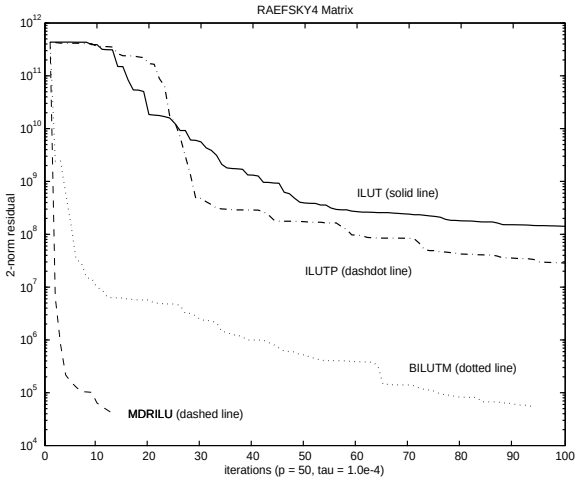


FIG. 6.1. *Convergence history of preconditioned GMRES for solving the* RAEFSKY4 *matrix.*

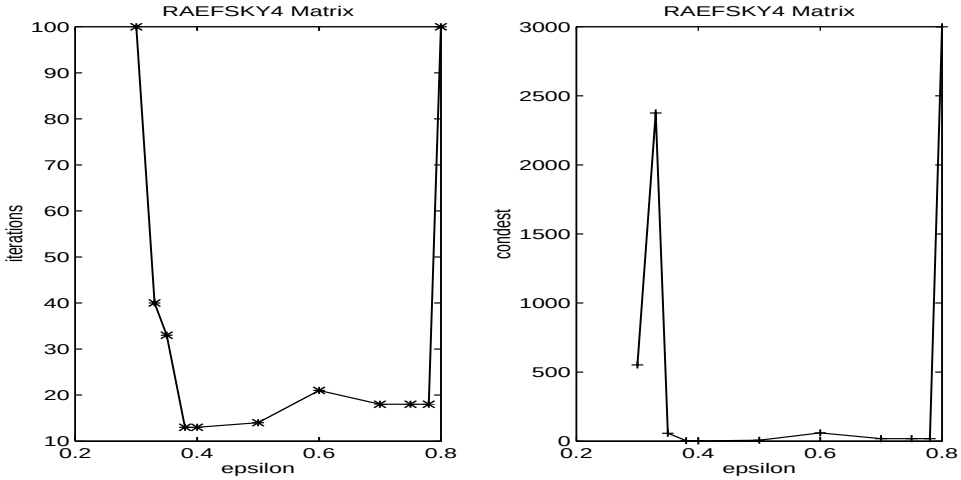


FIG. 6.2. *Iteration counts (left) and condition estimates (right) of MDRILU with different values of* ϵ *for solving the* RAEFSKY4 *matrix.*

UTM5940 *matrix.* The UTM5940 matrix[6] is the largest matrix from the TOKAMAK collection and was provided by P. Brown of Lawrence Livermore National Laboratory. Table 6.6 contains a few runs with MDRILU and ILUT with different sparsity ratios. It is clear that MDRILU is more efficient than ILUT when the sparsity ratios are low. The results are also consistent with other test results, indicating that MDRILU is able to solve this problem with less storage cost than ILUT. If sufficient memory space is available, ILUT may be efficient in certain cases. Note that if both MDRILU and ILUT converge with similar iteration counts, MDRILU is more expensive than ILUT to construct. But the total CPU times for MDRILU are less than those for ILUT in most cases.

[6]The UTM5940 matrix is available online from the MatrixMarket of the National Institute of Standards and Technology at http://math.nist.gov/MatrixMarket.

TABLE 6.6
Comparison of MDRILU and ILUT for solving the UTM5940 *matrix, and ILUTP and ILUT using the first level of MDRILU ordering.*

		MDRILU						ILUT (original ordering)				
p	τ	ϵ	iter	prec	totl	spar	cond	iter	prec	totl	spar	cond
15	1.0e-3	0.40	82	1.44	5.94	2.26	3.1e+6	–	–	–	–	1.7e+14
30	1.0e-3	0.30	63	2.48	6.56	3.66	7.4e+6	–	–	–	–	1.3e+11
30	1.0e-4	0.30	60	4.43	8.47	4.00	9.7e+6	82	1.97	8.08	3.50	1.5e+6
50	1.0e-2	0.30	63	1.36	5.28	3.42	2.9e+7	94	0.62	7.45	3.35	1.6e+6
50	1.0e-3	0.30	45	3.83	7.46	5.56	3.4e+7	78	2.08	9.26	5.22	1.7e+7
50	1.0e-4	0.30	42	7.49	11.16	6.26	2.2e+7	86	3.67	12.19	5.72	1.3e+7
			ILUTP (MDRILU ordering)					ILUT (MDRILU ordering)				
p	τ	ϵ	iter	prec	totl	spar	cond	iter	prec	totl	spar	cond
15	1.0e-3	0.40	94	1.30	6.07	1.58	4.4e+6	–	–	–	–	1.5e+7
30	1.0e-3	0.30	–	–	–	–	3.9e+7	–	–	–	–	1.1e+8
30	1.0e-4	0.30	–	–	–	–	2.9e+7	–	–	–	–	1.2e+10
50	1.0e-2	0.30	–	–	–	–	5.4e+6	–	–	–	–	1.3e+8
50	1.0e-3	0.30	93	4.98	12.64	4.48	3.7e+8	89	3.39	10.61	4.42	3.4e+8
50	1.0e-4	0.30	96	10.66	19.04	4.87	1.7e+10	73	6.98	13.02	4.84	4.2e+7

TABLE 6.7
Test results of BILUTM for solving the UTM5940 *matrix using parameters corresponding to those in Table* 6.6.

BILUTM						
p	τ	iter	prec	totl	spar	cond
15	1.0e-3	98	1.27	7.41	2.44	1.1e+7
30	1.0e-3	72	2.45	7.67	4.10	2.5e+6
30	1.0e-4	51	2.72	6.57	4.73	2.6e+7
50	1.0e-2	86	1.74	7.58	3.41	9.3e+6
50	1.0e-3	38	3.35	6.35	5.26	2.0e+7
50	1.0e-4	25	4.43	6.70	6.48	5.7e+6

The lower part of Table 6.6 contains test data of ILUT and ILUTP using the first level dual reordering strategy of MDRILU corresponding to the parameters in the upper part of Table 6.6. This time, we see that the dual reordering strategy did not improve the quality of ILUT and ILUTP. Although ILUTP converged with the first set of parameters, the overall robustness of ILUTP is still inferior to that of MDRILU.

We also used BILUTM of Saad and Zhang [41] to solve the UTM5940 matrix with the parameters corresponding to those in Table 6.6. Comparing data in Tables 6.6 and 6.7, we see that MDRILU and BILUTM performed comparably for solving this matrix. MDRILU did better when memory cost was low. BILUTM converged faster when more memory space was allowed.

Figure 6.3 shows the convergence history of MDRILU with different values of dropping tolerance τ to solve the UTM5940 matrix, keeping $p = 30$ and $\epsilon = 0.3$ fixed. We note that the number of iterations did not change very much when τ changed from 1.0e-2 to 1.0e-5 and the sparsity ratio changed from 2.67 to 4.15. It seems that MDRILU worked quite well with a relatively strict dropping tolerance.

FIDAP matrices. The FIDAP matrices[7] were extracted from the test problems provided in the FIDAP package [23]. They were generated by I. Hasbani of Fluid

[7]All FIDAP matrices are available online from the MatrixMarket of the National Institute of Standards and Technology at http://math.nist.gov/MatrixMarket.

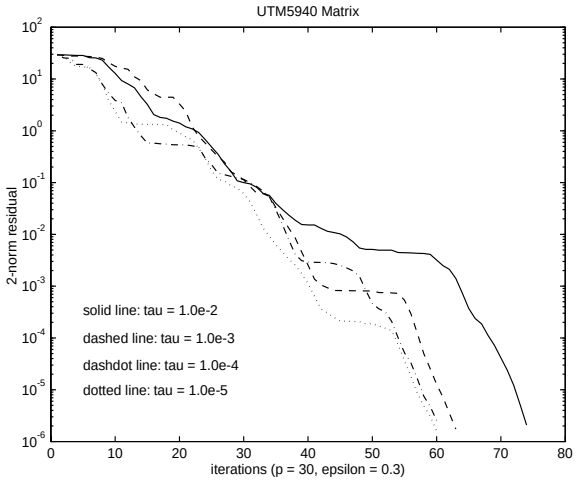


FIG. 6.3. *Convergence history of MDRILU with different values of dropping tolerance τ for solving the* UTM5940 *matrix.*

Dynamics International and B. Rackner of Minnesota Supercomputer Center. The matrices were resulted from modeling the incompressible Navier–Stokes equations and were generated using whatever solution method was specified in the input decks. However, if the penalty method was used, there is usually a corresponding FIDAPM matrix, which was constructed using a fully coupled solution method (mixed *u-p* formulation). The penalty method gives very ill-conditioned matrices, whereas the mixed *u-p* method gives indefinite, larger systems (they include pressure variables).

Many of these matrices contain small or zero diagonal values.[8] The zero diagonals are due to the incompressibility condition of the Navier–Stokes equations [9]. The large amount of zero diagonals makes these matrices indefinite. It is remarked in [6] that the FIDAP matrices are difficult to solve with ILU preconditioning techniques, which require high level of fill-in to be effective, and the performance of the preconditioners is unstable with respect to the amount of fill-in. Many of them cannot be solved by the standard BILUM preconditioner and in some cases even the construction of BILUM failed due to the occurrence of very ill-conditioned blocks. Nevertheless, some of them may be solved by the enhanced version of BILUM using singular value decomposition-based regularized inverse technique and variable block size [44].

The details of all of the largest 31 FIDAP matrices ($n > 2000$) are listed in Table 6.8 and the corresponding test results are given in Table 6.9. The second column of Table 6.9 lists the number of zero diagonals of the given matrix. In our tests, we first set $p = 20, \tau =$ 1.0e-4 and tested $\epsilon = 0.5, 0.3, 0.1, 0.01$. If none of these ϵ values showed any promise, we increased the p value or decreased the τ value. If for a given pair of (p, τ), MDRILU with a certain value of ϵ converged or showed some convergence, we adjusted the value of ϵ to get improved convergence rates if possible. However, there was no effort made to find the best parameters. We stopped refining the parameters when we found the iteration count was reasonable and the sparsity ratio was not high, or the computations took too much time in case of large matrices.

[8] The FIDAP matrices have structural zeros added on the offdiagonals to make them structurally symmetric. Structural zeros were also added to the diagonals.

Once MDRILU was tested, the same pair (p, τ) was used to test ILUTP and ILUT. For ILUTP, we varied the value of σ analogously to what we did to choose the value of ϵ.

TABLE 6.8
Description of the largest 31 *FIDAP matrices.*

Matrix	Order	Nonzeros	Description
FIDAP008	3 096	106 302	Developing flow in a vertical channel
FIDAP009	3 363	99 397	Jet impingment cooling
FIDAP010	2 410	54 816	2D flow over multiple steps in a channel
FIDAP011	16 614	1 091 362	3D steady flow, heat exchanger
FIDAP012	3 973	80 151	Flow in lid-driven wedge
FIDAP013	2 568	75 628	Axisymmetric poppet valve
FIDAP014	3 251	66 647	Isothermal seepage flow
FIDAP015	6 867	96 421	Spin up of a liquid in an annulus
FIDAP018	5 773	69 335	2D turbulent flow over a backward-facing step
FIDAP019	12 005	259 863	Developing pipe flow, turbulent
FIDAP020	2 203	69 579	Attenuation of a surface disturbance
FIDAP024	2 283	48 733	Unsymmetric forward roll coating
FIDAP026	2 163	93 749	Surface tension, thermal convection
FIDAP028	2 603	77 653	Two merging liquids with an interior interface
FIDAP029	2 870	23 754	Turbulent flow in axisymmetric U-bend
FIDAP031	3 909	115 299	Dilute species deposition on a heated plate
FIDAP035	19 716	218 308	Turbulent flow in a heated channel
FIDAP036	3 079	53 851	Chemical vapor deposition
FIDAP037	3 565	67 591	Flow of plastic in a profile extrusion die
FIDAP040	7 740	456 226	3D die-swell (square die Re = 1, Ca = ∞)
FIDAPM03	2 532	50 380	Flow past a cylinder in free stream (Re = 40)
FIDAPM07	2 065	53 533	Natural convection in a square enclosure
FIDAPM08	3 876	103 076	Developing flow in a vertical channel
FIDAPM09	4 683	95 053	Jet impingment cooling
FIDAPM10	3 046	53 842	2D flow over multiple heat sources in a channel
FIDAPM11	22 294	623 554	3D steady flow, heat exchanger
FIDAPM13	3 549	71 975	Axisymmetric poppet valve
FIDAPM15	9 287	98 519	Spin up of a liquid in an annulus
FIDAPM29	13 668	186 294	Turbulent flow is axisymmetric U-bend
FIDAPM33	2 353	23 765	Radiation heat transfer in a square cavity
FIDAPM37	9 152	765 944	Flow of plastic in a profile extrusion die

Table 6.9 shows that MDRILU can solve 27 out of the 31 largest FIDAP matrices, in which "levl" indicates the actual number of levels of dual reorderings of MDRILU. To the best of our knowledge, this is the first time that so many FIDAP matrices were solved by a single iterative technique. (20 were solved in [44], 18 in [50], 9 in [42], and 8 in [9].) In Table 6.9 the term "unstable" means that convergence was not reached in 100 iterations and the condition estimate was greater than 10^{15}. Similarly the term "inaccurate" means that convergence was not reached, but the condition estimate did not exceed 10^{15}. They are categorized according to Chow's and Saad's arguments [9]. We remark that the results of "inaccurate" or "unstable" in Table 6.9 do not indicate that ILUT or ILUTP can or cannot solve the given matrices with different parameters. The results only mean that they did not converge with the parameters that made MDRILU converge. It is worth pointing out that, in several tests, we observed that ILUTP encountered zero pivots when ILUT did not. As we remarked in section 1, the reason is that a poor pivoting sequence can occasionally trap a factorization into zero pivot even if the factorization would have succeeded without pivoting, as observed by Chow and Saad [9]. However, statistically ILUT

TABLE 6.9
Solving the FIDAP matrices by MDRILU, ILUTP, and ILUT.

				MDRILU					ILUTP		ILUT	
Matrix	zero-d	p	τ	ϵ	iter	totl	spar	levl	iter	spar	iter	spar
FIDAP008	0	90	2.0e-4	0.20	76	3.57	1.46	4	unstable		inaccurate	
FIDAP009	0	90	3.0e-4	0.35	14	0.53	0.48	3	unstable		unstable	
FIDAP010	220	90	1.0e-3	0.20	44	0.78	0.92	3	unstable		inaccurate	
FIDAP011	0	50	1.0e-4	0.10	unstable				inaccurate		inaccurate	
FIDAP012	1134	30	1.0e-4	0.10	50	2.94	2.75	3	unstable		unstable	
FIDAP013	0	300	2.0e-6	0.01	80	3.13	1.61	2	unstable		unstable	
FIDAP014	900	200	1.0e-9	0.30	87	15.92	8.91	4	40	7.89	9	6.55
FIDAP015	0	50	1.0e-4	0.10	unstable				unstable		inaccurate	
FIDAP018	0	50	1.0e-4	0.10	unstable				unstable		unstable	
FIDAP019	0	500	6.0e-6	0.30	60	19.64	4.02	7	unstable		unstable	
FIDAP020	600	20	1.0e-4	0.20	24	0.83	1.11	3	94	1.23	inaccurate	
FIDAP024	648	20	1.0e-4	0.10	46	0.97	1.62	3	45	1.82	inaccurate	
FIDAP026	457	20	1.0e-4	0.30	84	2.63	0.77	4	unstable		unstable	
FIDAP028	750	30	1.0e-4	0.20	21	1.56	1.80	3	55	1.96	59	1.94
FIDAP029	0	10	1.0e-4	0.10	4	0.20	2.15	2	3	2.20	3	2.20
FIDAP031	630	20	1.0e-4	0.20	30	0.20	1.10	3	29	1.30	20	1.30
FIDAP018	0	50	1.0e-4	0.10	unstable				unstable		inaccurate	
FIDAP036	504	20	1.0e-4	0.10	23	1.01	1.75	3	83	1.91	unstable	
FIDAP037	0	20	1.0e-4	0.01	6	0.55	0.90	2	5	1.01	5	1.01
FIDAP040	1824	70	1.0e-4	0.30	40	40.12	2.43	3	33	2.35	33	2.35
FIDAPM03	711	20	1.0e-4	0.01	32	1.18	1.72	3	57	1.65	unstable	
FIDAPM07	432	300	1.0e-4	0.20	78	7.75	6.66	2	80	7.71	inaccurate	
FIDAPM08	780	20	1.0e-4	0.10	25	2.25	1.70	3	78	2.22	unstable	
FIDAPM09	438	60	1.0e-5	0.50	32	4.57	3.74	3	unstable		unstable	
FIDAPM10	636	20	1.0e-4	0.10	28	1.21	1.79	4	29	2.13	inaccurate	
FIDAPM11	5680	300	1.0e-3	0.10	83	217.45	10.31	3	16	12.89	16	12.79
FIDAPM13	981	50	1.0e-4	0.001	24	3.33	3.69	3	34	3.29	inaccurate	
FIDAPM15	2420	50	1.0e-4	0.50	43	11.35	7.51	7	21	7.37	unstable	
FIDAPM29	2760	200	1.0e-4	0.001	28	78.80	14.38	3	11	7.47	13	7.46
FIDAPM33	620	20	1.0e-4	0.23	43	1.16	3.61	3	unstable		unstable	
FIDAPM37	0	500	1.0e-5	0.15	31	660.91	9.20	4	7	4.14	6	4.13

had more zero pivots than ILUTP did for solving all the 31 FIDAP matrices.

Although we allowed 10 levels of maximum dual reorderings to be performed, there were very few cases that 10 levels of reorderings were actually needed. With only 2 exceptions, 2 to 4 levels of dual reorderings were performed for the FIDAP matrices. Note that no case was reported for MDRILU with less than 2 levels of dual reorderings. This observation seems to suggest that the multilevel dual reordering is necessary for MDRILU to achieve good performance. In many cases, the first Schur complement matrix did not have any zero diagonal, even if the original matrix A did have many zero diagonals. We listed in Table 6.10 those matrices that did have zero diagonals in their Schur complement matrices. For all the FIDAP matrices solved by MDRILU, only the FIDAP026 matrix had 12 zero diagonals in the last Schur complement A_4. The test results show that the multilevel dual reordering strategy does have the effect of removing small and zero pivots from ILU factorizations.

TABLE 6.10
Number of zero diagonals in the Schur complement matrices.

Matrix	A_0	A_1	A_2	A_3	A_4
FIDAP026	457	44	12	12	12
FIDAPM09	1320	886	438	0	
FIDAPM10	636	4	4	0	
FIDAPM15	2420	742	0		

Remark. Ironically, the four matrices FIDAP011, FIDAP015, FIDAP018, and FIDAP035 that were not solved by MDRILU do not have any zero diagonals. (In Table 6.9, we just listed one set of parameters and the reasons why the preconditioners failed using these parameters.) They may be solved by ILUT with small values of τ. Some of them may even be solved by GMRES without preconditioning if enough iterations are allowed. We think this is because these matrices are very nonsymmetric and the preconditioned matrices were worse conditioned than the original matrices, causing GMRES iteration to converge extremely slowly. Our strong feeling in these numerical experiments is that, in general, MDRILU does not seem to work well when τ is very small. Large values of p usually improve convergence. This observation can be seen in Figure 6.4, which depicts the convergence history of MDRILU for solving the largest FIDAP matrix, FIDAPM11. We used $p = 300, \epsilon = 0.1$ and tested two values of $\tau =$ 1.0e-2 and $\tau =$ 1.0e-3. It is clear that more *accurate* (in terms of dropping tolerance) ILU factorization does not help and sometimes hampers convergence. Good values for the parameter ϵ are between 0.1 and 0.5. For most problems, the performance of MDRILU is not very sensitive to the choice of ϵ, as long as it is in the range of 0.1 and 0.5.

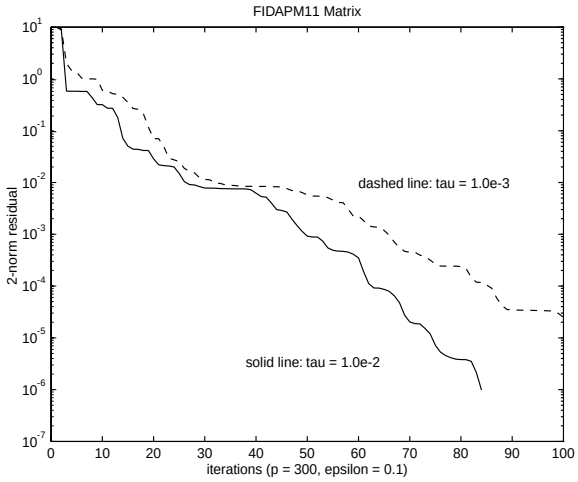


FIG. 6.4. *Convergence history of MDRILU with different values of dropping tolerance τ for solving the* FIDAPM11 *matrix.*

7. Conclusion. We have proposed a multilevel dual reordering strategy for constructing robust ILU preconditioners for solving general sparse indefinite matrices. This reordering strategy is combined with a partial ILU factorization procedure to construct recursive Schur complement matrices. The preconditioner is a multilevel ILU preconditioner. However, the constructed preconditioner (MDRILU) is different from all existing multilevel preconditioners in a fundamental concept [40, 49]. MDRILU never intends to utilize any traditional multilevel property; it uses the Schur complement approach solely for the purpose of removing small pivots. The idea used in this paper departs from traditional concepts of multilevel treatment of different error components. Thus preconditioners constructed from multilevel dual reordering strategies are more like approximate direct solvers. It is our understanding that a preconditioned iterative scheme absorbs strength from both iterative method (Krylov subspace accelerators) and direct method (preconditioners). Our idea of constructing

preconditioners from the point of view of a direct solver is therefore justified. Such a viewpoint directly pinpoints the major weakness of current iterative methods, i.e., their lack of robustness. As Gilbert and Toledo remarked [25], current iterative solvers are not as robust as the state-of-the-art direct solvers to be used as blackbox solvers. More robust preconditioners may be developed by extracting robustness strength from strategies used in modern direct solvers.

We conducted analyses on simplified model problems to find out how the size of the small diagonal elements and other elements is modified when these elements become the elements of the Schur complement matrix. We gave an upper bound on the size of general elements of the Schur complement matrix to show that their size will not grow uncontrollably if a suitable threshold reordering strategy based on the diagonal dominance measure is implemented. We also showed that under certain conditions, a zero or very small diagonal element is likely to be modified to favor a stable ILU factorization by the Schur complement procedure.

We further studied the quality of a preconditioning step. We showed that the quality of a preconditioning step is directly related to the size of both $(LU)^{-1}$ and R (the error matrix). Hence, a high quality preconditioner must have a stable ILU factorization and stable triangular solutions, as well as a small size error matrix. In other words, both accuracy and stability affect the quality of a preconditioner.

We performed numerical experiments to compare MDRILU with two popular ILU preconditioners [37] and a multilevel block ILUT preconditioner (BILUTM) [41]. Our numerical results show that MDRILU is much more robust than both ILUT and ILUTP for solving most indefinite matrices under current consideration. It also outperformed BILUTM in some tests. The most valuable advantage of MDRILU is that it can construct a sparse high quality preconditioner with low storage cost. The preconditioners computed by MDRILU are more stable than those computed by ILUT and ILUTP, thanks to the ability of MDRILU to remove small diagonal values.

Both analytic and numerical results strongly support our conclusion that the multilevel dual reordering strategy developed in this paper is a very useful strategy to construct robust ILU preconditioners for solving general sparse indefinite matrices. Due to the time and space limit, we have not tested other graph reordering algorithms in the multilevel dual reordering algorithm. Some of the popular reordering strategies such as Cuthill-McKee and reverse Cuthill-McKee algorithms may be useful in such applications to further improve the quality of the ILU preconditioner. However, we feel the robustness of MDRILU is mainly a result of using threshold tolerance reordering strategy and partial ILU factorization to remove small pivots. The difference arising from using different graph algorithms may be significant in terms of the number of iterations, but such a difference is unlikely to alter the stability problem in a systematic manner in the ILU factorization.

REFERENCES

[1] G. A. Behie and P. A. Forsyth, *Comparison of fast iterative methods for symmetric systems*, IMA J. Numer. Anal., 3 (1983), pp. 41–63.

[2] G. A. Behie and P. A. Forsyth, Jr., *Incomplete factorization methods for fully implicit simulation of enhanced oil recovery*, SIAM J. Sci. Statist. Comput, 5 (1984), pp. 543–561.

[3] M. Benzi, D. B. Szyld, and A. van Duin, *Orderings for incomplete factorization preconditioning of nonsymmetric problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1652–1670.

[4] C. I. W. Brand, *An incomplete-factorization preconditioning using red-black ordering*, Numer. Math., 61 (1992), pp. 433–454.

[5] 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.
[6] A. Chapman, Y. Saad, and L. Wigton, *High-order ILU preconditioners for CFD problems*, Internat. J. Numer. Methods Fluids, 33 (2000), pp. 767–788.
[7] M. P. Chernesky, *On preconditioned Krylov subspace methods for discrete convection-diffusion problems*, Numer. Methods Partial Differential Equations, 13 (1997), pp. 321–330.
[8] E. Chow and M. A. Heroux, *An object-oriented framework for block preconditioning*, ACM Trans. Math. Software, 24 (1998), pp. 159–183.
[9] E. Chow and Y. Saad, *Experimental study of ILU preconditioners for indefinite matrices*, J. Comput. Appl. Math., 86 (1997), pp. 387–414.
[10] S. S. Clift and W.-P. Tang, *Weighted graph based ordering techniques for preconditioned conjugate gradient methods*, BIT, 35 (1995), pp. 30–47.
[11] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang, *Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 944–961.
[12] M. A. Delong and J. M. Ortega, *SOR as a preconditioner*, Appl. Numer. Math., 18 (1995), pp. 431–440.
[13] G. Di Lena and D. Trigiante, *Stability and spectral properties of incomplete factorization*, Japan J. Appl. Math., 7 (1990), pp. 145–163.
[14] S. Doi, *On parallelism and convergence of incomplete LU factorizations*, Appl. Numer. Math., 7 (1991), pp. 417–436.
[15] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, New York, 1986.
[16] I. S. Duff and G. A. Meurant, *The effect of reordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–657.
[17] L. C. Dutto, *The effect of reordering on the preconditioned GMRES algorithm for solving the compressible Navier-Stokes equations*, Internat. J. Numer. Methods Engrg., 36 (1993), pp. 457–497.
[18] H. C. Elman, *A stability analysis of incomplete LU factorization*, Math. Comp., 47 (1986), pp. 191–217.
[19] H. C. Elman, *Relaxed and stabilized incomplete factorization for nonselfadjoint linear systems*, BIT, 29 (1989), pp. 890–915.
[20] H. C. Elman and E. Agron, *Ordering techniques for the preconditioned conjugate gradient method on parallel computers*, Comput. Phys. Comm., 53 (1989), pp. 253–269.
[21] H. C. Elman and G. H. Golub, *Iterative methods for cyclically reduced nonselfadjoint linear systems*, Math. Comp., 54 (1990), pp. 671–700.
[22] H. C. Elman and G. H. Golub, *Iterative methods for cyclically reduced nonselfadjoint linear systems.* II, Math. Comp., 56 (1991), pp. 215–242.
[23] M. Engelman, *FIDAP: Examples Manual, Revision* 6.0, Tech. report, Fluid Dynamics International, Evanston, IL, 1991.
[24] J. A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[25] J. R. Gilbert and S. Toledo, *An assessment of incomplete-LU preconditioners for nonsymmetric linear systems*, Tech. report, Xerox Palo Alto Research Center, Palo Alto, CA, 1997.
[26] D. S. Kershaw, *On the problem of unstable pivots in the incomplete LU-conjugate gradient method*, J. Comput. Phys., 38 (1980), pp. 114–123.
[27] H. P. Langtangen, *Conjugate gradient methods and ILU preconditioning of non-symmetric matrix systems with arbitrary sparsity patterns*, Internat. J. Numer. Methods Fluids, 9 (1989), pp. 213–233.
[28] J. W.-H. Liu and A. H. Sherman, *Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices*, SIAM J. Numer. Anal., 13 (1976), pp. 198–213.
[29] M.-M. Magolu, *Ordering strategies for modified block incomplete factorizations*, SIAM J. Sci. Comput., 16 (1995), pp. 378–399.
[30] T. A. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Math. Comput., 34 (1980), pp. 473–497.
[31] 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.

[32] A. Messaoudi, *On the stability of the incomplete LU-factorization and characterizations of H-matrices*, Numer. Math., 69 (1995), pp. 321–331.
[33] Y. Notay, *Ordering Methods for Approximate Factorization Preconditioning*, Tech. Report IT/IF/14-11, Universite Libre de Bruxelles, Brussels, Belgium, 1993.
[34] J. M. Ortega, *Orderings for conjugate gradient preconditionings*, SIAM J. Optim., 1 (1991), pp. 565–582.
[35] Y. Saad, *ILUT: A dual threshold incomplete LU preconditioner*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
[36] Y. Saad, *ILUM: A multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput., 17 (1996), pp. 830–847.
[37] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, New York, 1996.
[38] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[39] Y. Saad, M. Sosonkina, and J. Zhang, *Domain decomposition and multi-level type techniques for general sparse linear systems*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X. Cai, eds., Contemp. Math. 218, AMS, Providence, RI, 1998, pp. 174–190.
[40] Y. Saad and J. Zhang, *BILUM: Block versions of multielimination and multilevel ILU preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 20 (1999), pp. 2103–2121.
[41] Y. Saad and J. Zhang, *BILUTM: A domain-based multilevel block ILUT preconditioner for general sparse matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 279–299.
[42] Y. Saad and J. Zhang, *Diagonal threshold techniques in robust multi-level ILU preconditioners for general sparse linear systems*, Numer. Linear Algebra Appl., 6 (1999), pp. 257–280.
[43] Y. Saad and J. Zhang, *A multi-level preconditioner with applications to the numerical simulation of coating problems*, in Iterative Methods in Scientific Computing II, D. R. Kincaid and A. C. Elster, eds., IMACS, New Brunswick, NJ, 1999, pp. 437–449.
[44] Y. Saad and J. Zhang, *Enhanced multilevel block ILU preconditioning strategies for general sparse linear systems*, J. Comput. Appl. Math., to appear.
[45] S. A. Sauter, *On the stability of the incomplete Cholesky decomposition for a singular perturbed problem, where the coefficient matrix is not an M-matrix*, Numer. Linear Algebra Appl., 2 (1995), pp. 17–28.
[46] H. D. Simon, *Incomplete LU preconditioners for conjugate-gradient-type iterative methods*, in Proceedings of the 1985 SPE Resevoir Simulation Symposium, Richardson, TX, 1985, Society of Petroleum Engineers, pp. 387–396.
[47] H. A. van der Vorst, *Iterative solution methods for certain sparse linear systems with a nonsymmetric matrix arising from PDE-problems*, J. Comput. Phys., 44 (1981), pp. 1–19.
[48] H. A. van der Vorst, *Stabilized incomplete LU-decompositions as preconditionings for the Tchebycheff iteration*, in Preconditioning Methods: Analysis and Applications, in Topics in Comput. Math. 1, Gordon and Breach, New York, 1983, pp. 243–263.
[49] J. Zhang, *A grid based multilevel incomplete LU factorization preconditioning technique for general sparse matrices*, Appl. Math. Comput., to appear.
[50] J. Zhang, *Preconditioned Krylov subspace methods for solving nonsymmetric matrices from CFD applications*, Comput. Methods Appl. Mech. Engrg., 189 (2000), pp. 825–840.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 948–964

CONVERGENCE OF PSEUDOCONTRACTIONS AND APPLICATIONS TO TWO-STAGE AND ASYNCHRONOUS MULTISPLITTING FOR SINGULAR M-MATRICES*

YANGFENG SU† AND AMIT BHAYA‡

Abstract. Pseudocontractions, which are generalizations of paracontractions in the linear case, are introduced in this paper in order to study the convergence of nonstationary iterative methods for linear systems in which the coefficient matrices are singular M-matrices. A general convergence theorem for pseudocontractions is developed. This theorem is used to analyze the convergence of two nonstationary parallel iterations: two-stage multisplitting iterations and asynchronous multisplitting iterations for singular M-matrices, without other contractivity conditions on the iteration matrices.

Key words. pseudocontractions, paracontractions, two-stage iteration, asynchronous iteration, multisplitting, nonnegative matrices, singular M-matrices

AMS subject classifications. 15A48, 65F10, 65Y05

PII. S0895479898339414

1. Introduction.

Singular M-systems, which are linear systems of equations $Ax = b$, with A a singular M-matrix, may appear in many applications such as elliptic equations with periodic boundary conditions, finite Markov chains, etc. [6]. Stationary iterative methods to solve singular M-systems have been well studied; see, e.g., [6, 36]. For nonstationary iterative methods, in which the iteration operators change during the iterative procedure, for example, two-stage iterative methods and asynchronous iterative methods (see sections 3 and 4 below, respectively, for details), some assumptions on these operators have been made in the literature to guarantee the convergence. Bru, Elsner, and Neumann [11] assumed that the iterative operators are *paracontractive*, Migallón, Penadés, and Szyld [30] assumed that the operators are uniformly contractive.

In this paper, a new property of operators, called *pseudocontractivity*, is proposed. It is shown that pseudocontractivity is a generalization of the paracontractivity property in the linear case. A general convergence theorem for pseudocontractive iterations is proved in section 2. In section 3 it is proved that the product of iteration matrices associated with a class of stationary iterative methods is pseudocontractive and therefore that this class of methods is convergent. In sections 4 and 5, this theory is used to analyze two parallel nonstationary iterative methods for singular M-systems, specifically, nonstationary two-stage multisplitting methods and asynchronous multisplitting methods, and, under reasonable conditions, convergence results are obtained. No other contractivity condition on the iteration operators is required.

*Received by the editors May 29, 1998; accepted for publication (in revised form) by I. Ipsen October 2, 2000; published electronically January 19, 2001. This research was partially supported by CNPq, the Brazilian National Council for Scientific and Technological Research, as well as a grant from the PRONEX Program of MCT.

http://www.siam.org/journals/simax/22-3/33941.html

†Department of Mathematics, Fudan University, Shanghai, China (yfsu@fudan.edu.cn). This research was carried out while the first author was visiting the second on a fellowship awarded by CNPq, the Brazilian National Council for Scientific and Technological Research.

‡Federal University of Rio de Janeiro, Department of Electrical Engineering, PEE/COPPE/UFRJ, P.O. Box 68504, RJ 21945-970, Brazil (amit.bhaya@na-net.ORNL.gov). The second author was supported by CNPq, PRONEX, and CAPES.

2. Pseudocontractive operators and a general convergence theorem. Let X^* be a nonempty closed convex subset of $\mathbb{R}^n$, and let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. For any vector $x \in \mathbb{R}^n$, $y^* \in X^*$ is a *projection* vector of x onto X^* if

$$\|x - y^*\| = \min_{y \in X^*} \|x - y\|.$$

Remark. Since X^* is closed, the minimum is always attained. The projection vector may be not unique. For example, let $\|\cdot\| = \|\cdot\|_\infty$, $X^* = \{x \in \mathbb{R}^2 \mid \|x\| \le 1\}$, $x = (2,0)^T$, then all vectors $(1,a)^T$ with $a \in [-1,1]$ are projection vectors of x onto X^*. We use $P(x)$ to denote an arbitrary but fixed projection vector of x and $\operatorname{dist}(x, X^*)$ to denote $\|x - P(x)\|$. Note that $\operatorname{dist}(x, X^*)$ is independent of the choice of $P(x)$.

Let T be an operator on $\mathbb{R}^n$. It is *nonexpansive* (with respect to $\|\cdot\|$ and X^*) if

$$\|Tx - x^*\| \le \|x - x^*\| \quad \text{for all } x \in \mathbb{R}^n, x^* \in X^* \tag{2.1}$$

and *pseudocontractive* (with respect to $\|\cdot\|$ and X^*) if, in addition,

$$\operatorname{dist}(Tx, X^*) < \operatorname{dist}(x, X^*) \quad \text{for all } x \notin X^*. \tag{2.2}$$

We use $\mathcal{T}$ to denote the set of all pseudocontractive operators.

PROPOSITION 2.1. *If T is pseudocontractive with respect to $\|\cdot\|$ and X^*, then X^* is the set of all fixed points of T.*

Proof. For any fixed $x^* \in X^*$, in (2.1), let $x = x^*$, and thus we have $Tx^* = x^*$, which means that all points in X^* are fixed points of T. For any $x \notin X^*$, from (2.2), $Tx - P(Tx) \ne x - P(x)$, therefore, $Tx \ne x$, i.e., x is not a fixed point of T. □

Example 1. Let $T \in \mathbb{R}^{n\times n}$, $X^* = \{\alpha e \mid \alpha \in \mathbb{R}\}$, and let $e \in \mathbb{R}^n$ denote the vector with all components equal to 1. Then T is pseudocontractive with respect to X^* and the infinity norm $\|\cdot\|_\infty$ if and only if $Te = e$ and for any $x \in \mathbb{R}^n$ such that $\min_i x_i < \max_i x_i$, $\max_i (Tx)_i - \min_i (Tx)_i < \max_i x_i - \min_i x_i$.

Remark (paracontractivity versus pseudocontractivity). Paracontractive operators have been used mainly in the study of systems with multiple solutions; see, for example, [32, 14, 15, 11, 35, 8]. An operator is *paracontractive* if

$$\|Tx\| \le \|x\| \quad \text{for all } x \in \mathbb{R}^n$$

and equality holds if and only if $Tx = x$. In the case that T is linear, if T is paracontractive, then

$$\|Tx - x^*\| = \|T(x - x^*)\| < \|x - x^*\| \quad \text{for all } x \notin X^*, x^* \in X^*,$$

where X^* is the subspace consisting of all fixed points of T. Thus, T is pseudocontractive. So in the linear case, pseudocontractive operators are generalizations of paracontractive ones. But the converse is not true. Consider the following inequalities:

$$\|Tx - P(Tx)\| \le \|Tx - P(x)\| \le \|x - P(x)\| \text{ for } x \notin X^*.$$

Paracontractivity requires the second inequality to be strict, while pseudocontractivity requires any one of these two inequalities to be strict.

Example 2. For the operator

$$T = \begin{pmatrix} .5 & .5 & 0 \\ .25 & .5 & .25 \\ 0 & .5 & .5 \end{pmatrix},$$

the norm, the vector e, and the set X^* are the same as in Example 1. For any x, $P(x) = 0.5(\max_i x_i + \min_i x_i)e$. For $x = (2, 2, 1)^T$, $Tx = (2, 1.75, 1.5)^T$, $P(x) = 1.5e$, and $P(Tx) = 1.75e$. Thus the first inequality in the equation above is strict while the second one is an equality. So this operator is pseudocontractive, not paracontractive.

Another simple property of pseudocontractions is given below.

PROPOSITION 2.2. *Let T_i be a set of nonexpansive or pseudocontractive operators (with respect to the same norm and the same set X^*). A product of any number of operators from this set that contains at least one pseudocontractive operator is pseudocontractive.*

Proof. Let T_1 be pseudocontractive and T_2 be nonexpansive. First consider the case that $T = T_1T_2$. For $x \notin X^*$, if $T_2x \in X^*$, then $\|Tx - P(Tx)\| = 0 < \|x - P(x)\|$. Suppose $T_2x \notin X^*$; from the definition of the operator P, it follows that

$$\begin{aligned}\|Tx - P(Tx)\| &= \|T_1T_2x - P(T_1T_2x)\| \\ &< \|T_2x - P(T_2x)\| \\ &\le \|T_2x - P(x)\| \\ &\le \|x - P(x)\|.\end{aligned}$$

Thus, T is pseudocontractive. Now consider the case that $T = T_2T_1$.

$$\|Tx - P(Tx)\| \le \|Tx - P(T_1x)\| \le \|T_1x - P(T_1x)\|.$$

Since T_1 is pseudocontractive, T is also pseudocontractive. ∎

Note that this proposition implies that if a product of operators is pseudocontractive, then only one of its factors need be pseudocontractive; the others may be nonexpansive.

In the following theorem, the operators are not necessarily linear, although in our later applications the operators are linear.

THEOREM 2.3. *Let $\{T_k\}$ be a sequence of nonexpansive operators (with respect to $\|\cdot\|$ and X^*), and let there exist a subsequence $\{T_{k_i}\}$ which converges to $T \in \mathcal{T}$. If T is pseudocontractive and uniformly Lipschitz continuous, then for any initial vector $x(0)$, the sequence of vectors*

$$x(k+1) = T_kx(k), \quad k = 0, 1, 2, \ldots,$$

converges to some $x^* \in X^*$.

Proof. Consider the subsequence of vectors $\{x(k_i)\}_{i=0}^{\infty}$ of the sequence $\{x(k)\}_{k=0}^{\infty}$. As T_k is nonexpansive, this subsequence is bounded, and it contains a convergent subsequence which, without loss of generality, can be taken to be $\{x(k_i)\}_{i=0}^{\infty}$ itself. Assume therefore that

$$\lim_{i\to\infty} x(k_i) = \xi.$$

If $\xi \in X^*$, as all T_i are nonexpansive,

$$\|x(k+1) - \xi\| \le \|x(k) - \xi\|, \quad k = 0, 1, 2 \ldots,$$

and we are done.

Suppose $\xi \notin X^*$, and therefore $\|\xi - P(\xi)\| > 0$. As T is pseudocontractive,

$$\beta := \frac{\|T\xi - P(T\xi)\|}{\|\xi - P(\xi)\|} < 1.$$

For arbitrary fixed $\varepsilon > 0$ small enough, there exists an integer k_ε such that

$$\|x(k_i) - \xi\| \le \varepsilon, \quad \|T_{k_i} - T\| \le \varepsilon \quad \text{for all } i \text{ such that } k_i \ge k_\varepsilon.$$

Consider i: $k_i \ge k_\epsilon$.

$$\begin{aligned}
&\|x(k_i+1) - P(x(k_i+1))\| \\
&\le \|x(k_i+1) - P(T\xi)\| \\
&= \|T_{k_i}x(k_i) - P(T\xi)\| \\
&\le \|T_{k_i}x(k_i) - Tx(k_i)\| + \|Tx(k_i) - T\xi\| + \|T\xi - P(T\xi)\| \qquad (2.3)\\
&\le \varepsilon\|x(k_i)\| + \varepsilon\|T\| + \beta\|\xi - P(\xi)\| \qquad (2.4)\\
&\le \beta\|\xi - P(\xi)\| + C\varepsilon,
\end{aligned}$$

where, from (2.3) to (2.4), we use the convergence of T_{k_i} for the first term, the uniform Lipschitz continuity of T for the second term ($\|T\|$ is used to denote the Lipschitz constant), the pseudocontractive property of T for the third term, and C is a positive constant scalar. As all T_k are nonexpansive, for $k \ge 0$,

$$\begin{aligned}
&\|x(k+1) - P(x(k+1))\| \le \|x(k+1) - P(x(k))\| \\
&= \|Tx(k) - P(x(k))\| \le \|x(k) - P(x(k))\|,
\end{aligned}$$

therefore,

$$\|x(k_{i+1}) - P(x(k_{i+1}))\| \le \|x(k_i+1) - P(x(k_i+1))\| \le \beta\|\xi - P(\xi)\| + C\varepsilon.$$

On the other hand,

$$\begin{aligned}
\|\xi - P(\xi)\| &\le \|\xi - P(x(k_{i+1}))\| \\
&\le \|\xi - x(k_{i+1})\| + \|x(k_{i+1}) - P(x(k_{i+1}))\| \\
&\le \varepsilon + \beta\|\xi - P(\xi)\| + C\varepsilon,
\end{aligned}$$

i.e., for any ε small enough,

$$\|\xi - P(\xi)\| \le \frac{C+1}{1-\beta}\,\varepsilon.$$

This contradicts $\|\xi - P(\xi)\| > 0$. □

This theorem is applied to prove the convergence of two-stage multisplitting iteration algorithms and asynchronous multisplitting iteration algorithms for singular M-matrices in the subsequent sections. The following corollary relates pseudocontractivity to the existence of limits of powers of a matrix.

COROLLARY 2.4. *If T is a pseudocontractive matrix, then $\lim_{n\to\infty} T^n$ exists.*

Proof. In the matrix case, $\lim_{n\to\infty} T^n$ exists if and only if for all x, $\lim_{n\to\infty} T^n x$ exists. The conclusion is now drawn from the above theorem. □

Theoretically, the convergence of the powers of a matrix is equivalent to the existence of a vector norm such that the matrix is paracontractive with respect to

this norm. However, in practice, it is usually necessary to prove that a matrix has some contractive property with respect to a given norm, for example, with respect to the 2-norm in the case of symmetric matrices [32], or the weighted infinity norm in the case of nonnegative matrices. By the remark following Example 1, if a matrix is paracontractive with respect to a given norm, then it is always pseudocontractive with respect to this norm. On the other hand, if a matrix is not paracontractive with respect to some norm, it may still be pseudocontractive with respect to this norm. In this paper, it is proved that the product of a sufficiently large number (at most $n-1$, where n is the order of the matrix) of nonnegative matrices, which are induced from (possibly different) weak regular splittings of a singular M-matrix, is pseudocontractive. This is used to prove the convergence of some parallel iterative methods for singular M-matrices.

3. Pseudocontractive operators and weak regular splittings of an irreducible singular M-matrix. Let B be a nonnegative matrix (denoted $B \geq 0$), i.e., each element of B is nonnegative. From nonnegative matrix theory, see, e.g., [6, 42], $\rho(B)$, the spectral radius of B, is an eigenvalue of B, and there exists a nonnegative eigenvector, which is termed the Perron vector of B, associated to it. If B is irreducible, there is only one eigenvalue equal to $\rho(B)$ and the Perron vector is positive (componentwise, denoted as $\gg 0$) and unique (up to a scalar factor). A matrix $A \in \mathbb{R}^{n\times n}$ is a singular M-matrix if there exists a nonnegative matrix B such that $A = \rho(B)I - B$. Therefore, if A is an irreducible singular M-matrix, there exists a unique (up to a scalar factor) v which is positive such that $Av = 0$. In what follows, the following assumption will always be made.

Assumption. The vector v, referred to above, is equal to e, the vector with all components equal to 1.

Remark. This assumption makes our notation simpler and our demonstrations more intuitive: for example, $Av = 0$ means that all sums of elements in the same row of A are equal to zero. Furthermore, it entails no loss of generality. For, if $v \neq e$, then $A' = D^{-1}AD$ with $D = \mathrm{diag}(v)$ is a singular M-matrix as well, satisfying $A'e = 0$. If the same similarity transformation is applied to all the other matrices involved, then all the important properties assumed in this paper are preserved. For example, if $A = M - N$ is a weak regular or regular splitting (as defined below), then $A' = M' - N'$ is weak regular or regular as well. Consequently, the iterates $x(k)$ of the nonstationary two-stage multisplitting iteration described in section 4 (with splittings $A = M - N$, $M = F_l - N_l$) are related to those of the same algorithm with splittings $A' = M' - N'$, $M' = F'_l - G'_l$ through $x'(k) = D^{-1}x(k)$, provided that $x'(0) := D^{-1}x(0)$. Thus, the convergence proved in Theorem 4.2 for the special case $v = e$ actually also holds in the general case. Exactly the same situation arises for Theorem 5.1.

A splitting of the matrix $A : A = M - N$ is *weak regular* if M is nonsingular, $M^{-1} \geq 0$, and $M^{-1}N \geq 0$, and is *regular* if $M^{-1} \geq 0$ and $N \geq 0$.

PROPOSITION 3.1. *Let A be an irreducible singular M-matrix, the splitting $A = M - N$ be weak regular, and $T = M^{-1}N$. Then either T is irreducible, or there exists a permutation matrix P_e such that*

$$P_e{}^T T P_e = \begin{pmatrix} T_{11} & \\ T_{21} & T_{22} \end{pmatrix},$$

where T_{11} is irreducible, $\rho(T_{11}) = 1$, and $\rho(T_{22}) < 1$.

Proof. Suppose that T is reducible. For $v \gg 0$, $Av = 0$, we have $Tv = v$, therefore, from [29, p. 728], there exists a permutation matrix P_e such that

$$P_e{}^T T P_e = \begin{pmatrix} Q_{11} & & & \\ & \ddots & & \\ & & Q_{ii} & \\ Q_{i+1,1} & \cdots & \cdots & Q_{i+1,i+1} \end{pmatrix},$$

where $Q_{11}, \ldots, Q_{ii}$ are irreducible, $i \geq 1$, $\rho(Q_{11}) = \cdots = \rho(Q_{ii}) = 1$, $\rho(Q_{i+1,i+1}) < 1$, and $Q_{i+1,i+1}$ may be missing. If $i \geq 2$, let $u = P_e{}^T v = (u_1^T, \ldots, u_i^T, u_{i+1}^T)^T$, then any vector

$$\tilde{u} = \left(\alpha_1 u_1^T, \ldots, \alpha_i u_i^T, \left[(I - Q_{i+1,i+1})^{-1} \sum_{j=1}^{i} Q_{i+1,j} \alpha_j u_j \right]^T \right)^T, \quad \alpha_j > 0,$$

is also the Perron vector of $P_e{}^T T P_e$, which contradicts the fact that T has a unique (up to a scalar factor) Perron vector. Therefore $i = 1$. ∎

We use $S(T)$ to denote the set of all row indices of T such that after the permutation in the above proposition, these rows become the rows of T_{11}. For any $x \in \mathbb{R}^n$, denote

$$\overline{x} \equiv \max_i x_i, \quad \underline{x} \equiv \min_i x_i.$$

PROPOSITION 3.2. *Let x be a vector such that $\underline{x} < \overline{x}$, A be an irreducible singular M-matrix such that $Ae = 0$ with $e = (1, 1, \ldots, 1)^T$, the splitting $A = M - N$ be weak regular, $T \equiv (t_{ij}) = M^{-1}N$ with $t_{ii} > 0$ for $1 \leq i \leq n$, $y = Tx$. Then we have the following:*

(i) *For all components of y,*

$$\underline{x} \leq y_i \leq \overline{x}, \quad 1 \leq i \leq n,$$

$$\{i \,|\, y_i = \underline{x}\} \subset \{i \,|\, x_i = \underline{x}\}, \tag{3.1}$$

$$\{i \,|\, y_i = \overline{x}\} \subset \{i \,|\, x_i = \overline{x}\}. \tag{3.2}$$

(ii) *Equation*

$$\{i \,|\, y_i = \underline{x}\} = \{i \,|\, x_i = \underline{x}\} \tag{3.3}$$

holds if and only if

$$\{i \,|\, y_i = \underline{x}\} = \{i \,|\, x_i = \underline{x}\} = S(T);$$

similarly, equation

$$\{i \,|\, y_i = \overline{x}\} = \{i \,|\, x_i = \overline{x}\} \tag{3.4}$$

holds if and only if

$$\{i \,|\, y_i = \overline{x}\} = \{i \,|\, x_i = \overline{x}\} = S(T).$$

(iii) *The number of the elements in set $\{i \,|\, y_i = \underline{x}$ or $y_i = \overline{x}\}$ is at least one less than the number of the elements in $\{i \,|\, x_i = \underline{x}$ or $x_i = \overline{x}\}$.*

Proof. Write y_i as

$$y_i = \sum_{x_j=\underline{x}} t_{ij}x_j + \sum_{x_j=\overline{x}} t_{ij}x_j + \sum_{\underline{x}<x_j<\overline{x}} t_{ij}x_j, \quad i = 1, \ldots, n. \tag{3.5}$$

Using $t_{ii} > 0$ and $Te = e$, we have

$$\begin{array}{ll} \underline{x} \le y_i < \overline{x} & \text{for } i \text{ such that } x_i = \underline{x}; \\ \underline{x} < y_i < \overline{x} & \text{for } i \text{ such that } \underline{x} < x_i < \overline{x}; \\ \underline{x} < y_i \le \overline{x} & \text{for } i \text{ such that } x_i = \overline{x}. \end{array}$$

This is part (i) of this proposition.

From (3.5), $\{i \,|\, y_i = \underline{x}\} = \{i \,|\, x_i = \underline{x}\}$ if and only if

$$\sum_{j:\ x_j>\underline{x}} t_{ij} = 0 \quad \text{for all } i \text{ such that } x_i = \underline{x} \tag{3.6}$$

or equivalently,

$$\sum_{j:\ x_j=\underline{x}} t_{ij} = 1 \quad \text{for all } i \text{ such that } x_i = \underline{x}. \tag{3.7}$$

We use $\widetilde{T}_{11}$ to denote the principal submatrix of T consisting of those rows and columns whose indices belong to $\{i \,|\, y_i = \underline{x}\}$. From (3.7), e is a Perron vector of $\widetilde{T}_{11}$. Using Proposition 3.1, we have $\{i \,|\, x_i = \underline{x}\} = S(T)$; thus (3.3) holds. Similar arguments are valid for the necessary and sufficient condition of (3.4).

As $\underline{x} < \overline{x}$, equalities (3.3) and (3.4) cannot hold simultaneously, therefore, at least one of the inclusions (3.1), (3.2) is strict, and part (iii) is also proved. ∎

Remark. If $t_{ii} = 0$, then for any $\omega : 0 < \omega < 1$, all diagonal elements of $(1-\omega)I + \omega T$, the Jacobi extrapolating matrix of T, are positive. Let T be an irreducible nonnegative matrix with $\rho(T) = 1$. If at least one diagonal element of T is positive (thus T is primitive), then T is semiconvergent, i.e., $\lim_{k\to\infty} T^k$ exists, the Jacobi iterative method converges; see [6, Chapter 2]. If this is not satisfied, T may not be semiconvergent, for example,

$$T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

PROPOSITION 3.3. *Let $A \in \mathbb{R}^{n\times n}$ be an irreducible singular M-matrix such that $Ae = 0$, let $A = M_k - N_k$, $k = 1, \ldots, n-1$, be arbitrary weak regular splittings of A, and let $T_k = M_k^{-1}N_k$, with all diagonal elements of T_k positive. Then, with respect to $\|\cdot\|_\infty$ and $X^* = \{\alpha e \,|\, \alpha \in \mathbb{R}\}$,*

$$T = T_{n-1}T_{n-2}\cdots T_1$$

is pseudocontractive.

Proof. For any $x \notin X^*$, i.e., $\underline{x} < \overline{x}$, denote $y = Tx$, as $Te = e$, it is easy to prove that T is nonexpansive. Applying part (iii) of Proposition 3.2 repeatedly ($n-1$ times), we have that at least one set of $\{i \,|\, y_i = \underline{x}\}$ and $\{i \,|\, y_i = \overline{x}\}$ is empty. If, say, $\{i \,|\, y_i = \underline{x}\}$ is empty, then $y_i > \underline{x}$ for all $1 \le i \le n$, and furthermore

$$\|y - P(y)\| = \frac{\max_i y_i - \min_i y_i}{2} \le \frac{\overline{x} - \min_i y_i}{2} < \frac{\overline{x} - \underline{x}}{2} = \|x - P(x)\|;$$

therefore T is pseudocontractive. ∎

Remark. This T may be not paracontractive with respect to $\|\cdot\|_\infty$. For example, for $n = 3$, let

$$A = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix},\ M = \begin{pmatrix} 4 & -2 & -2 \\ 0 & 4 & -2 \\ 0 & -2 & 4 \end{pmatrix},\ N = M - A,$$

$$T_1 = T_2 = M^{-1}N = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix},\ T = T_2T_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.75 & 0.25 & 0 \\ 0.75 & 0 & 0.25 \end{pmatrix}.$$

T_k and T satisfy all conditions in Proposition 3.3, so T is pseudocontractive. For $x = (2, 0, 0)^T \notin X^*$, $Tx = (2, 1.5, 1.5)^T \neq x$, the projection vectors $P(x) = (1, 1, 1)^T$, $P(Tx) = (1.75, 1.75, 1.75)^T$, thus

$$0.25 = \|Tx - P(Tx)\| < \|x - P(x)\| = 1, \quad \text{but } \|x\| = \|Tx\|.$$

Thus, T is pseudocontractive, but not paracontractive. In [11] it was shown that T positive guarantees that T is paracontractive. For the matrix T in Example 2, T^2 is positive. Here we do not require T to be positive or even to be nonnegative irreducible.

4. Two-stage multisplitting iterative methods. Since multisplitting iteration was first proposed by O'Leary and White [34] to solve systems of linear equations in a parallel computer, it has been studied for many types of systems, for example, nonsingular M-systems [1, 13, 33], H-systems [37], SPD-systems [31, 44], nonlinear systems [17, 16], linear or nonlinear complementarity problems [2, 4, 28], etc., combining with many kinds of methods, e.g., extrapolating methods [43, 18, 3], CG methods with preconditioning [9, 24, 23], two-stage methods [12, 25, 40], etc. However, multisplitting iteration should be viewed more as an analysis tool to study a variety of block iterative methods, including the Schwarz method, rather than as a competitive computational method. In this section, we discuss a nonstationary two-stage multisplitting method for solving singular M-systems and its convergence from this point of view. In the next section, another nonstationary multisplitting method called the asynchronous multisplitting method is discussed.

Let A be an singular M-matrix, $A = M - N$ be a weak regular splitting, $M = F_l - G_l$ be r splittings, E_l be r nonnegative diagonal matrices such that $\sum_{l=1}^r E_l = I$. $(F_l, G_l, E_l)_{l=1}^r$ is called a weak regular multisplitting of M if the r splittings $M = F_l - G_l$, $l = 1, \ldots, r$, are weak regular. The following nonstationary two-stage multisplitting method for solving singular M-systems was given in [30].

ALGORITHM (nonstationary two-stage multisplitting). Given the initial vector $x(0)$, and a sequence of numbers of inner iterations $q(l, k), 1 \le l \le r, k = 1, 2, \ldots,$

For $k = 1, 2, \ldots$, until convergence.
  For $l = 1$ to r % outer stage
    $y(k, 0) = x(k - 1)$
    For $j = 1$ to $q(l, k)$ % inner stage
      $F_l y(l, j) = G_l y(l, j - 1) + N x(k - 1)$
  $x(k) = \sum_{l=1}^r E_l y(l, q(l, k))$.

The parallel implementation of this algorithm is obvious. Before we state our convergence theorem for it, we give a lemma.

LEMMA 4.1. *Let A be an irreducible singular M-matrix, $A = M - N$ be a weak regular splitting, and $(F_l, G_l, E_l)_{l=1}^r$ be a weak regular multisplitting of M. Then the sequence of iterative vectors $x(k)$ in the nonstationary two-stage multisplitting iteration satisfies*

$$x(k) = T_{k-1}x(k-1),$$

where

$$T_{k-1} = \sum_{l=1}^{r} E_l \left[R_l^{q(l,k)} + (I + R_l + \cdots + R_l^{q(l,k)-1}) F_l^{-1} N \right] \tag{4.1}$$

with

$$R_l = F_l^{-1} G_l.$$

Assume further that the diagonal elements of $M^{-1}N$ and $F_l^{-1}G_l$ are positive, and that $F_l^{-1}N \geq 0$, then there exists $\widetilde{M}_{k-1}$ and $\widetilde{N}_{k-1}$ such that $A = \widetilde{M}_{k-1} - \widetilde{N}_{k-1}$ is a weak regular splitting, $T_{k-1} = \widetilde{M}_{k-1}^{-1}\widetilde{N}_{k-1} \geq 0$, all diagonal elements of T_{k-1} are positive.

Proof. Equation (4.1) can be proved by induction. Since $R_l = F_l^{-1}G_l \geq 0$, $F_l N \geq 0$, we have $T_{k-1} \geq 0$. At the same time, since all diagonal elements of R_l are positive, all diagonal elements of T_{k-1} are also positive. Furthermore, as $M^{-1} \geq 0$, $M = F_l - G_l$ is weak regular, we know [42] that $\rho(R_l) < 1$ and

$$\begin{aligned} T_{k-1} &= \sum_{l=1}^{r} E_l \left[R_l^{q(l,k)} + (I - R_l)^{-1}(I - R_l^{q(l,k)}) F_l^{-1} N \right] \\ &= \sum_{l=1}^{r} E_l \left[R_l^{q(l,k)} + (I - R_l^{q(l,k)}) M^{-1} N \right]. \end{aligned} \tag{4.2}$$

Define

$$\widetilde{F}_{l,k} = M(I - R_l^{q(l,k)})^{-1},$$

$$\widetilde{G}_{l,k} = \widetilde{F}_{l,k} R_l^{q(l,k)};$$

we have $\widetilde{F}_{l,k} - \widetilde{G}_{l,k} = M$ and

$$\widetilde{F}_{l,k}^{-1} = (I + R_l + \cdots + R_l^{q(l,k)-1}) F_l^{-1} \geq 0$$

and

$$\widetilde{F}_{l,k}^{-1} \widetilde{G}_{l,k} = R_l^{q(l,k)} \geq 0,$$

thus $(\widetilde{F}_{l,k}, \widetilde{G}_{l,k}, E_l)_{l=1}^r$ is also a weak regular multisplitting of M. Since $M^{-1} \geq 0$, from [19, Theorem 2.1(ii)], we know that $\sum_{l=1}^r E_l \widetilde{F}_{l,k}^{-1}$ is nonsingular, thus we can define

$$\widetilde{M}_{k-1} = \left(\sum_{l=1}^{r} E_l \widetilde{F}_{l,k}^{-1} \right)^{-1}, \quad \widetilde{N}_{k-1} = \widetilde{M}_{k-1} - A.$$

It is easy to verify that $\widetilde{M}_{k-1}, \widetilde{N}_{k-1}$ satisfy the requirements of the lemma. ∎

Remark. If A is nonsingular, results similar to this lemma have appeared in, e.g., [40, 22]. If A is singular, the expression of $A = \widetilde{M}_{k-1} - \widetilde{N}_{k-1}$ is not unique; see [5].

THEOREM 4.2. *Suppose that the matrix A and all related splittings satisfy the conditions in Lemma* 4.1. *Then for any $x(0)$, the sequence of iterative vectors $x(k)$ converges to some x^* such that $Ax^* = 0$. More specifically, if $x(0)$ is positive, x^* is also positive.*

Proof. Without loss of generality, we assume that $Ae = 0$. Let $x(k) = T_{k-1}x(k-1)$, where T_{k-1} is defined by Lemma 4.1, and construct the following sequence of vectors iteratively:

$$\begin{aligned} y(0) &= x(0); \\ y(m+1) &= \widetilde{T}_m y(m), \quad m = 0, 1, 2, \ldots, \end{aligned}$$

where

$$\widetilde{T}_m = T_{m(n-1)+n-2} \cdots T_{m(n-1)}.$$

From Lemma 4.1, T_k is induced by a weak regular splitting of A for $k = 0, 1, \ldots$. From the assumption of this theorem, all diagonals of T_k are positive, thus from Proposition 3.3, $\widetilde{T}_m$ is pseudocontractive with respect to $\| \ \|_\infty$ and $X^* = \{\alpha e \,|\, \alpha \in \mathbb{R}\}$. To apply Theorem 2.3, we need to look for a convergent subsequence $\{\widetilde{T}_{m_i}\}_{i=0}^{\infty}$ of $\{\widetilde{T}_m\}_{m=0}^{\infty}$. Once the splitting $A = M - N$ and the multisplitting of $M : (F_l, G_l, E_l)_{l=1}^{r}$ are defined, T_{k-1} is uniquely determined by an integer vector

$$\hat{q}(k) = (q(1,k), \ldots, q(l,k));$$

cf. (4.2). Note that if some component of $\hat{q}(k)$, say $q(1,k)$, is replaced by $+\infty$, then

$$T_{k-1} = E_1 M^{-1} N + \sum_{l=2}^{r} E_l \left[R_l^{q(l,k)} + (I - R_l^{q(l,k)}) M^{-1} N \right],$$

the operator T_{k-1} is well defined, and it has the same properties as the one which has finite parameters $q(1,k), \ldots, q(l,k)$. Similarly, each $\widetilde{T}_m$ is uniquely determined by an integer vector

$$\tilde{q}(m) = \left(\hat{q}((m+1)(n-1)-1), \ldots, \hat{q}(m(n-1))\right).$$

Now we choose a subsequence $\{\tilde{q}(m_i)\}_{i=0}^{\infty}$ of $\{\tilde{q}(m)\}_{m=0}^{\infty}$ such that for each component sequence of $\{\tilde{q}(m_i)\}_{i=0}^{\infty}$, either this component sequence has equal value for all i, or this component sequence tends to infinity as $i \to \infty$. From the above analysis, the subsequence $\{\widetilde{T}(m_i)\}_{i=0}^{\infty}$ of $\{\widetilde{T}(m)\}_{m=0}^{\infty}$ is convergent, and its limit is pseudocontractive.

So, by applying Theorem 2.3, the sequence of vectors $y(m)$ converges to some $x^* = \alpha^* e$. More specifically,

$$\min_i x_i(0) \le \alpha^* \le \max_i x_i(0).$$

As every T_k is nonexpansive, $\{y(m)\}$ is a subsequence of $\{x(k)\}$, the sequence of iterative vectors $x(k)$ converges to x^* also. ∎

Remark. The condition that $A = M - N$ is a regular splitting ($M^{-1} \ge 0$, $N \ge 0$) is not necessary, since only a weaker condition $F_l^{-1} N \ge 0$ is needed in this theorem

(this has been observed earlier in [20]) . The condition that $q(l,k)$ is bounded for all k is not necessary either. The condition below was given in [30] for the convergence of the nonstationary iteration

$$\|T_k(I-T_k)(I-T_k)^{\#}\| \le \theta < 1, \quad k=0,1,2,\ldots, \tag{4.3}$$

where T_k are the iteration matrices such that $x(k+1) = T_k x(k)$, and $^{\#}$ denotes the group inverse. The following example shows that there is a matrix T_k, which is pseudocontractive with respect to $\|\cdot\|_\infty$, but does not satisfy the above condition with respect to $\|\cdot\|_\infty$:

$$T_k = \frac{1}{100}\begin{pmatrix} 99 & 1 \\ 40 & 60 \end{pmatrix}, T_k^\infty = \frac{1}{41}\begin{pmatrix} 40 & 1 \\ 40 & 1 \end{pmatrix},$$

and $\|T_k(I-T_k)(I-T_k)^{\#}\|_\infty = \|T_k - T_k^\infty\|_\infty = \frac{236}{205} > 1$. We should note that the norm in (4.3) can be any norm, although the infinity norm is widely used in nonnegative matrix theory.

5. Asynchronous multisplitting iterations. Parallel multisplitting iterative methods can be implemented asynchronously, avoiding synchronization overhead and thus saving computational time. Asynchronous multisplitting iterations to solve nonsingular systems have been widely discussed; see, e.g., [10, 12, 39, 38]. The effectiveness of asynchronization was shown by Frommer, Schwandt, and Szyld [21] with numerical examples. To solve singular systems, Lubachevski and Mitra [27] proposed an asynchronous iterative algorithm in the case of a single splitting and Pott [35] gave a different approach to prove convergence.

Let $\{M_l, N_l, E_l\}_{l=1}^r$ be a multisplitting of A. The following asynchronous multisplitting iteration (AMI) was given by Bru, Elsner, and Neumann [10] to solve nonsingular systems; here we use it to solve singular systems. Its convergence will be proved under reasonable conditions.

ALGORITHM (AMI). Given the initial vectors $x(0)$, ..., $x(-D)$, for $k=0,1,2,\ldots$,

$$x(k+1) = (I - E_{l(k)})x(k) + E_{l(k)}M_{l(k)}^{-1}N_{l(k)}y(k), \tag{5.1}$$

with

$$y(k) = (x_1(k-d(k,1)),\ldots,x_n(k-d(k,n)))^T, \tag{5.2}$$

where $l(k) \in \{1,\ldots,r\}$, $0 \le d(k,i) \le k+D$ are integers less than or equal k.

Suppose we have a parallel computer consisting of a host and r slaves. There is a global approximation in the host. Every slave has a local approximation and does the following repeatedly: retrieves a global approximation y from the host, forms a local approximation, say $M_l^{-1}N_l y$, and sends it to the host. The host does the following repeatedly: receives a local approximation from some slave and forms a new global approximation as in (5.1). The terms $d(k,i)$ can be interpreted as follows: Suppose that the $l(k)$th slave retrieves a global approximation at the $(k-d)$th iteration and forms a new local approximation, the host uses this to form a new global approximation $x(k+1)$; during this period, other slave(s) may send their local approximations to the host and the host forms global approximations $x(k-d+1),\ldots,x(k-d)$; thus d is the iteration drift.

A more universal asynchronous model for a distributed parallel machine can be found in [21]. Here we use a simple model in order to show how to use the analysis technique developed above.

THEOREM 5.1. *Let A be an irreducible singular M-matrix, and $\{M_l, N_l, E_l\}_{l=1}^r$ be a weak regular multisplitting of A. If, in the AMI* (5.1) *and* (5.2),

(i) $\sum_{l=1}^r E_l M_l^{-1}$ *is nonsingular,*

(ii) *there exists some integer D such that*

$$\{l(k)\} \cup \{l(k+1)\} \cup \cdots \cup \{l(k+D)\} = \{1, \ldots, r\} \quad \text{for all } k = 0, 1, \ldots \tag{5.3}$$

and

$$0 \le d(k,i) \le D \quad \text{for all } k \ge 0, 1 \le i \le n, \tag{5.4}$$

(iii) *for each $1 \le i \le n$, either*

$$(E_l)_{ii} < 1 \tag{5.5}$$

or

$$(E_l)_{ii} = 1, \quad (T_l)_{ii} > 0, \quad \text{and } d(k,i) = 0 \text{ for } k : l(k) = l, \tag{5.6}$$

where $T_l = M_l^{-1} N_l$, then AMI converges.

Condition (i) is necessary even in the case of synchronous multisplitting iterations, see [26], to guarantee the consistency between the iteration and the system $Ax = 0$. If $A = M - N$ is a weak regular splitting of A and $(F_l, G_l, E_l)_l$ is a weak regular multisplitting of M (cf. section 4), then the multisplitting $(M_l, N_l, E_l)_l$ with

$$M_l = F_l, \quad N_l = G_l + N, \quad l = 1, \ldots, r,$$

satisfies condition (i). Condition (ii) is referred to as the condition that the sequence $l(k)$ be regulated in [10]. If condition (5.5) is satisfied, the AMI can be viewed as an extrapolating one; cf. [14, 15]. Condition (5.6) is referred to as a partial asynchronism condition; see [27, 7, 41]. In practice, this condition can be satisfied if the ith component is updated by only one processor in a distributed parallel computer system.

Proof. For an arbitrary fixed k', denote

$$\overline{\alpha} \equiv \max_{1 \le i \le n} \{x_i(k' - D), \ldots, x_i(k')\},$$

$$\underline{\alpha} \equiv \min_{1 \le i \le n} \{x_i(k' - D), \ldots, x_i(k')\}.$$

For any $k \ge k'$,

$$x_i(k+1) = (1 - (E_{l(k)})_{ii}) x_i(k) + (E_{l(k)})_{ii} \sum_{j=1}^n (T_{l(k)})_{ij} x_j(k - d(k,j)),$$

with condition (iii), by induction, (as in Proposition 3.2) we have

$$\begin{aligned} \underline{\alpha} < x_i(k) < \overline{\alpha} &\Rightarrow \underline{\alpha} < x_i(k+1) < \overline{\alpha}, \\ x_i(k) = \overline{\alpha} &\Rightarrow \underline{\alpha} < x_i(k+1) \le \overline{\alpha}, \\ x_i(k) = \underline{\alpha} &\Rightarrow \underline{\alpha} \le x_i(k+1) < \overline{\alpha}. \end{aligned} \tag{5.7}$$

LEMMA 5.2. *At least one of the sets* $\{i \mid x_i(k'+(n-1)(2D+1)) = \overline{\alpha}\}$ *and* $\{i \mid x_i(k'+(n-1)(2D+1)) = \underline{\alpha}\}$ *is empty.*

Proof. Without loss of generality, we suppose that both $\{i \mid x_i(k') = \overline{\alpha}\}$ and $\{i \mid x_i(k') = \overline{\alpha}\}$ are not empty, otherwise, from (5.7), we are done. From (5.7),

$$\{i \mid x_i(k'+(n-1)(2D+1)) = \underline{\alpha}\} \subset \cdots \subset \{i \mid x_i(k'+1) = \underline{\alpha}\} \subset \{i \mid x_i(k') = \underline{\alpha}\}$$

and

$$\{i \mid x_i(k'+(n-1)(2D+1)) = \overline{\alpha}\} \subset \cdots \subset \{i \mid x_i(k'+1) = \overline{\alpha}\} \subset \{i \mid x_i(k') = \overline{\alpha}\}.$$

We first prove that

$$\begin{aligned} &\{i \mid x_i(k'+2D+1) = \overline{\alpha} \text{ or } x_i(k'+2D+1) = \underline{\alpha}\} \\ &\neq \{i \mid x_i(k') = \overline{\alpha} \text{ or } x_i(k') = \underline{\alpha}\}. \end{aligned} \tag{5.8}$$

If this is not the case, by (5.7),

$$\{i \mid x_i(k') = \overline{\alpha}\} = \cdots = \{i \mid x_i(k'+2D+1) = \overline{\alpha}\},$$

$$\{i \mid x_i(k') = \underline{\alpha}\} = \cdots = \{i \mid x_i(k'+2D+1) = \underline{\alpha}\},$$

and

$$\underline{\alpha} < x_i(k) < \overline{\alpha} \quad \text{for all } i \text{ such that } \underline{\alpha} < x_i(k') < \overline{\alpha}, k = k', \ldots, k'+2D+1.$$

For $k : k'+D \le k \le k'+2D$, using (5.4), we have

$$k' \le k - d(k,j) \le k'+2D, \quad j = 1, \ldots, n,$$

for i such that $x_i(k') = \overline{\alpha}$, use $\sum_j (T_{l(k)})_{ij} = 1$, we can write $x_i(k+1)$ as

$$\begin{aligned} \overline{\alpha} &= x_i(k+1) \\ &= (1 - (E_{l(k)})_{ii})\overline{\alpha} \\ &\quad + (E_{l(k)})_{ii} \times \left(\overline{\alpha} \sum_{j:\, x_j(k')=\overline{\alpha}} (T_{l(k)})_{ij} + \sum_{j:\, x_j(k')<\overline{\alpha}} (T_{l(k)})_{ij} x_j(k-d(k,j)) \right) \\ &= \overline{\alpha} + (E_{l(k)})_{ii} \times \sum_{j:\, x_j(k')<\overline{\alpha}} (T_{l(k)})_{ij} (x_j(k-d(i,j)) - \overline{\alpha}). \end{aligned}$$

This equation holds if and only if

$$(E_{l(k)})_{ii} (T_{l(k)})_{ij} = 0 \text{ for } j : x_j(k') < \overline{\alpha},$$

which means that

$$(E_{l(k)})_{ii} \left(1 - \sum_{j:\, x_j(k')=\overline{\alpha}} (T_{l(k)})_{ij} \right) = 0, \quad k = k'+D, \ldots, k'+2D,$$

i.e.,

$$(E_{l(k)})_{ii} \sum_{j:\, x_j(k')=\overline{\alpha}} (T_{l(k)})_{ij} = (E_{l(k)})_{ii}, \quad k = k'+D, \ldots, k'+2D.$$

From (5.3), we have

$$\bigcup_{k'+D\le k\le k'+2D} \{l(k)\} = \{1,\dots,r\},$$

so,

$$\sum_{l=1}^{r}(E_l)_{ii} \sum_{j:\, x_j(k')=\overline{\alpha}} (T_l)_{ij} = \sum_{l=1}^{r}(E_l)_{ii} = 1,$$

which means that

$$\sum_{j:\, x_j(k')=\overline{\alpha}} T_{ij} = 1 \quad \text{for all } i \text{ such that } x_i(k') = \overline{\alpha}, \tag{5.9}$$

where

$$T = \sum_{l=1}^{r} E_l T_l.$$

By the same argument, we have also that

$$\sum_{j:\, x_j(k')=\underline{\alpha}} T_{ij} = 1 \quad \text{for all } i \text{ such that } x_i(k') = \underline{\alpha}. \tag{5.10}$$

Note that under the condition (i) of the theorem, T is also an iterative matrix induced by a weak regular splitting [26], so these two equalities (5.9) and (5.10) contradict Proposition 3.1. Therefore the number of elements in $\{i \mid x_i(k'+2D+1) = \overline{\alpha} \text{ or } x_i(k'+2D+1) = \underline{\alpha}\}$ is at least one less than the number of elements in $\{i \mid x_i(k') = \overline{\alpha} \text{ or } x_i(k') = \underline{\alpha}\}$.

Repeating the above proof, we have that either one of $\{i \mid x_i(k'+n'(2D+1)) = \overline{\alpha}\}$ and $\{i \mid x_i(k'+n'(2D+1)) = \underline{\alpha}\}$ is empty for some $1 \le n' \le n-1$, in this case, the lemma has been proved, or the number of elements in $\{i \mid x_i(k'+n'(2D+1)) = \overline{\alpha} \text{ or } x_i(k'+n'(2D+1)) = \underline{\alpha}\}$ is at least one less than the number of elements in $\{i \mid x_i(k'+(n'-1)(2D+1)) = \overline{\alpha} \text{ or } x_i(k'+(n'-1)(2D+1)) = \underline{\alpha}\}$ for all $1 \le n' \le n-1$. After at most $n-1$ steps, we get the conclusion. □

Proof of Theorem 5.1 *(continued).* Construct a big vector

$$\mathbf{x}(k) \equiv (x^T(k-D),\dots,x^T(k))^T \in \mathbb{R}^{(D+1)n}, \quad k = 0,1,2,\dots.$$

The asynchronous iteration (5.1) and (5.2) is equivalent to

$$\mathbf{x}(k+1) = \mathbf{T}_k\mathbf{x}(k), k = 0,1,2,\dots,$$

where $\mathbf{T}_k$ has the form

$$\begin{pmatrix} I - E_{l(k)} + * & * & * & \cdots & * \\ I & 0 & 0 & \cdots & 0 \\ & I & 0 & \cdots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & I & 0 \end{pmatrix}.$$

Consider

$$X^* = \{\alpha \mathbf{e} \mid \alpha \in \mathbb{R}\},$$

where $\mathbf{e} \in \mathbb{R}^{(D+1)n}$ is the vector with all components equal to 1. For any $\mathbf{x}(k') \notin X^*$, from the above lemma, we know that, say, $\{i \mid x_i(k' + (n-1)(2D+1)) = \overline{\alpha}\}$ is empty, and therefore $\{i \mid x_i(k) = \overline{\alpha}\}$ is empty for all $k \geq k' + (n-1)(2D+1)$, which means that

$$\|\mathbf{x}(k' + (n-1)(2D+1) + D) - P(\mathbf{x}(k' + (n-1)(2D+1) + D))\| < \|\mathbf{x}(k') - P(\mathbf{x}(k'))\|.$$

So for all $k' \geq 0$, the product

$$\mathbf{T}_{k'+(n-1)(2D+1)+D-1} \cdots \mathbf{T}_{k'+1} \mathbf{T}_{k'}$$

of $(n-1)(2D+1) + D$ operators is pseudocontractive with respect to $\|\cdot\|_\infty$ and X^*. Because there is only a finite number of operators, there is always a convergent subsequence in this sequence of operators, and its limit is also pseudocontractive, from Theorem 2.3, we obtain the convergence of $\mathbf{x}(k)$, which is equivalent to asserting that the sequence $x(k)$ converges. □

6. Conclusions. This paper introduced a new property of operators called pseudocontractivity and showed that it is a generalization of the paracontractivity property. A general convergence theorem for pseudocontractive iterations was proved and it was shown that, under appropriate conditions, the product of at most $n-1$ (where n is the dimension of the matrix) iteration matrices induced from weak regular splittings is pseudocontractive. This analysis technique was applied to nonstationary iterative methods for solving singular M-systems, specifically, nonstationary two-stage multisplitting methods and asynchronous multisplitting methods, with no other contractivity condition on the iteration operators.

Acknowledgments. We thank Professor Daniel B. Szyld for a careful reading of a draft version of this paper and, among other helpful suggestions, pointing out the nonuniqueness of expression of the splitting in the first remark of section 4. We also thank the anonymous reviewers for several questions and suggestions that improved the paper.

REFERENCES

[1] G. Alefeld, I. Lenhardt, and G. Mayer, *On multisplitting methods for band matrices*, Numer. Math., 75 (1997), pp. 267–292.

[2] Z.-Z. Bai, *The monotone convergence of a class of parallel nonlinear relaxation methods for nonlinear complementarity problems*, Comput. Math. Appl., 31 (1996), pp. 17–33.

[3] Z.-Z. Bai, *The monotone convergence rate of the parallel nonlinear AOR method*, Comput. Math. Appl., 31 (1996), pp. 1–8.

[4] Z.-Z. Bai and D. Wang, *A class of parallel nonlinear multisplitting relaxation methods for the large sparse nonlinear complementarity problems*, Comput. Math. Appl., 32 (1996), pp. 79–95.

[5] M. Benzi and D. B. Szyld, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.

[6] A. Berman and B. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation—Numerical methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[8] W.-J. Beyn and L. Elsner, *Infinite products and paracontracting matrices*, Electron. J. Linear Algebra, 2 (1997), pp. 1–8.

[9] R. Bru, C. Corral, A. Martínez, and J. Mas, *Multisplitting preconditioners based on incomplete Choleski factorizations*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1210–1222.
[10] R. Bru, L. Elsner, and M. Neumann, *Models of parallel chaotic iteration methods*, Linear Algebra Appl., 103 (1988), pp. 175–192.
[11] R. Bru, L. Elsner, and M. Neumann, *Convergence of infinite products of matrices and inner-outer iteration schemes*, Electron. Trans. Numer. Anal., 2 (1994), pp. 183–193.
[12] R. Bru, V. Migallón, J. Penadés, and D. B. Szyld, *Parallel, synchronous and asynchronous two-stage multisplitting methods*, Electron. Trans. Numer. Anal., 3 (1995), pp. 24–38.
[13] L. Elsner, *Comparisons of weak regular splittings and multisplitting methods*, Numer. Math., 56 (1989), pp. 283–289.
[14] L. Elsner, I. Koltracht, and M. Neumann, *On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data*, Linear Algebra Appl., 130 (1990), pp. 65–82.
[15] L. Elsner, I. Koltracht, and M. Neumann, *Convergence of sequential and asynchronous nonlinear paracontractions*, Numer. Math., 62 (1992), pp. 305–319.
[16] A. Frommer, *Parallel nonlinear multisplitting methods*, Numer. Math., 56 (1989), pp. 269–282.
[17] A. Frommer and G. Mayer, *Parallel interval multisplittings*, Numer. Math., 56 (1989), pp. 255–267.
[18] A. Frommer and G. Mayer, *Safe bounds for the solutions of nonlinear problems using a parallel multisplitting method*, Computing, 42 (1989), pp. 171–186.
[19] A. Frommer and B. Pohl, *A comparison result for multisplittings and waveform relaxation methods*, Numer. Linear Algebra Appl., 2 (1995), pp. 335–346.
[20] A. Frommer and H. Schwandt, *A unified representation and theory of algebraic additive Schwarz and multisplitting methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 893–912.
[21] A. Frommer, H. Schwandt, and D. B. Szyld, *Asynchronous weighted additive Schwarz methods*, Electron. Trans. Numer. Anal., 5 (1997), pp. 48–61.
[22] A. Frommer and D. B. Szyld, *Asynchronous two-stage iterative methods*, Numer. Math., 69 (1994), pp. 141–153.
[23] A. Hadjidimos and A. K. Yeyios, *Some notes on multisplitting methods and m-step preconditioners for linear systems*, Linear Algebra Appl., 248 (1996), pp. 277–301.
[24] C.-M. Huang and D. P. O'Leary, *A Krylov multisplitting algorithm for solving linear systems of equations*, Linear Algebra Appl., (1993), pp. 9–29.
[25] M. T. Jones and D. B. Szyld, *Two-stage multisplitting methods with overlapping blocks*, Numer. Linear Algebra Appl., 3 (1996), pp. 113–124.
[26] J. P. Kavanagh and M. Neumann, *Consistency and convergence of the parallel multisplitting method for singular* M*-matrices*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 210–218.
[27] B. Lubachevsky and D. Mitra, *A chaotic asynchronous algorithm for computing the fixed point of a nonnegative matrix of unit spectral radius*, J. Assoc. Comput. Mach., 33 (1985), pp. 130–150.
[28] N. Machida, M. Fukushima, and T. Ibaraki, *A multisplitting method for symmetric linear complementarity problems*, J. Assoc. Comput. Mach., 62 (1995), pp. 217–227.
[29] I. Marek and D. B. Szyld, *Iterative and semi-iterative methods for computing stationary probability vectors of Markov operators*, Math. Comp., 61 (1993), pp. 719–731.
[30] V. Migallón, J. Penadés, and D. B. Szyld, *Block two-stage methods for singular systems and Markov chains*, Numer. Linear Algebra Appl., 3 (1996), pp. 413–426.
[31] R. Nabben, *A note on comparison theorems for splittings and multisplittings of Hermitian positive definite matrices*, Linear Algebra Appl., 233 (1996), pp. 67–80.
[32] S. Nelson and M. Neumann, *Generalization of the projection method with applications to SOR method for Hermitian positive semidefinite linear systems*, Numer. Math., 51 (1987), pp. 123–141.
[33] M. Neumann and R. J. Plemmons, *Convergence of parallel multisplitting iterative methods for* M*-matrices*, Linear Algebra Appl., 88/89 (1987), pp. 559–573.
[34] D. P. O'Leary and R. E. White, *Multisplittings of matrices and parallel solution of linear systems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 630–640.
[35] M. Pott, *On the convergence of asynchronous iteration methods for nonlinear paracontractions and consistent linear systems*, Linear Algebra Appl., 283 (1998), pp. 1–33.
[36] H. Schneider, *Theorems on* M*-splittings of a singular* M*-matrix which depend on graph structure*, Linear Algebra Appl., 58 (1984), pp. 407–424.
[37] Y. Song and D. Yuan, *On the convergence of relaxed parallel chaotic iterations for* H*-matrix*, Int. J. Comput. Math., 52 (1994), pp. 195–209.
[38] Y. Su, *Generalized multisplitting asynchronous iteration*, Linear Algebra Appl., 235 (1996), pp. 77–92.

[39] Y. Su and S. Zhu, *A model for parallel multisplitting chaotic iterations*, J. Fudan Univ. Natur. Sci., 30 (1991), pp. 444–450 (in Chinese).
[40] D. B. Szyld and M. T. Jones, *Two-stage and multisplitting methods for the parallel solution of linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 671–679.
[41] P. Tseng, D. Bertsekas, and J. Tsitsiklis, *Partially asynchronous, parallel algorithms for network flow and other problems*, SIAM J. Control Optim., 28 (1990), pp. 678–710.
[42] R. S. Varga, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.
[43] D. Wang, *On the convergence of the parallel multisplitting AOR algorithm*, Linear Algebra Appl., 154/156 (1991), pp. 473–486.
[44] R. E. White, *Multisplitting of a symmetric positive definite matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 69–82.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 965–970

ON REAL SOLUTIONS OF THE EQUATION $\Phi^t(A) = \frac{1}{n}J_n$*

YUMING CHEN†

Abstract. For a class of $n \times n$-matrices, we get related real solutions to the matrix equation $\Phi^t(A) = \frac{1}{n}J_n$ by generalizing the approach of and applying the results of Zhang, Yang, and Cao [*SIAM J. Matrix Anal. Appl.*, 21 (1999), pp. 642–645]. These solutions contain not only those obtained by Zhang, Yang, and Cao but also some which are neither diagonally nor permutation equivalent to those obtained by Zhang, Yang, and Cao. Therefore, the open problem proposed by Zhang, Yang, and Cao in the cited paper is solved.

Key words. Hadamard product, diagonally equivalent, permutation equivalent

AMS subject classifications. 15A24, 93A99, 65F99

PII. S0895479800372912

1. Introduction.

For a given positive integer n, let $M_n(\mathbb{R})$ and $GL_n(\mathbb{R})$ be the sets of all $n \times n$ real matrices and all $n \times n$ real nonsingular matrices, respectively. Two important members of $M_n(\mathbb{R})$ are the $n \times n$ identity and all-one matrix, denoted as I_n and J_n, respectively.

For $A = (a_{ij})$ and $B = (b_{ij})$ in $M_n(\mathbb{R})$, the Hadamard product of A and B is defined as $A \circ B = (a_{ij}b_{ij}) \in M_n(\mathbb{R})$. Then we define

$$\Phi : GL_n(\mathbb{R}) \to M_n(\mathbb{R})$$

by

$$\Phi(A) = A \circ A^{-T}, \qquad A \in GL_n(\mathbb{R}),$$

where A^{-T} means the inverse transpose, $(A^{-1})^T$, of A. The mapping Φ arises in mathematical control theory in chemical engineering design problems. The basic question about Φ is to determine its range.

It is easy to see that every matrix in the range of Φ has row and column sums 1. However, the converse is not true. In fact, Johnson and Shapiro [1] showed that the equation $\Phi(A) = \frac{1}{3}J_3$ has no real solutions. So they asked whether the equation

$$\Phi(A) = \frac{1}{n}J_n$$

has a real solution. This problem was solved by Zhang, Yang, and Cao [2]. In fact, they studied the more general problem, i.e., the existence of real solutions of the equation

$$\Phi^t(A) = \frac{1}{n}J_n \tag{1}$$

for any positive integer t, where Φ^t is the mapping Φ applied t times.

*Received by the editors June 5, 2000; accepted for publication (in revised form) by R. Brualdi September 16, 2000; published electronically January 19, 2001. This research was partially supported by the Izaak Walton Killam Memorial Postdoctoral Fellowship, University of Alberta.
http://www.siam.org/journals/simax/22-3/37291.html

†Department of Mathematics and Statistics, York University, 4700 Keele Street, Toronto, ON, M3J 1P3 Canada. Current address: Department of Mathematical Sciences, University of Alberta, Edmonton, AB, T6G 2G1 Canada (ychen@math.ualberta.ca).

The purpose of this note is to obtain some new solutions to (1) by generalizing the approach of and applying the results of [2]. Some of the solutions are neither diagonally equivalent nor permutation equivalent to those obtained in [2]. Hence, the open problem proposed in [2] is solved.

The organization of this note is as follows. First, by an example, we partially answer the open problem in [2] and introduce the notion of permutation equivalence. Then we obtain new solutions to (1) which are related to a class of $n \times n$-matrices. As a result, the above-mentioned open problem is solved.

2. An example. First, we recall some results about the mapping Φ.

LEMMA 1 (see [1, Observations 2 and 4]). *For $A \in GL_n(\mathbb{R})$,*

(i) *if D and E in $GL_n(\mathbb{R})$ are diagonal, then $\Phi(DAE) = \Phi(A)$;*

(ii) *if P and Q in $GL_n(\mathbb{R})$ are permutation matrices, then $\Phi(PAQ) = P\Phi(A)Q$.*

When $n = 4$ and $t = 1$, the unique solution to (1) with respect to diagonal equivalence obtained in [2] is

$$A = \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}. \tag{2}$$

Let P be the permutation

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, by Lemma 1,

$$\Phi(PA) = P\Phi(A) = P\left(\frac{1}{4}J_4\right) = \frac{1}{4}J_4.$$

But PA is not diagonally equivalent to A. If not, let $D = \operatorname{diag}(d_1, d_2, d_3, d_4)$ and $E = \operatorname{diag}(e_1, e_2, e_3, e_4)$ be two nonsingular diagonal matrices such that

$$DAE = PA.$$

Particularly, we have $d_1e_1 = -1$, $d_3e_1 = 1$, $d_1e_3 = 1$, and $d_3e_3 = 1$. This is impossible since the first two equations give $d_1 = -d_3$ while the last two give $d_1 = d_3$.

With (1) and the example in mind, Lemma 1 leads us to introduce the notion of permutation equivalence in a similar way to diagonal equivalence (see [2]).

Zhang, Yang, and Cao [2] also proved that, for $n = 2$ and $t = 1$, there is only one nondiagonally equivalent solution to (1) and no solution to (1) when $n = 2$ and $t \geq 2$. This, combined with the result of Johnson and Shapiro [1], intrigues us, and we naturally ask, for $n \geq 4$, whether there exist real solutions to (1) which are neither diagonally equivalent nor permutation equivalent to those found by Zhang, Yang, and Cao [2]. A positive answer will be given in the next section. In this section, we give a partial answer for the case where $n = 4$ and $t = 1$. For $a \neq 0$, let

$$A_a = \begin{pmatrix} 1 & a & -1 & a \\ a & 1 & a & -1 \\ -1 & a & 1 & a \\ a & -1 & a & 1 \end{pmatrix}. \tag{3}$$

Then $|A_a| = -16a^2 \neq 0$ and

$$A_a^{-1} = \frac{1}{4}\begin{pmatrix} 1 & a^{-1} & -1 & a^{-1} \\ a^{-1} & 1 & a^{-1} & -1 \\ -1 & a^{-1} & 1 & a^{-1} \\ a^{-1} & -1 & a^{-1} & 1 \end{pmatrix}.$$

Therefore,

$$\Phi(A_a) = \frac{1}{4}J_4.$$

Now, assume that $a \in \mathbb{R} \setminus \{0, -1, 1\}$. Obviously, A_a is not permutation equivalent to A given in (2). Now, we claim that A_a is not diagonally equivalent to A. In fact, if $D = \operatorname{diag}(d_1, d_2, d_3, d_4)$ and $E = \operatorname{diag}(e_1, e_2, e_3, e_4)$ are two nonsingular diagonal matrices such that

$$DAE = A_a,$$

then we have $e_1 = e_3$, $e_2 = e_4 = -ae_1$, $d_1 = d_3$, $d_2 = d_4 = -ad_1$, $d_1e_1 = -1$, and $d_2e_2 = -1$. Thus $-1 = d_2e_2 = a^2d_1e_1 = -a^2$ or $a^2 = 1$, a contradiction.

Remark 1. To obtain the form of A_a in (3), we tried the cyclic matrix generated by $(1, a, b, c)$. By requiring it satisfy $\Phi(A) = \frac{1}{4}J_4$, we get $b = 1$ or -1. But $b = 1$ is not suitable. Taking $b = -1$, we have $a = c$. Fortunately, the matrix we get satisfies $\Phi(A) = \frac{1}{4}J_4$. We believe that this approach is also applicable to deal with the general case. However, in the following section, by generalizing the approach of and using the results of Zhang, Yang, and Cao [2], we give a new approach to the general case.

3. New solutions to $\Phi^t(A) = \frac{1}{n}J_n$. In this section we always assume $n \geq 4$. Note that what is important in the presentation of Zhang, Yang, and Cao [2] are some special properties of J_n such as $J_n^2 = nJ_n$ and $J_n \circ J_n^T = J_n$. Inspired by this observation, for $k \in \mathbb{R}$ such that $k \leq 0$ or $k \geq 2n - 4$ (these restrictions on k will be clear later), we introduce a subset $M_{n,k}(\mathbb{R})$ of $M_n(\mathbb{R})$ as follows:

$$M_{n,k}(\mathbb{R}) = \{A \in M_n(\mathbb{R}); A \circ A^T = J_n, a_{ii} = 1 \text{ for } i = 1, \ldots, n, A^2 = kI_n + (n-k)A\}.$$

Generalizing the approach of and applying some results of Zhang, Yang, and Cao [2], we show that for each $A \in M_{n,k}(\mathbb{R})$ there exist corresponding solutions to (1).

Examples. Let $a = (a_1, \ldots, a_{n-1}) \in \mathbb{R}^{n-1}$ with $a_i \neq 0$ for $i \in \{1, \ldots, n-1\}$. Define $A^a \in M_n(\mathbb{R})$ by

$$A_{ij}^a = \begin{cases} 1 & \text{if } i = j, \\ \prod_{k=i}^{j-1} a_k & \text{if } i < j, \\ \frac{1}{\prod_{k=j}^{i-1} a_k} & \text{if } i > j. \end{cases}$$

Then it is easy to show that $A^a \in M_{n,0}(\mathbb{R})$. Particularly, when $a_i = 1$ for $i \in \{1, \ldots, n-1\}$, $A^a = J_n$. Furthermore, define $\tilde{A}^a \in M_n(\mathbb{R})$ by

$$\tilde{A}_{ij}^a = \begin{cases} A_{ij}^a & \text{if } i = j, \\ (-1)^{i-j+1}A_{ij}^a & \text{if } i \neq j. \end{cases}$$

Then $\tilde{A}^a \in M_{n,2n-4}(\mathbb{R})$. Note that neither A^a nor $\tilde{A}^a$ is symmetric if there exists an $i_0 \in \{1, \ldots, n-1\}$ such that $a_{i_0}^2 \neq 1$.

The following results can be easily proved and hence the proofs are omitted.

LEMMA 2. *Let $A \in M_{n,k}(\mathbb{R})$. If $a[a+b(n-k)]-b^2k \neq 0$, then $aI_n+bA \in GL_n(\mathbb{R})$ with*

$$(aI_n + bA)^{-1} = \frac{1}{a[a+b(n-k)]-b^2k}\{[a+b(n-k)]I_n - bA\}$$

and therefore

$$\Phi(aI_n + bA) = \frac{1}{a[a+b(n-k)]-b^2k}\{[a^2+ab(n-k)+b^2(n-k)]I_n - b^2J_n\}.$$

LEMMA 3. *Let $A \in M_{n,k}(\mathbb{R})$ and $\lambda \neq k$. Denote*

$$A(\lambda) = \frac{1}{\lambda - k}[(\lambda + n - k)I_n - A]. \tag{4}$$

Then

(i) *$A(\lambda) \in GL_n(\mathbb{R})$ if $\lambda \neq \frac{-(n-k)\pm\sqrt{(n-k)^2+4k}}{2}$;*

(ii) *if $\lambda \neq \frac{-(n-k)\pm\sqrt{(n-k)^2+4k}}{2}$, we have $\Phi(A(\lambda)) = J_n(\mu)$, where $\mu = \lambda(\lambda+n-k)-k$;*

(iii) *$A(\alpha)$ and $A(\beta)$ are not diagonally equivalent if $\alpha \neq \beta$.*

Proof. Since $J_n \in M_{n,0}(\mathbb{R})$, it follows from (4) that

$$J_n(\lambda) = \frac{1}{\lambda}[(\lambda + n)I_n - J_n] \tag{5}$$

for $\lambda \neq 0$. Now, (i) and (ii) follow easily from Lemma 2. The proof of (iii) is similar to that of (iii) of Lemma 2 of Zhang, Yang, and Cao [2]. This completes the proof of the lemma. □

THEOREM 1. *Let $A \in M_{n,k}(\mathbb{R})$ and t be a positive integer. Then if $n > 4$, there are 2^t distinct, real values of λ such that $\Phi^t(A(\lambda)) = \frac{1}{n}J_n$ and hence (1) has at least 2^t nondiagonally equivalent solutions. When $n = 4$, there are 2^{t-1} distinct, real values of λ such that $\Phi^t(A(\lambda)) = \frac{1}{4}J_4$ and hence (1) has at least 2^{t-1} nondiagonally equivalent solutions.*

Proof. We prove only the theorem for the case where $n > 4$. The proof is similar for the case where $n = 4$. First note that the nondiagonal equivalence follows from (iii) of Lemma 3. Second, it follows from (5) that

$$J_n(-n) = \frac{1}{n}J_n. \tag{6}$$

Now we distinguish two cases to complete the proof.

Case 1. $t = 1$. Consider the equation

$$\lambda(\lambda + n - k) - k = -n. \tag{7}$$

Note $\Delta = (n-k)^2 - 4(n-k) = (n-k)(n-k-4) > 0$ since $k \leq 0$ or $k \geq 2n-4$. Thus (7) has two distinct real solutions λ_1 and λ_2. By (6) and (ii) of Lemma 3,

$$\Phi(A(\lambda_1)) = \Phi(A(\lambda_2)) = J_n(-n) = \frac{1}{n}J_n.$$

Case 2. $t > 1$. It follows from (ii) of Lemma 3, for $\mu \neq -n$, that

$$\Phi(J_n(\mu)) = J_n(f(\mu)),$$

where $f(\mu) = \mu(\mu + n)$. Thus, if $\lambda \neq \frac{-(n-k)\pm\sqrt{(n-k)^2+4k}}{2}$, using (ii) of Lemma 3 again, we have

$$\Phi^t(A(\lambda)) = \Phi^{t-1}(\Phi(A(\lambda))) = \Phi^{t-1}(J_n(\mu)) = J_n(f^{t-1}(\mu)), \tag{8}$$

where $\mu = \lambda(\lambda+n-k)-k$. Lemma 3 of Zhang, Yang, and Cao [2] tells us that in the interval $(-\frac{n^2}{4}, 0]$ there are 2^{t-1} distinct, real solutions to the equation $f^{t-1}(\mu) = -n$, say $\mu_1, \ldots, \mu_{2^{t-1}}$. For $i = 1, \ldots, 2^{t-1}$, consider

$$\lambda(\lambda + n - k) - k = \mu_i. \tag{9}$$

Noting

$$\begin{aligned}\Delta &= (n-k)^2 + 4(k+\mu_i)\\ &= (n^2+4\mu_i) + (k^2 - 2nk + 4k)\\ &> k^2 - 2nk + 4k\\ &= k[k-(2n-4)]\\ &\geq 0\end{aligned}$$

(from here you see why we require $k \leq 0$ or $k \geq 2n-4$), we know that (9) has two distinct, real solutions, say $\lambda_{i,1}$ and $\lambda_{i,2}$. Moreover, it is easy to see that all $\lambda_{1,1}$, $\lambda_{1,2}$, …, $\lambda_{2^{t-1},1}$, and $\lambda_{2^{t-1},2}$ are distinct. Thus it follows from (6) and (8) that

$$\Phi^t(A(\lambda_{i,j})) = J_n(f^{t-1}(\mu_i)) = J_n(-n) = \frac{1}{n}J_n, \qquad i = 1, \ldots, 2^{t-1}, j = 1, 2,$$

and the proof is complete. □

Remark 2. For $A \in M_{n,k}(\mathbb{R})$, we can find 2^t and 2^{t-1} mutually nondiagonally equivalent real solutions $A(\lambda_0)$ to (1) for $n > 4$ and $n = 4$, respectively, where λ_0 satisfies the following inverted iteration (see Theorem 1 here and Remark 1 of Zhang, Yang, and Cao [2]):

$$\begin{cases}\lambda_t = -n,\\ \lambda_k = \frac{-n\pm\sqrt{n^2+4\lambda_{k+1}}}{2}, \quad k = 1, \ldots, t-1,\\ \lambda_0 = \frac{-(n-k)\pm\sqrt{(n-k)^2+4(\lambda_1+k)}}{2}.\end{cases}$$

Remark 3. For $a = (a_1, \ldots, a_{n-1}) \in \mathbb{R}^{n-1}$ with $a_i \neq 0$ for $i \in \{1, \ldots, n-1\}$, let A^a and $\tilde{A}^a$ be defined as in the examples above. Then we can easily show that the solutions associated with A^a and $\tilde{A}^a$ are diagonally equivalent to those associated with $A^{\mathbf{1}}$ and $\tilde{A}^{\mathbf{1}}$, respectively, where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^{n-1}$. But the solutions associated with $A^{\mathbf{1}}$ and $\tilde{A}^{\mathbf{1}}$ are neither diagonally equivalent nor permutation equivalent. Note that the solutions associated with $A^{\mathbf{1}}$ are just those obtained by Zhang, Yang, and Cao [2] and hence the open problem, whether there are real solutions to (1) which are not diagonally equivalent to those found in [2] when $n \geq 4$, proposed by them is

solved. Moreover, when $n = 4$, $t = 1$ and $a \in \mathbb{R} \setminus \{0, -1, 1\}$, the solutions given in section 2 are not associated with any $A \in M_{n,k}(\mathbb{R})$.

REFERENCES

[1] C. R. Johnson and H. M. Shapiro, *Mathematical aspects of the relative gain array* $(A \circ A^{-T})$, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 627–644.

[2] X. Zhang, Z. Yang, and C. Cao, *Real solutions of the equation* $\Phi^t(A) = \frac{1}{n} J_n$, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 642–645.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 3, pp. 971–971

ERRATUM: POINTWISE AND UNIFORMLY CONVERGENT SETS OF MATRICES*

ADAM L. COHEN†, LEIBA RODMAN‡, AND DAVID P. STANFORD‡

Abstract. An error in the paper [A. L. Cohen, L. Rodman, and D. P. Stanford, *SIAM J. Matrix Anal. Appl.*, 21 (1999), pp. 93–105] is corrected.

Key words. uniform convergence, matrix sets

AMS subject classification. 15A99

PII. S0895479800374674

We wish to point out that in Theorem 4.4 of our paper [1], as well as in the paragraph preceding Theorem 4.4, a hypothesis is inadvertently omitted. Namely, it should be assumed in Theorem 4.4 that the diagonal entries of the matrices in the set $\mathcal{A}$ are all nonzero. Without this assumption the statement of Theorem 4.4 is false, as the following example shows:

$$\mathcal{A} = \left\{ A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},\ A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \right\}.$$

The set $\mathcal{A}$ is in the closure of the set $\mathcal{U}_{2,2}$ (in the notation of [1]) of uniformly convergent ordered pairs of 2×2 real matrices. Indeed,

$$\mathcal{A} = \lim_{p \to \infty} \mathcal{A}_p,$$

where the sets

$$\mathcal{A}_p = \left\{ \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},\ \begin{bmatrix} 2 & 0 \\ 0 & 0.5^{1/p} \end{bmatrix} \right\}$$

are uniformly convergent. Nevertheless, $\rho(\mathcal{A}) = 2$.

This omission does not affect the rest of the paper [1].

REFERENCES

[1] A. L. COHEN, L. RODMAN, AND D. P. STANFORD, *Pointwise and uniformly convergent sets of matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 93–105.

*Received by the editors June 27, 2000; accepted for publication by P. Van Dooren June 29, 2000; published electronically January 19, 2001.

http://www.siam.org/journals/simax/22-3/37467.html

†Oregon State University, Corvallis, OR 97330 (cohena@ucs.orst.edu). Current address: 122 Campbell Avenue, Revere, MA 02151.

‡Department of Mathematics, College of William and Mary, P. O. Box 8795, Williamsburg, VA 23187-8795 (lxrodm@math.wm.edu, stanford@math.wm.edu).

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 973–996

ON ALGORITHMS FOR PERMUTING LARGE ENTRIES TO THE DIAGONAL OF A SPARSE MATRIX*

I. S. DUFF† AND J. KOSTER‡

Abstract. We consider bipartite matching algorithms for computing permutations of a sparse matrix so that the diagonal of the permuted matrix has entries of large absolute value. We discuss various strategies for this and consider their implementation as computer codes. We also consider scaling techniques to further increase the relative values of the diagonal entries. Numerical experiments show the effect of the reorderings and the scaling on the solution of sparse equations by a direct method and by preconditioned iterative techniques.

Key words. sparse matrices, bipartite weighted matching, shortest path algorithms, direct methods, iterative methods, preconditioning

AMS subject classifications. 05C70, 65F05, 65F10, 65F50

PII. S0895479899358443

1. Introduction.

We say that an $n \times n$ matrix A has a large diagonal if the absolute value of each diagonal entry is large relative to the absolute values of the off-diagonal entries in its row and column. Permuting large nonzero entries onto the diagonal of a sparse matrix can be useful in several ways. If we wish to solve the system

$$Ax = b, \tag{1.1}$$

where A is a nonsingular square matrix of order n and x and b are vectors of length n, then a preordering of this kind can be useful whether direct or iterative methods are used for solution (see [17, 33]).

The work in this paper is a continuation of the work reported in [17]. In that paper the authors presented an algorithm that maximizes the smallest entry on the diagonal and relies on repeated applications of the depth first search algorithm MC21 [15] in the Harwell Subroutine Library [28]. In the current paper, we will be concerned with other bipartite matching algorithms for permuting the rows and columns of the matrix so that the diagonal of the permuted matrix is large. The algorithm that is central to this paper computes a matching that corresponds to a permutation of a sparse matrix such that the product (or sum) of the diagonal entries is maximized. This algorithm is already mentioned and used in [17], but it is not fully described. We describe the algorithm and its implementation in more detail in this paper. We also consider a modified version of this algorithm to compute a permutation of the matrix that maximizes the smallest diagonal entry. We compare the performance of this algorithm with that in [17]. We also investigate the influence of scaling the matrix. Scaling can be used before or after computation of the matching to make the diagonal entries even larger relative to the off-diagonals. In particular, we look at a sparse variant of a bipartite matching and scaling algorithm of Olschowka and

*Received by the editors July 12, 1999; accepted for publication (in revised form) by E. Ng September 6, 2000; published electronically January 31, 2001.
http://www.siam.org/journals/simax/22-4/35844.html
†Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX England, and CERFACS, Toulouse, France (I.Duff@rl.ac.uk).
‡Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX England. Current address: Parallab, University of Bergen, 5020 Bergen, Norway (jak@ii.uib.no).

Neumaier [33] that first maximizes the product of the diagonal entries and then scales the matrix so that these entries are one and all other entries are no greater than one.

The paper is organized as follows. In section 2, we describe some concepts of bipartite matching that we need for the descriptions of the algorithms. In section 3, we review the basic properties of an algorithm (MC21) that computes a matching that corresponds to a permutation of the matrix that puts as many entries as possible onto the diagonal. The algorithm operates on a bipartite graph that has no weights; i.e., the numerical values of the matrix entries are not taken into account. Section 4 describes the algorithm that computes a matching for permuting a matrix such that the product of the diagonal entries is maximized. It is based on finding a minimum weight matching in a bipartite graph with nonnegative edge weights. The algorithm computes a sequence of shortest (augmenting) paths in this graph, each of which is used to extend a partial matching. Properties of the weighted bipartite graph and the partial matching are discussed. Details on the construction of the augmenting paths are also given. In section 5, we modify this algorithm such that it maximizes the smallest diagonal entry of the permuted matrix. In section 6, we consider the scaling of the reordered matrix. Computational experience for the algorithms applied to some practical problems and the effect of the reorderings and scaling on direct and iterative methods of solution are presented in section 7. Finally, we consider some of the implications of this current work in section 8.

2. Bipartite matching. Let $A = (a_{ij})$ be a general $n \times n$ sparse matrix. With the matrix A, we associate a bipartite graph $G_A = (V_r, V_c, E)$ that consists of two disjoint node sets V_r and V_c and an edge set E, where $(u, v) \in E$ implies that $u \in V_r$, $v \in V_c$. The sets V_r and V_c have cardinality n and correspond to the rows and columns of A, respectively. Edge $(i, j) \in E$ if and only if $a_{ij} \neq 0$. We define the sets $ROW(i) = \{j|(i, j) \in E\}$, for $i \in V_r$, and $COL(j) = \{i|(i, j) \in E\}$, for $j \in V_c$. These sets correspond to the positions of the entries in row i and column j of the sparse matrix, respectively. We use $|\ldots|$ both to denote the absolute value and to signify the number of entries in a set, sequence, or matrix. The meaning should always be clear from the context.

A subset $M \subseteq E$ is called a *matching* (or assignment) if no two edges of M are incident to the same node. A matching containing the largest number of edges possible is called a *maximum cardinality* matching (or simply maximum matching). A maximum matching is a *perfect* matching if every node is incident to a matching edge. Obviously, not every bipartite graph allows a perfect matching. However, if the matrix A is nonsingular, then there exists a perfect matching for G_A. A perfect matching M has cardinality n and defines an $n \times n$ permutation matrix $Q = (q_{ij})$ with

$$\begin{cases} q_{ji} = 1 & \text{for } (i, j) \in M, \\ q_{ji} = 0 & \text{otherwise,} \end{cases}$$

so that both QA and AQ are matrices with the matching entries on the (zero-free) diagonal. Bipartite matching problems can be viewed as a special case of network flow problems (see, for example, [21]). We refer the reader to [1] for an introduction to bipartite matching and network flow algorithms.

The more efficient algorithms for finding maximum matchings in bipartite graphs make use of augmenting paths. Let M be a matching in G_A. A node v is matched if it is incident to an edge in M. A path P in G_A is defined as an ordered set of edges in which successive edges are incident to the same node. A path P is called

an *M-alternating path* if the edges of P are alternately in M and not in M. An M-alternating path P is called an *M-augmenting* path if it connects an unmatched row node with an unmatched column node. In the bipartite graph in Figure 2.1, there exists an M-augmenting path from column node 8 to row node 8. If it is clear from the context which matching M is associated with the M-alternating and M-augmenting paths, then we will simply refer to them as alternating and augmenting paths.

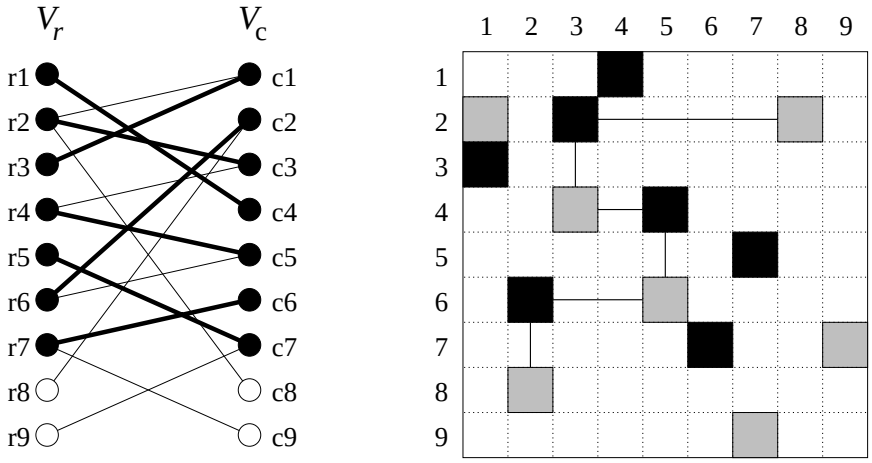


FIG. 2.1. *Example of an M-augmenting path. The bipartite graph contains nine row nodes* r1, . . . , r9 *and nine column nodes* c1, . . . , c9. *The white nodes are unmatched, and the black nodes are matched. The matching M (of cardinality 7) is represented by the thick edges in the bipartite graph and by the black entries in the matrix. The connected matrix entries form the augmenting path* $\{(8,2),(6,2),(6,5),(4,5),(4,3),(2,3),(2,8)\}$.

Let M and P be subsets of E. We define

$$M \oplus P := (M \setminus P) \cup (P \setminus M).$$

If M is a matching and P is an M-augmenting path, then $M \oplus P$ is again a matching, and $|M \oplus P| = |M| + 1$. If P is an M-alternating cyclic path, i.e., an alternating path whose first and last edges are incident to the same node, then $M \oplus P$ is also a matching and $|M \oplus P| = |M|$. A key observation for the construction of a maximum or perfect matching is that a matching M is maximum if and only if there is no augmenting path relative to M.

In what follows, a matching M will often be represented by a pointer array $m : V_r \cup V_c \to V_r \cup V_c \cup \{\mathbf{null}\}$ with

$$\begin{cases} m_i = j \text{ and } m_j = i & \text{for } (i,j) \in M, \\ m_i = \mathbf{null} & \text{for } i \text{ unmatched.} \end{cases}$$

Augmenting paths in a bipartite graph G can be found by constructing alternating trees. An *alternating tree* $T = (T_r, T_c, E_T)$ is a subgraph of G rooted at a row or column node in which each path from the root of T is an alternating path that begins with an edge not in M. An alternating tree rooted at an unmatched column node j_0 can be grown in the following way. We start with the initial alternating tree $(\emptyset, \{j_0\}, \emptyset)$ and consider all the column nodes $j \in T_c$ in turn. Initially $j = j_0$. For each node j, we check the row nodes $i \in COL(j)$ for which an alternating path from i to j_0 does not yet exist. If node i is already matched, we add row node i, column node m_i, and

edges (i, j) and (i, m_i) to T. If i is not matched, we extend T by row node i and edge (i, j). Now the path in T from node i to the root forms an augmenting path.

Alternating trees can be implemented using a pointer array $p : V_c \to V_c$ such that, given an edge $(i, j) \in E_T \setminus M$, node j is either the root node of the tree, or the edges (i, j), (m_j, j), and (m_j, p_j) are consecutive edges in an alternating path towards the root. Augmenting paths in an alternating tree (provided they exist) can thus easily be obtained from p and m.

Alternating trees are not unique. In general, one can construct several alternating trees starting from the same root node that have equal node sets, but different edge sets. Different alternating trees, in general, will contain different augmenting paths. The matching algorithms that we describe in the next sections impose different criteria on the order in which the paths in the alternating trees are grown in order to obtain augmenting paths and maximum matchings with special properties.

3. Unweighted matching. The asymptotically fastest currently known algorithm for finding a maximum matching is by Hopcroft and Karp [27]. It has a worst-case complexity of $\mathcal{O}(\sqrt{n}\tau)$, where $\tau = |E|$ is the number of entries in the sparse matrix. An efficient implementation of this algorithm can be found in [19]. The algorithm MC21 implemented by Duff [15] has a theoretically worst-case behavior of $\mathcal{O}(n\tau)$, but in practice it behaves more like $\mathcal{O}(n + \tau)$. Because this latter algorithm is simpler, we concentrate on this in what follows although we note that it is possible to use the algorithm of Hopcroft and Karp [27] in a similar way to how we will use MC21 in later sections.

```
for j0 ∈ Vc do
  j := j0;  pj := null;  iap := null;
  B := ∅;
  repeat
    if there exists i ∈ COL(j) and i is unmatched then
      iap := i;
    else
      if there exists i ∈ COL(j) \ B then
        B := B + {i};
        p_{m_i} := j;
        j := m_i;
      else
        j := p_j;
      end if;
    end if;
  until iap ≠ null or j = null;
  if iap ≠ null then augment along path from node iap to node j0;
end for
```

FIG. 3.1. *Outline of* MC21.

MC21 is a depth first search algorithm with look-ahead. It starts off with an empty matching M, and hence all column nodes are unmatched initially. See Figure 3.1. For each unmatched column node j_0 in turn, an alternating tree is grown until an augmenting path with respect to the current matching M is found (provided one exists). A set B is used to mark all the matched row nodes that have been visited so far. Initially, $B = \emptyset$. First, the row nodes in $COL(j_0)$ are searched (look-ahead) for an unmatched

node i_0. If one is found, the singleton path $P = \{(i_0, j_0)\}$ is an M-augmenting path. If there is no such unmatched node, then an unmarked matched node $i_0 \in COL(j_0)$ is chosen, i_0 is marked, and the nodes i_0 and j_1, $j_1 = m_{i_0}$, and the edges (i_0, j_0), (i_0, j_1) are added to the alternating tree (by setting $p_{j_1} = j_0$). The search then continues with column node j_1. For node j_1, the row nodes in $COL(j_1)$ are first checked for an unmatched node. If one exists, say, i_1, then the path $P = \{(i_0, j_0), (i_0, j_1), (i_1, j_1)\}$ forms an augmenting path. If there is no such unmatched node, a remaining unmarked node i_1 is picked from $COL(j_1)$, i_1 is marked, p_{j_2} is set to j_1, $j_2 = m_{i_1}$, and the search moves to node j_2. This continues in a similar (depth first search) fashion until either an augmenting path $P = \{(i_0, j_0), (i_0, j_1), (i_1, j_1), \ldots, (i_k, j_k)\}$ is found (with nodes j_0 and i_k unmatched) or until for some $k > 0$, $COL(j_k)$ does not contain an unmarked node. In the latter case, MC21 backtracks by resuming the search at the previously visited column node j_{k-1} for some remaining unmarked node $i'_{k-1} \in COL(j_{k-1})$. Backtracking for $k = 0$ is not possible; if MC21 resumes the search at column node j_0 and $COL(j_0)$ does not contain an unmarked node, then an M-augmenting path starting at node j_0 does not exist. In this case, MC21 continues with the construction of a new alternating tree starting at the next unmatched column node. (The final maximum matching will have cardinality at most $n - 1$, and hence it will not be perfect.)

4. Weighted matching. In this section, we describe an algorithm that computes a matching for permuting a sparse matrix A such that the product of the diagonal entries of the permuted matrix is maximum in absolute value. That is, the algorithm determines a matching that corresponds to a permutation σ that maximizes

$$\prod_{i=1}^{n} |a_{i\sigma_i}|. \tag{4.1}$$

This maximization multiplicative problem can be translated into a minimization additive problem by defining a matrix $C = (c_{ij})$ as

$$c_{ij} = \begin{cases} \log a_j - \log |a_{ij}|, & a_{ij} \neq 0, \\ \infty & \text{otherwise}, \end{cases}$$

where $a_j = \max_i |a_{ij}|$ is the maximum absolute value in column j of matrix A. Maximizing (4.1) is equal to minimizing

$$\begin{aligned} \log \frac{\prod_{i=1}^{n} a_i}{\prod_{i=1}^{n} |a_{i\sigma_i}|} = \log \frac{\prod_{i=1}^{n} a_{\sigma_i}}{\prod_{i=1}^{n} |a_{i\sigma_i}|} &= \sum_{i=1}^{n} \log a_{\sigma_i} - \sum_{i=1}^{n} \log |a_{i\sigma_i}| \\ &= \sum_{i=1}^{n} c_{i\sigma_i}. \end{aligned} \tag{4.2}$$

Minimizing (4.2) is equivalent to finding a minimum weight perfect matching in an edge weighted bipartite graph. This is known in the literature of linear programming and combinatorial optimization as the bipartite weighted matching or linear sum assignment problem. Numerous algorithms have been proposed for computing minimum weight perfect matchings; see, for example, [6, 7, 8, 13, 22, 24, 29, 30]. A practical example of an assignment problem is the allocation of tasks to people; entry c_{ij} in the cost matrix C represents the cost or benefit of assigning person i to task j.

Let $C = (c_{ij})$ be a real-valued $n \times n$ matrix, $c_{ij} \geq 0$. Let $G_C = (V_r, V_c, E)$ be the corresponding bipartite graph each of whose edges $(i, j) \in E$ has weight c_{ij}. The weight of a matching M in G_C, denoted by $c(M)$, is the sum of its edge weights; i.e.,

$$c(M) = \sum_{(i,j)\in M} c_{ij}.$$

A perfect matching M is said to be a minimum weight perfect matching if it has smallest possible weight; i.e., $c(M) \leq c(M')$ for all possible maximum matchings M'.

The key concept for finding a minimum weight perfect matching is the so-called *shortest* augmenting path. An M-augmenting path P starting at an unmatched column node j is called shortest if $c(M \oplus P) \leq c(M \oplus P')$ for all other possible M-augmenting paths P' starting at node j. We define the length of an alternating path P as

$$l(P) := c(M \oplus P) - c(M) = c(P \setminus M) - c(M \cap P).$$

If P is an augmenting path, $l(P)$ is the cost incurred by changing the matching by augmenting along the path P. A matching M is called *extreme* if and only if there exists no alternating cyclic path with negative length.

The following two relations hold [13, 33]. First, a perfect matching has minimum weight if (and, obviously, only if) it is extreme. Second, if the matching M is extreme and P is a shortest M-augmenting path, then $M \oplus P$ is also extreme. These two relations form the basis for many algorithms for solving the bipartite weighted matching problem: start from any (possibly empty) extreme matching M and successively augment M along shortest augmenting paths until M is maximum (or perfect).

Furthermore (see [30]), a matching M is extreme if and only if there exist *dual* variables u_i and v_j with

$$\begin{cases} u_i + v_j \leq c_{ij} & \text{for } (i,j) \in E, \\ u_i + v_j = c_{ij} & \text{for } (i,j) \in M. \end{cases} \tag{4.3}$$

We define the reduced weight matrix $\overline{C} = (\overline{c}_{ij})$ by

$$\overline{c}_{ij} := c_{ij} - u_i - v_j \geq 0.$$

The weights $\overline{c}_{ij}$ are nonnegative. Finding a minimum weight matching in the graph G_C is equivalent to finding a minimum weight matching in the graph $G_{\overline{C}}$ because

$$\sum_{i=1}^{n} c_{i\sigma_i} = \sum_{i=1}^{n} (\overline{c}_{i\sigma_i} + u_i + v_{\sigma_i}) = \sum_{i=1}^{n} \overline{c}_{i\sigma_i} + \Delta,$$

where Δ is a constant. The reduced weight $\overline{c}(M)$ for the matching M is zero. The reduced length $\overline{l}(P)$ of any M-alternating path P is nonnegative; i.e.,

$$\overline{l}(P) = \sum_{(i,j)\in P\setminus M} \overline{c}_{ij} \ \geq 0,$$

and, if $M \oplus P$ is a matching, the reduced weight of $M \oplus P$ equals

$$\overline{c}(M \oplus P) = \overline{l}(P).$$

Thus, finding a shortest augmenting path in the graph $G_{\overline{C}}$ is equivalent to finding an augmenting path with minimum reduced length. Since $\overline{c}_{ij} = 0$ for every edge $(i,j) \in M$ and the graph $G_{\overline{C}}$ contains no alternating paths P with negative length, $\overline{l}(P') \leq \overline{l}(P)$ for every principal leading subpath P' of P.

The algorithm that we describe in this paper for solving the bipartite weighted matching problem finds shortest augmenting paths by using these reduced weights. Each time a shortest augmenting path with minimum reduced length is found, the algorithm augments the matching M and updates the dual variables u and v (and thereby the reduced weight matrix $\overline{C}$), such that (4.3) again holds. We will describe this update in more detail later in this section.

Shortest augmenting paths in a weighted bipartite graph $G = (V_r, V_c, E)$ can be obtained by means of a shortest alternating path tree. A shortest alternating path tree T is an alternating tree each of whose paths is a shortest path in G. For any node $i \in V_r \cup V_c$, we define d_i as the length of the shortest path in T from node i to the root node ($d_i = \infty$ if no such path exists). T is a shortest alternating path tree if and only if $d_i + \overline{c}_{ij} \geq d_j$ for every edge $(i,j) \in E$ and tree nodes i, j.

An outline of an algorithm for constructing a shortest alternating path tree rooted at column node j_0 is given in Figure 4.1. Figure 4.2 illustrates the search for a shortest augmenting path in an 8×8 matrix, starting at the unmatched column (root) node j_0=c7. Since the reduced weights $\overline{c}_{ij}$ are nonnegative, and the graph $G_{\overline{C}}$ contains no alternating paths with negative length, we can use a sparse variant of Dijkstra's algorithm [14].

Intuitively, the algorithm works as follows. The set of row nodes is partitioned into three sets B, Q, and W. B is the set of (marked) nodes whose shortest alternating paths and distances to node j_0 are known. Q is the set of nodes for which an alternating path to the root is known that is not necessarily the shortest possible. W is the set of nodes for which an alternating path does not exist or is not yet known. (Since W is defined implicitly as $V_r \setminus (B \cup Q)$, it is not actually used in Figure 4.1.) Initially, $B = Q = \emptyset$. We now develop our algorithm using the example in Figure 4.2.

In the first step of the algorithm, the neighboring row nodes of the root node are considered, that is, the nodes in $COL(\text{c7}) = \{\text{r2}, \text{r4}, \text{r6}, \text{r8}\}$. For each matched row node i, the distance $d_i (= \overline{c}_{ij})$ to the root node is set and the node is added to Q. The singleton path $\{(8,7)\}$ is an augmenting path with length 8. Since the edge weights $\overline{c}_{ij}$ are nonnegative, we know that for the row node $i \in Q$ that is closest to j_0, there cannot be another row node in Q or W that has a shorter distance to j_0 and there does not exist a path from j_0 to i that is shorter than the one that is already computed. Therefore, node i can be moved to B. This corresponds to node r6 in Figure 4.2. The same procedure is now repeated with its matched column node m_i=c6. That is, for each neighbor row node i that is matched (and not already in B), the distance d_i to j_0 is updated and the node is added to Q (if it was not already there). If a neighbor row node i is unmatched, a new augmenting path (from root node j_0 to node i) has been found. This path will be marked as the shortest augmenting path if it has length smaller than the shortest augmenting path currently computed (that is, if $d_i < lsap$ in Figure 4.1). After column node c6 has been processed, node r5 has been added to Q and the shortest path currently computed is $\{(7,6),(6,6),(6,7)\}$ and has length 7. Row node r2 of Q is now closest to the root node, and the algorithm continues the search with column node c1. (Dijkstra's algorithm is sometimes referred to as a shortest first search algorithm.)

```
B := ∅;  Q := ∅;
for i ∈ V_r do d_i := ∞;
lsp := 0;   /* length of shortest path from j_0 to any node in Q */
lsap := ∞;   /* length of shortest augmenting path */
j := j_0;  p_j := null;
while true do
   for i ∈ COL(j) \ B do
      dnew := lsp + c̄_ij;
      if dnew < lsap then
         if i unmatched then
            lsap := dnew;  isap := i;
         else
            if dnew < d_i then
               d_i := dnew;  p_{m_i} := j;
               if i ∉ Q then Q := Q + {i};
            end if;
         end if;
      end if;
   end for;
   if Q = ∅ then exit while-loop;
   choose i ∈ Q with minimal d_i;
   lsp := d_i;
   if lsap ≤ lsp then exit while-loop;
   Q := Q − {i};  B := B + {i};
   j := m_i;
end while;
if lsap ≠ ∞ then augment along path from node isap to node j_0;
```

FIG. 4.1. *Construction of a shortest augmenting path.*

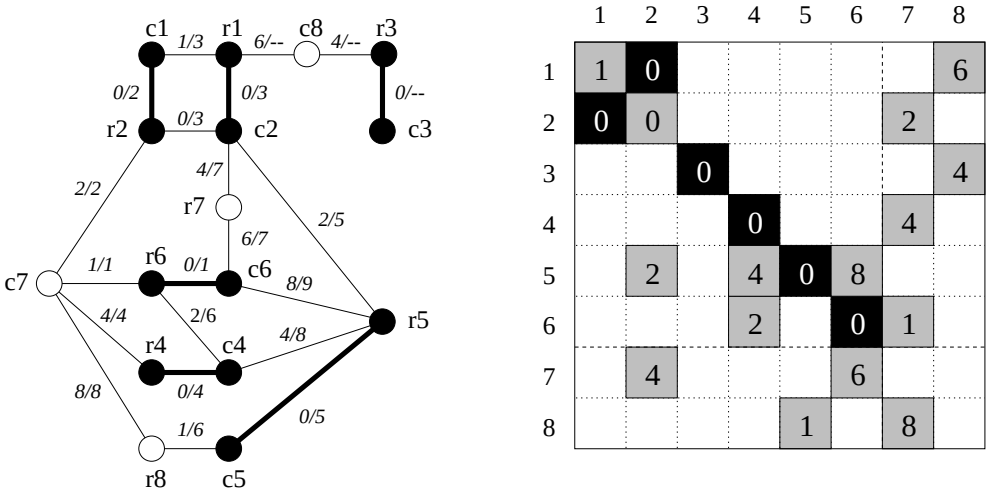


FIG. 4.2. *Example of a shortest augmenting path for an* 8×8 *matrix (with reduced weights* $\bar{c}_{ij}$*) and a matching of cardinality* 6. *The white nodes in the bipartite graph are unmatched, and the black nodes are matched. The matching is represented by the thick edges in the graph and by the black entries in the matrix. The edges in the graph are labelled with* $\bar{c}_{ij}/d_i$*; "--" stands for* $d_i = \infty$*. The shortest augmenting path from the unmatched column node* c7 *is* $\{(2,7),(2,1),(1,1),(1,2),(5,2),(5,5),(8,5)\}$.

At each step of the algorithm, after all the neighboring row nodes of a column node have been considered, the nodes in Q form a front between the nodes in B and the matched row nodes in W. That is, each arbitrary path in the graph $G_{\overline{C}}$ from root node j_0 to a node w in W must contain a (matched) row node i that is in Q and $d_i \leq d_w$, or it must contain an unmatched row node in W. As soon as the distance d_i of the node in Q that is closest to j_0 is larger than or equal to the length $lsap$ of the shortest augmenting path currently computed, we know that there cannot be another (not yet computed) augmenting path with length smaller than $lsap$. At this point, the shortest augmenting path is known (if one exists) and we can stop the search. The search is also stopped when there are no more row node lists of column nodes to be examined; i.e., $Q = \emptyset$. If an augmenting path was found, it is used to augment the current matching. For the example of Figure 4.2, the algorithm continues by scanning the row node lists of column nodes c2, c4, and c5. Then $Q = \emptyset$. The shortest augmenting path is {(2,7),(2,1),(1,1),(1,2),(5,2),(5,5),(8,5)} from row node r7 to column c7. It has length 6.

Dijkstra's algorithm (intended for dense graphs) has $\mathcal{O}(n^2)$ complexity. For sparse problems, the complexity can be reduced to $\mathcal{O}(\tau \log n)$ by implementing the set Q as a k-heap in which the nodes i are sorted by increasing distance d_i from the root (see, for example, [25] and [37]). The theoretical run time of the algorithm is dominated by the operations on the heap Q of which there are $\mathcal{O}(n)$ delete operations, $\mathcal{O}(n)$ insert operations, and $\mathcal{O}(\tau)$ modification operations (these are necessary each time a distance d_i is updated). Each insert and modification operation runs in $\mathcal{O}(\log_k n)$ time, and a delete operation runs in $\mathcal{O}(k \log_k n)$ time. Consequently, the algorithm for finding a shortest augmenting path in a sparse bipartite graph has run time $\mathcal{O}((\tau + kn) \log_k n)$, and the total run time for the sparse bipartite weighted algorithm is $\mathcal{O}(n(\tau + kn) \log_k n)$. If we choose $k = 2$, the algorithm uses binary heaps and we obtain a time bound of $\mathcal{O}(n(\tau + n) \log_2 n)$. If we choose $k = \lceil \tau/n \rceil$ (and $k \geq 2$), we obtain a bound of $\mathcal{O}(n\tau \log_{\tau/n} n)$. To our knowledge, the best known polynomial time bound for solving the assignment problem is $\mathcal{O}(n(\tau + n \log n))$, achieved by Fredman and Tarjan [22] with the use of Fibonacci heaps. (See also [10].)

The actual implementation that we use for the heap Q is similar to the implementation proposed in [13]. Q is a pair (Q_1, Q_2), where Q_1 is an array that contains all the row nodes for which the distance to the root is shortest (lsp), and $Q_2 = Q \setminus Q_1$ is a 2-heap. By separating the nodes in Q that are closest to the root, we may reduce the number of operations on the heap, especially in those situations where the cost matrix C has only a few different numerical values and many alternating paths have the same length. More precisely, if $d_{min} = \min\{d_i | i \in Q\}$, then Q_1 contains the nodes i for which $d_i \leq d_{min} \cdot (1 + \alpha)$. The real parameter α, $\alpha \geq 0$, ensures that round-off error in calculating (the real-valued) path lengths does not lead to a large increase in the number of operations on the heap Q_2. For our experiments in section 7, $\alpha = 10^{-14}$ is sufficient. Deleting a node from Q for which d_i is smallest (see Figure 4.1) now consists of choosing an (arbitrary) element from Q_1. If Q_1 is empty, then we first move all the nodes in Q_2 that are closest to the root to Q_1.

After the matching M is augmented, the reduced weights $\bar{c}_{ij}$ must be updated to ensure that relation (4.3) is satisfied for the new matching M'. This is done by modifying the dual vectors u and v by

$$\begin{cases} u'_i := u_i + d_i - lsap & \text{for } i \in B, \\ v'_j := c_{ij} - u'_i & \text{for all } (i,j) \in M'. \end{cases}$$

The new reduced weights $\bar{c}_{ij} = c_{ij} - u'_i - v'_j$ are nonnegative. Suppose, for simplicity,

that for the example in Figure 4.2, the dual variables for the extreme matching (of cardinality 6) are $u = 0$ and $v = 0$. Then $\bar{c}_{ij} = c_{ij}$. After the search for a shortest augmenting path starting at column node c7 has finished, $d = \{3, 2, \infty, 4, 5, 1, 7, 6\}$, $B = \{1, 2, 4, 5, 6\}$, $isap = 8$, and $lsap = 6$. The dual variables for the new (extreme) matching M' are $u' = \{-3, -4, 0, -2, -1, -5, 0, 0\}$ and $v' = \{4, 3, 0, 2, 0, 5, 6, 0\}$.

The run time of the weighted matching algorithm can be decreased considerably by means of a cheap heuristic that determines an initial extreme matching M that is large. We use the strategy proposed in [7]. We calculate

$$u_i := \min_{j \in ROW(i)} c_{ij} \quad \text{for } i \in V_r$$

and

$$v_j := \min_{i \in COL(j)} (c_{ij} - u_i) \quad \text{for } j \in V_c.$$

An initial extreme matching M can be determined from the edges for which the reduced weight $\bar{c}_{ij} = c_{ij} - u_i - v_j$ is zero. This can be done by scanning the set $COL(j)$ for each column node j to see whether it contains an unmatched row node i for which $\bar{c}_{ij} = 0$. If such a node i exists, edge (i, j) is added to the initial matching M. Then, for each remaining unmatched column node j, every row node $i \in COL(j)$ is considered for which $\bar{c}_{ij} = 0$, and that is matched to a column node other than j, say, j_1. So $(i, j_1) \in M$. If a row node $i_1 \in COL(j_1)$ can be found that is not yet matched and for which $\bar{c}_{i_1 j_1} = 0$, then edge (i, j_1) in M is replaced by edges (i, j) and (i_1, j_1). After having repeated this for all unmatched columns, the search for shortest augmenting paths starts with respect to the current matching.

Finally, we note that the weighted matching algorithm above can also be used for maximizing the sum of the diagonal entries of the matrix A (instead of maximizing the product of the diagonal entries). To do this, we again minimize (4.2), but we redefine the matrix C as

$$c_{ij} = \begin{cases} a_j - |a_{ij}|, & a_{ij} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Maximizing the sum of the diagonal entries is equal to minimizing (4.2), since

$$\sum_{i=1}^{n} a_{\sigma_i} - \sum_{i=1}^{n} |a_{i\sigma_i}| = \sum_{i=1}^{n} (a_{\sigma_i} - |a_{i\sigma_i}|) = \sum_{i=1}^{n} c_{i\sigma_i}.$$

5. Bottleneck matching. We describe a modification of the weighted bipartite matching algorithm from the previous section for permuting rows and columns of a sparse matrix A such that the smallest ratio between the absolute value of a diagonal entry and the maximum absolute value in its column is maximized. That is, the modification computes a permutation σ that maximizes

$$\min_{1 \leq i \leq n} \frac{|a_{i\sigma_i}|}{a_{\sigma_i}}, \tag{5.1}$$

where a_j is the maximum absolute value in column j of the matrix A. Similar to the previous section, we transform this into a minimization problem. We define the matrix $C = (c_{ij})$ as

$$c_{ij} = \begin{cases} 1 - \dfrac{|a_{ij}|}{a_j}, & a_{ij} \neq 0, \\ \infty & \text{otherwise.} \end{cases}$$

Then, maximizing (5.1) is equal to minimizing

$$\max_{1\le i\le n} \frac{a_{\sigma_i} - |a_{i\sigma_i}|}{a_{\sigma_i}} = \max_{1\le i\le n} c_{i\sigma_i}.$$

Given a matching M in the bipartite graph $G_C = (V_r, V_c, E)$, the bottleneck value of M is defined as

$$c(M) = \max_{(i,j)\in M} c_{ij}.$$

The problem is to find a perfect (or maximum) bottleneck matching M for which $c(M)$ is minimal, i.e., $c(M) \le c(M')$ for all possible maximum matchings M'.

Recall that a matching M is called extreme if and only if it does not allow any alternating cyclic path P for which $c(M \oplus P) < c(M)$. The bottleneck algorithm begins with any extreme matching M. The initial bottleneck value b is set to $c(M)$. For each pass through the main loop, an alternating tree is constructed until an augmenting path P is found for which either $c(M\oplus P) = c(M)$ or $c(M\oplus P) - c(M) > 0$ is as small as possible. That is, the algorithm tries to find an augmenting path from an unmatched column node to any unmatched row node such that the bottleneck value of the (augmented) matching does not increase. This will be the case if the weight of the largest edge on the augmenting path is less than or equal to the bottleneck value. If such a path does not exist, then the algorithm will compute an augmenting path that increases the bottleneck value with the smallest possible (nonnegative) amount.

The initializations and the main loop for constructing an augmenting path for the bottleneck algorithm are those of Figure 4.1. Figure 5.1 shows the inner loop of the weighted matching algorithm of Figure 4.1 modified to the case of the bottleneck objective function. There are two main differences. First, instead of an augmenting path with shortest possible length (sum of edge weights), the bottleneck algorithm computes, at each step of the algorithm, an augmenting path whose largest weight of any of its edges is as small as possible. Therefore, the algorithm constructs an alternating path tree, rooted at unmatched column node j_0, for which the largest weight on any of its paths is as small as possible (or less than or equal to the tentative bottleneck value b). The largest weight on the alternating path from root node j_0 to row node i is stored in d_i. As a consequence, the sum operation on the path lengths in Figure 4.1 is replaced by the "max" operation. Second, as soon as an augmenting path P is found whose cost $lsap$ is less than or equal to the current bottleneck value b, the main loop can be exited and the path P is used to augment the current matching M. The bottleneck value b does not change in this case. If such an augmenting path cannot be found, the algorithm continues the search and will eventually exit the main loop with an augmenting path P whose largest edge weight $lsap$ is larger than the current bottleneck value b but smaller than or equal to the largest edge weight of any other augmenting path. The current matching M is then augmented with path P and the bottleneck value b is adjusted to $lsap$. The bottleneck algorithm does not modify the edge weights c_{ij}, that is, it does not use dual variables or reduced edge weights.

Similar to the implementation discussed in section 4, the set Q is implemented as a pair (Q_1, Q_2). Now, the array Q_1 contains all the row nodes i for which $d_i \le b$. Q_2 contains the nodes (not in Q_1) for which d_i is larger than b (but not infinity). Q_2 is again implemented as a 2-heap. If Q_1 is empty and the heap Q_2 is not, then we first move all the nodes $i \in Q_2$ to Q_1 for which d_i is as small as possible, say, d_{min} ($d_{min} > b$), and we adjust the bottleneck value b to d_{min}. The search for augmenting

paths does not require floating-point arithmetic. Therefore, a parameter α that takes into account numerical round-off (as used in the weighted matching algorithm) is not necessary.

The worst-case run time of this bottleneck algorithm is the same as for the weighted matching algorithm, namely $\mathcal{O}(n(\tau + n)\log_2 n)$.

```
for i ∈ COL(j) \ B do
  dnew := max(lsp, c_ij);
  if dnew < lsap then
    if i unmatched then
      lsap := dnew;  isap := i;
      if lsap ≤ b then exit while-loop;
    else
      if dnew < d_i then
        d_i := dnew;  p_{m_i} := j;
        if i ∉ Q then Q := Q + {i};
      end if;
    end if;
  end if;
end for;
```

FIG. 5.1. *Modified inner loop of Figure* 4.1.

A large initial extreme matching can be found in the following way. We define

$$r_i := \min_{j \in ROW(i)} c_{ij} \quad \text{for } i \in V_r$$

and

$$s_j := \min_{i \in COL(j)} c_{ij} \quad \text{for } j \in V_c,$$

as the smallest entry in row i and column j, respectively. Obviously, any perfect matching (with cardinality n) will contain an edge that has a weight that is at least

$$b_0 := \max\left\{\max_i r_i, \max_j s_j\right\}.$$

That is, b_0 is a lower bound for the bottleneck value. An extreme matching M can be obtained from the edges (i, j) for which $c_{ij} \leq b_0$; we scan all nodes $j \in V_c$ in turn and for each node $i \in COL(j)$ that is unmatched and for which $c_{ij} \leq b_0$, edge (i, j) is added to M. Then, for each remaining unmatched column node j, every node $i \in COL(j)$ matched to a column node other than j, say, j_1, and for which $c_{ij} \leq b_0$ is considered. So $(i, j_1) \in M$. If an unmatched row node $i_1 \in COL(j_1)$ can be found for which $c_{i_1 j_1} \leq b_0$, then (i, j_1) in M is replaced by (i, j) and (i_1, j_1). After having done this for all unmatched columns, the search for shortest augmenting paths starts with respect to the current matching.

Other initialization procedures can be found in the literature. For example, a slightly more complicated initialization strategy is used by Finke and Smith [20] in the context of solving transportation problems. For every $i \in V_r$, $j \in V_c$, they use

$$g_i := |\{c_{ik} \mid k \in ROW(i) \text{ and } c_{ik} \leq b_0\}|$$

and

$$h_j \;:=\; |\{c_{kj} \mid k \in COL(j) \text{ and } c_{kj} \le b_0\}|$$

as the number of admissible edges incident to row node i and column node j, respectively. The idea behind using g_i and h_j is that once an admissible edge (i, j) is added to M, all the other admissible edges that are incident to nodes i and j are no longer candidates to be added to M. Therefore, the method tries to pick admissible edges such that the number of admissible edges that become unusable is minimal. First, a row node i with minimal g_i is determined. From the set $ROW(i)$ an admissible entry (i, j) (provided one exists) is chosen for which h_j is minimal and (i, j) is added to M. After deleting the edges (i, k), $k \in ROW(i)$, and the edges (k, j), $k \in COL(j)$, the method repeats the same for another row node i' with minimal $g_{i'}$. This continues until all admissible edges are deleted from the graph.

Finally, we note that instead of maximizing (5.1) we could have also maximized the smallest absolute value on the diagonal. That is, we maximize

$$\min_{1 \le i \le n} |a_{i\sigma_i}|,$$

and define the matrix C as

$$c_{ij} = \begin{cases} a_j - |a_{ij}|, & a_{ij} \neq 0, \\ \infty & \text{otherwise.} \end{cases}$$

Note that this problem is rather sensitive to the scaling of the matrix A. Suppose for example that the matrix A has a column containing only one nonzero entry whose absolute value v is the smallest absolute value present in A. Then, after applying the bottleneck algorithm, the bottleneck value b will be equal to this small value. The smallest entry on the diagonal of the permuted matrix is maximized, but the algorithm did not have any influence on the values of the other diagonal values. Scaling the matrix prior to applying the bottleneck algorithm avoids this difficulty.

In [17], a different approach is taken to obtain a bottleneck matching. Let A_ϵ denote the matrix that is obtained by setting to zero in A all entries a_{ij} for which $|a_{ij}| < \epsilon$ (thus $A_0 = A$) and let M denote any maximum matching for A. Throughout the algorithm, ϵmax and ϵmin are such that a maximum matching of size $|M|$ does not exist for $A_{\epsilon max}$ but does exist for $A_{\epsilon min}$. At each step, ϵ is chosen in the interval $(\epsilon min, \epsilon max)$, and a maximum matching for the matrix A_ϵ is computed using a variant of MC21. If this matching has size $|M|$, then ϵmin is set to ϵ, otherwise ϵmax is set to ϵ. Hence, the size of the interval decreases at each step and ϵ will converge to the bottleneck value. When the algorithm terminates, the last computed matching of size $|M|$ is the bottleneck matching and ϵmin the corresponding bottleneck value. The worst-case run time of this algorithm is $\mathcal{O}(n\tau \log_2 n)$. The term $n\tau$ is the cost for finding a maximum matching with the variant of MC21. The number of times a maximum matching is computed depends on the number q of different numerical values in the matrix. If at each step ϵ is chosen close to the median of the interval $(\epsilon min, \epsilon max)$, then approximately $\log_2 q < \log_2 \tau < 2 \log_2 n$ matchings will be computed.

6. Scaling. Olschowka and Neumaier [33] use the dual solution produced by the weighted matching algorithm to scale the matrix. Let u and v be such that they satisfy relation (4.3). If we define the diagonal matrices

$$D_r \;=\; \text{diag}(p_1, p_2, \ldots, p_n), \quad p_i = \exp(u_i)$$

and

$$D_c \;=\; \mathrm{diag}(q_1, q_2, \ldots, q_n), \quad q_j = \exp(v_j)/a_j,$$

then we have

$$\begin{aligned} & p_i \cdot |a_{ij}| \cdot q_j \\ & \quad = \exp(u_i + \log(|a_{ij}|) + v_j - \log(a_j)) \\ & \quad = \exp(u_i + v_j - (\log(a_j) - \log(|a_{ij}|))) \\ & \quad = \exp(u_i + v_j - c_{ij}) \leq 1. \end{aligned}$$

Equality holds when $u_i + v_j = c_{ij}$, and this is true for all $(i, j) \in M$. The scaled and permuted matrix QD_rAD_c is a matrix whose diagonal entries are one in absolute value and whose off-diagonal entries are all less than or equal to one. Olschowka and Neumaier call such a matrix an *I-matrix* and use this in the context of dense Gaussian elimination to reduce the amount of pivoting that is needed for numerical stability. The more dominant the diagonal of a matrix, the higher the chance that diagonal entries are stable enough to serve as pivots for elimination.

For iterative methods, the transformation of a matrix to an I-matrix is also of interest. For example, from Gershgorin's theorem we know that the union of all discs

$$K_i = \left\{ \mu \in C \;\; | \;\; |\mu - a_{ii}| \leq \sum_{k \neq i} |a_{ik}| \right\}$$

contains all eigenvalues of the $n \times n$ matrix A. Disc K_i has center at a_{ii} and radius that is equal to the sum of the absolute off-diagonal values in row i. If we scale the columns of an I-matrix such that the diagonal entries of an I-matrix are all one, the n discs all have their center at one. The estimate of the eigenvalues will be sharper as A deviates less from a diagonal matrix. That is, the smaller the radii of the discs, the better we know where the eigenvalues are situated. If we are able to reduce the radii of the discs of an I-matrix, i.e., reduce the off-diagonal values, then we tend to cluster the eigenvalues more around one. In the ideal case, all the discs of an I-matrix have a radius smaller than one, in which case the matrix is strictly row-wise diagonally dominant. This guarantees that many types of iterative methods will converge (in exact arithmetic), even simple ones like Jacobi and Gauss–Seidel. However, if at least one disc remains with radius larger than or close to one, zero eigenvalues or small eigenvalues are possible.

Note, however, that this scaling strategy does not guarantee that all the off-diagonal entries of an I-matrix are strictly smaller than one in absolute value.

7. Experimental results. In this section, we discuss cases where the reordering algorithms from the previous section can be useful. These include the solution of sparse equations by a direct method and by preconditioned iterative techniques.

The set of matrices that we used for our experiments are unsymmetric matrices from the Harwell–Boeing Sparse Matrix Test Collection [16] and from the sparse matrix collection at the University of Florida [11].

All matrices are initially row and column scaled. By this we mean that the matrix is scaled so that the maximum entry in each row and in each column is one.

The computer used for the experiments is a SUN UltraSparc with 256 Mbytes of main memory. The algorithms are implemented in Fortran 77.

We use the following acronyms. MC21 is the matching algorithm from the Harwell Subroutine Library [28] for computing a matching such that the corresponding permuted matrix has a zero-free diagonal (see section 3). BT is the bottleneck bipartite matching algorithm from section 5 for permuting a matrix such that the smallest ratio between the absolute value of a diagonal entry and the maximum absolute value in its column is maximized. BT′ is the bottleneck bipartite matching algorithm from [17]. MPD is the weighted matching algorithm from section 4 and computes a permutation such that the product of the diagonal entries of the permuted matrix is maximum in absolute value. MPS is equal to the MPD algorithm but, after the permutation, the matrix is scaled to an I-matrix (see section 6).

TABLE 7.1
Times (in seconds) for the matching algorithms. The order of the matrix is n; the number of entries in the matrix is τ.

Matrix	n	τ	MC21	BT′	BT	MPD
WEST1505	1505	5445	<0.01	<0.01	<0.01	0.02
WEST2021	2021	7353	<0.01	<0.01	<0.01	0.02
MAHINDAS	1258	7682	<0.01	<0.01	<0.01	0.01
ORANI678	2529	90158	0.01	0.07	0.02	0.07
GEMAT11	4929	33185	<0.01	0.03	<0.01	0.03
BAYER01	57735	277774	0.55	1.68	0.37	0.74
LHR01	1477	18592	0.01	0.03	0.05	0.04
LHR02	2954	37206	0.03	0.08	0.07	0.08
LHR14C	14270	307858	0.12	0.71	0.55	1.57
LHR71C	70304	1528092	0.86	6.06	6.18	16.90
ONETONE1	36057	341088	1.30	0.47	0.11	0.39
ONETONE2	36057	227628	1.36	0.35	0.08	0.27
GOODWIN	7320	324784	0.11	1.21	2.12	0.82
AV41092	41092	1683902	17.64	6.44	20.66	29.73

Table 7.1 shows, for our set of large sparse matrices, the order, number of entries, and the time for the algorithms to compute a matching. The times for MPS are not given, because they are almost identical to those for MPD. In general, MC21 needs the least time to compute a matching, except for the ONETONE matrices. For these matrices, the search heuristic that is used in MC21 (a depth first search with look-ahead) does not perform well. There is not a clear winner between the bottleneck algorithms BT and BT′, although we note that BT′ requires the entries inside the columns to be sorted by value. This sorting can be expensive for relatively dense matrices. MPD is in general the most expensive algorithm. We attribute this to the more selective way in which this algorithm constructs augmenting paths.

7.1. Experiments with a direct solution method. For direct methods, putting large entries on the diagonal suggests that pivoting down the diagonal might be more stable. Indeed, stability cannot be guaranteed, but, if we have a solution scheme like the multifrontal method [18], where a symbolic phase chooses the initial pivotal sequence and the subsequent factorization phase then modifies this sequence for stability, it can mean that less modification is required than if the permutation were not applied.

In the multifrontal approach of Duff and Reid [18], later developed by Amestoy and Duff [2], an analysis is performed on the pattern (adjacency graph) of the symmetric matrix $A + A^T$ to obtain an elimination order for the variables that reduces fill-in. Based on this analysis, estimates are computed for the amount of work and the size of the integer and real workspaces that will be needed during the subsequent

TABLE 7.2
Number of delayed pivots during the factorization by MA41. *An "–" indicates that* MA41 *needed more than* 200 *Mbytes of real workspace.*

Matrix	Matching algorithm				
	None	MC21	BT	MPD	MPS
WEST1505	1617	19	8	1	1
WEST2021	2447	27	6	0	0
MAHINDAS	1154	13	0	0	0
ORANI678	2343	9	0	0	0
GEMAT11	–	76	0	0	0
BAYER01	–	28115	8315	3493	0
LHR01	1378	171	42	18	0
LHR02	3432	388	143	56	0
LHR14C	–	7608	1042	169	174
LHR71C	–	35354	7424	2643	3190
ONETONE1	–	16261	298	100	0
ONETONE2	40916	8310	411	100	0
GOODWIN	536	1622	427	53	41
AV41092	–	10151	2141	1730	1722

TABLE 7.3
Number of entries ($\times 10^3$) *in the factors computed by* MA41.

Matrix	Matching algorithm				
	None	MC21	BT	MPD	MPS
WEST1505	531	24	24	24	24
WEST2021	932	32	32	32	32
MAHINDAS	418	46	51	52	52
ORANI678	1923	360	416	423	423
GEMAT11	–	128	79	78	78
BAYER01	–	6272	3534	2945	2801
LHR01	997	137	210	113	111
LHR02	2299	333	374	235	230
LHR14C	–	3111	2676	2164	2165
LHR71C	–	18787	17528	11600	11630
ONETONE1	–	10359	7329	4715	4713
ONETONE2	14083	2876	2298	2170	2168
GOODWIN	1263	2673	2058	1282	1281
AV41092	–	16226	14968	14110	14111

factorization. The numerical factorization is guided by an assembly tree based on the ordering and generated in the analysis phase. At each node of the tree, some steps of Gaussian elimination are performed on a dense submatrix whose Schur complement is then passed to the parent node in the tree where it is assembled (or summed) with Schur complements from the other children and original entries of the matrix. If, however, numerical considerations prevent the selection of a stable pivot in this dense submatrix, then the elimination of a variable is delayed. As a consequence, the Schur complement that is passed to the parent is larger and usually more work and storage will be needed than was predicted by the analysis phase. By permuting the matrix so that there are large entries on the diagonal, before computing the fill reducing ordering, we try to reduce the number of delayed pivots. We show the effect of this in Table 7.2 where we can see that even using MC21 can be very beneficial although the other algorithms can show significant further gains. Tables 7.3 and 7.4 show the effect on the number of entries in the factors and the solution time, respectively. We sometimes observe a dramatic reduction in the fill-in and in the solution time for MA41 when preceded by a permutation.

TABLE 7.4
Solution time (in seconds) required by MA41.

Matrix	Matching algorithm				
	None	MC21	BT	MPD	MPS
WEST1505	1.96	0.04	0.04	0.04	0.04
WEST2021	4.56	0.05	0.05	0.05	0.05
MAHINDAS	1.42	0.09	0.09	0.09	0.09
ORANI678	15.79	2.69	3.14	3.21	3.21
GEMAT11	–	0.15	0.10	0.10	0.10
BAYER01	–	17.81	8.09	8.53	8.33
LHR01	6.04	0.23	0.41	0.16	0.16
LHR02	15.16	0.62	0.65	0.33	0.32
LHR14C	–	7.42	6.10	3.51	3.48
LHR71C	–	73.12	59.41	24.01	24.11
ONETONE1	–	134.98	58.17	26.13	25.58
ONETONE2	44.53	9.26	6.45	6.31	6.12
GOODWIN	2.14	7.77	4.55	2.10	2.10
AV41092	–	124.08	98.91	88.88	88.61

TABLE 7.5
Structural symmetry after permutation (1.00 = *symmetric*).

Matrix	Matching algorithm			
	None	MC21	BT	MPD/MPS
WEST1505	0.002	0.297	0.295	0.292
WEST2021	0.004	0.289	0.297	0.290
MAHINDAS	0.030	0.248	0.183	0.177
ORANI678	0.073	0.077	0.091	0.090
GEMAT11	0.002	0.530	0.947	0.957
BAYER01	<0.01	0.265	0.268	0.255
LHR01	0.009	0.302	0.133	0.168
LHR02	0.009	0.302	0.141	0.168
LHR14C	0.007	0.336	0.125	0.150
LHR71C	0.002	0.384	0.182	0.207
ONETONE1	0.099	0.368	0.427	0.434
ONETONE2	0.148	0.461	0.564	0.574
GOODWIN	0.642	0.288	0.365	0.583
AV41092	0.001	0.101	0.082	0.082

In general, the multifrontal code MA41 will do better on matrices whose structure is symmetric or nearly so. Here, we define the structural symmetry for a matrix A as the number of entries a_{ij} for which a_{ji} is also an entry, divided by the total number of entries. The structural symmetry after the permutations is shown in Table 7.5. We see that in some cases the symmetry of the resulting reordered matrix has increased with respect to the original matrix. This is particularly apparent for very sparse matrices with many zeros on the diagonal, for example, the matrices WEST1505 and WEST2021. For such matrices, the reduction in the number of off-diagonal entries in the reordered matrix has a significant influence on the symmetry.

Our implementations of the algorithms described in this paper have been used successfully by Li and Demmel [31] to avoid the need for numerical pivoting in sparse Gaussian elimination in a distributed-memory environment. Their method partitions the matrix into an $N \times N$ block matrix $A[1 : N, 1 : N]$ by using the notion of unsymmetric supernodes [12]. The blocks are mapped cyclically (in both row and column dimensions) onto the nodes (processors) of a two-dimensional rectangular processor grid. The mapping is such that at step k, $k = 1, \ldots, N$, of the numerical

factorization, a column of processors factorizes the block column $A[k:N,k]$, a row of processors participates in the triangular solves to obtain the block row $U[k,k+1:N]$, and all processors participate in the subsequent multiple-rank update of the remaining matrix $A[k+1:N,k+1:N]$.

The numerical factorization phase in this method does not use (dynamic) partial pivoting on the block columns. This allows the a priori computation of the nonzero structure of the factors, the distributed data structures, the communication pattern, and a static load balancing scheme, which makes the factorization potentially more scalable on distributed-memory machines than factorizations in which the computational and communication tasks only become apparent during the elimination process. To help with numerical stability, the matrix is permuted and scaled before the factorization to make the diagonal entries large compared to the off-diagonal entries, any tiny pivots encountered during the factorization are perturbed, and a few steps of iterative refinement are performed during the triangular solution phase if the solution is not accurate enough. Numerical experiments demonstrate that the method (using our implementation of the MPS algorithm) is as stable as partial pivoting for a wide range of problems [31].

7.2. Experiments with iterative solution methods. For iterative methods, simple techniques like Jacobi or Gauss–Seidel converge more quickly if the diagonal entry is large relative to the off-diagonals in its row or column, and techniques like block iterative methods can benefit if the entries in the diagonal blocks are large. Additionally, for preconditioning techniques, for example, for diagonal preconditioning or incomplete LU preconditioning, it is intuitively evident that large diagonals should be beneficial.

In incomplete factorization preconditioners, pivots are often taken from the diagonal and fill-in is discarded if it falls outside a prescribed sparsity pattern. Incomplete factorizations are used so that the resulting factors are more economical to store, to compute, and to use in the solution process. See [35] for an overview.

One of the reasons incomplete factorizations can behave poorly is that pivots can be arbitrarily small [5, 9]. Pivots may even be zero, in which case the incomplete factorization fails. Small pivots allow the numerical values of the entries in the incomplete factors to become very large, which leads to unstable, and therefore inaccurate, factorizations. In such cases, the norm of the residual matrix $R = A - \hat{L}\hat{U}$ will be large. (Here, $\hat{L}$ and $\hat{U}$ denote the computed incomplete factors.)

A way to improve the stability of the incomplete factorization is to preorder the matrix to put large entries onto the diagonal. Obviously, a successful factorization still cannot be guaranteed, because nonzero diagonal entries may become very small (or even zero) *during* the factorization, but the reordering may mean that zero or small pivots are less likely to occur.

Table 7.6 shows results for three preconditioned Krylov subspace methods GMRES(20) [36], Bi-CGSTAB [38], and TFQMR [23]. The preconditioners considered are the incomplete factorizations ILU(0) and ILU(1) and the threshold incomplete factorization ILUT(tol, p) with drop tolerance $tol = 0.1$ and at most $p = 5$ off-diagonal entries in each row of the incomplete factor (see [35]). The column permutations that were obtained from the matching algorithms were applied to the sparse matrix prior to computing the incomplete factorization. The iteration was stopped when the l_2 norm of the residual was less than 10^{-9} times the l_2 norm of the initial residual. The table shows experimental results for eight matrices of Table 7.1. For the other matrices we did not achieve convergence of the iterative methods. The results show

TABLE 7.6
Number of iterations required by preconditioned iterative methods after unsymmetric (matching) reordering of the matrix. An "–" indicates no convergence within $\min(n, 1000)$ *iterations.*

Matrix + Preconditioner	Iterative method + Matching algorithm											
	GMRES(20)				Bi-CGSTAB				TFQMR			
	MC21	BT	MPD	MPS	MC21	BT	MPD	MPS	MC21	BT	MPD	MPS
WEST1505												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	–	–	–	–	94	68	–	–	–	–
ILUT	–	–	–	–	–	–	–	–	–	–	–	–
WEST2021												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	–	–	–	–	–	–	–	–	–	–
ILUT	–	–	–	–	–	–	–	–	–	–	–	–
MAHINDAS												
ILU(0)	–	–	60	55	–	–	33	31	–	–	33	30
ILU(1)	–	–	36	34	–	–	22	18	–	–	23	19
ILUT	–	178	26	16	–	93	19	10	–	80	17	11
ORANI678												
ILU(0)	–	298	37	36	–	88	24	22	–	100	22	23
ILU(1)	–	34	17	16	–	23	11	11	–	23	13	11
ILUT	–	51	20	15	–	33	15	10	–	33	15	10
GEMAT11												
ILU(0)	–	–	–	–	–	–	246	211	–	–	252	259
ILU(1)	–	–	380	375	–	–	112	85	–	–	101	100
ILUT	–	–	–	–	–	764	401	446	–	–	679	826
BAYER01												
ILU(0)	–	–	–	–	–	–	–	–	–	–	632	466
ILU(1)	–	–	–	–	–	–	–	300	–	–	470	357
ILUT	–	–	–	49	–	–	–	29	–	–	–	25
LHR01												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	95	87	–	–	37	31	–	–	32	31
ILUT	–	–	496	39	–	–	–	24	–	–	381	23
LHR02												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	236	154	–	–	96	60	–	–	53	53
ILUT	–	–	496	39	–	–	534	29	–	–	170	28

that a permutation (and scaling) applied to the initial matrix sometimes greatly improves the convergence. In all cases, the best results in terms of number of iterations are achieved by either the MPD or MPS algorithm. Overall, the MPS ordering and scaling seems to produce the best results.

Obviously, permuting large entries to the diagonal of matrix does not guarantee the accuracy and stability of the incomplete factorization. An inaccurate factorization can also occur in the absence of small pivots, for example, when many fill-ins are dropped from the incomplete factors. Another kind of instability in incomplete factorizations, which can occur with and without small pivots, is severe ill-conditioning of the triangular factors. (In this situation, $||R||_F$ need not be very large, but $||I - A(\hat{L}\hat{U})^{-1}||_F$ will be.) This is also a common situation when the coefficient matrix is far from diagonally dominant.

More accurate and more stable incomplete factors may be achieved by combining the unsymmetric permutations that place large entries onto the diagonal with a symmetric permutation. A symmetric permutation $P^T BP$ of an unsymmetric matrix B reorders the rows and columns of B in the same way and can be obtained by using the adjacency graph of the symmetric matrix $B + B^T$. A sparse linear system $Ax = b$ is

TABLE 7.7

Number of iterations required by preconditioned iterative methods after unsymmetric (matching) reordering followed by a symmetric reverse Cuthill–McKee reordering of the matrix.

Matrix + Preconditioner	Iterative method + Matching algorithm											
	GMRES(20)				Bi-CGSTAB				TFQMR			
	MC21	BT	MPD	MPS	MC21	BT	MPD	MPS	MC21	BT	MPD	MPS
WEST1505												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	–	317	–	–	–	–	–	–	96	73
ILUT	–	–	–	–	–	–	139	–	–	–	389	–
WEST2021												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	–	–	–	–	74	85	–	–	108	94
ILUT	–	–	–	–	–	–	359	–	–	–	–	–
MAHINDAS												
ILU(0)	–	–	55	53	–	–	62	30	–	–	35	28
ILU(1)	–	176	28	27	–	105	18	18	–	121	17	18
ILUT	–	191	19	16	–	114	22	9	–	156	17	10
ORANI678												
ILU(0)	–	86	35	34	–	55	26	19	–	54	21	23
ILU(1)	–	33	18	17	–	23	12	12	–	23	13	12
ILUT	–	31	19	18	–	19	14	12	–	21	16	14
GEMAT11												
ILU(0)	–	–	572	624	–	–	141	170	–	–	148	151
ILU(1)	–	178	90	90	–	60	50	49	–	65	56	56
ILUT	–	–	–	–	–	490	263	265	–	499	407	399
BAYER01												
ILU(0)	–	–	–	–	–	–	–	–	–	–	481	317
ILU(1)	–	–	–	–	–	–	–	770	–	–	535	408
ILUT	–	–	–	16	–	–	–	11	–	–	–	11
LHR01												
ILU(0)	–	–	218	137	–	–	47	40	–	–	60	48
ILU(1)	–	–	180	123	–	–	43	39	–	–	39	37
ILUT	–	–	474	35	–	–	265	19	–	–	267	21
LHR02												
ILU(0)	–	–	–	–	–	–	–	–	–	–	–	–
ILU(1)	–	–	–	369	–	–	90	104	–	–	69	66
ILUT	–	–	395	197	–	–	195	61	–	–	143	55

then transformed into a system $P^T(QD_rAD_c)Py = P^TQD_rb$; i.e., the symmetric permutation is applied to the matrix QD_rAD_c. The solution to the original linear system is $x = D_cPy$. The incomplete factorization preconditioners may benefit from a symmetric permutation since they are sensitive to the ordering of the rows and columns of the matrix. Table 7.7 shows the results that we obtained by combining the unsymmetric matching permutations with the reverse Cuthill–McKee ordering [26, 32]. In most cases, the reverse Cuthill–McKee ordering has a positive effect, but this is not always true. Benzi, Haws, and Tůma [4] have experimented extensively with our matching algorithms and combined them with the reverse Cuthill–McKee ordering, the multiple minimum degree ordering, and a generalized nested dissection ordering. Their main motivation for also using a symmetric permutation is that the number of entries that is dropped in incomplete factorization preconditioners can be reduced by applying a reordering of the matrix that reduces fill-in. The extensive experimental results presented in [4] show that the reliability and performance of preconditioned iterative solvers can be further enhanced by such combined preprocessing.

Finally, we mention that we also performed numerous experiments with the implementation of the block Cimmino method that is described in [3]. This iterative

scheme is equivalent to using a block Jacobi algorithm on the normal equations. The subproblems corresponding to blocks of rows from the matrix are solved by the sparse direct method MA27 [28]. In the experiments, the matching algorithm was followed by a reverse Cuthill–McKee algorithm to obtain a block tridiagonal form. We partitioned the matrix into blocks of rows of varying sizes (2, 4, 8, and 16). The accelerations used were block CG algorithms, also of varying sizes (1, 4, and 8). We chose the block rows to be of equal (or nearly equal) size. Table 7.8 shows a small set of results that is representative for many of the experimental results that we obtained. In general, we noticed in our experiments that the block Cimmino method often was more sensitive to the scaling (in MPS) and less to the reorderings. The convergence properties of the block Cimmino method are independent of row scaling. However, the sparse direct solver MA27 used for solving the augmented systems performs numerical pivoting during the factorizations of the augmented matrices. Row scaling might well change the choice of the pivot order and affect the fill-in in the factors and the accuracy of the solution. Column scaling should affect convergence of the method since it can be considered as a diagonal preconditioner. For more details, see [34].

TABLE 7.8
Number of iterations required by the block Cimmino algorithm with CG(4) acceleration for the matrix MAHINDAS.

# block rows	Matching algorithm				
	None	MC21	BT	MPD	MPS
2	148	112	130	133	68
4	212	190	199	194	92
8	261	235	232	233	111
16	281	245	253	253	112

8. Conclusions and future directions of work. We have considered, in sections 3–5, techniques for permuting a sparse matrix so that the diagonal of the permuted matrix has entries of large absolute value. We discussed various criteria for this and considered their implementation as computer codes. We also considered in section 6 possible scaling strategies to further improve the weight of the diagonal with respect to the off-diagonal values. In section 7, we indicated cases where such a permutation (and scaling) can be useful. These include the solution of sparse equations by a direct method and by an iterative technique. We also considered its use in generating a preconditioner for an iterative method.

The experimental results in section 7 show that the proposed reordering and scaling algorithms can have a significant effect on the performance of various methods for solving sparse systems. However, at present it is still somewhat less clear that there is a universal strategy that performs well in general. One reason for this is that increasing the size of only the diagonal is not always sufficient to improve the performance of the method. For example, for the incomplete preconditioners that we used for the numerical experiments in section 7.2, it is not only the size of the diagonal but also the amount and size of the discarded fill-in that plays an important role.

It is, therefore, interesting to combine the reordering and scaling strategies described in sections 3–6 with other reordering strategies. The experiments performed by Benzi, Haws, and Tůma [4], who used our algorithms in combination with symmetric matrix orderings, can be seen as a step in that direction. A combination with other scaling strategies is also possible. As an example of this, we mention a possible way of decreasing large off-diagonal entries of an I-matrix by row and column equal-

ization [33]. Let A be an I-matrix that (for simplicity) does not contain zero entries. We define the matrix $C = (c_{ij})$ as $c_{ij} = \log |a_{ij}|$. Equalization consists of repeatedly equalizing the largest absolute values in row i and column i:

$t := 0$;
for $k := 1, 2, \ldots$ **do**
  for $j := 1$ **to** n **do**
    $y_1 := \max\{c_{jr} + t_j - t_r \mid r \neq j \text{ and } c_{jr} \neq 0\}$;
    $y_2 := \max\{c_{rj} + t_r - t_j \mid r \neq j \text{ and } c_{rj} \neq 0\}$;
    $t_j := t_j + (y_2 - y_1)/2$;
  end;
end;

For $k = \infty$, this algorithm minimizes $\max\{c_{ij} + t_i - t_j \mid i \neq j,\ c_{ij} \neq 0\}$. If we define diagonal scaling matrices D_r and D_c as in section 6, but now $p_i := \exp(t_i)$ and $q_j := 1/\exp(t_j)$, then the algorithm minimizes the largest off-diagonal absolute value in the matrix $D_r A D_c$. The diagonal entries do not change. Unfortunately, this equalization strategy is costly and perhaps not practical in an actual implementation.

The run time of the minimum weighted matching algorithms can be reduced by using heuristics that more quickly find matchings of small but not minimum weight. This will be of interest as long as the benefit of using the weighted matching (with, for example, the incomplete factorization preconditioners) is not too much reduced. We note that the weighted matching algorithm described in section 4 can be used to find matchings with small weight by using larger values for the parameter α. A similar parameter can be inserted in the bottleneck matching algorithm of section 5.

Another interesting direction of research would be to develop an algorithm to obtain a partitioned matrix where the diagonal blocks have entries of large magnitude relative to the entries in the off-diagonal blocks. This is of particular interest for the block Cimmino method. One could also build other criteria into the weighting for obtaining a bipartite matching, for example, to incorporate a Markowitz cost so that sparsity would also be preserved by the choice of the resulting diagonal as a pivot. Such combination would make the resulting ordering suitable for a wider class of sparse direct solvers.

Finally, we plan to continue the development of the algorithms for the bipartite matchings. Preliminary experimental results with a tuned version of the weighted matching algorithm show that the time required by the MPD algorithm for matrix AV41092 can be reduced by about a factor of two, without degrading the quality of the matching. Up-to-date copies of the software can be obtained by contacting one of the authors.

Acknowledgments. We are grateful to Michele Benzi of Los Alamos National Laboratory and Miroslav Tůma of the Czech Academy of Sciences for their assistance on the preconditioned iterative methods and to Daniel Ruiz of ENSEEIHT for his help on block iterative methods. We would also like to thank two anonymous referees for their constructive criticism on an early draft of the paper.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[2] P. R. Amestoy and I. S. Duff, *Vectorization of a multiprocessor multifrontal code*, Internat. J. Supercomputer Appl., 3 (1989), pp. 41–59.

[3] M. Arioli, I. S. Duff, J. Noailles, and D. Ruiz, *A block projection method for sparse matrices*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 47–70.
[4] M. Benzi, J. C. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.
[5] M. Benzi, D. B. Szyld, and A. van Duin, *Orderings for incomplete factorization preconditioning of nonsymmetric problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1652–1670.
[6] R. E. Burkard and U. Derigs, *Assignment and Matching Problems: Solution Methods with FORTRAN-Programs*, Lecture Notes in Econom. and Math. Systems 184, Springer, Berlin, Heidelberg, New York, 1980.
[7] G. Carpaneto and P. Toth, *Solution of the assignment problem (Algorithm* 548*)*, ACM Trans. Math. Software, 1980, pp. 104–111.
[8] P. Carraresi and C. Sodini, *An efficient algorithm for the bipartite matching problem*, European J. Oper Res., 23 (1986), 86–93.
[9] E. Chow and Y. Saad, *Experimental Study of ILU Preconditioners for Indefinite Matrices*, Technical Report TR 97/95, Department of Computer Science, University of Minnesota, and Minnesota Supercomputer Institute, Minneapolis, MN, 1997.
[10] T. Corman, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
[11] T. A. Davis, *University of Florida sparse matrix collection*, http://www.cise.ufl.edu/∼davis and ftp://ftp.cise.ufl.edu/pub/faculty/davis (1997).
[12] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.
[13] U. Derigs and A. Metz, *An efficient labeling technique for solving sparse assignment problems*, Computing, 36 (1986), pp. 301–311.
[14] E. W. Dijkstra, *A note on two problems in connection with graphs*, Numer. Math., 1 (1959), pp. 269–271.
[15] I. S. Duff, *Algorithm* 575. *Permutations for a zero-free diagonal*, ACM Trans. Math. Software, 7 (1981), pp. 387–390.
[16] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release* 1*)*, Technical Report RAL-92-086, Rutherford Appleton Laboratory, Oxfordshire, England, 1992.
[17] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.
[18] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.
[19] I. S. Duff and T. Wiberg, *Remarks on implementations of* $O(n^{1/2}\tau)$ *assignment algorithms*, ACM Trans. Math. Software, 14 (1988), pp. 267–287.
[20] G. Finke and P. Smith, *Primal equivalents for the threshold algorithm*, in Proceedings of the Third Symposium on Operations Research, University of Mannheim, Mannheim, Germany, 1978, Operations Res. Verfahren 31, 1979, pp. 185–198.
[21] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
[22] M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, J. Assoc. Comput. Mach., 34 (1987), pp. 596–615.
[23] R. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 470–482.
[24] H. N. Gabow, *Scaling algorithms for network problems*, J. Comput. System Sci., 31 (1985), pp. 148–168.
[25] G. Gallo and S. Pallottino, *Shortest path algorithms*, Ann. Oper. Res., 13 (1988), pp. 3–79.
[26] A. George, *Computer Implementation of the Finite-Element Method*, Ph.D. thesis, Report STAN CS-71-208, Department of Computer Science, Stanford University, Stanford, CA, 1971.
[27] J. E. Hopcroft and R. M. Karp, *An* $n^{5/2}$ *algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.
[28] HSL, *A collection of Fortran codes for large scale scientific computation*, http://www.numerical.rl.ac.uk/hsl (2000).
[29] R. Jonker and A. Volgenant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing, 38 (1987), pp. 325–340.
[30] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.
[31] X. S. Li and J. W. Demmel, *A scalable sparse direct solver using static pivoting*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San

Antonio, Texas, 1999, CD-ROM, SIAM, Philadelphia, PA, 1999.
[32] W. H. Liu and A. H. Sherman, *Comparative analysis of the Cuthill–McKee and the reverse Cuthill–McKee ordering algorithms for sparse matrices*, SIAM J. Numer. Anal., 13 (1976), pp. 198–213.
[33] M. Olschowka and A. Neumaier, *A new pivoting strategy for Gaussian elimination*, Linear Algebra Appl., 240 (1996), pp. 131–151.
[34] D. Ruiz, *Solution of Large Sparse Unsymmetric Linear Systems with a Block Iterative Method in a Multiprocessor Environment*, Ph.D. thesis, CERFACS, Toulouse, France, 1992.
[35] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, MA, 1996.
[36] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[37] R. E. Tarjan, *Data Structures and Network Algorithms*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 44, SIAM, Philadelphia, PA, 1983.
[38] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 997–1013

MULTIPLE-RANK MODIFICATIONS OF A SPARSE CHOLESKY FACTORIZATION*

TIMOTHY A. DAVIS† AND WILLIAM W. HAGER‡

Abstract. Given a sparse symmetric positive definite matrix $\mathbf{AA}^\mathsf{T}$ and an associated sparse Cholesky factorization $\mathbf{LDL}^\mathsf{T}$ or $\mathbf{LL}^\mathsf{T}$, we develop sparse techniques for updating the factorization after either adding a collection of columns to $\mathbf{A}$ or deleting a collection of columns from $\mathbf{A}$. Our techniques are based on an analysis and manipulation of the underlying graph structure, using the framework developed in an earlier paper on rank-1 modifications [T. A. Davis and W. W. Hager, *SIAM J. Matrix Anal. Appl.*, 20 (1999), pp. 606–627]. Computationally, the multiple-rank update has better memory traffic and executes much faster than an equivalent series of rank-1 updates since the multiple-rank update makes one pass through $\mathbf{L}$ computing the new entries, while a series of rank-1 updates requires multiple passes through $\mathbf{L}$.

Key words. numerical linear algebra, direct methods, Cholesky factorization, sparse matrices, mathematical software, matrix updates

AMS subject classifications. 65F05, 65F50, 65-04

PII. S0895479899357346

1. Introduction.

This paper presents a method for evaluating a multiple-rank update or downdate of the sparse Cholesky factorization $\mathbf{LDL}^\mathsf{T}$ or $\mathbf{LL}^\mathsf{T}$ of the matrix $\mathbf{AA}^\mathsf{T}$, where $\mathbf{A}$ is $m \times n$. More precisely, given an $m \times r$ matrix $\mathbf{W}$, we evaluate the Cholesky factorization of $\mathbf{AA}^\mathsf{T} + \sigma\mathbf{WW}^\mathsf{T}$ where either σ is $+1$ (corresponding to an update) and $\mathbf{W}$ is arbitrary, or σ is -1 (corresponding to a downdate) and $\mathbf{W}$ consists of columns of $\mathbf{A}$. Both $\mathbf{AA}^\mathsf{T}$ and $\mathbf{AA}^\mathsf{T} + \sigma\mathbf{WW}^\mathsf{T}$ must be positive definite. It follows that $n \geq m$ in the case of an update, and $n - r \geq m$ in the case of a downdate.

One approach to the multiple-rank update is to express it as a series of rank-1 updates and use the theory developed in [10] for updating a sparse factorization after a rank-1 change. This approach, however, requires multiple passes through $\mathbf{L}$ as it is updated after each rank-1 change. In this paper, we develop a sparse factorization algorithm that makes only one pass through $\mathbf{L}$.

For a dense Cholesky factorization, a one-pass algorithm to update a factorization is obtained from Method C1 in [18] by making all the changes associated with one column of $\mathbf{L}$ before moving to the next column, as is done in the following algorithm that overwrites $\mathbf{L}$ and $\mathbf{D}$ with the new factors of $\mathbf{AA}^\mathsf{T} + \sigma\mathbf{WW}^\mathsf{T}$. Algorithm 1 performs $2rm^2 + 4rm$ floating-point operations.

ALGORITHM 1 (dense rank-r update/downdate).

for $i = 1$ **to** r **do**
    $\alpha_i = 1$
end for
for $j = 1$ **to** m **do**
    for $i = 1$ **to** r **do**

*Received by the editors June 17, 1999; accepted for publication (in revised form) by S. Vavasis August 16, 2000; published electronically January 31, 2001. This work was supported by the National Science Foundation.

http://www.siam.org/journals/simax/22-4/35734.html

†Department of Computer and Information Science and Engineering, University of Florida, P.O. Box 116120, Gainesville, FL 32611-6120 (davis@cise.ufl.edu, http://www.cise.ufl.edu/~davis).

‡Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/~hager).

```
        ᾱ = α_i + σ w_ji^2 / d_j  (σ = +1 for update or −1 for downdate)
        d_j = d_j ᾱ
        γ_i = w_ji / d_j
        d_j = d_j / α_i
        α_i = ᾱ
    end for
    for p = j + 1 to m do
        for i = 1 to r do
            w_pi = w_pi − w_ji l_pj
            l_pj = l_pj + σ γ_i w_pi
        end for
    end for
end for
```

We develop a sparse version of this algorithm that only accesses and modifies those entries in $\mathbf{L}$ and $\mathbf{D}$ which can change. For $r = 1$, the theory in our rank-1 paper [10] shows that those columns which can change correspond to the nodes in an elimination tree on a path starting from the node k associated with the first nonzero element w_{k1} in $\mathbf{W}$. For $r > 1$ we show that the columns of $\mathbf{L}$ which can change correspond to the nodes in a subtree of the elimination tree, and we express this subtree as a modification of the elimination tree of $\mathbf{AA}^{\mathsf{T}}$. Also, we show that with a reordering of the columns of $\mathbf{W}$, it can be arranged so that in the inner loop where elements in row p of $\mathbf{W}$ are updated, the elements that change are adjacent to each other. The sparse techniques that we develop lead to sequential access of matrix elements and to efficient computer memory traffic. These techniques to modify a sparse factorization have many applications, including the linear program dual active set algorithm (LPDASA) [20], least-squares problems in statistics, the analysis of electrical circuits and power systems, structural mechanics, sensitivity analysis in linear programming, boundary condition changes in partial differential equations, domain decomposition methods, and boundary element methods (see [19]).

Section 2 describes our notation. In section 3, we present an algorithm for computing the *symbolic factorization* of $\mathbf{AA}^{\mathsf{T}}$ using multisets, which determines the location of nonzero entries in $\mathbf{L}$. Sections 4 and 5 describe our multiple-rank symbolic update and downdate algorithms for finding the nonzero pattern of the new factors. Section 6 describes our algorithm for computing the new numerical values of $\mathbf{L}$ and $\mathbf{D}$, for either an update or downdate. Our experimental results are presented in section 7.

2. Notation and background. Given the location of the nonzero elements of $\mathbf{AA}^{\mathsf{T}}$, we can perform a *symbolic factorization* (this terminology is introduced by George and Liu in [15]) of the matrix to predict the location of the nonzero elements of the Cholesky factor $\mathbf{L}$. In actuality, some of these predicted nonzeros may be zero due to numerical cancellation during the factorization process. The statement "$l_{ij} \neq 0$" will mean that l_{ij} is *symbolically* nonzero. The main diagonals of $\mathbf{L}$ and $\mathbf{D}$ are always nonzero since the matrices that we factor are positive definite (see [26, p. 253]). The nonzero pattern of column j of $\mathbf{L}$ is denoted $\mathcal{L}_j$,

$$\mathcal{L}_j = \{i \,:\, l_{ij} \neq 0\},$$

while $\mathcal{L}$ denotes the collection of patterns

$$\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m\}.$$

Similarly, $\mathcal{A}_j$ denotes the nonzero pattern of column j of $\mathbf{A}$,

$$\mathcal{A}_j = \{i \,:\, a_{ij} \neq 0\},$$

while $\mathcal{A}$ is the collection of patterns

$$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}.$$

The *elimination tree* can be defined in terms of a *parent map* π (see [22]). For any node j, $\pi(j)$ is the row index of the first nonzero element in column j of $\mathbf{L}$ beneath the diagonal element

$$\pi(j) = \min\ \mathcal{L}_j \setminus \{j\},$$

where "min $\mathcal{X}$" denotes the smallest element of $\mathcal{X}$:

$$\min\ \mathcal{X} = \min_{i \in \mathcal{X}}\ i.$$

Our convention is that the min of the empty set is zero. Note that $j < \pi(j)$ except in the case where the diagonal element in column j is the only nonzero element. The children of node j is the set of nodes whose parent is j:

$$\{c : j = \pi(c)\}.$$

The *ancestors* of a node j, denoted $\mathcal{P}(j)$, is the set of successive parents:

$$\mathcal{P}(j) = \{j, \pi(j), \pi(\pi(j)), \ldots\}.$$

Since $\pi(j) > j$ for each j, the ancestor sequence is finite. The sequence of nodes $j, \pi(j), \pi(\pi(j)), \ldots$, forming $\mathcal{P}(j)$, is called the *path* from j to the associated tree root, the final node on the path. The collection of paths leading to a root form an *elimination tree*. The set of all trees is the *elimination forest*. Typically, there is a single tree whose root is m; however, if column j of $\mathbf{L}$ has only one nonzero element, the diagonal element, then j will be the root of a separate tree.

The number of elements (or size) of a set $\mathcal{X}$ is denoted $|\mathcal{X}|$, while $|\mathcal{A}|$ or $|\mathcal{L}|$ denote the sum of the sizes of the sets they contain.

3. Symbolic factorization. For a matrix of the form $\mathbf{A}\mathbf{A}^{\mathsf{T}}$, the pattern $\mathcal{L}_j$ of column j is the union of the patterns of each column of $\mathbf{L}$ whose parent is j and each column of $\mathbf{A}$ whose smallest row index of its nonzero entries is j (see [16, 22]):

$$\mathcal{L}_j = \{j\} \cup \left(\bigcup_{\{c : j = \pi(c)\}} \mathcal{L}_c \setminus \{c\} \right) \cup \left(\bigcup_{\min\ \mathcal{A}_k = j} \mathcal{A}_k \right). \tag{3.1}$$

To modify (3.1) during an update or downdate, without recomputing it from scratch, we need to keep track of how each entry i entered into $\mathcal{L}_j$ [10]. For example, if $\pi(c)$ changes, we may need to remove a term $\mathcal{L}_c \setminus \{c\}$. We cannot simply perform a set subtraction, since we may remove entries that appear in other terms. To keep track of how entries enter and leave the set $\mathcal{L}_j$, we maintain a multiset associated with column j. It has the form

$$\mathcal{L}_j^{\sharp} = \{(i, m(i,j)) : i \in \mathcal{L}_j\},$$

where the multiplicity $m(i,j)$ is the number of children of j that contain row index i in their pattern plus the number of columns of $\mathcal{A}$ whose smallest entry is j and that contain row index i. Equivalently, for $i \neq j$,

$$m(i,j) = |\{c : j = \pi(c) \text{ and } i \in \mathcal{L}_c\}| + |\{k : \min\ \mathcal{A}_k = j \text{ and } i \in \mathcal{A}_k\}|\,.$$

For $i = j$, we increment the above equation by one to ensure that the diagonal entries never disappear during a downdate. The set $\mathcal{L}_j$ is obtained from $\mathcal{L}_j^\sharp$ by removing the multiplicities.

We define the addition of a multiset $\mathcal{X}^\sharp$ and a set $\mathcal{Y}$ in the following way:

$$\mathcal{X}^\sharp + \mathcal{Y} = \{(i, m'(i))\ :\ i \in \mathcal{X} \text{ or } i \in \mathcal{Y}\},$$

where

$$m'(i) = \begin{cases} 1 & \text{if } i \notin \mathcal{X} \text{ and } i \in \mathcal{Y}, \\ m(i) & \text{if } i \in \mathcal{X} \text{ and } i \notin \mathcal{Y}, \\ m(i)+1 & \text{if } i \in \mathcal{X} \text{ and } i \in \mathcal{Y}. \end{cases}$$

Similarly, the subtraction of a set $\mathcal{Y}$ from a multiset $\mathcal{X}^\sharp$ is defined by

$$\mathcal{X}^\sharp - \mathcal{Y} = \{(i, m'(i))\ :\ i \in \mathcal{X} \text{ and } m'(i) > 0\},$$

where

$$m'(i) = \begin{cases} m(i) & \text{if } i \notin \mathcal{Y}, \\ m(i)-1 & \text{if } i \in \mathcal{Y}. \end{cases}$$

The multiset subtraction of $\mathcal{Y}$ from $\mathcal{X}^\sharp$ undoes a prior addition. That is, for any multiset $\mathcal{X}^\sharp$ and any set $\mathcal{Y}$, we have

$$((\mathcal{X}^\sharp + \mathcal{Y}) - \mathcal{Y}) = \mathcal{X}^\sharp.$$

In contrast $((\mathcal{X} \cup \mathcal{Y}) \setminus \mathcal{Y})$ is equal to $\mathcal{X}$ if and only if $\mathcal{X}$ and $\mathcal{Y}$ are disjoint sets.

Using multiset addition instead of set union, (3.1) leads to the following algorithm for computing the symbolic factorization of $\mathbf{A}\mathbf{A}^\mathsf{T}$.

ALGORITHM 2 (symbolic factorization of $\mathbf{A}\mathbf{A}^\mathsf{T}$, using multisets).

for $j = 1$ **to** m **do**
  $\mathcal{L}_j^\sharp = \{(j,1)\}$
  for each c such that $j = \pi(c)$ **do**
    $\mathcal{L}_j^\sharp = \mathcal{L}_j^\sharp + (\mathcal{L}_c \setminus \{c\})$
  end for
  for each k where $\min\ \mathcal{A}_k = j$ **do**
    $\mathcal{L}_j^\sharp = \mathcal{L}_j^\sharp + \mathcal{A}_k$
  end for
  $\pi(j) = \min\ \mathcal{L}_j \setminus \{j\}$
end for

4. Multiple-rank symbolic update. We consider how the pattern $\mathcal{L}$ changes when $\mathbf{A}\mathbf{A}^\mathsf{T}$ is replaced by $\mathbf{A}\mathbf{A}^\mathsf{T} + \mathbf{W}\mathbf{W}^\mathsf{T}$. Since

$$\mathbf{A}\mathbf{A}^\mathsf{T} + \mathbf{W}\mathbf{W}^\mathsf{T} = [\mathbf{A}|\mathbf{W}][\mathbf{A}|\mathbf{W}]^\mathsf{T},$$

we can in essence augment $\mathbf{A}$ by $\mathbf{W}$ in order to evaluate the new pattern of column j in $\mathbf{L}$. According to (3.1), the new pattern $\overline{\mathcal{L}}_j$ of column j of $\mathbf{L}$ after the update is

$$\overline{\mathcal{L}}_j = \{j\} \cup \left(\bigcup_{\{c: j = \overline{\pi}(c)\}} \overline{\mathcal{L}}_c \setminus \{c\} \right) \cup \left(\bigcup_{\min\, \mathcal{A}_k = j} \mathcal{A}_k \right) \cup \left(\bigcup_{\min\, \mathcal{W}_i = j} \mathcal{W}_i \right), \tag{4.1}$$

where $\mathcal{W}_i$ is the pattern of column i in $\mathbf{W}$. Throughout, we put a bar over a matrix or a set to denote its new value after the update or downdate.

In the following theorem, we consider a column j of the matrix $\mathbf{L}$ and how its pattern is modified by the sets $\mathcal{W}_i$. Let $\overline{\mathcal{L}}_j^\sharp$ denote the multiset for column j after the rank-r update or downdate has been applied.

THEOREM 4.1. *To compute the new multiset $\overline{\mathcal{L}}_j^\sharp$, initialize $\overline{\mathcal{L}}_j^\sharp = \mathcal{L}_j^\sharp$ and perform the following modifications.*

- *Case A: For each i such that $j = \min \mathcal{W}_i$, add $\mathcal{W}_i$ to the pattern for column j,*

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + \mathcal{W}_i.$$

- *Case B: For each c such that $j = \pi(c) = \overline{\pi}(c)$, compute*

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \mathcal{L}_c)$$

  (c is a child of j in both the old and new elimination tree).
- *Case C: For each c such that $j = \overline{\pi}(c) \neq \pi(c)$, compute*

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \{c\})$$

  (c is a child of j in the new tree, but not the old one).
- *Case D: For each c such that $j = \pi(c) \neq \overline{\pi}(c)$, compute*

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \{c\})$$

  (c is a child of j in the old tree, but not the new one).

Proof. Cases A–D account for all the adjustments we need to make in $\mathcal{L}_j$ in order to obtain $\overline{\mathcal{L}}_j$. These adjustments are deduced from a comparison of (3.1) with (4.1). In case A, we simply add in the $\mathcal{W}_i$ multisets of (4.1) that do not appear in (3.1). In case B, node c is a child of node j both before and after the update. In this case, we must adjust for the deviation between $\overline{\mathcal{L}}_c$ and $\mathcal{L}_c$. By [10, Prop. 3.2], after a rank-1 update, $\mathcal{L}_c \subseteq \overline{\mathcal{L}}_c$. If $\mathbf{w}_i$ denotes the ith column of $\mathbf{W}$, then

$$\mathbf{W}\mathbf{W}^\mathsf{T} = \mathbf{w}_1\mathbf{w}_1^\mathsf{T} + \mathbf{w}_2\mathbf{w}_2^\mathsf{T} + \cdots + \mathbf{w}_r\mathbf{w}_r^\mathsf{T}.$$

Hence, updating $\mathbf{A}\mathbf{A}^\mathsf{T}$ by $\mathbf{W}\mathbf{W}^\mathsf{T}$ is equivalent to r successive rank-1 updates of $\mathbf{A}\mathbf{A}^\mathsf{T}$. By repeated application of [10, Prop. 3.2], $\mathcal{L}_c \subseteq \overline{\mathcal{L}}_c$ after a rank-r update of $\mathbf{A}\mathbf{A}^\mathsf{T}$. It follows that $\overline{\mathcal{L}}_c$ and $\mathcal{L}_c$ deviate from each other by the set $\overline{\mathcal{L}}_c \setminus \mathcal{L}_c$. Consequently, in case B we simply add in $\overline{\mathcal{L}}_c \setminus \mathcal{L}_c$.

In case C, node c is a child of j in the new elimination tree, but not in the old tree. In this case we need to add in the entire set $\overline{\mathcal{L}}_c \setminus \{c\}$ since the corresponding term does not appear in (3.1). Similarly, in case D, node c is a child of j in the old elimination tree, but not in the new tree. In this case, the entire set $\mathcal{L}_c \setminus \{c\}$ should be deleted. The case where c is not a child of j in either the old or the new elimination

tree does not result in any adjustment since the corresponding $\mathcal{L}_c$ term is absent from both (3.1) and (4.1). ∎

An algorithm for updating a Cholesky factorization that is based only on this theorem would have to visit all nodes j from 1 to m, and consider all possible children $c < j$. On the other hand, not all nodes j from 1 to m need to be considered since not all columns of $\mathbf{L}$ change when $\mathbf{AA}^\mathsf{T}$ is modified. In [10, Thm. 4.1] we show that for $r = 1$, the nodes whose patterns can change are contained in $\overline{\mathcal{P}}(k_1)$, where we define $k_i = \min \mathcal{W}_i$. For a rank-r update, let $\mathcal{P}^{(i)}$ be the ancestor map associated with the elimination tree for the Cholesky factorization of the matrix

$$\mathbf{AA}^\mathsf{T} + \sum_{j=1}^{i} \mathbf{w}_j \mathbf{w}_j^\mathsf{T}. \tag{4.2}$$

Again, by [10, Thm. 4.1], the nodes whose patterns can change during the rank-r update are contained in the union of the patterns $\mathcal{P}^{(i)}(k_i)$, $1 \le i \le r$. Although we could evaluate $\mathcal{P}^{(i)}(k_i)$ for each i, it is difficult to do this efficiently since we need to perform a series of rank-1 updates and evaluate the ancestor map after each of these. On the other hand, by [10, Prop. 3.1] and [10, Prop. 3.2], $\mathcal{P}^{(i)}(j) \subseteq \mathcal{P}^{(i+1)}(j)$ for each i and j, from which it follows that $\mathcal{P}^{(i)}(k_i) \subseteq \overline{\mathcal{P}}(k_i)$ for each i. Consequently, the nodes whose patterns change during a rank-r update are contained in the set

$$\overline{\mathcal{T}} = \bigcup_{1 \le i \le r} \overline{\mathcal{P}}(k_i).$$

Theorem 4.2, below, shows that any node in $\overline{\mathcal{T}}$ is also contained in one or more of the sets $\mathcal{P}^{(i)}(k_i)$. From this it follows that the nodes in $\overline{\mathcal{T}}$ are precisely those nodes for which entries in the associated columns of $\mathbf{L}$ can change during a rank-r update. Before presenting the theorem, we illustrate this with a simple example shown in Figure 4.1. The left of Figure 4.1 shows the sparsity pattern of original matrix $\mathbf{AA}^\mathsf{T}$, its Cholesky factor $\mathbf{L}$, and the corresponding elimination tree. The nonzero pattern of the first column of $\mathbf{W}$ is $\mathcal{W}_1 = \{1, 2\}$. If performed as a single rank-1 update, this causes a modification of columns 1, 2, 6, and 8 of $\mathbf{L}$. The corresponding nodes in the original tree are encircled; these nodes form the path $\mathcal{P}^{(1)}(1) = \{1, 2, 6, 8\}$ from node 1 to the root (node 8) in the second tree. The middle of Figure 4.1 shows the matrix after this rank-1 update, and its factor and elimination tree. The entries in the second matrix $\mathbf{AA}^\mathsf{T} + \mathbf{w}_1\mathbf{w}_1^\mathsf{T}$ that differ from the original matrix $\mathbf{AA}^\mathsf{T}$ are shown as small pluses. The second column of $\mathbf{W}$ has the nonzero pattern $\mathcal{W}_2 = \{3, 4, 7\}$. As a rank-1 update, this affects columns $\mathcal{P}^{(2)}(3) = \overline{\mathcal{P}}(3) = \{3, 4, 5, 6, 7, 8\}$ of $\mathbf{L}$. These columns form a single path in the final elimination tree shown in the right of the figure.

For the first rank-1 update, the set of columns that actually change are $\mathcal{P}^{(1)}(1) = \{1, 2, 6, 8\}$. This is a subset of the path $\overline{\mathcal{P}}(1) = \{1, 2, 6, 7, 8\}$ in the final tree. If we use $\overline{\mathcal{P}}(1)$ to guide the work associated with column 1 of $\mathbf{W}$, we visit all the columns that need to be modified, plus column 7. Node 7 is in the set of nodes $\overline{\mathcal{P}}(3)$ affected by the second rank-1 update, however, as shown in the following theorem.

THEOREM 4.2. *Each of the paths $\mathcal{P}^{(i)}(k_i)$ is contained in $\overline{\mathcal{T}}$ and conversely, if $j \in \overline{\mathcal{T}}$, then j is contained in $\mathcal{P}^{(i)}(k_i)$ for some i.*

Proof. Before the theorem, we observe that each of the paths $\mathcal{P}^{(i)}(k_i)$ is contained in $\overline{\mathcal{T}}$. Now suppose that some node j lies in the tree $\overline{\mathcal{T}}$. We need to prove that it is contained in $\mathcal{P}^{(i)}(k_i)$ for some i. Let s be the largest integer such that $\overline{\mathcal{P}}(k_s)$ contains j, and let c be any child of j in $\overline{\mathcal{T}}$. If c lies on the path $\overline{\mathcal{P}}(k_i)$ for some i, then j lies

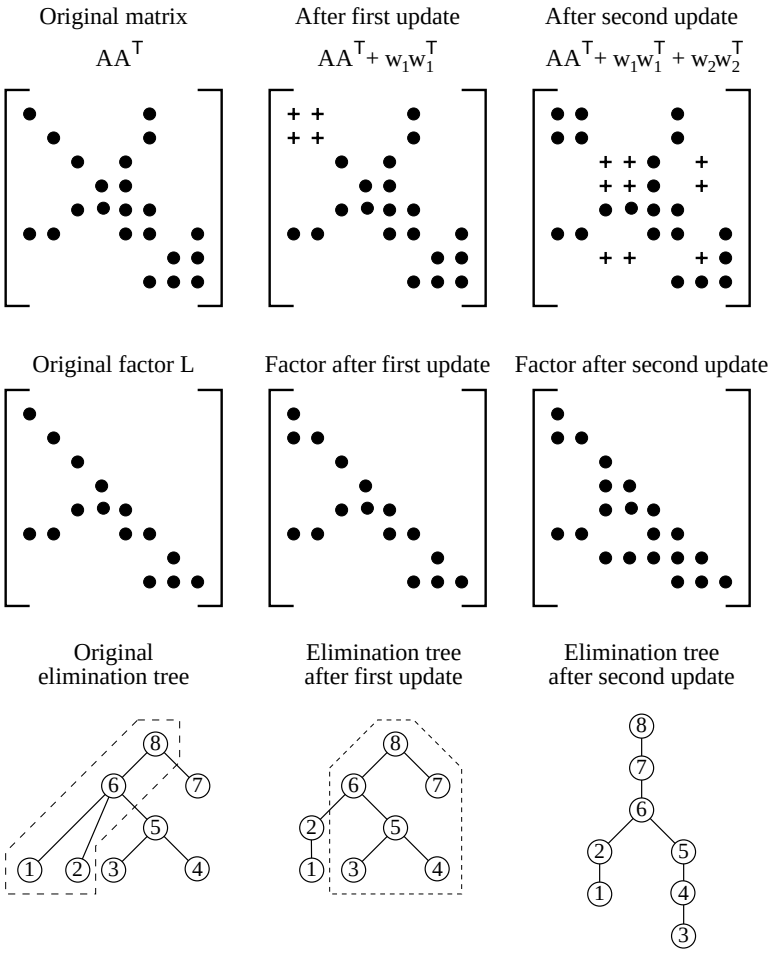


FIG. 4.1. *Example rank-2 update.*

on the path $\overline{\mathcal{P}}(k_i)$ since j is the parent of c. Since j does not lie on the path $\overline{\mathcal{P}}(k_i)$ for any $i > s$, it follows that c does not lie on the path $\overline{\mathcal{P}}(k_i)$ for any $i > s$. Applying this same argument recursively, we conclude that none of the nodes on the subtree of $\overline{\mathcal{T}}$ rooted at j lie on the path $\overline{\mathcal{P}}(k_i)$ for any $i > s$. Let $\overline{\mathcal{T}_j}$ denote the subtree of $\overline{\mathcal{T}}$ rooted at j. Since $\mathcal{P}^{(i)}(k_i)$ is contained in $\overline{\mathcal{P}}(k_i)$ for each i, none of the nodes of $\overline{\mathcal{T}_j}$ lie on any of the paths $\mathcal{P}^{(i)}(k_i)$ for $i > s$. By [10, Thm. 4.1], the patterns of all nodes outside the path $\mathcal{P}^{(i)}(k_i)$ are unchanged for each i. Let $\mathcal{L}_c^{(i)}$ be the pattern of column c in the Cholesky factorization of (4.2). Since any node c contained in $\overline{\mathcal{T}_j}$ does not lie on any of the paths $\mathcal{P}^{(i)}(k_i)$ for $i > s$, $\mathcal{L}_c^{(i)} = \mathcal{L}_c^{(l)}$ for all $i, l \geq s$. Since k_s is a node of $\overline{\mathcal{T}_j}$, the path $\mathcal{P}^{(s)}(k_s)$ must include j. □

Figure 4.2 depicts a subtree $\overline{\mathcal{T}}$ for an example rank-8 update. The subtree consists of all those nodes and edges in one or more of the paths $\overline{\mathcal{P}}(k_1), \overline{\mathcal{P}}(k_2), \ldots, \overline{\mathcal{P}}(k_8)$. These paths form a subtree, and not a general graph, since they are all paths from an initial node to the root of the elimination tree of the matrix $\overline{\mathbf{L}}$. The subtree $\overline{\mathcal{T}}$ might actually be a forest, if $\overline{\mathbf{L}}$ has an elimination forest rather than an elimination tree. The first nonzero positions in $\mathbf{w}_1$ through $\mathbf{w}_8$ correspond to nodes k_1 through k_8. For this

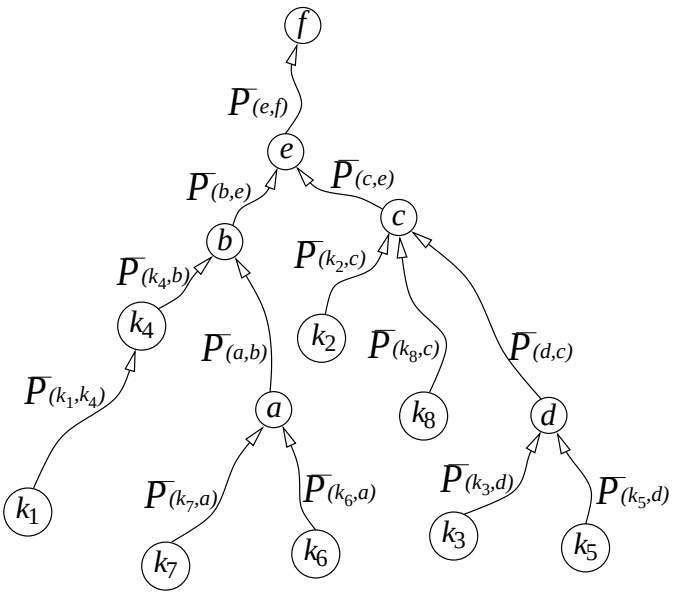


FIG. 4.2. *Example rank-8 symbolic update and subtree* $\overline{\mathcal{T}}$.

example, node k_4 happens to lie on the path $\mathcal{P}^{(1)}(k_1)$. Nodes at which paths first intersect are shown as smaller circles and are labeled a through f. Other nodes along the paths are not shown. Each curved arrow denotes a single subpath. For example, the arrow from nodes b to e denotes the subpath from b to e in $\overline{\mathcal{P}}(b)$. This subpath is denoted as $\overline{\mathcal{P}}(b,e)$ in Figure 4.2.

The following algorithm computes the rank-r symbolic update. It keeps track of an array of m "path-queues," one for each column of $\mathbf{L}$. Each queue contains a set of path-markers in the range 1 to r, which denote which of the paths $\overline{\mathcal{P}}(k_1)$ through $\overline{\mathcal{P}}(k_r)$ will modify column j next. If two paths have merged, only one of the paths needs to be considered. (We arbitrarily select the higher-numbered path to represent the merged paths.) This set of path-queues requires $O(m+r)$ space. Removing and inserting a path-marker in a path-queue takes $O(1)$ time. The only outputs of the algorithm are the new pattern of $\overline{\mathbf{L}}$ and its elimination tree, namely, $\overline{\mathcal{L}}_j^\sharp$ and $\overline{\pi}(j)$ for all $j \in [1,m]$. Not all columns are affected by the rank-r update. We define $\overline{\mathcal{L}}_j^\sharp = \mathcal{L}_j^\sharp$ and $\overline{\pi}(j) = \pi(j)$ for any node j not in $\overline{\mathcal{T}}$.

Case C will occur for c and j prior to visiting column $\pi(c)$, since $j = \overline{\pi}(c) < \pi(c)$. Thus we place c in the lost-child-queue of column $\pi(c)$ when encountering case C for nodes c and j. When the algorithm visits node $\pi(c)$, its lost-child-queue will contain all those nodes for which case D holds. This set of lost-child-queues is not the same as the set of path-queues (although there is exactly one lost-child-queue and one path-queue for each column j of $\mathbf{L}$).

ALGORITHM 3 (symbolic rank-r update; add new matrix $\mathbf{W}$).

Find the starting nodes of each path
for $i = 1$ **to** r **do**
    $\mathcal{W}_i = \{k : w_{ki} \neq 0\}$
    $k_i = \min\ \mathcal{W}_i$
    place path-marker i in path-queue of column k_i

end for
Consider all columns corresponding to nodes in the paths $\overline{\mathcal{P}}(k_1)$ *through* $\overline{\mathcal{P}}(k_r)$
for $j = \min_{i\in[1,r]} k_i$ **to** m **do**
if path-queue of column j is nonempty **do**
$\overline{\mathcal{L}}_j^\sharp = \mathcal{L}_j^\sharp$
for each path-marker i on path-queue of column j **do**
Path $\overline{\mathcal{P}}(k_i)$ *includes column* j
Let c be the prior column on this path (if any), where $\overline{\pi}(c) = j$
if $j = k_i$ **do**
Case A: j *is the first node on the path* $\overline{\mathcal{P}}(k_i)$*, no prior* c
$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + \mathcal{W}_i$
else if $j = \pi(c)$ **then**
Case B: c *is an old child of* j*, possibly changed*
$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \mathcal{L}_c)$
else
Case C: c *is a new child of* j *and a lost child of* $\pi(c)$
$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \{c\})$
place c in lost-child-queue of column $\pi(c)$
end if
end for
Case D: consider each lost child of j
for each c in lost-child-queue of column j **do**
$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \{c\})$
end for
Move up one step in the path(s)
$\overline{\pi}(j) = \min\ \overline{\mathcal{L}}_j \setminus \{j\}$
if $\overline{\mathcal{L}}_j \setminus \{j\} \neq \emptyset$ **then**
Let i be the largest path-marker in path-queue of column j
Place path-marker i in path-queue of column $\overline{\pi}(j)$
end if
end if *path-queue of column* j *nonempty*
end for

The optimal time for a general rank-r update is

$$O\left(\sum_{j\in\overline{\mathcal{T}}} |\overline{\mathcal{L}}_j|\right).$$

The actual time taken by Algorithm 3 is only slightly higher, namely,

$$O\left(m + \sum_{j\in\overline{\mathcal{T}}} |\overline{\mathcal{L}}_j|\right),$$

because of the $O(m)$ bookkeeping required for the path-queues. In most practical cases, the $O(m)$ term will not be the dominant term in the run time.

Algorithm 3 can be used to compute an entire symbolic factorization. We start by factorizing the identity matrix $\mathbf{I} = \mathbf{I}\mathbf{I}^\mathsf{T}$ into $\mathbf{L}\mathbf{D}\mathbf{L}^\mathsf{T} = \mathbf{I}\mathbf{I}\mathbf{I}$. In this case, we have $\mathcal{L}_j^\sharp = \{(j,1)\}$ for all j. The initial elimination tree is a forest of m nodes and no

edges. We can now determine the symbolic factorization of $\mathbf{I}+\mathbf{A}\mathbf{A}^{\mathsf{T}}$ using the rank-r symbolic update algorithm above, with $r = m$. This matrix has identical symbolic factors as $\mathbf{A}\mathbf{A}^{\mathsf{T}}$. Case A will apply for each column in $\mathbf{A}$, corresponding to the

$$\bigcup_{\min\ \mathcal{A}_k=j} \mathcal{A}_k$$

term in (3.1). Since $\pi(c) = 0$ for each c, cases B and D will not apply. At column j, case C will apply for all children in the elimination tree, corresponding to the

$$\bigcup_{\{c:j=\pi(c)\}} \mathcal{L}_c \setminus \{c\}$$

term in (3.1). Since duplicate paths are discarded when they merge, we modify each column j once, for each child c in the elimination tree. This is the same work performed by the symbolic factorization algorithm, Algorithm 2, which is $O(|\mathcal{L}|)$. Hence, Algorithm 3 is equivalent to Algorithm 2 when we apply it to the update $\mathbf{I}+\mathbf{A}\mathbf{A}^{\mathsf{T}}$. Its run time is optimal in this case.

5. Multiple-rank symbolic downdate. The downdate algorithm is analogous. The downdated matrix is $\mathbf{A}\mathbf{A}^{\mathsf{T}} - \mathbf{W}\mathbf{W}^{\mathsf{T}}$, where $\mathbf{W}$ is a subset of the columns of $\mathbf{A}$. In a downdate, $\overline{\mathcal{P}}(k) \subseteq \mathcal{P}(k)$, and thus rather than following the paths $\overline{\mathcal{P}}(k_i)$, we follow the paths $\mathcal{P}(k_i)$. Entries are dropped during a downdate, and thus $\overline{\mathcal{L}}_j \subseteq \mathcal{L}_j$ and $\pi(j) \le \overline{\pi}(j)$. We start with $\overline{\mathcal{L}}_j^\sharp = \mathcal{L}_j^\sharp$ and make the following changes.

- Case A: If $j = \min \mathcal{W}_i$ for some i, then the pattern $\mathcal{W}_i$ is removed from column j,

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - \mathcal{W}_i.$$

- Case B: If $j = \pi(c) = \overline{\pi}(c)$ for some node c, then c is a child of j in both the old and new tree. We need to remove from $\overline{\mathcal{L}}_j^\sharp$ entries in the old pattern $\mathcal{L}_c$ but not in the new pattern $\overline{\mathcal{L}}_c$,

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \overline{\mathcal{L}}_c).$$

- Case C: If $j = \pi(c) \ne \overline{\pi}(c)$ for some node c, then c is a child of j in the old elimination tree, but not the new tree. We compute

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \{c\}).$$

- Case D: If $j = \overline{\pi}(c) \ne \pi(c)$ for some node c, then c is a child of j in the new tree, but not the old one. We compute

$$\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \{c\}).$$

Case C will occur for c and j prior to visiting column $\overline{\pi}(c)$, since $j = \pi(c) < \overline{\pi}(c)$. Thus we place c in the new-child-queue of $\overline{\pi}(c)$ when encountering case C for nodes c and j. When the algorithm visits node $\overline{\pi}(c)$, its new-child-queue will contain all those nodes for which case D holds.

ALGORITHM 4 (symbolic rank-r downdate; remove matrix $\mathbf{W}$).

Find the starting nodes of each path
for $i = 1$ **to** r **do**
  $\mathcal{W}_i = \{k : w_{ki} \neq 0\}$
  $k_i = \min\ \mathcal{W}_i$
  place path-marker i in path-queue of column k_i
end for
Consider all columns corresponding to nodes in the paths $\mathcal{P}(k_1)$ through $\mathcal{P}(k_r)$
for $j = \min_{i\in[1,r]} k_i$ **to** m **do**
  if path-queue of column j is nonempty **do**
    $\overline{\mathcal{L}}_j^\sharp = \mathcal{L}_j^\sharp$
    for each path-marker i on path-queue of column j **do**
      Path $\mathcal{P}(k_i)$ includes column j
      Let c be the prior column on this path (if any), where $\pi(c) = j$
      if $j = k_i$ **do**
        Case A: j is the first node on the path $\mathcal{P}(k_i)$, no prior c
        $\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - \mathcal{W}_i$
      else if $j = \overline{\pi}(c)$ **then**
        Case B: c is an old child of j, possibly changed
        $\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \overline{\mathcal{L}}_c)$
      else
        Case C: c is a lost child of j and a new child of $\overline{\pi}(c)$
        $\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp - (\mathcal{L}_c \setminus \{c\})$
        place c in new-child-queue of column $\overline{\pi}(c)$
      end if
    end for
    Case D: consider each new child of j
    for each c in new-child-queue of j **do**
      $\overline{\mathcal{L}}_j^\sharp = \overline{\mathcal{L}}_j^\sharp + (\overline{\mathcal{L}}_c \setminus \{c\})$
    end for
    Move up one step in the path(s)
    $\overline{\pi}(j) = \min\ \overline{\mathcal{L}}_j \setminus \{j\}$
    if $\mathcal{L}_j \setminus \{j\} \neq \emptyset$ **then**
      Let i be the largest path-marker in path-queue of column j
      Place path-marker i in path-queue of column $\pi(j)$
    end if
  end if *path-queue of column j nonempty*
end for

The time taken by Algorithm 4 is

$$O\left(m + \sum_{j\in\mathcal{T}} |\mathcal{L}_j|\right),$$

which is slightly higher than the optimal time,

$$O\left(\sum_{j\in\mathcal{T}} |\mathcal{L}_j|\right).$$

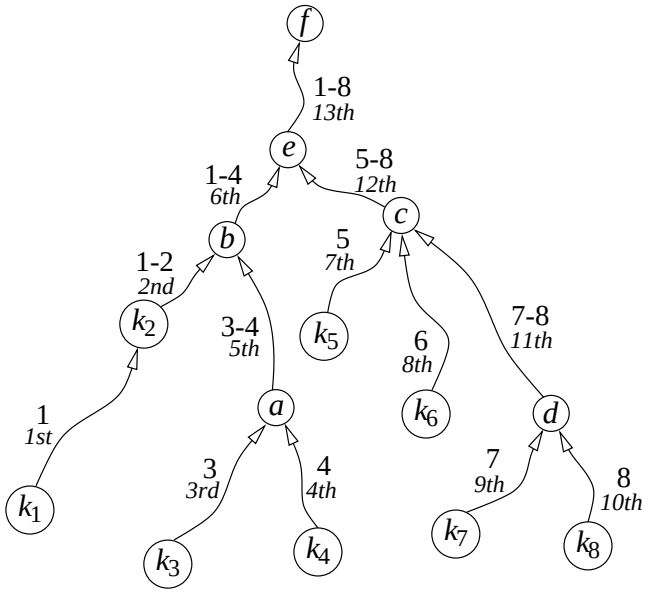


FIG. 6.1. *Example rank-8 update after depth-first-search reordering.*

In most practical cases, the $O(m)$ term in the asymptotic run time for Algorithm 4 will not be the dominant term.

6. Multiple-rank numerical update and downdate. The following numerical rank-r update/downdate algorithm, Algorithm 5, overwrites $\mathbf{L}$ and $\mathbf{D}$ with the updated or downdated factors. The algorithm is based on Algorithm 1, the one-pass version of Method C1 in [18] presented in section 1. The algorithm is used after the symbolic update algorithm (Algorithm 3) has found the subtree $\overline{\mathcal{T}}$ corresponding to the nodes whose patterns can change, or after the symbolic downdate algorithm (Algorithm 4) has found $\mathcal{T}$. Since the columns of the matrix $\mathbf{W}$ can be reordered without affecting the product $\mathbf{W}\mathbf{W}^{\mathsf{T}}$, we reorder the columns of $\mathbf{W}$ using a depth-first search [6] of $\overline{\mathcal{T}}$ (or $\mathcal{T}$) so that as we march through the tree, consecutive columns of $\mathbf{W}$ are utilized in the computations. This reordering improves the numerical update/downdate algorithm by placing all columns of $\mathbf{W}$ that affect any given subpath next to each other, eliminating an indexing operation. Reordering the columns of a sparse matrix prior to Cholesky factorization is very common [3, 22, 23, 25]. It improves data locality and simplifies the algorithm, just as it does for reordering $\mathbf{W}$ in a multiple-rank update/downdate. The depth-first ordering of the tree changes as the elimination tree changes, so columns of $\mathbf{W}$ must be ordered for each update or downdate.

To illustrate this reordering, consider the subtree $\overline{\mathcal{T}}$ in Figure 4.2 for a rank-8 update. If the depth-first-search algorithm visits child subtrees from left to right, the resulting reordering is as shown in Figure 6.1. Each subpath in Figure 6.1 is labeled with the range of columns of $\mathbf{W}$ that affect that subpath, and with the order in which the subpath is processed by Algorithm 5. Consider the path from node c to e. In Figure 4.2, the columns of $\mathbf{L}$ corresponding to nodes on this subpath are updated by columns 2, 8, 3, and 5 of $\mathbf{W}$, in that order. In the reordered subtree (Figure 6.1), the columns on this subpath are updated by columns 5 through 8 of the reordered $\mathbf{W}$.

ALGORITHM 5 (sparse numeric rank-r modification; add $\sigma \mathbf{W}\mathbf{W}^{\mathsf{T}}$).
The columns of $\mathbf{W}$ *have been reordered.*
for $i = 1$ **to** r **do**
  $\alpha_i = 1$
end for
for each subpath in depth-first-search order in $\overline{\mathcal{T}}$ ($\sigma = 1$) or $\mathcal{T}$ ($\sigma = -1$) **do**
  Let c_1 through c_2 be the columns of $\mathbf{W}$ that affect this subpath
  for each column j in the subpath **do**
    for $i = c_1$ **to** c_2 **do**
      $\overline{\alpha} = \alpha_i + \sigma w_{ji}^2 / d_j$
      $d_j = d_j \overline{\alpha}$
      $\gamma_i = w_{ji} / d_j$
      $d_j = d_j / \alpha_i$
      $\alpha_i = \overline{\alpha}$
    end for
    for all $p \in \overline{\mathcal{L}}_j \setminus \{j\}$ ($\sigma = 1$) or $p \in \mathcal{L}_j \setminus \{j\}$ ($\sigma = -1$) **do**
      for $i = c_1$ **to** c_2 **do**
        $w_{pi} = w_{pi} - w_{ji} l_{pj}$
        $l_{pj} = l_{pj} + \sigma \gamma_i w_{pi}$
      end for
    end for
  end for
end for

The time taken by r rank-1 updates [10] is

$$O\left(\sum_{i=1}^{r} \sum_{j \in \mathcal{P}^{(i)}(k_i)} |\mathcal{L}_j^{(i)}| \right), \tag{6.1}$$

where $\mathcal{L}_j^{(i)}$ is the pattern of column j after the ith rank-1 update. This time is asymptotically optimal. A single rank-r update cannot determine the paths $\mathcal{P}^{(i)}(k_i)$, but uses $\overline{\mathcal{P}}(k_i)$ instead. Thus, the time taken by Algorithm 5 for a rank-r update is

$$O\left(\sum_{i=1}^{r} \sum_{j \in \overline{\mathcal{P}}(k_i)} |\overline{\mathcal{L}}_j| \right).$$

This is slightly higher than (6.1), because $\mathcal{P}^{(i)}(k_i) \subseteq \overline{\mathcal{P}}(k_i)$ and $\mathcal{L}_j^{(i)} \subseteq \overline{\mathcal{L}}_j$. Since $\mathcal{P}^{(i)}(k_i) \subseteq \overline{\mathcal{P}}(k_i)$, the ith column of $\mathbf{W}$ does not necessarily affect all of the columns in the path $\overline{\mathcal{P}}(k_i)$. If $\mathbf{w}_i$ does not affect column j, then w_{ji} and γ_i will both be zero in the inner loop in Algorithm 5. An example of this occurs in Figure 4.1, where column 1 of $\mathbf{W}$ does not affect column 7 of $\mathbf{L}$. We could check this condition, and reduce the asymptotic run time to

$$O\left(\sum_{i=1}^{r} \sum_{j \in \mathcal{P}^{(i)}(k_i)} |\overline{\mathcal{L}}_j| \right).$$

In practice, however, we found that the paths $\mathcal{P}^{(i)}(k_i)$ and $\overline{\mathcal{P}}(k_i)$ did not differ much. Including this test did not improve the overall performance of our algorithm. The

time taken by Algorithm 5 for a rank-r downdate is similar, namely,

$$O\left(\sum_{i=1}^{r}\sum_{j\in\mathcal{P}(k_i)}|\mathcal{L}_j|\right).$$

The numerical algorithm for updating and downdating $\mathbf{LL}^\mathsf{T}$ is essentially the same as that for $\mathbf{LDL}^\mathsf{T}$ [4, 24]; the only difference is a diagonal scaling. For either $\mathbf{LL}^\mathsf{T}$ or $\mathbf{LDL}^\mathsf{T}$, the symbolic algorithms are identical.

7. Experimental results. To test our methods, we selected the same experiment as in our earlier paper on the single-rank update and downdate [10], which mimics the behavior of the LPDASA [20]. The first matrix is $10^{-6}\mathbf{I}+\mathbf{A}_0\mathbf{A}_0{}^\mathsf{T}$, where $\mathbf{A}_0$ consists of 5446 columns from a larger 6071-by-12,230 matrix $\mathbf{B}$ with 35,632 nonzeros arising in an airline scheduling problem (DFL001) [13]. The 5446 columns correspond to the optimal solution of the linear programming problem. Starting with an initial $\mathbf{LDL}^\mathsf{T}$ factorization of the matrix $10^{-6}\mathbf{I}+\mathbf{A}_0\mathbf{A}_0{}^\mathsf{T}$, we added columns from $\mathbf{B}$ (corresponding to an update) until we obtained the factors of $10^{-6}\mathbf{I}+\mathbf{BB}^\mathsf{T}$. We then removed columns in a first-in-first-out order (corresponding to a downdate) until we obtained the original factors. The LPDASA algorithm would not perform this much work (6784 updates and 6784 downdates) to solve this linear programming problem.

Our experiment took place on a Sun Ultra Enterprise running the Solaris 2.6 operating system, with eight 248 MHz UltraSparc-II processors (only one processor was used) and 2 GB of main memory. The dense matrix performance in millions of floating-point operations per second (Mflops) of the BLAS [12] is shown in Table 7.1. All results presented below are for our own codes (except for `colmmd`, `spooles`, and the BLAS) written in the C programming language and using double precision floating-point arithmetic.

TABLE 7.1
Dense matrix performance for 64*-by-*64 *matrices and* 64*-by-*1 *vectors.*

BLAS operation	Mflops
DGEMM (matrix-matrix multiply)	171.6
DGEMV (matrix-vector multiply)	130.0
DTRSV (solve $\mathbf{Lx}=\mathbf{b}$)	81.5
DAXPY (the vector computation $\mathbf{y}=\alpha\mathbf{x}+\mathbf{y}$)	78.5
DDOT (the dot product $\alpha=\mathbf{x}^\mathsf{T}\mathbf{y}$)	68.7

We first permuted the rows of $\mathbf{B}$ to preserve sparsity in the Cholesky factors of $\mathbf{BB}^\mathsf{T}$. This can be done efficiently with `colamd` [7, 8, 9, 21], which is based on an approximate minimum degree ordering algorithm [1]. However, to keep our results consistent with our prior rank-1 update/downdate paper [10], we used the same permutation as in those experiments (from `colmmd` [17]). Both `colamd` and MATLAB's `colmmd` compute the ordering without forming $\mathbf{BB}^\mathsf{T}$ explicitly. A symbolic factorization of $\mathbf{BB}^\mathsf{T}$ finds the nonzero counts of each column of the factors. This step takes an amount of space that is proportional to the number of nonzero entries in $\mathbf{B}$. It gives us the size of a static data structure to hold the factors during the updating and downdating process. The numerical factorization of $\mathbf{BB}^\mathsf{T}$ is not required. A second symbolic factorization finds the first nonzero pattern $\mathcal{L}$. An initial numerical factorization computes the first factors $\mathbf{L}$ and $\mathbf{D}$. We used our own nonsupernodal factorization code (similar to SPARSPAK [5, 15]), since the update/downdate algorithms do not use supernodes. A supernodal factorization code such as `spooles` [3] or

TABLE 7.2
Average update and downdate performance results.

rank r	rank-r time / r in seconds		Mflops	
	Update	Downdate	Update	Downdate
1	0.0840	0.0880	30.3	29.6
2	0.0656	0.0668	38.9	39.0
3	0.0589	0.0597	43.3	43.6
4	0.0513	0.0549	49.7	47.5
5	0.0500	0.0519	51.0	50.2
6	0.0469	0.0487	54.4	53.5
7	0.0451	0.0468	56.6	55.7
8	0.0434	0.0448	58.8	58.2
9	0.0431	0.0458	59.1	57.0
10	0.0426	0.0447	60.0	58.3
11	0.0415	0.0437	61.5	59.6
12	0.0413	0.0432	61.8	60.3
13	0.0403	0.0424	63.2	61.4
14	0.0402	0.0420	63.6	62.1
15	0.0395	0.0413	64.6	63.1
16	0.0392	0.0408	65.1	63.9

TABLE 7.3
Dense matrix performance for 64*-by-*64 *matrices and* 64*-by-*1 *vectors.*

Operation	Time (sec)	Mflops	Notes
`colamd` ordering	0.45	-	
Symbolic factorization (of $\mathbf{BB}^\mathsf{T}$)	0.07	-	1.49 million nonzeros
Symbolic factorization for first $\mathcal{L}$	0.46	-	831 thousand nonzeros
Numeric factorization for first $\mathbf{L}$ (our code)	20.07	24.0	
Numeric factorization for first $\mathbf{L}$ (`spooles`)	18.10	26.6	
Numeric factorization of $\mathbf{BB}^\mathsf{T}$ (our code)	61.04	18.5	not required
Numeric factorization of $\mathbf{BB}^\mathsf{T}$ (`spooles`)	17.80	63.3	not required
Average rank-16 update	0.63	65.1	compare with rank-1
Average rank-5 update	0.25	51.0	compare with solve step
Average rank-1 update	0.084	30.3	
Average solve $\mathbf{LDL}^\mathsf{T}\mathbf{x} = \mathbf{b}$	0.27	18.2	

a multifrontal method [2, 14] can get better performance. The factorization method used has no impact on the performance of the update and downdate algorithms.

We ran 16 different experiments, each one using a different rank-r update and downdate, where r varied from 1 to 16. After each rank-r update, we solved the sparse linear system $\mathbf{LDL}^\mathsf{T}\mathbf{x} = \mathbf{b}$ using a dense right-hand side $\mathbf{b}$. To compare the performance of a rank-1 update with a rank-r update ($r > 1$), we divided the run time of the rank-r update by r. This gives us a normalized time for a single rank-1 update. The average time and Mflops rate for a normalized rank-1 update and downdate for the entire experiment is shown in Table 7.2. The time for the update, downdate, or solve increases as the factors become denser, but the performance in terms of Mflops is fairly constant for all three operations. The first rank-16 update when the factor $\mathbf{L}$ is sparsest takes 0.47 seconds (0.0294 seconds normalized) and runs at 65.5 Mflops compared to 65.1 Mflops in Table 7.2 for the average speed of all the rank-16 updates.

The performance of each step is summarized in Table 7.3. A rank-5 update takes about the same time as using the updated factors to solve the sparse linear system $\mathbf{LDL}^\mathsf{T}\mathbf{x} = \mathbf{b}$, even though the rank-5 update performs 2.6 times the work.

The work, in terms of floating-point operations, varies only slightly as r changes.

With rank-1 updates, the total work for all the updates is 17.293 billion floating-point operations, or 2.55 million per rank-1 update. With rank-16 updates (the worst case), the total work increases to 17.318 billion floating-point operations. The rank-1 downdates take a total of 17.679 billion floating-point operations (2.61 million per rank-1 downdate), while the rank-16 downdates take a total of 17.691 billion operations. This confirms the near-optimal operation count of the multiple-rank update/downdate, as compared to the optimal rank-1 update/downdate.

Solving $\mathbf{Lx} = \mathbf{b}$ when $\mathbf{L}$ is sparse and $\mathbf{b}$ is dense, and computing the sparse $\mathbf{LDL}^\mathsf{T}$ factorization using a nonsupernodal method, both give a rather poor computation-to-memory-reference ratio of only 2/3. We tried the same loop unrolling technique used in our update/downdate code for our sparse solve and sparse $\mathbf{LDL}^\mathsf{T}$ factorization codes, but this resulted in no improvement in performance.

A sparse rank-r update or downdate can be implemented in a one-pass algorithm that has much better memory traffic than that of a series of r rank-1 modifications. In our numerical experimentation with the DFL001 linear programming test problem, the rank-r modification was more than twice as fast as r rank-1 modifications for $r \geq 11$. The superior performance of the multiple-rank algorithm can be explained using the computation-to-memory-reference ratio. If $c_1 = c_2$ in Algorithm 5 (a subpath affected by only one column of $\mathbf{W}$), it can be shown that this ratio is about 4/5 when $\overline{\mathcal{L}}_j$ is large. The ratio when $c_2 = c_1 + 15$ (a subpath affected by 16 columns of $\mathbf{W}$) is about 64/35 when $\overline{\mathcal{L}}_j$ is large. Hence, going from a rank-1 to a rank-16 update improves the computation-to-memory-reference ratio by a factor of about 2.3 when column j of $\mathbf{L}$ has many nonzeros. By comparison, the level-1 BLAS routines for dense matrix computations (vector computations such as DAXPY and DDOT) [11] have computation-to-memory-reference ratios between 2/3 and 1. The level-2 BLAS (DGEMV and DTRSV, for example) have a ratio of 2.

8. Summary. Because of improved memory locality, our multiple-rank sparse update/downdate method is over twice as fast as our prior rank-1 update/downdate method. The performance of our new method (65.1 Mflops for a sparse rank-16 update) compares favorably with both the dense matrix performance (81.5 Mflops to solve the dense system $\mathbf{Lx} = \mathbf{b}$) and the sparse matrix performance (18.0 Mflops to solve the sparse system $\mathbf{Lx} = \mathbf{b}$ and an observed peak numerical factorization of 63.3 Mflops in `spooles`) on the computer used in our experiments. Although not strictly optimal, the multiple-rank update/downdate method has nearly the same operation count as the rank-1 update/downdate method, which has an optimal operation count.

REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] P. R. Amestoy and I. S. Duff, *Vectorization of a multiprocessor multifrontal code*, Internat. J. Supercomputer Appl., 3 (1989), pp. 41–59.

[3] C. Ashcraft and R. G. Grimes, *SPOOLES: An object-oriented sparse matrix library*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX, 1999, CD-ROM, SIAM, Philadelphia, 1999.

[4] C. H. Bischof, C.-T. Pan, and P. T. P. Tang, *A Cholesky up- and downdating algorithm for systolic and SIMD architectures*, SIAM J. Sci. Comput., 14 (1993), pp. 670–676.

[5] E. Chu, A. George, J. W. H. Liu, and E. Ng, *SPARSPAK: Waterloo Sparse Matrix Package, User's Guide for SPARSPAK-A*, Technical report, Department of Computer Science, University of Waterloo, Waterloo, ON, Canada, 1984.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, and McGraw–Hill, New York, 1990.

[7] T. A. Davis, J. R. Gilbert, S. I. Larimore, E. Ng, and B. Peyton, *A column approximate minimum degree ordering algorithm*, in Proceedings of the Sixth SIAM Conference on Applied Linear Algebra, Snowbird, UT, 1997, p. 29.

[8] T. A. Davis, J. R. Gilbert, E. Ng, and B. Peyton, *A column approximate minimum degree ordering algorithm*, in Abstracts of the Second SIAM Conference on Sparse Matrices, Snowbird, UT, 1996.

[9] T. A. Davis, J. R. Gilbert, E. Ng, and B. Peyton, *A column approximate minimum degree ordering algorithm*, presented at the 13th Householder Symposium on Numerical Linear Algebra, Pontresina, Switzerland, 1996.

[10] T. A. Davis and W. W. Hager, *Modifying a sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 606–627.

[11] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[12] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, *A set of level-3 basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[13] J. J. Dongarra and E. Grosse, *Distribution of mathematical software via electronic mail*, Comm. ACM, 30 (1987), pp. 403–407.

[14] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[15] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[16] A. George, J. Liu, and E. Ng, *A data structure for sparse QR and LU factorizations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 100–121.

[17] J. R. Gilbert, C. Moler, and R. Schreiber, *Sparse matrices in MATLAB: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.

[18] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[19] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.

[20] W. W. Hager, *The LP dual active set algorithm*, in High Performance Algorithms and Software in Nonlinear Optimization, R. D. Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., Kluwer, Dordrecht, The Netherlands, 1998, pp. 243–254.

[21] S. I. Larimore, *An Approximate Minimum Degree Column Ordering Algorithm*, Technical Report TR-98-016, University of Florida, Gainesville, FL, 1998; also available online at http://www.cise.ufl.edu/tech-reports/.

[22] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[23] E. G. Ng and B. W. Peyton, *A supernodal Cholesky factorization algorithm for shared-memory multiprocessors*, SIAM J. Sci. Comput., 14 (1993), pp. 761–769.

[24] C.-T. Pan, *A modification to the LINPACK downdating algorithm*, BIT, 30 (1990), pp. 707–722.

[25] E. Rothberg, A. Gupta, E. G. Ng, and B. W. Peyton, *Parallel sparse Cholesky factorization algorithms for shared-memory multiprocessor systems*, in Advances in Computer Methods for Partial Differential Equations - VII, R. Vichnevetsky, D. Knight, and G. Richter, eds., IMACS, 1992, pp. 622–628.

[26] G. Strang, *Linear Algebra and Its Applications*, Academic Press, New York, 1980.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1014–1026

ON GREEN'S MATRICES OF TREES*

REINHARD NABBEN†

Abstract. The inverse $C = [c_{i,j}]$ of an irreducible nonsingular symmetric tridiagonal matrix is a so-called Green's matrix. A Green's matrix is a symmetric matrix which is given by two sequences of real numbers $\{u_i\}$ and $\{v_i\}$ such that $c_{i,j} = u_i v_j$ for $i \le j$. A similar result holds for nonsymmetric matrices. An open problem on nonsingular sparse matrices is whether there exists a similar structure for their inverses as in the tridiagonal case. Here we positively answer this question for irreducible acyclic matrices, i.e., matrices whose undirected graphs are trees. We prove that the inverses of irreducible acyclic symmetric matrices are given as the Hadamard product of three matrices, a type D matrix, a flipped type D matrix, and a matrix of tree structure which is closely related to the graph of the original matrix itself. For nonsymmetric matrices we obtain a similar structure. Moreover, our results include the result for symmetric and nonsymmetric tridiagonal matrices.

Key words. acyclic matrices, trees, Green's matrices, tridiagonal matrices

AMS subject classifications. 15A48, 15A57, 65F10

PII. S0895479899365732

1. Introduction.

In many mathematical problems nonsingular sparse matrices arise. An important class of sparse matrices is the class of tridiagonal matrices. For tridiagonal matrices many theoretical results are known. One of the most important results is established by Gantmacher and Krein in [4] and [5]. They proved that a symmetric, irreducible nonsingular matrix A is tridiagonal if and only if $A^{-1} =: C = [c_{i,j}]$ is given by two sequences $\{u_i\}_{i=1}^n$, $\{v_i\}_{i=1}^n$ of numbers such that

$$C = \begin{bmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_n \\ u_1v_2 & u_2v_2 & \cdots & u_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_1v_n & u_2v_n & \cdots & u_nv_n \end{bmatrix}, \quad \text{i.e.,} \quad c_{i,j} = \begin{cases} u_iv_j & i \le j, \\ u_jv_i & i \ge j. \end{cases} \tag{1.1}$$

Matrices of the form (1.1) are called Green's matrices by Karlin [8]. Gantmacher and Krein [5] called them matrices á la couple in [4]. It was observed in [14] that Green's matrices can be described more elegantly as the Hadamard product (elementwise product) of a so-called type D matrix [11] and a flipped type D matrix:

$$C = \begin{bmatrix} u_1 & u_1 & \cdots & u_1 \\ u_1 & u_2 & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_1 & u_2 & \cdots & u_n \end{bmatrix} \circ \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_2 & v_2 & \cdots & v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n & v_n & \cdots & v_n \end{bmatrix}. \tag{1.2}$$

A similar result holds for the inverse of a nonsymmetric irreducible tridiagonal matrix. There the inverse $C = A^{-1}$ can be described by four sequences $\{u_i\}, \{v_i\}, \{x_i\}, \{y_i\}$ which satisfy $u_iv_i = x_iy_i$ [6]. In detail we have

$$c_{i,j} = \begin{cases} u_iv_j, & i \le j, \\ x_iy_j, & i \ge j. \end{cases} \tag{1.3}$$

*Received by the editors January 19, 2000; accepted for publication (in revised form) by I. Ipsen July 28, 2000; published electronically January 31, 2001.
http://www.siam.org/journals/simax/22-4/36573.html
†Fakultät für Mathematik, Universität Bielefeld, Postfach 10 01 31, 33 501 Bielefeld, Germany (nabben@mathematik.uni-bielefeld.de).

As in the symmetric case matrices of the form (1.3) can be written nicely as the Hadamard product of two matrices:

$$C = \begin{bmatrix} u_1 & u_1 & \cdots & \cdots & u_1 \\ x_1 & u_2 & \cdots & \cdots & u_2 \\ x_1 & x_2 & u_3 & \cdots & u_3 \\ \vdots & \vdots & & \ddots & \vdots \\ x_1 & x_2 & \cdots & \cdots & u_n \end{bmatrix} \circ \begin{bmatrix} v_1 & v_2 & \cdots & \cdots & v_n \\ y_2 & v_2 & \cdots & \cdots & v_n \\ y_3 & y_3 & v_3 & \cdots & v_n \\ \vdots & \vdots & & \ddots & \vdots \\ y_n & y_n & \cdots & \cdots & v_n \end{bmatrix}. \tag{1.4}$$

A frequently asked question about nonsingular sparse matrices is whether there exists a similar structure for their inverses as in the tridiagonal case. Or in other words, what are generalizations of Green's matrices for arbitrary sparse matrices? Probably the answer of the first question in general is negative. However, here we establish the structure of the inverses of irreducible acyclic matrices, i.e., irreducible matrices whose graphs are trees. It turns out that the Hadamard product form (1.2) and (1.4) is the most promising approach for generalizations. We prove that the inverses of irreducible acyclic symmetric matrices are given as the Hadamard product of three matrices, a type D matrix, a flipped type D matrix (as in (1.2)), and a matrix of tree structure which is closely related to the graph of A itself. A similar result holds for nonsymmetric matrices (see Theorem 3.4 and Corollary 3.5). Moreover, our result includes the result by Gantmacher and Krein and also the result for nonsymmetric matrices. We also extend some results by Kirkland, Neumann, and Shader [10] as well as by Fiedler [3]. There, inverses of $(n-1)\times(n-1)$ principal submatrices of combinatorically symmetric [10] or symmetric [3] singular M-matrices whose graphs are trees are considered.

2. Notations and preliminary results. We start this section with some notations and definitions taken from [7]. A *weighted graph* $G = (V, E)$ of $n+1$ vertices is a graph with vertex set $V = \{0, \ldots, n\}$, edges $e_{ij} \in E$ between the vertices $(i, j \in \{0, 1, \ldots, n\})$ labeled by nonzero weights $w_{i,j} \in \mathbb{R}$. Note that we allow negative weights. Here we consider only undirected graphs.

A *path* from vertex i to vertex j, denoted by $P_{i,j}$, is the set of edges $\{(k_1, k_2), (k_2, k_3), \ldots, (k_{r-1}, k_r)\}$, where $k_1 = i$ and $k_r = j$. The path $P_{i,j}$ is a cycle if $k_1 = i = j = k_r$, $r \geq 3$, and $k_1, \ldots, k_{r-1}$ are distinct.

A graph is acyclic if it has no cycles. A *tree* is a connected acyclic graph. Equivalently, a tree is a graph for which there exists a unique path between any two vertices i and j. A *rooted tree* is a tree with a prominent vertex called the root.

Here we distinguish between trees Γ and rooted trees $\Gamma_{(0)}$. We always assume that trees have vertex set $\{1, \ldots, n\}$ while for rooted trees $\{0, 1, \ldots, n\}$ is the vertex set and the root is labeled by 0. For a given tree Γ we construct a (special) rooted tree $\Gamma_{(0)}$ by adding a new vertex 0, the root, and a new edge $e_{0,1}$ from the root to vertex 1 to the old tree.

The numbering of the vertices of the tree has no impact on our results. However, for convenience, we number the vertices of the trees in the following way. We start with an arbitrary vertex which is then labeled with 1. We then proceed recursively by a depth-first search (dfs) or numbering; see [1]. In other words we label the vertices recursively branch by branch.

A *branch* starting at vertex i of the tree Γ is the connected subgraph of Γ including vertex i obtained by deleting the unique edge $e_{j,i}$ with $j < i$.

The *graph* $G(A)$ of an $n \times n$ matrix $A = [a_{i,j}]$ is the undirected graph consisting of n vertices $\{1, \ldots, n\}$ such that there is an edge between vertex i and vertex j if and

only if $a_{i,j} \neq 0$ or $a_{j,i} \neq 0$. A is called *treediagonal* by Klein [9] if $G(A)$ is a forest, i.e., a collection of trees. Here we prefer the name *acyclic matrices* for matrices A whose graphs $G(A)$ are forests.

In one of our main theorems (Theorem 3.4) we consider irreducible acyclic matrices. Irreducible acyclic matrices are matrices whose graphs are trees. However, matrices whose graphs are trees are in general not irreducible and acyclic. For example consider the matrix

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

In Lemma 3.2 and Theorem 3.3 the more general assumption that the graph of a matrix is a tree is used.

We denote by e the vector with all ones and $e_1 = (1, 0, \ldots, 0)^T$.

Klein established in [9] some results on matrices whose graphs are trees. We will use these results in the next section.

DEFINITION 2.1. *An $n \times n$ matrix $C = [c_{i,j}]$ satisfies the treeangle property with respect to a given tree Γ if for every $i, j, k \in V$ with $P_{i,k} \subseteq P_{i,j}$*

$$c_{i,j}c_{k,k} = c_{i,k}c_{k,j}. \tag{2.1}$$

Klein then proved the following theorems.

THEOREM 2.2. *Let Γ be a tree. If a nonsingular matrix C satisfies the treeangle property with respect to Γ and if $c_{i,i} \neq 0$ for all interior vertices, then C^{-1} is treediagonal with respect to Γ, i.e., C^{-1} is acyclic.*

THEOREM 2.3. *Let Γ be a tree. If A is a nonsingular treediagonal matrix with respect to Γ, then A^{-1} satisfies the treeangle property with respect to Γ.*

The above theorems describe the structure of inverses of acyclic matrices. However, it is not clear at all how one can describe the treeangle property in terms of matrices as in the tridiagonal case. Moreover, what are the generalizations of Green's matrices? To do so we need the following class of matrices.

DEFINITION 2.4. *Let $\Gamma_{(0)}$ be a weighted rooted tree with vertex set $\{0, 1, \ldots, n\}$, where 0 denotes the root. A matrix $A = [a_{i,j}] \in \mathbb{R}^{n,n}$ is of tree structure with respect to $\Gamma_{(0)}$ if for all $i, j = 1, \ldots, n$*

$$a(i,j) := \sum_{\{r,s\} \in P_{i,0} \cap P_{j,0}} w_{r,s}, \tag{2.2}$$

where $\{r, s\} \in P_{i,0} \cap P_{j,0}$ denotes a common edge of the paths $P_{i,0}$ and $P_{j,0}$.

Note that (2.2) defines a "distance" or better an "inverse distance" between the vertices of the tree. This distance was already used by Nabben and Varga in [16] for leaves of trees.

We immediately obtain the following characterization of matrices of tree structure.

THEOREM 2.5. *Let $\Gamma_{(0)}$ be a weighted rooted tree. Then the following are equivalent:*

(1) *$A = [a_{i,j}] \in \mathbb{R}^{n,n}$ is of tree structure with respect to $\Gamma_{(0)}$.*

(2) *A can be decomposed as*

$$A = \sum_{i=1}^{n} \tau_i u_i u_i^T, \tag{2.3}$$

where $\tau_i \in \mathbb{R}$ and $\tau_i = w_{j,i}$; here j is the unique vertex which is connected with i and $j < i$. The vectors $u_i \in \mathbb{R}^{n,n}$ satisfy $(u_i)_j = 1$ for all j (including i) belonging to the branch of $\Gamma_{(0)}$ starting at vertex i, and $(u_i)_j = 0$ otherwise.

Proof. First assume that A is of tree structure. Consider the branches starting at the root 0. If there are $s, s > 1$, branches and i and j are vertices of different branches then $a(i,j) = 0$ and with our numbering of the vertices of the tree A is a block diagonal matrix

$$A = \begin{bmatrix} C_1 & & 0 \\ & \ddots & \\ 0 & & C_s \end{bmatrix},$$

where $C_1, \ldots, C_s$ are square matrices. We then proceed by induction. We can write $C_t = \tilde{C}_t - \tau_{i_t} ee^T$, $t \in \{1, \ldots, s\}$, where $\tau_{i_t} = \omega_{0,i_t}$. Here ω_{0,i_t} is the weight of edge from the root 0 to its neighbor $i_t \in \{1, \ldots, n\}$. The matrices are again of tree structure. The trees are the separate branches starting at the vertices i_t which become the new roots.

If there is just one branch, then $a(1,j) = w_{0,1}$ for all $j = 1, \ldots, n$. Hence

$$A = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} + w_{0,1} ee^T.$$

Again we can proceed by induction. The matrix C is of tree structure with respect to the branch starting at the neighbor of the root.

The other implication can be proved by induction also. If i and j are in different branches connected with the root, then

$$a(i,j) = 0 = \sum_{\{r,s\} \in P_{i,0} \cap P_{j,0}} w_{r,s}.$$

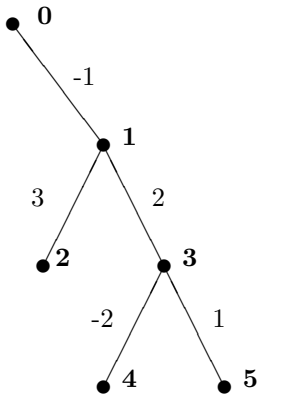
If there is just one neighbor of the root, vertex 1, then for all j

$$a(1,j) = a(j,1) = \tau_1 = w_{0,1} = \sum_{\{r,s\} \in P_{1,0} \cap P_{j,0}} w_{r,s}. \qquad \square$$

Note that matrices of tree structure can have negative entries. Thus the τ_i of (2.3) can be negative. But if all τ_i in (2.3) are nonnegative, we obtain a subclass of certain so-called pre-ultrametric matrices (see [3] and [17]). If all τ_i are positive, (2.3) gives certain so-called strictly ultrametric matrices defined by [12] and characterized in [15]. Pre-ultrametric matrices as well as strictly ultrametric matrices are nonnegative inverse M-matrices which can be decomposed as the sum of $2n-1$ rank one matrices similar to (2.3). For more details on ultrametric matrices, see [15], [17], [16].

Example 2.1. For illustration consider the weighted rooted tree in Figure 1. The 5×5 matrix of tree structure is then given by

$$A = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 2 \end{bmatrix}.$$



$u_1 := (1,1,1,1,1)^T$

$u_2 := (0,1,0,0,0)^T$; $u_3 := (0,0,1,1,1)^T$

$u_4 := (0,0,0,1,0)^T$; $u_5 := (0,0,0,0,1)^T$

FIG. 1.

Moreover, A can be decomposed as

$$A = \sum_{i=1}^{5} \tau_i \mathbf{u}_i \mathbf{u}_i^T,$$

where the u_i are given as in Figure 1 and the $\{\tau_1, \ldots, \tau_5\}$ in (2.3) are $\{-1, 3, 2, -2, 1\}$.

3. Main results. We start this section with a simple observation.

PROPOSITION 3.1. *Let $A = [a_{i,j}] \in \mathbb{R}^{n,n}$ be nonsingular, irreducible, and acyclic. Assume that the diagonal entries of $C = [c_{i,j}] := A^{-1}$ are nonzero. Then $c_{i,j} \neq 0$ for all $i, j \in \{1, \ldots, n\}$.*

Proof. Obviously, A is treediagonal. With Theorem 2.3, C satisfies the treeangle property. Now let $P_{i,j} = \{(k_1, k_2), (k_2, k_3), \ldots, (k_{r-1}, k_r)\}$, where $k_1 = i$ and $k_r = j$, be the unique path from i to j. As mentioned in [9] the treeangle property implies

$$c_{i,j} = c_{k_1,k_2} \prod_{s=2}^{r-1} \frac{c_{k_s,k_{s+1}}}{c_{k_s,k_s}}. \tag{3.1}$$

Thus, if there is one $c_{i,i+1} = 0$, then C is reducible. Hence all $c_{i,i+1}$ are nonzero which implies that all other entries are nonzero also. ∎

Note that Proposition 3.1 is not true for matrices whose graph is a tree.

LEMMA 3.2. *Let Γ be a tree and let $C = [c_{i,j}] \in \mathbb{R}^{n,n}$. Then the following are equivalent:*

1. *C satisfies the treeangle property with respect to Γ with $Ce_1 = \alpha e$ and $e_1^T C = \alpha e^T$ for some $\alpha \in \mathbb{R}$.*
2. *C is symmetric and C is of tree structure with respect to the weighted tree $\Gamma_{(0)}$, where $w_{0,1} = \alpha$.*

Proof. (1) ⇒ (2): First, observe that the first row and column of C are the same. We then proceed by induction on n. For $n = 2$ the statement is true.

Now delete the first row and column of C and let $C_{[1]}$ be the remaining matrix. Obviously, $C_{[1]}$ satisfies the treeangle property with respect to $\Gamma_{[1]}$, i.e., the graph obtained from Γ by deleting vertex 1 and all edges connecting 1 with other vertices. First assume that $\Gamma_{[1]}$ is connected. This implies that there was just one edge from

vertex 1 to vertex 2. Since the first row and column of C are the same we can write C as

$$C = \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix} + \alpha e e^T$$

for some $B \in \mathbb{R}^{n-1,n-1}$ and $e = (1, \ldots, 1)^T \in \mathbb{R}^{n,n}$. The treeangle property for C says

$$\begin{aligned} c_{1,j}c_{2,2} &= c_{1,2}c_{2,j} \quad \text{for} \quad j > 2, \\ c_{i,1}c_{2,2} &= c_{i,2}c_{2,1} \quad \text{for} \quad i > 2. \end{aligned}$$

Thus

$$c_{2,2} = c_{2,j} = c_{i,2} \neq 0 \quad \text{for} \quad j > 2,\ i > 2.$$

Hence, $C_{[1]}e_1 = \tilde{c}e$ and $e_1^T C_{[1]} = \tilde{c}e^T$. Therefore $Be_1 = (\tilde{c} - \alpha)e$ and $e_1^T B = (\tilde{c} - \alpha)e^T$. With the induction hypothesis and Theorem 2.5 we obtain the result.

Now assume that $C_{[1]}$ is reducible, i.e., there is more than one vertex connected with 1. Thus we have different branches connected with vertex 1. Now let i and j be two vertices of different branches. The treeangle property implies

$$c_{i,j}c_{1,1} = c_{i,1}c_{1,j}.$$

Thus

$$c_{i,j} = \alpha.$$

Hence C is of the form

$$C = \begin{bmatrix} C_1 & & 0 \\ & \ddots & \\ 0 & & C_s \end{bmatrix} + \alpha e e^T,$$

for some square matrices $C_1, \ldots, C_s$. Now consider the submatrices of C corresponding to the different branches. Obviously they satisfy the treeangle property with respect to the subgraphs. The first submatrix, say A_{11}, which includes vertex 1 obviously satisfies that the first row and column are the same. Now consider a submatrix consisting of vertices $k, k+1, \ldots, t$, where k is connected with 1. Then for $j \in \{k+1, \ldots, t\}$

$$\begin{aligned} c_{1,j}c_{k,k} &= c_{1,k}c_{k,j}, \\ c_{j,1}c_{k,k} &= c_{j,k}c_{k,1}. \end{aligned}$$

Hence

$$c_{k,k} = c_{k,j} = c_{j,k}.$$

Thus the submatrices satisfy the inductive assumptions. Again, with the induction hypothesis and Theorem 2.5 we get (2).

(2) $\Rightarrow$ (1): Since C is of tree structure, it is clear that $Ce_1 = \alpha e$ and $e_1^T C = \alpha e^T$. We then prove that C satisfies the treeangle property with respect to Γ. Let $i, j, k \in \{1, \ldots, n\}$ with $P_{i,k} \subseteq P_{i,j}$. If $P_{0,k} \subseteq P_{0,i}$, then $c_{i,k} = c_{k,k}$ and $c_{i,j} = c_{k,j}$. If

$P_{0,k} \subseteq P_{0,j}$, then $c_{j,k} = c_{k,j} = c_{k,k}$ and $c_{i,j} = c_{k,i} = c_{i,k}$. In both cases the treeangle property is fulfilled. ∎

The next theorem gives a characterization of acyclic matrices which satisfies $Ae = \gamma e_1$ and $e^T A = \gamma e_1^T$. Moreover, we give the exact formula for the inverse.

THEOREM 3.3. *Let Γ be a tree and let $A = [a_{i,j}] \in \mathbb{R}^{n,n}$ be nonsingular. Then the following are equivalent:*

(1) *$G(A) = \Gamma$ and $Ae = \gamma e_1$ and $e^T A = \gamma e_1^T$.*
(2) *A^{-1} is symmetric and A^{-1} is of tree structure with respect to the weighted rooted tree $\Gamma_{(0)}$, where the weights are $w_{i,j} = -1/a_{i,j}$ and $w_{0,1} = 1/\gamma$.*

Proof. (1) $\Rightarrow$ (2): Obviously the first row and column of A^{-1} are $\gamma^{-1}e$ and $\gamma^{-1}e^T$. Therefore we have

$$A^{-1} = \begin{bmatrix} \gamma^{-1} & \dots & \gamma^{-1} \\ \vdots & B & \\ \gamma^{-1} & & \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \tilde{B} & \\ 0 & & \end{bmatrix} + \frac{1}{\gamma} e e^T$$

for some $B, \tilde{B} \in \mathbb{R}^{n-1,n-1}$. We partition A similarly:

$$A = \begin{bmatrix} a_{11} & b \\ c^T & A_{22} \end{bmatrix}. \tag{3.2}$$

Since A is nonsingular we obtain that the Schur complement $A^{-1}/\gamma^{-1} = B - \gamma^{-1} e_{n-1} e_{n-1}^T$ is also nonsingular. Using the formula for the inverse of 2×2 block matrices we obtain

$$A_{22} = (B - \gamma^{-1} e_{n-1} e_{n-1}^T)^{-1} = \tilde{B}^{-1},$$

where $e_{n-1} = [1, \dots, 1]^T \in \mathbb{R}^{n-1}$.

Now first let vertex 1 be only connected with vertex 2. Then since $Ae = \gamma e_1$ and A is acyclic we have

$$A_{22} e = -a_{12} e_1 \quad \text{or} \quad -\frac{1}{a_{12}} e = A_{22}^{-1} e_1,$$
$$e^T A_{22} = -a_{21} e_1^T \quad \text{or} \quad -\frac{1}{a_{21}} e^T = e_1^T A_{22}^{-1}.$$

Hence, $a_{21} = a_{12} \neq 0$ and

$$A^{-1} = \frac{1}{\gamma} e e^T - \frac{1}{a_{12}} (0, e_{n-1}^T)^T (0, e_{n-1}^T) + \begin{bmatrix} O & O \\ O & \hat{B} \end{bmatrix}$$

for some $\hat{B} \in \mathbb{R}^{n-2 \times n-2}$.

If there is more than one branch connected with vertex 1, we can apply the above proof for each branch. We then proceed by induction.

(2) $\Rightarrow$ (1): Since A^{-1} is of tree structure we have as above

$$A^{-1} = \begin{bmatrix} \gamma^{-1} & \dots & \gamma^{-1} \\ \vdots & B & \\ \gamma^{-1} & & \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \tilde{B} & \\ 0 & & \end{bmatrix} + \frac{1}{\gamma} e e^T$$

for some $B, \tilde{B} \in \mathbb{R}^{n-1,n-1}$. Hence $Ae = \gamma e_1$ and $e^T A = \gamma e_1^T$. We partition A as in (3.2) and get $A_{22} = \tilde{B}^{-1}$. Now first let vertex 1 be only connected with vertex 2.

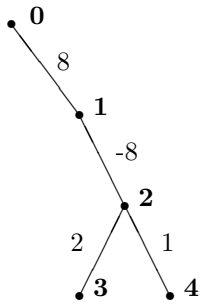


FIG. 2.

Then $\tilde{B}e_1 = w_{12}e_{n-1}$ and $e_1^T\tilde{B} = w_{12}e_{n-1}^T$ and $\tilde{B}$ is of tree structure with respect to the branch starting at vertex 2. Moreover, since

$$Ae = \gamma e_1 \quad \text{and} \quad A_{22}e_{n-1} = \tilde{B}^{-1}e_{n-1} = w_{12}^{-1}e_1,$$

we obtain

$$a_{21} = -\frac{1}{w_{12}} \quad \text{and} \quad a_{i,1} = 0 \text{ for } i > 2.$$

If there is more than one branch connected with vertex 1, we can apply the above steps for each branch connected with vertex 1. By induction we thus obtain the result. □

Note that in Theorem 3.3 there is no restriction on the signs of the entries of the matrix. Moreover, Theorem 3.3 also gives formulas for the entries of the inverse of a matrix of tree structure. The diagonal entries of the inverse can be obtained using $Ae = \gamma e_1$.

Theorem 3.3 extends some results by Kirkland, Neumann, and Shader [10] as well as by Fiedler [3]. There, inverses of $(n-1)\times(n-1)$ principal submatrices of combinatorically symmetric [10] or symmetric [3] singular M-matrices whose graphs are trees are considered.

Next we compare Theorem 3.3 with Klein's Theorems 2.2 and 2.3. First, of course, here we give formulas for the inverse in both ways. But on the other hand, the assumption for A^{-1} being of tree structure which implies $A^{-1}e_1 = w_{01}e$ seems to be more restrictive then the assumption of nonvanishing diagonal entries. But the following example shows that Theorem 3.3 is not included in Theorem 2.2.

Example 3.1. Let A^{-1} be as follows:

$$A^{-1} = \begin{bmatrix} 8 & 8 & 8 & 8 \\ 8 & 0 & 0 & 0 \\ 8 & 0 & 2 & 0 \\ 8 & 0 & 0 & 1 \end{bmatrix}.$$

A^{-1} is of tree structure with respect to the tree given in Figure 2.

The inverse of A^{-1} is

$$A = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 11 & -4 & -8 \\ 0 & -4 & 4 & 0 \\ 0 & -8 & 0 & 8 \end{bmatrix},$$

and $G(A)$ is the tree given in Figure 2.

We then obtain our generalization of the Gantmacher and Krein result.

THEOREM 3.4. *Let $A \in \mathbb{R}^{n,n}$ be nonsingular, irreducible, and acyclic and let $\Gamma = G(A)$. Assume that the diagonal entries of A^{-1} are nonzero. Then there exist matrices T and R of the form*

$$T = \begin{bmatrix} d_1 & d_1 & \cdots & \cdots & d_1 \\ f_1 & d_2 & \cdots & \cdots & d_2 \\ f_1 & f_2 & d_3 & \cdots & d_3 \\ \vdots & \vdots & & \ddots & \vdots \\ f_1 & f_2 & \cdots & \cdots & d_n \end{bmatrix}, \qquad R = \begin{bmatrix} f_1 & f_2 & \cdots & \cdots & f_n \\ d_2 & f_2 & \cdots & \cdots & f_n \\ d_3 & d_3 & f_3 & \cdots & f_n \\ \vdots & \vdots & & \ddots & \vdots \\ d_n & d_n & \cdots & \cdots & f_n \end{bmatrix} \tag{3.3}$$

and a matrix U of tree structure with respect to the weighted tree $\Gamma_{(0)}$ such that

$$A^{-1} = T \circ U \circ R.$$

Conversely, let $U = [u_{i,j}] \in \mathbb{R}^{n,n}$ be nonsingular. If U is of tree structure with respect to a given rooted tree $\Gamma_{(0)}$, then for any matrices T and R of the form (3.3) *with $f_i d_i \neq 0$ for all i, the matrix $(T \circ U \circ R)^{-1}$ is irreducible, acyclic, and $G((T \circ U \circ R)^{-1}) = \Gamma$.*

Note again that here $\Gamma_{(0)}$ is obtained from Γ by adding the root 0 and a new edge $e_{0,1}$.

Proof. Let $F = \operatorname{diag}(f_1, \ldots, f_n) = \operatorname{diag}(e_1^T A^{-1})$ and $D = \operatorname{diag}(d_1, \ldots, d_n) = \operatorname{diag}(A^{-1} e_1)$. Since A is irreducible, we can apply Proposition 3.1. Thus D and F are nonsingular. Hence FAD is nonsingular and $G(FAD) = \Gamma$. Moreover, $(FAD)^{-1} e_1 = ce$ and $e_1^T (FAD)^{-1} = ce^T$ for some $c \in \mathbb{R}$. Thus with Theorem 3.3

$$(FAD)^{-1} = U$$

for a matrix U of tree structure with respect to $\Gamma_{(0)}$. Hence

$$A^{-1} = DUF.$$

But we then observe that

$$A^{-1} = DUF = T \circ U \circ R.$$

For the converse we obtain with Theorem 3.3 that U^{-1} is acyclic with $G(U^{-1}) = \Gamma$. Since $T \circ U \circ R = DUF$ with $D = \operatorname{diag}(d_1, \ldots, d_n)$ and $F = \operatorname{diag}(f_1, \ldots, f_n)$, we obtain the desired result. ∎

For the symmetric case we obtain the following.

COROLLARY 3.5. *Let $A \in \mathbb{R}^{n,n}$ be symmetric nonsingular, acyclic, and irreducible with $G(A) = \Gamma$. Assume that the diagonal entries of A^{-1} are nonzero. Then there exist matrices T and R of the form*

$$T = \begin{bmatrix} d_1 & d_1 & \cdots & \cdots & d_1 \\ d_1 & d_2 & \cdots & \cdots & d_2 \\ d_1 & d_2 & d_3 & \cdots & d_3 \\ \vdots & \vdots & & \ddots & \vdots \\ d_1 & d_2 & \cdots & \cdots & d_n \end{bmatrix}, \qquad R = \begin{bmatrix} d_1 & d_2 & \cdots & \cdots & d_n \\ d_2 & d_2 & \cdots & \cdots & d_n \\ d_3 & d_3 & d_3 & \cdots & d_n \\ \vdots & \vdots & & \ddots & \vdots \\ d_n & d_n & \cdots & \cdots & d_n \end{bmatrix} \tag{3.4}$$

and a matrix U of tree structure with respect to the weighted tree $\Gamma_{(0)}$ such that

$$A^{-1} = T \circ U \circ R.$$

Conversely, let $U = [u_{i,j}] \in \mathbb{R}^{n,n}$ be nonsingular. If U is of tree structure with respect to a given rooted tree $\Gamma_{(0)}$, then for any matrices T and R of the form (3.4) *with $d_i \neq 0$ for all i, the matrix $(T \circ U \circ R)^{-1}$ is irreducible, acyclic and $G((T \circ U \circ R)^{-1}) = \Gamma$.*

Theorem 3.4 implies several well-known results as corollaries.

COROLLARY 3.6. *Let $A \in \mathbb{R}^{n,n}$ be nonsingular, irreducible, and tridiagonal. Assume that the diagonal entries of A^{-1} are nonzero. Then there exist four vectors $u, v, x, y \in \mathbb{R}^n$ with $u_i v_i = x_i y_i$ for all i, such that $A^{-1} =: C = [c_{i,j}]$ is given by*

$$c_{i,j} = \begin{cases} u_i v_j, & i \leq j, \\ x_i y_j, & i \geq j, \end{cases} \tag{3.5}$$

i.e.,

$$C = \begin{bmatrix} u_1 & u_1 & \cdots & \cdots & u_1 \\ x_1 & u_2 & \cdots & \cdots & u_2 \\ x_1 & x_2 & u_3 & \cdots & u_3 \\ \vdots & \vdots & & \ddots & \vdots \\ x_1 & x_2 & \cdots & \cdots & u_n \end{bmatrix} \circ \begin{bmatrix} v_1 & v_2 & \cdots & \cdots & v_n \\ y_2 & v_2 & \cdots & \cdots & v_n \\ y_3 & y_3 & v_3 & \cdots & v_n \\ \vdots & \vdots & & \ddots & \vdots \\ y_n & y_n & \cdots & \cdots & v_n \end{bmatrix}. \tag{3.6}$$

Conversely, if C is nonsingular and of the form (3.5) *with $u_i v_i = x_i y_i \neq 0$ for all i, then C^{-1} is tridiagonal.*

Proof. Consider the Hadamard product form of A given in Theorem 3.4. The matrix U is of tree structure with respect to $\Gamma_{(0)}$, where $\Gamma = G(A)$. But $G(A)$ is just a path. Hence U is a type D matrix as in (1.2). Therefore

$$C = \begin{bmatrix} d_1 & d_1 & \cdots & \cdots & d_1 \\ f_1 & d_2 & \cdots & \cdots & d_2 \\ f_1 & f_2 & d_3 & \cdots & d_3 \\ \vdots & \vdots & & \ddots & \vdots \\ f_1 & f_2 & \cdots & \cdots & d_n \end{bmatrix} \circ \begin{bmatrix} g_1 & g_1 & \cdots & \cdots & g_1 \\ g_1 & g_2 & \cdots & \cdots & g_2 \\ g_1 & g_2 & g_3 & \cdots & g_3 \\ \vdots & \vdots & & \ddots & \vdots \\ g_1 & g_2 & \cdots & \cdots & g_n \end{bmatrix} \circ \begin{bmatrix} f_1 & f_2 & \cdots & \cdots & f_n \\ d_2 & f_2 & \cdots & \cdots & f_n \\ d_3 & d_3 & f_3 & \cdots & f_n \\ \vdots & \vdots & & \ddots & \vdots \\ d_n & d_n & \cdots & \cdots & f_n \end{bmatrix}.$$

Thus with $u_i = d_i g_i, x_i = f_i g_i$ and $v_i = f_i, y_i = d_i$ we get the required form of C. Moreover, $u_i v_i = x_i y_i$ for all i. For the converse we set $f_i = v_i, d_i = y_i$, and $g_i = u_i / d_i$. With Theorem 3.4 we obtain that C^{-1} is tridiagonal. ∎

For the symmetric case we obtain the well-known result below.

COROLLARY 3.7. *Let $A \in \mathbb{R}^{n,n}$ be nonsingular, symmetric, irreducible, and tridiagonal. Assume that the diagonal entries of A^{-1} are nonzero. Then there exist two vectors $u, v \in \mathbb{R}^n$ such that $A^{-1} =: C = [c_{i,j}]$ is given by*

$$C = \begin{bmatrix} u_1 & u_1 & \cdots & u_1 \\ u_1 & u_2 & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \\ u_1 & u_2 & \cdots & u_n \end{bmatrix} \circ \begin{bmatrix} v_1 & v_2 & \cdots & v_n \\ v_2 & v_2 & \cdots & v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n & v_n & \cdots & v_n \end{bmatrix}. \tag{3.7}$$

Conversely, if C is nonsingular and of the form (3.7) *with $u_i v_i \neq 0$, then C^{-1} is tridiagonal.*

Proof. The result follows immediately from Corollary 3.6 and the fact that $d_i = f_i$ for all i. ∎

In the above corollaries tridiagonal matrices are considered. In some sense the counterpart of these matrices are matrices whose graphs are stars. For these matrices we obtain the following.

COROLLARY 3.8. *Let $A \in \mathbb{R}^{n,n}$ be nonsingular and irreducible and let $G(A)$ be a star, i.e., without loss of generality $a_{i,j} = 0$ whenever i or j is not 1, $i \neq j$. Assume that diagonal entries of A^{-1} are nonzero. Then there exist diagonal matrices D_1, D_2, D_3 and a constant α such that*

$$A^{-1} = D_1(\alpha e e^t + D_3)D_2. \tag{3.8}$$

Moreover, there exist matrices T and R as in (3.3) *such that*

$$A^{-1} = T \circ (\alpha e e^T + D_3) \circ R. \tag{3.9}$$

Proof. We apply Theorem 3.4. Since the center of the star is vertex 1, the related matrix of tree structure is $\alpha e e^T + D_3$. ∎

Example 3.2.

$$A = \begin{bmatrix} 10 & -2 & -4 & -4 & -4 \\ -1 & 1 & 0 & 0 & 0 \\ -2 & 0 & 4 & 0 & 0 \\ -1 & 0 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & 4 \end{bmatrix}.$$

We have

$$A^{-1} = \frac{1}{4}\begin{bmatrix} 2 & 4 & 2 & 4 & 2 \\ 2 & 8 & 2 & 4 & 2 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 1 & 4 & 1 \\ 1 & 2 & 1 & 2 & 2 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 1 & & & & \\ & 2 & & & \\ & & 1 & & \\ & & & 2 & \\ & & & & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 2 & & & & \\ & 2 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}.$$

Example 3.3.

$$B = \begin{bmatrix} -1 & -2 & -3 & 0 & 0 \\ -2 & 2 & 0 & 0 & 0 \\ -3 & 0 & 6 & 3 & -6 \\ 0 & 0 & 3 & -3 & 0 \\ 0 & 0 & -6 & 0 & 6 \end{bmatrix}.$$

The graph $G(B)$ is just the graph given in Figure 1. The inverse of B is $\frac{1}{6} * A$, with A as in Example 2.1:

$$A = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 2 \end{bmatrix}.$$

Note that A is of tree structure with respect to $G(B)$. Moreover, we have $D = F = -I$, where I denotes the identity matrix.

Theorem 3.4 describes a simple way to construct matrices A for which $G(A^{-1})$ is a tree. However, one has to guarantee that the matrices of tree structure are nonsingular. The next theorem gives a useful characterization of nonsingularity. This characterization was already observed in [13] for so-called generalized ultrametric matrices.

THEOREM 3.9. *Let $C = [c_{i,j}] \in \mathbb{R}^{n,n}$ be of tree structure with respect to a given rooted tree. Then C is nonsingular if and only if C does not contain a row or column of zeros, and no two rows or two columns are the same.*

Proof. It is clear that C is singular if C does contain a row or column of zeros, or if two rows or two columns are the same. We prove the other implication with an induction on the dimension of C. For $n = 2$ this is obviously true.

So assume that C does not contain a row or column of zeros, and that no two rows or two columns are the same. Moreover, C has the structure

$$C = \begin{bmatrix} w_{0,1} & \dots & w_{0,1} \\ \vdots & B & \\ w_{0,1} & & \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \tilde{B} & \\ 0 & & \end{bmatrix} + w_{0,1} e e^T$$

for some $B, \tilde{B} \in \mathbb{R}^{n-1,n-1}$ and some $w_{0,1} \neq 0$. For the Schur complement $C/w_{0,1}$ we obtain

$$C/w_{0,1} = B - w_{0,1} e_{n-1} e_{n-1}^T = \tilde{B}.$$

Obviously, $\tilde{B}$ does not contain a row or column of zeros, and no two rows or two columns are the same. Otherwise C would do so. On the other hand, $\tilde{B}$ is of tree structure or the direct sum of matrices of tree structure. Thus by the induction hypothesis we get the result. ∎

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[2] W. W. BARRETT, *A theorem on inverses of tridiagonal matrices*, Linear Algebra Appl., 27 (1979), pp. 211–217.

[3] M. FIEDLER, *Some characterizations of symmetric inverse M–matrices*, Linear Algebra Appl., 275/276 (1998), pp. 179–187.

[4] F. R. GANTMACHER AND M. G. KREIN, *Sur les matrices complètement non neǵatives et oscillatoires*, Compositio Math., 4 (1937), pp. 445–470.

[5] F. R. GANTMACHER AND M. G. KREIN, *Oszillationsmatrizen, Oszillationskerne und kleine Schwingungen mechanischer Systeme*, Akademie-Verlag, Berlin, 1960 (in German).

[6] Y. IKEBE, *On inverses of Hessenberg matrices*, Linear Algebra Appl., 24 (1979), pp. 93–97.

[7] F. HARRAY, *Graph Theory*, Addison-Wesley, Reading, MA, 1972.

[8] S. KARLIN, *Total Positivity*, Stanford University Press, Stanford, CA, 1968.

[9] D. J. KLEIN, *Treediagonal matrices and their inverses*, Linear Algebra Appl., 42 (1982), pp. 109–117.

[10] S. J. KIRKLAND, M. NEUMANN, AND B. L. SHADER, *Distances in weighted trees and group inverse of Laplacian matrices*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 827–841.

[11] T. L. MARKHAM, *Nonnegative matrices whose inverses are M-matrices*, Proc. Amer. Math. Soc., 36 (1972), pp. 326–330.

[12] S. MARTÍNEZ, G. MICHON, AND J. SAN MARTÍN, *Inverses of ultrametric matrices are of Stieltjes type*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 98–106.

[13] J. J. McDonald, M. Neumann, H. Schneider, and M. J. Tsatsomeros, *Inverse M-matrix inequalities and generalized ultrametric matrices*, Linear Algebra Appl., 220 (1995), pp. 329–349.

[14] J. J. McDonald, R. Nabben, M. Neumann, H. Schneider, and M. Tsatsomeros, *Inverse tridiagonal Z–matrices*, Linear and Multilinear Algebra, 45 (1998), pp. 75–97.

[15] R. Nabben and R. S. Varga, *A linear algebra proof that the inverse of strictly ultrametric matrix is a strictly diagonally dominant Stieltjes matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 107–113.

[16] R. Nabben and R. S. Varga, *Generalized ultrametric matrices—a class of inverse M–matrices*, Linear Algebra Appl., 220 (1995), pp. 365–390.

[17] R. S. Varga and R. Nabben, *On symmetric ultrametric matrices*, in Numerical Linear Algebra, L. Reichel, A. Ruttan, R. S. Varga, eds., Walter de Gruyter, New York, 1993, pp. 193–199.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1027–1037

A CLASS OF P-MATRICES WITH APPLICATIONS TO THE LOCALIZATION OF THE EIGENVALUES OF A REAL MATRIX*

J. M. PEÑA†

Abstract. A matrix with positive row sums and all its off-diagonal elements bounded above by their corresponding row means is called a B-matrix. It is proved that the class of B-matrices is a subset of the class of P-matrices. Properties of B-matrices are used to localize the real eigenvalues of a real matrix and the real parts of all eigenvalues of a real matrix.

Key words. Gerschgorin circles, P-matrix, eigenvalues localization, sign-regular matrix, symmetric matrix

AMS subject classifications. 15A18, 15A42, 15A48, 65F15

PII. S0895479800370342

1. Introduction. Weakest linear conditions on the rows of an $n \times n$ matrix that ensure that its determinant is positive were described and analyzed in [4]. One such set of $n2^{n-1}$ linear inequalities is that the matrix is strictly diagonal dominant by rows with positive diagonal elements. Another such set of n^2 linear inequalities is that the row means are positive and larger than all the off-diagonal entries in that row. Here a matrix satisfying this property is called a B-matrix. In [6] it was already proved that these matrices have positive determinants, and a first application to the localization of the real eigenvalues of a real matrix was included. In this paper we study properties of B-matrices. We use these properties to localize the eigenvalues of a real matrix. We also analyze some cases where these alternative regions are more advantageous than Gerschgorin circles.

In section 2 we provide several characterizations of B-matrices. We also prove some properties satisfied by B-matrices and their relationship with other classes of matrices. It is shown that B-matrices are P-matrices (i.e., all their principal minors are positive), a property which will be used in section 4.

We start section 3 by introducing the class of $\bar{B}$-matrices. These matrices can be factorized as a product of a nonsingular diagonal matrix and a B-matrix. Theorem 3.5 uses this class of nonsingular matrices to derive a result for localizing the real eigenvalues of a real matrix by means of a set of intervals called row $\bar{B}$-intervals. This result has a nature similar to the Gerschgorin circle theorem. The second part of Theorem 3.5 can be applied to several classes of real matrices, including symmetric matrices. We finish section 3 by comparing $\bar{B}$-intervals with the real intervals provided by Gerschgorin circles, specializing the comparison to several classes of matrices.

If a set of linear conditions on the rows of a complex matrix implies nonsingularity then we can derive a localization result for the eigenvalues of a matrix. Analogously, if a set of linear conditions on the rows of a real matrix implies nonsingularity then we can derive a localization result for the real eigenvalues of a real matrix A. However, since the matrix $A - \lambda I$ ($\lambda \in \mathbf{C}$) is a complex matrix, in this last case we have no information on the complex eigenvalues of the real matrix A. Section 4 deals with

*Received by the editors April 7, 2000; accepted for publication (in revised form) by R. Brualdi July 28, 2000; published electronically February 23, 2001. This research was partially supported by the Research grant DGES PB96-0730, Spain.
http://www.siam.org/journals/simax/22-4/37034.html
†Departamento de Matemática Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain (jmpena@posta.unizar.es).

the localization of the real parts of all eigenvalues of a real matrix. This is obtained in Theorem 4.3, whose second part can be applied to any real matrix, in contrast to the second part of Theorem 3.5. Each of the intervals obtained in Theorem 4.3 uses information on one row and one column of the matrix. In order to obtain Theorem 4.3 we apply some properties of B-matrices. Finally, by Proposition 4.6, the arguments given in this section to derive information on the real parts of all eigenvalues of a real matrix can be extended to any class of P-matrices closed under addition, multiplication by positive scalars and transposition which is formed by matrices satisfying linear conditions on rows and columns.

2. B-matrices. Let us introduce some basic notations. Given k, $n \in \mathbf{N}$, $1 \leq k \leq n$, $Q_{k,n}$ will denote the set of all increasing sequences of k natural numbers less than or equal to n. Given $\alpha \in Q_{k,n}$, the complement $\alpha' \in Q_{n-k,n}$ is the increasingly rearranged $\{1, 2, \ldots, n\} \setminus \alpha$. Let A be a real $m \times n$ matrix. For $k \leq m$, $l \leq n$, and for any $\alpha \in Q_{k,m}$ and $\beta \in Q_{l,n}$, we denote by $A[\alpha|\beta]$ the $k \times l$ submatrix of A containing rows numbered by α and columns numbered by β. The principal submatrices will be written in the form $A[\alpha] := A[\alpha|\alpha]$. The identity matrix will be denoted by I.

The next definition introduces B-matrices, a class of nonsingular matrices which will be crucial in this paper.

DEFINITION 2.1. *We say that a square real matrix $A = (a_{ik})_{1 \leq i,k \leq n}$ with positive row sums is a B-matrix if all of its off-diagonal elements are bounded above by the corresponding row means, i.e., for all $i = 1, \ldots, n$*

$$\sum_{k=1}^{n} a_{ik} > 0 \quad \textit{and} \quad \frac{1}{n}\left(\sum_{k=1}^{n} a_{ik}\right) > a_{ij} \ \ \forall\, j \neq i.$$

In [6] it was proved that a B-matrix is nonsingular and has positive determinant (see also [4, Corollary 4.5]). Let us observe that the definition of an $n \times n$ B-matrix involves n^2 linear inequalities. In [4] it was proved that this set of inequalities forms a weakest set of linear conditions on the rows of a real $n \times n$ matrix to ensure positive determinant. In that paper it was also proved that another weakest set of linear conditions to ensure positive determinant is provided by the $n2^{n-1}$ inequalities corresponding to strict diagonal dominance by rows with positive entries. In this section we show several properties satisfied by B-matrices, including that they are P-matrices, a property which is also shared by matrices which are strictly diagonally dominant by rows with positive entries.

From the previous definition we can deduce that the diagonal elements of a B-matrix satisfy for all $i = 1, \ldots, n$

$$a_{ii} > \max\{0, a_{ij} \mid j \neq i\}, \tag{2.1}$$

and therefore each row mean of a B-matrix is bounded below by any off-diagonal element of the row and bounded above by the diagonal element of the row.

Let $A = (a_{ik})_{1 \leq i,k \leq n}$ be a real matrix. From now on, we shall use the following notations: for each $i = 1, \ldots, n$

$$r_i^+ := \max\{0, a_{ij} \mid j \neq i\}, \quad r_i^- := \min\{0, a_{ij} \mid j \neq i\}, \quad r_i := \begin{cases} r_i^+ & \text{if } a_{ii} > 0, \\ r_i^- & \text{if } a_{ii} < 0 \end{cases} \tag{2.2}$$

and for each $j = 1, \ldots, n$

$$c_j^+ := \max\{0, a_{ij} \mid i \neq j\}, \quad c_j^- := \min\{0, a_{ij} \mid i \neq j\}, \quad c_j := \begin{cases} c_j^+ & \text{if } a_{jj} > 0, \\ c_j^- & \text{if } a_{jj} < 0. \end{cases} \tag{2.3}$$

The next result provides a characterization of B-matrices which can be derived from Definition 2.1.

PROPOSITION 2.2. *Let $A = (a_{ik})_{1\le i,k\le n}$ be a real matrix and, for each $i = 1,\dots,n$, let r_i^+ be the number given in (2.2). Then A is a B-matrix if and only if for all $i \in \{1,\dots,n\}$*

$$\sum_{k=1}^{n} a_{ik} > n r_i^+. \tag{2.4}$$

By (2.1) and the previous proposition we can characterize $n \times n$ B-matrices with $n > 2$ by the following property, which localizes each row mean in an open interval:

$$\frac{\sum_{k=1}^{n} a_{ik}}{n} \in (r_i^+, a_{ii}), \quad i = 1,\dots,n. \tag{2.5}$$

By rearranging the terms of (2.4), Proposition 2.2 also provides the following characterization, which will be very useful in the next section.

PROPOSITION 2.3. *Let $A = (a_{ik})_{1\le i,k\le n}$ be a real matrix and, for each $i = 1,\dots,n$, let r_i^+ be the number given in (2.2). Then A is a B-matrix if and only if for all $i \in \{1,\dots,n\}$*

$$a_{ii} - r_i^+ > \sum_{j\neq i} (r_i^+ - a_{ij}). \tag{2.6}$$

The following results collect some properties of B-matrices.

PROPOSITION 2.4. *Let $A = (a_{ik})_{1\le i,k\le n}$ be a real matrix and, for each $i = 1,\dots,n$, let r_i^+ be the number given in (2.2). Then the following properties hold for all $i = 1,\dots,n$:*

(i) $a_{ii} > \sum_{h\in H} |a_{ih}|$, *where* $H = \{h \mid 1 \le h \le n \quad \text{and} \quad a_{ih} < 0\}$.

(ii) $a_{ii} > |a_{ij}| \quad \forall\, j \neq i$.

Proof. (i) follows from (2.6), taking into account that $a_{ii} \ge a_{ii} - r_i^+$, $r_i^+ - a_{ij} \ge 0$ for all $j \neq i$ and $r_i^+ - a_{ij} \ge |a_{ij}|$ if $a_{ij} < 0$.

(ii) is a consequence of (i) and (2.1). ∎

From Definition 2.1 it is straightforward to check that the sum of two B-matrices is a B-matrix and that the multiplication of a positive real number and a B-matrix is a B-matrix. The following result shows that being a B-matrix is a property inherited by principal submatrices.

PROPOSITION 2.5. *The principal submatrices of a B-matrix are B-matrices.*

Proof. Let $A[\alpha]$, $\alpha \in Q_{k,n}$, be any principal submatrix of an $n \times n$ B-matrix A. By Proposition 2.4(i), $A[\alpha]$ has positive row sums. It is now sufficient to assume that, for some two different indices $i, j \in \alpha$, $k a_{ij} \ge \sum_{s\in\alpha} a_{is}$, deriving a contradiction. Let α' be the complement of α. If a_{il} is the greatest off-diagonal element of the ith row of A (i.e., $l \neq i$ and $a_{il} \ge a_{it}$ for any $t \neq i$, $1 \le t \le n$) then $k a_{il} \ge k a_{ij}$ and so

$$n a_{il} \ge k a_{il} + \sum_{r\in\alpha'} a_{ir} \ge \sum_{s\in\alpha} a_{is} + \sum_{r\in\alpha'} a_{ir} \ge \sum_{p=1}^{n} a_{ip},$$

which contradicts the fact that A is a B-matrix. ∎

A matrix with positive principal minors is called a P-matrix. From the previous result joint with the fact that B-matrices have positive determinant (see [4, Corollary 4.5]) we obtain the following consequence.

COROLLARY 2.6. *B-matrices are P-matrices.*

We can now add to our list of properties of B-matrices all properties known for the class of P-matrices (see [2, Chapter 10, section 2]). Corollary 2.6 also implies the following result.

COROLLARY 2.7. *A symmetric B-matrix is positive definite.*

Recall that a square real matrix is a Z-matrix if all of its off-diagonal elements are nonpositive. We finish this section by showing that for Z-matrices, the concept of being a B-matrix coincides with the strict diagonal dominance by rows.

PROPOSITION 2.8. *Let A be a Z-matrix. Then the following properties are equivalent:*

(i) *A is a B-matrix.*

(ii) *The row sums of A are positive.*

(iii) *A is strictly diagonally dominant by rows with positive diagonal entries.*

Proof. By Definition 2.1, (i) implies (ii). Taking into account that the off-diagonal elements of A are nonpositive, (ii) implies (iii) and (iii) implies (i). ∎

Recall that a nonsingular Z-matrix whose inverse is nonnegative is called an M-matrix. We deduce from the equivalence of (i) and (iii) in Proposition 2.8 and property (M_{35}) of [2, Chapter 6, Theorem 2.3] that a B-matrix is a Z-matrix if and only if it is an M-matrix.

3. Applications to the localization of real eigenvalues of a real matrix. We start this section by introducing a class of nonsingular matrices closely related to B-matrices.

DEFINITION 3.1. *We say that a real matrix is a $\bar{B}$-matrix if it is of the form DA where D is a diagonal matrix whose diagonal elements belong to the set $\{1,-1\}$ and A is a B-matrix.*

Remark 3.2. If $A=(a_{ij})_{1\le i,j\le n}$ is a $\bar{B}$-matrix then it is a nonsingular matrix because B-matrices are nonsingular (see [4, Corollary 4.5]). On the other hand, since by (2.1) the diagonal elements of a B-matrix are positive, we conclude that the diagonal elements of a $\bar{B}$-matrix are nonzero. Finally, observe that a $\bar{B}$-matrix with positive diagonal elements is a B-matrix.

The following result, motivated by Proposition 2.3, characterizes $\bar{B}$-matrices.

PROPOSITION 3.3. *Let $A=(a_{ik})_{1\le i,k\le n}$ be a real matrix and let r_i be as in* (2.2). *Then A is a $\bar{B}$-matrix if and only if for all $i=1,\dots,n$*

$$|a_{ii}-r_i| > \sum_{j\neq i} |r_i - a_{ij}|. \tag{3.1}$$

Proof. Let D be the matrix of Definition 3.1 and let r_i^+, r_i^- be the numbers given in (2.2). Observe that A is a $\bar{B}$-matrix if and only if DA is a B-matrix. The ith row of DA is given by $(a_{i1},\dots,a_{in})$ if $a_{ii}>0$ and by $(-a_{i1},\dots,-a_{in})$ if $a_{ii}<0$. By Proposition 2.3 DA is a B-matrix if and only if (2.6) holds if $a_{ii}>0$ and

$$-a_{ii}-(-r_i^-) > \sum_{j\neq i}(-r_i^- - (-a_{ij}))$$

if $a_{ii}<0$. Both cases are equivalent to (3.1) and the result follows. ∎

If a diagonal element a_{ii} of a $\bar{B}$-matrix $A=(a_{ij})_{1\le i,j\le n}$ has opposite sign to the off-diagonal elements of its row then $r_i=0$ (see (2.2)) and by (3.1) the corresponding row satisfies the strict diagonal dominance condition. If a_{kk} does not satisfy the previous property and we define $A'=(a'_{ij})_{1\le i,j\le n}$ by $a'_{ij}:=a_{ij}$ if $i\neq k$ and $a'_{kj}:=$

$a_{kj} - r_k$ for $1 \leq j \leq n$, then the kth row of A' also satisfies the strict diagonal dominance condition.

Remark 3.4. The fact that A^T is a $\bar{B}$-matrix is equivalent, by Proposition 3.3, to the fact that for all $j = 1, \ldots, n$

$$|a_{jj} - c_j| > \sum_{i \neq j} |c_j - a_{ij}|, \tag{3.2}$$

where c_j is given by (2.3). A matrix satisfying (3.2) for all $j = 1, \ldots, n$ is nonsingular because, by Remark 3.2, its transpose is nonsingular.

Given a matrix $B = (b_{ik})_{1 \leq i,k \leq n}$, let us define the family of matrices

$$B_t := D + t(B - D), \quad t \in [0, 1], \tag{3.3}$$

where D is the diagonal matrix $\operatorname{diag}\{b_{11}, \ldots, b_{nn}\}$. We can now prove a result on the localization of the real eigenvalues of a real matrix of a nature similar to the Gerschgorin circles theorem.

THEOREM 3.5. *Let $A = (a_{ik})_{1 \leq i,k \leq n}$ be a real matrix; let r_i^+, r_i^- be as in* (2.2)*; and let λ be a real eigenvalue of A. Then*

(i) $\lambda \in S := \bigcup_{i=1}^n [a_{ii} - r_i^+ - \sum_{k \neq i} |r_i^+ - a_{ik}|\,,\, a_{ii} - r_i^- + \sum_{k \neq i} |r_i^- - a_{ik}|]$.

(ii) *Let $\mathcal{C}$ be a class of real matrices such that if $B \in \mathcal{C}$ then all eigenvalues of B are real and all matrices of the form* (3.3) *belong to $\mathcal{C}$ and let us assume that $A \in \mathcal{C}$. If S' is the union of m intervals of S such that S' is disjoint from all other intervals, then S' contains precisely m eigenvalues (counting multiplicities) of A.*

Proof. (i) Taking into account that $A - \lambda I$ has the same off-diagonal elements as A and Proposition 3.3, we can deduce that if $\lambda \notin S$ then $A - \lambda I$ is a $\bar{B}$-matrix and, by Remark 3.2, $A - \lambda I$ is nonsingular. Therefore (i) holds.

(ii) For each $i = 1, \ldots, n$ and for every $t \in [0, 1]$, let $S_t := \cup_{i=1}^n [\alpha_{i,t}, \beta_{i,t}]$, where $\alpha_{i,t} := a_{ii} - t r_i^+ - t \sum_{k \neq i} |r_i^+ - a_{ik}|$, $\beta_{i,t} := a_{ii} - t r_i^- + t \sum_{k \neq i} |r_i^- - a_{ik}|$. Then we can write $S' = \cup_{j=1}^m [\alpha_{i_j,1}, \beta_{i_j,1}]$, and let $S'_t := \cup_{j=1}^m [\alpha_{i_j,t}, \beta_{i_j,t}]$ and $S''_t := S_t \setminus S'_t$. By our hypotheses, S''_1 is disjoint from S' and, since $t \in [0, 1]$ and $[\alpha_{i,t}, \beta_{i,t}] \subseteq [\alpha_{i,1}, \beta_{i,1}]$ for all $i = 1, \ldots, n$, this implies that S''_t and S'_t are disjoint for all $t \in [0, 1]$. By our hypotheses on $\mathcal{C}$, all eigenvalues of A_t are real and, by (i), they are contained in $S'_t \cup S''_t$ for all $t \in [0, 1]$. Since S''_t and S'_t are disjoint, the continuity of the eigenvalues as functions of the elements of the matrix joint with the fact that S'_0 contains m eigenvalues of the diagonal matrix A_0 (namely, $a_{i_1}, \ldots, a_{i_m}$) imply that S'_t must contain m eigenvalues of A_t for all $t \in [0, 1]$ and (ii) follows. ∎

The following concept plays in this paper a role similar to Gerschgorin row-regions (see [3]) in the Gerschgorin theorem.

DEFINITION 3.6. *The intervals appearing in Theorem* 3.5(i) *will be called row $\bar{B}$-intervals.*

Part (i) of Theorem 3.5 was also derived in [6], with an equivalent expression of S. The expression of S given here will be used later to analyze when row $\bar{B}$-intervals are more advantageous than Gerschgorin row-regions for the localization of the real eigenvalues of a real matrix.

The following remark shows some classes of matrices $\mathcal{C}$ satisfying the hypotheses required in part (ii) of Theorem 3.5.

Remark 3.7. The second part of Theorem 3.5 can be applied to any class of real matrices $\mathcal{C}$ such that if $B \in \mathcal{C}$ then all eigenvalues of B are real and all matrices B_t of the form (3.3) also belong to $\mathcal{C}$. This happens with the following classes of real matrices:

(I) The class of symmetric matrices.

(II) The class of matrices with disjoint Gerschgorin circles.

(III) The class $\mathcal{C}$ of matrices such that the row $\bar{B}$-intervals are disjoint. Let us justify this case with an argument very similar to the known argument of case II. Given $B \in \mathcal{C}$, since for all $t \in [0,1]$ the row $\bar{B}$-intervals of matrices B_t of (3.3) are contained in the row $\bar{B}$-intervals of $B_1 = B$ then all matrices B_t belong to $\mathcal{C}$ whenever $B \in \mathcal{C}$, and it is sufficient to see that the eigenvalues of matrices in this class $\mathcal{C}$ are real. Let us assume that $B = B_1 \in \mathcal{C}$ has some nonreal eigenvalues and let $\hat{t}$ be the infimum number in $\{t \in [0,1] \mid B_t \text{ has some nonreal eigenvalues}\}$. Since B_0 is a diagonal matrix, its eigenvalues are real and its row $\bar{B}$-intervals are its diagonal elements. If $\hat{t} > 1$, for all $t < \hat{t}$ the eigenvalues of B_t are real and by Theorem 3.5 and our definition of $\mathcal{C}$, each of them belongs to an interval which is disjoint from the other ones. The eigenvalues of $B_{\hat{t}}$ are also real by the continuity of the eigenvalues as functions of the coefficients of the matrix. By our choice of $\hat{t}$, we have, for $t > \hat{t}$ close enough to $\hat{t}$, that there are two nonreal (conjugate) eigenvalues of B_t converging to a real number when t converges to $\hat{t}^+$, contradicting the mentioned continuity of the eigenvalues. This contradiction shows that this class of matrices satisfies the properties required above.

In section 4 we shall obtain an extension of Theorem 3.5 in order to localize the real parts of all eigenvalues of a real matrix. This extension uses intervals containing the row $\bar{B}$-intervals (in fact it uses the union of row $\bar{B}$-intervals and column $\bar{B}$-intervals), and the corresponding second part of the theorem will be applied to all real matrices (without the requirement of classes $\mathcal{C}$ of Theorem 3.5(ii)).

We now include two examples showing that, in some cases, row $\bar{B}$-intervals provide sharp bounds to localize the real eigenvalues, even in spite that the bounds obtained from the union of intervals derived from the Gerschgorin circles are not sharp.

Example 3.8. Let us consider the matrix $A = (a_{ik})_{1 \le i,k \le n}$ with $a_{ik} = 1$ for all i, k. Its eigenvalues are $\lambda = 0$ (with multiplicity $n-1$) and $\lambda = n$. The $\bar{B}$-intervals are all $[0, n]$ and they are sharp, in contrast with the interval derived from the Gerschgorin circles: $[-n+2, n]$. This also happens with the matrix $B = (b_{ik})_{1 \le i,k \le n}$ with $b_{ik} = 1$ if $i = k$ and $b_{ik} = -1$ otherwise. Its eigenvalues are $\lambda = 2$ (with multiplicity $n-1$) and $\lambda = -n+2$. The $\bar{B}$-intervals now coincide with $[-n+2, 2]$ and the interval derived from the Gerschgorin circles is $[-n+2, n]$.

By using A^T instead of A and applying Remark 3.4, Theorem 3.5 is valid if we change rows by columns and the elements r_i^+, r_i^- of (2.2) by c_i^+, c_i^- of (2.3). Then we could define a set of column $\bar{B}$-intervals, which would also contain all real eigenvalues of the matrix. Information obtained from row $\bar{B}$-intervals can be complemented with the information obtained from column $\bar{B}$-intervals, and this information also can be complemented with the information derived from Gerschgorin circles. A natural question arises in this context in order to localize the real eigenvalues of a real matrix: can we combine Theorem 3.5(i) with the union of intervals provided by Gerschgorin row-regions so that we get a criterium which improves both mentioned criteria? The following example gives a negative answer to this question by showing that we cannot choose for each row the intersection of the real interval provided by the Gerschgorin circle theorem and the row $\bar{B}$-interval.

Example 3.9. The matrix

$$A = \begin{pmatrix} 5 & 4 & 4 \\ 4 & 5 & 0 \\ 4 & 0 & 5 \end{pmatrix}$$

has a negative eigenvalue. The $\bar{B}$-interval corresponding to the first row is [1,13] and the interval provided by Gerschgorin circle theorem for the second and third row is [1,9]. Therefore the union of these intervals does not contain all real eigenvalues of A.

It is well known (see [5], [9]) that Gerschgorin circles possess optimal properties among other possible results on localization of eigenvalues by means of circles which depend on the absolute value of the off-diagonal elements. Let us consider the question about which matrices are more suitable using row $\bar{B}$-intervals than Gerschgorin row-regions (and conversely) in order to localize its real eigenvalues. The sizes of the row $\bar{B}$-intervals depend on the size of the greatest and least off-diagonal elements and on the dispersion of off-diagonal elements. In fact, if all off-diagonal elements of each row are very similar then the distance from the diagonal element to one of the endpoints of each row $\bar{B}$-interval is approximately given by one off-diagonal element ($n-1$ times smaller than using Gerschgorin circles), although the distance to the other endpoint is the same as using Gerschgorin circles. For the case of nonnegative matrices and Z-matrices, the following remark describes the set of matrices for which all row $\bar{B}$-intervals are smaller than the real intervals provided by Gerschgorin row-regions, and the set of matrices satisfying the converse condition.

Remark 3.10. Given a nonnegative matrix $A = (a_{ik})_{1\le i,k\le n}$, the right endpoints of the row $\bar{B}$-intervals and the right endpoints of the real intervals provided by Gerschgorin row-regions coincide. Since A is nonnegative, for each $i = 1, \dots, n$, there exists $j \neq i$ such that $r_i^+ = a_{ij}$. Let us now compare the corresponding left endpoints. The left endpoints of the real intervals provided by Gerschgorin row-regions are given by $a_{ii} - a_{ij} - \sum_{k\neq i,j} a_{ik}$, $i = 1, \dots, n$, and the left endpoints of the row $\bar{B}$-intervals can be written as

$$a_{ii} - nr_i^+ + \sum_{k\neq i} a_{ik} = a_{ii} - a_{ij} - (n-2)r_i^+ + \sum_{k\neq i,j} a_{ik}, \quad i = 1, \dots, n.$$

One of these left endpoints is greater than the Gerschgorin left endpoint if and only if

$$r_i^+ < \frac{2\sum_{k\neq i,j} a_{ik}}{n-2}. \tag{3.4}$$

Taking into account that $\sum_{k\neq i,j} a_{ik} \in [0, (n-2)r_i^+]$, (3.4) is equivalent to

$$\sum_{k\neq i,j} a_{ik} \in \left(\frac{n-2}{2} r_i^+, (n-2)r_i^+\right]. \tag{3.5}$$

Conversely, one of the left endpoints of the real intervals provided by Gerschgorin circles is greater than the corresponding one of the row $\bar{B}$-intervals if for the corresponding index i one has

$$\sum_{k\neq i,j} a_{ik} \in \left[0, \frac{n-2}{2} r_i^+\right). \tag{3.6}$$

Therefore, for the localization of the real eigenvalues of a nonnegative matrix such that all its rows satisfy (3.6) it is more convenient to use Gerschgorin circles, and if all of its rows satisfy (3.5) it is more convenient to apply Theorem 3.5(i). Analogous conclusions can be derived if A is a Z-matrix. In this case, for each $i = 1, \dots, n$, there exists $j \neq i$ such that $r_i^- = a_{ij}$. Then the left endpoints of the row $\bar{B}$-intervals and the left endpoints of the real intervals provided by Gerschgorin row-regions coincide.

As for the right endpoints, take into account that now $\sum_{k\neq i,j} a_{ik} \in [(n-2)r_i^-, 0]$, and replace (3.5) by

$$\sum_{k\neq i,j} a_{ik} \in \left[(n-2)r_i^-, \frac{n-2}{2}r_i^-\right) \tag{3.7}$$

and (3.6) by

$$\sum_{k\neq i,j} a_{ik} \in \left(\frac{n-2}{2}r_i^-, 0\right]. \tag{3.8}$$

4. On the localization of the real parts of the eigenvalues. Theorem 3.5(i) provides a region containing all real eigenvalues of a real matrix. However, a region containing all real eigenvalues of a real matrix does not necessarily include the real parts of all of its eigenvalues. For instance, let us recall that the real eigenvalues of a P-matrix are contained in $(0,\infty)$ although there exist P-matrices with eigenvalues λ such that $\mathrm{Re}(\lambda) \notin (0,\infty)$. In Theorem 4.3 we shall extend Theorem 3.5 in order to localize the real parts of all eigenvalues of a real matrix.

Given a matrix $A = (a_{ij})_{1\leq i,j\leq n}$, let $\mathrm{Re}(A) := (\mathrm{Re}(a_{ij}))_{1\leq i,j\leq n}$. The matrix

$$H(A) = \left(\frac{1}{2}(a_{ij} + \bar{a}_{ji})\right)_{1\leq i,j\leq n} \tag{4.1}$$

is called the Hermitian part of A. The following result will use the fact that a B-matrix is a P-matrix (in fact, its consequence Corollary 2.7) and will allow us to analyze the localization of the real parts of the eigenvalues of a real matrix.

PROPOSITION 4.1. *If $A = (a_{ik})_{1\leq i,k\leq n}$ is a complex matrix such that its off-diagonal entries are real and $\mathrm{Re}(A)$ and $\mathrm{Re}(A^T)$ are B-matrices then A is nonsingular.*

Proof. Given the matrix A, let r_i^+ and c_i^+ as in (2.2) and (2.3), respectively. Observe that the matrix $H(A)$ of (4.1) is a symmetric real matrix. Let us first see that $H(A)$ is a B-matrix.

Taking into account that the off-diagonal entries of $\mathrm{Re}(A)$ and $\mathrm{Re}(A^T)$ coincide with those of A and A^T, respectively, and applying Proposition 2.2 to $\mathrm{Re}(A)$ and $\mathrm{Re}(A^T)$ we derive for all $i \in \{1,\dots,n\}$

$$\mathrm{Re}(a_{ii}) + \sum_{j\neq i}^{n} a_{ij} > nr_i^+ \quad \text{and} \quad \mathrm{Re}(a_{ii}) + \sum_{j\neq i}^{n} a_{ji} > nc_i^+,$$

and so

$$2\mathrm{Re}(a_{ii}) + \sum_{j\neq i}^{n}(a_{ij} + a_{ji}) > n(r_i^+ + c_i^+).$$

Hence we obtain

$$\mathrm{Re}(a_{ii}) + \frac{\sum_{j\neq i}^{n}(a_{ij}+a_{ji})}{2} > n\frac{(r_i^+ + c_i^+)}{2} \geq n\max\left\{0, \frac{a_{ij}+a_{ji}}{2} \mid j \neq i\right\}$$

and, again by Proposition 2.2, $H(A)$ is a B-matrix.

Therefore the symmetric matrix $H(A)$ is a B-matrix and, by Corollary 2.7, it is positive definite. So $H(A)$ has only positive eigenvalues. Since by [7, Corollary 3.15] the least singular value of a matrix A is greater than or equal to the least eigenvalue of $H(A)$, we now deduce that the least singular value of A is positive and therefore A is nonsingular. □

COROLLARY 4.2. *If A is a singular complex matrix whose off-diagonal entries are real then either* $\mathrm{Re}(A)$ *or* $\mathrm{Re}(A^T)$ *is not a $\bar{B}$-matrix.*

Proof. Let us assume that $\mathrm{Re}(A)$ and $\mathrm{Re}(A^T)$ are $\bar{B}$-matrices and we shall prove that A is nonsingular. By our assumption there exists a nonsingular diagonal matrix D such that $D\mathrm{Re}(A) = \mathrm{Re}(DA)$ is a B-matrix and so, by (2.1), it has only positive diagonal elements. Then its transpose $\mathrm{Re}(A^T)D = \mathrm{Re}(A^TD) = \mathrm{Re}((DA)^T)$ is a B-matrix with positive diagonal elements and, by Remark 3.2, it is also a B-matrix. Then, by Proposition 4.1, the matrix DA with real off-diagonal entries is nonsingular and A is nonsingular. □

By using information on rows and columns we can now derive the following extension of Theorem 3.5 to localize the real parts of all eigenvalues of a real matrix.

THEOREM 4.3. *Let $A = (a_{ik})_{1\le i,k\le n}$ be a real matrix; let $r_i^+, r_i^-, c_i^+, c_i^-$ be as in* (2.2), (2.3)*; and let λ be an eigenvalue of A. Then*

(i) $\mathrm{Re}(\lambda) \in S^* := \cup_{i=1}^n [\alpha_i, \beta_i]$, *where for each $i = 1, \dots, n$*

$$
\begin{aligned}
\alpha_i &:= \min\left\{ a_{ii} - r_i^+ - \sum_{k\neq i} |r_i^+ - a_{ik}|, a_{ii} - c_i^+ - \sum_{k\neq i} |c_i^+ - a_{ki}| \right\}, \\
\beta_i &:= \max\left\{ a_{ii} - r_i^- + \sum_{k\neq i} |r_i^- - a_{ik}|, a_{ii} - c_i^- + \sum_{k\neq i} |c_i^- - a_{ki}| \right\};
\end{aligned} \tag{4.2}
$$

(ii) *if S' is the union of m intervals of S^* such that S' is disjoint from all other intervals, then S' contains precisely the real part of m eigenvalues (counting multiplicities) of A.*

Proof. (i) Since $A-\lambda I$ is singular, by Corollary 4.2 either $\mathrm{Re}(A-\lambda I) = A-\mathrm{Re}(\lambda)I$ or $\mathrm{Re}((A-\lambda I)^T) = A^T - \mathrm{Re}(\lambda)I$ is not a $\bar{B}$-matrix. Part (i) is now a consequence of Proposition 3.3, taking into account that $A - \mathrm{Re}(\lambda)I$ and $A^T - \mathrm{Re}(\lambda)I$ have the same off-diagonal entries as A and A^T, respectively.

(ii) For each $i = 1, \dots, n$ and for every $t \in [0,1]$, let $S_t := \cup_{i=1}^n [\alpha_{i,t}, \beta_{i,t}]$, where

$$
\begin{aligned}
\alpha_{i,t} &:= \min\left\{ a_{ii} - tr_i^+ - t\sum_{k\neq i} |r_i^+ - a_{ik}|, a_{ii} - tc_i^+ - t\sum_{k\neq i} |c_i^+ - a_{ki}| \right\}, \\
\beta_{i,t} &:= \max\left\{ a_{ii} - tr_i^- + t\sum_{k\neq i} |r_i^- - a_{ik}|, a_{ii} - tc_i^- + t\sum_{k\neq i} |c_i^- - a_{ki}| \right\}.
\end{aligned}
$$

If $S' = \cup_{j=1}^m [\alpha_{i_j}, \beta_{i_j}]$, let $S'_t := \cup_{j=1}^m [\alpha_{i_j,t}, \beta_{i_j,t}]$ and $S''_t := S_t \setminus S'_t$. By our hypotheses, S''_1 is disjoint from S' and, since $[\alpha_{i,t}, \beta_{i,t}] \subseteq [\alpha_i, \beta_i]$ for all $i = 1, \dots, n$ and $t \in [0,1]$, this implies that S''_t and S'_t are disjoint for all $t \in [0,1]$. By (i) and our hypotheses, the eigenvalues of A_t have their real parts contained in $S'_t \cup S''_t$ for all $t \in [0,1]$. Since S''_t and S'_t are disjoint, the continuity of the eigenvalues as functions of the elements of the matrix joint with the fact that S'_0 contains m eigenvalues of the diagonal matrix

A_0 imply that S'_t must contain the real part of m eigenvalues of A_t for all $t \in [0,1]$ and (ii) follows. ∎

If all eigenvalues of the real matrix A are real then we can deduce the following consequence from Theorem 4.3.

COROLLARY 4.4. *Let $A = (a_{ik})_{1\leq i,k\leq n}$ be a real matrix whose eigenvalues are real, let $r_i^+, r_i^-, c_i^+, c_i^-$ be as in (2.2), (2.3), and let λ be an eigenvalue of A. Then*

(i) *$\lambda \in S^* := \cup_{i=1}^n [\alpha_i, \beta_i]$, where α_i, β_i are given in (4.2);*

(ii) *if S' is the union of m intervals of S^* such that S' is disjoint from all other intervals, then S' contains precisely m eigenvalues (counting multiplicities) of A.*

Observe that if we specialize Corollary 4.4 to the class of symmetric matrices we obtain the same information as in Theorem 3.5.

Remark 4.5. Matrices with all minors of each order with the same sign are called *sign-regular* (see [1], [8]). This class of matrices has many applications and contains the class of totally positive matrices (matrices whose minors are nonnegative). In addition to the classes of matrices considered in Remark 3.7, we can now apply Corollary 4.4 to localize all eigenvalues of sign-regular matrices since all their eigenvalues are real (cf. [1, Corollary 6.6]). However, this class of matrices does not satisfy the conditions stated in Theorem 3.5(ii): the matrix

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ 4 & 9 & 16 \end{pmatrix}$$

is sign-regular (even totally positive) but the matrix B_t given by (3.3) with $t = 1/2$,

$$\begin{pmatrix} 1 & 1/2 & 1/2 \\ 1 & 3 & 2 \\ 2 & 9/2 & 16 \end{pmatrix},$$

is not sign-regular because it has 2×2 minors with different strict sign. On the other hand, observe that Corollary 4.4 can also be applied to localize all eigenvalues of inverses of sign-regular matrices.

We finish this section by showing that the arguments used in this section can be extended to any set of linear conditions on the rows which ensure positive determinants are preserved under summation and multiplication by positive scalars and are inherited by principal submatrices. For this purpose, we consider the class $\mathcal{B}$ of matrices whose rows and columns satisfy such set of linear conditions. Then $\mathcal{B}$ is a class of P-matrices which is closed under addition, multiplication by positive scalars, and transposition, and then Proposition 4.1 can be generalized to $\mathcal{B}$.

PROPOSITION 4.6. *Let $\mathcal{B}$ be any class of P-matrices which is closed under addition, multiplication by positive scalars, and transposition, and let A be a complex matrix whose off-diagonal entries are real. If $\mathrm{Re}(A) \in \mathcal{B}$, then A is nonsingular.*

Proof. By our hypotheses, the matrix $H(A)$ of (4.1) is real and symmetric. By our assumptions on $\mathcal{B}$, $(\mathrm{Re}(A))^T = \mathrm{Re}(A^T) \in \mathcal{B}$, $\mathrm{Re}(A)+\mathrm{Re}(A^T) \in \mathcal{B}$, and $H(A) \in \mathcal{B}$. Then $H(A)$ is a symmetric P-matrix and therefore it has only positive eigenvalues. Since by [7, Corollary 3.15] the least singular value of a matrix A is greater than or equal to the least eigenvalue of $H(A)$, we conclude that the least singular value of A is positive and therefore A is nonsingular. ∎

REFERENCES

[1] T. Ando, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.
[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1996.
[3] R. A. Brualdi and S. Mellendorf, *Regions in the complex plane containing the eigenvalues of a matrix*, Amer. Math. Monthly, 101 (1994), pp. 975–985.
[4] J. M. Carnicer, T. N. T. Goodman, and J. M. Peña, *Linear conditions for positive determinants*, Linear Algebra Appl., 292 (1999), pp. 39–59.
[5] K. Fan, *Note on circular disks containing the eigenvalues of a matrix*, Duke Math. J., 25 (1958), pp. 441–445.
[6] A. J. Hoffman, *On the nonsingularity of real matrices*, Math. Comp., 19 (1965), pp. 56–61.
[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[8] J. M. Peña, *Backward stability of a pivoting strategy for sign-regular linear systems*, BIT, 37 (1997), pp. 910–924.
[9] R. S. Varga, *Minimal Gerschgorin sets*, Pacific J. Math., 15 (1965), pp. 719–729.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1038–1057

NEWTON'S METHOD IN FLOATING POINT ARITHMETIC AND ITERATIVE REFINEMENT OF GENERALIZED EIGENVALUE PROBLEMS*

FRANÇOISE TISSEUR†

Abstract. We examine the behavior of Newton's method in floating point arithmetic, allowing for extended precision in computation of the residual, inaccurate evaluation of the Jacobian and unstable solution of the linear systems. We bound the limiting accuracy and the smallest norm of the residual. The application that motivates this work is iterative refinement for the generalized eigenvalue problem. We show that iterative refinement by Newton's method can be used to improve the forward and backward errors of computed eigenpairs.

Key words. Newton's method, generalized eigenvalue problem, iterative refinement, Cholesky method, backward error, forward error, rounding error analysis, limiting accuracy, limiting residual

AMS subject classifications. 65F15, 65F35

PII. S0895479899359837

1. Introduction.

This work is motivated by the symmetric definite generalized eigenvalue problem $Ax = \lambda Bx$ (A and B symmetric and one of them positive definite), for which no method is known that takes advantage of the symmetry, is efficient, and is backward stable. For the special case where both matrices are positive definite, such a method is available [26]. The aim is to show that iterative refinement by Newton's method can be used to improve the forward and backward errors of computed eigenpairs. An important question is how accurately the residuals must be evaluated in order to improve the relative forward error and/or the backward error.

For added generality we give a detailed analysis of the general Newton method in floating point arithmetic, allowing for extended precision in computation of the residual, possibly inaccurate evaluation of the Jacobian and unstable linear system solvers. We bound the limiting accuracy that can be obtained and the smallest norm of the residual.

Lancaster [19], Woźniakowski [28], Ypma [29], [30], and Dennis and Walker [6] have also considered the effects of inaccuracy, computational or otherwise, on Newton's method for solving nonlinear algebraic equations. None of these authors analyzed the behavior of the residual. Lancaster and Ypma were interested in how the approximate iterate is related to the exact one rather than the error in the approximate iterate. Woźniakowski carried out his analysis with the big-Oh notation and therefore his results contain unknown constants. We follow the same approach as Dennis and Walker [6] in that our results are based directly on the error in the computed iterates. The analysis in [6] is very general and uses several assumptions and constants that are difficult to interpret and understand even for the special case discussed therein (iterative refinement for linear systems of equations).

The residual contains information that is crucial for improving an approximate solution by Newton's method. Thus it should be computed as accurately as possible.

*Received by the editors August 4, 1999; accepted for publication (in revised form) by J. Varah October 5, 2000; published electronically February 23, 2001.
http://www.siam.org/journals/simax/22-4/35983.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). This work was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

Recently, mixed precision BLAS (XBLAS) routines have been proposed as a standard [2], where extended precision arithmetic is used internally to the BLAS and then the output is rounded to working precision. These new BLAS make the computation of the residual in mixed precision feasible for many problems, including the generalized eigenvalue problem considered here.

We first rework the forward error analysis of [6] for Newton's method in floating point arithmetic. We use different assumptions that are more appropriate when we have access to extended precision in computation of the residual and when we are using a possibly unstable linear system solver. The results we obtain are of more practical use than those in [6], [19], [28], [29] but consistent with them. We also estimate the limiting accuracy that can be obtained near a solution.

Next, we study the convergence of the norm of the residual, bounding the smallest norm. For many problems the backward error is a scaled residual norm, in which case we can use our results to bound the backward error. The idea of using iterative refinement to obtain a small backward error with a potentially unstable solution method has been investigated for linear systems by several authors, including Jankowski and Woźniakowski [18], Skeel [22], and Higham [17], and more recently for the algebraic Riccati equation by Ghavimi and Laub [11]. The idea does not seem to have been applied previously to the generalized eigenvalue problem.

In section 3 we apply our results to linear systems and to the standard and generalized eigenvalue problems. In section 4 we present numerical examples for the symmetric definite eigenvalue problem that motivated the whole analysis.

2. Newton's method in floating point arithmetic.

2.1. Basics and notation. We begin by describing our notation. Let $F : \mathbb{R}^m \mapsto \mathbb{R}^m$ be continuously differentiable on $\mathbb{R}^m$. We denote by J the Jacobian matrix $(\partial F_i/\partial v_j)$ of F and assume that J is Lipschitz continuous with constant β in $\mathbb{R}^m$, that is,

$$\|J(w) - J(v)\| \le \beta\|w - v\| \quad \text{for all} \quad v, w \in \mathbb{R}^m,$$

where $\|\cdot\|$ denotes any vector norm and the corresponding operator norm. We denote by $\kappa(J) = \|J\|\|J^{-1}\|$ the condition number of the matrix J. We attempt to solve the system of nonlinear equations $F(v) = 0$ by Newton's method:

$$J(v_i)(v_{i+1} - v_i) = -F(v_i), \quad i \ge 0, \tag{2.1}$$

where v_0 is given. We implement (2.1) as

$$\begin{aligned} &\text{Solve } J(v_i)d_i = -F(v_i),\\ &v_{i+1} = v_i + d_i. \end{aligned}$$

Newton's method is attractive because under appropriate conditions it converges rapidly from any sufficiently good initial guess. In particular, if the Jacobian is nonsingular at the solution, local quadratic convergence can be proved [5, Thm. 5.2.1]. The Kantorovich theorem yields a weaker bound on the convergence rate but makes no assumption on the nonsingularity of Jacobian at the solution [5, Thm. 5.3.1], [24].

We use hats to denote computed quantities. We work with the standard model of floating point arithmetic [16, section 2.3]

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1 + \delta), \qquad |\delta| \le u, \quad \operatorname{op} = +, -, *, /,$$

where u is the unit roundoff.

In floating point arithmetic, we have

$$\widehat{v}_{i+1} = \widehat{v}_i - (J(\widehat{v}_i) + E_i)^{-1}\,(F(\widehat{v}_i) + e_i) + \varepsilon_i, \tag{2.2}$$

where

- e_i is the error made when computing the residual $F(\widehat{v}_i)$,
- E_i is the error incurred in forming $J(\widehat{v}_i)$ and solving the linear system for d_i,
- ε_i is the error made when adding the correction $\widehat{d}_i$ to $\widehat{v}_i$.

We assume that $F(\widehat{v}_i)$ is computed in the possibly extended precision $\bar{u} \le u$ before rounding back to working precision u, and that $\widehat{d}_i, \widehat{v}_i$ are computed at precision u. Hence we assume that there exists a function ψ depending on $F, \widehat{v}_i, u$, and $\bar{u}$ such that

$$\|e_i\| \le u\|F(\widehat{v}_i)\| + \psi(F, \widehat{v}_i, u, \bar{u}). \tag{2.3}$$

Note that standard error analysis shows that $\|e_i\| \le u\|F(\widehat{v}_i)\|$ is the best we can obtain in practice for both mixed and fixed precision. Later, we will give an explicit formula for ψ in the case of linear systems and the generalized eigenvalue problem. We assume that the error E_i satisfies

$$\|E_i\| \le u\phi(F, \widehat{v}_i, n, u) \tag{2.4}$$

for some function ϕ that reflects both the instability of the linear solver and the error made when approximating or forming $J(\widehat{v}_i)$. In practice, we certainly have $\phi(F, \widehat{v}_i, n, u) \ge \|J(\widehat{v}_i)\|$. For the error ε_i we have

$$\|\varepsilon_i\| \le u(\|\widehat{v}_i\| + \|\widehat{d}_i\|).$$

We will make use of the constants

$$\gamma_n = \frac{cnu}{1 - cnu} \quad \text{and} \quad \bar{\gamma}_n = \frac{cn\bar{u}}{1 - cn\bar{u}}, \tag{2.5}$$

where c is a small integer constant.

2.2. Forward error. First we consider the change in error for a single step of an iteration of the form (2.2). For notational convenience we write $v = \widehat{v}_i$, $\bar{v} = \widehat{v}_{i+1}$, and

$$\bar{v} = v - (J + E)^{-1}(r + e) + \varepsilon, \tag{2.6}$$

where $r = F(v), J = J(v)$, and

$$\|E\| \le u\phi(F, v, n, u), \tag{2.7}$$

$$\|e\| \le u\|r\| + \psi(F, v, u, \bar{u}), \quad \|\varepsilon\| \le u(\|v\| + \|d\|),$$

with

$$d = (J + E)^{-1}(r + e). \tag{2.8}$$

We will often refer to the following lemma.

LEMMA 2.1 (see [5, Lem. 4.1.12]). *For any $v, w \in \mathbb{R}^m$,*

$$\|F(w) - F(v) - J(v)(w - v)\| \le \frac{\beta}{2}\|w - v\|^2. \tag{2.9}$$

THEOREM 2.2. *Assume that there is a v_* such that $F(v_*) = 0$, $J_* = J(v_*)$ is nonsingular, and*

$$\|J^{-1}E\| \le \nu < 1. \tag{2.10}$$

Then, for all v such that

$$\beta\|J_*^{-1}\|\|v - v_*\| \le \mu < 1, \tag{2.11}$$

$\bar{v}$ in (2.6) is well defined and

$$\|\bar{v} - v_*\| \le G\|v - v_*\| + g,$$

where

$$G = \frac{1}{1-\nu}\|J^{-1}E\| + \frac{(1+u)^2}{2(1-\mu)(1-\nu)}\beta\|J_*^{-1}\|\|v - v_*\| + \frac{u(2+u)}{(1-\mu)(1-\nu)}\kappa(J_*) + u$$

and

$$g = \frac{1+u}{(1-\mu)(1-\nu)}\|J_*^{-1}\|\psi(F, v, u, \bar{u}) + u\|v_*\|.$$

Proof. From assumption (2.11) and the Lipschitz property of J we have

$$\|J_*^{-1}(J - J_*)\| \le \beta\|J_*^{-1}\|\|v - v_*\| \le \mu < 1. \tag{2.12}$$

From the identity

$$J = J_*(I + J_*^{-1}(J - J_*)) \tag{2.13}$$

it then follows that J is nonsingular with inverse given by

$$J^{-1} = (I + J_*^{-1}(J - J_*))^{-1}J_*^{-1}$$

and with

$$\|J^{-1}\| \le \frac{\|J_*^{-1}\|}{1 - \|J_*^{-1}(J - J_*)\|} \le \frac{1}{1-\mu}\|J_*^{-1}\|. \tag{2.14}$$

Similarly, assumption (2.10) guarantees that $J + E$ is nonsingular and that, using (2.14),

$$\|(J+E)^{-1}\| \le \frac{\|J^{-1}\|}{1 - \|J^{-1}E\|} \le \frac{1}{(1-\mu)(1-\nu)}\|J_*^{-1}\|. \tag{2.15}$$

Since $(J+E)^{-1}$ exists, $\bar{v}$ in (2.6) is well defined. We have

$$\begin{aligned}\bar{v} - v_* &= v - v_* - (J+E)^{-1}(r+e) + \varepsilon \\ &= (I - (J+E)^{-1}J)(v - v_*) - (J+E)^{-1}(r - J(v - v_*) + e) + \varepsilon,\end{aligned}$$

which gives

$$\|\bar{v} - v_*\| \le \|I - (J+E)^{-1}J\|\|v - v_*\| + \|(J+E)^{-1}\|(\|r - J(v - v_*)\| + \|e\|) + \|\varepsilon\|.$$

From

$$I - (J+E)^{-1}J = (J+E)^{-1}E = (I + J^{-1}E)^{-1}J^{-1}E$$

it follows that

$$\|I - (J+E)^{-1}J\| \le \frac{1}{1-\nu}\|J^{-1}E\|.$$

From Lemma 2.1,

$$\|r - J(v-v_*)\| \le \frac{\beta}{2}\|v-v_*\|^2 \quad \text{and} \quad \|r - J_*(v-v_*)\| \le \frac{\beta}{2}\|v-v_*\|^2,$$

so that

$$\|r\| \le \|r - J_*(v-v_*)\| + \|J_*(v-v_*)\| \le \frac{\beta}{2}\|v-v_*\|^2 + \|J_*\|\|v-v_*\| \tag{2.16}$$

and hence

$$\|e\| \le u\left(\frac{\beta}{2}\|v-v_*\|^2 + \|J_*\|\|v-v_*\|\right) + \psi(F,v,u,\bar{u}).$$

We have

$$\|\varepsilon\| \le u(\|v-v_*\| + \|v_*\| + \|d\|)$$

with

$$\begin{aligned}\|d\| &\le \|(J+E)^{-1}\|(\|r\| + \|e\|) \\ &\le \|(J+E)^{-1}\|\big((1+u)\|r\| + \psi(F,v,u,\bar{u})\big) \\ &\le \frac{1}{(1-\mu)(1-\nu)}\|J_*^{-1}\|\left[(1+u)\left(\frac{\beta}{2}\|v-v_*\| + \|J_*\|\right)\|v-v_*\| \right. \\ &\qquad\qquad \left. + \psi(F,v,u,\bar{u})\right],\end{aligned} \tag{2.17}$$

using (2.15) and (2.16). Hence,

$$\|\bar{v} - v_*\| \le G\|v-v_*\| + g,$$

where G and g are given in the statement of the theorem. ∎

Assumptions (2.10) and (2.11) are necessary for $\bar{v}$ in (2.6) to be defined. Assumption (2.10) is a condition on the stability of the linear system solver and the accuracy of the Jacobian.

In exact arithmetic we have $u = \psi(F,v,u,\bar{u}) = \nu = 0$ and $E = 0$. Then, for $\mu \le 1/2$, Theorem 2.2 reduces to the local quadratic convergence theorem for Newton's method [5, Thm. 5.2.1] applied to a single step.

Clearly, for $\mu \le \frac{1}{8}$, $\nu \le \frac{1}{8}$, if J_* is not too ill conditioned, say, $u\kappa(J_*) \le \frac{1}{8}$, then we have $G \le \frac{1}{2}$. Thus the error contracts unless $g \gtrsim \|v - v_*\|$. Hence, the best limiting normwise accuracy we can guarantee is

$$\frac{g}{\|v_*\|} = \frac{1+u}{(1-\mu)(1-\nu)}\frac{\|J_*^{-1}\|}{\|v_*\|}\psi(F,v,u,\bar{u}) + u,$$

which depends on the accuracy with which the residual is computed. If $\|J_*^{-1}\|\psi(F, v, u, \bar{u}) \leq cu\|v_*\|$ for some constant c, then we can expect to obtain a normwise relative error of order cu.

Note that the rate of convergence depends on the accuracy of the Jacobian and on the stability of the linear system solver, since G depends strongly on E, but the limiting accuracy is essentially independent of the solver (for $\nu < \frac{1}{8}$, say). Note also that G is independent of $\bar{u}$, which means that the rate of convergence is bounded independent of the precision used to compute the residual.

COROLLARY 2.3. *Assume that there is a v_* such that $F(v_*) = 0$ and $J_* = J(v_*)$ is nonsingular and satisfies*

$$u\kappa(J_*) \leq \frac{1}{8}. \tag{2.18}$$

Assume also that for ϕ in (2.4),

$$u\|J(\widehat{v}_i)^{-1}\|\phi(F, \widehat{v}_i, n, u) \leq \frac{1}{8} \quad \textit{for all } i. \tag{2.19}$$

Then, for all v_0 such that

$$\beta\|J_*^{-1}\|\|v_0 - v_*\| \leq \frac{1}{8}, \tag{2.20}$$

Newton's method in floating point arithmetic generates a sequence $\{\widehat{v}_{i+1}\}$ whose normwise relative error decreases until the first i for which

$$\frac{\|\widehat{v}_{i+1} - v_*\|}{\|v_*\|} \approx \frac{\|J_*^{-1}\|}{\|v_*\|}\psi(F, v_*, u, \bar{u}) + u. \tag{2.21}$$

Proof. For $i = 0$, the assumptions (2.10) and (2.11) hold with $\nu = \frac{1}{8}$ and $\mu = \frac{1}{8}$ and Theorem 2.2 applies to the first step. Using the values for μ, ν, and the bound (2.18), we find that $G < 1$ so the error contracts if (2.21) does not already hold. Thus, (2.20) is also satisfied with v_0 replaced by $\widehat{v}_1$. The result follows by induction. ∎

Example 1. To illustrate the corollary, we use Newton's method to compute a zero of the polynomial

$$F(v) = (v-1)^{10} - 10^{-8}.$$

At the solution $v_* = 1 - 10^{-0.8} \approx 0.8415$, $|J(v_*)^{-1}| \approx 1.6 \times 10^6$. To increase the rounding errors when computing the residual, we expand $(v-1)^{10}$ as

$$(v-1)^{10} = v^{10} - 10v^9 + 45v^8 - 120v^7 + 210v^6 - 252v^5 + 210v^4 - 120v^3 + 45v^2 - 10v + 1$$

and use this expression to evaluate $F(v)$. For $v \approx 1$ we have $\psi(F, v, u, \bar{u}) \approx 10^3\bar{u}$ (which is roughly the sum of the absolute values of the coefficients in the expansion of $(v-1)^{10}$). Corollary 2.3 predicts that if v_0 is not too far from v_*, the forward error decreases until $|\widehat{v}_{i+1} - v_*|/|v_*| \approx 10^9\bar{u} + u$.

We carried out some numerical experiments in MATLAB, for which the unit roundoff is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. We used the Symbolic Math Toolbox to evaluate $F(v)$ at precision $\bar{u}$. We tried both $\bar{u} = u$ and $\bar{u} = u^{3/2} \approx 3.3 \times 10^{-24}$.[1] The theory predicts

[1] In the BLAST document [2], the term "extended precision" is used for $\bar{u} \leq u^{3/2}$.

limiting accuracy $|\widehat{v}_{i+1} - v_*|/|v_*| \approx 10^{-7}$ if $\bar{u} = u$ and $|\widehat{v}_{i+1} - v_*|/|v_*| \approx 10^{-15}$ if $\bar{u} = u^{3/2}$. For both values of $\bar{u}$, we used two different starting values for v_0, one for which $|v_0 - v_*|/|v_*| > 10^9\bar{u} + u$ and the second one for which the forward error is smaller than the expected limiting accuracy. We plot the behavior of the normwise forward error for $\bar{u} = u$ and $\bar{u} = u^{3/2}$ in Figure 2.1. The results are as predicted by the theory. They also illustrate Wilkinson's remark [27, p. 55]:

> It is perhaps worth remarking that if we start with an approximation to a zero which is appreciably more accurate than the limiting accuracy ... a single iteration will usually spoil this very good approximation and produce one with an error which is typical of the limiting accuracy.

2.3. Residual. We now turn to bounding the residual for a single step of the form (2.6). As before, we write $r = F(v)$ and $J = J(v)$. Note that if $\widehat{v}_* = fl(v_*) = v_* + \Delta v_*$ with $\|\Delta v_*\| \le u\|v_*\|$, then Lemma 2.1 gives

$$F(\widehat{v}_*) = F(v_* + \Delta v_*) = J(v_*)\Delta v_* + \theta, \quad \text{where} \quad \|\theta\| \le \frac{\beta}{2}\|\widehat{v}_* - v_*\|^2.$$

Thus

$$\|F(\widehat{v}_*)\| \le u\|J(v_*)\|\|v_*\| + \frac{\beta}{2}u^2\|v_*\|^2$$

is the best bound we can hope to obtain for the norm of the residual.

THEOREM 2.4. *Assume that there is a v_* such that $F(v_*) = 0$, $J_* = J(v_*)$ is nonsingular, and*

$$\beta\|J_*^{-1}\|\|v - v_*\| \le \mu < 1, \tag{2.22}$$

$$u\|J^{-1}\|\phi(F, v, n, u) \le \nu < 1. \tag{2.23}$$

Let

$$\tau = \beta g\|J_*^{-1}\|,$$

where g is defined in Theorem 2.2. Then

$$\|F(\bar{v})\| \le H\|F(v)\| + h,$$

where

$$H = c_0\left[\mu + \tau + u\kappa(J_*)\right]$$

and

$$h = c_1\left(\mu + \tau + u\kappa(J_*)\right)\psi(F, v, u, \bar{u}) + c_2\left(\mu + \tau + 1\right)u\|J\|\|v\|,$$

with c_0, c_1, and c_2 constants of order 1.

Proof. We have

$$\|J^{-1}E\| \le u\|J^{-1}\|\phi(F, v, n, u) \le \nu < 1 \tag{2.24}$$

using (2.7) and (2.23). Thus, we can apply Theorem 2.2 to deduce that $\bar{v}$ is well defined. Let $\bar{r} = F(\bar{v})$, and define $w \in \mathbb{R}^m$ by $w = \bar{r} - r - J(\bar{v} - v)$. Note that from

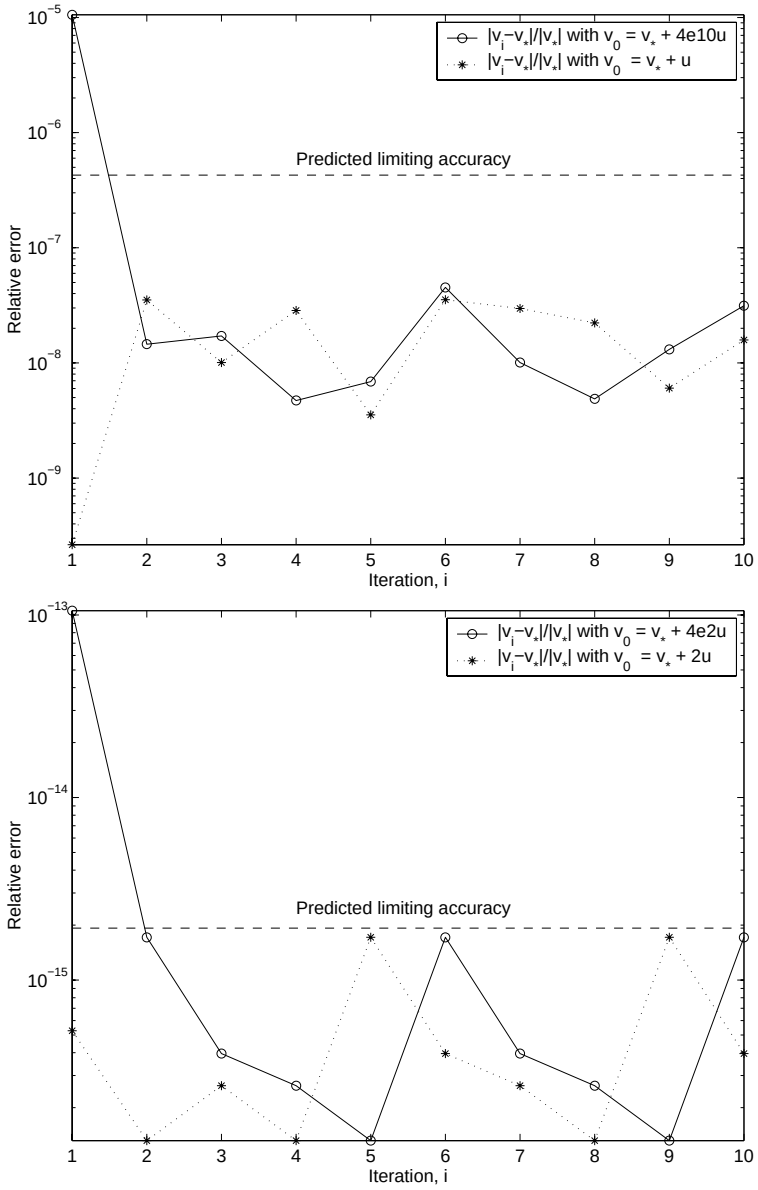


FIG. 2.1. *Behavior of the forward error for $\bar{u} = u$ (top) and $\bar{u} = u^{3/2}$ (bottom).*

(2.6) and (2.8) $\bar{v} - v = -d + \varepsilon$ and $Jd = r + e - Ed$, so that $\bar{r} = r + J(-d + \varepsilon) + w = -e + Ed + J\varepsilon + w$, which yields

$$\begin{aligned} \|\bar{r}\| &\le \|e\| + \|E\|\|d\| + \|J\|\|\varepsilon\| + \|w\| \\ &\le u\|r\| + \psi(F, v, u, \bar{u}) + u\|d\|(\phi(F, v, n, u) + \|J\|) + u\|J\|\|v\| + \|w\|. \end{aligned} \tag{2.25}$$

From (2.12) and (2.13) it follows that

$$\|J\| \le (1 + \mu)\|J_*\|. \tag{2.26}$$

Using (2.17) and (2.24), we have

$$\|d\| \le \|(J+E)^{-1}\|(\|r\|+\|e\|) \le \frac{1}{1-\nu}\|J^{-1}\|\left((1+u)\|r\|+\psi(F,v,u,\bar{u})\right), \tag{2.27}$$

which gives, using (2.14) and (2.26),

$$\begin{aligned} u\|d\|(\phi(F,v,n,u)+\|J\|) \le{}& \frac{1+u}{1-\nu}\left\{u\|J^{-1}\|\phi(F,v,n,u)+\frac{1+\mu}{1-\mu}u\kappa(J_*)\right\}\|r\| \\ &+\frac{1}{1-\nu}\left\{u\|J^{-1}\|\phi(F,v,n,u)+\frac{1+\mu}{1-\mu}u\kappa(J_*)\right\}\psi(F,v,u,\bar{u}). \end{aligned} \tag{2.28}$$

From Lemma 2.1 we have

$$\|w\| \le \frac{\beta}{2}\|\bar{v}-v\|^2. \tag{2.29}$$

First, from (2.6), (2.8), (2.27), and (2.14)

$$\begin{aligned} \|\bar{v}-v\| &\le (1+u)\|d\|+u\|v\| \\ &\le \|J_*^{-1}\|\left(\frac{(1+u)^2}{(1-\mu)(1-\nu)}\|r\|+\frac{1+u}{(1-\mu)(1-\nu)}\psi(F,v,u,\bar{u})\right)+u\|v\|. \end{aligned} \tag{2.30}$$

Second, from the triangle inequality and Theorem 2.2 we have

$$\|\bar{v}-v\| \le (G+1)\|v-v_*\|+g. \tag{2.31}$$

Substituting the product of (2.30) and (2.31) into (2.29) yields

$$\begin{aligned} \|w\| \le{}& \frac{(1+u)^2(G+1)}{2(1-\mu)(1-\nu)}\beta\|J_*^{-1}\|\|v-v_*\|\|r\|+\frac{(1+u)^2}{2(1-\mu)(1-\nu)}\beta\|J_*^{-1}\|g\|r\| \\ &+\frac{(1+u)(G+1)}{2(1-\mu)(1-\nu)}\beta\|J_*^{-1}\|\|v-v_*\|\psi(F,v,u,\bar{u}) \\ &+\frac{(1+u)}{2(1-\mu)(1-\nu)}\beta\|J_*^{-1}\|g\psi(F,v,u,\bar{u}) \\ &+\frac{(G+1)}{2(1-\mu)}\beta\|J_*^{-1}\|\|v-v_*\|u\|J\|\|v\|+\frac{1}{2(1-\mu)}\beta g\|J_*^{-1}\|u\|J\|\|v\|, \end{aligned} \tag{2.32}$$

where the penultimate and last terms on the right-hand side of the inequality are obtained using $\|J\|\|J^{-1}\| \ge 1$ and (2.14). Substituting (2.28) and (2.32) into (2.25) yields

$$\|\bar{r}\| \le H\|r\|+h,$$

with H and h as in the statement of the theorem. ∎

The theorem shows that if the problem is not too ill conditioned, the solver is not too unstable, the approximation of the Jacobian is accurate enough, and v is sufficiently close to the solution, then the norm of the residual reduces after one step of Newton's method in floating point arithmetic. Note that H does not depend on $\bar{u}$ so that, as for the forward error analysis, the use of extended precision for computing the residual has no effect on the rate of convergence of Newton's method. With a careful analysis of the constants in Theorem 2.4 we can derive the following corollary.

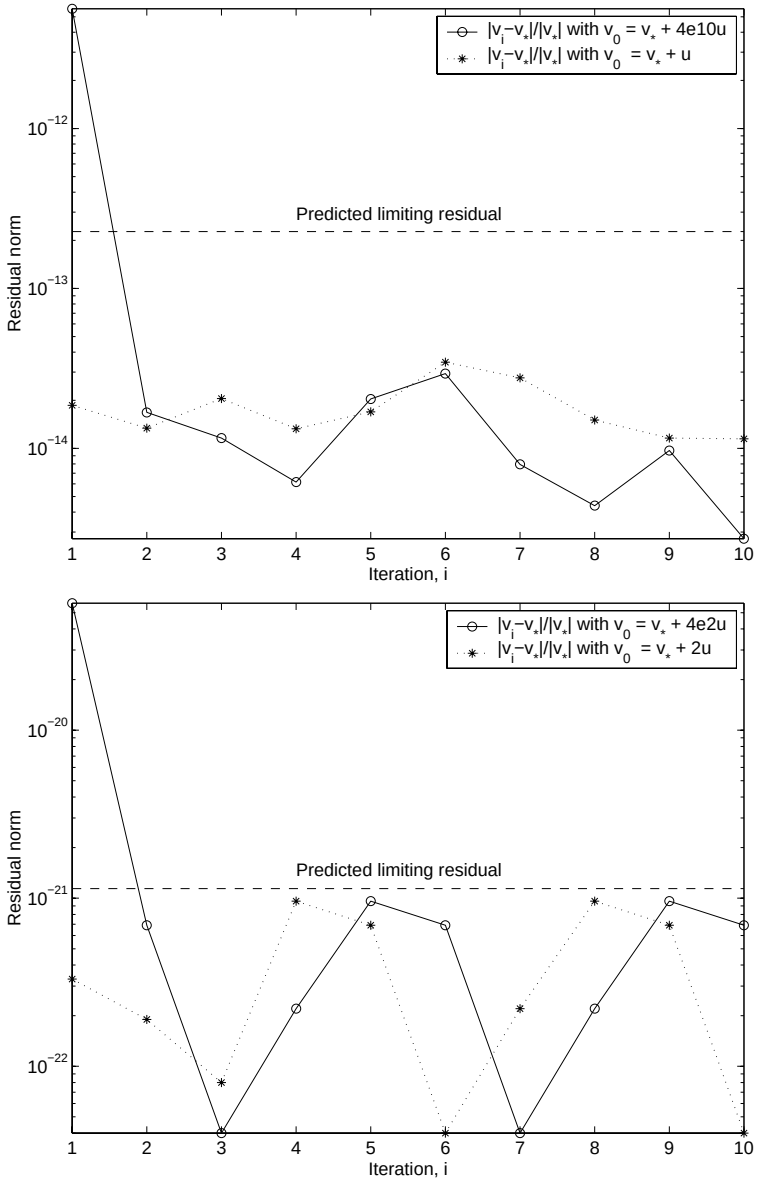


FIG. 2.2. *Behavior of the norm of the residual for $\bar{u} = u$ (top) and for $\bar{u} = u^{3/2}$ (bottom).*

COROLLARY 2.5. *Assume that there is a v_* such that $F(v_*) = 0$, $J_* = J(v_*)$ is nonsingular, and*

$$u\kappa(J_*) < 1/8. \tag{2.33}$$

Assume also that for ϕ in (2.4)

$$u\|J(\widehat{v}_i)^{-1}\|\phi(F, \widehat{v}_i, n, u) < \frac{1}{8} \quad \textit{for all } i \tag{2.34}$$

and that the limiting accuracy $g \approx \|J_*^{-1}\|\psi(F, v_*, u, \bar{u}) + u\|v_*\|$ satisfies $\beta g\|J_*^{-1}\| <$

1/8. *Then, for all* v_0 *such that* $\beta\|J_*^{-1}\|\|v_0 - v_*\| < 1/8$, *the sequence* $\{F(\widehat{v}_i)\}$ *of residual norms generated by Newton's method in floating point arithmetic decreases until*

$$\|F(\widehat{v}_{i+1})\| \approx \psi(F, \widehat{v}_i, u, \bar{u}) + u\|J(\widehat{v}_i)\|\|\widehat{v}_i\|. \tag{2.35}$$

Note that the second term in (2.35) is independent of the accuracy with which the residual is computed.

We consider again Example 1, for which $\psi(F, \widehat{v}_i, u, \bar{u}) \approx 10^3\bar{u}$ and $u\|J(v_*)\|\|v_*\| \approx 10^{-7}u$. As before, we tried both $\bar{u} = u$ and $\bar{u} = u^{3/2} \approx 3.3 \times 10^{-24}$. The theory predicts that

$$\|F(\widehat{v}_i)\| \lesssim \begin{cases} 10^{-13} & \text{if} \quad \bar{u} = u, \\ 10^{-21} & \text{if} \quad \bar{u} = u^{3/2}. \end{cases}$$

We used the same starting values as before. We plot the behavior of $|F(\widehat{v}_i)|$ for $\bar{u} = u$ and $\bar{u} = u^{3/2}$ in Figure 2.2. The results agree well with the predictions.

3. Applications. In this section, we consider several applications. For each of them, we define F and the function ψ and apply our results. We are particularly interested in the effect of mixed precision versus fixed precision for the computation of the residual. The proposed mixed precision BLAS routines (XBLAS) [2] make possible the use of mixed precision in a portable manner.

3.1. Linear systems. We consider the linear system $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is nonsingular and $b \in \mathbb{R}^n$. Iterative refinement for a computed solution $\widehat{x}$ is simple to describe: compute the residual $r = b - A\widehat{x}$, solve the system $Ad = r$ for the correction d, and form the updated solution $y = \widehat{x} + d$. If necessary, repeat the process with $\widehat{x}$ replaced by y. This process is equivalent to Newton's method with $F(x) = b - Ax$ for which $J(x) = A$ and thus $\beta = 0$.

If the residual $r = F(\widehat{x})$ is computed with the XBLAS routine `GEMV_X` at precision $\bar{u}$, then for ψ in (2.3) we can take

$$\psi(F, \widehat{x}, u, \bar{u}) = \bar{\gamma}_n(\|A\|\|\widehat{x}\| + \|b\|),$$

where $\bar{\gamma}_n$ is defined in (2.5). Corollary 2.3 then yields the following result.

COROLLARY 3.1. *If* $u\kappa(A)$ *is sufficiently less than* 1 *and if the linear system solver is not too unstable, then iterative refinement reduces the relative forward error until*

$$\frac{\|\widehat{x}_i - x\|}{\|x\|} \approx u + \kappa(A)\bar{\gamma}_n.$$

If $\bar{u} = u^2$, *then the relative error is of order* u *provided* $n\kappa(A)u \le 1$.

A backward error of an approximate solution $\widehat{x}$ is a measure of the smallest perturbations ΔA and Δb such that $(A+\Delta A)\widehat{x} = b+\Delta b$. The most popular definition of the normwise backward error is

$$\eta(\widehat{x}) = \min\{\varepsilon : (A + \Delta A)\widehat{x} = b + \Delta b,\ \|\Delta A\| \le \varepsilon\|A\|,\ \|\Delta b\| \le \varepsilon\|b\|\}.$$

It can be shown [21] that

$$\eta(\widehat{x}) = \frac{\|r\|}{\|A\|\|\widehat{x}\| + \|b\|}.$$

Corollary 2.5 thus yields the following result.

COROLLARY 3.2. *Let iterative refinement be applied to the nonsingular linear system $Ax = b$ of order n with $u\kappa(A) < 1/8$ and using a solver satisfying $u\|A^{-1}\|\phi(A, b, n, u) \leq 1/8$. Then the norm of the residual decreases until*

$$\|\widehat{r}_i\| \approx \max(\bar{\gamma}_n, u)(\|A\|\|\widehat{x}\| + \|b\|),$$

so that iterative refinement yields a small normwise backward error $\eta(\widehat{x}) \approx \max(\bar{\gamma}_n, u)$.

Corollaries 3.1 and 3.2 are standard normwise results in the literature [17], [18], [20], [22], [27]. They show that we do not lose anything by using our general analysis.

3.2. Generalized eigenvalue problem. Newton's method and its variants have been considered for improving the accuracy of computed eigenvalues and eigenvectors for the standard eigenvalue problem [10], [7], [8], [23], the singular value problem [9], and refining estimates of invariant subspaces [4], [10]. The error analysis in [10] applies to the standard eigenvalue problem $Ax = \lambda x$ and requires that the problem be scaled ($\|A\| = 1$), that the residual be computed in extended precision, and that the linear solver be stable. A lengthy analysis leads to the conclusion that if the problem is not too ill conditioned and the initial guess is good enough, then their refinement procedure yields a relative error of the order of the working precision.

Here, we consider the generalized eigenvalue problem (GEP)

$$Ax = \lambda Bx \quad \text{with} \quad e_s^T x = 1 \quad \text{for some fixed } s, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times n}$. Newton-based refinement algorithms for this problem have been proposed [7], [23] but no error analysis has been done.

Define $F : \mathbb{R}^{n+1} \mapsto \mathbb{R}^{n+1}$ by

$$F\left(\begin{bmatrix} x \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} (A - \lambda B)x \\ \alpha e_s^T x - \alpha \end{bmatrix}, \tag{3.2}$$

where $\alpha = \max(\|A\|, \|B\|)$. Then (3.1) can be stated as finding the zeros of $F(v)$, where $v = [x^T, \lambda]^T$. The function F is continuously differentiable in $\mathbb{R}^{n+1}$ with Jacobian

$$J(v) = \begin{bmatrix} A - \lambda B & -Bx \\ \alpha e_s^T & 0 \end{bmatrix}. \tag{3.3}$$

The scalar α is introduced to make F and J scale linearly when A and B are multiplied by a scalar. For all $v, w \in \mathbb{R}^{n+1}$ and any absolute vector norm we have

$$\|J(w) - J(v)\| \leq 2\|B\|\|w - v\|$$

so that J is Lipschitz continuous in $\mathbb{R}^{n+1}$ with constant $\beta = 2\|B\|$.

The next lemma concerns the singularity of J at a zero of F. This result is more general than the one given in [23, p. 120] as it applies to the generalized eigenvalue problem rather than the standard eigenvalue problem and no assumption is made on the nonsingularity of B.

LEMMA 3.3. *Let $v_* = [x_*^T, \lambda_*]^T$ be a zero of F as defined by (3.2) with λ finite. Then $J(v_*)$ is singular if and only if λ_* is a multiple eigenvalue of (A, B).*

Proof. Suppose that $J(v_*)$ is singular. Using the formula (see [13])

$$\det\left(\begin{bmatrix} M & u \\ v_*^T & \mu \end{bmatrix}\right) = \mu \det(M) - v_*^T M^A u,$$

where M^A is the adjugate (or adjoint) of M, we obtain

$$0 = \det(J(v_*)) = \alpha e_s^T (A - \lambda_* B)^A B x_*. \tag{3.4}$$

The adjugate has the property that

$$M^A M = \det(M) I.$$

Define $y^T = e_s^T (A - \lambda_* B)^A$. Then

$$y^T (A - \lambda_* B) = e_s^T \det(A - \lambda_* B) I = 0,$$

because λ_* is an eigenvalue of (A, B). Thus y is a left eigenvector corresponding to λ_*. Using (3.4),

$$y^T B x_* = e_s^T (A - \lambda_* B)^A B x_* = 0.$$

If λ_* were a simple eigenvalue, we would have $y^T B x_* \neq 0$ [1, Thm. 3.2]. So λ_* must be an eigenvalue of multiplicity at least two.

For the converse, suppose that λ_* is a multiple eigenvalue of (A, B). Then, there exists a left eigenvector y corresponding to λ_* that is B-orthogonal to x_*. We have

$$\begin{bmatrix} y^T & 0 \end{bmatrix} \begin{bmatrix} A - \lambda_* B & -B x_* \\ \alpha e_s^T & 0 \end{bmatrix} = 0,$$

which means that $J(v_*)$ is singular. ∎

In exact arithmetic, Theorem 2.2 applies with $E = 0$ and $\nu = u = 0$ so that for all v_0 such that $\|v_0 - v_*\| \leq 1/(4\|B\|\|J_*\|^{-1})$ the Newton iteration is well defined and converges quadratically to zero.

The residual $F(\widehat{v}_i)$ can be computed in mixed precision by the XBLAS routine `GE_SUM_MV`. Then we can take

$$\psi(F, v, u, \bar{u}) = \bar{\gamma}_n(\|A\| + |\lambda|\|B\|)\|x\|. \tag{3.5}$$

COROLLARY 3.4. *Let λ_* be a simple eigenvalue of (A, B), and let x_* be the corresponding eigenvector normalized such that $\|x_*\|_\infty = |x_{*s}| = 1$. Assume that J in* (3.3) *is not too ill conditioned, the linear system solver is not too unstable, and (x_0, λ_0) is a sufficiently good approximation to (x_*, λ_*) so that assumptions* (2.18)–(2.20) *with $\beta = 2\|B\|_\infty$ are satisfied. Then Newton's method for* (3.2) *in floating point arithmetic is well defined and the limiting forward error is bounded by*

$$\frac{\|(\widehat{x}_i^T, \widehat{\lambda}_i) - (x_*^T, \lambda_*)\|_\infty}{\|(x_*^T, \lambda_*)\|_\infty} \lesssim \bar{\gamma}_n \|J(v_*)^{-1}\|_\infty \max(\|A\|_\infty, \|B\|_\infty) + u.$$

If $\bar{u} = u^2$, then

$$\frac{\|(\widehat{x}_i^T, \widehat{\lambda}_i) - (x_*^T, \lambda_*)\|_\infty}{\|(x_*^T, \lambda_*)\|_\infty} \lesssim \gamma_n.$$

Proof. We apply Corollary 2.3 using (3.5) for $\psi(F, v, u, \bar{u})$. We have

$$\begin{aligned} \frac{\|J(v_*)^{-1}\|_\infty}{\|v_*\|_\infty} \psi(F, v_*, u, \bar{u}) &= \frac{\|J(v_*)^{-1}\|_\infty}{\|v_*\|_\infty} \bar{\gamma}_n(\|A\|_\infty + |\lambda_*|\|B\|_\infty)\|x_*\|_\infty \\ &\leq \bar{\gamma}_n \|J(v_*)^{-1}\|_\infty \max(\|A\|_\infty, \|B\|_\infty) \frac{(1 + |\lambda_*|)}{\max(1, |\lambda_*|)} \\ &\leq 2\bar{\gamma}_n \|J(v_*)^{-1}\|_\infty \max(\|A\|_\infty, \|B\|_\infty). \end{aligned}$$

Since $J(v_*)_{n+1,s} = \alpha$, we have $\|J(v_*)\|_\infty \geq \max(\|A\|_\infty, \|B\|_\infty)$. From (2.18), we have $u\kappa(J(v_*)) < 1$ and if $\bar{\gamma}_n \approx nu^2$, then $\bar{\gamma}_n \|J(v_*)^{-1}\|_\infty \lesssim nu \max(\|A\|_\infty, \|B\|_\infty)^{-1}$, which proves the last part of the corollary. ∎

Our result is consistent with the one of Dongarra, Moler, and Wilkinson [10] concerning the standard eigenvalue problem. They showed that their iterative refinement procedure, which is a recasting of Newton's method, yields a forward error of the order of the working precision assuming that $\|A\|_\infty = 1$ and that the residual is computed at precision $\bar{u} = u^2$.

The normwise backward error for an approximate eigenpair $(\widehat{x}, \widehat{\lambda})$ is defined by

$$\eta(\widehat{x}, \widehat{\lambda}) = \min\{\varepsilon : (A + \Delta A)\widehat{x} = \widehat{\lambda}(B + \Delta B)\widehat{x}, \|\Delta A\| \leq \varepsilon\|A\|, \|\Delta B\| \leq \varepsilon\|B\|\},$$

and it can be shown [14], [25] that

$$\eta(\widehat{x}, \widehat{\lambda}) = \frac{\|r\|}{(\|A\| + |\widehat{\lambda}|\|B\|)\|\widehat{x}\|},$$

where $r = A\widehat{x} - \widehat{\lambda}B\widehat{x}$.

COROLLARY 3.5. *Under the same assumptions as in Corollary* 3.4*, Newton's method for* (3.2) *in floating point arithmetic yields a backward error for the* ∞*-norm bounded by*

$$\eta_\infty(\widehat{x}_i, \widehat{\lambda}_i) \lesssim \bar{\gamma}_n + u(3 + |\lambda|) \max\left(\frac{\|A\|_\infty}{\|B\|_\infty}, \frac{\|B\|_\infty}{\|A\|_\infty}\right).$$

Proof. We assume $\|\widehat{x}_i\|_\infty \approx 1$. We have $\psi(F, \widehat{v}_i, u, \bar{u}) \approx \bar{\gamma}_n(\|A\|_\infty + |\widehat{\lambda}_i|\|B\|_\infty)$ and

$$\|\widehat{v}_i\|_\infty \lesssim 1 + |\widehat{\lambda}_i|, \quad \|J(\widehat{v}_i)\|_\infty \lesssim (3 + |\widehat{\lambda}_i|) \max(\|A\|_\infty, \|B\|_\infty),$$

and $(\|A\|_\infty + |\widehat{\lambda}_i|\|B\|_\infty)\|\widehat{x}_i\|_\infty \gtrsim \min(\|A\|_\infty, \|B\|_\infty)(1 + |\widehat{\lambda}_i|)$. Then applying Corollary 2.5 yields the result. ∎

The corollary shows that if $|\lambda| \max(\|A\|_\infty/\|B\|_\infty, \|B\|_\infty/\|A\|_\infty)$ is large, then we cannot guarantee a small backward error. In numerical experiments, we have found that the backward error is small independent of the size of $|\lambda| \max(\|A\|_\infty/\|B\|_\infty, \|B\|_\infty/\|A\|_\infty)$, but we have not been able to prove that this must always be the case.

Note that for the standard eigenvalue problem, $|\lambda_*| \leq 1$ if $\|A\|_\infty = 1$, as was assumed in [10]. Then the eigenpairs refined by Newton's method have a small backward error.

For the GEP, if the problem is scaled and replaced by $\widetilde{A}x = \widetilde{\lambda}Bx$ with $\widetilde{A}$ and $\widetilde{\lambda}$ such that $\|\widetilde{A}\|_\infty = \alpha\|A\|_\infty = \|B\|_\infty$ and $\widetilde{\lambda} = \alpha\lambda$, then, for this problem, the backward error depends only on the size of $|\widetilde{\lambda}|$. A small $|\widetilde{\lambda}|$ ensures a small backward error. If $|\widetilde{\lambda}|$ is large, then we can consider the problem $Bx = \widetilde{\mu}\widetilde{A}x$ for which $|\widetilde{\mu}|$ is small and Corollary 3.5 guarantees that iterative refinement will yield a small backward error.

4. Numerical experiments. We show how iterative refinement can be used to improve the stability of an unstable solver for the symmetric definite generalized eigenvalue problem $Ax = \lambda Bx$, with A symmetric and B symmetric positive definite.

All our tests have been performed with MATLAB for which the working precision is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. We approximate the eigenpairs using the Cholesky-QR method, which consists of the following.

1. Compute the Cholesky factorization $B = GG^T$.
2. Compute $C = G^{-1}AG^{-T}$.
3. Compute the eigendecomposition $W^TCW = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ using the symmetric QR algorithm.

The matrix $X = G^{-T}W$ is nonsingular and satisfies $X^TBX = I$ and $X^TAX = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. This algorithm can be unstable. The computed $\widehat{C}$ from step 2 satisfies [3]

$$\widehat{C} = C + \Delta C, \quad \|\Delta C\|_2 \le \gamma_{n^2}\|B^{-1}\|_2\|A\|_2,$$

so if B is ill conditioned, then $\|\Delta C\|_2/\|C\|_2$ can be large, even if the eigenvalue problem itself is well conditioned.

For problem (3.1), the Newton iteration (2.1) can be written as

$$(A - \lambda_i B)\Delta x_{i+1} - \Delta\lambda_{i+1}Bx_i = r_i, \quad e_s^T x_{i+1} = e_s^T x_i = 1, \tag{4.1}$$

where $\Delta x_{i+1} = x_{i+1} - x_i$ and $\Delta\lambda_{i+1} = \lambda_{i+1} - \lambda_i$. As in [10], [23] we note that $e_s^T x_0 = 1$ implies $e_s^T \Delta x_{i+1} = 0$ for $i \ge 0$, and thus the sth column of $A - \lambda_i B$ does not participate in the product with Δx_{i+1}. We can replace the sth column of $A - \lambda_i B$ by $-Bx_i$ and the component s of Δx_{i+1} by $\Delta\lambda_{i+1}$. We define

$$\delta_i = \Delta x_i + \Delta\lambda_i e_s \quad \text{and} \quad M_i = (A - \lambda_i B) - ((A - \lambda_i B)e_s + Bx_i)e_s^T.$$

Then we can rewrite (4.1) as

$$M_i\delta_{i+1} = r_i, \quad \lambda_{i+1} = \lambda_i + e_s^T\delta_{i+1}, \quad x_{i+1} = x_i + \delta_{i+1} - e_s^T\delta_{i+1}e_s. \tag{4.2}$$

Algorithm 4.1 is a straightforward implementation of iteration (4.2).

ALGORITHM 4.1. *Given A, B, and an approximate eigenpair (x, λ) with $\|x\|_\infty = x_s = 1$, this algorithm applies iterative refinement to λ and x:*

repeat until convergence
  $r = \lambda Bx - Ax$ (possibly extended precision used)
  Form M: the matrix $A - \lambda B$ with column s replaced by $-Bx$.
  Factor $PM = LU$ (LU factorization with partial pivoting)
  Solve $M\delta = r$ using the LU factors
  $\lambda = \lambda + \delta_s$; $\delta_s = 0$
  $x = x + \delta$
end

This algorithm is expensive as each iteration requires $O(n^3)$ flops for the factorization of M. If the eigenpairs are approximated by a Cholesky reduction of $A - \lambda B$, then a nonsingular matrix X such that $X^TAX = D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ and $X^TBX = I$ is available. Then

$$\begin{aligned} X^T r_i &= X^T M_i \delta_{i+1} \\ &= \left((D - \lambda_i I) - X^T((A - \lambda_i B)e_s + Bx_i)e_s^T X\right) X^{-1}\delta_{i+1}. \end{aligned} \tag{4.3}$$

Defining

$$D_{\lambda_i} = D - \lambda_i I, \quad v_i = X^T((A - \lambda_i B)e_s + Bx_i),$$

$$f = X^T e_s, \quad w_{i+1} = X^{-1}\delta_{i+1}, \quad g_i = X^T r_i,$$

(4.3) becomes

$$(D_{\lambda_i} - v_i f^T) w_{i+1} = g_i. \tag{4.4}$$

The matrix in (4.4) is a rank-one modification of a diagonal matrix. As D_{λ_i} is nearly singular when λ_i approaches the solution λ_*, we cannot use the Sherman–Morrison–Woodbury formula. However, we can define rotations $J_{n-1}, \ldots, J_1$ such that

$$J_1^T \ldots J_{n-1}^T v_i = \pm \|v_i\|_2 e_1,$$

where J_k is a rotation in the $(k, k+1)$ plane. Then $H = J_1^T \ldots J_{n-1}^T D_{\lambda_i}$ is upper Hessenberg, as is the matrix

$$J_1^T \ldots J_{n-1}^T (D_{\lambda_i} - v_i f^T) = H \pm \|v_i\|_2 e_1 f^T = H_1.$$

Using a QR factorization of H_1, the solution of (4.4) can be computed in $O(n^2)$ flops.

ALGORITHM 4.2. *Given A, B, X, and D such that $X^T A X = D$ and $X^T B X = I$ and an approximate eigenpair (x, λ) with $\|x\|_\infty = x_s = 1$, this algorithm applies iterative refinement to λ and x at a cost of $O(n^2)$ flops per iteration.*

repeat until convergence
  $r = \lambda B x - A x$ (possibly extended precision used)
  $D_\lambda = D - \lambda I$
  $d = -Bx - c_{\lambda s}$ where $c_{\lambda s}$ is the sth column of $A - \lambda B$
  $v = X^T d$; $f = X^T e_s$
  Compute Givens rotations J_k in the $(k, k+1)$ plane, such that
    $Q_1^T v := J_1^T \ldots J_{n-1}^T v = \|v\|_2 e_1$
  Compute orthogonal Q_2 such that
    $T = Q_2^T Q_1^T (D_\lambda + v f^T)$ is upper triangular
  $z = Q_2^T Q_1^T X^T r$
  Solve $Tw = z$ for w
  $\delta = Xw$
  $\lambda = \lambda + \delta_s$; $\delta_s = 0$
  $x = x + \delta$
end

When B is ill conditioned, the computed $\widehat{X}$ may be inaccurate, so that $\widehat{X}^T A \widehat{X} = D + \Delta D$, $\widehat{X}^T B \widehat{X} = I + \Delta I$, with possibly large $\|\Delta D\|$ and $\|\Delta I\|$. Then the procedure used in Algorithm 4.2 to solve $M\delta = r$ may be unstable: δ is the exact solution of $(M + \Delta M)\delta = r$ with a possibly large $\|\Delta M\|$. However, the theory shows that allowing some instability in the solver and inaccurate evaluation of the Jacobian (assumptions (2.19) and (2.23)) may affect the rate of convergence of the Newton process but not the limiting accuracy and backward error.

We use the hat notation $(\widehat{x}, \widehat{\lambda})$ for approximate eigenpairs obtained with the Cholesky-QR method and the tilde notation $(\widetilde{x}, \widetilde{\lambda})$ for the refined eigenpairs obtained after a few iterations with Algorithm 4.1 or 4.2 starting with $(\widehat{x}, \widehat{\lambda})$ as initial guess. We need to define several quantities:

$$E_{rel}(\widehat{x}, \widehat{\lambda}) = \|(x, \lambda) - (\widehat{x}, \widehat{\lambda})\|_\infty / \|(x, \lambda)\|_\infty$$

is the relative forward error;

$$\mathrm{cond}(\lambda) = (\|A\|_\infty + |\lambda| \|B\|_\infty) \|x\|_\infty^2 / (|\lambda|\ |y^T B x|)$$

TABLE 4.1
Relative errors, condition numbers, and backward error for Example 1.

	λ_i	$E_{rel}(\widehat{x}_i,\widehat{\lambda}_i)$	$\mathrm{cond}(\lambda_i)$	$\eta(\widehat{x}_i,\widehat{\lambda}_i)$
1	-0.62	6e-5	41	4e-6
2	1.63	6e-5	120	2e-6
3	$9e17$	9e-5	$6e18$	2e-20

TABLE 4.2
Backward error and relative error for the two smallest eigenpairs of Example 1.

			Algorithm 4.1			Algorithm 4.2		
λ_i	η^{est}	E_{rel}^{est}	it	$\eta(\widetilde{x}_i,\widetilde{\lambda}_i)$	$E_{rel}(\widetilde{x}_i,\widetilde{\lambda}_i)$	it	$\eta(\widetilde{x}_i,\widetilde{\lambda}_i)$	$E_{rel}(\widetilde{x}_i,\widetilde{\lambda}_i)$
-0.62	1e-16	1e-14	3	2e-17	2e-16	4	6e-17	4e-16
1.63	1e-16	1e-14	3	3e-17	4e-16	4	4e-17	7e-16

is the condition number of the eigenvalue λ, where y is a left eigenvector corresponding to λ [14];

$$\eta(\widehat{x},\widehat{\lambda}) = \|A\widehat{x} - \widehat{\lambda}B\widehat{x}\|_\infty / \big((\|A\|_\infty + |\widehat{\lambda}|\|B\|_\infty)\|\widehat{x}\|_\infty\big)$$

is the backward error of the approximate eigenpair $(\widehat{x},\widehat{\lambda})$;

$$E_{rel}^{est} = \|J^{-1}\|_\infty \bar{u}(\|A\|_\infty + |\lambda|\|B\|_\infty)\|x\|_\infty / \|(x^T,\lambda)\|_\infty + u$$

is an approximation of the theoretical bound (2.21) for the relative forward error, where the Jacobian matrix J is given by (3.3) and $\psi(F,v,u,\bar{u})$ is given by (3.5) with $\bar{\gamma}_n \approx \bar{u}$; and finally, η^{est} is the theoretical bound of the backward error for the refined eigenpair $(\widetilde{x},\widetilde{\lambda})$ from Corollary 3.5.

Example 1. First we consider

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}, \quad G = \begin{bmatrix} .001 & 0 & 0 \\ 1 & .001 & 0 \\ 2 & 1 & 0.001 \end{bmatrix},$$

and $B = GG^T$. This example is used in [12] to illustrate the instability of the Cholesky-QR method when B is ill conditioned. Results are displayed in Table 4.1. The two smallest eigenvalues have a small condition number, but their backward error is large because of the ill conditioning of B ($\kappa_\infty(B) = 7 \times 10^{18}$).

We refined the two smallest eigenvalues using Algorithm 4.1 and Algorithm 4.2 with the approximate eigenpairs as initial guess and the residual computed at working precision ($\bar{u} = u \approx 1.1 \times 10^{-16}$). We terminated the iteration when the norm of the correction stopped decreasing. The results are given in Table 4.2, where it is the number of iterations required for convergence. Algorithm 4.2 uses an unstable solver and therefore requires one more iteration. However, the accuracy and stability are unaffected by this unstable solver. Both algorithms produce refined eigenpairs with a small backward error and a relative error as predicted by the theory.

Example 2. We would like to test the sharpness of the residual bound in Corollary 2.5 and the backward error bound in Corollary 3.5. We consider an example with large $\|J_*\|$, a large ratio $\|A\|_\infty/\|B\|_\infty$, and large eigenvalues. We denote by M the Moler matrix from the Test Matrix Toolbox [15]:

$$m_{ij} = \begin{cases} i \ \text{ if } \ i = j, \\ \min(i,j) - 2 \ \text{ otherwise.} \end{cases}$$

TABLE 4.3
Estimated and computed residuals and backward errors for Example 2.

		Before refinement	From theory		After refinement		
λ_i	cond(λ_i)	$\eta(\widehat{x}_i, \widehat{\lambda}_i)$	$\|r^{est}\|$	η^{est}	$\|r\|$	$\eta(\widetilde{x}_i, \widetilde{\lambda}_i)$	it
7.1e5	2.0	1e-5	3.1e-4	9.1e-5	1.2e-10	5.2e-17	5
5.6e6	9.0	2e-6	1.1e-2	7.3e-4	4.7e-10	4.3e-17	4
2.0e7	29.2	9e-7	1.5e-1	2.6e-3	1.0e-9	2.9e-17	3
3.3e7	48.7	7e-7	4.3e-1	4.3e-3	1.6e-9	2.7e-17	5
4.3e7	62.9	2e-7	7.4e-1	5.6e-3	1.7e-9	2.2e-17	3

TABLE 4.4
Relative error for the computed and refined eigenpairs of Example 3 *using working and double precision in the computation of the residual.*

		Before refinement	After refinement			
			$\bar{u} = u$		$\bar{u} = u^2$	
λ_i	cond(λ_i)	$E_{rel}(\widehat{x}_i, \widehat{\lambda}_i)$	E_{rel}^{est}	$E_{rel}(\widetilde{x}_i, \widetilde{\lambda}_i)$	E_{rel}^{est}	$E_{rel}(\widetilde{x}_i, \widetilde{\lambda}_i)$
2.4e-7	1.8e6	1.3e-8	1.0e-11	2.0e-13	2.2e-16	1.1e-16
2.2e-5	2.0e4	2.1e-8	1.3e-11	7.3e-13	2.2e-16	2.2e-16
8.2e-4	5.3e2	1.0e-9	3.3e-13	1.8e-14	2.2e-16	1.1e-16
1.4e-2	4.0e1	6.9e-11	3.4e-14	2.0e-15	2.2e-16	1.1e-16
2.9e-2	4.6e0	4.3e-11	2.8e-14	5.6e-16	2.2e-16	1.1e-16
1.2e-1	1.5e1	2.6e-11	1.7e-14	5.6e-16	2.2e-16	1.1e-16
1.7e-1	7.4e0	3.6e-11	3.0e-14	1.3e-15	2.2e-16	1.1e-16
3.0e-1	1.1e1	3.0e-11	2.0e-13	2.2e-15	2.2e-16	5.6e-17
3.1e-1	1.2e1	3.4e-11	2.1e-13	7.8e-16	2.2e-16	5.6e-17
9.2e4	3.7e6	1.6e-16	1.5e-9	4.1e-12	2.2e-16	0.0e0

We took $n = 20$, $A = 10^6 I$, and $B = 10^{-2} M$ and computed the approximate eigenpairs using the Cholesky reduction. Instabilities are expected as $\kappa(B) = 2 \times 10^{13}$. All the eigenpairs have a large backward error and a small condition number except the largest one. We refined using Algorithm 4.1. Results for some eigenpairs are given in Table 4.3, where

$$\|r^{est}\| = \bar{u}(\|A\|_\infty + |\lambda|\|B\|_\infty)\|x\|_\infty + u\|J\|_\infty \|(x^T, \lambda)\|_\infty$$

is the theoretical bound (2.35) for the norm of the residual. This example corresponds to the "bad case" where $|\lambda| \max\left(\|A\|/\|B\|, \|B\|/\|A\|\right)$ is large, which explains why the theoretical estimates are so pessimistic. The estimates are sharp when the pair (A, B) is scaled such that $\|A\| = \|B\|$ and the eigenpair is refined on the reverse problem (B, A) if $|\widehat{\lambda}_i|$ is large. We have generated many pairs (A, B) with a large value of $\max\left(\|A\|/\|B\|, \|B\|/\|A\|\right)$ and large eigenvalues, for which the theory predicts a large backward error. For all of them, iterative refinement yields a small backward error as long as the initial guess is good enough for Newton's method to converge.

Example 3. We illustrate how using extended precision in computation of the residual yields a small relative error. Let A be the Prolate matrix of size $n = 10$ of the Test Matrix Toolbox [15], and let B be the Moler matrix. We used the Symbolic Math Toolbox of MATLAB to compute the exact eigenpairs of (A, B) and the Cholesky reduction method to approximate the eigenpairs. We give the results in Table 4.4. We refined using both working precision ($\bar{u} = u$) and double precision ($\bar{u} = u^2$) for the computation of the residual. For eigenpairs such that $E_{rel}(\widehat{x}_i, \widehat{\lambda}_i) > E_{rel}^{est}$, iterative refinement leads to $E_{rel}(\widetilde{x}_i, \widetilde{\lambda}_i) < E_{rel}^{est}$ after two iterations. For the largest eigenvalue, $E_{rel}(\widehat{x}_i, \widehat{\lambda}_i) \ll E_{rel}^{est}$ of $\bar{u} = u$, which means that the approximate eigenpair is appreciably more accurate than the limiting accuracy. In this case, one single step

of iterative refinement is enough to spoil the good initial approximation. If $\bar{u} = u^2$, all the eigenpairs are computed to high relative accuracy as expected from the theory (Corollary 3.4).

For further numerical examples of iterative refinement for the Cholesky-QR method, see [3].

5. Conclusions. We have analyzed Newton's method in floating point arithmetic, allowing for extended precision in computation of the residual, inaccurate evaluation of the Jacobian, and a possibly unstable solver. We estimated the limiting accuracy and the smallest residual norm. We showed that the accuracy with which the residual is computed affects the limiting accuracy. The limiting residual norm depends on two terms, one of them independent of the accuracy used in evaluating the residual.

We applied our results to iterative refinement for the generalized eigenvalue problem. We showed that high accuracy for the refined eigenpairs is guaranteed, under suitable assumptions, if twice the working precision is used for the computation of the residual. We also showed that if the pair (A, B) is well balanced ($\|A\| \approx \|B\|$), working precision in evaluating the residual is enough for iterative refinement to yield a small backward error.

Finally, we examined in detail how iterative refinement can be used to improve the forward and backward error of computed eigenpairs for the symmetric definite GEP. We used two refinement algorithms, one of them with an unstable solver. We confirmed that the unstable solver affects the convergence but not the limiting accuracy and backward error. In practice, the assumption that the pair (A, B) is well balanced does not seem to be necessary. We have not been able to generate an example for which iterative refinement fails to yield a small backward error for pairs (A, B) for which $\max(\|A\|/\|B\|, \|B\|/\|A\|)$ is large. This suggests that the bound of Corollary 3.5 is pessimistic. Deriving a sharper bound remains an open problem.

In future work, we plan to investigate iterative refinement for the quadratic eigenvalue problem, for which there are no proven backward stable algorithms [25].

Acknowledgments. I thank the referees for valuable suggestions that improved the paper.

REFERENCES

[1] A. L. Andrew, K.-W. E. Chu, and P. Lancaster, *Derivatives of eigenvalues and eigenvectors of matrix functions*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 903–926.

[2] *BLAS Technical Forum Standard*, International Journal of High Performance Computing Applications, to appear. Available online at http://www.netlib.org/blas/blast-forum/.

[3] P. I. Davies, N. J. Higham, and F. Tisseur, *Analysis of the Cholesky Method with Iterative Refinement for Solving the Symmetric Definite Generalized Eigenproblem*, Numerical Analysis Report No. 360, Manchester Centre for Computational Mathematics, Manchester, UK, 2000.

[4] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.

[5] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] J. E. Dennis, Jr. and H. F. Walker, *Inaccuracy in quasi-Newton methods: Local improvement theorems*, Math. Programming Stud., 22 (1984), pp. 70–85.

[7] J. J. Dongarra, *Improving the accuracy of computed matrix eigenvalues*, Preprint ANL-80-84, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1980.

[8] J. J. Dongarra, *Algorithm* 589 *SICEDR: A FORTRAN subroutine for improving the accuracy of computed matrix eigenvalues*, ACM Trans. Math. Software, 8 (1982), pp. 371–375.
[9] J. J. Dongarra, *Improving the accuracy of computed singular values*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 712–719.
[10] J. J. Dongarra, C. B. Moler, and J. H. Wilkinson, *Improving the accuracy of computed eigenvalues and eigenvectors*, SIAM J. Numer. Anal., 20 (1983), pp. 23–45.
[11] A. R. Ghavimi and A. J. Laub, *Backward error, sensitivity, and refinement of computed solutions of algebraic Riccati equations*, Numer. Linear Algebra Appl., 2 (1995), pp. 29–49.
[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[13] H. V. Henderson and S. R. Searle, *On deriving the inverse of a sum of matrices*, SIAM Rev., 23 (1981), pp. 53–60.
[14] D. J. Higham and N. J. Higham, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 493–512.
[15] N. J. Higham, *The Test Matrix Toolbox for* Matlab *(version* 3.0*)*, Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, UK, 1995.
[16] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[17] N. J. Higham, *Iterative refinement for linear systems and LAPACK*, IMA J. Numer. Anal., 17 (1997), pp. 495–509.
[18] M. Jankowski and H. Woźniakowski, *Iterative refinement implies numerical stability*, BIT, 17 (1977), pp. 303–311.
[19] P. Lancaster, *Error analysis for the Newton-Raphson method*, Numer. Math., 9 (1966), pp. 55–68.
[20] C. B. Moler, *Iterative refinement in floating point*, J. Assoc. Comput. Mach., 14 (1967), pp. 316–321.
[21] J. L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. Assoc. Comput. Mach., 14 (1967), pp. 543–548.
[22] R. D. Skeel, *Iterative refinement implies numerical stability for Gaussian elimination*, Math. Comp., 35 (1980), pp. 817–832.
[23] H. J. Symm and J. H. Wilkinson, *Realistic error bounds for a simple eigenvalue and its associated eigenvector*, Numer. Math., 35 (1980), pp. 113–126.
[24] R. A. Tapia, *The Kantorovich theorem for Newton's method*, Amer. Math. Monthly, 78 (1971), pp. 389–392.
[25] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[26] S. Wang and S. Zhao, *An algorithm for* $Ax = \lambda Bx$ *with symmetric and positive-definite* A *and* B, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 654–660.
[27] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963. Also published by Prentice-Hall, Englewood Cliffs, NJ, 1963. Reprinted by Dover, New York, 1994.
[28] H. Woźniakowski, *Numerical stability for solving nonlinear equations*, Numer. Math., 27 (1977), pp. 373–390.
[29] T. J. Ypma, *The effect of rounding errors on Newton-like methods*, IMA J. Numer. Anal., 3 (1983), pp. 109–118.
[30] T. J. Ypma, *Local convergence of inexact Newton methods*, SIAM J. Numer. Anal., 21 (1984), pp. 583–590.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1058–1078

AN ANALYSIS OF SPARSE APPROXIMATE INVERSE PRECONDITIONERS FOR BOUNDARY INTEGRAL EQUATIONS*

KE CHEN†

Abstract. Preconditioning techniques for dense linear systems arising from singular boundary integral equations are described and analyzed. A particular class of approximate inverse based preconditioners related to the mesh neighbor methods is known to be efficient. This paper shows that it is an operator splitting preconditioner and clusters eigenvalues for the normal equation matrix thus ensuring a fast convergence of the conjugate gradient normal method. Clustering of the eigenvalues of the preconditioned matrix and fast convergence of the generalized minimal residual method are also observed. For the type of problems considered, we demonstrate a crucial connection between two essential features of eigenvalue clustering for a sparse preconditioner—approximate inversion for a small cluster radius and operator splitting for a small cluster size. Experimental results from several boundary integral equations are presented.

Key words. preconditioning, operator splitting, approximate inversion, singular boundary elements, least squares solution, conjugate gradients, conjugate gradient normal method, GMRES

AMS subject classifications. 65F10, 65R20

PII. S0895479898348040

1. Introduction.

In this paper we consider the efficient solution of dense linear systems $A\underline{x} = \underline{b}$ by preconditioned iterative methods, where A is an $n \times n$ unsymmetric matrix. We are concerned with those systems arising from numerical solution of singular boundary integral equations (BIEs), where preconditioning is essential for convergence of iterative methods. Special attention is given to a justification of sparse approximate inverse related preconditioners.

Most BIEs possess singularities. When the underlying integral operator is smooth or only weakly singular, it is compact and iterative methods have been proved to be efficient even without preconditioning; see [1, 5, 6, 27, 29, 38, 40]. In the special case when the singularities are due to geometric nonsmoothness (e.g., corners), the operator can be noncompact but the singularities occur at fixed points. Preconditioners based on separating these fixed singularities have been studied and iterative methods for the preconditioned systems have been shown to be effective; see [8, 12, 15]. Here we consider the general case of singular BIEs, e.g., an integral reformulation of the Helmholtz equations with Neumann's boundary conditions. To iteratively solve the dense linear system from discretization of singular BIEs, we consider three types of sparse preconditioners: the operator splitting preconditioner (OSP), the least squares approximate inverse preconditioner (LSAI), and the diagonal block approximate inverse preconditioner (DBAI).

All these sparse preconditioners have been used in the literature. The OSPs [13, 14, 30, 46], although admitting a number of isolated eigenvalues (see section 2.2), can cluster eigenvalues of both the original matrix and its normal matrix but further improvements are difficult. The LSAI is a general technique and its effectiveness lies in a knowledge of a dominant sparsity pattern of the true inverse (otherwise the technique can be expensive); see [10, 18, 3, 16, 17]. The LSAIs can also cluster eigenvalues

*Received by the editors November 23, 1998; accepted for publication (in revised form) by S. Vavasis August 15, 2000; published electronically February 23, 2001.
http://www.siam.org/journals/simax/22-4/34804.html
†Department of Mathematical Sciences, The University of Liverpool, M & O Building, Peach Street, Liverpool L69 7ZL, UK (k.chen@liv.ac.uk, http://www.liv.ac.uk/∼cmchenke).

collectively but with less tight clusters unless the approximate inverse approaches the true inverse. The DBAIs, under different names and contexts, have been used by many researchers. These include the mesh neighbor preconditioner (MN) of [43], the local least squares inverse approximation preconditioner of [42], the truncated Green's function preconditioner of [26], and the nearest neighbor preconditioner of [44] among others. The effectiveness of DBAIs has been noted in the work of [3, 4, 7, 14, 17]. However, despite many successful experiments, there was no analysis done to indicate why DBAIs should be effective. Here we show that the preconditioner implements a useful operator splitting, possessing the advantages of both LSAI and OSP. Thus, both the preconditioned matrix and its normal matrix have eigenvalue clustering patterns. Further this is also true for the generalized case with more mesh neighbors and in the three dimensional (3D) case.

The rest of the paper is presented as follows. Section 2 provides some background information on solving BIEs and iterative solvers with preconditioning. Section 3 discusses a general OSP that can be used to establish eigenvalue clustering. Section 4 discusses the LSAI in order to introduce the DBAI. Section 5 shows that the DBAI is a version of OSP and a brief discussion is followed in section 6 generalizing the results to the 3D case. Finally section 7 gives some experimental results for three examples solved by two preconditioned conjugate gradient methods.

Define a linear system by $A\underline{u} = \underline{f}$ and write the preconditioned system as $M^{-1}A\underline{u} = M^{-1}\underline{f}$ (left preconditioned) or $AM^{-1}\underline{y} = \underline{f}$ with $\underline{y} = M\underline{u}$ (right preconditioned), with a preconditioner M^{-1}. As is known, there are two somewhat self-conflicting requirements of the preconditioner. The first one follows from convergence estimates of iterative methods, namely, we require M^{-1} be close to A^{-1} (or M to A). Similarly (but not always equivalently), the eigenspectrum and singular value spectrum of the preconditioned matrix (or its normal matrix if conjugate gradient for the normal equation (CGN) is considered) should be clustered, provided all eigenvalues are not sensitive to small perturbations. The second requirement is efficiency; that is, the extra work required to find M^{-1} (or solve $M\underline{x} = \underline{z}$) should be at a minimum. Practically this means that either M or M^{-1} must be sparse or a product of sparse matrices implicitly or explicitly.

We remark that the three classes of preconditioning methods discussed in this paper all satisfy the second requirement but satisfy the first requirement in different ways—this is the key to a successful preconditioner. The first class, OSP, aims to achieve eigenvalue clustering for most eigenvalues of singular BIEs but does not approximate the inverse well. The second class, LSAI, aims to approximate A^{-1} (as the name suggests) and cluster all eigenvalues but may not cluster eigenvalues as tightly as OSP for singular equations. The third class, DBAI, appears as a special case of LSAI but is shown here to be an OSP. Therefore, DBAI can be viewed as an ideal combination of LSAI and OSP producing good approximate inversion (clustering for all eigenvalues) and operator splitting (tight clustering for most eigenvalues) for our class of singular BIEs.

2. Dense linear systems. Let $\Omega \in R^2$ denote[1] a closed domain that may be interior and bounded, or exterior and unbounded, and $\Gamma = \partial\Omega$ be its (finite part) boundary that can be parameterized by $p = (x, y) = (x(s), y(s)), a \le s \le b$. Then a boundary integral equation that usually arises from reformulating a partial differential

[1]The 3D case can be described similarly; see [2] and sections 6 and 7 of this paper.

equation in Ω can be written as

$$\alpha u(p) - \int_\Gamma \overline{k}(p,q)u(q)d\Gamma = f(p), \qquad p \in \Gamma, \tag{1}$$

or

$$\alpha u(s) - \int_a^b k(s,t)u(t)dt = f(s), \qquad s \in [a,b], \tag{2}$$

i.e., simply,

$$(\alpha I - \mathcal{K})u = f. \tag{3}$$

Here u may be a density function; see [2, 7]. We do not assume that $\alpha \neq 0$ so our methods will work for both first and second kind boundary integral equations of the Fredholm type. For the latter type, $\alpha = 1/2$ at smooth points on Γ. We assume that $\mathcal{K}$ is the full operator (not just a principal part); for the Helmholtz equation this refers to the unique formulation which is valid for all wavenumbers (see [2] and section 7).

To solve the above equation numerically, we divide the boundary Γ (interval $[a,b]$) into m boundary elements (nonintersecting subintervals $I_i = (s_{i-1}, s_i)$). On each interval I_i, we may either approximate the unknown u by an interpolating polynomial of order τ that leads to a collocation method or apply a quadrature method of τ nodes and weights w_i, that gives rise to the Nyström method. Both discretization methods approximate (3) by

$$(\alpha I - \mathcal{K}_n)u_n = f, \tag{4}$$

where we can write

$$\mathcal{K}_n u = \mathcal{K}_n u_n = \sum_{j=1}^{m} \left[\sum_{i=1}^{\tau} w_i k(s, t_{ji}) u_{ji} \right], \quad u_n(t_{ji}) = u(t_{ji}) = u_{ji}, \quad \text{and} \quad n = m\tau.$$

We use the vector $\underline{u}$ to denote u_{ji}'s at all nodes. By a collocation step in (4), we obtain a linear system of equations

$$(\alpha I - K)\underline{u} = \underline{f}, \qquad \text{or} \qquad A\underline{u} = \underline{f}, \tag{5}$$

where matrices K and A are dense and unsymmetric (in general). The conditioning of A depends on the smoothness of kernel function $k(s,t)$. A strong singularity (as $t \to s$) leads to noncompactness of operator $\mathcal{K}$ and consequently the iterative solution of (5) requires preconditioning.

2.1. Iterative methods. For a general unsymmetric linear system, there exist many useful iterative solvers of the conjugate gradient type. See [22, 37, 41]. Here we select the following two methods for testing our preconditioners: CGN and the generalized minimum residuals (GMRES); see [37]. We use the usual notation: $\lambda(A)$ denotes the eigenvalue spectrum of A, $\sigma(A) = \sqrt{\lambda(A^*A)}$ the singular value spectrum of A, and $\mathcal{P}_j$ the space of all polynomials of degree up to j. Similarly $\lambda^\epsilon(A)$ denotes the ϵ-pseudospectrum of A; see [37]. For $\lambda(A)$, as in [9], we assume that its members λ_j are ordered such that $|\lambda_{j+1} - \ell| \leq |\lambda_j - \ell|$ for some cluster center ℓ (usually $\ell = 1$). We now briefly review the convergence estimates of these two methods to motivate the need of preconditioning.

CGN. Recall that the error of CGN at the jth iteration is determined by

$$E_j = \inf_{\Psi\in\mathcal{P}_j:\Psi(0)=1} \max_{\lambda_i\in\lambda(A^*A)} |\Psi(\lambda_i)|.$$

GMRES. The relative residual error of the GMRES method, at step j, is generally determined by

$$F_j^\epsilon = \inf_{\Psi\in\mathcal{P}_j:\Psi(0)=1} \sup_{\lambda_i\in\lambda^\epsilon(A)} |\Psi(\lambda_i)|.$$

For the case where A is diagonalizable, this residual error is bounded by the alternative and simpler quantity

$$F_j = \inf_{\Psi\in\mathcal{P}_j:\Psi(0)=1} \max_{\lambda_i\in\lambda(A)} |\Psi(\lambda_i)|,$$

provided the condition number of the matrix of A's eigenvectors is not large.

Clearly smaller values of E_j and F_j^ϵ lead to faster convergence. A good preconditioner should force these quantities to be small. For matrices whose eigenvalues are insensitive to small perturbations, one aims to design preconditioners for $\lambda(A)$ and $\lambda(A^*A)$ to cluster and thus produce small values of E_j and F_j.

By "cluster" or "clustering" we mean that there are a large number of eigenvalues that are inside a small interval [35] or close to a fixed point [9]. If we define, for any $\mu_1 \le \mu_2$, a complex row vector set by $\sum_{[\ell,\mu_2]}^{[n_1,\mu_1]} = \{\mathbf{a}^\top \mid \mathbf{a} = (a_1,\ldots,a_n)^\top \in C^n,\ |a_j - \ell| \le \mu_2$ for $j \ge 1$ and $|a_k - \ell| \le \mu_1$ for $k \ge n_1\}$,then a more precise statement can be made as follows.

DEFINITION 1. *Given a square matrix $A_{n\times n}$, if $\lambda(A) \in \sum_{[\ell,\mu_2]}^{[n_1,\mu_1]}$ for some relatively small n_1 (with respect to n), we say $\lambda(A)$ is clustered at point ℓ with a cluster size μ_1 and cluster radius μ_2.*

Here μ_2 is the radius of a disk, centering at ℓ, containing all the eigenvalues and μ_1 is the radius of a smaller disk that contains most of the eigenvalues (i.e., all eigenvalues except the first $n_1 - 1$).

Remark. As far as convergence of conjugate gradients methods is concerned, point clusterings imply that at step n_1 the underlying approximation in $\mathcal{P}_{n_1}$ is almost as accurate as in $\mathcal{P}_n$. In this sense, both condition number estimates (popular in the literature) and interval clusterings are not as effective as point clusterings (Definition 1) in measuring convergence.

2.2. Compactness, eigenvalue clustering, and preconditioning. We recall that for a compact operator, its eigenvalues cluster at zero. For compact $\mathcal{K}$, the adjoint $\mathcal{K}^*$ and product $\mathcal{K}^*\mathcal{K}$ are also compact. Therefore in this case for A in (5), $\lambda(A) \in \sum_{[\alpha,\mu_2]}^{[n_1,\mu_1]}$ and $\lambda(A^*A) \in \sum_{[|\alpha|^2,\mu_2]}^{[n_1,\mu_1]}$, i.e., clustered at α and $|\alpha|^2$, respectively, with μ_1 arbitrarily small for some suitable and fixed n_1. Further the superlinear convergence of both CGN and GMRES has been established in [45] and [36], respectively.

When $\mathcal{K}$ is not compact, we wish to use preconditioning to achieve compactness and thus eigenvalue clustering and superlinear convergence. To this end, we shall consider on one hand preconditioning a dense matrix for the general case and on the other hand preconditioning the operator. If the preconditioned operator is compact (plus an identity), we expect the preconditioned matrix and its normal to have clustered eigenvalues at 1, that in turn imply fast convergence of iterative solvers.

Before we proceed we show in Figures 1 and 2 two typical examples of eigenvalue distributions $\lambda(A)$ and singular value distributions $\sigma(A)$ of an original matrix

A, compared to preconditioned counterparts $C = M^{-1}A$ for the three main types of preconditioners (OSP, DBAI, LSAI). (Note the different scalings in the figures.) Clearly the preconditioned cases show better clustering patterns, although of different cluster sizes and radii, than the unpreconditioned case. We have also computed the ϵ-pseudospectra for all cases and found that the preconditioned eigenvalues are not sensitive to small perturbations. This implies that all preconditioned systems can be solved efficiently by either CGN or other solvers such as GMRES because eigenvalues can be used as convergence indicators [20, 37].

Moreover, DBAI has the best singular value clustering pattern among all cases in terms of distances from 1 (both cluster size and radius). The following sections explain the different reasons why such clustering patterns occur.

3. Operator splitting preconditioners. Operator splitting is a useful technique in solving singular BIEs; see [13, 14, 30, 46] and the references therein. There are two main approaches. Both make use of the following elementary lemma.

LEMMA 1. *Let linear operators $\mathcal{A}$ and $\mathcal{C}$ be defined in a normed space with $\mathcal{A}$ bounded and $\mathcal{C}$ compact. Then*

1. *operator $\mathcal{A}^{-1}\mathcal{C}$ is also compact;*
2. *operator $\mathcal{B} = \mathcal{A} - \mathcal{C}$ has a bounded inverse if $\mathcal{B}$ is injective.*

The first splitting approach is based on expanding the singular kernel into a principal term of simple forms and a smooth part of remaining terms, giving rise to two splitting operators. The latter part gives rise to a compact operator while the former to a bounded operator. Further because of the simple forms in the kernel, fast algorithms (e.g., the FFT; see [34] and [46]) are used to invert the former operator which serves as a preconditioner.

Here we apply the second idea of operator splitting, previously used in [13]. This is based on domain decomposition rather than kernel decomposition. Use the partition of section 2, $[a, b] = \bigcup_{i=1}^{m} I_i$. Accordingly we can partition the variable u and vector $\underline{u}$ as follows: $u = (u_1, u_2, \ldots, u_m)^T$ and $\underline{u} = (\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_m)^T$. Similarly writing the operator $\mathcal{A}$ in matrix form, we obtain the splitting $\mathcal{A} = \alpha\mathcal{I} - \mathcal{K} = \mathcal{B} - \mathcal{C}$ with $\mathcal{B} = \mathcal{I} + \overline{\mathcal{K}}$) and

$$\overline{\mathcal{K}} = \begin{pmatrix} \mathcal{K}_{1,1} & \mathcal{K}_{1,2} & & & \mathcal{K}_{1,m} \\ \mathcal{K}_{2,1} & \mathcal{K}_{2,2} & \mathcal{K}_{2,3} & & \\ & \mathcal{K}_{3,2} & \ddots & \ddots & \\ & & \ddots & \ddots & \mathcal{K}_{m-1,m} \\ \mathcal{K}_{m,1} & & & \mathcal{K}_{m,m-1} & \mathcal{K}_{m,m} \end{pmatrix}.$$

Observe that all singularities of $\mathcal{A}$ are contained in the above operator and so the smooth operator $\mathcal{C}$ is compact. Note also that, after discretization, the corresponding matrix out of K is

$$\overline{K} = \begin{pmatrix} K_{1,1} & K_{1,2} & & & K_{1,m} \\ K_{2,1} & K_{2,2} & K_{2,3} & & \\ & K_{3,2} & \ddots & \ddots & \\ & & \ddots & \ddots & K_{m-1,m} \\ K_{m,1} & & & K_{m,m-1} & K_{m,m} \end{pmatrix}.$$

Also define matrix $B = \alpha I - \overline{K}$ and $C = K - \overline{K}$. Then from the above lemma, it can be shown that the operator $\mathcal{B}$ is bounded. Since the operator $\mathcal{B}^{-1}\mathcal{C}$ is also

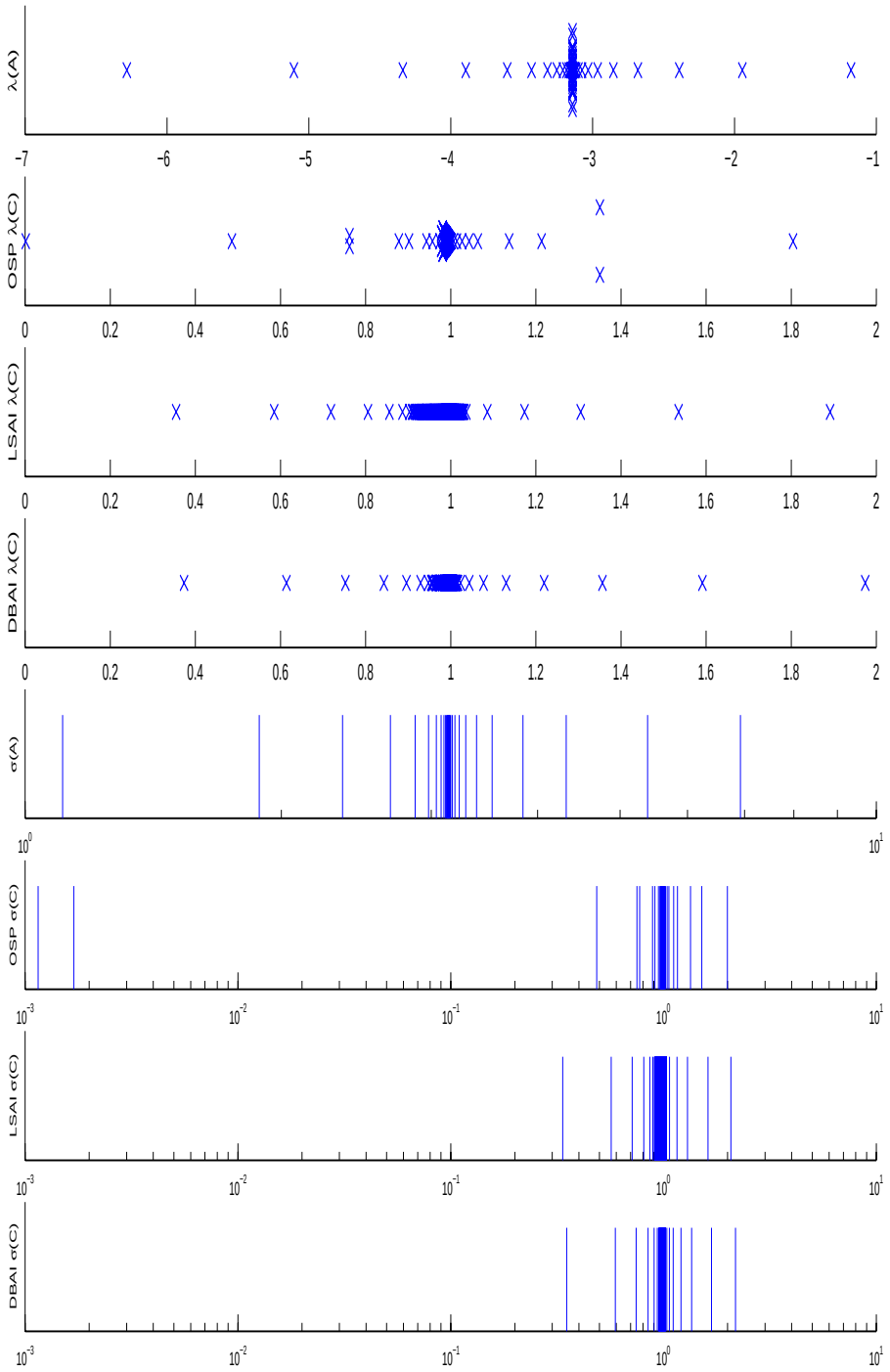


FIG. 1. *Example* 1 *of eigenvalue and singular value distributions of* 4 *cases: original matrix A, OSP, LSAI, DBAI.*

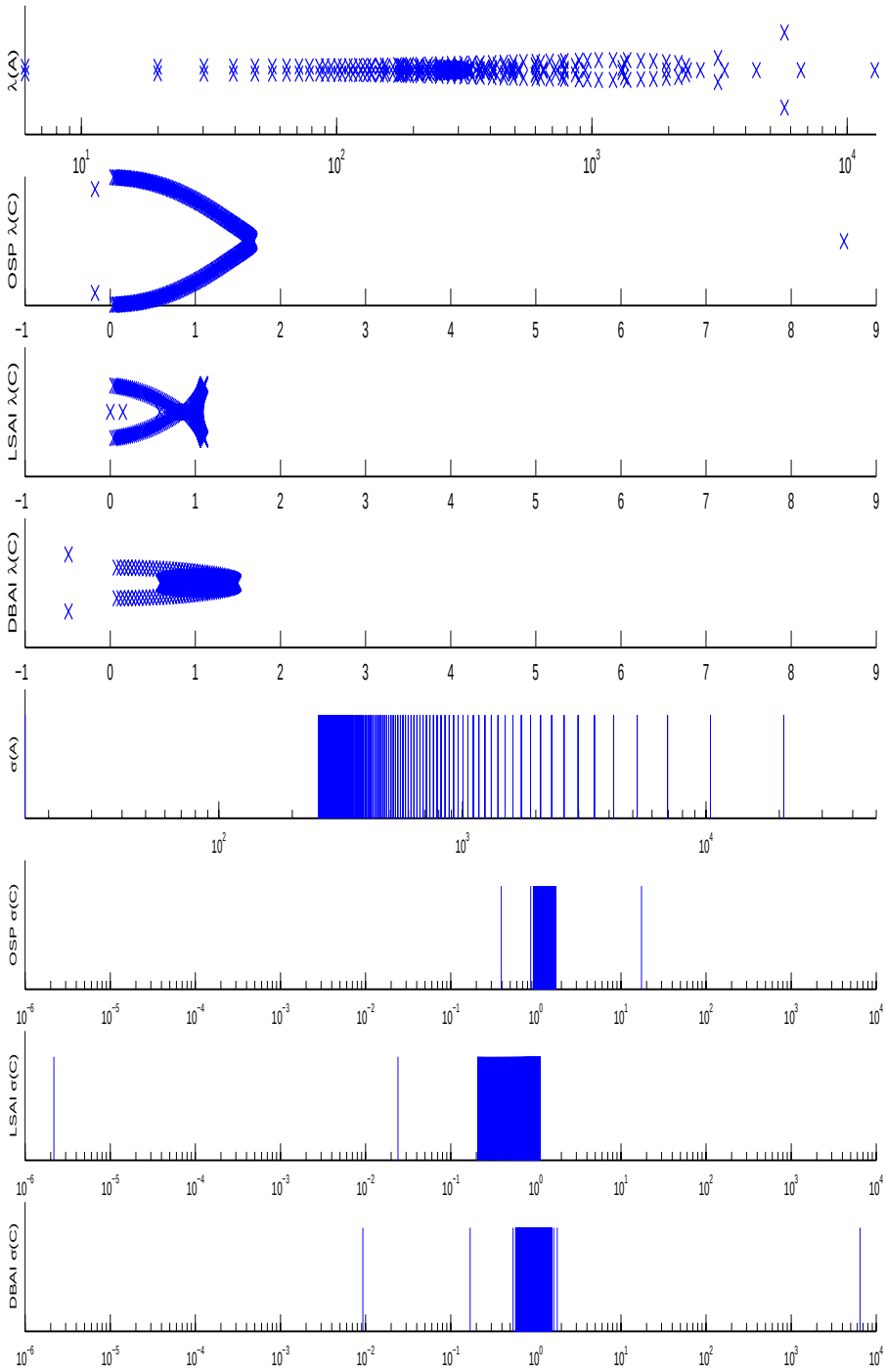


FIG. 2. *Example* 2 *of eigenvalue and singular value distributions of* 4 *cases: original matrix* A*, OSP, LSAI, DBAI.*

compact, we can use $\mathcal{M} = \mathcal{B}$ as an operator preconditioner and $M = B$ as a matrix preconditioner. This is what we call OSP.

Thus the solution of $A\underline{u} = \underline{f}$ is reduced to that of $M^{-1}A\underline{u} = M^{-1}\underline{f}$, i.e., $[I - M^{-1}C]\underline{u} = M^{-1}\underline{f}$. Here B is in general a block quasi-tridiagonal matrix and the solution of $B\underline{x} = \underline{y}$ is via $B = LU$, where L, U are of the same sparsity structure as B apart from the last row of L and the last column of U; see [3, 14].

We remark that our techniques of constructing M, for singular operators, may be viewed as efficient regularization methods. Therefore from properties of compact operators, the preconditioned matrix $(I - M^{-1}C)$ and its normal matrix should have most of its eigenvalues clustered as demonstrated in Figures 1 and 2. Most sparse preconditioners (e.g., approximate inverses [31]) do not possess the latter property of the normal matrix having clustered eigenvalues because of the unsymmetric nature of the matrix (where $\lambda(M^{-1}A)$ and $\sigma(M^{-1}A)$ are not related).

Note that the underlying preconditioner $M^{-1} = B^{-1}$, although not sparse, does not approximate A^{-1} well because $\|M^{-1}A - I\|$ is not small. Solving $B\underline{x} = \underline{y}$ takes $O(kn)$ flops at each step of iterative methods where k (= 3 here) denotes the total bandwidth of B. Using Definition 1, we may write $\lambda(M^{-1}A) \in \sum_{[1,\mu_2]}^{[n_1,\mu_{\text{OSP}}]}$ for a small μ_{OSP} and some relatively small n_1 (with respect to n), but the poor approximation of A by M means that we cannot estimate μ_2. It has been found that, to further improve on the above preconditioner, i.e., reduce n_1, μ_{OSP}, μ_2, increasing the bandwidth k alone is not efficient (or sufficient) as the improvements are only marginal unless $k \approx n$. We now consider a different class of preconditioners (i.e., LSAI) before establishing the connection of mesh neighbor preconditioners with OSP and LSAI.

4. Approximate inversion techniques. Sparse approximate inverse preconditioners are widely used for preconditioning sparse linear systems, since the earlier work of [10] and [33]. See also [18, 25, 31, 41, 16, 17] and the references therein for more details. Here we review the method briefly in order to introduce the DBAI and consider its applications to iterative solution of dense linear systems $A\underline{u} = \underline{f}$ from section 2.

4.1. The LSAI approach. The problem is to construct approximations, M^{-1}, to the inverse of matrix A for which $\|AM^{-1} - I\|$ is small in some norm. Once such an M^{-1} has been constructed, we solve the preconditioned linear system $AM^{-1}\underline{w} = \underline{f}$, $\underline{u} = M^{-1}\underline{w}$.

To specify the matrix M^{-1}, denote $\mathcal{N} = \{1, 2, \ldots, n\}$, and let $\mathcal{S}$ be a given set of (i, j)'s with $[i, j] \in \mathcal{N}$, and $\mathcal{G}_{\mathcal{S}}$ be the space of all $n \times n$ matrices that have entries in positions indexed by $\mathcal{S}$. For each column index $j = 1, \ldots, n$, define its row indices $\mathcal{S}_j = \{i : (i, j) \in \mathcal{S}\}$ and let its vector space $\mathcal{G}_{\mathcal{S}_j}$ contain all vectors that have entries in positions indexed by $\mathcal{S}_j$. Then the approximate inverse M^{-1} is calculated by solving the least squares problem

$$\min_{M^{-1} \in \mathcal{G}_{\mathcal{S}}} \|AM^{-1} - I\|_F^2 = \sum_{j=1}^{n} \min_{m_j \in \mathcal{G}_{\mathcal{S}_j}} \|Am_j - e_j\|_2^2, \tag{6}$$

where $M^{-1} = [m_1, \ldots, m_n]$ and $I = [e_1, \ldots, e_n]$. In theory, such a problem can be posed in any norm but the Frobenius norm leads to an easier solution. The full problem of finding M^{-1} is reduced to n standard least squares problems. The above equation (6) has been used by various researchers. The resulting preconditioner M^{-1} is obviously the right preconditioner since A is multiplied on its right.

But we can equally attempt to search for the left preconditioner M^{-1} by trying to minimize $\|M^{-1}A - I\|$:

$$\min_{M^{-1}\in\mathcal{G}_{\mathcal{S}}} \|M^{-1}A - I\|_F^2 = \min_{M^{-1}\in\mathcal{G}_{\mathcal{S}}} \|A^T M^{-T} - I\|_F^2 = \sum_{j=1}^{n} \min_{m_j\in\mathcal{G}_{\mathcal{V}_j}} \|A^T m_j - e_j\|_2^2, \tag{7}$$

where $M^{-T} = [m_1, \ldots, m_n]$, $I = [e_1, \ldots, e_n]$, $\mathcal{V}_j = \{i : (j, i) \in \mathcal{S}\}$, and $\mathcal{G}_{\mathcal{V}_j}$ contain all vectors that have entries in positions indexed by $\mathcal{V}_j$. We find it slightly convenient to describe and implement the right preconditioner.

The suitable specification of $\mathcal{S}$ will then become essential. For PDEs solved by domain-type discretization methods (mainly finite element and finite difference methods), $\mathcal{S}$ is usually chosen so that M^{-1} is of some sparse structure, e.g., a band matrix. Its choice may also be coupled with the solution process of least squares problems so we could start from an initial sparse specification (say a diagonal matrix) and increase the number of nonzeros adaptively in order to control the residual errors of each least squares solution under some tolerance. See [25, 31, 41].

4.2. Application to singular integral equations. For singular BIEs, we have shown in section 3 that for a suitable preconditioner $M^{-1} = B^{-1}$ (left or right), its inverse $M = B$ can be of the specific sparsity structure

$$\begin{pmatrix}
\times & \times & & & & & & & \times \\
\times & \times & \times & & & & & & \\
 & \times & \times & \times & & & & & \\
 & & \ddots & \ddots & \ddots & & & & \\
 & & & \ddots & \ddots & \ddots & & & \\
 & & & & \times & \times & \times & & \\
 & & & & & \times & \times & \times & \\
 & & & & & & \times & \times & \times \\
\times & & & & & & & \times & \times
\end{pmatrix}. \tag{8}$$

As the diagonal entries of B are large, numerical evidence suggests that the structure of the largest entries in B^{-1} is similar to the structure of B. In fact, if B is strictly diagonal dominant, B^{-1} can be shown to be exponentially decaying from the diagonal. In practice, this strong condition may not be strictly satisfied. For 3D problems, see section 6 for further discussions.

However, it appears reasonable to seek a preconditioner M^{-1} of sparsity structure (8) that can be used as an approximate inverse of A. Thus $\mathcal{S}$ will represent all nonzero positions in (8).

4.3. Solution of the least squares problem. We now consider the solution of the least squares problem for finding the right preconditioner; the left preconditioner can be found similarly. Since matrix $M^{-1} = [m_1,\ m_2,\ \ldots,\ m_n]$ is consisted of column vectors, for each column j, the least squares problem is to solve

$$\min_{m_j\in\mathcal{G}_{\mathcal{S}_j}} \|Am_j - e_j\|_2^2 = \min_{m_j\in\mathcal{G}_{\mathcal{S}_j}} \|\hat{A}_j m_j - e_j\|_2^2$$

or

$$\begin{pmatrix} A_{1j_1} & A_{1j_2} & A_{1j_3} \\ \vdots & \vdots & \vdots \\ A_{j_1j_1} & A_{j_1j_2} & A_{j_1j_3} \\ A_{j_2j_1} & A_{j_2j_2} & A_{j_2j_3} \\ A_{j_3j_1} & A_{j_3j_2} & A_{j_3j_3} \\ \vdots & \vdots & \vdots \\ A_{nj_1} & A_{nj_2} & A_{nj_3} \end{pmatrix} \begin{pmatrix} M_{j_1j} \\ M_{j_2j} \\ M_{j_3j} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{9}$$

or simply

$$\hat{A}_j \hat{m}_j = e_j,$$

where $j_2 = j$, $\hat{m}_j = [M_{j_1j} \;\; M_{j_2j} \;\; M_{j_3j}]^T$, $m_j = [0^T \;\; \hat{m}_j \;\; 0^T]^T$, $j_1 = j-1$, $j_3 = j+1$ for $j = 2, \ldots, n$, $j_1 = n, j_3 = 2$ for $j = 1$, and $j_1 = n-1$, $j_3 = 1$ for $j = n$ due to the choice of $\mathcal{S}$ and the wrap-around nature of M^{-1}.

The least squares problem (9) may be solved by the QR method [11, 24]. For the approximation using this specific pattern $\mathcal{S}$, we have the following theorem.

THEOREM 1. *For the least squares problem* (9) *with* $\hat{A}_j$ *of size* $n \times 3$,

1. *the residual for the solution* $\hat{m}_j$ *satisfies* $\|r_j\|_2 \leq 1$ *because the right-hand side of* (9) *is a unit vector;*
2. *problem* (9) *is equivalent in the least squares sense to the following problem with* B_j *of a smaller size (i.e.,* 4×3*),*

$$B_j \hat{m}_j = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ & b_{32} & b_{33} \\ & & b_{43} \end{pmatrix} \begin{pmatrix} M_{j_1j} \\ M_{j_2j} \\ M_{j_3j} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Further, the residual for the solution $\hat{m}_j$ *can be written more specifically as*

$$r_j = [0 \;\; \bar{r}_j]^T \qquad \text{and} \qquad \bar{r}_j = -\sin\theta_1 \sin\theta_2 \sin\theta_3$$

for some θ_i*'s (so* $\|r_j\|_2 < 1$ *if* $A_{1j_1} \neq 0$*).*

Therefore the matrix residual for the approximate inverse M^{-1} *will be* $E = I - AM^{-1}$ *and its F-norm satisfies* $\|E\|_F^2 = \sum_{j=1}^n \|r_j\|_2^2 < n$ *or* $\|E\|_F < \sqrt{n}$.

Proof. See the appendix. □

Remark. This theorem illustrates the accuracy of inverse approximation using structure (8). More general results of this type and on eigenvalue bounds can be found in [18, 31] among others. In particular, note that the residual error $\|E\|$ is directly linked to the eigenspectrum $\lambda(AM^{-1})$. Using Definition 1, we may write

$$\lambda(AM^{-1}) \in \sum_{[1,\mu_{\text{LSAI}}]}^{[1,\mu_{\text{LSAI}}]},$$

where μ_{LSAI} is generally small depending on the approximation accuracy. This behavior of LSAI having the same (maybe small) cluster radius as the cluster size is different from OSP having a very small cluster size but not necessarily small cluster radius. We shall show that the DBAI is an interesting method that has both small cluster size and small cluster radius.

5. The DBAI and an analysis. In (9), we expect three rows $(j_1,\ j_2,\ j_3)$ of $\hat{A}_j$ to play a dominant role due to the singular nature of the original operator. Therefore we may approximately reduce (9) to a 3×3 system

$$\overline{A}_j \hat{m}_j = \begin{pmatrix} A_{j_1 j_1} & A_{j_1 j_2} & A_{j_1 j_3} \\ A_{j_2 j_1} & A_{j_2 j_2} & A_{j_2 j_3} \\ A_{j_3 j_1} & A_{j_3 j_2} & A_{j_3 j_3} \end{pmatrix} \begin{pmatrix} M_{j_1 j} \\ M_{j_2 j} \\ M_{j_3 j} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tag{10}$$

which of course makes sense from a computational point of view. This modified preconditioner M^{-1}, of form (8), is a DBAI preconditioner. This is the so-called method of mesh neighbors in [43]. The same idea was used in the local least squares inverse approximation preconditioner of [42], the truncated Green's function preconditioner of [26], and the nearest neighbor preconditioner of [44], among others.

While heuristically reasonable, computationally simple, and experimentally successful, the DBAI method has not been justified in theory. Here we present results on an analysis for the method before discussing the generalized version using more mesh neighbors.

To simplify the presentation, we first give two definitions and then a simple lemma.

DEFINITION 2 ($\mathbf{Band}^+(d_L, d_U, b_L, b_U)$). *A band matrix $A_{n\times n}$ with wrap-around boundaries is called $\mathbf{Band}^+(d_L, d_U, b_L, b_U)$ if its lower and upper bandwidths are b_L and b_U, respectively, and if furthermore the first d_L bands below the main diagonal are all zeros and the first d_U bands above the main diagonal are also all zeros.*

DEFINITION 3 ($\mathbf{Band}^-(d_L, d_U, b_L, b_U)$). *A simple band matrix $A_{n\times n}$ (without wrap-around boundaries) is called $\mathbf{Band}^-(d_L, d_U, b_L, b_U)$ if its lower and upper bandwidths are b_L and b_U, respectively, and if furthermore the first d_L bands below the main diagonal are all zeros and the first d_U bands above the main diagonal are also all zeros.*

Note that the first definition here is for matrices with wrap-around boundaries while the second is for simple band matrices without wrap-arounds. In both definitions, the parameters are nonnegative integers, not exceeding $(n-1)$. Here if $d_L d_U \neq 0$, both $\mathbf{Band}^+(d_L, d_U, b_L, b_U)$ and $\mathbf{Band}^-(d_L, d_U, b_L, b_U)$ matrices have a zero diagonal. But if $d_L d_U = 0$ the diagonal information will be stated in the context. With $n = 6$ we may illustrate $\mathbf{Band}^+(0, 1, 2, 1)$ and $\mathbf{Band}^-(0, 1, 2, 1)$, respectively, by

$$\begin{bmatrix} 0 & 0 & \times & 0 & \times & \times \\ \times & 0 & 0 & \times & 0 & \times \\ \times & \times & 0 & 0 & \times & 0 \\ 0 & \times & \times & 0 & 0 & \times \\ \times & 0 & \times & \times & 0 & 0 \\ 0 & \times & 0 & \times & \times & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 & \times & 0 & 0 & 0 \\ \times & 0 & 0 & \times & 0 & 0 \\ \times & \times & 0 & 0 & \times & 0 \\ 0 & \times & \times & 0 & 0 & \times \\ 0 & 0 & \times & \times & 0 & 0 \\ 0 & 0 & 0 & \times & \times & 0 \end{bmatrix}.$$

Therefore, for matrices from section 3, B and $\overline{K}$ are $\mathbf{Band}^+(0, 0, 1, 1)$, and C is $\mathbf{Band}^-(1, 1, n-3, n-3)$. One may verify that, for example, $\mathbf{Band}^+(0, 0, 2, 2) = \mathbf{Band}^+(0, 0, 1, 1) + \mathbf{Band}^+(1, 1, 1, 1)$, and $\mathbf{Band}^+(d_L, d_U, b_L + 3, b_U + 4) = \mathbf{Band}^+(d_L, d_U, 3, 4) + \mathbf{Band}^+(d_L + 3, d_U + 4, b_L, b_U)$.

LEMMA 2 (multiplication of band matrices). *If matrix $A_{n\times n}$ is $\mathbf{Band}^+(0, 0, b_{L_1}, b_{U_1})$, $B_{n\times n}$ is $\mathbf{Band}^+(0, 0, b_{L_2}, b_{U_2})$ and $C_{n\times n}$ is $\mathbf{Band}^-(d_{L_3}, d_{U_3}, b_{L_3}, b_{U_3})$, then*

1. *AB is $\mathbf{Band}^+(0, 0, b_{L_4}, b_{U_4})$, with $b_{L_4} = b_{L_1} + b_{L_2}$ and $b_{U_4} = b_{U_1} + b_{U_2}$;*

2. *AC is* $\mathbf{Band}^-(d_{L_5}, d_{U_5}, b_{L_5}, b_{U_5})$, *with* $d_{L_5} = \max(0, d_{L_3} - b_{L_1})$, $d_{U_5} = \max(0, d_{U_3} - b_{U_1})$, $b_{L_5} = b_{L_1} + b_{U_1} + b_{L_3}$, *and* $b_{U_5} = b_{L_1} + b_{U_1} + b_{U_3}$.

Proof. The proof is made by simple inductions. ∎

We are now in a position to study the singularity separation property of the preconditioned matrix AM^{-1}.

THEOREM 2. *The DBAI preconditioner admits a diagonal operator splitting. Therefore for singular BIEs, the preconditioned matrix and its normal matrix have clustered eigenvalues.*

Proof. Partition matrix A as follows (as illustrated in Figure 3)

$$A = D + B_2 + C_2,$$

where D is the diagonal matrix of A, B_2 is $\mathbf{Band}^+(0, 0, 2, 2)$ (with a zero diagonal), and C_2 is $\mathbf{Band}^-(2, 2, n-5, n-5)$. That is,

$$B_2 = \begin{pmatrix}
 & A_{12} & A_{13} & & & & A_{1n-1} & A_{1n} \\
A_{21} & & A_{23} & A_{24} & & & & A_{2n} \\
A_{31} & A_{32} & & A_{34} & A_{35} & & & \\
 & A_{42} & A_{43} & & A_{45} & \ddots & & \\
 & & \ddots & \ddots & & \ddots & \ddots & \\
 & & & \ddots & \ddots & & \ddots & A_{n-2n} \\
A_{n-11} & & & & \ddots & \ddots & & A_{n-1n} \\
A_{n1} & A_{n2} & & & & A_{nn-2} & A_{nn-1} &
\end{pmatrix}.$$

First, from a similar matrix splitting of the operator $\mathcal{A}$, we can show that the off diagonal operators are compact due to smooth kernels. Therefore assuming the original operator $\mathcal{A}$ is bounded, using Lemma 1, we see that $\overline{A}_j$ has a bounded inverse and so M^{-1} is bounded.

Secondly, to work out an explicit formula for $AM^{-1} - I$ in terms of D, B_2, C_2, we have

$$AM^{-1} = DM^{-1} + B_2M^{-1} + C_2M^{-1},$$

where DM^{-1} is $\mathbf{Band}^+(0, 0, 1, 1)$ as with M^{-1}. From Lemma 2, B_2M^{-1} is $\mathbf{Band}^+(0, 0, 3, 3)$ with a nonzero diagonal and C_2M^{-1} is $\mathbf{Band}^-(1, 1, n-3, n-3)$. Now do a simple splitting $B_2M^{-1} = B_2^{(1)} + B_2^{(2)}$ with $B_2^{(1)}$ as a $\mathbf{Band}^+(0, 0, 1, 1)$ matrix and $B_2^{(2)}$ as $\mathbf{Band}^+(1, 1, 2, 2)$. So defining $C_3 = B_2^{(2)} + C_2M^{-1}$ gives

$$AM^{-1} = DM^{-1} + B_2^{(1)} + C_3. \tag{11}$$

From the construction of M^{-1}, we see that

$$DM^{-1} + B_2^{(1)} = I.$$

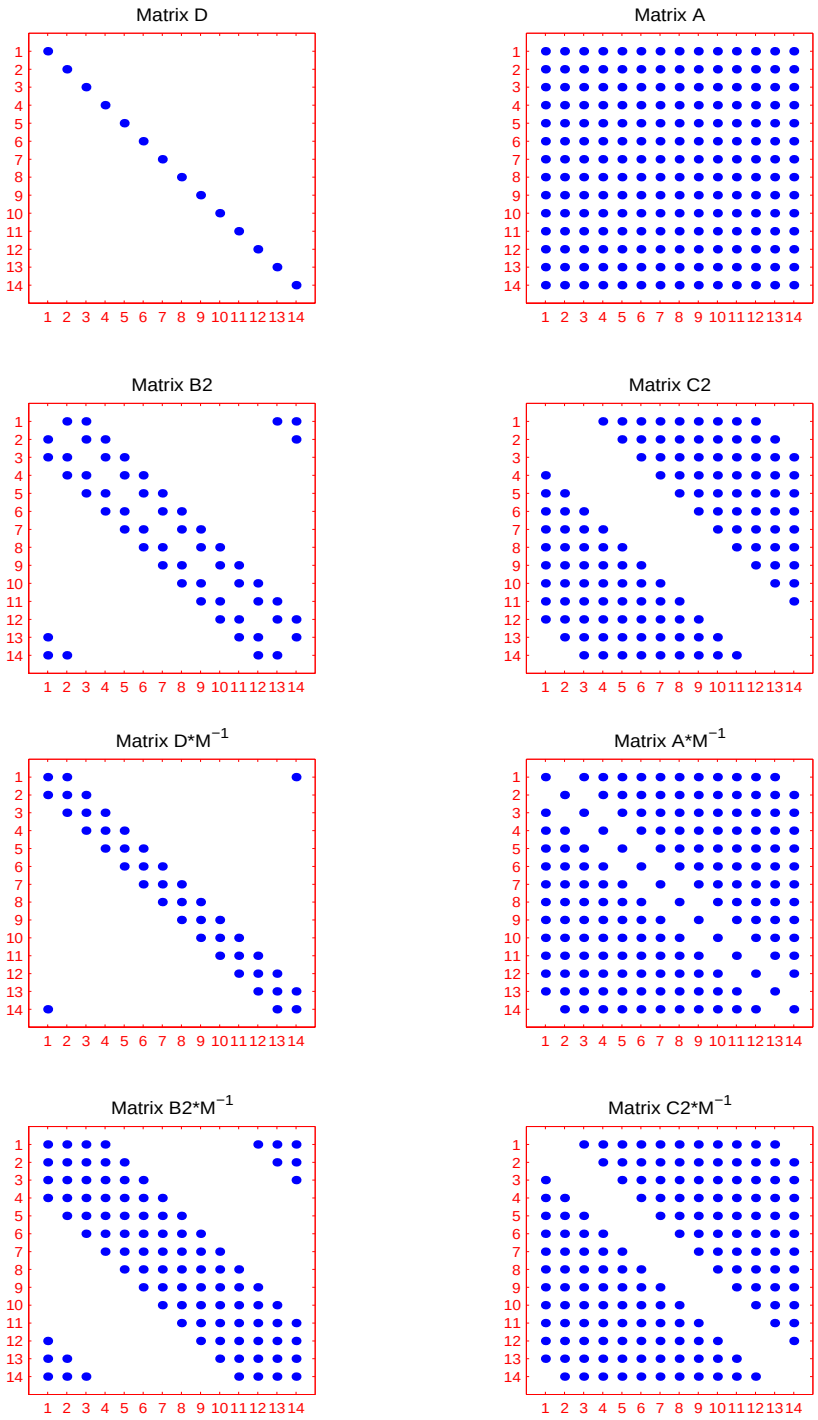


FIG. 3. *Illustration of operator splitting of DBAI* ($n = 14$).

Therefore the matrix D is implicitly inverted because (11) becomes

$$AM^{-1} = I + C_3. \tag{12}$$

In this formula, notice that C_3 is solely determined by terms B_2 and C_2 which correspond to compact operators. Thus matrix C_3 can be viewed as from a discretization of a compact operator and its eigenvalues and those of its normal matrix are thus clustered at 1. ∎

The present DBAI is specified by the pattern in (8), that is, using the nearest neighbors. As is known from [42, 26, 44, 16], one may use more than one level of neighbors. In our notation, this means that we use the new pattern of a $\mathbf{Band}^+(0,0,s,s)$ matrix or a band $k = 2s+1$ matrix instead of a $\mathbf{Band}^+(0,0,1,1)$ matrix or a band 3 matrix. For brevity, we name such a preconditioner DBAI(k). Thus $s = 1$ (or $k = 3$) gives the same DBAI as before. We now consider $s > 1$ (or odd $k \geq 5$).

Then to solve for the jth column of M^{-1}, we solve a new $k \times k$ system

$$\overline{A}_j \hat{m}_j = \begin{pmatrix} A_{j_1 j_1} & \cdots & A_{j_1 j_{s+1}} & \cdots & A_{j_1 j_k} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ A_{j_{s+1} j_1} & \cdots & A_{j_{s+1} j_{s+1}} & \cdots & A_{j_{s+1} j_k} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ A_{j_k j_1} & \cdots & A_{j_k j_{s+1}} & \cdots & A_{j_k j_k} \end{pmatrix} \begin{pmatrix} M_{j_1 j} \\ \vdots \\ M_{j_{s+1} j} \\ \vdots \\ M_{j_k j} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{13}$$

where $j_{s+1} = j$ always, $j_\ell = j + \ell - s - 1$ for $\ell = 1, \ldots, k$, and j_ℓ's take wrap-around index values outside the range of $[1, n]$ as with (9). Compare to (10).

It remains to identify the operator splitting implied in this DBAI(k). We can prove the following.

THEOREM 3. *The DBAI(k) admits the same diagonal operator splitting as DBAI. Therefore for singular BIEs, the preconditioned matrix and its normal have clustered eigenvalues.*

Proof. Follow the similar lines of proving Theorem 2, partition matrix A as follows:

$$A = D + B_{2s} + C_{2s},$$

where B_{2s} is $\mathbf{Band}^+(0,0,2s,2s)$ and C_{2s} is $\mathbf{Band}^-(2s,2s,n-2s-1,n-2s-1)$, to complete the proof. ∎

We have thus shown that DBAI is an OSP (having a small cluster size), although it appears more like an LSAI method (having a small cluster radius). So DBAI possesses advantages of both methods: inverse approximation (of LSAI) and operator splitting (of OSP). Using Definition 1, we may write for DBAI $\lambda(AM^{-1}) \in \sum_{[1,\mu_{\rm LSAI}]}^{[n_1,\mu_{\rm OSP}]}$, where the cluster radius $\mu_{\rm LSAI}$ is related to the approximation error (that can be made smaller by increasing k) and $\mu_{\rm OSP}$ is small due to operator splitting.

It remains to specify what k should be used. Since working out the preconditioner M^{-1} takes $\mathrm{O}(k^3 n)$ operations, to ensure that this work does not exceed n^2 operations (one step of matrix vector multiplication), we suggest to choose k as an odd integer satisfying $3 \leq k \leq cn^{1/3}$ for some fixed constant c (say $c = 1$). This will be used in the experiments below.

6. Analysis of the 3D case. The analysis presented so far is mainly for two dimensional (2D) problems. However, for 3D problems, a similar analysis can be done. For LSAI and DBAI, the essential difference is that the sparsity pattern $\mathcal{S}$ due to mesh neighbors, depending on the geometry of the surface and ordering, is more irregular and complex than that from (8). This is because the mesh neighbors are not always related to neighboring entries in matrix A. In the 3D example of section 7, the number of mesh neighbors varies from element to element (say one case with 4 neighbors and another with at least 9 neighbors).

However, it is not difficult to understand why the analysis presented for DBAI can be generalized to this case in a similar way since all we need to do is to replace band matrices by pattern matrices. Let $\mathcal{S}$ denote the sparsity pattern of a mesh neighboring strategy (see section 7 for both edge and edge/vertex based strategies). This includes the case of introducing levels of neighbors as in the 2D case.

DEFINITION 4. *For any matrix B, given the sparsity pattern $\mathcal{S}$, define the pattern $\mathcal{S}$ splitting of B as*

$$B = Patt_{\mathcal{S}}(B) + Pato_{\mathcal{S}}(B),$$

where $Patt_{\mathcal{S}}(B)$ is the sparse matrix taking elements of B at location $\mathcal{S}$ and zeros elsewhere and $Pato_{\mathcal{S}}(B)$ is the complement matrix for B.

If we use M^{-1} to denote the DBAI preconditioner based on $\mathcal{S}$, then $M^{-1} = \mathrm{Patt}_{\mathcal{S}}(M^{-1})$.

We can now establish that the DBAI preconditioner admits a diagonal splitting. As in the proof of Theorem 2, partition matrix A as follows:

$$A = D + C,$$

where $D = \mathrm{diag}(A)$. Then

$$\begin{aligned} AM^{-1} &= DM^{-1} + CM^{-1} \\ &= \mathrm{Patt}_{\mathcal{S}}(DM^{-1} + CM^{-1}) + \mathrm{Pato}_{\mathcal{S}}(CM^{-1}) \\ &= I + \mathrm{Pato}_{\mathcal{S}}(CM^{-1}) \\ &= I + C_3, \end{aligned}$$

because $\mathrm{Patt}_{\mathcal{S}}$ is not affected by diagonal scaling and it also has the simple summation property. As with (12), matrix C_3 is solely determined by matrix C, which corresponds to a compact operator. Thus DBAI admits a diagonal operator splitting. Therefore the DBAI preconditioned matrix and its normal matrix have clustered eigenvalues at 1 with a small cluster size. Also from the approximation inversion property of DBAI, we know that the eigenvalues have a small cluster radius.

Remark. For both OSP- and LSAI-type methods, as is known, one may improve the eigenvalue clustering (in particular the cluster size for OSP and cluster radius for LSAI). However, as our analysis shows, the DBAI using a more complex sparsity pattern $\mathcal{S}$ does not imply a similar operator splitting beyond the diagonal splitting (i.e., one cannot observe much change in the cluster size) although the cluster radius will be reduced. It is straightforward to establish that a block matrix version of DBAI admits a block diagonal splitting. More work is needed to find a DBAI-like method admitting more than the block diagonal splitting (say, tridiagonal in two dimensions).

7. Numerical experiments. We describe the application of three types of preconditioners, OSP, LSAI, and DBAI(k), discussed in the paper to three examples. For simplicity, we shall use the following abbreviations.

Code name	Preconditioning method	Set up work (flops)
OSP	OSP preconditioning A ($M_1^{-1}A$, section 3)	O(n)
LSAI	LSAI preconditioning A (AM_2^{-1}, section 4)	$3n^2$
DBAI(k)	DBAI preconditioning A (AM_3^{-1}, section 5)	n^2
No	Unpreconditioned case	0

On the above list, column three indicates the amount of work needed to compute the preconditioner M^{-1} (or to prepare for solving $M\underline{x} = \underline{z}$) and this set up is only done once. The amount of work needed (by all three methods) at each iterative step is only O(n). We remark that DBAI(k), as a preconditioner, has been used successfully to solve other examples. See [43, 26, 44] for solving the 3D Laplace's equation in potential theory and [42] for solving differential equations with multigrid methods.

Example 1. Cauchy SIE. Singular integral equations (SIE) of Cauchy type are important in fracture mechanics. Consider a Cauchy SIE

$$(14)\quad \begin{cases} \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} \frac{w(t)\phi(t)}{t-x}dt + \int_{-1}^{1} \frac{(t^2-x^2)^2}{t^2+x^2} w(t)\phi(t)dt = f(x), & x \in (-1,1), \\ \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} w(t)\phi(t)dt = 0, \end{cases}$$

with the exact solution $\phi(x) = x|x|$, where $w(t) = (1-t^2)^{-1/2}$; see [19, 21, 24, 32] for full details. Here

$$f(x) = \frac{2}{\pi}\left(1 + x^2\omega \log\left|\frac{(1-x)\omega+1}{(x-1)\omega+1}\right|\right) \qquad \text{with} \qquad \omega = \frac{1}{\sqrt{1-x^2}}.$$

We choose this example because the integral equation resembles a singular BIE in the sense that the operator has a singular principal part and a smooth part.

Example 2. Singular BIE for 2D Helmholtz equation. The exterior Helmholtz equation (see [2])

$$(\nabla^2 + \mathtt{k}^2)\phi(p) = 0$$

is of importance in acoustic scattering problems. The interior boundary Γ is the ellipse $(x/0.5)^2 + (y/2)^2 = 1$. For Neumann's boundary conditions, a unique BIE formulation due to Burton and Miller is the following:

$$\left(-\frac{1}{2}\mathcal{I} + \mathcal{M}_{\mathtt{k}} + i\eta\mathcal{N}_{\mathtt{k}}\right)\phi = \left[\mathcal{L}_{\mathtt{k}} + i\eta\left(\frac{1}{2}\mathcal{I} + \mathcal{M}_{\mathtt{k}}^T\right)\right]\frac{\partial\phi}{\partial n}.$$

Here $\mathcal{L}_{\mathtt{k}}$ and $\mathcal{M}_{\mathtt{k}}$ are the usual single and double layer potential operators, respectively,

$$(\mathcal{L}_{\mathtt{k}}\phi)(p) = \int_\Gamma G_{\mathtt{k}}(p,q)\phi(q)dS, \qquad (\mathcal{M}_{\mathtt{k}}\phi)(p) = \int_\Gamma \frac{\partial G_{\mathtt{k}}}{\partial n_q}\phi(q)dS,$$

and $\mathcal{M}_{\mathtt{k}}^T$ is the adjoint of $\mathcal{M}_{\mathtt{k}}$ and $\mathcal{N}_{\mathtt{k}}$ is the hypersingular operator. Recall that the 2D Green function is $G_{\mathtt{k}}(p,q) = \frac{i}{4}H_0^{(1)}(\mathtt{k}|p-q|)$ with $H_0^{(1)}$ the Hankel function. Refer to [2, 3, 4, 28]. Here we have tested the case of wavenumbers $\mathtt{k} = 1, 5, 10$ using $\eta = 1/\mathtt{k}$ and the collocation method.

Example 3. Singular BIE for 3D Helmholtz equation. Following the last example, we solve the exterior Helmholtz equation with Neumann's boundary conditions in a 3D domain of a cylinder of radius 0.6 and height 1.8. Again the Burton and Miller formulation is used. Here the free space Green function is $G_k = e^{ik|p-q|}/(4\pi|p-q|)$. The 3D surface is approximated by quadratic functions over a triangular mesh; see [2, 6, 28]. In this 3D case, for DBAI(k), k can be a fixed integer depending on n as discussed above and used in [26]. However, we have found that making k variable for the number of columns of M^{-1} and dependent on the actual number of mesh neighbors is a better strategy. This does not complicate the algorithm. Specifically, we have implemented two cases: type I—near neighbors share a common edge ($k = 4$) and type II—near neighbors share either a common edge or a common vertex ($k \geq 9$). Correspondingly, the OSP and DBAI preconditioners are given code names: OSP(I), OSP(II) and DBAI(I), DBAI(II). As the unpreconditioned case and the LSAI method are not competitive, we omit the details but show the cpu time from using a direct solver (Gaussian elimination with partial pivoting) instead.

Tables 1–3 show both the number of iteration steps and the corresponding cpu seconds required to reduce the residual error to be of the same magnitude as the discretization error, respectively, for the above three examples; see [39] for an earlier use of this kind of stopping criteria. This error refers to the root mean square (RMS) error of the computed solution against the exact solution at all nodal points. An "*" entry indicates that the method cannot converge within N iteration steps or diverges. The tests were carried out on a SGI IP30 using double precision (Fortran). For DBAI(k), k varies with N as discussed. The performance of the CGN and GMRES are similar, although CGN is slightly better. As the preconditioning step takes O(n) operations (negligible), cpu times for each case are proportional to the number of iterations.

The results clearly demonstrate that for dense linear systems arising from singular BIEs, all preconditioners are effective. In particular, as our theory predicted, DBAI(k) type preconditioners perform better than OSP and LSAI, and OSP is better than LSAI. Because LSAI has more "*" entries, it is the least robust method. Note that although our choice of k for DBAI(k) appears to be adequate, it may be possible that an optimal (and better) choice exists. For the restarted GMRES(m), we have used a fixed number $m = 5$ and it may be possible to find a better value; this is beyond the scope of the paper. For Helmholtz equations, all preconditioners show some dependence on the wavenumbers (`k`) and CGN is less sensitive to `k` than GMRES. Although for practical applications the size N should depend on `k` (or frequency) to achieve a certain accuracy, it is of interest to find a reason for such behaviors.

8. Conclusions. We have discussed three types of sparse approximate inverse preconditioners suitable for dense linear systems arising from singular integral equations: the operator splitting preconditioner, the least squares approximate inverse preconditioner, and the diagonal block approximate inverse preconditioner. Both the operator splitting preconditioner and the least squares approximate inverse preconditioner can cluster eigenvalues—the former gives a small cluster size but not necessarily a small cluster radius, and the latter produces a small cluster radius but not necessarily a very small cluster size. We have shown that DBAI, appearing to be a LSAI preconditioner, is an OSP. Therefore it can cluster eigenvalues of the preconditioned matrix giving a small cluster size as well as a small cluster radius and it can also cluster eigenvalues of the normal matrix.

Numerical experiments show that, for the type of problems considered here, the

TABLE 1
Iterative solution of Example 1.

CGN method								
Size N	No	[cpu]	OSP	[cpu]	LSAI	[cpu]	DBAI(k)	[cpu]
256	129	5.76	97	4.40	138	6.55	61	2.85
512	274	80.2	132	38.8	163	50.1	72	20.0
1024	582	714	167	208	185	226	84	105
2048	1546	8154	215	1149	206	1075	100	530
GMRES(5) method								
Size N	No	[cpu]	OSP	[cpu]	LSAI	[cpu]	DBAI(k)	[cpu]
256	385	48.86	28	3.52	*	*	22	2.85
512	*	*	37	41.7	*	*	29	33.1
1024	*	*	51	251	*	*	31	140
2048	*	*	58	1275	*	*	43	974

TABLE 2
Iterative solution of Example 2.

Size N	No	[cpu]	OSP	[cpu]	LSAI	[cpu]	DBAI(k)	[cpu]
CGN method							(wavenumber k=1)	
256	128	14.4	28	3.44	*	*	16	1.73
512	257	118	58	28.4	*	*	26	15.2
1024	513	954	118	225	*	*	41	91.2
2048	1027	7120	258	1957	*	*	68	538
CGN method							(wavenumber k=5)	
256	91	9.63	17	2.09	*	*	12	1.91
512	199	83.8	42	18.9	*	*	17	10.2
1024	421	717	86	151	*	*	30	65.4
2048	925	6413	183	1394	*	*	54	441
CGN method							(wavenumber k=10)	
256	90	9.53	23	2.72	*	*	25	3.14
512	198	83.4	41	18.5	*	*	27	14.4
1024	419	714	87	153	*	*	31	67.1
2048	898	6226	181	1275	*	*	53	434
GMRES(5) method							(wavenumber k=1)	
N = 256	70	34.8	12	6.37	89	44.6	10	5.72
N = 512	132	271	16	34.5	142	293	13	30.2
N =1024	252	2049	26	218	252	1957	18	162
N =2048	526	15459	33	995	410	12080	24	777
GMRES(5) method							(wavenumber k=5)	
N = 256	62	28.5	25	11.8	95	43.9	18	8.94
N = 512	140	255	28	52.5	145	266	27	52.4
N =1024	255	1852	34	253	269	1960	36	275
N =2048	559	16428	43	1289	455	13401	43	1335
GMRES(5) method							(wavenumber k=10)	
N = 256	57	26.2	34	15.9	121	55.8	17	8.48
N = 512	198	83.4	41	18.5	173	317	29	56.1
N =1024	305	2214	50	369	319	2322	47	356
N =2048	640	18806	62	1847	542	15956	67	2040

preconditioner DBAI(k) is indeed more robust than LSAI and OSP. For other types of problems where diagonal operator splitting is not appropriate, we expect LSAI may be more useful. Experiments involving further generalizations are in progress.

Appendix. Proof of Theorem 1.

Proof. For an orthogonal matrix $Q = [q_1, \ldots, q_n]$, let the QR-decomposition of

TABLE 3
Iterative solution of Example 3.

Size	Direct	OSP (I)		OSP (II)		DBAI (I)		DBAI (II)	
N	cpu	Steps	cpu	Steps	cpu	Steps	cpu	Steps	cpu
CGN method								(wavenumber k=1)	
576	102	31	20.2	15	24.4	41	19.5	25	15.0
2304	7188	63	641	31	943	83	605	51	430
CGN method								(wavenumber k=5)	
576	102	15	13.5	14	24.0	16	9.1	17	11
2304	7188	24	377	15	835	31	253	19	213
CGN method								(wavenumber k=10)	
576	102	17	14.3	15	24.4	19	10.3	18	21.1
2304	7188	18	337	17	848	18	165	19	213
GMRES(5) method								(wavenumber k=1)	
576	102	14	33.9	7	31.6	17	34.7	12	27.5
2304	7188	26	1011	15	1195	36	1143	21	728
GMRES(5) method								(wavenumber k=5)	
576	102	6	18.8	4	26.0	7	15.9	5	14.3
2304	7188	11	554	7	951	14	474	9	363
GMRES(5) method								(wavenumber k=10)	
576	102	4	15.1	3	24.1	4	10.2	3	10.6
2304	7188	7	433	4	860	9	322	6	272

$\hat{A}_j$ be

$$\hat{A}_j = Q^T \begin{pmatrix} R \\ 0 \end{pmatrix}$$

and let $Qe_j = q_j$. Define $q_j = [\hat{c}^T \quad \hat{d}^T]^T$ where $\hat{c}^T$ is of size 3 and $\hat{d}^T$ is of size $n-3$. Then (9) is equivalent in the least squares sense to $R\hat{m}_j = \hat{c}^T$. The solution is $\hat{m}_j = R^{-1}\hat{c}^T$. The residual error will be $r_j = e_j - \hat{A}_j\hat{m}_j = Q^T[0 \quad \hat{d}^T]^T$.

The first result is trivial because q_j is a unit vector and so $\|r_j\|_2 = \|\hat{d}\|_2 \leq \|q_j\|_2 = 1$. For the second result, we first multiply a permutation matrix $P_{1,j}$ (also orthogonal), permuting rows 1 and j, to (9). The resulting equation can be applied by three Householder transformations giving rise to the reduced 4×3 problem.

Further, a sequence of three successive Givens transformations

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \cos\theta_3 & \sin\theta_3 \\ & & -\sin\theta_3 & \cos\theta_3 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & \cos\theta_2 & \sin\theta_2 & \\ & -\sin\theta_2 & \cos\theta_2 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_1 & \sin\theta_1 & & \\ -\sin\theta_1 & \cos\theta_1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

can reduce the 4×3 matrix B_j to an upper triangular matrix $(R^T \quad 0^T)^T$ and the right-hand side to

$$\hat{c} = [\cos\theta_1 \quad -\sin\theta_1\cos\theta_2 \quad \sin\theta_1\sin\theta_2\cos\theta_3 \quad -\sin\theta_1\sin\theta_2\sin\theta_3]^T$$

for some θ_i's. Note that $|\bar{r}_j| \leq 1$ but if $b_{11} = A_{1j_1} \neq 0$, $|\sin\theta_1| \neq 1$ so $\|r_j\|_2 < 1$. Thus the second result follows. □

Acknowledgments. The author wishes to thank the anonymous referees who gave critical and useful comments to help clarify and improve the paper in a substantial way. He is also grateful to Peter Appleby for reading a draft version of the manuscript and making comments.

REFERENCES

[1] S. Amini and K. Chen, *Conjugate gradient method for second kind integral equations—applications to the exterior acoustic problem*, Engrg. Anal. Bound. Elem., 6 (1989), pp. 72–77.

[2] S. Amini, P. J. Harris, and D. T. Wilton, *Coupled Boundary and Finite Element Methods for the Solution of the Dynamic Fluid-Structure Interaction Problem*, Springer-Verlag, New York, 1992.

[3] S. Amini and N. Maines, *Preconditioned Krylov subspace methods for boundary element solution of the Helmholtz equation*, Internat. J. Numer. Meth. Engrg., 41 (1998), pp. 875–898.

[4] S. Amini and N. Maines, *Qualitative Properties of Boundary Integral Operators and Their Discretizations*, Mathematics Technical report MCS-95-12, University of Salford, UK, 1995.

[5] K. E. Atkinson, *Iterative variants of the Nystrom method for the numerical solution of integral equations*, Numer. Math., 22 (1973), pp. 17–31.

[6] K. E. Atkinson, *Two-grid iteration methods for linear integral equations of the second kind on piecewise smooth surfaces in R^3*, SIAM J. Sci. Comput., 15 (1994), pp. 1083–1104.

[7] K. E. Atkinson, *The Numerical Solution of Fredholm Integral Equations of the Second Kind*, Cambridge University Press, Cambridge, UK, 1996.

[8] K. E. Atkinson and I. G. Graham, *Iterative solution of the linear systems arising from the boundary integral method*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 694–722.

[9] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.

[10] M. W. Benson and P. O. Frederickson, *Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems*, Util. Math., 22 (1982), pp. 127–140.

[11] A. Bjorck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[12] K. Chen, *Conjugate gradient methods for the solution of boundary integral equations on a piecewise smooth boundary*, J. Comput. Phys., 97 (1991), pp. 127–143.

[13] K. Chen, *Efficient iterative solution of linear systems from discretizing singular integral equations*, Electron. Tran. Numer. Anal., 2 (1994), pp. 76–91.

[14] K. Chen, *On a class of preconditioning methods for dense linear systems from boundary elements*, SIAM J. Sci. Comput., 20 (1998), pp. 684–698.

[15] K. Chen and S. Amini, *Numerical analysis of boundary integral solution of the Helmholtz equation in domains with non-smooth boundaries*, IMA J. Numer. Anal., 13 (1994), pp. 43–66.

[16] E. Chow, *A priori sparsity patterns for parallel sparse approximate inverse preconditioners*, SIAM J. Sci. Comput., 21 (2000), pp. 1804–1822.

[17] E. Chow and Y. Saad, *Parallel approximate inverse preconditioners*, in Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, March 1997, pp. 14–17; also available online from http://www.llnl.gov/CASC/people/chow/pubs/history.ps.

[18] J. D. F. Cosgrove, J. C. Diaz, and A. Griewank, *Approximate inverse preconditioners for sparse linear systems*, Int. J. Comput. Math., 44 (1992), pp. 91–110.

[19] J. A. Cuminato, *On the uniform convergence of a collocation method for a class of singular integral equations*, BIT, 27 (1987), pp. 190–202.

[20] M. Embree, *How Descriptive Are GMRES Convergence Bounds?*, University of Oxford Computing Lab., Oxford, UK, NA Report 99/08 1999; http://web.comlab.ox.ac.uk/oucl/work/mark.embree/estimates.ps.gz.

[21] F. Erdogan and G. D. Gupta, *On the numerical solution of singular integral equations*, Quart. Appl. Math., 30 (1972), pp. 525–534.

[22] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numerica, Cambridge University Press, UK, 1992, pp. 57–100.

[23] A. Gerasoulis, *Nyström's iterative variant methods for the solution of Cauchy singular integral equations*, SIAM J. Numer. Anal., 26 (1989), pp. 430–441.

[24] G. H. Golub and C. Van Loan, *Matrix Computation*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[25] N. I. M. Gould and J. A. Scott, *Sparse approximate-inverse preconditioners using norm-minimization techniques*, SIAM J. Sci. Comput., 19 (1998), pp. 605–625.

[26] A. Grama, V. Kumar, and A. Sameh, *Parallel hierarchical solvers and preconditioners for boundary element methods*, SIAM J. Sci. Comput., 20 (1999), pp. 337–358.

[27] W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, New York, 1985.

[28] P. J. Harris, *A boundary element method for the Helmholtz equation using finite part integration*, Comput. Methods Appl. Mech. Engrg., 95 (1992), pp. 331–342.

[29] P. W. Hemker and H. Schippers, *Multigrid methods for the solution of Fredholm integral equations of the second kind*, Math. Comp., 36 (1981), pp. 215–232.
[30] H. Holm, M. Maischak, and E. P. Stephan, *The hp-Version of the Boundary Element Method for Helmholtz Screen Problems*, Institute For Applied Mathematics report, IFAM 7, University of Hannover, Germany, 1995. (Available via ftp from ftp.ifam.uni-hannover.de/pub/preprints.)
[31] T. Huckle and M. Grote, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1998), pp. 838–853.
[32] N. I. Ioakimidis and P. S. Theocaris, *A comparison between the direct and the classical numerical methods for the solution of Cauchy type singular integral equations*, SIAM J. Numer. Anal., 17 (1980), pp. 115–118.
[33] L. Y. Kolotilina and A. Y. Yeremin, *On a family of two-level preconditionings of the incomplete block factorization type*, Soviet J. Numer. Anal. Math. Modelling, 1 (1986), pp. 293–320.
[34] F. R. Lin, M. K. Ng, and R. Chan, *Preconditioners for Wiener-Hopf equations with high-order quadrature rules*, SIAM J. Numer. Anal., 34 (1997), pp. 1418–1431.
[35] S. V. Parter and S. P. Wong, *Preconditioning second-order elliptic operators: Condition numbers and the distribution of the singular values*, J. Sci. Comput., 6 (1991), pp. 129–157.
[36] I. Moret, *A note on the superlinear convergence of GMRES*, SIAM J. Numer. Anal., 34 (1997), pp. 513–516.
[37] N. M. Nachtigal, S. Reddy, and N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.
[38] L. Reichel, *A method for preconditioning matrices arising from linear integral equations for elliptic boundary value problems*, Computing, 37 (1986), pp. 125–136.
[39] L. Reichel, *Parallel iterative methods for the solution of Fredholm integral equations of the second kind*, in Hypercube Multiprocessors 1987, M. T. Heath, ed., SIAM, Philadelphia, 1987, pp. 520–529.
[40] L. Reichel, *A matrix problem with application to rapid solution of integral equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 263–280.
[41] Y. Saad, *Iterative Solution for Sparse Linear Systems*, PWS, Boston, 1996.
[42] W. P. Tang and W. L. Wan, *Sparse approximate inverse smoother for multigrid*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1236–1252.
[43] S. Vavasis, *Preconditioning for boundary integral equations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 905–925.
[44] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory*, SIAM J. Sci. Comput., 15 (1994), pp. 713–735.
[45] R. Winther, *Some superlinear convergence results for the conjugate gradient method*, SIAM J. Numer. Anal., 17 (1980), pp. 14–17.
[46] Y. Yan, *Sparse preconditioned iterative methods for dense linear systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1190–1200.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1079–1088

SUCCESSIVELY ORDERED ELEMENTARY BIDIAGONAL FACTORIZATION*

CHARLES R. JOHNSON†, D. D. OLESKY‡, AND P. VAN DEN DRIESSCHE§

Abstract. Let D be a diagonal matrix and E_{ij} denote the n-by-n matrix with a 1 in entry (i,j) and 0 in every other entry. An n-by-n matrix A has a successively ordered elementary bidiagonal (SEB) factorization if it can be factored as

$$A = \left(\prod_{k=1}^{n-1} \prod_{j=n}^{k+1} L_j(s_{jk}) \right) D \left(\prod_{k=n-1}^{1} \prod_{j=k+1}^{n} U_j(t_{kj}) \right),$$

in which $L_j(s_{jk}) = I + s_{jk}E_{j,j-1}$ and $U_j(t_{kj}) = I + t_{kj}E_{j-1,j}$ for some scalars s_{jk}, t_{kj}. Note that some of the parameters s_{jk}, t_{kj} may be zero, and the order of the bidiagonal factors is fixed. If this factorization corresponds to reduction of A to D via successive row/column operations in the specified order, it is called an elimination SEB factorization. New rank conditions are formulated that are proved to be necessary and sufficient for matrix A to have such a factorization. These conditions are related to known but more restrictive properties that ensure a bidiagonal factorization as above, but with all parameters s_{jk}, t_{kj} nonzero.

Key words. bidiagonal matrix, elimination, factorization, rank

AMS subject classification. 15A23

PII. S0895479800373322

1. Introduction.

We begin by defining an elementary bidiagonal factorization and then focus on a particular order for the factors. Let I denote the n-by-n identity matrix, and $E_{ij}, 1 \le i,j \le n$, denote the n-by-n 0,1 matrix whose (i,j) entry, and no other, is 1. Define $L_i(s) = I + sE_{i,i-1}$ and $U_j(t) = I + tE_{j-1,j} = L_j^T(t)$, for $2 \le i,j \le n$ and parameters $s,t \in F$, a given field. Matrices of the form $L_i(s)$ or $U_j(t)$ are called *elementary bidiagonal (EB) matrices.* A matrix $A \in M_n(F)$ has an EB factorization if A can be written as a product of EB matrices and at most one diagonal matrix. Without restriction upon order or number of factors, such a factorization of A always exists [6].

Most previous work on EB factorization has dealt with the factorization of totally nonnegative matrices (that is, matrices in which all minors are nonnegative); see, for, example, [1, 3, 5, 7, 8]. In this context, a factorization that corresponds to elimination of entries in the following order

$$\begin{aligned} &(n,1), \dots, (2,1),\ (n,2), \dots, (3,2), \dots, (n,n-1), \\ &(1,n),\ \dots, (1,2),\ (2,n), \dots, (2,3), \dots, (n-1,n) \end{aligned} \tag{1}$$

*Received by the editors June 7, 2000; accepted for publication (in revised form) by R. Brualdi November 13, 2000; published electronically March 7, 2001.
http://www.siam.org/journals/simax/22-4/37332.html

†Department of Mathematics, College of William and Mary, PO Box 8795, Williamsburg, VA 23187-8795 (crjohnso@math.wm.edu).

‡Department of Computer Science, University of Victoria, Victoria, BC, V8W 3P6 Canada (dolesky@csr.uvic.ca). The research of this author was supported in part by an NSERC research grant.

§Department of Mathematics and Statistics, University of Victoria, Victoria, BC, V8W 3P4 Canada (pvdd@math.uvic.ca). The research of this author was supported in part by an NSERC research grant.

is particularly appropriate. This factorization always exists for a nonsingular totally nonnegative matrix, and we study it for more general matrices, a special case of which has already been considered in [4].

A *successively ordered EB (SEB) factorization* of a general n-by-n matrix A is a factorization

$$A = \left(\prod_{k=1}^{n-1} \prod_{j=n}^{k+1} L_j(s_{jk}) \right) D \left(\prod_{k=n-1}^{1} \prod_{j=k+1}^{n} U_j(t_{kj}) \right) \tag{2}$$

in which each parameter s_{jk}, t_{kj} may be zero or nonzero. (We adopt the convention in $\prod$ that factors are written from left to right with the one corresponding to the bottom index of $\prod$ on the left.) For example, if a 4-by-4 matrix A has an SEB factorization, then it is of the form

$$\begin{aligned} A = {} & L_4(s_{41})L_3(s_{31})L_2(s_{21})\ L_4(s_{42})L_3(s_{32})\ L_4(s_{43})\ D \\ & \times U_4(t_{34})\ U_3(t_{23})U_4(t_{24})\ U_2(t_{12})U_3(t_{13})U_4(t_{14}) \end{aligned} \tag{3}$$

in which D is diagonal and some EB factors may not be included.

Example 1.

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

has an EB factorization $A = L_2(-1)L_3(1)L_2(1)L_3(-1)$; see [6, Theorem 11]. However, A has no SEB factorization.

In the event that all possible EB factors are included (with nonzero parameters), the SEB factorization (2) is called *generic*. Matrices with generic SEB factorizations were studied in [4, 5]. For lists α, β from $1, 2, \ldots, n$, the submatrix of the n-by-n matrix A lying in the rows α and columns β is denoted by $A[\alpha|\beta]$. As in [4, (1)] and earlier with different terminology in [5, (2.3)], the n-by-n matrix A is said to have the *consecutive-column (CC) property* if the $\binom{n}{2}$ submatrices $A[r-s+1, \ldots, r|1, \ldots, s]$ are nonsingular, for $1 \le r \le n$ and $1 \le s \le r$, and A has the *consecutive-row (CR) property* if A^T has the CC property. It was shown in [4] that a nonsingular matrix A has a generic SEB factorization if and only if it has the CC and CR properties, and when the factorization exists, it is unique.

Our interest is in the SEB factorization of matrices, but in which not all of the factors $L_j(s_{jk})$ and $U_j(t_{kj})$ are necessarily included (i.e., nongeneric SEB). The following example shows that for a unit lower triangular matrix, all $\binom{n}{2}$ EB factors may not be needed in an SEB factorization with the ordering (2).

Example 2.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = L_3(1)L_2(1)$$

requires only 2 EB factors instead of the generic number 3, and $D = I$. Since s_{32} must be 0 and $L_3(0) = I$, this factor is not included. Note that A does not have the CC property, since $A[2, 3|1, 2]$ is singular.

Unfortunately, nongeneric SEB factorizations of a given matrix (when they exist) are *not* in general unique; it is also natural to consider factorization of singular

matrices (those of [4, 5] are necessarily nonsingular). Thus we narrow somewhat the SEB factorizations that we study in order to understand existence and to achieve uniqueness. We call an SEB factorization (2) an *elimination SEB* ($ESEB$) if it results from the reduction of an n-by-n matrix A to the diagonal matrix D by eliminating nonzero entries via elementary bidiagonal row/column operations performed in the order (1). Each such operation in the elimination is called *a successive row (column) operation*. An elimination process consisting of such operations has been called Neville elimination by some authors. Note that the subscripts on s_{jk} and t_{kj} correspond to the entry being eliminated. If s_{jk} or t_{kj} can be 0 in any factor, then that factor can be I, and it is not included in the $ESEB$ factorization.

Example 3.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & bc & c & 1 \end{bmatrix}$$

with $a, b, c \neq 0$ has an SEB factorization $A = L_4(c)L_2(a)L_3(b)$. However, the $ESEB$ factorization is $A = L_2(a)L_4(c)L_3(b)$. This illustrates the fact that SEB factorizations are not necessarily unique.

Example 4. The singular matrix

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

has the $ESEB$ factorization

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = L_2(1)D.$$

Note that $A = L_2(1)L_3(s_{32})D$ for all values of s_{32} (including $s_{32} = 0$), which is an SEB factorization of A.

In an SEB factorization, the factors L_j (or U_j) occur in "stretches" left to right (right to left). The kth *L-stretch* , $k = 1, \ldots, n-1$, is the product

$$\prod_{j=n}^{k+1} L_j(s_{jk}) = L_n(s_{nk})L_{n-1}(s_{n-1,k}) \cdots L_{k+1}(s_{k+1,k}),$$

and the kth *U-stretch* , $k = 1, \ldots, n-1$, is the product

$$\prod_{j=k+1}^{n} U_j(t_{kj}) = U_{k+1}(t_{k,k+1})U_{k+2}(t_{k,k+2}) \cdots U_n(t_{kn}).$$

For example, if A is a 4-by-4 matrix with an SEB factorization as in (3), then the 1st L-stretch is $L_4(s_{41})L_3(s_{31})L_2(s_{21})$. We sometimes refer to a stretch without specifying L or U. In an $ESEB$ factorization, the parameters s_{jk} (t_{kj}) in the kth stretch are chosen to eliminate the entries in the kth column (row) of A. The smallest indexed L or U that may appear in the kth stretch is $k + 1$ and the largest is n, although some factors may not be included in a stretch in which they are allowed, and in fact a stretch may be empty.

We call an SEB factorization *proper* ($PSEB$) if, in each nonempty stretch, the included EB factors are indexed consecutively and include the smallest index allowed (i.e., if $L_i(s_{ik})$ or $U_i(t_{ki})$, $i > k$, does not appear in the kth stretch, then neither does $L_j(s_{jk})$ or $U_j(t_{kj})$ for all $j > i$). This turns out to be a natural requirement that narrows the possible SEB factorizations from the point of view of elimination, and corresponds to the $ESEB$ factorization for a nonsingular matrix (see section 3).

2. Existence of ESEB factorization. We formulate our characterization of an $ESEB$ factorization in terms of new rank conditions given in the following definition.

DEFINITION 5. *An n-by-n matrix A satisfies the column descending rank condition if for all l such that $1 \le l \le n-1$, for all m such that $0 \le m \le l-1$, and for all p such that $l-m \le p \le n-m-1$,*

$$\text{rank } A[p, \dots, p+m | 1, \dots, l] \ge \text{rank } A[p+1, \dots, p+m+1 | 1, \dots, l]. \tag{4}$$

Matrix A satisfies the *row descending rank condition* if A^T satisfies the column descending rank condition, namely,

$$\text{rank } A[1, \dots, l | p, \dots, p+m] \ge \text{rank } A[1, \dots, l | p+1, \dots, p+m+1], \tag{5}$$

in which the indices are as above.

The relation between the CC property and the column descending rank condition is now considered. An analogous relation holds between the CR property and the row descending rank condition.

Observation 6. Let A be an n-by-n nonsingular matrix. If A has the CC property, then A satisfies the column descending rank condition (and, in fact, all of the rank conditions (4) hold with equality and every submatrix in (4) has full rank).

Proof. All of the square submatrices (the case $m = l-1$) in (4) are included in the CC property and are therefore nonsingular. For each of the inequalities in (4) between the ranks of two nonsquare submatrices, each of the submatrices has full rank, as it consists of one or more complete rows of a nonsingular matrix. Thus all of the submatrices in (4) have full rank, and the result follows. ∎

Example 2 shows that the column descending rank condition may hold without the CC property. The proof of the above observation leads to the following further observation.

Observation 7. If all of the rank conditions in (4) on square submatrices hold with equality and all of the submatrices involved have full rank, then all of the other rank conditions in (4) also hold with equality.

The following lemma, showing that the rank conditions are preserved under a successive row (column) operation during the computation of an $ESEB$ factorization, is central to the proof of our characterization.

LEMMA 8. *Assume that A is an n-by-n matrix with i, r such that $1 \le i \le n-1$, $i \le r \le n-1$ and every entry in A below the main diagonal in columns $1, \dots, i-1$ and entries $a_{r+2,i}, \dots, a_{ni}$ are zero, but $a_{ri} \ne 0$. (Entry $a_{r+1,i}$ is arbitrary.) Let $A' = L_{r+1}(s)A$. Then A' satisfies the column and row descending rank conditions if and only if A satisfies these rank conditions. The conclusion also holds if A^T satisfies the above assumptions on A and $A' = AU_{r+1}(t)$.*

Proof. Since $L_{r+1}(0) = I$, the result is obvious for $s = 0$; thus suppose that $s \ne 0$. Assume that A satisfies the column and row descending rank conditions. Since the submatrices on both sides of the inequality (5) have the same row indices, the row descending rank condition (5) is unchanged by any successive row operation.

Thus A' satisfies the row descending rank condition. To show that A' satisfies the column descending rank condition, let $B = A[p+1,\dots,p+m+1|1,\dots,l]$ and $C = A[p,\dots,p+m|1,\dots,l]$ for $1 \le l \le n-1, 0 \le m \le l-1$, and $l-m \le p \le n-m-1$. By assumption rank $C \ge$ rank B, and it is necessary to show that rank $C' \ge$ rank B', where C' and B' are the corresponding submatrices of A'. Assume that $i \le l$ (otherwise $C' = C$ and $B' = B$, and the result is obvious). Depending on the value of r, there are 5 ways that multiplication by $L_{r+1}(s)$ (i.e., adding multiple s of row r of A to row $r+1$ of A) may change B or C; each is considered separately.

(i) $p+1 \le r \le p+m-1$ with $m \ge 2$ and $1 \le i \le n-3$ (i.e., both rows r and $r+1$ of A are in B and C). The ranks of B and C are unchanged by multiplication by $L_{r+1}(s)$. Thus rank C' = rank $C \ge$ rank B = rank B'.

(ii) $i \le r \le p-2$ or $p+m+1 \le r \le n-1$ (i.e., neither row r nor row $r+1$ of A are in B or C, but they are above C or below B). Multiplication by $L_{r+1}(s)$ leaves B and C unchanged, thus rank $C' \ge$ rank B'.

(iii) $r = p+m$ with $m \ge 1$ (i.e., the bottom rows of B and C are involved). Since the successive row operation is within B, rank B' = rank B. Also $C' = C$. Thus rank $C' \ge$ rank B'.

(iv) $r = p$ (i.e., the top rows of B and C are involved). If $m = 0$, then $C' = C$ with rank $C' = 1 \ge$ rank B'. If $m \ge 1$, then the successive row operation is within C, and so rank C' = rank C. If rank $B' \le$ rank B, then rank $C' \ge$ rank B'. Otherwise rank B' = rank $B+1$ (since a row operation can change the rank by at most one). Let $R_j = A[j|1,\dots,l]$ for $p \le j \le p+m+1$. Since rank B = rank $span\{R_{p+1},\dots,R_{p+m+1}\}$, and rank B' = rank $span\{sR_p+R_{p+1}, R_{p+2},\dots, R_{p+m+1}\}$, it must be that $R_p \notin span\{R_{p+1},\dots,R_{p+m+1}\}$. Also $\{R_{p+1},\dots,R_{p+m+1}\}$ must be linearly dependent, as rank B' = rank $B+1$ implies that rank $B < m+1$. If R_{p+m+1} is not a linear combination of $\{R_{p+1},\dots,R_{p+m}\}$, then there exists R_{p+q}, such that, R_{p+q} is a linear combination of $\{R_{p+q+1},\dots,R_{p+m}\}$, where $1 \le q \le m-1$. Then

$$\text{rank } span\{R_{p+q},\dots,R_{p+m}\} = \text{rank } span\{R_{p+q+1},\dots,R_{p+m+1}\} - 1,$$

which violates the column descending rank condition. Therefore rank $B' \le$ rank B. Finally, if R_{p+m+1} is a linear combination of $\{R_{p+1},\dots,R_{p+m}\}$, then rank B' = rank C, and thus rank $C' \ge$ rank B'.

(v) $r = p-1$ (i.e., the row above C and the top row of C are involved). B remains unchanged, thus $B' = B$. If rank $C' \ge$ rank C, then the result is obvious. So assume that rank C' = rank $C-1$. Note that this implies that $a_{r+1,i} \ne 0$, since by assumption $a_{ri} \ne 0$. Let $B^+ = A[p,\dots,p+m+1|1,\dots,l]$ and $C^+ = A[p-1,\dots,p+m|1,\dots,l]$. Then rank $B^+ = 1 +$ rank B' as column i of B' is zero but $a_{pi} = a_{r+1,i} \ne 0$. By the column descending rank condition, rank $B^+ \le$ rank C^+. Since rank C' = rank $C-1$, it follows that $sR_{p-1} + R_p \in span\{R_{p+1},\dots,R_{p+m}\}$. Thus $R_{p-1} \in span\{R_p,\dots,R_{p+m}\}$, which gives rank C^+ = rank C. Therefore rank $B' <$ rank$B^+ \le$ rank C^+ = rank C = rank $C' + 1$, giving rank $C' \ge$ rank B'.

Collecting (i)–(v), $A' = L_{r+1}(s)A$ satisfies the column descending rank condition.

For the converse, assume that $A' = L_{r+1}(s)A$ satisfies the rank conditions. Then $A = L_{r+1}^{-1}(s)A' = L_{r+1}(-s)A'$. Thus the arguments in the 5 cases above show that A also satisfies the rank conditions. The last statement of the theorem follows by applying the above to A^T. ∎

THEOREM 9. *An n-by-n matrix A has an ESEB factorization if and only if it satisfies the column and row descending rank conditions. Furthermore, if an ESEB factorization exists, then it is unique.*

Proof. Suppose that A satisfies the rank conditions. Let i, r be such that $1 \leq i \leq n-1$, $i \leq r \leq n-1$ and every entry in A below the main diagonal in columns $1, \ldots, i-1$ and entries $a_{r+2,i}, \ldots, a_{ni}$ are zero, but $a_{r+1,i} \neq 0$. The column descending rank condition implies that a_{ri} is nonzero, so a successive row operation using a_{ri} to eliminate $a_{r+1,i}$ can be applied to A yielding A'. Lemma 8 implies that A' satisfies the column and row descending rank conditions. Clearly this procedure can be repeated (a total of at most $\binom{n}{2}$ times) until A is reduced to an upper triangular matrix T that satisfies the row descending rank condition. Similarly, using at most $\binom{n}{2}$ successive column operations, T can be reduced to diagonal form, completing the $ESEB$ factorization. Since the order of elimination is fixed by (1), each parameter s_{jk}, t_{jk} is uniquely determined and so the $ESEB$ factorization is unique.

For the converse, assume that A has an $ESEB$ factorization, written as $A = LDU$, where D is a diagonal matrix, $L = \prod_{k=1}^{n-1} \prod_{j=n}^{k+1} L_j(s_{jk})$ and $U = \prod_{k=n-1}^{1} \prod_{j=k+1}^{n} U_j(t_{kj})$. Thus $D = L^{-1}AU^{-1}$ and D satisfies the rank conditions. Working from column $n-1$, the lower triangular part of A can be reconstructed from D by successive row operations. After each operation the resulting matrix satisfies the conditions of Lemma 8, and thus the rank conditions hold. Similarly, starting from row $n-1$, the upper triangular part of A is reconstructed, and the rank conditions remain satisfied. ∎

The following example illustrates that determinant conditions alone may not suffice to determine when a matrix has an $ESEB$ factorization; cf. the CC and CR properties.

Example 10. Consider the unit lower triangular matrix

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

which satisfies all the column descending rank conditions on *square* submatrices. However, rank $A[3|1,2] = 0 <$ rank $A[4|1,2] = 1$; thus by Theorem 9 A has no $ESEB$ factorization.

3. Relations between SEB factorizations. As we have seen it is possible for some of the parameters to be zero in the $ESEB$ factorization, and the corresponding factors are then not included. In (i) of the following result, it is shown that an $ESEB$ factorization is in fact a $PSEB$ factorization; this was observed (with different terminology) in [5, Theorem 2.2]. In (ii) a necessary rank condition corresponding to a zero parameter is given.

THEOREM 11. *Given the ESEB factorization of an n-by-n matrix A, let* $1 \leq q \leq n-1$, $q+1 \leq i \leq n$ *and assume that* $s_{iq} = 0$ *(thus* $L_i(s_{iq})$ *is not included in the qth L-stretch of the ESEB factorization of A). It follows that*

(i) *if i is the smallest such index, then the qth L-stretch is just* $\prod_{j=i-1}^{q+1} L_j(s_{jq})$ *(in the case* $i = q+1$*, the qth L-stretch has no factors); and*

(ii) $A[i-q+1, \ldots, i|1, \ldots, q]$ *does not have full rank.*

Analogous statements hold for the qth U-stretch.

Proof. From (2), all entries below the main diagonal in the first $q-1$ columns of $\hat{A} = (\prod_{k=1}^{q-1} \prod_{j=n}^{k+1} L_j(s_{jk}))^{-1} A = [\hat{a}_{ij}]$ are 0. The product of the inverses of the factors in the qth L-stretch are used to zero out the entries of $\hat{A}$ in column q below the main diagonal. Since $L_i(s_{iq})$ does not occur in the qth L-stretch, $\hat{a}_{iq}$ must already be zero. Since A has an $ESEB$ factorization, it satisfies the column descending rank

condition (by Theorem 9), and thus so does $\hat{A}$ (by Lemma 8). Therefore, $\hat{a}_{rq} = 0$ for $i+1 \le r \le n$. Thus none of the factors $\prod_{j=n}^{i+1} L_j(s_{jq})$ are included in the qth L-stretch, proving (i).

To prove (ii), suppose that $L_i(s_{iq})$ is not included in the qth L-stretch, but that

$$\text{rank } A[i-q+1,\dots,i|1,\dots,q] = q,$$

i.e., this submatrix has full rank. Since A has an $ESEB$ factorization, A satisfies the column descending rank condition (by Theorem 9). Consider the reduction of A to the diagonal matrix D as in the $ESEB$ factorization of A. After elimination of entries in positions $(n,1),\ (n-1,1),\dots,(i-q+2,1)$, we obtain $A_1 = (\prod_{j=n}^{i-q+2} L_j(s_{j1}))^{-1}A$, in which

$$\text{rank } A_1[i-q+1,\dots,i|1,\dots,q] = q.$$

Thus

$$\text{rank } A_1[i-q+2,\dots,i|2,\dots,q] = q-1,$$

i.e., this submatrix has full rank. Now continue with the elimination of entries of A as in the $ESEB$ factorization, eliminating the entries in positions $(i-q+1,1)\dots,(2,1),\ (n,2),\ (n-1,2),\dots,(i-q+3,2)$, giving

$$A_2 = \left(\prod_{j=i-q+1}^{2} L_j(s_{j1}) \prod_{j=n}^{i-q+3} L_j(s_{j2})\right)^{-1} A_1.$$

After these elementary row operations

$$\text{rank } A_2[i-q+2,\dots,i|2,\dots,q] = q-1,$$

and thus

$$\text{rank } A_2[i-q+3,\dots,i|3,\dots,q] = q-2,$$

which is full rank. Proceeding with the $ESEB$ factorization, after the elimination of all entries below the main diagonal in columns $1,\dots,q-2$ and the elimination of entries in positions $(n,q-1),\ (n-1,q-1),\dots,(i,q-1)$ we obtain the matrix A_{q-1} in which rank $A_{q-1}[i|q] = 1$. Continuing with the $ESEB$ factorization, the elementary row operations used to eliminate the entries in positions $(i-1,q-1),\dots,(q,q-1),\ (n,q),\ (n-1,q),\dots,(i+1,q)$ leave the (i,q) entry of A_{q-1} unchanged. But this implies that $L_i(s_{iq})$ is included in the qth L-stretch of the $ESEB$ factorization of A, which is a contradiction. Thus rank $A[i-q+1,\dots,i|1,\dots,q]$ cannot be full. ∎

The results of Theorem 11 are illustrated in the following example.

Example 12. Let A be the 5-by-5 unit lower triangular matrix with $ESEB$ factorization

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 4 & 4 & 1 \end{bmatrix} = L_5(1)L_4(1)L_3\left(\frac{1}{2}\right)L_2(2)\ L_3\left(\frac{1}{2}\right)\ L_5(2)L_4(1)\ L_5(1).$$

Here the 2nd L-stretch is just $L_3(\frac{1}{2})$ since $s_{52} = s_{42} = 0$, thus $L_5(s_{52})$ and $L_4(s_{42})$ are not included in the 2nd L-stretch of the $ESEB$ factorization (2). Furthermore, rank $A[3,4|1,2] = $ rank $A[5,4|1,2] = 1$, illustrating Theorem 11(ii). However, rank $A[3,4,5|1,2,3] = 2$ (i.e., is not full), but $L_5(s_{53}) = L_5(2)$ is included in the $ESEB$ factorization. Thus the converse of Theorem 11(ii) is not in general true.

By Theorem 11(i), the $ESEB$ factorization of a matrix A is also a $PSEB$ factorization. The following result addresses the converse.

THEOREM 13. *If A is an n-by-n matrix with a $PSEB$ factorization* (2)*, then A also has an $ESEB$ factorization. Furthermore, if A is nonsingular, then any $PSEB$ factorization coincides with the (unique) $ESEB$ factorization.*

Proof. Assume that A has a $PSEB$ factorization. We prove the results by induction for an n-by-n lower triangular matrix A. An analogous proof holds if A is upper triangular, and these two cases give the general results. The theorem is clearly true if $n = 2$. Assume that it is true for lower triangular matrices of order $n-1$, and let A be an n-by-n lower triangular matrix.

If the 1st L-stretch in the $PSEB$ factorization (2) is empty, then $a_{21} = \cdots = a_{n1} = 0$ (but a_{11} may be zero or nonzero). By the induction hypothesis, since a $PSEB$ factorization of $A[2,\ldots,n|2,\ldots,n]$ exists, this leads to an $ESEB$ factorization of A. If A is nonsingular, then $a_{11} \neq 0$ and $A[2,\ldots,n|2,\ldots,n]$ is nonsingular. The second statement of the theorem then follows by the inductive hypothesis. If the 1st L-stretch is nonempty, then it is $F = L_m(s_{m1})L_{m-1}(s_{m-1,1})\cdots L_2(s_{21})$, with $m \geq 2$. Then A can be written as $A = FA'D$, in which either

(a) $a_{11},\ldots,a_{n1}$ are all zero and the first diagonal entry of D is 0,

or

(b) $a_{11},\ldots,a_{m1}$ are all nonzero, but $a_{q1} = 0$ for all $q > m$.

In either case $(A'D)[2,\ldots,n|2,\ldots,n]$ has a $PSEB$ factorization, given from the $PSEB$ factorization of A. Thus, $(A'D)[2,\ldots,n|2,\ldots,n]$ has an $ESEB$ factorization by the induction hypothesis. In the event (a), A clearly has an $ESEB$ factorization (although it need not coincide with the given $PSEB$ factorization). In the event (b), since the inverses of the EB matrices of the 1st L-stretch are exactly the elementary factors that eliminate up the first column, A also has an $ESEB$ factorization, which coincides with the given $PSEB$ factorization if A is nonsingular. ∎

Example 4 shows that the conclusion of Theorem 13 in the nonsingular case is not in general true for a singular matrix, since $A = L_2(1)L_3(s_{32})D$ is a $PSEB$ factorization of A for all values of s_{32}, but the (unique) $ESEB$ factorization is $A = L_2(1)D$.

The following example shows that (for $n \geq 4$) it is possible for a nonsingular matrix A to have an SEB factorization, but not an $ESEB$ factorization (and thus not a $PSEB$ factorization).

Example 14.

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

has an SEB factorization $A = L_3(-1)L_4(1)L_3(1)L_4(-1)$. However, by Theorem 9, A has no $ESEB$ factorization, since rank $A[3|1,2] = 0 <$ rank $A[4|1,2]$, and thus no $PSEB$ factorization by Theorem 13. If the $(4,4)$ entry of A is changed to 0, then the new matrix is singular, and has an SEB factorization $L_3(-1)L_4(1)L_3(1)L_4(-1)D$, with $D = \mathrm{diag}(1,1,1,0)$, but no $ESEB$ or $PSEB$ factorization.

By using the relations (see, e.g., [1, p. 57])

$$L_i(r)L_j(s) = L_j(s)L_i(r), \quad |i-j| \geq 2,$$

and

$$L_i(r)L_{i\pm1}(s)L_i(t) = L_{i\pm1}\left(\frac{st}{r+t}\right)L_i(r+t)L_{i\pm1}\left(\frac{rs}{r+t}\right), \quad r+t \neq 0, \tag{6}$$

an SEB factorization can usually be transformed into a $PSEB$ factorization. For example, if $a+c \neq 0$, then

$$\begin{aligned} L_3(a)L_4(b)L_3(c)L_4(d) &= L_4\left(\frac{bc}{a+c}\right)L_3(a+c)L_4\left(\frac{ab}{a+c}\right)L_4(d) \\ &= L_4\left(\frac{bc}{a+c}\right)L_3(a+c)L_4\left(d+\frac{ab}{a+c}\right), \end{aligned}$$

by using (6). This is a $PSEB$ factorization (with the 1st stretch empty) of

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a+c & 1 & 0 \\ 0 & bc & b+d & 1 \end{bmatrix}.$$

Matrices (such as A in Example 14, in which $a+c=0$) that have an SEB factorization but no $PSEB$ factorization are a set of measure zero in the vector space of n-by-n matrices.

For the generic SEB factorization, the parameters in (2) are given in our notation by the following formulae from [4, Theorems 3.2, 3.3].

THEOREM 15. *Let A be an n-by-n nonsingular matrix with the generic factorization* (2). *Then for $1 \leq k \leq n-1$,*

$$s_{k+1,1} = \frac{a_{k+1,1}}{a_{k1}}, \tag{7}$$

and for $2 \leq q \leq n-1$ and $1 \leq k \leq n-q$,

$$s_{k+q,q} = \frac{1}{P_{kq}} \frac{\det(A[k+1,\ldots,k+q|1,\ldots,q])}{\det(A[k+1,\ldots,k+q-1|1,\ldots,q-1])}, \tag{8}$$

where $P_{kq} = s_{k+q-1,q}\cdots s_{q+1,q}$, with $P_{1q}=1$. Similar formulae hold for $t_{q,k+q}$ with the rows and columns of the minors of A interchanged.

For fixed q, the result of Theorem 11(ii) can be obtained from (7) and (8) only if i is the smallest index such that $s_{iq}=0$. (Note that if $s_{iq}=0$, then (8) is undefined for s_{jq} if $j>i$.)

4. Additional remarks on the ESEB factorization. As remarked in the introduction, if A is a nonsingular totally nonnegative matrix, then it is known to have an $ESEB$ factorization; see [7, 8]. The following observation follows from Theorem 9 and generalizes known facts about the zero pattern of a nonsingular totally nonnegative matrix; see the double echelon form in [2].

Observation 16. If A is a nonsingular totally nonnegative matrix, then A satisfies the column and row descending rank conditions.

It is clear from the definition that if A has an $ESEB$ factorization, then A has an LU factorization. However, the matrix A in Example 1 shows that the converse of this statement is not in general true, since $a_{21} = 0$ but $a_{31} \neq 0$. In fact there is no permutation matrix P such that PA has an $ESEB$ factorization.

We conclude with an inheritance property of the leading principal submatrices of matrices that have an $ESEB$ factorization, which is just like the corresponding property for LU factorization. We denote the leading principal submatrix $A[1, \ldots, w|1, \ldots, w]$ by $A[w]$.

THEOREM 17. *Let A be an n-by-n matrix that has the $ESEB$ factorization* (2). *Then $A[w]$ has the $ESEB$ factorization*

$$A[w] = \left(\prod_{k=1}^{w-1} \prod_{j=w}^{k+1} L_j(s_{jk})[w] \right) D[w] \left(\prod_{k=w-1}^{1} \prod_{j=k+1}^{w} U_j(t_{kj})[w] \right).$$

Proof. From the $ESEB$ factorization (2) of A,

$$A[w] = \left(\left(\prod_{k=1}^{n-1} \prod_{j=n}^{k+1} L_j(s_{jk}) \right) D \left(\prod_{k=n-1}^{1} \prod_{j=k+1}^{n} U_j(t_{kj}) \right) \right) [w].$$

Let $M = \max\{w+1, k+1\}$. In the above matrix product, the factors

$$\prod_{k=1}^{n-1} \prod_{j=n}^{M} L_j(s_{jk}), \quad \prod_{k=n-1}^{1} \prod_{j=M}^{n} U_j(t_{kj}),$$

and the diagonal entries of D in rows $w+1, \ldots, n$ have no effect on this product in rows and columns $1, \ldots, w$, which gives the result. ∎

REFERENCES

[1] A. BERENSTEIN, S, FOMIN, AND A. ZELEVINSKY, *Parameterizations of canonical bases and totally positive matrices*, Adv. Math., 122 (1996), pp. 49–149.

[2] A.S. CRANS, S.M. FALLAT, AND C.R. JOHNSON, *The Hadamard core of the totally nonnegative matrices*, Linear Algebra Appl., to appear.

[3] C.W. CRYER, *Some properties of totally nonnegative matrices*, Linear Algebra Appl., 15 (1976), pp. 1–25.

[4] M. FIEDLER AND T.L. MARKHAM, *Consecutive-column and -row properties of matrices and the Loewner-Neville factorization*, Linear Algebra Appl., 266 (1997), pp. 243–259.

[5] M. GASCA AND J.M. PEÑA, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, M. Gasca and C.A. Micchelli, eds., Kluwer, Norwell, MA, 1996, pp. 109–130.

[6] C.R. JOHNSON, D.D. OLESKY, AND P. VAN DEN DRIESSCHE, *Elementary bidiagonal factorizations*, Linear Algebra Appl., 292 (1999), pp. 233–244.

[7] C. LOEWNER, *On totally positive matrices*, Math Z., 63 (1955), pp. 338–340.

[8] A.M. WHITNEY, *A reduction theorem for totally positive matrices*, J. Anal. Math., 2 (1952), pp. 88–92.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1089–1094

NONSTATIONARY MULTISPLITTINGS WITH GENERAL WEIGHTING MATRICES*

VIOLETA MIGALLÓN†, JOSÉ PENADÉS†, AND DANIEL B. SZYLD‡

Abstract. In the convergence theory of multisplittings for symmetric positive definite (s.p.d.) matrices it is usually assumed that the weighting matrices are scalar matrices, i.e., multiples of the identity. In this paper, this restrictive condition is eliminated. In its place it is assumed that more than one (inner) iteration is performed in each processor (or block). The theory developed here is applied to nonstationary multisplittings for s.p.d. matrices, as well as to two-stage multisplittings for symmetric positive semidefinite matrices.

Key words. iterative methods, linear systems, symmetric positive definite matrices, block methods, parallel algorithms, multisplitting, two-stage, nonstationary

AMS subject classifications. 65F10, 65F15

PII. S0895479800367038

1. Introduction.

Multisplitting methods, first introduced by O'Leary and White [18], developed into an important theoretical tool in the study of parallel block iterative methods for the solution of linear (and nonlinear) systems of equations; see, e.g., [4], [12], [16], and the extensive references therein. In these methods, for the solution of a nonsingular system $Ax = b$, several splittings $A = M_\ell - N_\ell$, $\ell = 1, \ldots, p$, (M_ℓ nonsingular) are used, together with a set of diagonal nonnegative (weighting) matrices E_ℓ, such that they add up to the identity; see Algorithm 1.1 below.

Most of the convergence results obtained with the philosophy of multisplittings throughout the literature relate to nonsymmetric matrices. The reason for this is that in these cases, general convergence results were obtained for quite general weighting matrices thus allowing the study of truly parallel methods (with or without overlap), i.e., methods in which each processor computes an approximation to the solution of a problem which is much smaller than the original problem.

In contrast, most results for general symmetric positive definite (s.p.d.), or more generally, Hermitian positive definite, linear systems require the assumption that weighting matrices are multiples of the identity

$$E_\ell = \alpha_\ell I, \quad \ell = 1, \ldots, p \tag{1.1}$$

(see, e.g., [7], [8], [15], [18]), thus these results have little applicability for analysis of parallel processing. We note here that in [21], condition (1.1) is not present, but the splittings of A have a very special structure. We also mention [6] where a nonstandard multisplitting is used for s.p.d. matrices.

The main reason for the requirement (1.1) becomes apparent with examples found in the literature, e.g., in [7], [18], showing that relaxing (1.1) may lead to a divergent

*Received by the editors February 4, 2000; accepted for publication (in revised form) by D. O'Leary October 19, 2000; published electronically March 7, 2001.

http://www.siam.org/journals/simax/22-4/36703.html

†Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, E-03080 Alicante, Spain (violeta@dccia.ua.es, jpenades@dccia.ua.es). This research was supported by Spanish DGESIC grant PB98-0977.

‡Department of Mathematics, Temple University, Philadelphia, PA 19122-2585 (szyld@math.temple.edu). This author was supported by National Science Foundation grants INT-9521226 and DMS-9973219.

multisplitting method. In this note we prove new convergence theorems which, for the first time, show that one can use the multisplitting idea on s.p.d. matrices without the restriction (1.1), i.e., with quite general weighting matrices. As we show later, the price we pay for this generality is a few extra inner iterations.

In the following version of the multisplitting method, which is a special case of the nonstationary two-stage multisplitting method [20] (see also section 3 for another special case), the sequence $s(\ell,k)$ indicates, e.g., the number of local iterations used to approximate the solution of the ℓth system, at the kth iteration; see also [3]. We call them local iterations, since they correspond to work performed in each processor. In the standard multisplitting method of [18], one has $s(\ell,k)=1$ for all ℓ and k.

ALGORITHM 1.1 (NONSTATIONARY MULTISPLITTING). *Given an initial vector x_0 and a sequence of numbers of local iterations $s(\ell,k)$, $\ell=1,\dots,p$, $k=1,2,\dots$*

For $k=1,2,\dots$, until convergence.
  For $\ell=1$ to p
    $y_{\ell,0}=x_{k-1}$
    For $j=1$ to $s(\ell,k)$
      $M_\ell y_{\ell,j}=N_\ell y_{\ell,j-1}+b$ (1.2)
  $x_k=\sum_{\ell=1}^{p} E_\ell y_{\ell,s(\ell,k)}$.

The iteration matrix at the kth step of this multisplitting method is

$$T_k=\sum_{\ell=1}^{p} E_\ell (M_\ell^{-1}N_\ell)^{s(\ell,k)}, \tag{1.3}$$

i.e., $e_k=T_k e_{k-1}$, where the error at each step is $e_k=x_k-x_\star$ and $x_\star$ is the solution of $Ax=b$. Convergence of the method is obtained for any initial vector x_0 by showing that $H_k\to O$ as $k\to\infty$, where $H_k=T_kT_{k-1}\cdots T_2T_1$; see, e.g., [4].

The strength of these methods stems from having many zeros in the weighting matrices, indicating that only a small number of variables of $y_{\ell,j}$ in (1.2) need to be computed. This is why multisplittings developed into such a valuable tool for the analysis of block methods, with or without overlap. By overlap we mean that a variable received contributions from more than one processor, i.e., that the corresponding diagonal entry is nonzero in more than one weighting matrix; see, e.g., [5], [9], [10], [13].

In this note we present a convergence theorem for (nonstationary) multisplittings (section 2), where the condition (1.1) is not needed. We apply this general theorem to the case of s.p.d. matrices. Then, in section 3, we use this result to prove convergence of two-stage multisplittings for symmetric positive semidefinite linear systems.

2. Convergence with general weighting matrices. We begin with our general convergence result. To that end, consider *any matrix norm* such that the norm of the identity is equal to one. Thus, from the fact that $\sum_{\ell=1}^{p} E_{\ell=I}$, we have that

$$\sum_{\ell=1}^{p}\|E_\ell\|\geq 1. \tag{2.1}$$

Furthermore, if each splitting $A=M_\ell-N_\ell$ is convergent, i.e., if $\rho(M_\ell^{-1}N_\ell)<1$, where ρ is the spectral radius, then, since $\lim_{k\to\infty}\|(M_\ell^{-1}N_\ell)^k\|=0$, given any positive

number $\eta < 1$ there is an integer $\tilde{s} = \tilde{s}(\eta)$ (which also depends on the chosen norm), so that

$$\|(M_\ell^{-1}N_\ell)^s\| \le \eta \text{ for all } \ s \ge \tilde{s}, \ \ \ell = 1, \ldots, p. \tag{2.2}$$

THEOREM 2.1. *Let A be nonsingular, and let every splitting $A = M_\ell - N_\ell$, $\ell = 1, \ldots, p$, be convergent. Given a fixed positive number $\theta < 1$, let $\eta = \theta / (\sum_{\ell=1}^{p} \|E_\ell\|)$. Let $\tilde{s}$ be such that* (2.2) *holds. If the sequence of number of local iterations satisfies $s(\ell, k) \ge \tilde{s}$, $\ell = 1, \ldots, p$, $k = 1, 2, \ldots$, then the nonstationary multisplitting Algorithm* 1.1 *converges to the solution of $Ax = b$ with convergence factor θ.*

Proof. From (1.3) it follows that

$$\|T_k\| \le \sum_{\ell=1}^{p} \|E_\ell\| \|(M_\ell^{-1}N_\ell)^{s(\ell,k)}\| \le \left(\max_\ell \|(M_\ell^{-1}N_\ell)^{s(\ell,k)}\| \right) \sum_{\ell=1}^{p} \|E_\ell\| \le \theta < 1.$$

Thus $\|H_k\| = \|T_k T_{k-1} \cdots T_2 T_1\| \le \theta^k \to 0$ as $k \to \infty$. ∎

It follows from Theorem 2.1 that even if the standard multisplitting algorithm ($s(\ell, k) = 1$) does not converge, the price to pay for convergence is more local iterations. Furthermore, we can have convergence as fast as desired, i.e., we can prescribe a smaller convergence factor θ, and obtain the desired convergence by performing more local iterations to satisfy the corresponding condition (2.2).

Of course, we do not know a priori how many local iterations are needed for condition (2.2) to hold, and thus, Theorem 2.1 can be seen more as a theoretical result than a computational tool. On the other hand, Theorem 2.1 implies that one can experiment by increasing a value of $\tilde{s}$, until convergence is achieved. In fact, our discussion before Theorem 2.1 guarantees that such $\tilde{s}$ exists.

We remark that unlike some results in the literature, here we do not need that the sequence $s(\ell, k)$ go to infinity; see [4] and the references therein. On the contrary, all we need is that this sequence be bounded from below by $\tilde{s}$ defined by (2.2). We also mention that similar theorems exist, with the sequence either going to infinity or bounded for two-stage iterative methods; see [11], [17].

In the remainder of this section we apply Theorem 2.1 specifically to the s.p.d. case, using the A-norm, for which (2.1) holds.

If a matrix A is s.p.d., it induces a vector norm $\|x\|_A = (x^T A x)^{1/2}$. A splitting $A = M - N$ of A is called P-regular if $M^T + N$ is positive definite [19]. The following characterization can be found in [2], [12], or [22].

THEOREM 2.2. *Let A be s.p.d. A splitting $A = M - N$ is P-regular if and only if $\|M^{-1}N\|_A < 1$.*

Thus, condition (2.2) can be easily satisfied for P-regular splittings of an s.p.d. matrix.

COROLLARY 2.3. *Let $A = M_\ell - N_\ell$ be P-regular splittings of the symmetric positive definite matrix A, $\ell = 1, \ldots, p$. Given a fixed positive number $\theta < 1$, let $\eta = \theta / (\sum_{\ell=1}^{p} \|E_\ell\|_A)$. Let $\tilde{s}$ be such that $\|(M_\ell^{-1}N_\ell)^s\|_A \le \eta$, for all $s \ge \tilde{s}$, $\ell = 1, \ldots, p$. If the sequence of number of local iterations satisfies $s(\ell, k) \ge \tilde{s}$, $\ell = 1, \ldots, p$, $k = 1, 2, \ldots$, then the nonstationary multisplitting Algorithm* 1.1 *converges to the solution of $Ax = b$ with convergence factor θ. Furthermore, we have that for each iteration k, $\|T_k\|_A \le \theta < 1$.*

We emphasize that in Corollary 2.3 no condition is imposed on the weighting matrices other than adding to the identity, so that (2.1) holds. In other words, we do not have the restriction (1.1).

In Example 2.3 of [7], where there is no convergence, the two splittings are P-regular, and the smallest integer $\tilde{s}$ for which $\|(M_\ell^{-1}N_\ell)^{\tilde{s}}\|_A < 1/\sum_{\ell=1}^p \|E_\ell\|_A$, is $\tilde{s} = 8$. Thus, for $s(\ell,k) \geq 8$, $\|T_k\|_A < 1$. We mention here also that if $s(1,k) = s(2,k) = 4$, $k = 1, 2, \ldots$, then $\rho(T_k) < 1$, and this is the smallest integer for which this is true.

3. Two-stage multisplittings for symmetric positive semidefinite matrices. In this section we extend the convergence theory of multisplittings with general weighting matrices to two-stage multisplittings, when the coefficient matrix of the linear system is s.p.d. (and in particular our result applies to the symmetric positive semidefinite case). Here we assume that $s(\ell,k) = s(\ell)$, i.e., that the number of inner iterations may change from one (inner) splitting to another (or from block to block), but it is the same for all (outer) iterations; it is a stationary method.

In the nonsingular case, as in Theorem 2.1, convergence of an algorithm was shown by having the iteration matrix having norm less than one. Here, in the singular case, we consider a consistent linear system $Ax = b$. The iteration matrix T has spectral radius equal to one, and convergence to some solution is achieved when the iteration matrix T is convergent, i.e., when the limit $\lim_{k\to\infty} T^k$ exists; see, e.g., [2].

Let $A = M - N$ be the outer splitting, and let $M = F_\ell - G_\ell$, $\ell = 1, \ldots, p$.

ALGORITHM 3.1 (TWO-STAGE MULTISPLITTING). *Given an initial vector x_0, and a sequence of numbers of inner iterations $s(\ell)$, $\ell = 1, \ldots, p$.*

For $k = 1, 2, \ldots$, until convergence.

For $\ell = 1$ to p

$y_{\ell,0} = x_{k-1}$

For $j = 1$ to $s(\ell)$

$$F_\ell y_{\ell,j} = G_\ell y_{\ell,j-1} + N x_{k-1} + b$$

$$x_k = \sum_{\ell=1}^{p} E_\ell y_{\ell,s(\ell)} \ .$$

For this two-stage multisplitting algorithm, it follows, e.g., as in [4], that the iteration matrix is

$$T = \sum_{\ell=1}^{p} E_\ell (F_\ell^{-1}G_\ell)^{s(\ell)} + \left(I - \sum_{\ell=1}^{p} E_\ell (F_\ell^{-1}G_\ell)^{s(\ell)} \right) M^{-1}N. \tag{3.1}$$

For our convergence proof we will use the following two results. The first can be found, e.g., in [1], and the second, e.g., in [2].

LEMMA 3.2. *Given a nonsingular matrix A and a matrix T such that $(I - T)^{-1}$ exists, there exists a unique pair of matrices P and Q, P nonsingular, such that $A = P - Q$ and $T = P^{-1}Q$. The matrices are $P = A(I - T)^{-1}$ and $Q = P - A$.*

THEOREM 3.3. *Let $A = M - N$ be a P-regular splitting of a symmetric matrix A. Then the matrix $M^{-1}N$ is convergent if and only if A is positive semidefinite.*

THEOREM 3.4. *Let A be a symmetric positive semidefinite matrix. Let the splitting $A = M - N$ be such that M is a s.p.d. matrix and N is a positive semidefinite matrix. Let $M = F_\ell - G_\ell$, $\ell = 1, \ldots, p$, be P-regular splittings. Given a fixed positive number $\theta < 1$, let $\eta = \theta / \left(\sum_{\ell=1}^p \|E_\ell\|_M\right)$. Let $\tilde{s}$ be such that*

$$\|(F_\ell^{-1}G_\ell)^s\|_M \leq \eta \quad \text{for all} \quad s \geq \tilde{s},\ \ell = 1, \ldots, p. \tag{3.2}$$

If the numbers of inner iterations satisfies $s(\ell) \geq \tilde{s}$, $\ell = 1, \ldots, p$, then the two-stage multisplitting Algorithm 3.1 converges to a solution of the consistent linear system $Ax = b$ for any initial vector x_0.

Proof. All we need to prove is that T defined in (3.1) is convergent. Observe first that the matrix $S = \sum_{\ell=1}^{p} E_\ell (F_\ell^{-1} G_\ell)^{s(\ell)}$ can be viewed as the iteration matrix of a nonstationary multisplitting method based on the splittings $M = F_\ell - G_\ell$ and $s(\ell, k) = s(\ell),\ k = 1, 2, \ldots,\ \ell = 1, \ldots, p$; cf. (1.3). Furthermore, since $M = F_\ell - G_\ell$, $\ell = 1, \ldots, p$ are P-regular splittings of the s.p.d. matrix M, from Corollary 2.3 it follows that $\|S\|_M < 1$. Then, using Lemma 3.2, this iteration matrix induces a unique splitting $M = P - Q$ such that $P^{-1}Q = S$. Moreover, by Theorem 2.2, this splitting is P-regular. Thus, with these matrices, we have

$$\begin{aligned} T &= P^{-1}Q + (I - P^{-1}Q)M^{-1}N \\ &= P^{-1}(Q + (P - Q)M^{-1}N) = P^{-1}(Q + N). \end{aligned}$$

Thus, the splitting $A = P - (Q + N)$ is a splitting induced by T (this splitting is not unique [1]). Since $P^T + Q$ is positive definite and N is positive semidefinite, $P^T + Q + N$ is positive definite, and thus this splitting is P-regular. Therefore, by Theorem 3.3, T is a convergent matrix and the proof is complete. ∎

We mention that the second part of the proof of Theorem 3.4 resembles the proof of Theorem 2.1 of [14], although the context is different.

As in Corollary 2.3, no special condition is imposed on the weighting matrices. Instead we may need to increase the number of inner iterations so that (3.2) holds.

Acknowledgments. We thank Michele Benzi for helpful comments on an early manuscript. We also thank the referees for their suggestions. These comments and suggestions helped improve our presentation.

REFERENCES

[1] M. Benzi and D. B. Szyld, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.
[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, 3rd ed., Academic Press, New York, 1979. Reprinted by SIAM, Philadelphia, 1994.
[3] R. Bru, L. Elsner, and M. Neumann, *Models of parallel chaotic iteration methods*, Linear Algebra Appl., 103 (1988), pp. 175–192.
[4] R. Bru, V. Migallón, J. Penadés, and D. B. Szyld, *Parallel, synchronous and asynchronous two-stage multisplitting methods*, Electron. Trans. Numer. Anal., 3 (1995), pp. 24–38.
[5] Z.-H. Cao, *Nonstationary two-stage multisplitting methods with overlapping blocks*, Linear Algebra Appl., 285 (1998), pp. 153–163.
[6] Z.-H. Cao and Z.-Y. Liu, *Symmetric multisplitting of a symmetric positive definite matrix*, Linear Algebra Appl., 285 (1998), pp. 309–319.
[7] M. J. Castel, V. Migallón, and J. Penadés, *Convergence of non-stationary multisplitting methods for Hermitian positive definite matrices*, Math. Comp., 67 (1998), pp. 209–220.
[8] J.-J. Climent and C. Perea, *Convergence and comparison theorems for multisplittings*, Numer. Linear Algebra Appl., 6 (1999), pp. 93–107.
[9] A. Frommer and B. Pohl, *A comparison result for multisplittings and waveform relaxation methods*, Numer. Linear Algebra Appl., 2 (1995), pp. 335–346.
[10] A. Frommer, H. Schwandt, and D. B. Szyld, *Asynchronous weighted additive Schwarz methods*, Electron. Trans. Numer. Anal., 5 (1997), pp. 48–61.
[11] A. Frommer and D. B. Szyld, *Asynchronous two-stage iterative methods*, Numer. Math., 69 (1994), pp. 141–153.
[12] A. Frommer and D. B. Szyld, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numer. Math., 83 (1999), pp. 259–278.
[13] M. T. Jones and D. B. Szyld, *Two-stage multisplitting methods with overlapping blocks*, Numer. Linear Algebra Appl., 3 (1996), pp. 113–124.
[14] V. Migallón and J. Penadés, *Convergence of two-stage iterative methods for Hermitian positive definite matrices*, Appl. Math. Lett., 10 (1997), pp. 79–83.
[15] R. Nabben, *A note on comparison theorems of splittings and multisplittings of Hermitian positive definite matrices*, Linear Algebra Appl., 233 (1996), pp. 67–80.

[16] M. Neumann and R. J. Plemmons, *Convergence of parallel multisplitting iterative methods for M-matrices*, Linear Algebra Appl., 88/89 (1987), pp. 559–573.
[17] N. K. Nichols, *On the convergence of two-stage iterative processes for solving linear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 460–469.
[18] D. P. O'Leary and R. E. White, *Multisplittings of matrices and parallel solution of linear systems*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 630–640.
[19] J. M. Ortega, *Numerical Analysis, A Second Course*, Academic Press, New York, 1972. Reprinted by SIAM, Philadelphia, 1990.
[20] D. B. Szyld and M. T. Jones, *Two-stage and multisplitting methods for the parallel solution of linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 671–679.
[21] R. E. White, *Multisplitting of a symmetric positive definite matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 69–82.
[22] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1095–1111

A DECOMPOSITION METHOD FOR POSITIVE SEMIDEFINITE MATRICES AND ITS APPLICATION TO RECURSIVE PARAMETER ESTIMATION*

LIYU CAO† AND HOWARD M. SCHWARTZ†

Abstract. A matrix decomposition method for positive semidefinite matrices based on a given subspace is proposed in this paper. It is shown that any positive semidefinite matrix can be decomposed uniquely into two positive semidefinite parts with specified rank one of which is orthogonal to the subspace. This method is then compared with the rank-additivity decomposition, and the difference as well as the close connection between these two decompositions are given. Finally, the proposed decomposition method is used to develop a new recursive parameter estimation algorithm for linear systems.

Key words. positive semidefinite matrices, matrix decomposition, rank additivity, least squares method, recursive estimation

AMS subject classifications. 15A23, 15A24, 93E12, 93E24

PII. S0895479899364027

1. Introduction.

It is assumed that all matrices involved in this paper have real elements. The identity matrix and the null matrix are designated as I and 0, respectively, and their sizes are determined by the context. For any matrix A, we designate the image and the kernel space of A, respectively, as ImA and Ker$A$.

In this paper, we consider the problem of decomposing a positive semidefinite matrix A into the form $A = B + C$, where B and C are required to be positive semidefinite, and, furthermore, C should satisfy $CV = 0$(or $BV = AV$), where V is a full column rank matrix. The columns of V define a subspace, which is its image space. Therefore, the above decomposition requires that one of the decomposed parts is "orthogonal" to a given subspace. It will be shown that if the rank of B is required to be equal to the dimension of the given subspace, then such a decomposition exists and is unique. It will also be shown that such a decomposition has the *rank-additivity property*, that is, rank(A)=rank(B)+rank(C).

Our motivation for considering such a matrix decomposition method comes from the authors' effort in developing a new recursive parameter estimation algorithm for linear systems [1], where the decomposition is the key in establishing the new algorithm that can overcome the main drawbacks of the well-known exponentially weighted least squares algorithm (see section 4).

In the recent paper by Chu, Funderlic, and Golub [2], rank modifications of semidefinite matrices are analyzed. In particular, the following rank reduction problem is addressed in their paper. Given a positive semidefinite matrix A, seek a matrix B such that $C = A - B$ is positive semidefinite and that rank(C)=rank(A)−rank(B). The sufficient and necessary condition for such a rank reduction is given in their paper. Obviously, the matrix rank reduction problem in the form $A - B$ can be viewed as a matrix decomposition problem: decompose a given matrix A into the form $A = B+C$ so that rank(A)=rank(B)+rank(C). To distinguish this decomposition from the one

*Received by the editors November 16, 1999; accepted for publication (in revised form) by P. Van Dooren November 10, 2000; published electronically March 7, 2001.
http://www.siam.org/journals/simax/22-4/36402.html
†Department of Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada (cao@sce.carleton.ca, schwartz@sce.carleton.ca).

addressed in this paper, we call the former the *rank-additivity decomposition* and the latter the *orthogonal decomposition along a subspace*. It will be shown that (see section 3) although the starting points of these two decomposition are different, they achieve almost the same results in nature.

The most fundamental result for the rank-additivity decomposition is the so-called *symmetric rank-subtractivity lemma*, which is given in [2] (also refer to Theorem 3.1 in this paper). This result is obtained from a general rank-subtractivity lemma, which is proven in [3, Corollary 3.1], where the rank of $A - B$ is characterized for arbitrary A and B. Although the main results in this paper can be proven by using the symmetric rank-subtractivity lemma and related results in [3], we will provide alternative proofs that do not depend on these existing results. We believe such a treatment is useful both theoretically and practically. The proofs provided in section 2 show that the rank-additivity decomposition can be obtained from another starting point: the orthogonal decomposition along a subspace. On the other hand, these proofs are easy to understand because they use only basic principles of matrix analysis which can be found in a standard textbook.

This paper is organized as follows. We begin in section 2 with a description of the orthogonal decomposition along a given subspace and then indicate some basic facts related with the decomposition. We present our main result in Theorem 2.1, which shows that the decomposition exists and is unique. Based on Theorem 2.1, we further get Lemma 2.6 and Lemma 2.9, which show that the decomposition is rank additive and image space additive. In section 3, we compare the orthogonal decomposition along a subspace with the rank-additivity decomposition and show the difference as well as the close connection between these two decompositions. It is proven that given the rank of B(or C) there are infinite pairs of positive semidefinite matrices B and C which satisfy $A = B + C$ and rank(A)=rank(B)+rank(C). It is also shown how to select such a B so that the rank-additivity condition is satisfied. In section 4 an application of the new theoretical contributions presented in sections 2 and 3 is illustrated. The orthogonal decomposition method along a subspace is used to develop a new recursive parameter estimation algorithm and, furthermore, to prove that the new algorithm can overcome the main drawbacks of the widely used exponentially weighted least squares algorithm.

2. Decomposition of nonnegative definite matrices along a given subspace.

2.1. Decomposition of a positive semidefinite matrix. Given an $n \times n$ positive semidefinite matrix A and an m-dimensional subspace S in R^n such that $S \cap \mathrm{Ker}A = 0$, the problem is to decompose A as

$$A = B + C \tag{2.1}$$

in such a way that for any vector $x \in S, x \neq 0$,

$$Cx = 0. \tag{2.2}$$

In other words, it is required that $S \subseteq \mathrm{Ker}C$. Let $v_1, v_2, \ldots, v_m$ be a basis of S. Define the matrix V as follows:

$$V = [\ v_1 \quad v_2 \quad \cdots \quad v_m\ ]. \tag{2.3}$$

Obviously, (2.2) means that

$$CV = 0, \tag{2.4}$$

and hence B satisfies

$$BV = AV. \tag{2.5}$$

Equations (2.4) and (2.5) are equivalent. We can solve one of them to determine the decomposition. Consider the matrix equation (2.5). If there are not any restrictions on B, then, according to Prasolov [4, p. 193], many solutions exist for the equation. Here, we are interested in the symmetric positive semidefinite solutions. In the following, it will be shown that if we seek the positive semidefinite solution to (2.5) with rank m, then such a solution is unique. Furthermore, the solution to (2.4) under the restriction of (2.1) is also positive semidefinite.

Before we present the main results, we should note three facts concerning the decomposition.

Fact 2.1. The condition $S \cap \mathrm{Ker}A = 0$ means that the rank of A is not less than m, the dimension of S.

Proof. Obviously, $u_i = Av_i \neq 0, i = 1, \ldots, m$, and u_i belongs to the image space of A. Consider the linear combination of u_i

$$\begin{aligned} u &= a_1u_1 + a_2u_2 + \cdots + a_mu_m \\ &= A(a_1v_1 + a_2v_2 + \cdots + a_mv_m) \\ &= Av, \end{aligned}$$

where $v = a_1v_1 + a_2v_2 + \cdots + a_mv_m$. Obviously, $v \in S$. From the above equation, we can see that $u = 0$ means $v \in \mathrm{Ker}A$. Since $S \cap \mathrm{Ker}A = 0$, we get $v = 0$. However, since the set of $v_1, \ldots, v_m$ is linearly independent, $v = 0$ means $a_i = 0, i = 1, \ldots, m$. Therefore, we conclude that the set of vectors $u_1, \ldots, u_m$ is linearly independent. Thus there are at least m linearly independent vectors in the image space of A, which indicates the rank of A is not less than m. □

Fact 2.2. Equation (2.5) implies the rank of B is not less than m.

Proof. Refer to Lemma 2.2 in the following; it can be seen that the rank of AV is equal to m. Then from (2.5) we get

$$m = \mathrm{rank}(AV) \leq \min\{\mathrm{rank}(B), \mathrm{rank}(V)\},$$

which means $\mathrm{rank}(B) \geq m$. □

Fact 2.3. Equation (2.4) means $\mathrm{Im}C \subseteq S^{\perp}$, where $S^{\perp}$ denotes the orthogonal complement of S.

Proof. For any vector $x \in \mathrm{Im}C$, there exists a vector y such that $x = Cy$. Then for the matrix V defined by (2.3), we have $x^TV = y^TCV = 0$ (because C is assumed to be symmetric), which means $x \in S^{\perp}$. □

We now give the main results on the positive semidefinite decomposition of a positive semidefinite matrix.

THEOREM 2.1. *Assume that A is an $n \times n$ positive semidefinite matrix with rank $r < n$, and S is an m-dimensional subspace S in R^n such that $S \cap \mathrm{Ker}A = 0$. Given the decomposition* (2.1)*, where B and C are required to be positive semidefinite, and B satisfies* (2.5) *and its rank is equal to the dimension of S, then the decomposition defined in* (2.1) *is unique. Furthermore, the matrices B and C are given by*

$$B = AV(V^TAV)^{-1}V^TA, \tag{2.6}$$

$$C = A - B, \tag{2.7}$$

where V is defined by (2.3).

To prove this theorem, we need the following two lemmas.

LEMMA 2.2. *Assume that the $n \times n$ matrix A is positive semidefinite with rank r, and V is an $n \times m$ matrix ($m \leq r$). Then the matrix $D = V^T AV$ is positive definite if and only if V has full column rank and $KerA \cap ImV = 0$.*

Proof. Obviously, D is positive semidefinite. If V has full column rank, then for any $x \in R^m, x \neq 0$, we have $y = Vx \neq 0$ and $y \in \mathrm{Im}V$. Furthermore, if $\mathrm{Ker}A \cap \mathrm{Im}V = 0$, then $y \notin \mathrm{Ker}A$. Thus we have $AVx = Ay \neq 0$. We know that for the positive semidefinite matrix A, $y^T Ay = 0$ if and only if $Ay = 0$. Therefore,

$$x^T Dx = (Vx)^T A(Vx) = y^T Ay \neq 0. \tag{2.8}$$

Since D is nonnegative definite, from (2.8) we get $x^T Dx > 0$, which shows that D is positive definite.

Conversely, if D is positive definite, then for any $x \in R^m, x \neq 0$, $x^T Dx = (Vx)^T A(Vx) > 0$. Obviously, $y = Vx \neq 0$, which shows that V has full column rank. We can easily see that $y \in \mathrm{Im}V$. From $x^T Dx = y^T Ay > 0$, we get $y \notin \mathrm{Ker}A$. Since the above arguments are applicable to all $x \in R^m$ and hence to all $y \in \mathrm{Im}V$, we conclude that $\mathrm{Ker}A \cap \mathrm{Im}V = 0$. □

Lemma 2.2 ensures that the matrix B defined by (2.6) is well defined.

The following lemma is from [5, p. 406].

LEMMA 2.3. *Assume that P is a positive definite matrix. Then any solution Q to the equation*

$$Q^T Q = P \tag{2.9}$$

is of the form $Q = UP^{\frac{1}{2}}$, where U is an orthogonal matrix and $P^{\frac{1}{2}}$ denotes the unique positive definite square root of P.

Lemma 2.3 is a key in obtaining the unique B.

With the above lemmas, we can now prove Theorem 2.1.

Proof of Theorem 2.1. From the assumption, B is a positive semidefinite matrix with rank m. According to [5, p. 407], B can be written in the form

$$B = X^T X, \tag{2.10}$$

where X is an $m \times n$ matrix with rank m.

Then X should satisfy the following equation:

$$X^T XV = AV. \tag{2.11}$$

Multiplying V^T to the above equation from the left, one gets

$$(XV)^T (XV) = V^T AV. \tag{2.12}$$

By Lemma 2.2, the matrix $V^T AV$ is positive definite. Matrix equation (2.12) has the same form as (2.9). By Lemma 2.3, we get

$$XV = U(V^T AV)^{\frac{1}{2}}, \tag{2.13}$$

where U is an arbitrary orthogonal matrix. By substituting (2.13) into (2.11) we can get

$$X^T = AV(V^T AV)^{-\frac{1}{2}} U^T. \tag{2.14}$$

Thus the positive semidefinite matrix B which satisfies (2.5) is given by

$$B = X^T X = AV(V^T AV)^{-1}V^T A. \tag{2.15}$$

Although X is dependent on the choice of U and hence is not unique, B is independent of U and is uniquely determined by A and V according to (2.15).

From Lemma 2.2, we can see that the rank of AV is m. Then from (2.14) it can be seen that the rank of X is also equal to m. Therefore, the rank of B given by (2.15) is equal to m.

To show the matrix C given by (2.7) is positive semidefinite, we use the approach given in [2, Acknowledgment 1]. Since A is positive semidefinite, it can be written as $A = E^T E$, where E is a positive semidefinite matrix. Then

$$C = A - B = E^T(I - EV(V^T AV)^{-1}V^T E^T)E. \tag{2.16}$$

Noting that the middle factor $I - EV(V^T AV)^{-1}V^T E^T$ is an idempotent matrix and hence is positive semidefinite, then from the above equation one can see that $C = A - B$ is positive semidefinite. □

Remark 2.4. The factorization given by (2.10) is the key in the proof of Theorem 2.1. Although the general solution of matrix equations in the form of (2.5) is available (refer to [5, p. 193]), it is not easy from the general solution to get a positive semidefinite solution with specified rank restriction. By factorizing B into the form (2.10), the original linear matrix equation (2.5) is transformed into the nonlinear matrix equation (2.11). The nonlinear matrix equation enables us to get a positive semidefinite solution with the specified rank and, furthermore, to prove that this positive semidefinite solution is unique.

Remark 2.5. It is worth noting that the decomposition is independent of the choice of the basis of S, which constitutes the columns of V. This can be shown as follows.

Let $w_1, w_2, \ldots, w_m$ be another basis of S, and define the matrix V_1 as

$$V_1 = [\ w_1 \quad w_2 \quad \cdots \quad w_m\ ]. \tag{2.17}$$

Then, according to Theorem 2.1, $B_1 = AV_1(V_1^T AV_1)^{-1}V_1^T A$, $C_1 = A - B_1$ are a pair of matrices that satisfy (2.4) and (2.5). Since $v_i, i = 1, \ldots, m$ and $w_i, i = 1, \ldots, m$ are the basis of S, there exists an invertible matrix W such that $V_1 = VW$. Then we have

$$\begin{aligned} B_1 &= AV_1(V_1^T AV_1)^{-1}V_1^T A \\ &= BVW(W^T V^T AVW)^{-1}W^T V^T A \\ &= BV(V^T AV)^{-1}V^T A = B \end{aligned}$$

and hence $C_1 = C$. Thus the decomposition is uniquely determined by the subspace S and is independent of the choice of its basis.

Theorem 2.1 shows C is a nonnegative definite matrix but does not give its rank. The rank of C is given by the following lemma.

LEMMA 2.6. *The rank of C given by* (2.7) *is equal to $r - m$, where r is the rank of A.*

Remark 2.7. Lemma 2.6 indicates that the decomposition given by Theorem 2.1 has the rank-additivity property, that is, rank(A)=rank(B)+rank(C). In [2] this property is proven based on a general rank-subtractivity lemma given by Cline and

Funderlic [3]. Here, we will give an alternative proof, which uses only standard matrix analysis results.

Proof. For any $x \in S$, we have $Bx = Ax$. For $y \in \mathrm{Ker}A$, we have $Ay = 0$ and $By = AV(V^TAV)^{-1}V^TAy = 0$. Thus, for any vector $x + y \in S+\mathrm{Ker}A$, we have

$$C(x + y) = Ax - Bx + Ay - By = 0. \tag{2.18}$$

This indicates $S+\mathrm{Ker}A \subseteq \mathrm{Ker}C$. Therefore,

$$\begin{aligned} \dim(\mathrm{Ker}C) &\geq \dim(S + \mathrm{Ker}A) \\ &= \dim(S) + \dim(\mathrm{Ker}A) - \dim(S \cap \mathrm{Ker}A) \\ &= m + n - r. \end{aligned} \tag{2.19}$$

From the above inequality, we get

$$\begin{aligned} \mathrm{rank}(C) &= \dim(\mathrm{Im}C) = n - \dim(\mathrm{Ker}C) \\ &\leq n - n - m + r = r - m. \end{aligned} \tag{2.20}$$

On the other hand, from the well-known rank inequality rank $(A = B + C) \leq$ rank(B)+rank(C), we have

$$\mathrm{rank}(C) \geq \mathrm{rank}(A) - \mathrm{rank}(B) = r - m. \tag{2.21}$$

The inequalities (2.20) and (2.21) show that $\mathrm{rank}(C) = r - m$. ∎

Remark 2.8. It should be noted that the conditions for the unique decomposition are (2.4) (or (2.5)) and $\mathrm{rank}(B) = m$. On the other hand, if we require that B satisfies (2.5) (or C satisfies (2.4)) restrict the rank of C, say, according to Lemma 2.6, let $\mathrm{rank}(C) = r - m$, and leave the rank of B free, then the decomposition is not unique. This can be explained as follows.

As in the proof of Theorem 2.1, C can be expressed as

$$C = X^TX,$$

where X is an $(r - m) \times n$ matrix with rank $r - m$. We have $CV = X^TXV = 0$. Since X^T is a full column rank matrix, we get $XV = 0$ or $V^TX^T = 0$. We can see that if the columns of X^T are a set of linearly independent vectors in the subspace $S^\perp$, then X is a solution of $V^TX^T = 0$ and the rank of X is $r - m$. There are many sets of linearly independent vectors in $S^\perp$. Therefore, there are many solutions for the equation $XV = 0$ or $CV = 0$. Thus the decomposition $A = B + C$ is not unique under the conditions $CV = 0$ and $\mathrm{rank}(C) = r - m$. As stated in Fact 2.2, all of the decompositions satisfy $\mathrm{rank}(B) \geq m$. Theorem 2.1 states that the unique decomposition is obtained when the rank of B takes its minimal value m.

For the decomposition described in Theorem 2.1, there are some striking relationships among the image spaces and kernel spaces of A, B, and C. These are demonstrated by the following lemma.

LEMMA 2.9. *Let A, B, and C be the matrices given in Theorem* 2.1. *Then*

$$\mathrm{Im}A = \mathrm{Im}B \oplus \mathrm{Im}C, \tag{2.22}$$

$$\mathrm{Ker}A = \mathrm{Ker}B \cap \mathrm{Ker}C, \tag{2.23}$$

$$R^n = \mathrm{Ker}B + \mathrm{Ker}C. \tag{2.24}$$

Proof. First, it is shown that B and AV have the same image space, that is, $\mathrm{Im}B = \mathrm{Im}AV$. For any vector $x \in \mathrm{Im}B$, there exists $y \in R^n$ such that

$$x = By = AV(V^TAV)^{-1}V^TAy = AVz,$$

where $z = (V^TAV)^{-1}V^TAy \in R^m$. This shows $x \in \mathrm{Im}AV$ and therefore $\mathrm{Im}B \subseteq \mathrm{Im}AV$. On the other hand, assume $x \in \mathrm{Im}AV$. Then there exists a vector $y \in R^m$ such that

$$\begin{aligned} x &= AVy = AV(V^TAV)^{-1}V^TAVy \\ &= Bz, \end{aligned}$$

where $z = Vy \in R^n$. The above equation shows $x \in \mathrm{Im}B$ and therefore $\mathrm{Im}AV \subseteq \mathrm{Im}B$. Thus, we conclude that $\mathrm{Im}B = \mathrm{Im}AV$.

Assume $x \in \mathrm{Im}B \cap \mathrm{Im}C$. Since $\mathrm{Im}B = \mathrm{Im}AV$, x can be written as $x = AVy$, where y is a vector in R^m. On the other hand, from Fact 2.3 we have $x \in S^{\perp}$. Therefore, $x^TV = y^TV^TAV = 0$. But Lemma 2.2 says the matrix V^TAV is positive definite. Therefore, we get $y = 0$ and hence $x = 0$. Then we conclude that

$$\mathrm{Im}B \cap \mathrm{Im}C = 0. \tag{2.25}$$

From $A = B + C$ we get $Ax = Bx + Cx$ for any x, which means

$$\mathrm{Im}A \subseteq \mathrm{Im}B + \mathrm{Im}C. \tag{2.26}$$

From Theorem 2.1 and Lemma 2.6, we have

$$\begin{aligned} \dim(\mathrm{Im}B + \mathrm{Im}C) &= \dim(\mathrm{Im}B) + \dim(\mathrm{Im}C) - \dim(\mathrm{Im}B \cap \mathrm{Im}C) \\ &= m + r - m = r. \end{aligned} \tag{2.27}$$

We know $\dim(\mathrm{Im}A)=r$. Then from (2.26) and (2.27) we get

$$\mathrm{Im}A = \mathrm{Im}B + \mathrm{Im}C. \tag{2.28}$$

Equations (2.25) and (2.28) show (2.22) is true.

For any $x \in \mathrm{Ker}A$, from (2.6) we get $Bx = 0$ and hence $x \in \mathrm{Ker}B$. Furthermore, $Cx = Ax - Bx = 0$ and hence $x \in \mathrm{Ker}C$. Thus we get

$$\mathrm{Ker}A \subseteq (\mathrm{Ker}B \cap \mathrm{Ker}C). \tag{2.29}$$

On the other hand, if $x \in \mathrm{Ker}B \cap \mathrm{Ker}C$, then $Ax = Bx + Cx = 0$. Therefore, we get

$$\mathrm{Ker}B \cap \mathrm{Ker}C \subseteq \mathrm{Ker}A. \tag{2.30}$$

From (2.29) and (2.30), (2.23) follows.

Obviously, $\mathrm{Ker}B + \mathrm{Ker}C \subset R^n$. Furthermore, we can get the dimension of $\mathrm{Ker}B + \mathrm{Ker}C$ as

$$\begin{aligned} \dim(\mathrm{Ker}B + \mathrm{Ker}C) &= \dim(\mathrm{Ker}B) + \dim(\mathrm{Ker}C) - \dim(\mathrm{Ker}B \cap \mathrm{Ker}C) \\ &= n - m + n - (r - m) - \dim(\mathrm{Ker}A) \\ &= 2n - r - (n - r) = n, \end{aligned}$$

which shows that (2.24) is true. □

In section 3, Lemma 2.9 is further developed and it is shown that (2.22), (2.23), and (2.24) are equivalent to the rank-additivity decomposition.

2.2. Decomposition of a positive definite matrix. The results in section 2.1 can be applied to positive definite matrices directly. In the following, some main results on the decomposition of a positive definite matrix along a subspace are given.

From Theorem 2.1 and Lemma 2.6 we can get the following theorem.

THEOREM 2.10. *Given an $n \times n$ positive definite matrix A and an m-dimensional subspace S in R^n, let V be an $n \times m$ matrix whose columns constitute a basis of S. Then there exists a unique pair of positive semidefinite matrices B and C such that $A = B + C$, where B is given by*

$$B = AV(V^TAV)^{-1}V^TA \tag{2.31}$$

and has rank m, and C satisfies $CV = 0$ and has rank $n - m$.

Proof. The proof is the same as that for Theorem 2.1. □

The next result is an analogue of Lemma 2.9.

LEMMA 2.11. *Let A, B, and C be the matrices defined in Theorem* 2.10. *Then*

$$R^n = \mathrm{Im}B \oplus \mathrm{Im}C, \tag{2.32}$$

$$R^n = \mathrm{Ker}B \oplus \mathrm{Ker}C. \tag{2.33}$$

Proof. The proof is the same as that for Lemma 2.9. Since A is positive definite and $\mathrm{Ker}A = 0$, (2.23) and (2.24) in Lemma 2.9 reduce to (2.33). □

3. Rank-additivity decomposition of nonnegative definite matrices. In this section, some insights are given into the rank-additivity decomposition addressed in [2], and a brief comparison with the decomposition developed in this paper is also provided. The rank-additivity decomposition is characterized in terms of the image spaces and the kernel spaces of the related matrices.

First, we rewrite the so-called symmetric rank-subtractivity lemma given in [2, Lemma 2.4] in the following form with a minor modification.

THEOREM 3.1. *Let A, B, and C be positive semidefinite matrices satisfying $A = B + C$. Then* rank (A) =rank(B)+rank(C) *if and only if B or C is in the form $AR(R^TAR)^{-1}R^TA$, where R is a full column rank matrix and satisfies* $\mathrm{Im}R \cap \mathrm{Ker}A = 0$.

Remark 3.2. In the original version of Theorem 3.1 [2, Lemma 2.4] the condition $\mathrm{Im}R \cap \mathrm{Ker}A = 0$ is not included. However, by Lemma 2.2 this condition is necessary and sufficient to ensure that the matrix R^TAR is invertible.

If the condition $\mathrm{Im}R \cap \mathrm{Ker}A = 0$ is satisfied and $B = AR(R^TAR)^{-1}R^TA$, then one can see rank(B)=rank(R), and therefore rank(A)=rank(R)+rank(C). By Fact 2.1 the condition $\mathrm{Im}R \cap \mathrm{Ker}A = 0$ implies that the number of the columns of R is not larger than the rank of A. Therefore, if one wants to seek a rank-additivity decomposition of A under the restriction that the matrix B has rank l, where $l < n$ is a given number, one can do it in the following way. First, select arbitrary l linearly independent vectors which do not belong to $\mathrm{Ker}A$. Second, use these vectors as columns to constitute a matrix R. Then, the matrix $B = AR(R^TAR)^{-1}R^TA$ is the intended matrix.

On the other hand, it is easy to see that there are infinite sets of the l linear independent vectors which do not belong to $\mathrm{Ker}A$. Thus, from Theorem 3.1 and the above arguments, we get the following corollary.

COROLLARY 3.3. *Let A, B, and C be $n \times n$ positive semidefinite matrices such that $A = B + C$. Furthermore, let the rank of B be l. Then there are infinite pairs of matrices B and C which satisfy* rank (A)=rank(B)+rank(C), *and each B is given by*

$$B = AR(R^TAR)^{-1}R^TA, \tag{3.1}$$

where R is an arbitrary $n \times l$ full column rank matrix satisfying $\mathrm{Im}R \cap \mathrm{Ker}A = 0$.

Corollary 3.3 shows that with only the rank-additivity condition, the pair of B and C cannot be determined uniquely. To determine B and C uniquely, an additional condition is necessary. From Theorem 2.1, it can be seen that the additional condition to determine B and C uniquely is given by a subspace S, which satisfies $\mathrm{Ker}A \cap S = 0$. Furthermore, the matrix R is determined by S in the sense of $\mathrm{Im}R = S$. With such a subspace, from Theorem 2.1 and Remark 2.5 it can be seen that the rank-additivity decomposition can be determined uniquely. Therefore, Theorem 2.1 determines a *unique* rank-additivity decomposition of a given positive semidefinite matrix, while Theorem 3.1 gives the *general* form of the rank-additivity decomposition.

The next result shows that the rank-additivity decomposition can be characterized in terms of the image spaces and the kernel spaces of the involved matrices.

THEOREM 3.4. *Let A, B, and C be $n \times n$ positive semidefinite matrices such that $A=B+C$. Then the following are equivalent:*

(i) $\mathrm{rank}(A)=\mathrm{rank}(B)+\mathrm{rank}(C)$,

(ii) $\mathrm{Im}A = \mathrm{Im}B \oplus \mathrm{Im}C$,

(iii) $\mathrm{Ker}A=\mathrm{Ker}B \cap \mathrm{Ker}C$ *and* $\dim(\mathrm{Ker}B+\mathrm{Ker}C)=n$.

Proof. For the proof that (i) leads to (ii) and (iii), refer to the proof of Lemma 2.9. Here we need to prove that (ii) leads to (i) and (iii) leads to (i).

First we prove that (ii) leads to (i). If $\mathrm{Im}A=\mathrm{Im}B \oplus \mathrm{Im}C$, then we have $\mathrm{Im}A=\mathrm{Im}B+\mathrm{Im}C$ and $\mathrm{Im}B \cap \mathrm{Im}C=0$. Therefore,

$$\mathrm{rank}(A) = \dim(\mathrm{Im}A) = \dim(\mathrm{Im}B + \mathrm{Im}C). \tag{3.2}$$

However, we have

$$\begin{aligned} \dim(\mathrm{Im}B + \mathrm{Im}C) &= \dim(\mathrm{Im}B) + \dim(\mathrm{Im}C) - \dim(\mathrm{Im}B \cap \mathrm{Im}C) \\ &= \mathrm{rank}(B) + \mathrm{rank}(C) - 0. \end{aligned} \tag{3.3}$$

From (3.2) and (3.3) we get (i).

Next, we prove (iii) leads to (i). From (iii) we get

$$\begin{aligned} n &= \dim(\mathrm{Ker}B + \mathrm{Ker}C) \\ &= \dim(\mathrm{Ker}B) + \dim(\mathrm{Ker}C) - \dim(\mathrm{Ker}B \cap \mathrm{Ker}C) \\ &= n - \mathrm{rank}(B) + n - \mathrm{rank}(C) - \dim(\mathrm{Ker}A) \\ &= 2n - \mathrm{rank}(B) - \mathrm{rank}(C) - n + \mathrm{rank}(A). \end{aligned}$$

From the equation above, we see that (i) is true. □

4. Application to recursive parameter estimation algorithm. In this section, we will use the orthogonal decomposition method developed in section 2 to derive a new recursive parameter estimation algorithm, which can overcome the main drawbacks of the widely used exponentially weighted least squares algorithm, that is, the algorithm gain increases unboundedly when the input is not sufficiently excited. Although this algorithm was originally presented in reference [1], it is derived here to illustrate the practical use of the original theoretical contributions developed in sections 2 and 3 of this paper. Here, we also provide some new results, such as the modified definition of persistency of excitation, Lemmas 4.4 and 4.5. These results are important in proving one of the key properties of the algorithm (refer to Theorem 4.3).

In adaptive systems such as adaptive control and adaptive signal processing, it is necessary to identify systems' parameters on-line in order to track the time-varying system dynamics and to keep the whole system adaptive. This leads to the world of recursive parameter estimation. For a detailed discussion on recursive parameter estimation methods, refer to [6]. One of the most popular recursive parameter estimation methods is the exponentially weighted recursive least squares algorithm. To explain how this algorithm works, consider a dynamic system (whose parameters are to be estimated) described by the following linear difference equation:

$$y(t) + a_1 y(t-1) + \cdots + a_p y(t-p) = b_1 u(t-1) + \cdots + b_q u(t-q) + e(t), \tag{4.1}$$

where $u(t)$ and $y(t)$ are the input and output of the system sampled at time t, and $e(t)$ is a disturbance term. Let

$$\varphi(t) = [-y(t-1) \cdots -y(t-p) u(t-1) \cdots u(t-q)]^T, \tag{4.2}$$

$$\theta = [a_1 \cdots a_p b_1 \cdots b_q]^T. \tag{4.3}$$

The vector $\varphi(t)$ represents the data measured up to $t-1$, while θ represents the unknown parameters which may be time-varying. Then (4.1) can be rewritten as

$$y(t) = \varphi^T(t)\theta + e(t). \tag{4.4}$$

This equation describes the measured output $y(t)$ as a product of the measured data vector $\varphi(t)$ and the unknown parameter vector θ plus a disturbance. The task of parameter estimation is to determine θ from the measurements $y(t)$ and $\varphi(t)$ based on a criterion function. One of the most popular ways to do this is to minimize the following criterion function:

$$V_t(\theta) = \sum_{k=1}^{t} \mu^{t-k} (y(k) - \varphi^T(k)\theta)^2, \tag{4.5}$$

where $\mu < 1$ is called the forgetting factor. Since the square of the error term $y(t) - \varphi^T(t)\theta$ is exponentially weighted, the corresponding algorithm is called the exponentially weighted least squares method.

The solution to the exponentially weighted least squares problem can be written in a recursive form as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + R^{-1}(t)\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)], \tag{4.6}$$

$$R(t) = \mu R(t-1) + \varphi(t)\varphi^T(t), \tag{4.7}$$

where $\hat{\theta}(t)$ denotes the estimated parameter vector at time t, and $R(t)$ is called the information matrix and plays an important role in determining the algorithm's performance. Equation (4.6) is called the parameter update equation and (4.7) is called the information matrix update equation. Equation (4.7) can be rewritten as

$$R(t) = \mu^t R(0) + \sum_{k=1}^{t} \mu^{t-k} \varphi(k)\varphi^T(k), \tag{4.8}$$

where $R(0)$ is the initial value of $R(t)$, which is necessary to start the algorithm.

In order to ensure the algorithm is well defined, $R(t)$ must be positive definite for all t. Therefore, $R(0)$ should be chosen as a positive definite matrix. As has

been shown in [10], a sufficient condition for $R(t)$ to be positive definite is the so-called persistency of excitation of the sequence $\varphi(t)$. The concept of the persistency of excitation is a key in this section and its definition is given as follows [8].

DEFINITION 4.1 (definition of persistency of excitation). *A sequence $x(t) \in R^n$ is said to be persistently exciting in N steps if there exists a positive number a such that*

$$\sum_{k=t+1}^{t+N} x(k)x^T(k) \geq aI \tag{4.9}$$

for all t.

The above definition states that R^n can be spanned by $x(t)$ uniformly in N steps when $x(t)$ is persistently exciting. In the other words, the sum of N matrices, $\sum_{k=t+1}^{t+N} x(k)x^T(k)$, is a full rank matrix when $x(t)$ is persistently exciting. Therefore, if the rank of $\sum_{k=t+1}^{t+N} x(k)x^T(k)$ is less than n, then $x(t)$ is not persistently exciting. Here, in order to handle both persistent excitation and nonpersistent excitation in a unified way, a modified version of the above definition is given, where the *order* of persistency of excitation is introduced.

DEFINITION 4.2 (modified definition of persistency of excitation). *A sequence $x(t) \in R^n$ is said to be persistently exciting of order $m (m \leq n)$ in N steps if*

(1)

$$\max_N \left[\operatorname{rank} \left(\sum_{k=t+1}^{t+N} x(k)x^T(k) \right) \right] = m \quad \text{for all } t; \tag{4.10}$$

(2) *there exists a positive number a such that all nonzero eigenvalues of $\sum_{k=t+1}^{t+N} x(k)x^T(k)$ are not less than a.*

By using the order of persistent excitation, most widely-used excitation signals in adaptive systems are included in the above definition. In addition, the concepts of excited subspace and unexcited subspace of a sequence are closely connected to its order of persistent excitation. When the order of a persistently exciting sequence $x(t)$ in R^n is less than n, then there exists a subspace S in R^n such that for any $u \in S, u^T x(t) = 0$ for all t. We call this subspace the unexcited subspace of $x(t)$ and its orthogonal complement the excited subspace. When the unexcited subspace of the sequence $x(t)$ is not 0, we say that $x(t)$ is not *sufficiently* exciting.

Equation (4.8) shows that the old data $\varphi(k)$ is forgotten according to the exponential function $\mu^{t-k}, k < t$; therefore, this algorithm is also called the exponentially forgetting recursive least squares algorithm. To simplify the notation, we call it the EFRLS algorithm. The forgetting mechanism is necessary to track time-varying parameters. This can be shown as follows. Assume that the sequence $\varphi(t)$ is sufficiently exciting; that is, it satisfies (4.9). Then without forgetting($\mu = 1$), from (4.8) it can be seen that all of the elements of $R(t)$ will tend to infinity and hence the algorithm gain $R^{-1}(t)\varphi(t)$ will tend to zero if $\varphi(t)$ is bounded. In such a case, $\hat{\theta}$ will tend to a constant and the algorithm will eventually turn itself off.

In practice, it is not guaranteed that the sequence $\varphi(t)$ satisfies the condition (4.9). When $\varphi(t)$ is not sufficiently exciting, a phenomenon known as estimator windup occurs in the EFRLS algorithm [7]. In such a situation, no matter how large a number t takes, the second term in (4.8) cannot be a positive definite matrix. The first term in (4.8) will tend to a zero matrix exponentially as $t \to \infty$. Therefore, some

eigenvalues of $R(t)$ will degenerate and the algorithm gain $R^{-1}(t)\varphi(t)$ will tend to be unbounded. Estimator windup is unacceptable because it makes the algorithm very sensitive to noise and thus the estimation may be completely unreliable. Therefore, for the algorithm to be well behaved, a necessary condition is that there exists a positive constant β such that

$$R(t) \geq \beta I. \tag{4.11}$$

Many methods have been suggested to overcome the above drawback of the EFRLS algorithm. For a survey on these methods refer to [9]. Here we will derive a new algorithm based on the decomposition method proposed in section 2. It will be shown that the new algorithm satisfies condition (4.11).

It can be observed from (4.6) that estimator windup in the EFRLS algorithm is due to the fact that the forgetting operation is applied to all elements in $R(t-1)$ through the forgetting factor μ. When $\varphi(t)$ is not sufficiently exciting, then some forgotten data in $R(t-1)$ cannot be compensated by $\varphi(t)$, and eventually some eigenvalues of the information matrix will tend to zero. This fact motivates us to derive an algorithm which forgets only a part of $R(t-1)$ which can be compensated by the new data $\varphi(t)$. For such a strategy it is necessary to decompose $R(t-1)$ into two parts before performing forgetting. In the following, it will be shown that the matrix decomposition method addressed in section 2 is suitable for such a strategy and the estimator windup phenomenon disappears in the resulting algorithm.

Under the direction of the above considerations, before forgetting is applied, $R(t-1)$ is divided into two parts as

$$R(t-1) = R_1(t-1) + R_2(t-1) \tag{4.12}$$

and $R_1(t-1)$ is required to satisfy the equation

$$R_1(t-1)\varphi(t) = 0, \quad \varphi(t) \neq 0, \tag{4.13}$$

which means that $\varphi(t)$ is in the kernel space of $R_1(t-1)$. If we further require that the rank of $R_1(t-1)$ is $n-1$ and the rank of $R_2(t-1)$ is 1, where $n = p+q$ is the size of $R(t-1)$, then by Theorem 2.10 we can get

$$R_2(t-1) = \frac{1}{\varphi^T(t)R(t-1)\varphi(t)}[R(t-1)\varphi(t)][R(t-1)\varphi(t)]^T \tag{4.14}$$

and

$$R_1(t-1) = R(t-1) - R_2(t-1), \tag{4.15}$$

where $R_1(t-1)$ and $R_2(t-1)$ are nonnegative definite. Performing forgetting only on $R_2(t-1)$, the update equation for the information matrix becomes

$$R(t) = R_1(t-1) + \mu R_2(t-1) + \varphi(t)\varphi^T(t). \tag{4.16}$$

Substituting (4.14) and (4.15) into (4.16), we get

$$R(t) = F(t)R(t-1) + \varphi(t)\varphi^T(t), \tag{4.17}$$

where $F(t)$ is a matrix and is given by

$$F(t) = I - (1-\mu)\alpha(t)R(t-1)\varphi(t)\varphi^T(t), \tag{4.18}$$

$$\alpha(t) = \frac{1}{\varphi^T(t)R(t-1)\varphi(t)}. \tag{4.19}$$

Here, in order to ensure that $\alpha(t)$ is well defined, we assume that $\varphi(t) \neq 0$. Equation (4.17) is the new update equation for $R(t)$. Comparing (4.17) with (4.7), we find that the forgetting factor μ in the EFRLS algorithm has been replaced by the matrix $F(t)$ in the new algorithm. Therefore, $F(t)$ is called the forgetting matrix. With $F(t)$, the various eigenvalues of $R(t-1)$ are forgotten with different scaling, which is in sharp contrast to the EFRLS algorithm. Assume $R(0) > 0$; then it can be shown that $R(t)$ given in (4.17) is positive definite for all t.

In the following it will be shown that even when $\varphi(t)$ is not sufficiently exciting condition (4.11) is also satisfied.

THEOREM 4.3. *Assume that $R(0) > 0$. Then the information matrix given by* (4.17) *satisfies the boundedness condition* (4.11) *when $\varphi(t)$ is not sufficiently exciting.*

Theorem 4.3 also appeared in [1]. However, the proof of Theorem 4.3 presented here is more complete than that presented in [1], and Lemmas 4.4 and 4.5 are new contributions.

The key idea to proving Theorem 4.3 is to decompose the information matrix along the excited subspace of $\varphi(t)$. In the excited subspace, $\varphi(t)$ behaves just like a sufficiently exciting signal, and therefore the known result in the case of sufficient excitation (refer to [10]) is applicable.

To prove the theorem, we need the following preliminary lemmas.

LEMMA 4.4. *Assume that A is a positive semidefinite matrix, u is a vector satisfying $Au \neq 0$, and $\rho < 1$ is a scalar. Then the following matrix*

$$B = A - \rho \frac{Auu^T A}{u^T Au} \tag{4.20}$$

is positive semidefinite and has the same rank as that of A.

Proof. The matrix B can be rewritten as

$$B = B_1 + (1-\rho)\frac{Auu^T A}{u^T Au}, \tag{4.21}$$

where B_1 is given by

$$B_1 = A - \frac{Auu^T A}{u^T Au}. \tag{4.22}$$

From Theorem 2.1 we know that B_1 is positive semidefinite. Therefore, B is also positive semidefinite.

Assume that the vector $x \in \mathrm{Ker}B$. Then from (4.20) we have

$$\begin{aligned} Bx &= Ax - \rho \frac{u^T Ax}{u^T Au} Au \\ &= A\left(x - \rho \frac{u^T Ax}{u^T Au} u\right) = 0. \end{aligned} \tag{4.23}$$

From the above equation we see that the vector $x - \rho \frac{u^T Ax}{u^T Au} u$ belongs to $\mathrm{Ker}A$. Therefore, x can be written in the form

$$x = y + au, \tag{4.24}$$

where $y \in \mathrm{Ker}A$ and a is a scalar. From (4.24) and (4.20) we get

$$\begin{aligned} Bx &= aAu - a\rho Au \\ &= a(1-\rho)Au = 0. \end{aligned} \tag{4.25}$$

However, we know that $\rho \neq 1$ and $Au \neq 0$. Therefore, the equation above indicates $a = 0$ and hence $x = y$. Thus we get $\mathrm{Ker}B \subseteq \mathrm{Ker}A$. On the other hand, from (4.20) we get $\mathrm{Ker}A \subseteq \mathrm{Ker}B$. Then we can conclude that B and A have the same kernel space and therefore the same rank. □

LEMMA 4.5. *Assume that the sequence $x(t) \in R^n$ is persistently exciting of order $m(< n)$ in N steps and W is an $m \times n$ matrix whose rows constitute a base of the excited subspace of $x(t)$. Then the sequence $y(t) = Wx(t) \in R^m$ is persistently exciting of order m in N steps (sufficiently exciting).*

Proof. From Definition 4.2 we know that in the vectors set $x(k), x(k+1), \ldots, x(k+N)$($k$ is an arbitrary integer) there exist m vectors which are linearly independent. Let these vectors be denoted by $x(k_i), i = 1, 2, \ldots, m$, and let $y(k_i) = Wx(k_i)$. Consider the linear combinations of $y(k_i)$ given by

$$\begin{aligned} &c_1 y(k_1) + \cdots + c_m y(k_m) \\ &= W(c_1 x(k_1) + \cdots + c_m x(k_m)), \end{aligned}$$

where $c_i, i = 1, \ldots, m$, are scalar. Since $x(k_i), i = 1, 2, \ldots, m$, are linearly independent and the rows of W are a base of the subspace spanned by $x(k_i), i = 1, 2, \ldots, m$, we can see that the above linear combination cannot be zero unless $c_i = 0$ for all i. Therefore, the set $y(k_i), i = 1, 2, \ldots, m$, are linearly independent. In other words, the matrix

$$\sum_{k=t+1}^{t+N} y(k) y^T(k) \tag{4.26}$$

has full rank m for all t, which shows that $y(t) = Wx(t)$ is persistently exciting of order m. □

Proof of Theorem 4.3. Assume that $\varphi(t)$ is persistently exciting of order m, $m < n$. We denote the excited subspace of $\varphi(t)$ by ϕ. The dimension of ϕ is m. The orthogonal complement of ϕ, denoted by $\phi^{\perp}$, is the unexcited subspace.

The update equation for the information matrix can be rewritten as

$$R(t) = R(t-1) - (1-\mu)\alpha(t) R(t-1)\varphi(t)\varphi^T(t) R(t-1) + \varphi(t)\varphi^T(t). \tag{4.27}$$

Therefore, we have

$$R(1) = R(0) - (1-\mu)\alpha(1) R(0)\varphi(1)\varphi^T(1) R(0) + \varphi(1)\varphi^T(1). \tag{4.28}$$

Assume $R(0)$ is positive definite; then, according to Theorem 2.10, it can be decomposed along the excited subspace ϕ into two parts,

$$R(0) = R_o(0) + R_p(0), \tag{4.29}$$

where $R_o(0)$ is orthogonal to the excited space in the following sense:

$$R_o(0)\varphi(t) = 0 \quad \text{for all } \varphi(t) \in \phi \tag{4.30}$$

and is positive semidefinite with rank $n - m$. R_p is also positive semidefinite with rank m.

Equation (4.28) can be rewritten as

$$\begin{aligned} R(1) &= R_o(0) + R_p(0) - (1-\mu)\alpha(1) R_p(0)\varphi(1)\varphi^T(1) R_p(0) \\ &\quad + \varphi(1)\varphi^T(1) \qquad (4.31) \\ &= R_o(1) + R_p(1), \qquad (4.32) \end{aligned}$$

where

$$R_o(1) = R_o(0), \tag{4.33}$$

$$R_p(1) = R_p(0) - (1-\mu)\alpha(1)R_p(0)\varphi(1)\varphi^T(1)R_p(0) + \varphi(1)\varphi^T(1). \tag{4.34}$$

The first two terms in (4.34) are in the same form as that of the matrix B in Lemma 4.4. Then from Lemma 4.4 it can be seen that the matrix $R_p(1)$ is positive semidefinite and its rank is not less than m.

Thus generally we have

$$R(t) = R_o(t) + R_p(t), \tag{4.35}$$

$$R_o(t) = R_o(t-1) = R_o(t-2) = \cdots = R_o(0), \tag{4.36}$$

$$\begin{aligned} R_p(p) = {} & R_p(t-1) - (1-\mu)\alpha(t)R_p(t-1)\varphi(t)\varphi^T(t)R_p(t-1) \\ & + \varphi(t)\varphi^T(t). \end{aligned} \tag{4.37}$$

Define the following matrix:

$$U = [\ U_1 \quad U_2\ ], \tag{4.38}$$

where U_1 is an $n \times m$ matrix whose columns are the orthonormal basis of the excited subspace ϕ, and U_2 is an $n \times (n-m)$ matrix whose columns are the orthonormal basis of the unexcited subspace $\phi^{\perp}$. U is an orthogonal matrix.

One can get

$$U^T R_o(0)U = \begin{bmatrix} 0 & 0 \\ 0 & U_2^T R_o(0)U_2 \end{bmatrix}, \tag{4.39}$$

$$U^T \varphi(t)\varphi^T(t)U = \begin{bmatrix} \psi(t)\psi^T(t) & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.40}$$

where $\psi(t) = U_1^T\varphi(t)$ is an $m \times 1$ column vector. By setting $x(t) = \varphi(t), W = U_1^T$ in Lemma 4.5 we can see that $\psi(t)$ is sufficiently exciting.

Thus we have

$$S(t) = U^T R(t)U = \begin{bmatrix} 0 & 0 \\ 0 & U_2^T R_o(0)U_2 \end{bmatrix} + U^T R_p(t)U. \tag{4.41}$$

We need to show that as $t \to \infty$, the eigenvalues of $S(t)$ and hence $R(t)$ keep bounded from below away from zero. For this purpose, recall the update equation in the EFRLS algorithm

$$R_p'(t) = \mu R_p'(t-1) + \varphi(t)\varphi^T(t). \tag{4.42}$$

Here $R_p'(t)$ corresponds to $R_p(t)$ in (4.37). Assume that the recursive equations (4.37) and (4.42) have the same initial condition, that is, $R_p(0) = R_p'(0) = R_{p0} \geq 0$. Then it can be seen that

$$R_p(1) - R_p'(1) = (1-\mu)\left(R_{p0} - \frac{R_{p0}\varphi(1)\varphi^T(1)R_{p0}}{\varphi^T(1)R_{p0}\varphi(1)}\right) \geq 0. \tag{4.43}$$

That is, $R_p(1) \geq R_p'(1)$. Similarly, in general we have

$$R_p(t) \geq R_p'(t). \tag{4.44}$$

In the following, it is shown how $R'_p(t)$ changes as $t \to \infty$. We have

$$U^T R'_p(t)U = \mu U^T R'_p(t-1)U + U^T \varphi(t)\varphi^T(t)U \tag{4.45}$$

$$= \mu \begin{bmatrix} U_1^T R'_p(t-1)U_1 & U_1^T R'_p(t-1)U_2 \\ U_2^T R'_p(t-1)U_1 & U_2^T R'_p(t-1)U_2 \end{bmatrix} + \begin{bmatrix} \psi(t)\psi^T(t) & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.46}$$

From the equation above it can be seen that when $\mu < 1$ and t is sufficiently large, $U^T R'_p(t)U$ can be approximately expressed as

$$U^T R'_p(t)U \approx \begin{bmatrix} \mu U_1^T R'_p(t-1)U_1 + \psi(t)\psi^T(t) & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.47}$$

The equation above shows that the update equation for $U_1^T R_p(t)U_1$ is the same as that of the EFRLS algorithm. The vector $\psi(t)$ acts as a sufficiently exciting signal. From the known result obtained in [10], we can see that as $t \to \infty$ the eigenvalues of $U_1^T R'_p(t)U_1$ keep bounded from below by a positive number.

From (4.41) and (4.44), we have

$$S(t) = U^T R(t)U \geq \begin{bmatrix} 0 & 0 \\ 0 & U_2^T R_o(0)U_2 \end{bmatrix} + U^T R'_p(t)U. \tag{4.48}$$

Thus when t is sufficiently large we can get

$$S(t) \geq \begin{bmatrix} 0 & 0 \\ 0 & U_2^T R_o(0)U_2 \end{bmatrix} + \begin{bmatrix} \mu U_1^T R'_p(t-1)U_1 + \psi(t)\psi^T(t) & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.49}$$

The columns of U_2 belong to $\phi^\perp$; therefore, $R_o(0)U_2 \neq 0$ and $U_2^T R_o(0)U_2$ is positive definite. The inequality (4.49) proves that as $t \to \infty$ all eigenvalues of $S(t)$ and hence $R(t)$ keep bounded from below by a positive number. □

Finally, the computational complexity of the new algorithm is discussed briefly. In order to avoid matrix inversion computation in (4.6), we need to rewrite the update equation (4.17) in terms of the inverse of $R(t)$. Let $P(t) = R^{-1}(t)$. By applying the matrix inversion lemma to (4.17), we can get

$$P(t) = \bar{P}(t-1) - \frac{\bar{P}(t-1)\varphi(t)\varphi^T(t)\bar{P}(t-1)}{1 + \varphi^T(t)\bar{P}(t-1)\varphi(t)}, \tag{4.50}$$

where $\bar{P}(t-1)$ is defined by the following equations:

$$\bar{P}(t-1) = \bar{R}^{-1}(t-1) = P(t-1)F^{-1}(t)$$

$$= P(t-1) + \frac{1-\mu}{\mu}\frac{\varphi(t)\varphi^T(t)}{\varphi^T(t)R(t-1)\varphi(t)}. \tag{4.51}$$

From (4.6), (4.50), and (4.51), we see that there are three matrices ($P(t)$, $\bar{P}(t)$, and $R(t)$) which need to be updated at each step. In terms of matrix computations, the computational requirement is three times as much as that of the EFRLS algorithm, where only $P(t)$ needs to be updated. This is the price paid for performing forgetting only in the excited subspace. This also motivates further effort to improve the algorithm's complexity.

5. Conclusions. In this paper, the problem of orthogonally decomposing a positive semidefinite matrix A along a given subspace into the form $A = B + C$ has been analyzed. It has been proven that when the rank of B is required to be equal to the dimension of the given subspace, then such a decomposition is unique and has the rank-additivity property $\text{rank}(A) = \text{rank}(B) + \text{rank}(C)$. The difference and close connection between this decomposition and the existing rank-additivity decomposition have been discussed. It has been shown that there are infinite pairs of matrices which have the rank-additivity property. In addition, the rank-additivity decomposition has been characterized in terms of the image space and the kernel space of the involved matrices, which has given some new insights into the rank-additivity property. As an application example, a new recursive parameter estimation algorithm has been developed based on the proposed matrix decomposition method. This algorithm can overcome the main drawbacks of the widely used exponentially weighted least squares algorithm.

REFERENCES

[1] L. Cao and H. Schwartz, *A directional forgetting algorithm based on the decomposition of the information matrix*, Automatica J. IFAC, 36 (2000), pp. 1725–1731.

[2] M. T. Chu, R. E. Funderlic, and G. H. Golub, *Rank modifications of semidefinite matrices associated with a secant update formula*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 428–436.

[3] R. E. Cline and R. E. Funderlic, *The rank of a difference matrices and associated generalized inverses*, Linear Algebra Appl., 24 (1979), pp. 185–215.

[4] V. V. Prasolov, *Problems and Theorems in Linear Algebra*, AMS, Providence, RI, 1994.

[5] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[6] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1983.

[7] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed., Addison-Wesley, Reading, MA, 1995.

[8] E. W. Bai and S. S. Sastry, *Persistency of excitation, sufficient richness and parameter convergence in discrete time adaptive control*, Systems Control Lett., 6 (1985), pp. 153–163.

[9] J. E. Parkum, N. K. Poulsen, and J. Holst, *Recursive forgetting algorithms*, Internat. J. Control, 55 (1992), pp. 109–128.

[10] R. M. Johnstone, C. R. Johnson, Jr., R. R. Bitmead, and B. D. O. Anderson, *Exponential convergence of recursive least squares with exponential forgetting factor*, Systems Control Lett., 2 (1982), pp. 77–82.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1112–1125

APPROXIMATING THE LOGARITHM OF A MATRIX TO SPECIFIED ACCURACY*

SHEUNG HUN CHENG†, NICHOLAS J. HIGHAM‡, CHARLES S. KENNEY§, AND ALAN J. LAUB¶

Abstract. The standard inverse scaling and squaring algorithm for computing the matrix logarithm begins by transforming the matrix to Schur triangular form in order to facilitate subsequent matrix square root and Padé approximation computations. A transformation-free form of this method that exploits incomplete Denman–Beavers square root iterations and aims for a specified accuracy (ignoring roundoff) is presented. The error introduced by using approximate square roots is accounted for by a novel splitting lemma for logarithms of matrix products. The number of square root stages and the degree of the final Padé approximation are chosen to minimize the computational work. This new method is attractive for high-performance computation since it uses only the basic building blocks of matrix multiplication, LU factorization and matrix inversion.

Key words. matrix logarithm, Padé approximation, inverse scaling and squaring method, matrix square root, Denman–Beavers iteration

AMS subject classification. 65F30

PII. S0895479899364015

1. Introduction.

Logarithms of matrices arise in various contexts. For example, for a physical system governed by a linear differential equation of the form

$$\frac{dy}{dt} = Xy,$$

we may be interested in determining the matrix X from observations of the state vector $y(t)$ [1], [20]. If $y(0) = y_0$ then $y(t) = e^{Xt}y_0$, where the exponential of a matrix is defined by

$$e^X = \sum_{k=0}^{\infty} \frac{X^k}{k!}.$$

By observing y at $t = 1$ for initial states consisting of the columns of the identity matrix, we obtain the matrix $A = e^X$. Under certain conditions on A and X, we can then solve for X as $X = \log A$. This raises the question of how to compute a logarithm of a matrix.

When A is near the identity matrix several methods can be used to approximate $\log A$ directly, that is, without any nontrivial transformation of A. For example,

*Received by the editors November 16, 1999; accepted for publication (in revised form) by A. Edelman August 23, 2000; published electronically March 13, 2001.

http://www.siam.org/journals/simax/22-4/36401.html

†Centre for Novel Computing, Department of Computer Science, University of Manchester, Manchester, M13 9PL, England (scheng@cs.man.ac.uk, http://www.cs.man.ac.uk/~scheng/). The work of this author was supported by Engineering and Physical Sciences Research Council grant GR/L94314.

‡Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/). The work of this author was supported by Engineering and Physical Sciences Research Council grant GR/L94314.

§ECE Department, University of California, Santa Barbara, CA 93106-9560 (kenney@seidel.ece.ucsb.edu).

¶College of Engineering, University of California, Davis, CA 95616-5294 (laub@ucdavis.edu). The work of this author was supported by NSF grant ECS-9633326.

we can truncate the Taylor series $\log(I-W) = -W - W^2/2 - W^3/3 - \cdots$, where $W = I - A$. Alternatively, we can use Padé approximations of $\log(I-W)$; see [16] and section 5 below. Unfortunately, if A is not near the identity then these methods either do not converge or converge so slowly that they are not of practical use. The standard way of dealing with this problem is to use the square root operator repeatedly to bring A near the identity:

$$\log A = 2^k \log A^{1/2^k}. \tag{1.1}$$

(Definitions of the logarithm and square root functions for matrices are given in the next section.) As k increases, $A^{1/2^k} \to I$, so for sufficiently large k we can apply a direct method to $A^{1/2^k}$. This procedure for the logarithm was introduced by Kenney and Laub [15] and is referred to as inverse scaling and squaring, since it reverses the usual scaling and squaring method of evaluating the matrix exponential: $e^X = \left(e^{X/2^k}\right)^{2^k}$ [19], [21].

Two related questions arise with the inverse scaling and squaring method. First, potentially the most expensive part of the method is the computation of the square roots. For cases where only modest accuracy is required in the logarithm it is natural to ask whether the cost of this part of the computation can be reduced by computing approximate square roots. The second question concerns the effect of errors in computing the square roots on the accuracy of the computed logarithm. In [15] the square roots are computed using the Schur method [4], [9], [12], which has essentially optimal accuracy and stability properties, but the effects of rounding errors are not analyzed.

In partial answer to these questions we develop an extension of the inverse scaling and squaring method with two key properties.

1. It aims for a specified accuracy in the computed logarithm, requiring less work when more modest accuracy is requested. When full accuracy (that of the underlying arithmetic) is requested, our method becomes a new and attractive implementation of the original inverse scaling and squaring method.
2. It can be implemented using only the basic building blocks of matrix multiplication, LU factorization, and matrix inversion. The method is therefore attractive for high-performance computation.

In view of these two properties our method may also be of interest for computing the logarithm in variable precision computing environments, such as in symbolic manipulation packages. Our bounds for the various truncation errors are developed for exact arithmetic. In floating point arithmetic, rounding errors also influence the accuracy. We do not rigorously bound the effect of rounding errors, but rather estimate it in terms of the conditioning of the problem.

Our new method is based on a splitting lemma for the logarithm (Lemma 2.1 below), which says that if $A = BC$ and B and C commute then, under certain conditions,

$$\log A = \log B + \log C.$$

In the special case $B = C = A^{1/2}$ we recover the basis of (1.1): $\log A = 2\log A^{1/2}$. We apply the splitting lemma to the Denman–Beavers (DB) iteration for the matrix square root [5]:

$$\begin{aligned} Y_{k+1} &= \left(Y_k + Z_k^{-1}\right)/2, \qquad Y_0 = A, \\ Z_{k+1} &= \left(Z_k + Y_k^{-1}\right)/2, \qquad Z_0 = I. \end{aligned}$$

The DB iteration converges quadratically with $Y_k \to A^{1/2}$ and $Z_k \to A^{-1/2}$. The splitting lemma can be used to show that

$$\log A = 2\log Y_k - \log Y_k Z_k.$$

The matrix product $Y_k Z_k$ converges to the identity and so its logarithm converges to zero. Our approach is to iterate until $\log Y_k Z_k$ is sufficiently small, then apply the process recursively on Y_k, monitoring the error build-up as we proceed. We thus apply an incomplete square root cascade that brings A close enough to the identity so that the logarithm can be approximated directly.

To increase the efficiency of our method we develop in section 3 a product form of the DB iteration that trades one of the matrix inversions for a matrix multiplication and automatically generates the products $Y_k Z_k$. We also incorporate scaling to reduce the overall number of iterations. The product form iteration turns out to be closely related to the standard Newton iteration for the matrix sign function, as explained in section 4. In section 5 we develop the implementation details for the incomplete square root cascade. Our method uses a Padé approximation, explained in section 6, whose order is chosen in section 7 together with the number of square root stages in order to minimize the computational work. Numerical experiments are described in section 8 and conclusions are given in section 9.

2. Splitting lemma. We begin by defining the matrix logarithm and square root functions. Let A be a real or complex matrix of order n with no eigenvalues on $\mathbb{R}^-$ (the closed negative real axis). Then there exists a unique matrix X such that [15]

1. $e^X = A$;
2. the eigenvalues of X lie in the strip $\{\, z : -\pi < \mathrm{Im}(z) < \pi \,\}$;

We refer to X as the (principal) logarithm of A and write $X = \log A$. Similarly, there is a unique matrix S such that [9], [15]

1. $S^2 = A$;
2. the eigenvalues of S lie in the open halfplane: $0 < \mathrm{Re}(z)$.

We refer to S as the (principal) square root of A and write $S = A^{1/2}$.

If A is real then its principal logarithm and principal square root are also real.

For our first result, we need to define the open halfplane associated with $z = \rho e^{i\theta}$, which is the set of complex numbers $w = \zeta e^{i\phi}$ such that $-\pi/2 < \phi - \theta < \pi/2$.

LEMMA 2.1 (splitting lemma). *Suppose that $A = BC$ has no eigenvalues on $\mathbb{R}^-$ and*

1. *$BC = CB$;*
2. *every eigenvalue of B lies in the open halfplane of the corresponding eigenvalue of $A^{1/2}$ (or, equivalently, the same condition holds for C).*

Then $\log A = \log B + \log C$.

Proof. First we show that the logarithms of B and C are well defined. Since $A = BC = CB$ it follows that A commutes with B and C. Thus there is a correspondence between the eigenvalues a, b, and c of A, B, and C: $a = bc$. Express these eigenvalues in polar form as

$$a = \alpha e^{i\theta}, \quad b = \beta e^{i\phi}, \quad c = \gamma e^{i\psi}.$$

Since A has no eigenvalues on $\mathbb{R}^-$,

$$-\pi < \theta < \pi. \tag{2.1}$$

The eigenvalues of B lie in the open halfplanes of the corresponding eigenvalues of $A^{1/2}$, that is,

$$-\frac{\pi}{2} < \phi - \frac{\theta}{2} < \frac{\pi}{2}. \tag{2.2}$$

The relation $a = bc$ gives $\theta = \phi + \psi$, from which we have $\psi - \theta/2 = \theta/2 - \phi$. It follows from (2.2) that the eigenvalues of C lie in the open halfplanes of the corresponding eigenvalues of $A^{1/2}$. Thus, in view of (2.1), B and C have no eigenvalues on $\mathbb{R}^-$ and their logarithms are well defined.

Next, we show that $e^{\log B + \log C} = A$. The matrices $\log B$ and $\log C$ commute since B and C do. Using the well-known result that the exponential of the sum of commuting matrices is the product of the exponentials [14, Thm. 6.2.38], we have

$$e^{\log B + \log C} = e^{\log B} e^{\log C} = BC = A.$$

It remains to show that the eigenvalues of $\log B + \log C$ have imaginary parts in $(-\pi, \pi)$. This follows since, in view of the commutativity of B and C, the eigenvalues of $\log B + \log C$ are $\log b + \log c = \log a$. □

Note that for $A = BC$ the commutativity condition $BC = CB$ is not enough to guarantee that $\log A = \log B + \log C$, as the following scalar example shows. Let $a = e^{-2\epsilon i}$ and $b = c = e^{(\pi - \epsilon)i}$ for ϵ small and positive. Then $a = bc$ but

$$\log a = -2\epsilon i \neq (\pi - \epsilon)i + (\pi - \epsilon)i = \log b + \log c.$$

The reason for this behavior is that b and c are equal to a nonprincipal square root of a, and hence are not in the halfplane of $a^{1/2}$.

3. DB square root iteration. The DB iteration [5] for the square root of a matrix A with no eigenvalues on $\mathbb{R}^-$ is

$$\begin{aligned} Y_{k+1} &= \left(Y_k + Z_k^{-1}\right)/2, \qquad Y_0 = A, \\ Z_{k+1} &= \left(Z_k + Y_k^{-1}\right)/2, \qquad Z_0 = I. \end{aligned} \tag{3.1}$$

The iteration has the properties [8] (and see Theorem 4.1, below)

$$\lim_{k\to\infty} Y_k = A^{1/2}, \qquad \lim_{k\to\infty} Z_k = A^{-1/2}$$

and, for all k,

$$\begin{aligned} Y_k &= AZ_k, \\ Y_k Z_k &= Z_k Y_k, \\ Y_{k+1} &= (Y_k + AY_k^{-1})/2. \end{aligned} \tag{3.2}$$

The next lemma is the basis for our use of the DB iteration for computing the logarithm.

LEMMA 3.1. *The DB iterates satisfy the splitting relations*

$$\begin{aligned} \log A &= \log Y_k - \log Z_k \\ &= 2\log Y_k - \log Y_k Z_k \\ &= -2\log Z_k + \log Y_k Z_k. \end{aligned}$$

Proof. Since $A = Y_k Z_k^{-1}$, Y_k and Z_k commute and $\log Z_k^{-1} = -\log Z_k$, the first equality follows from Lemma 2.1 if we can show that the eigenvalues of Y_k are in the halfplane of the corresponding eigenvalues of $A^{1/2}$. By (3.2), the individual eigenvalues of Y_k follow the scalar iteration

$$y_{k+1} = (y_k + a y_k^{-1})/2, \qquad y_0 = a,$$

where a is an eigenvalue of A. This is just the scalar Newton iteration for the square root of a and it has the property that the iterates y_k remain in the halfplane of $a^{1/2}$ (see, e.g., [8]). Similar arguments show that $\log Y_k Z_k = \log Y_k + \log Z_k$, which yields the remaining two equalities. ∎

To see how to use Lemma 3.1, note that since $Y_k \to A^{1/2}$ and $Z_k \to A^{-1/2}$, $Y_k Z_k \to I$ and $\log Y_k Z_k \to 0$. Suppose we terminate the DB iteration after k iterations; we can write

$$\log A = 2 \log Y_k - E_1,$$

where we wish $E_1 = \log Y_k Z_k$ to be suitably small. Define $Y^{(1)} = Y_k$, $Z^{(1)} = Z_k$. We now apply the DB iteration to $Y^{(1)}$, again for a finite number of iterations. Continuing this process leads after s steps to

$$\log A = 2^s \log Y^{(s)} - E_1 - 2E_2 - \cdots - 2^{s-1} E_s, \quad E_i = \log Y^{(i)} Z^{(i)}, \tag{3.3}$$

where $Y^{(i)}$ and $Z^{(i)}$ are the final iterates from the DB iteration applied to $Y^{(i-1)}$. Our aim is that $\log Y^{(s)}$ be easy to compute and the E_i terms be small enough to be ignored. Note that we could apply the DB iteration to the $Z^{(i)}$ instead of the $Y^{(i)}$; all the following analysis is easily adapted for this choice.

We need to bound the error terms $E_i = \log Y^{(i)} Z^{(i)}$ without computing a matrix logarithm. One way to do this is as follows. Using the Taylor expansion of $\log(1+x)$ it is easy to show that if $\|I - YZ\| < 1$ then

$$\|\log YZ\| \le |\log(1 - \|I - YZ\|)|. \tag{3.4}$$

Here, and throughout, the norm is any subordinate matrix norm. The terms $Y_k Z_k$ are not formed during the DB iteration. However, a little manipulation shows that $Y_{k+1} Z_{k+1} - I = (Y_{k+1} - Y_k)(Z_{k+1} - Z_k)$ and hence

$$\|Y_{k+1} Z_{k+1} - I\| \le \|Y_{k+1} - Y_k\| \|Z_{k+1} - Z_k\|. \tag{3.5}$$

Thus, if this upper bound does not exceed 1, we have a bound for $\|E_i\|$ that can be computed at no extra cost and can be used to decide when the E_i terms can be neglected. However, both the bounds (3.4) and (3.5), and hence the bound for $\|E_i\|$, can be weak. Fortunately, there is a better approach: we can reformulate the DB iteration in terms of Y_k (or Z_k) and the required product $M_k = Y_k Z_k$, as the next lemma shows.

LEMMA 3.2 (product form of DB iteration). *Let Y_k and Z_k be the DB iterates for A and define $M_k = Y_k Z_k$. Then*

$$\begin{aligned} M_{k+1} &= \frac{1}{2}\left(I + \frac{M_k + M_k^{-1}}{2}\right), && M_0 = A, \\ Y_{k+1} &= Y_k (I + M_k^{-1})/2, && Y_0 = A, \\ Z_{k+1} &= Z_k (I + M_k^{-1})/2, && Z_0 = I. \end{aligned} \tag{3.6}$$

In a high-performance computing environment, iterating with M_k and Y_k from (3.6), at the cost of one inversion and one multiplication per iteration, is preferable to iterating with Y_k and Z_k from (3.1), at the cost of two inversions per iteration, since matrix multiplication is faster than matrix inversion.

In practice, it is vital to scale matrix iterations to produce reasonably fast overall convergence. Higham [11] derives a scaling for the DB iteration based on $\theta = \det(Y_k)\det(Z_k)$: it requires Y_k and Z_k to be multiplied by $|\theta^{-1/(2n)}|$ at the start of the $(k+1)$st iteration, where A is of order n. For the product form of the iteration, since $\det(Y_k)\det(Z_k) = \det(M_k)$ and we invert and hence factorize M_k, θ is available at no extra cost.

Matrix iterations such as the DB iteration can suffer from numerical instability. Although an iteration may be globally convergent for the specified starting matrices, rounding errors can introduce perturbations that grow unboundedly, this phenomenon usually being associated with loss of commutativity of the iterates. We define an iteration $X_{k+1} = f(X_k)$ to be stable in a neighborhood of a solution $X = f(X)$ if the error matrices $E_k = X - X_k$ satisfy

$$E_{k+1} = L(E_k) + O(\|E_k\|^2),$$

where L is a linear operator that has bounded powers, that is, there exists a constant c such that for all $p > 0$ and arbitrary E of unit norm, $\|L^p(E)\| \le c$. The DB iteration is stable [8], [11]; the iteration (3.2), which is a standard Newton iteration for $A^{1/2}$, is unstable unless the eigenvalues λ_i of A satisfy [8] $\max_{i,j}\left|1 - (\lambda_i/\lambda_j)^{1/2}\right| \le 2$.

It is easy to show that the product form of the DB iteration is stable. Define the error terms $G_k = Y_k - A^{1/2}$, $H_k = Z_k - A^{-1/2}$, and $J_k = M_k - I$. Simple manipulations show that, to first order in G_k, H_k, and J_k,

$$\begin{bmatrix} G_{k+1} \\ H_{k+1} \\ J_{k+1} \end{bmatrix} = \begin{bmatrix} I & 0 & -A^{1/2}/2 \\ 0 & I & -A^{-1/2}/2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} G_k \\ H_k \\ J_k \end{bmatrix} \equiv C \begin{bmatrix} G_k \\ H_k \\ J_k \end{bmatrix}.$$

The coefficient matrix C is idempotent ($C^2 = C$) and hence has bounded powers. Thus the iteration is stable.

Before explaining the use of the modified DB iteration, we develop more insight into its properties by relating it to a well-known iteration for the matrix sign function.

4. Relation to matrix sign function iteration. For a matrix N with no eigenvalues on the imaginary axis the sign function is defined by [10], [18]

$$\operatorname{sign} N = N\left(N^2\right)^{-1/2}.$$

The standard approach to compute $\operatorname{sign} N$ is to use the Newton iteration

$$N_{k+1} = \left(N_k + N_k^{-1}\right)/2, \qquad N_0 = N.$$

This iteration converges quadratically to $S = \operatorname{sign} N$, with error evolving in the Cayley metric according to [17]

$$(N_{k+1} - S)(N_{k+1} + S)^{-1} = \left((N_k - S)(N_k + S)^{-1}\right)^2.$$

The following theorem shows that the DB iterates are scaled versions of the Newton iterates for $\operatorname{sign} A^{1/2}$.

THEOREM 4.1. *Let A have no eigenvalues on $\mathbb{R}^-$. Let N_k be the Newton iterates for* $\operatorname{sign} A^{1/2}$ $(= I)$ *and Y_k, Z_k, and $M_k = Y_k Z_k$ be the DB iterates for A in* (3.1) *and* (3.6). *Then*

$$Y_k = A^{1/2} N_k, \qquad Z_k = A^{-1/2} N_k, \qquad M_k = N_k^2.$$

Proof. A straightforward induction, making use of the fact that N_k commutes with $A^{1/2}$. ∎

Theorem 4.1 implies that the DB iterates Y_k, Z_k, and M_k converge quadratically to $A^{1/2}$, $A^{-1/2}$, and I, respectively, with errors evolving in the Cayley metric according to

$$\begin{aligned}
(Y_{k+1} - A^{1/2})(Y_{k+1} + A^{1/2})^{-1} &= ((Y_k - A^{1/2})(Y_k + A^{1/2})^{-1})^2, \\
(Z_{k+1} - A^{-1/2})(Z_{k+1} + A^{-1/2})^{-1} &= ((Z_k - A^{-1/2})(Z_k + A^{-1/2})^{-1})^2, \\
(N_{k+1} - I)(N_{k+1} + I)^{-1} &= ((N_k - I)(N_k + I)^{-1})^2,
\end{aligned}$$

where $N_k = M_k^{1/2}$.

From [18] we know that k steps of the Newton iteration for the sign function generate the kth diagonal Padé approximation to the sign function, which is given by $r_k = p_k/q_k$, where p_k and q_k are the even and odd parts, respectively, of the polynomial $(1+x)^{2^k}$. Using Theorem 4.1 we can therefore obtain explicit rational expressions for the DB iterates. For example, $Y_k = \widetilde{p}_k(A)\widetilde{q}_k^{-1}(A)$, where

$$\begin{aligned}
\widetilde{p}_k(A) &= \binom{2^k}{0} + \binom{2^k}{2} A + \binom{2^k}{4} A^2 + \cdots + \binom{2^k}{2^k} A^{2^{k-1}}, \\
\widetilde{q}_k(A) &= \binom{2^k}{1} I + \binom{2^k}{3} A + \binom{2^k}{5} A^2 + \cdots + \binom{2^k}{2^k - 1} A^{2^{k-1}-1}.
\end{aligned}$$

5. Incomplete square root cascade. We return now to the use of the product form of the DB iteration to compute the logarithm. The following algorithm describes how we use the DB iteration, but omits convergence tests.

ALGORITHM 5.1. *This algorithm runs an incomplete square root cascade on the matrix A of order n, using the product form of the DB iteration with scaling. The DB iteration is invoked s times, with k_i iterations on the ith invocation.*

for $i = 1\colon s$
  if $i = 1$
    $M_0 = A$, $Y_0 = A$
  else
    $M_0 = Y^{(i-1)}$, $Y_0 = Y^{(i-1)}$
  end
  for $k = 0\colon k_i - 1$
    $\gamma_k = |(\det(M_k))^{-1/(2n)}|$
    $M_{k+1} = \frac{1}{2}\Big(I + \dfrac{\gamma_k^2 M_k + \gamma_k^{-2} M_k^{-1}}{2}\Big)$
    $Y_{k+1} = \frac{1}{2}\gamma_k Y_k (I + \gamma_k^{-2} M_k^{-1})$
  end
  $M^{(i)} = M_{k_i}$, $Y^{(i)} = Y_{k_i}$
end

With the notation of Algorithm 5.1, we can rewrite (3.3) as

$$\log A = 2^s \log Y^{(s)} - \log M^{(1)} - 2\log M^{(2)} - \cdots - 2^{s-1}\log M^{(s)}. \tag{5.1}$$

Rather than simply discard the terms $\log M^{(i)} = \log M_k$, we can approximate them using

$$\log M_k \approx M_k - I. \tag{5.2}$$

The error in this approximation satisfies

$$\|\log M_k - (M_k - I)\| \approx \|(M_k - I)^2\|/2. \tag{5.3}$$

For comparison, the error resulting from continuing for one more iteration and then discarding $\log M_{k+1}$ is

$$\|\log M_{k+1}\| \approx \|M_{k+1} - I\|. \tag{5.4}$$

It can be shown that

$$M_{k+1} - I = \frac{1}{4}(M_k - I)^2 M_k^{-1},$$

and hence the error term in (5.4) is approximately half that in (5.3) close to convergence (recall that $M_k \to I$). The product form of the DB iteration thus has an advantage over the original iteration; because it generates M_k explicitly it allows us to use the approximation (5.2) and thus to obtain similar accuracy in the logarithm with one less iteration.

Define the approximation $L^{(s)}$ to $\log A$ by

$$L^{(s)} = 2^s \log Y^{(s)} - \big(M^{(1)} - I\big) - 2\big(M^{(2)} - I\big) - \cdots - 2^{s-1}\big(M^{(s)} - I\big). \tag{5.5}$$

Then, subtracting (5.5) from (5.1) gives

$$\log A = L^{(s)} - \widetilde{E}_1 - 2\widetilde{E}_2 - \cdots - 2^{s-1}\widetilde{E}_s, \tag{5.6}$$

where

$$\widetilde{E}_i = \log M^{(i)} - \big(M^{(i)} - I\big).$$

THEOREM 5.2. *Let $\delta > 0$. In the ith product DB square root stage of Algorithm* 5.1 *let k_i be large enough so that*

$$\Big|\, \|W^{(i)}\| + \log(1 - \|W^{(i)}\|) \,\Big| \le \delta/4^{i-1}, \tag{5.7}$$

where $W^{(i)} = I - M^{(i)}$. Then

$$\|\log A - L^{(s)}\| \le 2\delta\left(1 - \frac{1}{2^s}\right). \tag{5.8}$$

Proof. Using the bound

$$\|\widetilde{E}_i\| = \|W^{(i)} + \log(I - W^{(i)})\| \le |\|W^{(i)}\| + \log(1 - \|W^{(i)}\|)| \le \delta/4^{i-1}$$

in (5.6) and summing a geometric series yields the result. ∎

To obtain a logarithm approximation, the final step is to approximate $\log Y^{(s)}$, by $\widetilde{L}$, say. Then our approximation to $\log A$ is

$$\widetilde{X} = 2^s \widetilde{L} - \sum_{k=1}^{s} 2^{k-1}\big(M^{(k)} - I\big). \tag{5.9}$$

Assuming we choose the k_i as in Theorem 5.2, then (5.8) leads to

$$\|\widetilde{X} - \log A\| \le 2^s \|\widetilde{L} - \log Y^{(s)}\| + 2\delta\left(1 - \frac{1}{2^s}\right). \tag{5.10}$$

It is natural to require that the error due to our approximation of $\log Y^{(s)}$ satisfy the same bound as the error introduced by the incomplete square roots; thus we require

$$\|\widetilde{L} - \log Y^{(s)}\| \le 2^{1-s}\delta\left(1 - \frac{1}{2^s}\right). \tag{5.11}$$

Then we have the overall error bound

$$\|\widetilde{X} - \log A\| \le 4\delta\left(1 - \frac{1}{2^s}\right) < 4\delta. \tag{5.12}$$

Two questions arise: How shall we select s and how can we find $\widetilde{L}$ such that (5.11) is satisfied? These questions are treated in the next two sections. We close this section by noting that (5.10) shows that the error in approximating $\log Y^{(s)}$ is magnified by a factor 2^s. This is a fundamental limitation of the inverse scaling and squaring approach that is also identified in [7].

6. Padé approximants. If A is near the identity matrix then rational approximation of $\log A$ is practical. Diagonal Padé approximants preserve some important properties of the logarithm and offer rapid convergence as the degree of the approximant increases [16]. For a given scalar function

$$f(x) = \sum_{n=0}^{\infty} a_n x^n,$$

we say that the rational function $r_{km} = p_{km}/q_{km}$ is a $[k/m]$ Padé approximant of f if p_{km} is a polynomial in x of degree at most k, q_{km} is a polynomial in x of degree at most m, and $f(x) - r_{km}(x) = O\big(x^{k+m+1}\big)$. In addition, we usually require that p_{km} and q_{km} are relatively prime (have no common zeros) and that q_{km} has been normalized so that $q_{km}(0) = 1$. These conditions ensure that if a $[k/m]$ approximant exists then it is unique; see [2] and [3]. Following Kenney and Laub [16] we restrict our attention to the diagonal ($k = m$) Padé approximants of $f(x) = \log(1 - x)$, the first three of which are (here, for convenience we have not normalized q_{mm})

$$r_{11}(x) = \frac{-2x}{2 - x}, \quad r_{22}(x) = \frac{-6x + 3x^2}{6 - 6x + x^2}, \quad r_{33}(x) = \frac{-60x + 60x^2 - 11x^3}{60 - 90x + 36x^2 - 3x^3}.$$

Kenney and Laub [16] show that the error in the Padé approximant evaluated at a matrix argument X is bounded by the error in the scalar approximation with $x = \|X\|$, provided that $\|X\| < 1$:

$$\|r_{mm}(X) - \log(I - X)\| \le |r_{mm}(\|X\|) - \log(1 - \|X\|)|. \tag{6.1}$$

This bound can be evaluated at negligible cost given r_{mm}.

7. Inverse scaling and squaring with specified accuracy. The availability of the error bound (6.1) for the Padé approximation makes possible a strategy for choosing s (the number of incomplete DB square root stages) and the order m of the final Padé approximation in order to achieve the desired accuracy with minimal work. For Padé approximation to be applicable s must be large enough so that $\|I - Y^{(s)}\| < 1$. Once this point is reached, we can compare the work required to produce an acceptable Padé approximation at the current square root stage with the work required to carry out another square root stage and then evaluate a Padé approximation.

In view of (5.11) and (6.1), a suitable order m_k of the Padé approximation at the kth square root stage is the smallest m for which

$$|r_{mm}(\|X\|) - \log(1 - \|X\|)| \le 2^{1-k}\delta\left(1 - \frac{1}{2^k}\right), \qquad X = I - Y^{(k)}, \tag{7.1}$$

where r_{mm} is the Padé approximant of order m as described in section 6. With this choice of m, and with the number of DB iterations k_i chosen as in Theorem 5.2, we have the bound (5.12), that is,

$$\|\widetilde{X} - \log A\| < 4\delta, \tag{7.2}$$

where $\widetilde{X}$ is given by (5.9) with $\widetilde{L} = r_{mm}(I - Y^{(k)})$. Note that this bound does not incorporate the effects of rounding errors. We comment below on the effects of roundoff.

Having determined m_k, we can consider whether to iterate further or not, by examining the cost of evaluating the Padé approximation. Several methods of evaluation are described and compared with respect to cost, storage, and accuracy in [13]. The best overall method is based on the partial fraction expansion

$$r_{mm}(x) = \sum_{j=1}^{m} \frac{\alpha_j^{(m)} x}{1 + \beta_j^{(m)} x}, \tag{7.3}$$

where the $\alpha_j^{(m)}$ are the weights and the $\beta_j^{(m)}$ the nodes of the m-point Gauss–Legendre quadrature rule on $[0, 1]$. Evaluating r_{mm} at the matrix argument X with $m = m_k$ requires the solution of m_k linear systems each having n right-hand sides, which we will regard as equivalent to m_k matrix inversions.

To estimate the cost of proceeding for a further square root stage we need to know the number of iterations in that stage and the degree m_{k+1} of the Padé approximation at the end of the stage. Since $A^{1/2^k} \to I$ as k increases, the square roots become easier to compute with increasing k, but this is compensated for by the more stringent accuracy demanded by the condition (5.7). In practice, the number of square root iterations frequently stays the same or decreases by 1 from one stage to the next. For our calculations we assume that the number of square root iterations on the $(k+1)$st stage is the same as that on the kth stage, which we denote by it_k. The estimated cost of the next square root stage is therefore it_k matrix multiplications and it_k matrix inversions.

To estimate m_{k+1} we note that

$$\left(I - A^{1/2^{k+1}}\right)\left(I + A^{1/2^{k+1}}\right) = I - A^{1/2^k}. \tag{7.4}$$

Since $A^{1/2^k} \to I$ we have

$$\|I - A^{1/2^{k+1}}\| \approx \frac{1}{2}\|I - A^{1/2^k}\|. \tag{7.5}$$

We therefore use the approximation $\|I-Y^{(k+1)}\| \approx \|I-Y^{(k)}\|/2$ in (7.1) to determine m_{k+1}.

Denoting by α the ratio "cost of matrix inversion divided by cost of matrix multiplication," we terminate the square root iterations if

$$m_k \le m_{k+1} + (1+\alpha)\mathrm{it}_k. \tag{7.6}$$

For our tests we have taken $\alpha = 1$ (as suggested by the operation counts), but on many computers a value of α bigger than 1 and possibly depending on n would be more appropriate.

It is worth stressing that if any of the assumptions underlying our choice of s and m are not satisfied then the efficiency of the computation may be less than optimal but the error bound (7.2) still holds.

We summarize our overall algorithm as follows.

ALGORITHM 7.1. *Given a matrix A with no eigenvalues on $\mathbb{R}^-$, and a tolerance $\delta > 0$, this algorithm approximates $X = \log A$ to within absolute accuracy 4δ (ignoring roundoff).*

1. Run Algorithm 5.1 with the k_i chosen as in Theorem 5.2, choosing s, the number of DB iteration stages, as the first k for which $\|I-Y^{(k)}\| \le 0.99$, and (7.6) is satisfied with $m_k \le 16$.
2. Use (7.3) to evaluate X, the $[m_s/m_s]$ Padé approximation $r_{m_s,m_s}(B)$ to $\log(I-B)$, where $B = I - Y^{(s)}$.
3. $X = 2^s X - \sum_{k=1}^{s} 2^{k-1}\big(M^{(k)} - I\big)$.

Now we return to the effects of roundoff. We do not attempt here a full rounding error analysis of Algorithm 7.1, as experience shows that it is difficult to obtain useful error bounds for iterations for the matrix square root and sign function. However, several observations can be made. First, it is shown in [13] that with the parameters 0.99 and 16 in step 1 of Algorithm 7.1 the Padé approximation is evaluated to high accuracy, because the matrices that are inverted are very well conditioned. Second, for tolerances δ sufficiently larger than u the rounding errors can be subsumed in the truncation errors. Finally, even if the computed $\widehat{X}$ has perfect backward stability, that is,

$$\widehat{X} = \log(A + \Delta A), \qquad \|\Delta A\| \le u\|A\|, \tag{7.7}$$

where u is the unit roundoff, then the best forward error bound is [6], [15]

$$\frac{\|\widehat{X} - X\|}{\|X\|} \le \|G'(A)\|\frac{\|A\|}{\|X\|}u + O(u^2),$$

where $G'(A)$ is the Fréchet derivative of $G(A) = \log A = X$ at A. The term

$$\mathrm{cond}_G(A) = \|G'(A)\|\frac{\|A\|}{\|X\|}$$

is a condition number for the logarithm function and it is notably absent in (7.2). Since no numerical algorithm can be expected to do better than achieve (7.7), we must accept that the computed $\widehat{X}$ will at best satisfy the modified version of (7.2)

$$\|\widehat{X} - \log A\|_1 \le \mathrm{cond}_G(A)\|X\|u + 4\delta. \tag{7.8}$$

Methods for estimating $\mathrm{cond}_G(A)$ are developed in [15].

8. Numerical experiments. We have implemented Algorithm 7.1 in MATLAB, for which the unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$. The various tests in Algorithm 7.1 use the 1-norm. We describe results for three matrices $A \in \mathbb{R}^{16\times 16}$.

Matrix 1: $\kappa_2(A) = 10^8$, $\mathrm{cond}_G(A) \approx 10^8$. A is a random symmetric positive definite matrix with eigenvalues exponentially distributed between 10^{-8} and 1, formed using MATLAB's `gallery('randsvd',...)`.

Matrix 2: $\kappa_2(A) = 1.2 \times 10^6$, $\mathrm{cond}_G(A) \approx 5 \times 10^9$. $A = QTQ^T$, where Q is a random orthogonal matrix and T, obtained using `gallery('rschur',...)`, is in real Schur form with eigenvalues $\alpha_j + i\beta_j$, $\alpha_j = -j^2/10$, $\beta_j = -j$, $j = 1\colon n/2$ and $(2j, 2j+1)$ elements μ (thus μ controls the nonnormality of the matrix). We took $\mu = 25$.

Matrix 3: $\kappa_2(A) = 13$, $\mathrm{cond}_G(A) \approx 1$. $A = QTQ^T$ is the same as matrix 2, but with $\mu = 0$.

For the tests we needed the exact logarithm, which we approximated by X_* computed using our own implementation of the inverse scaling and squaring method. Our code computes a Schur decomposition, computes square roots by the Schur method [4], [9], [12], and uses the [8/8] Padé approximation once $\|A^{1/2^k} - I\|_1 \le 0.25$. (Then the Padé approximation has error safely less than u [16, sect. 3].)

For each matrix we applied Algorithm 7.1 with tolerance $\delta = \epsilon\|X_*\|_F/4$, with ϵ ranging from 10^{-16} to 10^{-1}. The results are shown in Figure 8.1. In each plot ϵ is on the x-axis. The plots in the first row show the total number $\sum_{i=1}^{s} k_i$ of inner DB iterations (using the notation of Algorithm 5.1), and those in the second row show the total number of matrix multiplications for the complete logarithm computation (counting a matrix inversion as a multiplication). In the third row is plotted an approximation $\|\widehat{X} - X_*\|_F/\|X_*\|_F$ to the relative error.

The number of square roots computed by the inverse scaling and squaring method for Matrices 1–3 was 7, 20, and 5, respectively. The corresponding operation counts are about $32n^3$–$40n^3$ flops. The number of incomplete square root stages used by Algorithm 7.1 for Matrices 1–3 was in the ranges 5–7, 18–20, and 4–5, respectively. From the second row of Figure 8.1 we can see that the operation count for Algorithm 7.1 varies between about $20n^3$ and $150n^3$ flops, and only for very relaxed tolerances does Algorithm 7.1 better the flop count of the inverse scaling and squaring method. However, these operation counts do not reflect the fact that Algorithm 7.1 is built from high-level computational kernels that can be implemented very efficiently.

The results reported are for the Y form of the DB iteration, as specified in Algorithm 5.1. The corresponding Z form performs similarly.

We make the following comments on the results.

1. The number of inner iterations and the number of matrix multiplications both vary with the tolerance δ by factors up to 3.2, confirming that incomplete square root iterations with careful choice of the degree of Padé approximation can produce substantial savings in work.
2. Ideally, the relative error would be approximately equal to ϵ. For Matrix 1 this is the case down to $\epsilon = 10^{-8}$, at which point the relative error levels off due to ill-conditioning: the cond_G term in (7.8) starts to dominate. For Matrix 2 the relative errors are approximately constant at about 10^{-6}. Given that $\mathrm{cond}_G(A) \approx 5 \times 10^9$ this is the level of relative error we would expect for the smallest δ. Why the relative error increases only slightly with increasing δ is unclear, but may be related to the large number of (incomplete) square roots required and the consequent rapid decrease in the convergence tolerance. For

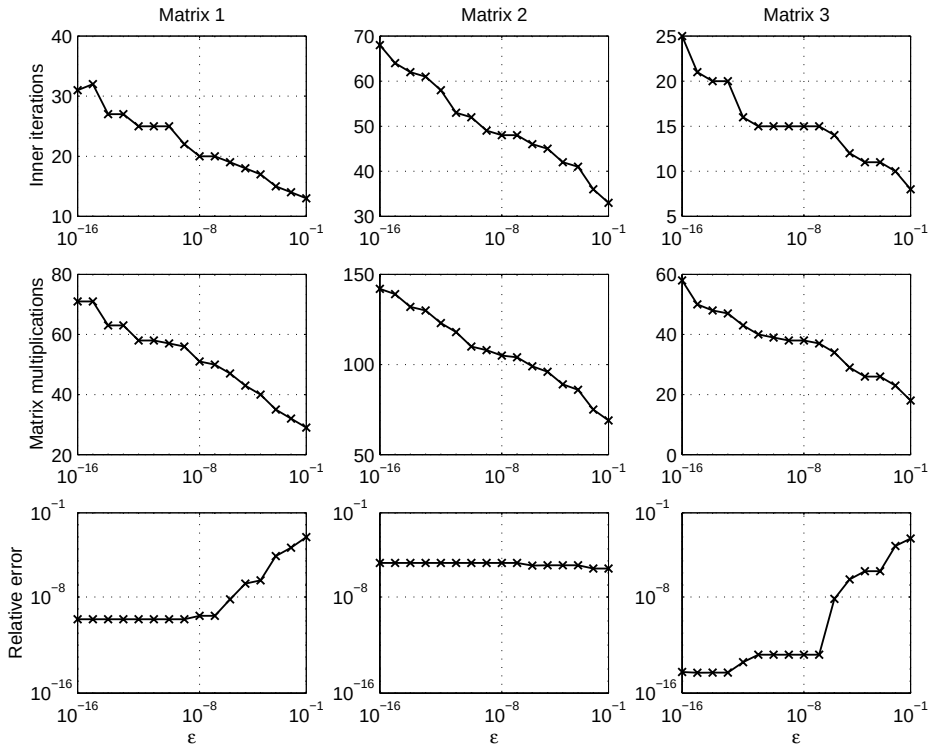


FIG. 8.1. *Results for Matrices* 1–3.

Matrix 3 the relative errors are somewhat less than ϵ, again for reasons that are not clear.

3. We also implemented the inverse scaling and squaring method using the DB iteration (3.1) with scaling, with the standard convergence test of the form $\|Y_{k+1} - Y_k\|/\|Y_{k+1}\| \le \theta$ and using the [8/8] Padé approximation. With $\theta = nu$ the number of inner iterations was 85, 506 (due to convergence problems, even with this relaxed tolerance), and 35 for Matrices 1–3, compared with 31, 68, and 25 for Algorithm 7.1 with $\delta = 10^{-16}\|X\|_F/4$; the accuracy of the computed logarithms was similar in both cases. The improved efficiency of Algorithm 7.1 is due to the better convergence test (based on $\|M_k - I\|$) and the use of the free approximation (5.2).

9. Conclusion. This work makes three main contributions. First, we have obtained a splitting result, Lemma 2.1, which gives conditions under which the logarithm of a matrix product is the sum of the logarithms. Second, we have derived a product form (3.6) of the DB iteration for the matrix square root; it trades a matrix inversion for a matrix multiplication and, unlike the original iteration, has a natural stopping test (based on $\|M_k - I\|$). We used the lemma and the iteration to derive a new version of the inverse scaling and squaring method for computing the matrix logarithm. The key features of our method are that it adapts itself to a specified accuracy (modulo the effects of roundoff) by carrying out incomplete square root computations and choosing a suitable Padé approximation, and that the computational kernels are matrix multiplication, LU factorization, and matrix inversion, making the method attractive for high-performance computation.

Acknowledgments. We thank the referees for their helpful comments.

REFERENCES

[1] R. C. Allen and S. A. Pruess, *An analysis of an inverse problem in ordinary differential equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 176–185.
[2] G. A. Baker, Jr., *Essentials of Padé Approximants*, Academic Press, New York, 1975.
[3] G. A. Baker, Jr. and P. Graves-Morris, *Padé Approximants*, Encyclopedia Math. Appl., 2nd ed., Cambridge University Press, Cambridge, England, 1996.
[4] A. Björck and S. Hammarling, *A Schur method for the square root of a matrix*, Linear Algebra Appl., 52/53 (1983), pp. 127–140.
[5] E. D. Denman and A. N. Beavers, Jr., *The matrix sign function and computations in systems*, Appl. Math. Comput., 2 (1976), pp. 63–94.
[6] L. Dieci, B. Morini, and A. Papini, *Computational techniques for real logarithms of matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 570–593.
[7] L. Dieci and A. Papini, *Conditioning and Padé approximation of the logarithm of a matrix*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 913–930.
[8] N. J. Higham, *Newton's method for the matrix square root*, Math. Comp., 46 (1986), pp. 537–549.
[9] N. J. Higham, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.
[10] N. J. Higham, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra Appl., 212/213 (1994), pp. 3–20.
[11] N. J. Higham, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.
[12] N. J. Higham, *A New* `sqrtm` *for MATLAB*, Numerical Analysis Report 336, Manchester Centre for Computational Mathematics, Manchester, England, January 1999.
[13] N. J. Higham, *Evaluating Padé approximants of the matrix logarithm*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1126–1135.
[14] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, England, 1991.
[15] C. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.
[16] C. Kenney and A. J. Laub, *Padé error estimates for the logarithm of a matrix*, Internat. J. Control, 50 (1989), pp. 707–730.
[17] C. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.
[18] C. S. Kenney and A. J. Laub, *The matrix sign function*, IEEE Trans. Automat. Control, 40 (1995), pp. 1330–1348.
[19] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.
[20] B. Singer and S. Spilerman, *The representation of social processes by Markov models*, Amer. J. Sociology, 82 (1976), pp. 1–54.
[21] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal., 14 (1977), pp. 600–610.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1126–1135

EVALUATING PADÉ APPROXIMANTS OF THE MATRIX LOGARITHM*

NICHOLAS J. HIGHAM†

Abstract. The inverse scaling and squaring method for evaluating the logarithm of a matrix takes repeated square roots to bring the matrix close to the identity, computes a Padé approximant, and then scales back. We analyze several methods for evaluating the Padé approximant, including Horner's method (used in some existing codes), suitably customized versions of the Paterson–Stockmeyer method and Van Loan's variant, and methods based on continued fraction and partial fraction expansions. The computational cost, storage, and numerical accuracy of the methods are compared. We find the partial fraction method to be the best method overall and illustrate the benefits it brings to a transformation-free form of the inverse scaling and squaring method recently proposed by Cheng, Higham, Kenney, and Laub [*SIAM J. Matrix Anal. Appl.*, 22 (2001), pp. 1112–1125]. We comment briefly on how the analysis carries over to the matrix exponential.

Key words. matrix logarithm, Padé approximation, inverse scaling and squaring method, Horner's method, Paterson–Stockmeyer method, continued fraction, partial fraction expansion

AMS subject classification. 65F30

PII. S0895479800368688

1. Introduction.

Any nonsingular matrix $A \in \mathbb{R}^{n\times n}$ having no eigenvalues on the negative real axis has a real logarithm, that is, a real matrix W such that $e^W = A$ [12, Thm. 6.4.15], [13]. Among all real logarithms there is a unique one whose eigenvalues have imaginary parts lying strictly between $-\pi$ and π; this is the principal logarithm, which we denote by $\log A$.

One of the most effective ways to compute $\log A$ is by inverse scaling and squaring combined with Padé approximation. The idea is to compute $Z = A^{1/2^k}$, with k large enough so that Z is close to the identity, and then to compute a Padé approximant of $\log Z$. The logarithm of A is then obtained from the identity [5], [13]

$$\log A = 2^k \log A^{1/2^k}. \tag{1.1}$$

We will refer to this method as the inverse scaling and squaring method. The method was proposed by Kenney and Laub [13], who suggested obtaining the square roots by computing a Schur decomposition of A and then taking square roots of the triangular Schur factor, using the methods of [2], [10]. Recently, Cheng, Higham, Kenney, and Laub [5] developed a transformation-free form of the inverse scaling and squaring method in which the square roots are approximated using a matrix iteration and certain parameters are chosen dynamically to minimize the computational cost subject to achieving a specified accuracy. This new version can be implemented using only matrix multiplication, LU factorization, and matrix inversion. The methods of [5] and [13] must evaluate a diagonal Padé approximant

$$r_m(x) = p_m(x)/q_m(x) = \log(1+x) + O(x^{2m+1})$$

*Received by the editors March 7, 2000; accepted for publication (in revised form) by D. Calvetti November 4, 2000; published electronically March 13, 2001.
http://www.siam.org/journals/simax/22-4/36868.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/). This work was supported by Engineering and Physical Sciences Research Council grant GR/L94314 and a Royal Society Leverhulme Trust Senior Research Fellowship.

at a matrix argument X with $\|X\| < 1$. Here, p_m and q_m are polynomials of degree m whose coefficients are known, and $m \leq 16$ in practice. The norm is any subordinate matrix norm. The question we consider here is how to evaluate the Padé approximant for a given m.

Evaluation of $r_m(X)$ by applying Horner's method to the numerator and denominator polynomials is the most obvious approach and was used in [13] and during the initial work of [5]. However, several alternatives are available and a hint that the use of a different representation of the rational r_m may be profitable is given by Dieci, Morini, and Papini [7], who comment that "for diagonal Padé approximants, it might instead be more desirable to pass to their quadrature formula equivalent ... to avoid ill-conditioning in the denominator of the rational function."

In the next section we describe the Paterson–Stockmeyer method for evaluating the p_m/q_m form and Van Loan's variant of it, together with methods based on continued fraction and partial fraction representations. We count the operations and storage required. The effect of rounding errors on the methods is described in section 3 and numerical experiments are given in section 4. We finish, in section 5, with a recommendation on the choice of method and a brief discussion of how the analysis carries over to the evaluation of Padé approximants of the matrix exponential.

2. Methods of evaluation. We consider methods of evaluating the Padé approximant $r_m(X)$ at $X \in \mathbb{R}^{n\times n}$ based on three representations. We note that in several of our equations matrices can be reordered, since rational functions of a matrix X commute, but such changes have no effect on the computational cost or accuracy. When counting storage we will include that for X and $r_m(X)$ and assume that X cannot be overwritten.

2.1. Rational evaluation. In this method the polynomials $p_m(X)$ and $q_m(X)$ are evaluated and $Y = r_m(X)$ is computed by solving $q_m Y = p_m$. We consider three possibilities. First, Horner's method can be used for the polynomial evaluations, as in [5], [13]. Thus

$$p_m(X) = \sum_{k=0}^{m} b_k X^k \tag{2.1}$$

is evaluated by

$S_m = b_m X + b_{m-1} I$
for $j = m-2\colon -1\colon 0$
  $S_j = X S_{j+1} + b_j$
end
$p_m = S_0$

and similarly for $q_m(X)$. The total cost is $2(m-1)M + I$, where we denote by M the cost of a matrix multiplication and I the cost of a matrix inversion or of solving a linear system with n right-hand sides.

Instead of using Horner's method we could explicitly compute the powers X^2, ..., X^m and evaluate p_m and q_m as linear combinations of the powers, at a cost of $(m-1)M + I$ (note that if the polynomial coefficients were matrices rather than scalars, this method would cost 50 percent more than Horner's method). However, a potentially greater reduction in cost over Horner's method is offered by a method of Paterson and Stockmeyer [9, sect. 11.2.4], [16]. It writes p_m as

$$p_m(X) = \sum_{k=0}^{r} B_k \cdot (X^s)^k, \qquad r = \mathrm{floor}(m/s), \tag{2.2}$$

where s is an integer parameter and

$$B_k = \begin{cases} b_{sk+s-1}X^{s-1} + \cdots + b_{sk+1}X + b_{sk}I, & k = 0\colon r-1, \\ b_m X^{m-sr} + \cdots + b_{sr+1}X + b_{sr}I, & k = r. \end{cases}$$

The powers $X^2, \ldots, X^s$ are computed, then the B_k, and finally (2.2) is evaluated by Horner's method. The cost of evaluating p_m is

$$(s + r - 1 - f(s,m))M, \qquad f(s,m) = \begin{cases} 1 & \text{if } s \text{ divides } m, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The cost of evaluating r_m by the Paterson–Stockmeyer method is $\big(s + 2r - 1 - 2f(s,m)\big)M + I$, which is approximately minimized[1] by $s = \sqrt{2m}$. We therefore take for s whichever of floor($\sqrt{2m}$) and ceil($\sqrt{2m}$) yields the smaller operation count. Unfortunately, the method requires $(s+2)n^2$ elements of storage. This can be reduced to $4n^2$ by computing p_m and q_m a column at a time, as shown by Van Loan [18], though the cost of evaluating r_m then increases to $(2s + 2r - 3 - 2f(s,m))M + I$. Since $s = \sqrt{m}$ approximately minimizes the cost of Van Loan's variant we take for s whichever of floor($\sqrt{m}$) or ceil($\sqrt{m}$) yields the smaller operation count.

2.2. Continued fraction. The Padé approximant r_m to $\log(1+x)$ has the continued fraction expansion [1, p. 174]

$$r_m(x) = \cfrac{c_1 x}{1 + \cfrac{c_2 x}{1 + \cfrac{c_3 x}{\cdots \cfrac{}{1 + \cfrac{c_{2m-1}x}{1 + c_{2m}x}}}}},$$

where

$$c_1 = 1, \qquad c_{2j} = \frac{j}{2(2j-1)}, \quad c_{2j+1} = \frac{j}{2(2j+1)}, \qquad j = 1, 2, \ldots.$$

This expansion can be evaluated at the matrix X in two ways. Top-down evaluation (which converts the continued fraction to rational form) is effected by the recurrence [3]

$A_1 = c_1 X$, $B_1 = I$, $A_2 = c_1 X$, $B_2 = I + c_2 X$
for $j = 3\colon 2m$
  $A_j = A_{j-1} + c_j X A_{j-2}$
  $B_j = B_{j-1} + c_j X B_{j-2}$
end
$r_m = A_{2m} B_{2m}^{-1}$.

The cost is $2(2m-2)M + I$.

Using bottom-up evaluation, $r_m(X)$ is evaluated by

$Y_{2m} = c_{2m} X$
for $j = 2m-1\colon -1\colon 1$
  Solve $(I + Y_{j+1})Y_j = c_j X$ for Y_j.
end
$r_m = Y_0$.

[1] In [7] $s = \sqrt{m}$ is chosen, which minimizes the cost of evaluating p_m or q_m alone, but not both together.

TABLE 1
Cost of evaluating $r_m(X)$. The optimal s are described in the text and f is defined in (2.3).

Method	Computational cost	Storage
Horner	$2(m-1)M + I$	$3n^2$
Paterson–Stockmeyer	$\big(s + 2r - 1 - 2f(s,m)\big)M + I \gtrsim (2\sqrt{2}\sqrt{m} - 1)M + I$	$(s+2)n^2$
Van Loan	$(2s + 2r - 3 - 2f(s,m))M + I \gtrsim (4\sqrt{m} - 3)M + I$	$4n^2$
Continued fraction	top-down: $2(2m-2)M + I$	$5n^2$
	bottom-up: $(2m-1)I$	$3n^2$
Partial fraction	mI	$3n^2$

This evaluation costs $(2m-1)I$.

Although the top-down evaluation is computationally expensive, it merits further consideration as it is well suited to situations in which the whole sequence $r_1(X)$, $r_2(X)$, ..., needs to be evaluated; in this case the bottom-up evaluation has to start afresh each time.

2.3. Partial fraction. The Padé approximant r_m can be expressed in partial fraction form as

$$r_m(x) = \sum_{j=1}^{m} \frac{\alpha_j^{(m)} x}{1 + \beta_j^{(m)} x}, \tag{2.4}$$

where the $\alpha_j^{(m)}$ are the weights and the $\beta_j^{(m)}$ the nodes of the m-point Gauss–Legendre quadrature rule on $[0,1]$ [7, Thm. 4.3]. The connection with quadrature stems from the integral representation

$$\log(1+x) = x \int_0^1 \frac{dt}{1+xt}.$$

Codes for computing the $\alpha_j^{(m)}$ and $\beta_j^{(m)}$ are given in [6, App. 2], [8], [17, sect. 4.5]; these computations are of negligible cost if $m \ll n$ and the coefficients can of course be precomputed and stored. The cost of evaluating (2.4) at the matrix X is mI. An advantage of (2.4) is its suitability for parallel evaluation; see [4] for a discussion and extensive bibliography on parallel evaluation of matrix partial fraction expansions.

Table 1 summarizes the cost of the methods. The Paterson–Stockmeyer and Van Loan methods clearly require the least computation for large m, since their costs grow as $\sqrt{m}$ for the optimal s rather than linearly with m as for the other methods. In fact, both methods are more efficient than Horner's method and the continued fraction methods for all m, as shown by Figure 1, in which the total number of matrix multiplications and inversions is plotted against m. For the range of m of interest the partial fraction method is competitive with the $O(\sqrt{m})$ methods.

The sensitivity of the methods to rounding errors is another important factor in the choice of method and we examine it in the next section.

3. Effects of rounding errors. Before beginning the error analysis we state some properties of $r_m = p_m/q_m$ that will be needed [14]. First, $q_m(x)$ is an increasing, positive function of x for $x > -1$. Second, the coefficients of p_m and q_m (with the normalization $q_m(0) = 1$) are nonnegative. To illustrate, in unnormalized form,

$$r_3(x) = \frac{60x + 60x^2 + 11x^3}{60 + 90x + 36x^2 + 3x^3}.$$

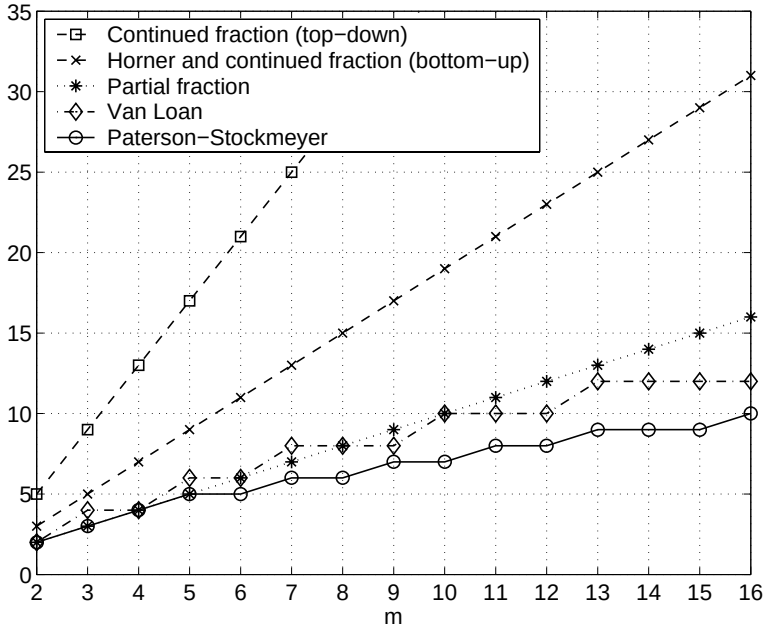


FIG. 1. *Total number of matrix multiplications and inversions to evaluate* $r_m(X)$.

It is straightforward to derive an error bound for Horner's method for evaluating a polynomial p_m of the form (2.1). The following result is a generalization of one for the scalar case [11, sect. 5.1]. We use the standard model of floating point arithmetic with unit roundoff u [11, sect. 2.2].

LEMMA 3.1. *The computed polynomial* $\widehat{p}_m$ *from Horner's method applied to* (2.1) *satisfies*

$$\|\widehat{p}_m - p_m\| \le m(n+1)u\,\widetilde{p}_m(\|X\|) + O(u^2),$$

where $\widetilde{p}_m(X) = \sum_{i=0}^m |b_k| X^k$.

The bound in the lemma is not the sharpest that can be obtained, but it is adequate for our application, in which $\|X\| < 1$.

In view of the lemma, the system that is solved to determine $Y = r_m(X)$ is

$$(q_m + \Delta Q)Y = p_m + \Delta P,$$
$$\|\Delta Q\| \le m(n+1)u q_m(\|X\|) + O(u^2), \quad \|\Delta P\| \le m(n+1)u p_m(\|X\|) + O(u^2),$$

where we have used the fact that our particular p_m and q_m have nonnegative coefficients. Assuming the system is solved by a stable method, the overall forward error bound will be of the form

$$\frac{\|Y - \widehat{Y}\|}{\|Y\|} \le d_1(m,n)u\kappa(q_m)\eta(X) + O(u^2), \tag{3.1}$$

where $d_j(m,n)$ denotes a constant depending on m and n and η is given by

$$\eta_1(X) = \left(\frac{p_m(\|X\|)}{\|q_m\|\|Y\|} + \frac{q_m(\|X\|)}{\|q_m\|}\right) \ge 1. \tag{3.2}$$

Kenney and Laub [14] show that

$$\kappa(q_m(X)) \le \frac{q_m(\|X\|)}{q_m(-\|X\|)}, \quad \|X\| < 1, \tag{3.3}$$

and this bound is easily evaluated for particular m and x.

For the Paterson–Stockmeyer and Van Loan methods it is not difficult to show that a bound of the same form as that in Lemma 3.1 holds, but with different constants. Therefore (3.1) applies to these methods too.

Next, we consider top-down evaluation of the continued fraction. We can express the recurrence for the B_j as

$$\begin{aligned}\begin{bmatrix} B_j \\ B_{j-1} \end{bmatrix} &= \begin{bmatrix} I & c_jX \\ I & 0 \end{bmatrix} \begin{bmatrix} B_{j-1} \\ B_{j-2} \end{bmatrix} \\ &= \begin{bmatrix} I & c_jX \\ I & 0 \end{bmatrix} \cdots \begin{bmatrix} I & c_2X \\ I & 0 \end{bmatrix} \begin{bmatrix} I \\ I \end{bmatrix}.\end{aligned}$$

From a standard error bound for matrix multiplication [11, Prob. 3.8] we have

$$\|\widehat{B}_{2m} - B_{2m}\| \le d_2(m,n)u \prod_{j=2}^{2m} (1 + c_j\|X\|).$$

Similarly,

$$\|\widehat{A}_{2m} - A_{2m}\| \le d_3(m,n)uc_1\|X\| \prod_{j=3}^{2m} (1 + c_j\|X\|).$$

Therefore (3.1) holds with η given by

$$\eta_2(X) = \frac{\prod_{j=3}^{2m}(1 + c_j\|X\|)}{\|q_m\|} \left(\frac{c_1\|X\|}{\|Y\|} + 1 + c_2\|X\| \right). \tag{3.4}$$

For the bottom-up evaluation of the continued fraction, in which Y_j is computed by solving $(I + Y_{j+1})Y_j = c_jX_j$, errors in Y_{j+1} can potentially be magnified by $\kappa(I + Y_{j+1})$ in passing to Y_j. Therefore it is essential that $\max_j \kappa(I+Y_j)$ is small. Assuming $\|Y_j\| < 1$, we have

$$\kappa(I + Y_j) \le \frac{1 + \|Y_j\|}{1 - \|Y_j\|}, \tag{3.5}$$

and the $\|Y_j\|$ satisfy, with $\|Y_{2m}\| = c_{2m}\|X\|$,

$$\|Y_j\| \le \frac{|c_j|\|X\|}{1 - \|Y_{j+1}\|}, \qquad j = 2m-1\colon -1\colon 1. \tag{3.6}$$

For a particular bound on $\|X\|$ we can therefore compute a bound on $\kappa(I + Y_j)$ and the overall error will be roughly bounded by $\max_j \kappa(I + Y_j)u$.

For the partial fraction method the accuracy is again dependent on the condition of the linear systems that are solved, and we expect the normwise relative error to be bounded approximately by $d_4(m,n)u\phi$, where

$$\phi = \max_j \left[\alpha_j^{(m)} \kappa(I + \beta_j^{(m)} X) \right] \tag{3.7}$$

TABLE 2

Terms from error analysis. $\epsilon(m,\|X\|)$ is defined in (3.8); η_1 in (3.2) and η_2 in (3.4) are terms from the Horner and top-down continued fraction methods; the bound for $\kappa(q_m(X))$ is from (3.3) and that for $\kappa(I+Y_j)$ from (3.5) and (3.6); ϕ for the partial fraction method is defined in (3.7).

			Approx. to		Bounds for		
$\|X\|$	m	$\epsilon(m,\|X\|)$	$\eta_1(X)$	$\eta_2(X)$	$\kappa(q_m(X))$	$\max_j \kappa(I+Y_j)$	ϕ
tol $= 2^{-24} \approx 6\times 10^{-8}$							
0.99	16	7.7e-3	6.8e2	1.9e3	4.5e10	8.3e0	1.8e0
0.95	16	1.9e-6	5.6e2	1.5e3	1.7e9	5.3e0	7.9e-1
0.90	14	3.5e-8	2.2e2	4.8e2	1.3e7	4.1e0	6.2e-1
0.75	8	4.7e-8	1.9e1	2.6e1	1.0e3	2.7e0	5.8e-1
0.50	5	2.3e-8	5.3e0	5.8e0	1.4e1	1.8e0	5.4e-1
0.25	3	5.7e-8	2.7e0	2.7e0	2.1e0	1.3e0	5.7e-1
0.10	3	5.1e-11	2.3e0	2.3e0	1.4e0	1.1e0	4.9e-1
tol $= 2^{-53} \approx 1\times 10^{-16}$							
0.90	16	2.9e-9	4.4e2	1.1e3	1.4e8	4.1e0	5.5e-1
0.75	16	2.6e-12	2.1e2	4.1e2	1.1e6	2.7e0	3.1e-1
0.50	16	3.4e-14	5.5e1	7.7e1	4.7e3	1.8e0	1.8e-1
0.25	7	0.0e0	4.2e0	4.4e0	5.9e0	1.3e0	2.7e-1
0.10	5	1.4e-17	2.5e0	2.5e0	1.7e0	1.1e0	3.1e-1
Largest $\|X\|$, m permitted in earlier version of [5].							
0.50	8	5.9e-13	1.0e1	1.2e1	6.9e1	1.8e0	3.5e-1

(note that $\alpha_j^{(m)} > 0$ and $\sum_j \alpha_j^{(m)} = 1$). We have

$$\kappa(I+\beta_j^{(m)}X) \le \frac{1+|\beta_j^{(m)}|\|X\|}{1-|\beta_j^{(m)}|\|X\|},$$

and since $\beta_j^{(m)} \in (0,1)$ the condition number is guaranteed to be small provided that $\|X\|$ is not too close to 1.

The two key parameters to consider when investigating the accuracy of the methods are the degree m of the Padé approximant and the norm of the matrix argument, X. In practice, these parameters are chosen so that $r_m(X)$ approximates $\log(I+X)$ to the desired accuracy, with either a fixed choice of m [7], [13] or a dynamic choice intended to minimize the overall computation time [5]. For a given X with $\|X\| < 1$ the bound

$$\|r_m(X) - \log(I+X)\| \le |r_m(-\|X\|) - \log(1-\|X\|)| =: \epsilon(m,\|X\|) \tag{3.8}$$

from [14] enables a suitable m to be determined.

In Table 2 we compare approximations to and bounds for the quantities arising in our analysis for a range of $\|X\|$ and m, with m chosen as the smaller of 16 and the minimal value for which $\epsilon(m,\|X\|) \le$ tol, where tol is a tolerance. The values of tol used for the table correspond to single and double precision accuracy in the Padé approximant, and for the η values we approximated $\|Y\| = \|\log(I+X)\| \approx \|X\|$ and $\|q_m(X)\| \approx q_m(0) = 1$.

The table implies that the effect of rounding errors on the bottom-up evaluation of the continued fraction and the partial fraction methods is negligible for all m and $\|X\|$ of interest. But Horner's method, the Paterson–Stockmeyer method, Van Loan's method, and the continued fraction evaluated top-down are all potentially unstable unless $\|X\|$ is much less than 1, as the denominator polynomial q_m has a condition number bound that grows rapidly with $\|X\|$ and the η terms from the error bounds

TABLE 3
Normwise relative errors. The pairs $(\|X\|, m)$ *correspond to those in Table* 2.

			Paterson–		Continued fraction		Partial
$\|X\|$	m	Horner	Stockmeyer	Van Loan	top-down	bottom-up	fraction
0.99	16	6.7e-12	3.5e-11	1.3e-11	2.9e-12	1.5e-16	4.2e-16
0.95	16	1.4e-15	3.0e-15	2.7e-15	1.2e-14	1.4e-16	3.1e-16
0.90	14	9.4e-14	5.9e-14	4.0e-14	7.9e-14	9.1e-17	2.0e-16
0.75	8	5.9e-16	1.0e-15	1.0e-15	1.6e-15	1.9e-16	3.7e-16
0.50	5	2.7e-16	2.3e-16	1.8e-16	3.9e-16	1.0e-16	5.7e-17
0.25	3	1.8e-16	7.9e-17	2.6e-16	1.7e-16	6.1e-17	4.1e-16
0.10	3	9.8e-17	1.0e-16	1.0e-16	9.8e-17	1.7e-16	3.2e-16
0.90	16	2.8e-13	2.8e-13	9.9e-14	2.1e-13	9.1e-17	2.6e-16
0.75	16	6.0e-15	1.3e-14	8.6e-15	1.1e-14	1.7e-16	3.4e-16
0.50	16	1.7e-15	1.2e-14	6.0e-15	1.6e-14	1.4e-16	4.1e-16
0.25	7	1.5e-16	1.8e-16	1.9e-16	4.3e-16	1.4e-16	4.5e-16
0.10	5	5.2e-17	1.4e-16	4.0e-17	2.8e-16	8.1e-17	8.6e-17

also become significant for $\|X\|$ close to 1. The last line of the table justifies a restriction on $\|X\|$ and m used in an earlier version of [5] in conjunction with Horner evaluation of r_m.

In the next section we check the actual errors via numerical experiments.

4. Numerical experiments. We report numerical experiments carried out in MATLAB, for which $u = 2^{-53} \approx 1 \times 10^{-16}$.

First we test the predictions from the analysis of the previous section. For random 4×4 matrices X with elements from the normal $N(0,1)$ distribution we computed the normwise relative errors $\|\widehat{Y} - Y\|_2 / \|Y\|_2$ in $Y = r_m(X)$ for a range of values of $\|X\|_2$ and m corresponding to Table 2. The "exact" logarithm was obtained using the variable precision arithmetic of MATLAB's Symbolic Math Toolbox. The results are shown in Table 3.

The results confirm that the Horner, Paterson–Stockmeyer, Van Loan, and top-down continued fraction methods do indeed suffer instability when $\|X\|$ is close to 1 and m is large, though the level of instability is much less than the bounds for $\kappa(q_m(X))$ in Table 2 would suggest. The actual $\kappa(q_m(X))$ values in this experiment are less than the square root of the bounds, showing that the bound (3.3) can be very weak. As expected, the bottom-up continued fraction and partial fraction methods give perfect accuracy.

Next we illustrate how the choice of method for evaluating the Padé approximant can affect the efficiency of Cheng, Higham, Kenney, and Laub's version of the inverse scaling and squaring method [5]. The implementation in [5] uses the partial fraction expansion with the restrictions that $\|X\| \le 0.99$ and $m \le 16$. An earlier implementation used Horner's method with the stronger restrictions that $\|X\| \le 1/2$ and $m \le 8$. In view of our analysis in the previous section and the value of ϕ in the first line of Table 2 these two implementations should have similar accuracy properties. We used both implementations to compute the logarithm of the 7×7 Frank matrix (MATLAB's `gallery('frank',7)`). The results are shown in Table 4 for two choices of tolerance in the method corresponding to approximation of the logarithm to single precision and double precision accuracy (all computations are carried out in double precision arithmetic). The partial fraction-based implementation is about 10 percent more efficient than the Horner-based implementation in this example. The improvement accrues from the algorithm being able to take fewer square roots and use a higher degree Padé approximant, as well as from the more efficient evaluation

TABLE 4
Comparison of current and earlier implementations of method from [5]. *"Roots" is the number of square roots, "Cost" the total number of matrix multiplications and matrix inversions, and m the degree of Padé approximant chosen.*

	$\text{tol} = 2^{-24}$			$\text{tol} = 2^{-53}$		
	Roots	Cost	Degree m	Roots	Cost	Degree m
Earlier (Horner)	9	63	5	9	93	8
Current (partial fraction)	8	58	8	9	86	8

of the Padé approximant.

5. Conclusions, and comments on the matrix exponential. We have analyzed alternatives to Horner's method for evaluating Padé approximants to the matrix logarithm. All but two of the alternatives are less expensive than Horner's method and the bottom-up continued fraction method and the partial fraction method have more favorable accuracy properties. Based on operation counts the choice narrows down to the Paterson–Stockmeyer method, Van Loan's version of it, and partial fraction expansion. For the degrees m of practical interest ($m \le 16$), the methods have similar computational cost, but the Paterson–Stockmeyer and partial fraction methods are rich in level 3 BLAS operations whereas Van Loan's method is inherently level 2 BLAS-based. If storage of size $(\sqrt{2m}+2)n^2$ is not available then the Paterson–Stockmeyer method must be ruled out. The partial fraction method has the advantage of being readily parallelizable and of allowing $\|X\|$ to be much closer to 1 without any loss of stability. Therefore the partial fraction expansion emerges as the best overall method.

In special cases a different choice may be appropriate. For example, if matrix multiplication is significantly faster than matrix inversion, as may be the case on certain high-performance machines, if sufficient storage is available, and if $\|X\|$ can be kept significantly less than 1, the Paterson–Stockmeyer method may be the most attractive choice.

An investigation similar to that given here can be done for the matrix exponential. Padé approximants $r_m = p_m/q_m$ of the matrix exponential e^A need to be evaluated in the scaling and squaring method, which approximates $e^{A/2^k}$ by $r_m(A/2^k)$ in the expression $e^A = (e^{A/2^k})^{2^k}$, where k is chosen so that $\|A/2^k\| \le 1$ [19] or $\|A/2^k\| \le 1/2$ [15], [9, sect. 11.3]. We briefly summarize some pertinent facts concerning the evaluation of $r_m(A/2^k)$. The coefficients $\alpha_j^{(m)}$ and $\beta_j^{(m)}$ in the partial fraction expansion (2.4) of r_m are not known explicitly, and the $\alpha_j^{(m)}$ can be very large [4], leading to numerical instability in the evaluation of the expansion. However, the techniques of [4] can be used to obtain an incomplete partial fraction expansion with suitably bounded coefficients. Ill conditioning of the denominator polynomial q_m is not an issue, as $\kappa(q_m(B)) < 5$ for $\|B\| \le 1$ [19, Thm. 1]. Finally, $q_m(X) = p_m(-X)$, and advantage can be taken of this when applying the Paterson–Stockmeyer and Van Loan methods. For the matrix exponential, then, the Paterson–Stockmeyer and Van Loan methods have the advantage over the partial fraction expansion except, possibly, in a parallel computing context.

Acknowledgments. Charlie Kenney suggested the possibility of using the continued fraction and partial fraction representations to evaluate r_m during our work on [5]. I thank Peter Graves-Morris for helpful comments on the manuscript.

REFERENCES

[1] G. A. Baker, Jr. and P. Graves-Morris, *Padé Approximants*, 2nd ed., Encyclopedia Math. Appl., Cambridge University Press, Cambridge, UK, 1996.

[2] A. Björck and S. Hammarling, *A Schur method for the square root of a matrix*, Linear Algebra Appl., 52/53 (1983), pp. 127–140.

[3] G. Blanch, *Numerical evaluation of continued fractions*, SIAM Rev., 6 (1964), pp. 383–421.

[4] D. Calvetti, E. Gallopoulos, and L. Reichel, *Incomplete partial fractions for parallel evaluation of rational matrix functions*, J. Comput. Appl. Math., 59 (1995), pp. 349–380.

[5] S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub, *Approximating the logarithm of a matrix to specified accuracy*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1112–1125.

[6] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed., Academic Press, Orlando, FL, 1984.

[7] L. Dieci, B. Morini, and A. Papini, *Computational techniques for real logarithms of matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 570–593.

[8] W. Gautschi, *Algorithm* 726: *ORTHPOL—A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules*, ACM Trans. Math. Software, 20 (1994), pp. 21–62.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[10] N. J. Higham, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, London, 1991.

[13] C. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.

[14] C. Kenney and A. J. Laub, *Padé error estimates for the logarithm of a matrix*, Internat. J. Control, 50 (1989), pp. 707–730.

[15] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.

[16] M. S. Paterson and L. J. Stockmeyer, *On the number of nonscalar multiplications necessary to evaluate polynomials*, SIAM J. Comput., 2 (1973), pp. 60–66.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, London, 1992.

[18] C. F. Van Loan, *A note on the evaluation of matrix polynomials*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 320–321.

[19] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal., 14 (1977), pp. 600–610.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1136–1152

JOINT APPROXIMATE DIAGONALIZATION OF POSITIVE DEFINITE HERMITIAN MATRICES*

DINH TUAN PHAM†

Abstract. This paper provides an iterative algorithm to jointly approximately diagonalize K Hermitian positive definite matrices $\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_K$. Specifically, it calculates the matrix $\mathbf{B}$ which minimizes the criterion $\sum_{k=1}^{K} n_k[\log\det \operatorname{diag}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*) - \log\det(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)]$, n_k being positive numbers, which is a measure of the deviation from diagonality of the matrices $\mathbf{B}\mathbf{C}_k\mathbf{B}^*$. The convergence of the algorithm is discussed and some numerical experiments are performed showing the good performance of the algorithm.

Key words. diagonalization, principal components, separation of sources

AMS subject classifications. 49M20, 65F30

PII. S089547980035689X

1. Introduction.

The need to diagonalize jointly approximately several positive definite matrices has arisen from (at least) two different problems: the common principal components estimation and the blind source separation. The first problem is statistical and has been introduced by Flury [5]. He considers k populations of multivariate observations of size $n_1, \ldots, n_K$, obeying the Gaussian distribution with zero means and covariance matrices $\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_K$, which are assumed to have common eigenvectors, that is, $\mathbf{\Gamma}_k = \mathbf{A}\mathbf{\Lambda}_k\mathbf{A}^*$, $k = 1, \ldots, K$, for some orthogonal matrix $\mathbf{A}$ and diagonal matrices $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_K$, the symbol * denoting the transpose. The goal is to estimate the $\mathbf{A}$ (the columns of which are the common principal components) from the sample covariance matrices $\mathbf{C}_k, \ldots, \mathbf{C}_K$ of the populations. As is well known, $n_k\mathbf{C}_k$ are distributed independently according to the Wishart distribution of n_k degrees of freedom and covariance matrices $\mathbf{\Gamma}_k$ (see, for example, Seber [9], noting that if the population means are unknown and have to be estimated, then n_k should be decreased by 1). Therefore the log likelihood function for $\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_K$ based on $\mathbf{C}_1, \ldots, \mathbf{C}_K$ equals

$$C - \frac{1}{2}\sum_{k=1}^{K} n_k[\log\det\mathbf{\Gamma}_k + \operatorname{tr}(\mathbf{\Gamma}_k^{-1}\mathbf{C}_k)],$$

where C is a constant and tr denotes the trace. Thus the maximum likelihood method for estimating $\mathbf{A}$ and the $\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_K$ amounts to minimizing

$$\frac{1}{2}\sum_{k=1}^{K} n_k[\log\det\mathbf{\Lambda}_k + \operatorname{tr}(\mathbf{B}^*\mathbf{\Lambda}_k^{-1}\mathbf{B}\mathbf{C}_k) - \log\det(\mathbf{B}\mathbf{B}^*)],$$

where we have put $\mathbf{B} = \mathbf{A}^{-1}$ for compatibility with other notations introduced later. Note that the term $\operatorname{tr}(\mathbf{B}^*\mathbf{\Lambda}_k^{-1}\mathbf{B}\mathbf{C}_k)$ in the above expression can be written as $\operatorname{tr}(\mathbf{\Lambda}_k^{-1}\mathbf{B}\mathbf{C}_k\mathbf{B}^*)$. Then it is not hard to see that for fixed $\mathbf{B}$, the above expression

*Received by the editors May 2, 2000; accepted for publication (in revised form) by A. Sayed August 25, 2000; published electronically March 13, 2001.
http://www.siam.org/journals/simax/22-4/35689.html
†Laboratory LMC/IMAG, C.N.R.S., University of Grenoble, B.P. 53X, 38041 Grenoble cedex, France (Dinh-Tuan.Pham@imag.fr).

is minimized when $\mathbf{\Lambda}_k = \operatorname{diag}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)$, where $\operatorname{diag}(\cdot)$ denotes the diagonal matrix with the same diagonal as its argument. Thus substituting $\mathbf{\Lambda}_k$ by this value, the above expression becomes

$$\frac{1}{2}\sum_{k=1}^{K} n_k[\log\det\operatorname{diag}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*) + K - \log\det(\mathbf{B}\mathbf{B}^*)],$$

which should now be minimized with respect to $\mathbf{B}$. Note that in Flury [5] the matrix $\mathbf{B}$ is assumed to be orthogonal and hence the term $\log\det(\mathbf{B}\mathbf{B}^*)$ disappears. However, we will not assume orthogonality and hence we have this term. Further, since the matrix $\mathbf{C}_k$ does not depend on the parameter $\mathbf{B}$, we may add $\log\det\mathbf{C}_k$ to this term, which becomes $\log\det(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)$. This leads to the following cost function, dropping the constant K:

$$\frac{1}{2}\sum_{k=1}^{K} n_k[\log\det\operatorname{diag}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*) - \log\det(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)]. \tag{1.1}$$

This function is precisely a measure of the global deviation of the matrices $\mathbf{B}^*\mathbf{C}_k\mathbf{B}$ from diagonality, since, from the Hadamard inequality (see, for example, Cover and Thomas [4, p. 233 or 502]), $\det\operatorname{diag}(\mathbf{M}) \geq \det\mathbf{M}$ with equality if and only if $\mathbf{M}$ is diagonal. Thus minimizing (1.1) can be viewed as trying to find a matrix $\mathbf{B}$ which diagonalizes jointly the matrices $\mathbf{C}_1, \ldots, \mathbf{C}_K$ as much as it can.

The blind source separation problem comes from the field of signal processing and has received much attention recently because of its many potential applications. In this problem, K linear mixtures of K sources have been recorded and the goal is to extract the sources from the observations, *without relying on any specific knowledge about the sources* other than that they are statistically independent (this is why the separation is called blind). Let $\mathbf{X}(t)$ and $\mathbf{S}(t)$ denote the vectors of measurements and of sources at time t, the mixture model can be written as $\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t)$ for some square matrix $\mathbf{A}$. Since one can only rely on the independence of the sources for their extraction, a natural idea is to find a matrix $\mathbf{B}$ such that the components of $\mathbf{B}\mathbf{X}(t)$ (which represent the reconstructed sources) are as independent as possible. As it is easier to work with noncorrelation rather than independence, a simple method would be to try to make the cross-covariances, including lagged cross-covariances, between the sources, vanish. This would lead to the joint approximate diagonalization of a certain set of covariance matrices, as proposed in Belouchrami et al. [1]. On the other hand, Cardoso and Souloumiac [3] do not consider lagged covariances but higher order cumulants between the sources instead. They construct a certain set of matrices in which such cumulants appear as off-diagonal elements and then separate the sources through a joint approximate diagonalization of these matrices.

It should be pointed out that the above authors use a different measure of deviation to diagonality than that of Flury. Their measure is simply the sum of squares of the off-diagonal elements of the considered matrices. But there is a common feature in all the above works in that the diagonalizing matrix $\mathbf{B}$ is taken to be orthogonal. In this work we shall *drop this restriction*. The orthogonality condition is part of the assumption of Flury [5] but there is no clear statistical reason why it should be satisfied. In principal components analysis, since the components are taken to be the eigenvectors of a symmetric matrix, their orthogonality is automatically satisfied. Thus this property is a mathematical property which happens to hold, but in our opinion is not a statistical requirement. If the components are to be interpreted as underlying

factors, in factor analysis for example, then one may need to drop the orthogonality constraint. In fact, in practice factor rotations (actually nonorthogonal rotation!) are performed quite often, based on a sparseness or parsimony criterion, to obtain interpretable factors. When there are many covariance matrices involved, we don't see an appealing reason to insist that the components be the common eigenvector of these matrices, while the requirement that these matrices be simultaneously diagonalized is of interest because it implies that the corresponding factors are statistically uncorrelated. In the blind source separation problem, the orthogonality constraint is introduced in the works of Cardoso and Souloumiac [3] and Belouchrami et al. [1] because these authors have prewhitened their observations so that they are uncorrelated and have unit variance. We want to avoid this prewhitening stage, which can adversely affect its performance of the method since the statistical error committed in this stage cannot be corrected in the following "effective separation" stage (see Cardoso [2]). In fact the method of these authors amounts to requiring a certain covariance matrix be exactly diagonalized while other (covariance or cumulant) matrices can only be approximately diagonalized. By dropping the orthogonality restriction, we obtain a single-stage separation procedure which is simpler and can perform better, since the matrices can be treated in equal footing. Note that without the orthogonality restriction, exact joint diagonalization is possible for two positive definite matrices (see, for example, Golub and van Loan [7, Algorithm 8.7.1]). This "double" diagonalization has in fact been exploited in Pham and Garat [8] for blind source separation. But for more than two matrices joint diagonalization can only be achieved approximately relative to some measure of deviation to diagonality. We take this measure to be (1.1) for two following reasons. First, it can be traced back to the likelihood criterion, widely used in statistics. Second, it is invariant with respect to scale change: it remains the same if the matrices to be diagonalized are pre- and postmultiplied by a same diagonal matrix. The other measure, adopted by Cardoso and Souloumiac [3] and Belouchrami et al. [1] does not have this nice invariant property. Of course, one can introduce this property by first normalizing the matrices so that they have unit diagonal elements, but then the resulting criterion would be very hard to manipulate. Flury [5] uses the same criterion as ours, but with the orthogonality constraint.

After completing this work, we have been aware of the recent work of Yeredor [10, 11], in which the author introduces a joint approximate diagonalization algorithm without the orthogonality constraint. But the author uses the same measure of deviation to diagonality as Cardoso and Souloumiac [3] and Belouchrami et al. [1], which differs from ours and hence his algorithm is completely different. Yeredor [11] has also introduced a set of weights in his criterion which takes into account the statistical property of the covariance matrices $\mathbf{C}_k$ and could make his criterion closer to the likelihood criterion. Our criterion, being likelihood-based, takes care of this in an automatic way.

The main result of this paper is the derivation of an algorithm for the joint approximate diagonalization in the sense of the criterion (1.1) and *without the restriction that the diagonalizing matrix be orthogonal*. Our algorithm has some similarity with that of Cardoso and Souloumiac [3] and even more with that of Flury and Gautschi [6]: it follows the classic Jacobi approach of making successive transformations on pairs of rows and columns of the matrices to be diagonalized. However, our elementary transformation are not (and cannot be) the same as that of these authors. Further, the convergence proof is completely different since we can no longer rely on the orthogonality property. Incidentally, our method of proof can be adapted to prove

the convergence result in Flury and Gautschi [6] in a much simpler way. For ease of reading, proofs of results are relegated to the appendix.

2. The algorithm. As complex data frequently arise in signal processing applications, we shall consider complex Hermitian (instead of real symmetric) positive definite matrices $\mathbf{C}_1, \ldots, \mathbf{C}_K$. (Note that Cardoso and Soulomiac [3], Bellouchrani et al. [1], and Yeredor [10] also work in a complex setting.) The problem is to find a complex matrix $\mathbf{B}$ such that the matrices $\mathbf{B}\mathbf{C}_1\mathbf{B}^*, \ldots, \mathbf{B}\mathbf{C}_K\mathbf{B}^*$ are as close to diagonal as possible, the notation * now denoting the transpose complex conjugate. The measure of deviation to diagonality is taken to be (1.1), where the n_k are positive weights (they need not be integers). Note that since the $\mathbf{C}_k$ do not depend of $\mathbf{B}$, minimizing (1.1) is the same as minimizing $\frac{1}{2}\sum_{k=1}^K n_k[\log\det\operatorname{diag}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*) - \log|\det\mathbf{B}|]$.

The algorithm consists of performing successive transformations, each time on a pair of rows of $\mathbf{B}$, the ith row $\mathbf{B}_{i\cdot}$ and the jth row $\mathbf{B}_{j\cdot}$, say, according to

$$\begin{bmatrix}\mathbf{B}_{i\cdot}\\ \mathbf{B}_{j\cdot}\end{bmatrix} \leftarrow \mathbf{T}_{ij}\begin{bmatrix}\mathbf{B}_{i\cdot}\\ \mathbf{B}_{j\cdot}\end{bmatrix}, \tag{2.1}$$

where $\mathbf{T}_{ij}$ is a 2×2 nonsingular matrix, chosen such that the criterion is sufficiently decreased. Whether a decrease is sufficient is a question which we shall return to in the next section. Once this is done, the procedure is repeated with another pair of rows. The processing of all the $K(K-1)/2$ is called a sweep. The algorithm consists of repeated sweeps until convergence is achieved.

The decrease of the criterion (1.1) induced by the transformation (2.1) is

$$\frac{1}{2}\sum_{k=1}^K n_k\Biggl\{2\log|\det\mathbf{T}_{ij}| - \log\det\operatorname{diag}\left(\mathbf{T}_{ij}\begin{bmatrix}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)ij\\ (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ji} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}\end{bmatrix}\mathbf{T}_{ij}^*\right) - \log[(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}\,(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}]\Biggr\},$$

where $(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij}$ denotes the element (i,j) of the matrix $\mathbf{B}\mathbf{C}_k\mathbf{B}^*$. A natural idea is to chose $\mathbf{T}_{ij}$ to maximize this decrease. However, it does not seem possible to derive explicit formulae for this maximization. Our idea is to maximize a lower bound of it instead. Since the logarithm function is convex, by the Jensen inequality (see, for example, Cover and Thomas [4, Theorem 2.6.2]) for any two sets of positive numbers $p_1, \ldots, p_K$ and $x_1, \ldots, x_K$ with $\sum_{k=1}^K p_k = 1$, one has $\sum_{k=1}^K p_k\log x_k \le \log(\sum_{k=1}^K p_k x_k)$. Applying this inequality twice with $p_k = n_k/\sum_{k=1}^K n_k$ and with x_k being the first and second diagonal elements of

$$\begin{bmatrix}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & 0\\ 0 & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}\end{bmatrix}^{-1}\left\{\mathbf{T}_{ij}\begin{bmatrix}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)ij\\ (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ji} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}\end{bmatrix}\mathbf{T}_{ij}\right\},$$

respectively, the above decrease can be seen to be bounded below by

$$(n/2)[2\log|\det\mathbf{T}_{ij}| - \log(\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*)_{11} - \log(\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*)_{22}], \tag{2.2}$$

where $n = \sum_{k=1}^K n_k$,

$$\mathbf{P} = \frac{1}{n}\sum_{k=1}^K \frac{n_k}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}}\begin{bmatrix}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij}\\ (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ji} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}\end{bmatrix}, \tag{2.3a}$$

$$\mathbf{Q} = \frac{1}{n}\sum_{k=1}^K \frac{n_k}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}}\begin{bmatrix}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij}\\ (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ji} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}\end{bmatrix}, \tag{2.3b}$$

and $(\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*)_{11}$ and $(\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*)_{22}$ denote the first and second diagonal elements of $\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*$ and $\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*$.

Since (2.2) clearly vanishes when $\mathbf{T}_{ij}$ is the identity matrix, its maximum (with respect to $\mathbf{T}_{ij}$) is nonnegative and can be zero only if the maximum is attained at the identity matrix. Thus the transformation (2.1) with $\mathbf{T}_{ij}$ being the matrix realizing the maximum of (2.2) will decreases the criterion (1.1) unless (2.2) attains it maximum at the identity matrix. The key point is that the maximization of (2.2) can be done analytically, using the following result.

PROPOSITION 2.1. *A necessary and sufficient condition that the nonsingular matrix* $\mathbf{T}_{ij}$ *maximizes* (2.2) *is that the matrices* $\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*$ *and* $\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*$ *are diagonal with diagonal elements* p_1', p_2' *and* q_1', q_2' *satisfying* $p_2'q_1' \geq p_1'q_2'$.

The above result shows that the only case where the criterion (1.1) cannot be decreased by the above technique is when both matrices $\mathbf{P}$ and $\mathbf{Q}$ are diagonal. If this holds for a pair (i, j), one just skips this pair and process other pairs. If this holds for all pairs, then the algorithm stops. Referring to the definition (2.3) of $\mathbf{P}$ and $\mathbf{Q}$, the last case can arise only when

$$g_{ij} \overset{\text{def}}{=} \sum_{k=1}^{K} \frac{n_k}{n} \frac{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij}}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}} = 0, \qquad 1 \leq i \neq j \leq K. \tag{2.4}$$

But it can be seen that the above system of equations merely expresses that $\mathbf{B}$ is a stationary point of the criterion (1.1). Indeed, consider a small change in $\mathbf{B}$ of the form $\boldsymbol{\delta}\mathbf{B}$, matrix $\boldsymbol{\delta}$ representing a *relative* change, then the corresponding change of the criterion (1.1) is

$$\sum_{k=1}^{K} \frac{n_k}{2} \log\det\{\operatorname{diag}^{-1}(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)\operatorname{diag}[\mathbf{B}\mathbf{C}_k\mathbf{B}^* + 2\Re(\mathbf{B}\mathbf{C}_k\mathbf{B}^*\boldsymbol{\delta}^*) + \boldsymbol{\delta}\mathbf{B}\mathbf{C}_k\mathbf{B}^*\boldsymbol{\delta}^*]\} - n\log|\det(\mathbf{I}+\boldsymbol{\delta})|, \tag{2.5}$$

where $\operatorname{diag}^{-1}(\cdot)$ denotes the inverse of $\operatorname{diag}(\cdot)$ and $\Re$ denotes the real part. Expanding (2.5) with respect to $\boldsymbol{\delta}$ up to the first order, one gets

$$\sum_{k=1}^{K} n_k \sum_i \Re\left[\frac{\sum_j (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij}\bar{\delta}_{ij}}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}} - \delta_{ii}\right] = n\Re\sum_{i\neq j}\sum g_{ij}\bar{\delta}_{ij},$$

where δ_{ij} denotes the general element of $\boldsymbol{\delta}$ and $\bar{\delta}_{ij}$ its complex conjugate and g_{ij} is given by (2.4). This shows that ng_{ij} are the components of the (relative) gradient of the criterion and our algorithm only stops when this vector vanishes.

Note. The Flury and Gautschi [6] algorithm operates on a similar principle. However, these authors iterate the transformation (2.1) with a fixed pair $(i,\ j)$ until convergence and only then they change to another pair. We feel that this is less efficient, because by using the same pair, the decrease of the criterion tends to be smaller each time while by changing it one can get a large decrease in the first few iterations. Our algorithm is also simpler to program.

2.1. Explicit formula for the transformation matrix. The application of Proposition (2.1) requires the joint diagonalization of two matrices for which the solution is known and, in the case of 2×2 matrices, can be written down explicitly.

PROPOSITION 2.2. *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *be two nonproportional Hermitian matrices of order two, with diagonal and upper off-diagonal elements* p_1, p_2, p *and* q_1, q_2, q,

respectively. Then

$$\alpha = p_2\bar{q} - \bar{p}q_2, \quad \beta = p_1q_2 - p_2q_1 + \bar{p}q - p\bar{q}, \quad \gamma = pq_1 - p_1q$$

are not all zero and $\Delta = \beta^2 - 4\alpha\gamma$ *is real and* $\mathbf{P}$ *and* $\mathbf{Q}$ *are jointly diagonalized by the matrix* $\mathbf{T}$ *if and only if one of the following conditions holds:*

(i) $\mathbf{T}$ *has a zero row,*

(ii) *the rows of* $\mathbf{T}$ *are nonzero and are proportional to* $[2\alpha \;\; \beta+\delta]$ *or* $[\beta-\delta \;\; 2\gamma]$ *and to* $[2\alpha \;\; \beta-\bar{\delta}]$ *or* $[\beta+\bar{\delta} \;\; 2\gamma]$*, where* δ *is any one of the two square roots of* Δ*.*

Note that for each choice of δ, the matrix $\mathbf{T}$ under the condition (ii) above has its rows uniquely defined up to a constant factor, since $[2\alpha \;\; \beta+\delta]$ and $[\beta-\delta \;\; 2\gamma]$ and $[2\alpha \;\; \beta-\bar{\delta}]$ and $[\beta+\bar{\delta} \;\; 2\gamma]$ are proportional if they are both nonzero (by the equality $\beta^2 - \delta^2 = 4\alpha\gamma$). We provide two vectors for symmetry reasons and because one of them *might be zero*, in this case the row of $\mathbf{T}$ should be proportional to the other.

The above result has been given in all its generality which allows $\Delta \leq 0$. In this case $\delta = -\bar{\delta}$ and hence the matrix $\mathbf{T}$ under the condition (ii) has proportional rows; therefore $\mathbf{P}$ and $\mathbf{Q}$ can only be diagonalized by a singular matrix which transforms them to a matrix with at most a nonzero term on the diagonal. This case, however, is excluded in the present application, by the following result.

LEMMA 2.3. *The quantity* Δ *in Proposition* 2.1 *is positive if* $\mathbf{P}$ *and* $\mathbf{Q}$ *are not proportional and* $\det \mathbf{P} > 0$ *or* $\det \mathbf{Q} > 0$*.*

If $\Delta > 0$, it has a positive root, denoted as usual by $\sqrt{\Delta}$. Then taking δ in Proposition 2.2 to be $\operatorname{sign}(p_2q_1 - q_1p_2)\sqrt{\Delta}$, $\operatorname{sign}(\cdot)$ denoting the sign function,[1] the vectors $[\beta-\delta \;\; 2\gamma]$ and $[2\alpha \;\; \beta-\delta]$ will be nonzero and hence the matrix $\mathbf{T}$ must have rows proportional to them (the trivial case where $\mathbf{T}$ has a zero row being excluded). Similarly, if one takes δ to be $-\operatorname{sign}(p_2q_1 - q_1p_2)\sqrt{\Delta}$, then $\mathbf{T}$ must have rows proportional to $[2\alpha \;\; \beta+\delta]$ and $[\beta+\delta \;\; 2\gamma]$, but δ now has opposite sign of that of the previous choice. Thus in all cases $\mathbf{T}$ must equal the product of a diagonal and a permutation matrix with

$$\begin{bmatrix} \beta - \operatorname{sign}(p_2q_1 - p_1q_2)\sqrt{\Delta} & 2\gamma \\ 2\alpha & \beta - \operatorname{sign}(p_2q_1 - p_1p_2)\sqrt{\Delta} \end{bmatrix}. \tag{2.6}$$

The ambiguity in $\mathbf{T}$ with respect to the premultiplication by a diagonal matrix is intrinsic since (2.2) is unchanged when $\mathbf{T}_{ij}$ is premultiplied by such a matrix. The ambiguity with respect to the permutation must be lifted by using the last condition in Proposition 2.1.

LEMMA 2.4. *With the same notations and under the same conditions as in Proposition* 2.2 *and supposing that* $\Delta > 0$*, then the diagonal elements* p_1'*,* p_2' *and* q_1'*,* q_2' *of the diagonalized matrices resulting from the pre- and postmultiplication of* $\mathbf{P}$ *and* $\mathbf{Q}$ *by the matrix* (2.6) *and its transpose conjugate satisfy*

$$p_2'q_1' - p_1'q_2' = 4\operatorname{sign}(p_2q_1 - p_1q_2)\sqrt{\Delta}\Delta|\beta - \operatorname{sign}(p_2q_1 - p_1p_2)\sqrt{\Delta}|^2.$$

We now apply the above results to the matrices $\mathbf{P}$ and $\mathbf{Q}$ defined in (2.3). Their diagonal and upper off-diagonal elements are 1, ω_{ij}, g_{ij} and ω_{ji}, 1, $\bar{g}_{ji}$, where

$$\omega_{ij} = \sum_{k=1}^{K} \frac{n_k}{n} \frac{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj}}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}} \tag{2.7}$$

[1] $\operatorname{sign}(0)$ could be either 1 or -1.

and g_{ij} is given in (2.4). Further, write $\omega_{ij}\omega_{ji}$ as

$$\sum_{k=1}^{K}\sum_{l=1}^{K}\frac{n_k}{n}\frac{n_l}{n}\frac{(\mathbf{BC}_k\mathbf{B}^*)_{jj}}{(\mathbf{BC}_k\mathbf{B}^*)_{ii}}\frac{(\mathbf{BC}_l\mathbf{B}^*)_{ii}}{(\mathbf{BC}_l\mathbf{B}^*)_{jj}}$$
$$=\frac{1}{2}\sum_{k=1}^{K}\sum_{l=1}^{K}\frac{n_k}{n}\frac{n_l}{n}\left[\frac{(\mathbf{BC}_k\mathbf{B}^*)_{jj}^{1/2}}{(\mathbf{BC}_k\mathbf{B}^*)_{ii}^{1/2}}\frac{(\mathbf{BC}_l\mathbf{B}^*)_{ii}^{1/2}}{(\mathbf{BC}_l\mathbf{B}^*)_{jj}^{1/2}}-\frac{(\mathbf{BC}_l\mathbf{B}^*)_{jj}^{1/2}}{(\mathbf{BC}_l\mathbf{B}^*)_{ii}^{1/2}}\frac{(\mathbf{BC}_k\mathbf{B}^*)_{ii}^{1/2}}{(\mathbf{BC}_k\mathbf{B}^*)_{jj}^{1/2}}\right]^2+1,$$

and one sees that $\omega_{ij}\omega_{ji}\geq 1$ with equality if and only if the ratio $(\mathbf{BC}_k\mathbf{B}^*)_{ii}/(\mathbf{BC}_k\mathbf{B}^*)_{jj}$ does not depend on k. The last condition is clearly equivalent to the condition that the matrices $\mathbf{P}$ and $\mathbf{Q}$ are proportional. It is then a matter of straightforward calculation, noting that the α, β, and γ in Proposition 2.1 satisfy $\beta=\Re\beta+(\alpha\gamma-\bar{\alpha}\bar{\gamma})/\Re\beta$, to obtain the following result.

COROLLARY 2.5. *Assuming that the sequence* $(\mathbf{BC}_k\mathbf{B}^*)_{ii}/(\mathbf{BC}_k\mathbf{B}^*)_{jj}$ *is not constant with respect to* k*, then a necessary and sufficient for the matrix* $\mathbf{T}_{ij}$ *to satisfy the conditions of Proposition* 2.1 *is that it is the product of a diagonal matrix with*

$$\begin{bmatrix}1&0\\0&1\end{bmatrix}-\frac{2}{1+h_{ij}h_{ji}-\bar{h}_{ij}\bar{h}_{ji}+\sqrt{(1+h_{ij}h_{ji}-\bar{h}_{ij}\bar{h}_{ji})^2-4h_{ij}h_{ji}}}\begin{bmatrix}0&h_{ij}\\h_{ji}&0\end{bmatrix},\tag{2.8}$$

where h_{ij} *and* h_{ji} *are the solution of*

$$\begin{bmatrix}\omega_{ij}&1\\1&\omega_{ji}\end{bmatrix}\begin{bmatrix}h_{ij}\\\bar{h}_{ji}\end{bmatrix}=\begin{bmatrix}g_{ij}\\\bar{g}_{ji}\end{bmatrix},\tag{2.9}$$

g_{ij} *and* ω_{ij} *being given in* (2.4) *and* (2.7).

2.2. Numerical considerations. The above corollary does not apply when the sequence $(\mathbf{BC}_k\mathbf{B}^*)_{ii}/(\mathbf{BC}_k\mathbf{B}^*)_{jj}$ is constant with respect to k, but in this case the matrices $\mathbf{P}$ and $\mathbf{Q}$ in (2.3) will be proportional and hence diagonalizing one would diagonalize the other and thus the conditions of Proposition 2.1 would be satisfied (the last one with an equality), only that there is now an infinite number of choices for the matrices $\mathbf{T}_{ij}$ (choices differing by the premultiplication with a diagonal matrix not counted as distinct).

However, when this sequence is nearly constant, direct computation of h_{ij} by $(\omega_{ji}g_{ij}-\bar{g}_{ji})/(\omega_{ij}\omega_{ji}-1)$ could be subjected to large error because of near cancellation of the numerator and denominator in this ratio. For better accuracy, it is preferable to solve (2.9) by the singular value decomposition. But since the matrix $\mathbf{B}$ have rows defined only up to a constant, the ω_{ij} and ω_{ji} can have widely different magnitudes, so we need to first "balance" the matrix in (2.9), by rewriting it as

$$\begin{bmatrix}\sqrt{\tilde{\omega}_{ij}}&0\\0&\sqrt{\tilde{\omega}_{ji}}\end{bmatrix}\begin{bmatrix}\sqrt{\omega_{ij}\omega_{ij}}&1\\1&\sqrt{\omega_{ij}\omega_{ji}}\end{bmatrix}\begin{bmatrix}\sqrt{\tilde{\omega}_{ij}}&0\\0&\sqrt{\tilde{\omega}_{ji}}\end{bmatrix},$$

where $\tilde{\omega}_{ij}=\sqrt{\omega_{ij}/\omega_{ji}}=1/\tilde{\omega}_{ji}$. The above middle matrix then has the singular value decomposition

$$\frac{1}{2}\begin{bmatrix}1&-1\\1&1\end{bmatrix}\begin{bmatrix}\sqrt{\omega_{ij}\omega_{ji}}+1&0\\0&\sqrt{\omega_{ji}\omega_{ij}}-1\end{bmatrix}\begin{bmatrix}1&1\\-1&1\end{bmatrix}.$$

It is thus inverted by simply inverting the diagonal elements of the above diagonal matrix. This yields as the solution to (2.9)

$$h_{ij}=\frac{\tilde{\omega}_{ji}g_{ij}+\bar{g}_{ji}}{2(\sqrt{\omega_{ij}\omega_{ji}}+1)}+\frac{\tilde{\omega}_{ji}g_{ij}-\bar{g}_{ji}}{2(\sqrt{\omega_{ij}\omega_{ji}}-1)}\tag{2.10}$$

and the same formula for h_{ji} by interchanging the indexes. When $\sqrt{\omega_{ij}\omega_{ji}} \approx 1$, the singular decomposition method would drop the second term in the above formula, as the numerator in this term should also be small.

There is an interesting interpretation of the above procedure. By the above singular decomposition, (2.9) is equivalent to

$$\begin{bmatrix} (\sqrt{\omega_{ij}\omega_{ji}}+1)\sqrt{\tilde{\omega}_{ij}} & (\sqrt{\omega_{ij}\omega_{ji}}+1)\sqrt{\tilde{\omega}_{ji}} \\ (1-\sqrt{\omega_{ij}\omega_{ji}})\sqrt{\tilde{\omega}_{ij}} & (\sqrt{\omega_{ij}\omega_{ji}}-1)\sqrt{\tilde{\omega}_{ji}} \end{bmatrix} \begin{bmatrix} h_{ij} \\ \bar{h}_{ji} \end{bmatrix} = \begin{bmatrix} \sqrt{\tilde{\omega}_{ij}}g_{ij} + \sqrt{\tilde{\omega}_{ij}}\bar{g}_{ji} \\ \sqrt{\tilde{\omega}_{ij}}\bar{g}_{ji} - \sqrt{\tilde{\omega}_{ji}}g_{ij} \end{bmatrix}.$$

One can recognize that the coefficients in the above equations and the corresponding right-hand sides are the second and first diagonal elements and upper off-diagonal elements of $\tilde{\mathbf{P}} = \sqrt{\tilde{\omega}_{ji}}\mathbf{P} + \sqrt{\tilde{\omega}_{ij}}\mathbf{Q}$ and of $\tilde{\mathbf{Q}} = \sqrt{\tilde{\omega}_{ij}}\mathbf{Q} - \sqrt{\tilde{\omega}_{ji}}\mathbf{P}$, where $\mathbf{P}$ and $\mathbf{Q}$ are defined in (2.3). Thus by the same argument leading to Corollary 2.5, the matrix (2.8) diagonalizes $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$. While this result is trivial since $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$ are just linear combinations of $\mathbf{P}$ and $\mathbf{Q}$, it shows that when the second term in the right-hand side of (2.10) is dropped, the matrix (2.8) would diagonalize $\tilde{\mathbf{P}}$ and $\operatorname{diag}\tilde{\mathbf{Q}}$, since $\tilde{\mathbf{P}}$ is positive definite and hence the result of Lemma 2.3 applies yielding that $(1+h_{ij}h_{ji}-\bar{h}_{ij}\bar{h}_{ji})^2 - 4h_{ij}h_{ji} > 0$. Thus this matrix also diagonalizes $\mathbf{P} + \sqrt{\tilde{\omega}_{ji}}(\tilde{\mathbf{Q}} - \operatorname{diag}\tilde{\mathbf{Q}})/2$ and $\mathbf{Q} - \sqrt{\tilde{\omega}_{ji}}(\tilde{\mathbf{Q}} - \operatorname{diag}\tilde{\mathbf{Q}})/2$. As the off-diagonal elements of $\tilde{\mathbf{Q}}$ are small, one sees that it still nearly diagonalizes $\mathbf{P}$ and $\mathbf{Q}$.

To avoid abrupt change in the way h_{ij} are computed, another possibility is to replace $\sqrt{\omega_{ij}\omega_{ji}} - 1$ in (2.10) by $\max(\sqrt{\omega_{ij}\omega_{ji}} - 1, \epsilon)$, where ϵ is a small number. The above argument can be repeated to show that $\mathbf{P}$ and $\mathbf{Q}$ are still nearly diagonalized.

3. Convergence of the algorithm. We have seen in the previous section that our algorithm decreases the criterion (1.1) and stops only when it reaches a stationary point of the criterion. However, we have not yet quantified this decrease.

3.1. Convergence of the gradient. We first derive a lower bound for the decrease of the criterion at each step of the algorithm, in terms of the gradient vector which is the vector with components ng_{ij} defined in (2.4).

LEMMA 3.1. *Let $\mathbf{T}_{ij}$ be the matrix satisfying the condition of Proposition* 2.1 *and let p'_1, p'_2, q'_1, q'_2 be as defined there, then the decrease of the criterion* (1.1) *associated with the transformation* (2.1) *is at least*

$$n \begin{bmatrix} \sqrt{q'_2/p'_2}\, g_{ij} \\ \sqrt{p'_1/q'_1}\, g_{ji} \end{bmatrix}^* \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sqrt{q'_2/p'_2}\, g_{ij} \\ \sqrt{p'_1/q'_1} g_{ji} \end{bmatrix}, \qquad \rho = \sqrt{p'_1 q'_2/(p'_2 q'_1)},$$

which can be bounded below by $n[(q'_2/p'_2)|g_{ij}|^2 + (p'_1/q'_1)|g_{ji}|^2]/2$.

Since the criterion always decreases during our algorithm, the decrease at each step must converge to zero, implying that $(q'_2/p'_2)|g_{ij}|^2 + (p'_1/q'_1)|g_{ji}|^2$ tends to zero. Still, this result hasn't proved the convergence to zero of the gradient vector. The difficulty is due to the lack of normalization. Indeed, our algorithm constructs the transformation matrix $\mathbf{B}$ *only up to a scaling of its rows*; hence a row of $\mathbf{B}$ can be arbitrary large or arbitrary small and this would affect the gradient. To avoid this, we shall renormalize the transformation matrices $\mathbf{B}$ after each step of the algorithm. Any reasonable normalization procedure will do, but for convenience, we will consider the normalization which makes the rows of $\mathbf{B}$ having unit norm. Then the diagonal elements of $\mathbf{B}\mathbf{C}_k\mathbf{B}^*$ will be bounded between the smallest and the largest eigenvalue of $\mathbf{C}_k$. Thus let m and M be the minimum and the maximum of all the eigenvalues of $\mathbf{C}_1, \ldots, \mathbf{C}_K$, then $m \le (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} \le M$ for all i and k. Note that $m > 0$ since the

matrices $\mathbf{C}_1, \ldots, \mathbf{C}_K$ are positive definite. It follows from the definition (2.3) of $\mathbf{P}$ and $\mathbf{Q}$ that both matrices $\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*$ and $\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*$ are bounded below by $1/M$ times the matrix

$$\mathbf{T}_{ij} \sum_{k=1}^{K} \begin{bmatrix} (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ij} \\ (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ji} & (\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{jj} \end{bmatrix} \mathbf{T}_{ij}^*$$

and above by $1/m$ times the same matrix; hence both p_1'/q_1' and p_2'/q_2' must lie in the interval $[m/M, M/m]$. This proves that the gradient vector of the criterion, evaluated at each step of the algorithm, converges to zero.

The above result shows that if the algorithm converges, then the limit must be a stationary point of the criterion. Further, since it always decreases the criterion, this point is actually a local minimum, unless the algorithm is started at a stationary point, in which case it stops immediately. Note that, the sequence of diagonalizing matrices constructed by the algorithm, being normalized and hence belonging to a compact set, will admit a convergent subsequence and this also holds for any of its subsequences. Therefore, *if the criterion admits a unique local minimum*, the algorithm will converge to it. However, Flury and Gautschi [6] have shown that in some extreme cases, the criterion (1.1) with orthogonality constraint admits more than one local minimum. Thus, it seems unlikely that the same criterion but without orthogonality constraint would admit a unique local minimum in all cases. Nevertheless, if there are only a finite number of local minima, one can still expect that the algorithm would converge to one of them. If this is not so, then since we have proved that the gradient vector converges to 0, the algorithm must jump continually from one local minimum to another, a highly implausible scenario.

3.2. Quadratic convergence. Our algorithm also has the nice properties that it behaves near the solution like the quasi-Newton–Raphson iteration, provided that the matrices $\mathbf{C}_1, \ldots, \mathbf{C}_K$ can be nearly jointly diagonalized. To derive the Newton–Raphson iteration, one makes a second order Taylor expansion of the criterion around the current point, then minimizes this expansion (instead of the true criterion) to obtain the new point. We have already derived the formula (2.5) for the change of the criterion corresponding to a change $\boldsymbol{\delta}\mathbf{B}$ of $\mathbf{B}$ and its expansion with respect to $\boldsymbol{\delta}$ up to the first order of the form $n\sum_{i\neq j} \Re(g_{ij}\bar{\delta}_{ij})$, where g_{ij} are given in (2.4) and δ_{ij} denote the elements of the matrix $\boldsymbol{\delta}$. We now need only to pursue the expansion to the second order. Thus we expand (2.5) as

$$n\sum_{i\neq j}\sum \Re(g_{ij}\bar{\delta}_{ij}) + \frac{1}{2}\sum_i\sum_l\sum_m \delta_{il}\bar{\delta}_{im} \sum_{k=1}^{K} n_k \frac{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{lm}}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}}$$
$$- \sum_i\sum_l\sum_m\sum_{k=1}^{K} n_k \frac{\Re[(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{il}\bar{\delta}_{il}]}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}} \frac{\Re[(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{im}\bar{\delta}_{im}]}{(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}} + n\sum_i\sum_j \Re(\delta_{ij}\delta_{ji}).$$

Assume that the matrices $\mathbf{C}_1, \ldots, \mathbf{C}_K$ can be nearly jointly diagonalized, then near the solution, the off-diagonal terms matrices $\mathbf{B}\mathbf{C}_k\mathbf{B}^*$ would be small relative to the diagonal terms. Hence we may neglect, in the above expression, the terms containing $(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{lm}/(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}$, $l \neq m$ or $(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{il}/(\mathbf{B}\mathbf{C}_k\mathbf{B}^*)_{ii}$, $l \neq i$ (other than the g_{ij}, of course). With this approximations, the above expression reduces to

$$n\sum_{i\neq j}\sum \Big[\Re(g_{ij}\bar{\delta}_{ij}) + \frac{1}{2}\omega_{ij}|\delta_{ij}|^2 + \frac{1}{2}\Re(\delta_{ij}\delta_{ji})\Big],$$

where ω_{ij} are as given in (2.7).

The quasi-Newton–Raphson algorithm consists of minimizing the above expression with respect to $\boldsymbol{\delta}$, then change $\mathbf{B}$ into $\mathbf{B} + \boldsymbol{\delta}\mathbf{B}$, $\boldsymbol{\delta}$ being the solution to this minimization. Note that the above expression can be written as

$$\frac{n}{2} \sum_{1 \le i < j \le K} \left([\delta_{ij} \;\; \bar{\delta}_{ji}] \begin{bmatrix} \bar{g}_{ij} \\ g_{ji} \end{bmatrix} + [\bar{\delta}_{ij} \;\; \delta_{ji}] \begin{bmatrix} g_{ij} \\ \bar{g}_{ji} \end{bmatrix} + [\bar{\delta}_{ij} \;\; \delta_{ji}] \begin{bmatrix} \omega_{ij} & 1 \\ 1 & \omega_{ji} \end{bmatrix} \begin{bmatrix} \delta_{ij} \\ \bar{\delta}_{ji} \end{bmatrix} \right).$$

Therefore, one can see that its minimization yields precisely $\delta_{ij} = -h_{ij}$, where h_{ij} are defined in (2.9). Note that the δ_{ii}, since they do not appear in the above expansion, can be anything as long as they are small. For convenience, we put them to zero. This is justified by the fact that by dividing the ith row of $\mathbf{B} + \boldsymbol{\delta}\mathbf{B}$ by $1 + \delta_{ii}$, one is led to the matrix $\mathbf{B} + \boldsymbol{\delta}'\mathbf{B}$, where $\boldsymbol{\delta}'$ has zero diagonal element and (i, j) off-diagonal element $\delta_{ij}/(1 + \delta_{ii})$, which is about the same as δ_{ij}.

The above quasi-Newton–Raphson algorithm appears very similar to our joint approximate diagonalization algorithm, with two differences as follows.

1. The off-diagonal term of $\mathbf{T}_{ij}$ in (2.1) is not δ_{ij} but contains the extra factor $2/[1 + h_{ij}h_{ji} - \bar{h}_{ij}\bar{h}_{ji} + \sqrt{(1 + h_{ij}h_{ji} - \bar{h}_{ij}\bar{h}_{ji})^2 - 4h_{ij}h_{ji}}]$.

2. Our algorithm operates on each pair of rows at a time while the quasi-Newton–Raphson algorithm operates on the whole matrix. Thus a sweep of the our algorithm is not quite the same as a Newton–Raphson iteration, since in our algorithm after a transformation (2.1) is made on the pair (i, j), the h_{kl} for k or l equal to i or j would undergo some change.

However, it can be seen that the above differences become negligible when the h_{ij} are small. Therefore, our algorithm should have about the same quadratic convergence speed as the quasi-Newton–Raphson iteration *near the solution*. But far from the solution (that is, at the beginning of the algorithm) it could have better convergence behavior since it always decreases the criterion while the Newton–Raphson iteration may not. It is true that we have been able to prove the convergence of our algorithm to a local minimum, but this cannot even be guaranteed in the Newton–Raphson iteration. Note, however, that our argument relies on the assumption that the matrices can be nearly jointly diagonalized. In the case where they cannot (which is also the case where more than one local minima often arise), this argument doesn't apply.

4. Some numerical examples. We consider the same example as in Flury and Gautschi [6]. The following 6×6 matrices are to be diagonalized:

$$\mathbf{C}_1 = \begin{bmatrix} 45 & 10 & 0 & 5 & 0 & 0 \\ 10 & 45 & 5 & 0 & 0 & 0 \\ 0 & 5 & 45 & 10 & 0 & 0 \\ 5 & 0 & 10 & 45 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.4 & -4.8 \\ 0 & 0 & 0 & 0 & -4.8 & 13.6 \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} 27.5 & -12.5 & -.5 & -4.5 & -2.04 & 3.72 \\ -12.5 & 27.5 & -4.5 & -.5 & 2.04 & -3.72 \\ -.5 & -4.5 & 24.5 & -9.5 & -3.72 & -2.04 \\ -4.5 & -.5 & -9.5 & 24.5 & 3.72 & 2.04 \\ -2.04 & 2.04 & -3.72 & 3.72 & 54.76 & -4.68 \\ 3.72 & -3.72 & -2.04 & 2.04 & -4.68 & 51.24 \end{bmatrix}.$$

We take $n_1 = n_2 = 1$ and start our algorithm with $\mathbf{B}$ being the identity matrix.

Table 1 reports the values of criterion after each sweep.

TABLE 1

Sweep	0	1	2	3	4
Criterion	0.809676	0.204339	0.00239435	$1.65756 \cdot 10^{-8}$	0

The last sweep produces a zero value of the criterion, because exact joint diagonalization can be achieved for two matrices. The above example is given only as a test to check that the algorithm works well, but it is not intended for jointly diagonalizing two matrices since in this case Algorithm 2.7.1 of Golub and van Loan [7] (based on the eigenvalue decomposition) would be more efficient. Our algorithm is, however, quite fast. Actually, after only 3 sweeps (sweep 0 corresponds to the initial matrices) the diagonalization is already rather good. We have

$$\mathbf{C}_1 = \begin{bmatrix} 39.0322 & -0.0038 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0038 & 29.8108 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \\ 0.0000 & 0.0000 & 60.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 50.0000 & -0.0000 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & 20.1550 & 0.0000 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 10.0449 \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} 30.7903 & 0.0023 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \\ 0.0023 & 40.0132 & -0.0000 & 0.0000 & -0.0000 & 0.0000 \\ -0.0000 & -0.0000 & 10.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 0.0000 & -0.0000 & 20.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & 0.0000 & 0.0000 & 59.1714 & -0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 48.4716 \end{bmatrix},$$

which corresponds to the transformation matrix

$$\mathbf{B} = \begin{bmatrix} 0.3975 & -0.3975 & -0.5754 & 0.5754 & -0.0664 & -0.1324 \\ -0.5529 & 0.5529 & -0.4204 & 0.4204 & -0.1688 & 0.0817 \\ 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.0000 & 0.0000 \\ -0.5000 & -0.5000 & 0.5000 & 0.5000 & 0.0000 & 0.0000 \\ -0.0749 & 0.0749 & -0.0443 & 0.0443 & 0.7790 & -0.6148 \\ 0.0073 & -0.0073 & -0.0272 & 0.0272 & 0.6082 & 0.7928 \end{bmatrix}.$$

(For definiteness, the rows of $\mathbf{B}$ have been normalized to have unit norm.) One can see that there are only two off-diagonal elements which are not quite zero, with a relative error (to the geometric mean of the corresponding diagonal elements) less than 10^{-4}. The fourth sweep zeros all off-diagonal elements of $\mathbf{C}_1$ and $\mathbf{C}_2$, up to 6 digits after the decimal point at least (we haven't checked further) with only a slight change in their diagonal elements: the first two diagonal elements of $\mathbf{C}_1$ and $\mathbf{C}_2$ become 39.0333 and 29.8099 and 30.7912 and 40.0120, the other are unchanged. The transformation matrix $\mathbf{B}$ is also almost unchanged: in fact only its first two rows have changed, to

$$\begin{matrix} 0.3977 & -0.3977 & -0.5752 & 0.5752 & -0.0664 & -0.1324 \\ -0.5527 & 0.5527 & -0.4206 & 0.4206 & -0.1688 & 0.0817 \end{matrix}.$$

This suggests that the algorithm has converged after the first transformation (2.1) of the fourth sweep and does not need this whole sweep. The Flury and Gautschi [6] algorithm needs 4 to 5 sweeps to converge and moreover it makes several iterations for each pair of indexes while we make only one. However, although we have used the same matrices, our algorithm does not solves quite the same problem, since we do not require the transformation matrix to be orthogonal. A simple way to implement the orthogonality constraint, at least approximately, is to add another matrix $\mathbf{C}_3$ which is the identity matrix and give it a large weight n_3. For $n_3 = 10$, ($n_1 = n_2 = 1$), the values of the criterion after each sweep are given in Table 2.

TABLE 2

Sweep	0	1	2	3	4	5
Criterion	0.809676	0.224022	0.0291158	0.0290464	0.0290454	0.0290454

The criterion does not decrease further after 4 sweeps. The change in the transformation matrix $\mathbf{B}$ produced by the fifth sweep is also very slight, affecting only the last digit and never more than 2 units. This matrix, after sweep 5, is

$$\mathbf{B} = \begin{bmatrix} 0.5000 & 0.5000 & -0.5000 & -0.5000 & 0.0000 & -0.0000 \\ -0.5556 & 0.5556 & -0.4227 & 0.4227 & -0.1327 & 0.0878 \\ 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.0000 & -0.0000 \\ 0.4219 & -0.4219 & -0.5664 & 0.5664 & -0.0088 & -0.0489 \\ -0.0918 & 0.0918 & -0.0523 & 0.0523 & 0.7918 & -0.5922 \\ 0.0085 & -0.0085 & -0.0186 & 0.0186 & 0.5979 & 0.8010 \end{bmatrix}$$

and the corresponding matrices $\mathbf{C}_1$, $\mathbf{C}_2$ are

$$\mathbf{C}_1 = \begin{bmatrix} 50.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 29.9224 & 0.0000 & -1.8186 & 2.1943 & 0.1092 \\ 0.0000 & 0.0000 & 60.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & -1.8186 & 0.0000 & 39.7221 & -0.7660 & 1.0341 \\ -0.0000 & 2.1943 & -0.0000 & -0.7660 & 20.2390 & -0.0385 \\ -0.0000 & 0.1092 & -0.0000 & 1.0341 & -0.0385 & 10.0240 \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} 20.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 \\ -0.0000 & 40.2097 & -0.0000 & -2.2700 & 4.3681 & 1.7309 \\ -0.0000 & -0.0000 & 10.0000 & -0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -2.2700 & -0.0000 & 31.7746 & -1.2684 & 7.1501 \\ -0.0000 & 4.3681 & -0.0000 & -1.2684 & 59.3949 & 0.5032 \\ -0.0000 & 1.7309 & -0.0000 & 7.1501 & 0.5032 & 48.3457 \end{bmatrix}.$$

These results are very similar to that of Flury and Gautschi [6]. (Note that our matrix $\mathbf{B}$ is the transpose of theirs.) Of course, the orthogonality constraint is not exactly satisfied here. We have

$$\mathbf{B}\mathbf{B}^* = \begin{bmatrix} 1.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 1.0000 & -0.0000 & 0.0117 & -0.0182 & -0.0047 \\ 0.0000 & -0.0000 & 1.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 0.0117 & -0.0000 & 1.0000 & 0.0059 & -0.0251 \\ 0.0000 & -0.0182 & 0.0000 & 0.0059 & 1.0000 & -0.0007 \\ 0.0000 & -0.0047 & 0.0000 & -0.0251 & -0.0007 & 1.0000 \end{bmatrix},$$

but the difference in this matrix from the identity matrix is slight. We should mention here that our algorithm is not designed to enforce orthogonality, the above numerical results are given only as examples showing its good convergence property.

Appendix. Proofs of results.

Proof of Proposition 2.1. Since the matrix $\mathbf{T}_{ij}$ is nonsingular, one can write any 2×2 matrix $\mathbf{T}$ in the form $[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}]\mathbf{T}_{ij}$ and hence (2.2) with $\mathbf{T}$ in place of $\mathbf{T}_{ij}$ equals

$$\begin{aligned} (\text{A.1}) \qquad & n \log \frac{|\det \mathbf{T}_{ij}|^2}{p'_1 q'_2} \\ & - n \log \frac{(|a|^2 p'_1 + a\bar{b}p' + \bar{a}b\bar{p}' + |b|^2 p'_2)(|d|^2 q'_2 + d\bar{c}\bar{q}' + \bar{d}cq' + |c|^2 q'_1)}{p'_1 q'_2 |ad - bc|^2}, \end{aligned}$$

where p'_1, p'_2, p' and q'_1, q'_2, q' are the diagonal and upper diagonal elements of $\mathbf{T}_{ij}\mathbf{P}\mathbf{T}_{ij}^*$ and $\mathbf{T}_{ij}\mathbf{Q}\mathbf{T}_{ij}^*$, respectively. One can recognize that the first term in (A.1) is the value of (2.2) at $\mathbf{T}_{ij}$, hence a necessary and sufficient condition that this point realizes the maximum of (2.2) is that the other term in (without the minus sign) is nonnegative for all a, b, c, d. But for $a = d = 1$ and $b,\ c \to 0$, this term can be seen to be equivalent to $n[(\bar{b}p' + b\bar{p}')/p'_1 + (\bar{c}\bar{q}' + cq')/q'_2]$; hence a necessary condition for it to be nonnegative is that $p' = q' = 0$. Under this condition, (A.1) reduces to

$$(\text{A.2}) \qquad n \log \frac{|\det \mathbf{T}_{ij}|^2}{p'_1 q'_2} - n \log \frac{(|a|^2 p'_1 + |b|^2 p'_2)(|d|^2 q'_2 + |c|^2 q'_1)}{p'_1 q'_2 (|ad|^2 - \bar{a}\bar{d}bc - ad\bar{b}\bar{c} + |bc|^2)}.$$

Again, for $a = d = 1$ and $b,\ c \to 0$, the last term in (A.2) can be seen to be equivalent to $n(|b|^2 p'_2/p'_1 + |c|^2 q'_1/q'_2 + bc + \bar{b}\bar{c})$. Therefore it is also necessary that this quadratic form in the variables $b,\ \bar{c}$ be nonnegative. This condition is satisfied if and only if $p'_2 q'_1 \geq p'_1 q'_2$. This yields the necessary part of the proposition.

To prove the sufficient part, note that by the inequality $-\log x \leq 1/x - 1$, (A.2) can be seen to be bounded above by

$$n \log \frac{|\det \mathbf{T}_{ij}|^2}{p'_1 q'_2} - n \frac{|bd|^2 p'_2 q'_2 + |ac|^2 p'_1 q'_1 + (\bar{a}\bar{d}bc + ad\bar{b}\bar{c}) p'_1 q'_2 + |bc|^2 (p'_2 q'_1 - p'_1 q'_2)}{(|a|^2 p'_1 + |b|^2 p'_2)(|d|^2 q'_2 + |c|^2 q'_1)}.$$

But for $p'_2 q'_1 \geq p'_1 q'_2$, this expression is bounded again by

$$(\text{A.3}) \quad n \log \frac{|\det \mathbf{T}_{ij}|^2}{p'_1 q'_2} - \frac{n}{(|a|^2 p'_1 + |b|^2 p'_2)(|d|^2 q'_2 + |c|^2 q'_1)} \begin{bmatrix} b\bar{d} \\ a\bar{c} \end{bmatrix}^* \begin{bmatrix} p'_2 q'_2 & p'_1 q'_2 \\ p'_1 q'_2 & p'_1 q'_1 \end{bmatrix} \begin{bmatrix} b\bar{d} \\ a\bar{c} \end{bmatrix}$$

and thus it cannot exceed $n \log[|\det \mathbf{T}_{ij}|^2/(p'_1 q'_2)]$ for all choices of $a,\ b,\ c,\ d$, since the matrix in its second term is positive definite. This yields the sufficient part of the proposition. □

Proof of Proposition 2.2. Since $\mathbf{P}$ and $\mathbf{Q}$ are nonproportional, the matrix $[\begin{smallmatrix} p_1 & p_2 & \Re p & \Im p \\ q_1 & q_2 & \Re q & \Im q \end{smallmatrix}]$ is of full rank. Hence it admits at least a 2×2 submatrix with nonzero determinant, which entails that α, β, γ are not all zero. To prove that Δ is real, we expand it as

(A.4)

$$\begin{aligned} & (p_1 q_2 - p_2 q_1 + \bar{p}q - p\bar{q})^2 - 4(p_2\bar{q} - \bar{p}q_2)(pq_1 - p_1 q) \\ & = (p_1 q_2 - p_2 q_1 + \bar{p}q - p\bar{q})^2 - 4(p_2 q_1 p\bar{q} + p_1 q_2 \bar{p}q) + 4(|p|^2 q_1 q_2 + p_1 p_2 |q|^2) \\ & = (p_1 q_2 - p_2 q_1)^2 + (\bar{p}q - p\bar{q})^2 - 2(p_1 q_2 + p_2 q_1)(\bar{p}q + p\bar{q}) + 4(|p|^2 q_1 q_2 + p_1 p_2 |q|^2). \end{aligned}$$

Consider now the solution to the joint diagonalization problem. Let $[a \;\; b]$ and $[c \;\; d]$ be the rows of the diagonalizing matrix $\mathbf{T}$. The condition that the transformed matrices be diagonal can be written as

$$\text{(A.5)} \qquad \begin{bmatrix} [a & b]\mathbf{P} \\ [a & b]\mathbf{Q} \end{bmatrix} \begin{bmatrix} \bar{c} \\ \bar{d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{or equivalently} \quad \begin{bmatrix} [c & d]\mathbf{P} \\ [c & d]\mathbf{Q} \end{bmatrix} \begin{bmatrix} \bar{a} \\ \bar{b} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We shall exclude the trivial case where $[a\ b]$ or $[c\ d]$ is zero. Then (A.5) implies that the matrices in the left-hand side have zero determinants, i.e.,

$$(pa + p_2 b)(\bar{q}b + q_1 a) - (qa + q_2 b)(\bar{p}b + p_1 a) = 0,$$

and the same equation but with a, b replaced by c, d. After expansion, one gets the equations $\alpha b^2 - \beta ab + \gamma a^2 = 0$ and $\alpha d^2 - \beta cd + \gamma c^2 = 0$.

The solution $[a \;\; b]$ to the equation $\alpha b^2 - \beta ab + \gamma a^2 = 0$ is clearly determined only up to a multiplicative factor. Let δ be any one of the two square roots of Δ, it can be seen that for $\alpha\gamma \neq 0$ the solution is proportional to $[2\alpha \;\; \beta + \delta]$ or $[\beta - \delta \;\; 2\gamma]$, these two vectors being proportional. If $\alpha = 0$, then the solution is proportional to $[0 \;\; 1]$ while if $\gamma = 0$, it is proportional to $[1 \;\; 0]$ and if both are zero, then it can be proportional to $[0 \;\; 1]$ or $[1 \;\; 0]$. Thus in all cases, the solution is proportional to $[2\alpha \;\; \beta + \delta]$ or $[\beta - \delta \;\; 2\gamma]$, since $\delta = \pm\beta$ when $\alpha\gamma = 0$. Similarly, the solutions $[c \;\; d]$ to the equation $\alpha d^2 - \beta cd + \gamma c^2 = 0$ must be proportional to $[2\alpha \;\; \beta + \delta']$ or $[\beta - \delta' \;\; 2\gamma]$, where δ' is also a square root of Δ.

The above results provide only the necessary form of the solutions to the diagonalization problem and further they don't say how δ' is related to δ. To see if a choice for $[a \;\; b]$ and $[c \;\; d]$ as given above is admissible, one must check that it satisfies (A.5).

We begin with the choice $[a \;\; b] = [2\alpha \;\; \beta + \delta]$ and $[c \;\; d] = [2\alpha \;\; \beta \pm \bar{\delta}]$. We have

$$\begin{aligned} [a \;\; b]\mathbf{P}[c \;\; d]^* &= (ap_1 + b\bar{p})\bar{c} + (ap + bp_2)\bar{d} \\ &= (2\alpha p_1 + \beta\bar{p} + \delta\bar{p})2\bar{\alpha} + (2\alpha p + \beta p_2 + \delta p_2)(\bar{\beta} \pm \delta). \end{aligned}$$

The last expression can be expanded as

$$\begin{aligned} &2\alpha(2p_1\bar{\alpha} + p\bar{\beta}) + 2\bar{\alpha}\bar{p}\beta + |\beta|^2 p_2 + [2\bar{\alpha}\bar{p} + \bar{\beta}p_2 \pm (2\alpha p + \beta p_2)]\delta \pm \Delta p_2 \\ &= 2\alpha(2p_1\bar{\alpha} + 2p_2\gamma + p\bar{\beta}) + (p_2\bar{\beta} - p_2\beta + 2\bar{p}\bar{\alpha})\beta + (\Delta \pm \Delta)p_2 + [2\bar{\alpha}\bar{p} + \bar{\beta}p_2 \pm (2\alpha p + \beta p_2)]\delta. \end{aligned}$$

Using the following relations, a consequence of the definition of α, β, and γ,

$$\text{(A.6)} \qquad 2(p_1\bar{\alpha} + p_2\gamma) + p(\beta + \bar{\beta}) = 0, \qquad 2(\bar{p}\bar{\alpha} - p\alpha) = p_2(\beta - \bar{\beta}),$$

the above expression can be seen to vanish if the minus sign is used in $\pm$. Hence $[a \;\; b]\mathbf{P}[c \;\; d]^* = 0$ for the choice $[c \;\; d] = [2\alpha \;\; \beta - \bar{\delta}]$. A similar calculation, with q_1, q_2, q in place of p_1, p_2, p and based on the relations

$$\text{(A.7)} \qquad q_1\bar{\alpha} + q_2\gamma + q(\beta + \bar{\beta})/2 = 0, \qquad \bar{q}\bar{\alpha} - q\alpha = q_2(\beta - \bar{\beta})/2,$$

(which is also a consequence of the definition of α, β, and γ) yields that $[a \;\; b]\mathbf{Q}[c \;\; d]^* = 0$ for the same choice of $[a \;\; b]$ and $[c \;\; d]$. One does not need to explore other choices of $[c \;\; d]$ for this $[a \;\; b]$, since for a given $[a \;\; b]$, the solutions $[c \;\; d]$ to the first equation in (A.5) must be all proportional to a same vector, unless $[a \;\; b]\mathbf{P} = [0 \;\; 0] = [a \;\; b]\mathbf{Q}$, which is excluded since this implies that $\mathbf{P}$ and $\mathbf{Q}$ are proportional.

For the choice $[a \;\; b] = [\beta - \delta \;\; 2\gamma]$, observe that $[a \;\; b]\mathbf{P}[c \;\; d]$ and $[a \;\; b]\mathbf{Q}[c \;\; d]$ remain the same by interchanging a with b, c with d, p_1 with p_2, p with $\bar{p}$, q_1 with q_2, and q with $\bar{q}$. But then one is led to the same calculations as before by interchanging α with γ and reversing the sign of δ. Noting that the second relations in (A.6) and (A.7) still hold under these interchanges while the first relations remain the same, one gets that the correct choice for $[c \;\; d]$ is $[\beta + \bar{\delta} \;\; 2\gamma]$. As before, for this $[a \;\; b]$, one need not explore other choices for $[c \;\; d]$. □

Proof of Lemma 2.3. We shall prove the result only for the case $\det \mathbf{Q} > 0$ since the proof for the other case is similar. Continue the calculation of Δ in (A.4) and noting that $q_1 q_2 = \det \mathbf{Q} + |q|^2 > 0$, we get

$$\begin{aligned}\Delta &= (p_1 q_2 - p_2 q_1)^2 + (\bar{p}q - p\bar{q})^2 + 4p_1 p_2 |q|^2 \\ &\quad + 4q_1 q_2 \Big| p - \frac{p_1 q_2 + p_2 q_1}{2q_1 q_2} q \Big|^2 - \frac{(p_1 q_2 + p_2 q_1)^2}{q_1 q_2} |q|^2 \\ &= (p_1 q_2 - p_2 q_1)^2 \Big(1 - \frac{|q|^2}{q_1 q_2}\Big) + (\bar{p}q - p\bar{q})^2 + 4q_1 q_2 \Big| p - \frac{p_1 q_2 + p_2 q_1}{2q_1 q_2} q \Big|^2.\end{aligned}$$

Putting $r = p - [(p_1 q_2 + p_2 q_1)/(2q_1 q_2)]q$, the last two terms in the above right-hand side can be written as

$$(\bar{r}q - r\bar{q})^2 + 4q_1 q_2 |r|^2 = 4q_1 q_2 |r|^2 - [2\Im(\bar{r}q)]^2 \geq 4|r|^2 (q_1 q_2 - |q|^2).$$

Thus $\Delta \geq 0$ with equality if and only if $p_1 q_2 - p_2 q_1 = 0$ and $r = 0$. Since $q_1 q_2 > 0$, the first condition implies that $p_1 = \lambda q_1$ and $p_2 = \lambda q_2$ for some λ. Then the second condition implies that $p - \lambda q = 0$. These two conditions thus entail that $\mathbf{P}$ is proportional to $\mathbf{Q}$, contradicting our assumption. □

Proof of Lemma 2.4. From the definition (2.6) of the diagonalizing matrix, one has, putting $D = \beta - \operatorname{sign}(p_2 q_1 - p_1 q_2)\sqrt{\Delta}$,

$$\begin{aligned} p_1' &= p_1 |D|^2 + 2(\bar{p}\gamma \bar{D} + p\bar{\gamma} D) + 4p_2 |\gamma|^2, \\ q_2' &= q_2 |D|^2 + 2(q\alpha \bar{D} + \bar{q}\bar{\alpha} D) + 4q_1 |\alpha|^2. \end{aligned}$$

Hence the product $p_2' q_1'$ equals

$$\begin{aligned} &p_2 q_1 |D|^4 + 4\Re(p_2 \bar{q}\gamma \bar{D} + q_1 p \alpha D)|D|^2 + 4(p_2 q_2 |\gamma D|^2 + p_1 q_1 |\alpha D|^2) \\ &+ 8\Re(p\bar{q}\alpha\gamma \bar{D}^2 + pq\alpha\bar{\gamma}|D|^2) + 16\Re(pq_2 |\gamma|^2 \alpha \bar{D} + p_1 \bar{q} |\alpha|^2 \gamma \bar{D}) + p_1 q_2 |4\alpha\gamma|^2. \end{aligned}$$

The product $p_1' q_2'$ can be obtained from the above formula by interchanging p_1, p_2, p with q_1, q_2, q. Therefore, the difference $p_2' q_1' - p_1' q_2'$ equals

$$\begin{aligned} p_2' q_1' - p_1' q_2' &= (p_2 q_1 - p_1 q_2)(|D|^4 - |4\alpha\gamma|^2) \\ &\quad + 8\Re(\alpha\gamma \bar{D})|D|^2 + 8\Re[(p\bar{q} - p\bar{q})\alpha\gamma \bar{D}^2] - 32\Re(|\alpha\gamma|^2 \bar{D}). \end{aligned} \tag{A.8}$$

The last three terms in the right-hand side above may be regrouped as

$$\Re[(8\alpha\gamma \bar{D}^2)(D + p\bar{q} - \bar{p}q - 4\bar{\alpha}\bar{\gamma}/\bar{D})]. \tag{A.9}$$

However, putting $\delta = \beta - D = \operatorname{sign}(p_1 q_2 - p_2 q_1)\sqrt{\Delta}$, one has $D = \beta + \delta$, $4\alpha\gamma = \beta^2 - \delta^2 = D(\beta + \delta)$; hence, noting that $p\bar{q} - \bar{p}q = (\bar{\beta} - \beta)/2$,

$$\begin{aligned} D + p\bar{q} - \bar{p}q - 4\bar{\alpha}\bar{\gamma}/\bar{D} &= \beta - \delta + (\bar{\beta} - \beta)/2 - (\bar{\beta} + \delta) = (\beta - \bar{\beta})/2 - 2\delta, \\ 4\alpha\gamma \bar{D}^2 &= |D|^2 (\beta + \delta)(\bar{\beta} - \delta) = |D|^2 [|\beta|^2 - \Delta + (\bar{\beta} - \beta)\delta]. \end{aligned}$$

Therefore (A.9) reduces to

$$|D|^2\Re\{[|\beta|^2 - \Delta + (\bar\beta - \beta)\delta](\beta - \bar\beta - 4\delta)\} = |D|^2\delta[4(\Delta - |\beta|^2) - (\beta - \bar\beta)^2]$$

since $\beta - \bar\beta$ is purely imaginary. Finally, from $4\alpha\gamma = D(\beta + \delta)$ and $p_2q_1 - p_1q_2 = -(\beta + \bar\beta)/2$, the first term in the right-hand side of (A.8) equals

$$\frac{1}{2}(\beta + \bar\beta)|D|^2(|\beta + \delta|^2 - |\beta - \delta|^2) = |D|^2\delta(\beta + \bar\beta)^2.$$

Combining the above result, one gets

$$p_2'q_1' - p_1'q_2' = |D|^2\delta[(\beta + \bar\beta)^2 + 4(\Delta - |\beta|^2) - (\beta - \bar\beta)^2] = 4|D|^2\delta\Delta,$$

yielding the result of the lemma. □

Proof of Lemma 3.1. We have shown in the proof of Proposition 2.1 that the expression (2.2) evaluated at the matrix $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]\mathbf{T}_{ij}$ is given by (A.2), which is bounded above by (A.3). Take a, b, c, d to be the elements of the inverse of $\mathbf{T}_{ij}$; then (2.2) becomes zero and thus (A.3) with these values of a, b, c, d is nonnegative. Therefore, noting that a, b, c, d by definition satisfy

$$1 = |a|^2p_1' + |b|^2p_2', \quad 1 = |d|^2q_2' + |c|^2q_1', \quad \begin{bmatrix} g_{ij} \\ g_{ji} \end{bmatrix} = \begin{bmatrix} p_2' & p_1' \\ q_2' & q_1' \end{bmatrix} \begin{bmatrix} b\bar d \\ a\bar c \end{bmatrix},$$

one gets

$$n \log \frac{|\det \mathbf{T}_{ij}|^2}{p_1'q_2'} \geq n \begin{bmatrix} g_{ij} \\ g_{ji} \end{bmatrix}^* \begin{bmatrix} p_2' & q_2' \\ p_1' & q_1' \end{bmatrix}^{-1} \begin{bmatrix} p_2'q_2' & p_1'q_2' \\ p_1'q_2' & p_1'q_1' \end{bmatrix} \begin{bmatrix} p_2' & p_1' \\ q_2' & q_1' \end{bmatrix}^{-1} \begin{bmatrix} g_{ij} \\ g_{ji} \end{bmatrix}.$$

But the left-hand side of the above inequality is a lower bound for the decrease of the criterion (1.1) associated with the transformation (2.1) while the right-hand side can be rearranged to obtain the same bound as given in the lemma. Since the matrix which appears there has eigenvalues $1 \pm \rho$ and $0 \leq \rho \leq 1$, one gets the second result of this lemma. □

REFERENCES

[1] A. Belouchrami, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Process., 45 (1977), pp. 434–444.

[2] J.-F. Cardoso, *On the performance of orthogonal source separation algorithms*, in Signal Processing VII, Proceedings of the European Association for Signal Processing '94, Edinburgh, Scotland, 1994, pp. 776–779.

[3] J.-F. Cardoso and A. Souloumiac, *Blind beam forming for non Gaussian signals*, IEE Proceedings-F, 140 (1993), pp. 362–370.

[4] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[5] B. N. Flury, *Common principal components in k groups*, J. Amer. Statist. Assoc., 79 (1984), pp. 892–897.

[6] B. N. Flury and W. Gautschi, *An algorithm for the simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly orthogonal form*, SIAM J. Sci. Statist. Comp., 7 (1986), pp. 169–184.

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

[8] D. T. Pham and P. Garat, *Blind separation of mixtures of independent sources through a quasi maximum likelihood approach*, IEEE Trans. Signal Process., 45 (1997), pp. 1712–1725.

[9] G. A. F. Seber, *Multivariate Observations*, Wiley, New York, 1984.
[10] A. Yeredor, *Approximate joint diagonalization using non orthogonal matrices*, in Proceedings of the ICA 2000 Conference, Helsinki, University of Technology, Helsinki, Finland, 2000, pp. 33–38.
[11] A. Yeredor, *Optimization of a second-order statistics blind separation algorithm for Gaussian signals*, in Signal Processing X, Proceedings of the European Association for Signal Processing 2000 Conference, Tempere, Finland, 2000, pp. 19–22.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1153–1174

ACCURATE SOLUTION OF WEIGHTED LEAST SQUARES BY ITERATIVE METHODS*

ELENA Y. BOBROVNIKOVA† AND STEPHEN A. VAVASIS‡

Abstract. We consider the weighted least-squares (WLS) problem with a very ill-conditioned weight matrix. WLS problems arise in many applications including linear programming, electrical networks, boundary value problems, and structures. Because of roundoff errors, standard iterative methods for solving a WLS problem with ill-conditioned weights may not give the correct answer. Indeed, the difference between the true and computed solution (forward error) may be large. We propose an iterative algorithm, called MINRES-L, for solving WLS problems. The MINRES-L method is the application of MINRES, a Krylov-space method due to Paige and Saunders [*SIAM J. Numer. Anal.*, 12 (1975), pp. 617–629], to a certain layered linear system. Using a simplified model of the effects of roundoff error, we prove that MINRES-L ultimately yields answers with small forward error. We present computational experiments for some applications.

Key words. weighted least squares, iterative method, MINRES, conjugate gradient, Krylov-space, achievable accuracy

AMS subject classifications. 65F10, 65N22

PII. S0895479897316576

1. Introduction.

Consider the *weighted least-squares* (WLS) problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} \|D^{1/2}(\mathbf{b}-A\mathbf{x})\|^2, \tag{1.1}$$

where $D \in \mathbf{R}^{m\times m}$, $A \in \mathbf{R}^{m\times n}$, $\mathbf{b} \in \mathbf{R}^m$, $m \geq n$, and $\mathbf{x} \in \mathbf{R}^n$ is the unknown. In this formula and for the remainder of this article, $\|\cdot\|$ indicates the 2-norm. The normal equations for (1.1) have the form

$$A^T D A\mathbf{x} = A^T D\mathbf{b}. \tag{1.2}$$

We make the following assumptions: D is a diagonal positive definite matrix and $\operatorname{rank} A = n$. These assumptions imply that (1.2) is a nonsingular linear system with a unique solution.

WLS problems arise in several application domains including linear programming, electrical power networks, elliptic boundary value problems, and structural analysis, as observed by Strang [27]. This article focuses on the case when D is severely ill-conditioned. This happens in certain classes of electrical power networks. In this case, A is a node-arc adjacency matrix, D is matrix of load conductivities, $\mathbf{b}$ is the

*Received by the editors February 12, 1997; accepted for publication (in revised form) by R. Freund November 27, 2000; published electronically March 20, 2001. This work has been supported in part by an NSF Presidential Young Investigator grant, with matching funds received from AT&T and Xerox Corp. Research supported in part by NSF through grant DMS-9505155 and ONR through grant N00014-96-1-0050. Support was also received from the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Dept. of Energy, under contract W-31-109-Eng-38 through Argonne National Laboratory. Support was also received from the J. S. Guggenheim Foundation. This work was also supported in part by NSF grants CCR-9619489 and EIA-9726388.

http://www.siam.org/journals/simax/22-4/31657.html

†Formerly of the Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. Part of this work was done while this author was visiting Lucent Bell Laboratories.

‡Department of Computer Science, Cornell University, Ithaca, NY 14853 (vavasis@ cs.cornell.edu). Part of this work was done while this author was visiting the Argonne National Laboratory.

vector of voltage sources, and $\mathbf{x}$ is the vector of voltages of the nodes. Ill-conditioning occurs when resistors are out of scale, for instance, when modeling leakage of current through insulators.

Ill-conditioning also occurs in linear programming when an interior-point method is used. To compute the Newton step for an interior-point method, we need to solve a weighted least-squares equation of the form (1.2). Matrix D becomes ill-conditioned as the iterates approach the boundary of the feasible region, which always happens in an interior point method. In section 8, we examine this application in more detail. Ill-conditioning also occurs in finite element methods for certain classes of boundary value problems, for example, in the heat equilibrium equation $\nabla \cdot (c\nabla u) = 0$ when thermal conductivity field c varies widely in scale.

An important property of problem (1.1) or (1.2) is the norm bound on the solution, which was obtained independently by Stewart [26], Todd [28], and several other authors. See [10] for a more complete bibliography. Here we state this result as in the paper by Stewart.

THEOREM 1.1. *Let $\mathcal{D}$ denote the set of all positive definite $m \times m$ real diagonal matrices. Let A be an $m \times n$ real matrix of rank n. If we define*

$$\chi_A = \sup\{\|(A^T DA)^{-1} A^T D\| : D \in \mathcal{D}\} \qquad \textit{and} \tag{1.3}$$

$$\bar{\chi}_A = \sup\{\|A(A^T DA)^{-1} A^T D\| : D \in \mathcal{D}\}, \tag{1.4}$$

then both $\chi_A, \bar{\chi}_A$ are finite.

Note that the matrix appearing in (1.3) is the solution operator for the normal equations (1.2). In other words, (1.2) can be rewritten as $\mathbf{x} = (A^T DA)^{-1} A^T D\mathbf{b}$.

Since the bounds (1.3), (1.4) exist, we can hope that there exist algorithms for (1.2) that possess the same property, namely, the forward error bound does not depend on D. We will call these algorithms stable, where *stability* as defined by Vavasis [29] means that forward error in the computed solution $\hat{\mathbf{x}}$ satisfies

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon \cdot f(A) \cdot \|\mathbf{b}\|, \tag{1.5}$$

where ϵ is machine precision and $f(A)$ is some function of A not depending on D. Note that the underlying rationale for this kind of bound is that the conditioning problems in (1.1) stem from an ill-conditioned D rather than an ill-conditioned A.

This stability property is not possessed by standard direct methods such as QR factorization, Cholesky factorization, symmetric indefinite factorization, and range-space and null-space methods nor by standard iterative methods such as conjugate gradient applied to (1.2). The only two algorithms in the literature that are proved to have this property are the NSH algorithm by Vavasis [29] and the complete orthogonal decomposition (COD) algorithm by Hough and Vavasis [17], both of them direct. In section 3 we consider previous work on iterative methods. See Björck [2] for more information about algorithms for least-squares problems.

We would like to have stable iterative methods for this problem because iterative methods can be much more efficient than direct methods for large sparse problems, which is the common setting in applications.

This article presents an iterative algorithm for WLS problems called MINRES-L. The description and theory of the algorithm is presented in sections 4 through 6. This is followed by computational experiments in section 7. Briefly, MINRES-L consists of applying the MINRES algorithm of Paige and Saunders [21] to a certain layered linear system. We prove that MINRES-L satisfies (1.5). This proof of the forward

error bound for MINRES-L is based on a simplified model (presented in section 5) of how roundoff error affects Krylov space methods. An analysis of roundoff in MINRES-L starting from first principles is not presented here because the effect of roundoff on the MINRES iteration is not fully understood.

MINRES-L imposes the additional assumption on the WLS problem instance that D is "layered." In section 2 we state the layering assumption, and also present the layered least-squares (LLS) problem. This assumption is made without loss of generality (i.e., every WLS problem can be rewritten in layered form), but the MINRES-L algorithm is inefficient for problems with many layers.

Our analysis concerns the ultimate achievable accuracy of MINRES-L (versus other iterative algorithms) and does not address the matter of convergence rate. In section 7 we remark on this matter and on the use of preconditioners.

2. The layering assumption. Recall that we have already assumed that the weight matrix D appearing in (1.1) is diagonal, positive definite, and ill-conditioned. For the rest of this article we impose an additional "layering" assumption: we assume, after a suitable permutation of the rows of $(A, \mathbf{b})$ and corresponding symmetric permutation of D, that D has the structure

$$D = \begin{pmatrix} \delta_1 D_1 & & \\ & \ddots & \\ & & \delta_p D_p \end{pmatrix}, \tag{2.1}$$

where each D_k is well-conditioned and scaled so that its smallest diagonal entry is 1, and where $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_p > 0$. Let κ denote the maximum diagonal entry among $D_1, \ldots, D_p$. The layering assumption is that κ is not much larger than 1.

Note that this assumption is made without any loss of generality (and we could assume $\kappa = 1$), since we could place each diagonal entry of D in its own layer. Unfortunately, the complexity of each iteration of our algorithm grows like a high power of p. In particular, the size of the system under consideration grows quadratically with p. The work per iteration grows approximately like p^3, and the number of iterations is also expected to grow with p. Thus, computational work is expected to grow at a rate of perhaps p^4. Furthermore, our upper bound on the forward error degrades as p increases (see (6.11) below). Thus, a tacit assumption is that the number of layers p is not too large.

From now on, we write A in partitioned form as

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix}$$

to correspond with the partitioning of D. We partition $\mathbf{b} = [\mathbf{b}_1; \ldots; \mathbf{b}_p]$ conformally.

Under this assumption, we say that (1.1) is a "layered WLS" problem. In the context of electrical networks, this assumption means that there are several distinct classes of wires in the circuit, where the resistance of wires in class l is of order $1/\delta_l$. For instance, one class of wires might be transmission lines, whereas the other class might consist of broken wires (open lines) where the resistance is much higher. In the context of the heat equilibrium equation, the layering assumption means that the object under consideration is composed of a small number of different materials. Within each material the conductivity δ_l is roughly constant, but the different materials have very

different conductivities. In linear programming, taking $p = 2$ means that some of the variables at the current interior-point iterate are "small" while others are "large."

A limiting case of layered WLS occurs when the gaps between the δ_l's tend to infinity, that is, δ_1 is infinitely larger than δ_2 and so on. As the weight gaps tend to infinity, the solution to (1.1) tends to the solution of the following problem, which we refer to as *layered least-squares* (LLS). Construct a sequence of nested affine subspaces $L_0 \supset L_1 \supset \cdots \supset L_p$ of $\mathbf{R}^n$. These spaces are defined recursively: $L_0 = \mathbf{R}^n$, and

$$L_l = \{\text{minimizers of } \|D_l^{1/2}(A_l\mathbf{x} - \mathbf{b}_l)\| \text{ subject to } \mathbf{x} \in L_{l-1}\}.$$

Finally, $\mathbf{x}$, the solution to the LLS problem, is the unique element in L_p. The LLS problem was first introduced by Vavasis and Ye [30] as a technique for accelerating the convergence of interior-point methods. They also established the result mentioned above in this paragraph: the solution to the WLS problem in the limit as $\delta_{l+1}/\delta_l \to 0$ for all l converges to the solution of the LLS problem.

3. Previous work. A standard iterative method for least-squares problems, including WLS problems, is the LSQR algorithm of Paige and Saunders [22]. Most of our test cases below in section 7 compare MINRES-L to LSQR.

LSQR is closely related to CGNR, which is analytically equivalent to the conjugate gradient method (see Golub and Van Loan [11] or Greenbaum [13]) applied to the normal equations (1.2). There are several variants of CGNR in the literature; see, e.g., Björck, Elfving, and Strakoš [3].

The difficulty with CGNR and LSQR is that an inaccurate solution can be returned because $A^T DA$ can be ill-conditioned when D is ill-conditioned. To understand the difficulty, consider the two-layered WLS problem, which is obtained by substituting (2.1) in the case $p = 2$ into (1.2):

$$\delta_1 A_1^T D_1 A_1 \mathbf{x} + \delta_2 A_2^T D_2 A_2 \mathbf{x} = \delta_1 A_1^T D_1 \mathbf{b}_1 + \delta_2 A_2^T D_2 \mathbf{b}_2. \tag{3.1}$$

Observe that if $\delta_1 \gg \delta_2$, then the Krylov sequence

$$A^T D\mathbf{b}, (A^T DA)A^T D\mathbf{b}, (A^T DA)^2 A^T D\mathbf{b}, \ldots$$

constructed by CGNR and LSQR is very close to

$$\delta_1 A_1^T D_1 \mathbf{b}_1, \delta_1^2 (A_1^T D_1 A_1) A_1^T D_1 \mathbf{b}_1, \delta_1^3 (A_1^T D_1 A_1)^2 A_1^T D_1 \mathbf{b}_1, \ldots .$$

Indeed, a naive implementation of CGNR would form $A^T D\mathbf{b}$ on the first iteration of the algorithm and never reintroduce any information about $\mathbf{b}$ again. This naive implementation would clearly lose information about $\mathbf{b}_2$ due to roundoff error when forming $A^T D\mathbf{b}$ if $\delta_1 \gg \delta_2$, and the lost information is never recovered.

LSQR uses $\mathbf{b}$ in a more sophisticated manner than naive CGNR but nonetheless still has a difficulty with loss of information in the Krylov space. The difficulty with LSQR (as well as with a good implementation of CGNR like CGLS1 [3]) is as follows. LSQR computes a basis for the Krylov space using a short recurrence. As noted in the last paragraph, initially each basis vector contains mostly information from A_1^T plus a small contribution from A_2^T. After a certain number of iterations, these algorithms converge upon an approximate solution to the (possibly rank-deficient) normal equations $A_1^T D_1 A_1 \mathbf{x} = A_1^T D_1 \mathbf{b}_1$. At this point, the residual drops abruptly. See, e.g., Figure 7.2 for an example of the sudden decrease in the residual. After this point, in exact arithmetic, the future basis vectors for the Krylov space will contain

information mostly about directions spanned by A_2^T orthogonal to the approximate basis already constructed for A_1^T. Unfortunately, the information about A_2^T, which is present in the first iterate, is not propagated accurately to this transition step. Much information is lost because of cancellation when the residual decreases abruptly. A different framework for interpreting this difficulty is described in section 5.

In light of this explanation of the difficulty, one could consider a remedy of a correction to the residual and search directions at the transition step mentioned in the preceding paragraph. Indeed, several authors have proposed an iterative method along these lines in the case of the heat equation for composite materials mentioned in the introduction, e.g., [31]. (Note that the use of the word "layered" in the title of [31] has a different meaning from the use of that word herein.) This class of methods uses knowledge about the form of A and D that arise in discretizing a second-order elliptic boundary value problem to come up with the correct adjustment.

We expect this class of algorithms to be more efficient than the MINRES-L algorithm proposed in subsequent sections of this paper. On the other hand, they require information about the problem domain; in contrast, MINRES-L requires only $(A, D, \mathbf{b})$ as problem input. Thus, MINRES-L appears to be more generally applicable than adjustment.

We proposed another method based on correcting the search directions in our own earlier work [4] that attempted to extract the necessary adjustment directly from $(A, D, \mathbf{b})$ and Krylov-space information computed by CG. We have not pursued that approach because we found a case that was not correctly handled by our proposed algorithm. That technique was reminiscent of reorthogonalization—a standard way to combat ill-conditioning in iterative methods; see, for example, Paige [23] and Parlett and Scott [24].

Another technique for addressing ill-conditioned linear systems with iterative methods is called "regularization." A typical regularization technique modifies the ill-conditioned system with additional terms. See Hanke [14]. Regularization does not appear to be a good approach for solving (1.1) because (1.1) already has a well-defined solution (in particular, Theorem 1.1 implies that solutions are not highly sensitive to perturbation of the data vector $\mathbf{b}$). A regularization technique would compute a completely different solution.

4. MINRES-L for two layers. In this section and the next we consider the two-layered case, that is, $p = 2$ in (2.1). We consider the two-layered case separately from the p-layered case because the two-layered case contains all the main ideas of the general case but is easier to write down and analyze. In the $p = 1$ case, our algorithm reduces to MINRES applied to (1.2) and hence is not novel. MINRES is not even considered a good algorithm for (1.2) because it operates with the normal-equation matrix $A_1^T D_1 A_1$. See the remarks in section 9 for more comments on this matter. Furthermore, the $p = 2$ case is expected to occur commonly in practice. We mention also that the two-layered WLS and LLS problems were considered in Chapter 22 of Lawson and Hanson [18].

As noted in the preceding section, the two-layered WLS problem is written in the form (3.1), in which the diagonal entries of D_1, D_2 are of order 1 and $\delta_1 \geq \delta_2$. Let us introduce a new variable $\mathbf{v}$ such that

$$A_1^T D_1 A_1 \mathbf{v} = (\delta_1/\delta_2)(A_1^T D_1 A_1 \mathbf{x} - A_1^T D_1 \mathbf{b}_1). \tag{4.1}$$

Note that this equation always has a solution $\mathbf{v}$ because the right-hand side is in the

range of A_1^T. Multiplying (4.1) by δ_2 and adding to (3.1) yields

$$A_1^T D_1 A_1 \mathbf{v} = A_2^T D_2 \mathbf{b}_2 - A_2^T D_2 A_2 \mathbf{x}. \tag{4.2}$$

Putting (4.1) and (4.2) together, we get

$$\begin{pmatrix} A_2^T D_2 A_2 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & (-\delta_2/\delta_1) A_1^T D_1 A_1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \end{pmatrix}. \tag{4.3}$$

Our algorithm, which we call MINRES-L (for MINRES "layered"), is the application of the MINRES iteration due to Paige and Saunders [21] to (4.3). Note that (4.3) is a symmetric indefinite linear system.

In general, this linear system is rank-deficient because if $(\mathbf{x}; \mathbf{v})$ is a solution and $\mathbf{v}'$ satisfies $A_1 \mathbf{v}' = A_1 \mathbf{v}$, then $(\mathbf{x}; \mathbf{v}')$ is also a solution. Thus, (4.3) is rank-deficient whenever the rank of A_1 is less than n. This means we must address existence and uniqueness of a solution. Existence follows because the original WLS problem (3.1) is guaranteed to have a solution. Uniqueness of $\mathbf{x}$ is established as follows: if we add δ_2 times the first row of (4.3) to δ_1 times the second row, we recover the original WLS problem (3.1). Since (3.1) has a unique solution, (4.3) must uniquely determine $\mathbf{x}$. Since $\mathbf{x}$ is uniquely determined, so is $A_1 \mathbf{v}$.

The question arises whether MINRES (in exact arithmetic) will find a solution of (4.3). MINRES can find a solution only if it lies in the Krylov space, which (because of rank deficiency) is not necessarily full dimensional. This question was answered affirmatively by Theorem 2.4 of Brown and Walker [6]. (Their analysis concerns GMRES, but the same result applies to MINRES in exact arithmetic.) Furthermore, their result states that, assuming the initial guess is $\mathbf{0}$, the computed solution $(\mathbf{x}; \mathbf{v})$ will have minimum norm over all possible solutions. Since $\mathbf{x}$ is uniquely determined, their result implies that $\mathbf{v}$ will have minimum norm.

It may seem paradoxical that we remedy a difficulty caused by ill-conditioning by transforming the problem to a truly rank-deficient system. One explanation of this paradox concerns the limiting behavior as $\delta_1/\delta_2 \to \infty$. In this case, (3.1) tends to the linear system $A_1^T D_1 A_1 \mathbf{x} = A_1^T D_1 \mathbf{b}_1$. This system will, in general, not have a unique solution (because A_1 is not assumed to have rank n), and LSQR and CGNR will not have accurate information about $\mathbf{b}_2$ in their Krylov spaces. Thus, the LSQR and CGNR solutions are not expected to have the forward accuracy that we demand.

On the other hand, as $\delta_1/\delta_2 \to \infty$, we see that (4.3) tends to

$$\begin{pmatrix} A_2^T D_2 A_2 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \end{pmatrix}.$$

This system is easily seen to be the Lagrange multiplier conditions for the two-layered LLS problem: recall from section 2 that the two-layered LLS problem is

$$\begin{array}{rl} \text{minimize} & \|D_2^{1/2}(A_2 \mathbf{x} - \mathbf{b}_2)\|^2 \\ \text{subject to} & A_1^T D_1 A_1 \mathbf{x} = A_1^T D_1 \mathbf{b}_1. \end{array}$$

This is the correct limiting behavior: the WLS solution tends to the LLS solution as $\delta_2/\delta_1 \to 0$. An in-depth explanation of MINRES-L's convergence behavior follows.

5. Analysis of the error for two layers. In this section we consider convergence of MINRES-L in the presence of roundoff error for the case $p = 2$. As mentioned

in the introduction, we make a simplifying assumption concerning the effect of roundoff error in Krylov-space methods. Consider the symmetric linear system $M\mathbf{x} = \mathbf{c}$. If M is singular, we then assume that $\mathbf{c}$ lies in its range-space. For the purpose of this analysis, we assume that there is no preconditioner, and the initial guess is $\mathbf{x}^{(0)} = \mathbf{0}$. These assumptions could be relaxed, although some conditions would need to be imposed on $\mathbf{x}^{(0)}$ and on the preconditioner to make this analysis work. Although our analysis needs these restrictions, there is no restriction imposed by the algorithm itself on the choice of $\mathbf{x}^{(0)}$. The only restriction imposed by the algorithm on the preconditioner is the same restriction that applies to MINRES generally, namely, that the preconditioner must be positive definite. (If the system matrix M is singular, then the preconditioner P may be positive semidefinite provided that its nullspace is contained in the nullspace of M.)

Our assumption about the effect of roundoff is that after a sufficient number of iterations, all of these Krylov methods will compute an iterate $\hat{\mathbf{x}}$ satisfying

$$\|\mathbf{c} - M\hat{\mathbf{x}}\| \leq C\epsilon \cdot \|M\| \cdot \|\mathbf{x}\|, \tag{5.1}$$

where C is a modest constant, ϵ is machine epsilon, and $\mathbf{x}$ is the true solution. (If multiple solutions exist, we take $\mathbf{x}$ to be the minimum-norm solution.) In other words, we assume that in all of these methods, the true residual is ultimately driven to machine epsilon in the relative sense.

As far as we know, this bound has not been rigorously proved for CG or LSQR, but it is related to a bound proved by Greenbaum [12]. In particular, Greenbaum's result implies that (5.1) would hold for CG if we were guaranteed that the recursively updated residual drops to well below machine precision, which always happens in our test cases.

A bound similar to (5.1) is known to hold for GMRES implemented with Householder transformations as shown by Drkošová et al. [8]. Little is known about floating point behavior of MINRES. GMRES is equivalent to MINRES augmented with a full reorthogonalization process. For this paper, we tentatively assert (5.1) for MINRES without proof. Our computational experiments in section 7 provide a bit of evidence to support the assertion, but a more thorough computational study on whether (5.1) applies to MINRES would be very useful.

This bound sheds light on why MINRES-L can attain much better accuracy than CGNR. For CGNR, the error bound (5.1) implies that $\|A^T D\mathbf{b} - A^T DA\hat{\mathbf{x}}\|$ gets very small, where $\hat{\mathbf{x}}$ is the computed solution. This latter quantity is the same as $\|(A^T DA)(\mathbf{x} - \hat{\mathbf{x}})\|$. But recall that we are seeking a bound on the forward error, that is, on $\|\mathbf{x} - \hat{\mathbf{x}}\|$. In this case, the factor $(A^T DA)$ can greatly skew the norm when δ_2/δ_1 is close to zero, so there is no bound on $\|\mathbf{x} - \hat{\mathbf{x}}\|$ independent of δ_1/δ_2, that is, (1.5) is not expected to be satisfied by CGNR. This is confirmed by our computational experiments.

It is not known what bound to expect on forward error for LSQR. Our experiments hint that LSQR satisfies (5.1) for the normal equations and may satisfy a stronger bound.

In contrast, an analysis of MINRES-L starting from (5.1) does yield the accuracy bound (1.5). We need the following preliminary lemma.

LEMMA 5.1. *Let A be an $m \times n$ matrix of rank n and $\bar{A}$ an $r \times n$ submatrix of A. Suppose the linear system $\bar{A}^T \bar{D} \bar{A}\mathbf{x} = A^T\mathbf{c}$ is consistent. Here, $\mathbf{c}$ is a given vector, and $\bar{D}$ is a given diagonal positive definite matrix. Then for any solution $\mathbf{x}$,*

$$\|\bar{A}\mathbf{x}\| \leq \|\bar{D}^{-1}\| \cdot \bar{\chi}_A \cdot \|\mathbf{c}\| \tag{5.2}$$

and

$$\|\bar{A}\mathbf{x}\| \le \|\bar{D}^{-1}\| \cdot \chi_A \cdot \|A^T\mathbf{c}\|. \tag{5.3}$$

Furthermore, there exists a solution $\mathbf{x}$ *satisfying*

$$\|\mathbf{x}\| \le \|\bar{D}^{-1}\| \cdot \chi_A \bar{\chi}_A \cdot \|\mathbf{c}\|. \tag{5.4}$$

Proof. First, note the following preliminary result. Let H, K be two symmetric positive semidefinite $n \times n$ matrices such that $H + K$ is positive definite. Let $\mathbf{b}$ be an n-vector in the range-space of H. Then $(H + \epsilon K)^{-1}\mathbf{b}$ converges to a solution of $H\mathbf{x} = \mathbf{b}$ as $\epsilon \to 0^+$. This is proved by reducing to the diagonal case using simultaneous diagonalization of H, K.

Let D be the extension of $\bar{D}$ to an $m \times m$ diagonal matrix obtained by padding with zeros, so that $A^T D A = \bar{A}^T \bar{D} \bar{A}$. Let M be an $m \times m$ diagonal matrix with 1's in diagonal positions corresponding to $\bar{D}$ and zeros elsewhere. Let N be the complementary projection, i.e., N is also a diagonal matrix such that $M + N = I$. Since $A^T D A\mathbf{x} = A^T\mathbf{c}$ is consistent, the limit of $(A^T(D + \epsilon N)A)^{-1}A^T\mathbf{c}$ as $\epsilon \to 0^+$ is some solution $\mathbf{x}$ of $\bar{A}^T\bar{D}\bar{A}\mathbf{x} = A^T\mathbf{c}$, as noted in the preceding paragraph. We have

$$\begin{aligned}
\|\bar{A}\mathbf{x}\| &= \|MA\mathbf{x}\| \\
&= \lim_{\epsilon\to 0^+} \|MA(A^T(D+\epsilon N)A)^{-1}A^T\mathbf{c}\| \\
&= \lim_{\epsilon\to 0^+} \|M(D+\epsilon N)^{-1}(D+\epsilon N)A(A^T(D+\epsilon N)A)^{-1}A^T\mathbf{c}\| \qquad (5.5)\\
&\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon N)^{-1}\| \cdot \sup_{\epsilon>0} \|(D+\epsilon N)A(A^T(D+\epsilon N)A)^{-1}A^T\| \cdot \|\mathbf{c}\| \\
&\le \|\bar{D}^{-1}\| \cdot \bar{\chi}_A \cdot \|\mathbf{c}\|.
\end{aligned}$$

The last line was obtained from the transpose of (1.4). This proves (5.2). Note that this holds for all $\mathbf{x}$ satisfying $\bar{A}^T\bar{D}\bar{A}\mathbf{x} = A^T\mathbf{c}$, since this latter equation uniquely determines $\bar{A}\mathbf{x}$. Similarly, to demonstrate (5.3), we start from (5.5):

$$\begin{aligned}
\|\bar{A}\mathbf{x}\| &\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon N)^{-1}(D+\epsilon N)A(A^T(D+\epsilon N)A)^{-1}A^T\mathbf{c}\| \\
&\le \lim_{\epsilon\to 0^+} \|M(D+\epsilon N)^{-1}\| \cdot \sup_{\epsilon>0} \|(D+\epsilon N)A(A^T(D+\epsilon N)A)^{-1}\| \cdot \|A^T\mathbf{c}\| \\
&\le \|\bar{D}^{-1}\| \cdot \chi_A \cdot \|A^T\mathbf{c}\|.
\end{aligned}$$

Turning to the proof of (5.4), observe that $A^T\mathbf{c} = \bar{A}^T\bar{D}\bar{A}\mathbf{x} = A^T D A\mathbf{x} = A^T D M A\mathbf{x} = A^T(D + \epsilon N)MA\mathbf{x}$ for any ϵ, since $MN = 0$. Hence,

$$\begin{aligned}
\mathbf{x} &= \lim_{\epsilon\to 0^+} (A^T(D+\epsilon N)A)^{-1}A^T\mathbf{c} \\
&= \lim_{\epsilon\to 0^+} (A^T(D+\epsilon N)A)^{-1}A^T(D+\epsilon N)MA\mathbf{x}
\end{aligned}$$

and thus

$$\begin{aligned}
\|\mathbf{x}\| &\le \sup_{\epsilon>0} \|(A^T(D+\epsilon N)A)^{-1}A(D+\epsilon N)\| \cdot \|MA\mathbf{x}\| \\
&\le \chi_A \|\bar{A}\mathbf{x}\|.
\end{aligned}$$

Combining this with (5.2) proves (5.4). ∎

To resume the analysis of MINRES-L, we define residual vectors

$$\mathbf{r}_1 = A_2^T D_2 A_2 \hat{\mathbf{x}} + A_1^T D_1 A_1 \hat{\mathbf{v}} - A_2^T D_2 \mathbf{b}_2 \text{ and} \tag{5.6}$$

$$\mathbf{r}_2 = A_1^T D_1 A_1 \hat{\mathbf{x}} - (\delta_2/\delta_1) A_1^T D_1 A_1 \hat{\mathbf{v}} - A_1^T D_1 \mathbf{b}_1, \tag{5.7}$$

where $(\hat{\mathbf{x}}; \hat{\mathbf{v}})$ is the solution of (4.3) computed by MINRES-L. Then (5.1) applied to (4.3) yields the bounds

$$\|\mathbf{r}_1\|, \|\mathbf{r}_2\| \leq C\epsilon \cdot \|H_2\| \cdot \|(\mathbf{x}; \mathbf{v})\|. \tag{5.8}$$

In this formula, H_2 is shorthand for the coefficient matrix of (4.3).

We can extract another equation from (5.6) and (5.7); in particular, if we multiply (5.6) by δ_2, multiply (5.7) by δ_1 and then add, we eliminate the terms involving $\hat{\mathbf{v}}$:

$$\delta_2 \mathbf{r}_1 + \delta_1 \mathbf{r}_2 = \delta_1 A_1^T D_1 A_1 \hat{\mathbf{x}} + \delta_2 A_2^T D_2 A_2 \hat{\mathbf{x}} - \delta_1 A_1^T D_1 \mathbf{b}_1 - \delta_2 A_2^T D_2 \mathbf{b}_2.$$

Let $\mathbf{x}$ be the exact solution to the WLS problem. The last two terms of this equation can be replaced with terms involving $\mathbf{x}$ by using (3.1). Interchanging the left- and right-hand sides yields

$$\delta_1 A_1^T D_1 A_1 (\hat{\mathbf{x}} - \mathbf{x}) + \delta_2 A_2^T D_2 A_2 (\hat{\mathbf{x}} - \mathbf{x}) = \delta_2 \mathbf{r}_1 + \delta_1 \mathbf{r}_2. \tag{5.9}$$

The goal is to derive an accuracy bound like (1.5) from (5.8) and (5.9). We start by bounding the quantity in the right-hand side of (5.8). Note that $\|H_2\|$ can be bounded by $2\kappa\|A\|^2$ because the largest entries in D_1, D_2 are bounded by κ. We can bound $\|\mathbf{x}\|$ by $\chi_A\|\mathbf{b}\|$ and $\|A\mathbf{x}\|$ by $\bar{\chi}_A\|\mathbf{b}\|$ using Theorem 1.1. Next we turn to bounding $\|\mathbf{v}\|$ in (5.8). Recall that, as mentioned in the preceding section, $\mathbf{v}$ is not uniquely determined, but MINRES will find the minimum-norm $\mathbf{v}$ satisfying (4.3). Recall that $\mathbf{v}$ is determined by the constraint

$$A_1^T D_1 A_1 \mathbf{v} = A_2^T D_2 \mathbf{b}_2 - A_2^T D_2 A_2 \mathbf{x}.$$

One way to pick such a $\mathbf{v}$ is to use Lemma 5.1 with $\bar{A}$ chosen to be A_1 and $\mathbf{c}$ chosen to be $[\mathbf{0}; D_2\mathbf{b}_2 - D_2 A_2 \mathbf{x}]$. In this case,

$$\begin{aligned} \|\mathbf{c}\| &\leq \kappa\|\mathbf{b}\| + \kappa\|A\mathbf{x}\| \\ &\leq \kappa(\bar{\chi}_A + 1)\|\mathbf{b}\|. \end{aligned}$$

Thus, by (5.4), we can select $\mathbf{v}$ so that

$$\|\mathbf{v}\| \leq \kappa\chi_A\bar{\chi}_A(\bar{\chi}_A + 1)\|\mathbf{b}\|.$$

Note that the derivation of this inequality used the fact that $\|D_1^{-1}\| \leq 1$, which follows from the assumption that diagonal entries of each D_i are 1 or greater. Combining the $\mathbf{x}$ and $\mathbf{v}$ contributions means that we have bounded the right-hand side of (5.8); let us rewrite (5.8) with the new bound:

$$\|\mathbf{r}_1\|, \|\mathbf{r}_2\| \leq 2C\epsilon \cdot \|A\|^2 \cdot \kappa^2 \cdot \chi_A\bar{\chi}_A(\bar{\chi}_A + 2)\|\mathbf{b}\|. \tag{5.10}$$

Next, we write new equations for $\mathbf{r}_1, \mathbf{r}_2$. Observe that $\mathbf{r}_1$ lies in the range of $[A_1^T, A_2^T]$, so we can find $\mathbf{h}_1$ satisfying

$$\mathbf{r}_1 = A_1^T D_1 A_1 \mathbf{h}_1 + A_2^T D_2 A_2 \mathbf{h}_1. \tag{5.11}$$

Similarly, by (5.7) there exists $\mathbf{h}_2$ satisfying

$$\mathbf{r}_2 = A_1^T D_1 A_1 \mathbf{h}_2. \tag{5.12}$$

By applying (5.3) to $\mathbf{r}_1$ and $\mathbf{r}_2$ separately, with "$A^T\mathbf{c}$" in the lemma taken to be first $\mathbf{r}_1$ and then $\mathbf{r}_2$, we conclude from (5.11) and (5.12) that

$$\|[A_1; A_2]\mathbf{h}_1\| \le \chi_A \cdot \|\mathbf{r}_1\| \text{ and} \tag{5.13}$$

$$\|A_1\mathbf{h}_2\| \le \chi_A \cdot \|\mathbf{r}_2\|. \tag{5.14}$$

Substituting (5.11) and (5.12) into (5.9) yields

$$\begin{aligned}\delta_1 A_1^T D_1 A_1(\hat{\mathbf{x}} - \mathbf{x}) + \delta_2 A_2^T D_2 A_2(\hat{\mathbf{x}} - \mathbf{x}) &= \delta_1 A_1^T D_1 A_1 \mathbf{h}_2 + \delta_2 A_1^T D_1 A_1 \mathbf{h}_1 \\ &\quad + \delta_2 A_2^T D_2 A_2 \mathbf{h}_1 \\ &= \delta_1 A_1^T D_1 (A_1\mathbf{h}_2 + (\delta_2/\delta_1) A_1 \mathbf{h}_1) \\ &\quad + \delta_2 A_2^T D_2 A_2 \mathbf{h}_1.\end{aligned}$$

Notice (by analogy with (3.1)) that the preceding equation is exactly a WLS computation where the "unknown" is $\hat{\mathbf{x}} - \mathbf{x}$ and the right-hand side data is $(A_1\mathbf{h}_2 + (\delta_2/\delta_1)A_1\mathbf{h}_1; A_2\mathbf{h}_1)$. Thus, by Theorem 1.1,

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \le \chi_A \|(A_1\mathbf{h}_2 + (\delta_2/\delta_1)A_1\mathbf{h}_1; A_2\mathbf{h}_1)\|.$$

We now build a chain of inequalities: the right-hand side of the preceding inequality is bounded by (5.13) and (5.14), and the right-hand side of (5.13) and (5.14) is bounded by (5.10). Combining all of this yields

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \le 4C\epsilon \cdot \chi_A^3 \|A\|^2 \cdot \kappa^2 \cdot \bar{\chi}_A(\bar{\chi}_A + 2) \cdot \|\mathbf{b}\|. \tag{5.15}$$

To obtain the preceding inequality, we used the assumption that $\delta_2/\delta_1 \le 1$. Thus, we have an error bound of the form (1.5) as desired; in particular, there is no dependence of the error bound on δ_2/δ_1. Note that this bound depends on κ. Recall that κ is defined to be the maximum entry in $D_1, \ldots, D_p$ and is assumed to be small. Indeed, as noted in section 2, we can always assume that $\kappa = 1$ if we are willing to divide the problem into many layers.

Also note that this bound depends on A because it includes factors like $\|A\|$ and χ_A, $\bar{\chi}_A$. The reader may wonder why the bound does not depend explicitly on the partitioning of A into $A_1, A_2, \ldots$. This is because the parameters $\chi_A, \bar{\chi}_A$ implicitly involve a supremum over all possible such partitions. (See, e.g., [26] and [20] for some theorems along these lines.) In particular, the parameter χ_A grows inversely with the smallest nonzero singular value of any square submatrix of A. Thus, it is expected that MINRES-L will perform poorly if A_1 is ill-conditioned (in the sense that its smallest nonzero singular value is close to zero). See further remarks in section 9. Note that the algorithm does not need to know the values of either χ_A or $\bar{\chi}_A$. In our test cases below, some examples have matrices A for which χ_A and $\bar{\chi}_A$ are known to be small. In others we do not have estimates of these parameters.

6. MINRES-L for *p* layers. In this section we present the MINRES-L algorithm for the p-layered WLS problem. In particular, we describe a layered linear system whose solution corresponds to the solution of the WLS problem. The proof that this algorithm is stable in the sense of (1.5) is omitted but is available in the

technical report version [5] of this paper. It is a generalization of the proof of (5.15) in the previous section. The algorithm is the application of MINRES to the symmetric linear system $H_p\mathbf{w} = \mathbf{c}_p$, where H_p is a square matrix of size $qn \times qn$, where $q = (1 + p(p-1)/2)$, $\mathbf{c}_p$ is a vector of order qn, and $\mathbf{w}$ is the vector of unknowns. Matrix H_p is partitioned into $q \times q$ blocks each of size $n \times n$. Vectors $\mathbf{c}_p$ and $\mathbf{w}$ are conformally partitioned. The WLS solution vector is the first subvector of $\mathbf{w}$.

In more detail, the vector $\mathbf{w}$ is composed of $\mathbf{x}$ concatenated with $p(p-1)/2$ n-vectors that we denote $\mathbf{v}_{i,j}$, where i lies in $2, \ldots, p$ and j lies in $1, \ldots, i-1$. Recall that the p-layered WLS problem may be written as

$$\delta_1 A_1^T D_1 A_1 \mathbf{x} + \cdots + \delta_p A_p^T D_p A_p \mathbf{x} = \delta_1 A_1^T D_1 \mathbf{b}_1 + \cdots + \delta_p A_p^T D_p \mathbf{b}_p. \tag{6.1}$$

Let $\mathbf{x}$ be the solution to this equation. Then we see from this equation that $A_p^T D_p A_p \mathbf{x} - A_p^T D_p \mathbf{b}_p$ lies in the span of $[A_1^T, \ldots, A_{p-1}^T]$. Therefore, there exists a solution $\bar{\mathbf{v}}$ to

$$A_p^T D_p A_p \mathbf{x} + (\delta_{p-1}/\delta_p) A_{p-1}^T D_{p-1} A_{p-1} \bar{\mathbf{v}} + \cdots + (\delta_1/\delta_p) A_1^T D_1 A_1 \bar{\mathbf{v}} = A_p^T D_p \mathbf{b}. \tag{6.2}$$

Therefore, there exists a solution $[\mathbf{v}_{p,p-1}; \ldots; \mathbf{v}_{p,1}]$ to the equation

$$A_p^T D_p A_p \mathbf{x} + A_{p-1}^T D_{p-1} A_{p-1} \mathbf{v}_{p,p-1} + \cdots + A_1^T D_1 A_1 \mathbf{v}_{p,1} = A_p^T D_p \mathbf{b} \tag{6.3}$$

which can be obtained from (6.2) by defining

$$\mathbf{v}_{p,j} = (\delta_j/\delta_p)\bar{\mathbf{v}}. \tag{6.4}$$

Equation (6.3) is the first block-row of $H_p\mathbf{w} = \mathbf{c}_p$. (Note that MINRES-L would likely not compute the particular solution to (6.3) given by (6.4) since the norm of $\mathbf{v}_{p,j}$ in (6.4) is large when $\delta_j \gg \delta_p$. But nonetheless, this definition of $\mathbf{v}_{p,j}$ is sufficient to prove that the linear system constructed in this section is consistent.) In other words, the first block-row of H_p contains one copy of each of the matrices $A_i^T D_i A_i$, and the first block of $\mathbf{c}_p$ is $A_p^T D_p \mathbf{b}$.

The next $p-1$ block-rows continue this pattern. Specifically, the $(p-i+1)$th block-row of $H_p\mathbf{w} = \mathbf{c}_p$, for $i = 1, \ldots, p$, is the equation

$$A_i^T D_i A_i \mathbf{x} + \sum_{j=1}^{i-1} A_j^T D_j A_j \mathbf{v}_{i,j} - \sum_{j=i+1}^{p} \frac{\delta_j}{\delta_i} A_i^T D_i A_i \mathbf{v}_{j,i} = A_i^T D_i \mathbf{b}_i. \tag{6.5}$$

This completes the description of block-rows $1, \ldots, p$ of $H_p\mathbf{w} = \mathbf{c}_p$. We now establish some properties of these block-rows, and we postpone the description of block-rows $p+1, \ldots, q$.

LEMMA 6.1. *Suppose* $\mathbf{w}$ *is a solution to the linear equation* (6.5) *for each* $i = 1, \ldots, p$, *where* $\mathbf{w}$ *denotes the concatenation of* $\mathbf{x}$ *and all of the* $\mathbf{v}_{i,j}$*'s. Then* $\mathbf{x}$ *is the solution to the WLS problem* (6.1).

Proof. For each i, multiply (6.5) by δ_i and then sum all p equations obtained in this manner. Observe that all the $\mathbf{v}_{i,j}$ terms cancel out and we end up exactly with (6.1). ∎

We also need the converse to be true.

LEMMA 6.2. *Suppose* $\mathbf{x}$ *is the solution to* (6.1). *Then there exist vectors* $\mathbf{v}_{i,j}$ *for* $1 \le j < i \le p$ *such that* (6.5) *is satisfied for each* $i = 1, \ldots, p$.

Proof. The proof is by induction on (decreasing) $k = p, \ldots, 1$. We assume that we have already determined $\mathbf{v}_{i,j}$ for all $i = k+1, \ldots, p$ and all $j = 1, \ldots, i-1$

so that (6.5) is satisfied for $i = k+1, \ldots, p$, and now we must determine $\mathbf{v}_{k,j}$ for $j = 1, \ldots, i-1$ to satisfy (6.5) for the particular value $i = k$. The base case of the induction is that we can select $\mathbf{v}_{p,1}, \ldots, \mathbf{v}_{p,p-1}$ to satisfy (6.5) in the case $i = p$ as argued above.

Now for the induction case of $k < p$. Rewrite (6.5) for the case $k = i$, and multiply through by δ_k:

$$\delta_k A_k^T D_k A_k \mathbf{x} + \delta_k \sum_{j=1}^{k-1} A_j^T D_j A_j \mathbf{v}_{k,j} - \sum_{j=k+1}^{p} \delta_j A_k^T D_k A_k \mathbf{v}_{j,k} = \delta_k A_k^T D_k \mathbf{b}_k. \tag{6.6}$$

Recall that our goal is to choose $\mathbf{v}_{k,j}$ for $j = 1, \ldots, k-1$ to make this equation valid.

Multiply (6.5) for each $i = k+1, \ldots, p$ by δ_i and add this to (6.6). After rearranging the summations and cancelling common terms on the left-hand side, we end up with

$$\sum_{i=k}^{p} \delta_i A_i^T D_i A_i \mathbf{x} + \sum_{i=k}^{p} \sum_{j=1}^{k-1} \delta_i A_j^T D_j A_j \mathbf{v}_{i,j} = \sum_{i=k}^{p} \delta_i A_i^T D_i \mathbf{b}_i. \tag{6.7}$$

Dividing through by δ_k and separating out the $\mathbf{v}_{k,j}$ terms from the second summation yields

$$\begin{aligned} A_1^T D_1 A_1 \mathbf{v}_{k,1} + \cdots + A_{k-1}^T D_{k-1} A_{k-1} \mathbf{v}_{k,k-1} \\ = \sum_{i=k}^{p} \frac{\delta_i}{\delta_k} A_i^T D_i (\mathbf{b}_i - A_i \mathbf{x}) - \sum_{i=k+1}^{p} \sum_{j=1}^{k-1} \frac{\delta_i}{\delta_k} A_j^T D_j A_j \mathbf{v}_{i,j}. \end{aligned} \tag{6.8}$$

But from (6.1) we know that $\sum_{i=k}^{p} \delta_i A_i^T D_i (\mathbf{b}_i - A_i \mathbf{x})$ lies in the range of $[A_1^T, \ldots, A_{k-1}^T]$. Clearly the rightmost summation of (6.8) also lies in the same range. Therefore, there exist $\mathbf{v}_{k,j}$ for $j = 1, \ldots, k-1$ to make (6.8) valid. But then these same choices will make (6.6) valid because the algebraic steps used to derive (6.8) from (6.6) can be reversed. This proves the lemma. ∎

We now explain the remaining $q - p = (p-1)(p-2)/2$ block-rows of H_p. These rows exist solely for the purpose of making H_p symmetric. First, we have to order the variables and equations correctly. The variables will be listed in the order $(\mathbf{x}; \mathbf{v}_{p,p-1}; \mathbf{v}_{p,p-2}; \ldots; \mathbf{v}_{p,1}; \mathbf{v}_{p-1,p-2}; \ldots; \mathbf{v}_{p-1,1}; \ldots; \mathbf{v}_{2,1})$. The first p block equations will be listed in the order (6.5) for $i = p, p-1, \ldots, 1$. This means that the first p block-rows of H_p have the format $[S_p, T_p]$, where S_p is a matrix of $p \times p$ blocks, each block $n \times n$, and T_p is a $p \times (p-1)(p-2)/2$ block matrix. Furthermore, it is easily checked that S_p is symmetric: its first block-row and first block-column both consist of $A_i^T D_i A_i$ listed in the order $i = p, \ldots, 1$; the $(p-i+1)$st entry of its main diagonal is $-(\delta_p/\delta_i) A_i^T D_i A_i$ for $i = 1, \ldots, p-1$; and all its other blocks are zeros. Then we define H_p to be

$$H_p = \begin{pmatrix} S_p & T_p \\ T_p^T & 0 \end{pmatrix}.$$

We define $\mathbf{c}_p$ as

$$\mathbf{c}_p = \begin{pmatrix} A_p^T D_p \mathbf{b}_p \\ \vdots \\ A_1^T D_1 \mathbf{b}_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix},$$

where there are $p(p-1)/2$ blocks of zeros. For example, the following linear system is $H_3\mathbf{w} = \mathbf{c}_3$:

$$\begin{pmatrix} A_3^T D_3 A_3 & A_2^T D_2 A_2 & A_1^T D_1 A_1 & 0 \\ A_2^T D_2 A_2 & -\frac{\delta_3}{\delta_2} A_2^T D_2 A_2 & 0 & A_1^T D_1 A_1 \\ A_1^T D_1 A_1 & 0 & -\frac{\delta_3}{\delta_1} A_1^T D_1 A_1 & -\frac{\delta_2}{\delta_1} A_1^T D_1 A_1 \\ 0 & A_1^T D_1 A_1 & -\frac{\delta_2}{\delta_1} A_1^T D_1 A_1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{v}_{3,2} \\ \mathbf{v}_{3,1} \\ \mathbf{v}_{2,1} \end{pmatrix} = \begin{pmatrix} A_3^T D_3 \mathbf{b}_3 \\ A_2^T D_2 \mathbf{b}_2 \\ A_1^T D_1 \mathbf{b}_1 \\ \mathbf{0} \end{pmatrix}. \tag{6.9}$$

We must now consider whether $H_p\mathbf{w} = \mathbf{c}_p$ has any solutions; in particular, we must demonstrate that the new group of equations $T_p^T \mathbf{w}' = \mathbf{0}$ is consistent with the first p block-rows. Here $\mathbf{w}'$ denotes the first p blocks of $\mathbf{w}$, that is, $\mathbf{w}' = (\mathbf{x}; \mathbf{v}_{p,p-1}; \ldots; \mathbf{v}_{p,1})$. Studying the structure of T_p, we see that there are $(p-1)(p-2)/2$ block-rows of T_p^T indexed by (i,j) for $1 \le j < i \le p-1$ (in correspondence with the columns of T_p, which correspond to variables $\mathbf{v}_{i,j}$ for i,j in that range). The row indexed by (i,j) has exactly two nonzero block entries that yield the equation

$$A_j^T D_j A_j \mathbf{v}_{p,i} - \frac{\delta_i}{\delta_j} A_j^T D_j A_j \mathbf{v}_{p,j} = \mathbf{0}. \tag{6.10}$$

Our task is therefore to show that we can simultaneously satisfy (6.5) for $i = 1, \ldots, p$ and (6.10) for (i,j) such that $1 \le j < i \le p-1$. But this follows immediately from (6.4), which shows that there is at least one way to pick $\mathbf{v}_{p,p-1}, \ldots, \mathbf{v}_{p,1}$ to satisfy (6.10).

This proof shows that the above method for selecting $\mathbf{v}_{p,1}, \ldots, \mathbf{v}_{p,p-1}$ is consistent and satisfies (6.10). We also see that (6.3) is satisfied. Thus, the arguments of this section have established the following theorem.

THEOREM 6.3. *There exists at least one solution* $\mathbf{w}$ *to* $H_p\mathbf{w} = \mathbf{c}_p$*, and furthermore, any such solution has as its first* n *entries the vector* $\mathbf{x}$ *that solves* (6.1).

As mentioned at the beginning of this section, we omit the analysis of the error of this method for the case of p layers. That analysis is available in [5]. Here we present only the final bound. Assuming that $\hat{\mathbf{x}}$ computed by MINRES-L satisfies (5.1), and letting $\mathbf{x}$ denote the exact solution to (6.1),

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \le C\epsilon p^6 \|A\|^2 \cdot \kappa(\bar{\chi}_A + 1)\chi_A^3 \cdot (4\kappa\bar{\chi}_A)^{p-1} \|\mathbf{b}\|. \tag{6.11}$$

This is a bound of the form (1.5) as desired. Note that the exponential growth of the right-hand side of (6.11) with respect to p is another reason why MINRES-L may not be suitable for problems where p is large. Therefore, all of our test cases assume two layers. We have also done a few experiments in the $p = 3$ case not reported here.

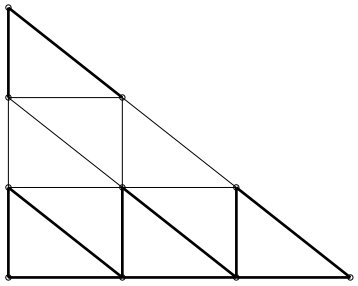


FIG. 7.1. *An* 18×9 *RNAI matrix based on this graph was used for the first group of tests. The column corresponding to the top node is deleted. Edges marked with heavy lines are weighted* 1*, and edges marked with light lines are weighted* δ_2*, where* δ_2 *varies from test to test.*

7. Computational experiments. In this section we present computational experiments on MINRES-L and LSQR to compare their achievable accuracy and efficiency. Some experiments also involve CG, CGNR, and GMRES. Three main experiments are presented. First is an idealized test on a small node-arc adjacency matrix that illustrates the behavior of MINRES-L under the most favorable conditions. Second is a slightly larger test case in which the theoretical bound still applies, but in which a deficiency of MINRES-L appears, namely, the loss of orthogonality among the Krylov residuals. Third is a large test-case in which MINRES-L eventually converges according to predictions of the theory, but only when a preconditioner is used, and even in that case, only after billions of flops.

The first test involves a small node-arc adjacency matrix. This test was conducted in Matlab 5.2. Matlab is a software package and programming language for numerical computation written by The Mathworks, Inc. All computations are in IEEE double precision with machine-epsilon approximately $2.2 \cdot 10^{-16}$. Matlab sparse matrix operations were used in all tests.

In this test we compare LSQR on (1.1) to MINRES-L (i.e., to MINRES applied to (4.3)). We also carried out this experiment with CGNR, based on CGLS1 as in (3.2) of Björck, Elfving, and Strakoš [3]. These authors conclude that CGLS1 is a good way to organize CGNR. Its behavior was almost identical to LSQR, so the results are not shown.

Our implementation of MINRES is based on [21], except Givens rotations were used instead of 2×2 Householder matrices (so that there are some inconsequential sign differences). The MINRES-L iteration terminates when the scaled computed residual $\|\mathbf{r}_k\|/\|[A_1^T D_1 \mathbf{b}_1; \ldots ; A_p^T D_p \mathbf{b}_p]\|$ drops below 10^{-13}. The LSQR routine uses the compound termination test described in [22].

The matrix A used in the following tests is the reduced node-arc adjacency matrix of the graph depicted in Figure 7.1. A "node-arc adjacency" matrix contains one column for each node of a graph and one row for each edge. Each row contains exactly two nonzero entries, a "+1" and a "−1" in the columns corresponding to the end-points of the edge. (The choice of which end-point is assigned +1 and which is assigned −1 induces an orientation on the edge, but often this orientation is irrelevant for the application.) A reduced node-arc incidence (RNAI) matrix is obtained from a node-arc incidence matrix by deleting one column. RNAI matrices arise in the analysis of an electrical network with batteries and resistors; see [29]. They also arise in network flow problems. In the case of Figure 7.1, the column corresponding to the

TABLE 7.1

Behavior of the two-layered MINRES-L algorithm compared to LSQR for decreasing values of δ_2. The error reported is the scaled error defined in the text. Note that the LSQR accuracy degrades while the MINRES-L accuracy stays about the same.

δ_2	MINRES-L Flops	MINRES-L Iterations	MINRES-L Error	LSQR Flops	LSQR Iterations	LSQR Error
10^{-3}	19443	30	1.4e-15	5608	12	3.2e-15
10^{-6}	17508	27	2.8e-15	6053	13	1.1e-11
10^{-9}	19443	30	1.2e-15	6053	13	3.5e-8
10^{-12}	18798	29	2.7e-15	6053	13	8.1e-6
10^{-15}	18153	28	1.5e-15	2938	6	8.2e-1
10^{-18}	18153	28	1.9e-15	2938	6	8.2e-1

top node was deleted. Thus, A is an 18×9 matrix. It is well known that the RNAI matrix for a connected graph always has full rank. RNAI matrices are known to have small values of χ_A and $\bar{\chi}_A$ [29].

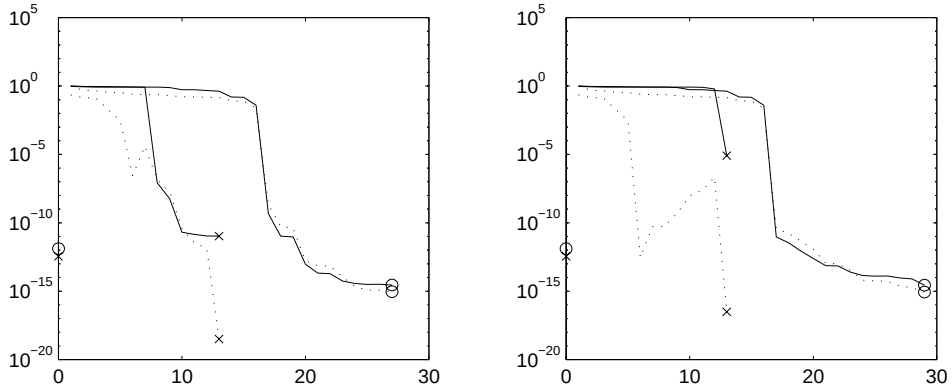
In all these tests, the weight matrix has two layers. We took $D_1 = I$, $D_2 = I$, and $\delta_1 = 1$, while we let δ_2 vary from experiment to experiment. The rows of A in correspondence with D_2 are drawn as thinner lines in Figure 7.1. Finally, the right-hand side $\mathbf{b}$ was chosen to be the first 18 prime numbers.

The results are displayed in Table 7.1, and the cases when $\delta_2 = 10^{-6}$ and $\delta_2 = 10^{-12}$ are plotted in Figure 7.2. The *scaled error* that is tabulated and plotted in all cases is defined to be $\|\hat{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{b}\|$. We choose this particular scaling for the error because our goal is to investigate stability bound (1.5). The true solution $\mathbf{x}$ is computed using the COD method [17]. Note that the accuracy of LSQR decays as δ_2 gets smaller, whereas MINRES-L's accuracy stays constant. MINRES-L requires many more flops than LSQR because the system matrix is larger and the number of iterations greater. The running time of LSQR is about the same for the first four rows of the table as the ill-conditioning increases. In the last two rows the running time of LSQR drops because the matrix $A^T D A$ masquerades as a low-rank matrix for small values of δ_2, causing early termination of the Lanczos process.

Besides returning an inaccurate solution, LSQR has the additional difficulty that its residual (the quantity normally measured in practical use of this algorithm) does not reflect the forward error, so there is no simple way to determine whether LSQR is computing good answers. In contrast, the error and residual in MINRES-L are closely correlated as indicated in Figure 7.2. This correlation is predicted by our theory.

We found that the results of the preceding experiment were fairly insensitive to termination tests. In particular, we found that iterating beyond the step when the residual is driven to a small number (in either MINRES-L or LSQR) does not appear to lead to further reduction in the error compared to the results presented here.

The next computational test involved a slightly larger matrix A taken from the Netlib linear programming test set, namely, the matrix in problem ADLITTLE, which is 138×56. The matrix D was defined to have 15 initial entries of size 10^{-13} followed by 56 entries of size 1 followed by 67 entries again of size 10^{-13}. (This pattern in D yielded more interesting results than simpler patterns.) The right-hand side vector $\mathbf{b}$ was chosen to contain the first 138 primes. MINRES-L required 1169 iterations and 5.6 Mflops and yielded a solution $\hat{\mathbf{x}}$ with scaled error $1.9 \cdot 10^{-11}$ with respect to the true solution computed by the COD method. For this matrix, χ_A and $\bar{\chi}_A$ are not known. LSQR on this problem required 135 iterations and 0.49 Mflops and returned an answer with scaled error 0.25. Interestingly, we found in this experiment that continuing to



Legend:

—× = LSQR scaled error
· · · × = LSQR scaled residual
—○ = MINRES-L scaled error
· · · ○ = MINRES-L scaled residual

FIG. 7.2. *Convergence behavior of LSQR and MINRES-L for the* 18×9 *RNAI test case. The plots are for* $\delta_2 = 10^{-6}$ *(left) and* $\delta_2 = 10^{-12}$ *(right). In these plots and all that follow, the* x*-axis is the iteration number. For both algorithms the computed (i.e., recursively updated) residual is plotted rather than the true residual. Other experiments (not reported here) indicate that these are usually indistinguishable.*

iterate with LSQR even after the termination test is satisfied increases the accuracy substantially. After approximately 300 more iterations, the error of LSQR dropped to $4.6 \cdot 10^{-6}$ and then appeared to stay fixed at that level. It is not clear how one could take advantage of this increased accuracy in practice since there is no obvious way to detect the reduction in forward error.

The convergence plots are depicted in Figure 7.3. This plot shows the behavior of LSQR extended beyond iteration 135 when the termination test was satisfied. In contrast to other plots which depict the computed residual, the LSQR residual depicted in Figure 7.3 is the true residual. In iterations after the usual termination test is satisfied, the computed and true residuals start to diverge substantially. In this example, CGNR (not plotted) performed worse than LSQR, even when CGNR was allowed to take many extra iterations.

As mentioned above, MINRES-L required 1169 iterations even though the system size, namely qn, is only 112. It is known that in exact arithmetic MINRES should never require more iterations than the system size. The excessive number of iterations required by MINRES-L is apparently caused by a loss of orthogonality in the Lanczos process. To verify this hypothesis, we ran GMRES on the same layered matrix (4.3). GMRES [25] on a symmetric matrix is equivalent to MINRES with full reorthogonalization. In exact arithmetic the two algorithms produce identical iterates, errors, and residuals. We call this algorithm GMRES-L. The same termination tests were used. The GMRES-L result is also depicted in Figure 7.3. In this case, GMRES-L ran for 99 iterations (fewer than qn) and returned a more accurate answer, one with forward error $4.3 \cdot 10^{-14}$. The number of flops, 2.6M, was also lower than the MINRES-L flops despite the expensive Gram–Schmidt process in the GMRES main loop.

Our final computational test involves a much larger matrix A arising from finite-

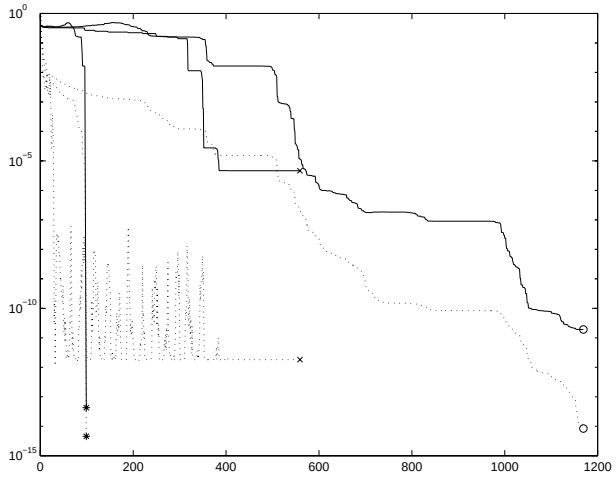


FIG. 7.3. *Convergence behavior of LSQR, MINRES-L, and GMRES-L for ADLITTLE. The MINRES-L and LSQR curves are labeled as in* Figure 7.2. *The GMRES-L curves are labeled "—∗" for the scaled error and "· · ·∗" for the residual.*

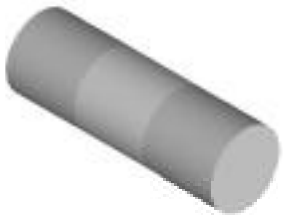


FIG. 7.4. *Geometry of model used for large test of MINRES-L, which is a cylinder in three sections.*

element analysis. In particular, we consider computation of the displacements given by linear elasticity in a three-dimensional cylindric rod. The domain is depicted in Figure 7.4. Although the domain happens to be axisymmetric, it was treated as a general three-dimensional object. The domain was meshed using QMG [19], a three-dimensional unstructured finite-element mesh generator that produces tetrahedral elements. The total number of elements in the mesh is 1090 and the number of nodes is 1802. Therefore, the number of unknowns in the system (degrees of freedom) is 5406. The middle segment of the domain is composed of a very flexible material compared to the outer two segments. In particular, the Young's modulus for the center segment is 10^{-12} times its value for the outer segments.

This system was obtained from the finite element formulator in FRANC3D [7] and is not available in WLS form. Nonetheless, it is known (see, e.g., [27]) that elasticity can be derived from first principles using least-squares theory. Furthermore, the MINRES-L algorithm does not actually need A and D explicitly; instead, it needs the partition of $A^T DA$ into the two terms $\delta_1 A_1^T D_1 A_1 + \delta_2 A_2^T D_2 A_2$. This partition can be obtained from the individual element stiffness matrices produced by FRANC3D without reference to A and D.

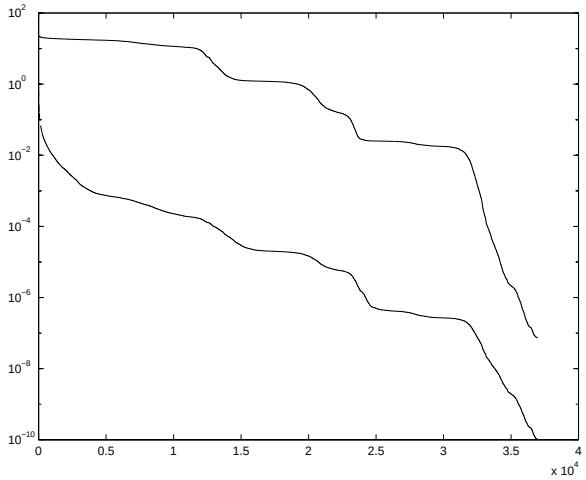


FIG. 7.5. *Convergence behavior of MINRES-L on a large problem. The upper curve is the scaled error, and the lower curve is the residual.*

The absence of A and D in explicit form, however, does mean that the COD method cannot be easily applied to give the "exact" solution. In addition, even if A and D were available, this problem is too large for the COD method, which uses all dense matrix operations. The COD method has not been extended to sparse matrices.

Therefore, we need a technique to measure the error in the MINRES-L solution that does not require knowledge of the exact solution. To address this, we performed the following experiment. We made up two random right-hand sides $\mathbf{b}_1$ and $\mathbf{b}_2$ and used MINRES-L to compute solutions $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. Then we ran MINRES-L a third time to solve for $A^T DA\mathbf{x}_3 = A^T DA(\mathbf{b}_1 + \mathbf{b}_2)$. The "error" in $\mathbf{x}_3$ was taken to be the difference $\|\mathbf{x}_3 - \mathbf{x}_2 - \mathbf{x}_1\|$.

A second difficulty was that MINRES-L did not converge (i.e., the residual was not driven to a small number) for this problem without a preconditioner. Therefore, we used a preconditioner of the form $[I, 0; 0, P]$. This form of preconditioner for MINRES-L is (loosely) motivated by the ideas of Fischer et al. [9] for preconditioning equilibrium (a.k.a. saddle-point or KKT) systems. In the limit $\delta_2/\delta_1 \to 0$, the two-layered MINRES-L system is a saddle-point system. The preconditioner P in turn is the globally extracted element-by-element preconditioner for linear elasticity from [15] and attributed to Bartelt [1]. This matrix P is based only on the rigid parts of the domain.

The plot in Figure 7.5 shows the convergence behavior of the residual and "error" (as explained above) for this problem. The total number of iterations (for obtaining $\mathbf{x}_3$) was 36977 for a total of 128 Gflops. The termination test for this study was a decrease in the computed residual by a factor 10^{-10}. It appears from the plot, however, that we could have extracted additional accuracy by continuing for more iterations. Although convergence is very slow, the plot nonetheless shows the desirable correlation between residual and error predicted by theory. We also tried the conjugate gradient method applied to the normal equations (1.2) of this problem in both preconditioned and unpreconditioned forms. CG diverged on this problem, so the results are not plotted. (Note that CGNR and LSQR are not applicable to this problem since, as mentioned above, we do not have access to its least-squares formulation.) The reason

that CG diverged is suspected to be as follows. The matrix $A^T DA$ masquerades as a very rank-deficient matrix. It is known that CG applied to a rank-deficient matrix with a consistent right-hand side will, in exact arithmetic, converge. But it is also known that for large problems, roundoff error causes components in the nullspace of A to enter the Krylov basis and slowly get magnified, potentially leading to divergence.

The number of iterations for MINRES-L in this test, even with the preconditioner, is still much larger than the size of the augmented system, which is 10812×10812. In fact, it seems that MINRES-L is probably not appropriate for problems of this scale until a better preconditioner can be found. We suspect that a preconditioner is necessary for the following reason. As observed in the medium-scale test described earlier, MINRES-L appears to suffer from loss of orthogonality. For large problems, the loss is so severe that it prevents convergence entirely. The only workaround known to us is to hasten convergence with a good preconditioner or with a novel reorthogonalization scheme.

We also tried unpreconditioned MINRES-L on some of the larger Netlib linear programming test problems. In all of the large cases it failed to converge, presumably for the same reason as in the last paragraph. We are not aware of a good way to precondition MINRES-L for that class of problems.

8. An issue for interior-point methods. In this section we describe an issue that arises when using the MINRES-L algorithm in an interior-point method for linear programming. Full consideration of this matter is postponed to future work.

It is well known that the system of equations for the Newton step in an interior-point method can be expressed as a WLS problem. To be precise, consider the linear programming problem

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A^T\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

whose dual is

$$\begin{array}{rl} \text{maximize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & A\mathbf{y} + \mathbf{s} = \mathbf{c}, \\ & \mathbf{s} \geq \mathbf{0} \end{array}$$

(which is standard form, except that we have transposed A to be consistent with least-squares notation). A primal-dual method starting at a feasible interior point $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ for this problem computes an update $\Delta\mathbf{y}$ to $\mathbf{y}$ satisfying

$$A^T DA\Delta\mathbf{y} = A^T D(\mathbf{s} - \sigma\mu X^{-1}\mathbf{e}), \tag{8.1}$$

where $X = \text{diag}(\mathbf{x})$, $S = \text{diag}(\mathbf{s})$, $D = XS^{-1}$, σ is an algorithm-dependent parameter usually in $[0, 1]$, μ is the duality gap, and $\mathbf{e}$ is the vector of all 1's. See Wright [32]. Since (8.1) has the form of a WLS problem, we can obtain $\Delta\mathbf{y}$ using the MINRES-L algorithm.

One way to compute $\Delta\mathbf{s}$ is via $\Delta\mathbf{s} := -A\Delta\mathbf{y}$. This method is not stable because $\Delta\mathbf{s}$ has very small entries in positions where $\mathbf{s}$ has very small entries; these small entries must be computed accurately with respect to the corresponding entry of $\mathbf{s}$. In contrast, the error in all components of $\Delta\mathbf{s}$ arising from the product $A\Delta\mathbf{y}$ is on the order of $\epsilon \cdot \|\mathbf{s}\|$ (where ϵ is machine-epsilon). A direct method for accurately computing

all components of $\Delta\mathbf{s}$ was proposed by Hough [16], who obtains a bound of the form

$$|\Delta s_i - \widehat{\Delta s_i}|/s_i \leq f(A)\cdot\epsilon \tag{8.2}$$

for each i. We will consider methods for extending MINRES-L to accurate computation of $\Delta\mathbf{s}$ in future work. As noted by Hough, $\Delta\mathbf{x}$ is easily computed from $\Delta\mathbf{s}$ with a similar accuracy bound assuming $\Delta\mathbf{s}$ satisfies (8.2).

9. Conclusions. We have presented an iterative algorithm MINRES-L for solving WLS. Theory and computational experiments indicate that the method is more accurate than LSQR and CGNR when the weight matrix is highly ill-conditioned. This work raises a number of questions.

1. The most pressing problem is the loss of orthogonality in MINRES. We have proposed to handle the problem with a preconditioner. Is there another approach, for instance, reorthogonalization? So far we have not found any other approach.
2. Speaking of preconditioners, what is the best way to precondition MINRES-L? The technique of Fischer et al. [9] for equilibrium systems seems promising, except that the technique must be generalized to the case that the upper-left block does not have full rank.
3. An additional issue concerning preconditioning is that the analysis of MINRES-L's achievable accuracy in section 5 assumes that no preconditioner is used. If a preconditioner is used, then the theorem of Brown and Walker [6] no longer applies.
4. Can this work be extended to componentwise accurate computation of $\Delta\mathbf{x}$ and $\Delta\mathbf{s}$ in an interior-point method? (This question was raised in section 8.)
5. Michael Saunders observed that MINRES-L uses normal-equation operators of the form $A_1^T D_1 A_1$. Use of normal equations in iterative methods is generally considered inferior to using the factors separately. For example, the CGNR and LSQR algorithms are more accurate than CG on the normal equations unless the latter are well-conditioned. Is there a method to carry out a layered computation without forming normal equations? Saunders devised exactly such a method for the case when $p = 2$. In addition to the variables $(\mathbf{x};\mathbf{v})$ already present in (4.3), Saunders proposed additional new variables:

$$\begin{aligned}\mathbf{u} &= -D_1^{1/2}A_1\mathbf{v},\\ \mathbf{r} &= D_1^{1/2}\mathbf{b}_1 - D_1^{1/2}A_1\mathbf{x} - \delta_2\mathbf{u}/\delta_1,\\ \mathbf{s} &= D_2^{1/2}\mathbf{b}_2 - D_2^{1/2}A_2\mathbf{x}.\end{aligned}$$

Then one checks that $(\mathbf{r};\mathbf{x};\mathbf{s};\mathbf{u};\mathbf{v})$ satisfies

(9.1)
$$\begin{pmatrix} & & & I & D_1^{1/2}A_1 \\ & & A_2^T D_2^{1/2} & A_1^T D_1^{1/2} & \\ & D_2^{1/2}A_2 & I & & \\ I & D_1^{1/2}A_1 & & \delta_2 I/\delta_1 & \\ A_1^T D_1^{1/2} & & & & \end{pmatrix}\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\\ \mathbf{s}\\ \mathbf{u}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\\ D_2^{1/2}\mathbf{b}_2\\ D_1^{1/2}\mathbf{b}_1\\ \mathbf{0}\end{pmatrix}.$$

Note that this matrix is symmetric and therefore may be solved with MINRES.

This method could be more accurate than MINRES-L when A is ill-conditioned. We constructed an artificial small example of this kind by adding a large multiple of the matrix of all 1's to the node-arc adjacency matrix used in the first test in section 7. We found that Saunders's algorithm was superior to MINRES-L for this test case. Unfortunately, we were able to test Saunders's method only on smaller problems because MINRES did not converge for (9.1) on larger problems (presumably because of loss of orthogonality), and we do not know how to precondition (9.1).

Finally, we mention more recent work by Howle and Vavasis, not yet published on preconditioned iterative methods for network problems. That paper raises (and solves) an issue about accuracy similar to the accuracy issue raised in this paper. The difference between the two approaches is that the Howle–Vavasis method iterates with (3.1) as the coefficient matrix, coupled with a particular preconditioner that prevents the early drop in the residual. The use of a preconditioner alone is not enough to fix the accuracy problem since there is still a loss of information in forming matrix-vector products. The Howle–Vavasis method requires further the ability to rapidly project a vector into the nullspace of A_1. In some combinatorial settings such as network problems, this projection is easy to compute. But it is not known how to generalize the Howle–Vavasis algorithm to the case when projection into $N(A_1)$ is not available. The MINRES-L method in this paper makes no such assumption.

Acknowledgments. We had helpful discussions of this work with Anne Greenbaum and Mike Overton of NYU; Roland Freund, David Gay, and Margaret Wright of Bell Labs; Patty Hough of Sandia; Rich Lehoucq and Steve Wright of Argonne; Vicki Howle of Cornell; Homer Walker of Utah State; and Zdeněk Strakoš of the Czech Academy of Sciences. We thank Patty Hough and Gail Pieper for carefully reading an earlier draft of this paper. Michael Saunders and the anonymous referee provided many helpful comments and suggestions for improvement. The FRANC3D finite element formulator for the large test case was provided by members of the Cornell Fracture Group. In addition, we received the Netlib linear programming test cases in Matlab format from Patty Hough.

REFERENCES

[1] P. Bartelt, *Finite Element Procedures on Vector/Tightly Coupled Parallel Computers*, Verlag der Fachvereine, Zürich, 1989.

[2] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[3] A. Björck, T. Elfving, and Z. Strakoš, *Stability of conjugate gradient and Lanczos methods for linear least squares problems*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 720–736.

[4] E. Bobrovnikova and S. Vavasis, *Iterative methods for weighted least squares,* in Proceedings of the Copper Mountain Conference on Iterative Methods, University of Colorado, Copper Mountain, CO, 1996.

[5] E. Y. Bobrovnikova and S. A. Vavasis, *Accurate Solution of Weighted Least Squares by Iterative Methods*, Tech. Report ANL/MCS-P644-0297, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1997.

[6] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

[7] Cornell Fracture Group, *Franc3d website*. See http://www.cfg.cornell.edu, 1999.

[8] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of GMRES*, BIT, 25 (1995), pp. 309–330.

[9] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.

[10] A. L. Forsgren, *On linear least-squares problems with diagonally dominant weight matrices*, SIAM J. Matrix Anal. App., 17 (1996), pp. 763–788.

[11] G. Golub and C. V. Loan, *Matrix Computations,* 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[12] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. App., 18 (1997), pp. 535–551.
[13] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[14] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman, Harlow, UK, 1995.
[15] I. Hladík, M. B. Reed, and G. Swoboda, *Robust preconditioners for linear elasticity FEM analysis*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 2109–2127.
[16] P. Hough, *Stable Computation of Search Directions for Near-Degenerate Linear Programming Problems*, Tech. Report SAND97-8243, Sandia National Laboratories, Livermore, CA, 1997.
[17] P. D. Hough and S. A. Vavasis, *Complete orthogonal decomposition for weighted least squares*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 369–392.
[18] C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974. Republished by SIAM, Philadelphia, 1995.
[19] S. A. Mitchell and S. A. Vavasis, *Quality mesh generation in higher dimensions*, SIAM J. Comput., 29 (2000), pp. 1334–1370.
[20] D. P. O'Leary, *On bounds for scaled projections and pseudoinverses*, Linear Algebra Appl., 132 (1990), pp. 115–117.
[21] C. Paige and M. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[22] C. Paige and M. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[23] C. C. Paige, *Practical use of the symmetric Lanczos process with re-orthogonalization*, BIT, 10 (1970), pp. 183–195.
[24] B. Parlett and D. Scott, *The Lanczos algorithm with selective reorthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[26] G. W. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra Appl., 112 (1989), pp. 189–193.
[27] G. Strang, *A framework for equilibrium equations*, SIAM Rev., 30 (1988), pp. 283–297.
[28] M. J. Todd, *A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm*, Oper. Res., 38 (1990), pp. 1006–1018.
[29] S. A. Vavasis, *Stable numerical algorithms for equilibrium systems*, SIAM J. Matrix Anal. Appl, 15 (1994), pp. 1108–1131.
[30] S. A. Vavasis and Y. Ye, *A primal-dual interior point method whose running time depends only on the constraint matrix*, Math. Programming, 74 (1996), pp. 79–120.
[31] C. Vuik, A. Segal, and J. A. Meijerink, *An Efficient Preconditioned CG Method for the Solution of Layered Problems with Extreme Contrasts in the Coefficients*, J. Comput. Phys., 152 (1999), pp. 385–403.
[32] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1175–1189

INVERSION OF ANALYTIC MATRIX FUNCTIONS THAT ARE SINGULAR AT THE ORIGIN*

KONSTANTIN E. AVRACHENKOV†, MOSHE HAVIV‡, AND PHIL G. HOWLETT§

Abstract. In this paper we study the inversion of an analytic matrix valued function $A(z)$. This problem can also be viewed as an analytic perturbation of the matrix $A_0 = A(0)$. We are mainly interested in the case where A_0 is singular but $A(z)$ has an inverse in some punctured disc around $z = 0$. It is known that $A^{-1}(z)$ can be expanded as a Laurent series at the origin. The main purpose of this paper is to provide efficient computational procedures for the coefficients of this series. We demonstrate that the proposed algorithms are computationally superior to symbolic algebra when the order of the pole is small.

Key words. matrix inversion, matrix valued functions, analytic perturbation, Laurent series

AMS subject classifications. 15A09, 41A58, 47A55, 47A56

PII. S0895479898337555

1. Introduction.

Let $\{A_k\}_{k=0,1,\ldots} \subseteq \mathbb{R}^{n\times n}$ be a sequence of matrices that defines the analytic matrix valued function

$$A(z) = A_0 + zA_1 + z^2A_2 + \cdots . \tag{1}$$

The above series is assumed to converge in some nonempty neighborhood of $z = 0$. We will also say that $A(z)$ is an *analytic perturbation* of the matrix $A_0 = A(0)$. Assume the inverse matrices $A^{-1}(z)$ exist in some (possibly punctured) disc centered at $z = 0$. In particular, we are primarily interested in the case where A_0 is singular. In this case it is known that $A^{-1}(z)$ can be expanded as a Laurent series in the form

$$A^{-1}(z) = \frac{1}{z^s}(X_0 + zX_1 + \cdots), \tag{2}$$

where $X_0 \neq 0$ and s is a natural number, known as the order of the pole at $z = 0$. The main purpose of this paper is to provide efficient computational procedures for the Laurent series coefficients X_k, $k \geq 0$. As one can see from the following literature review, few computational methods have been considered in the past.

The inversion of nearly singular operator valued functions was probably first studied in the paper by Keldysh [22]. In that paper he studied the case of a polynomial perturbation

$$A(z) = A_0 + zA_1 + \cdots + z^mA_m, \tag{3}$$

where A_k, $1 \leq k \leq m$, are compact operators on Hilbert space. In particular, he showed that the principal part of the Laurent series expansion for the inverse operator

*Received by the editors April 16, 1998; accepted for publication (in revised form) by A. Ran August 4, 2000; published electronically March 20, 2001. This work was supported in part by Australian Research Council grant A49532206.

http://www.siam.org/journals/simax/22-4/33755.html

†INRIA Sophia Antipolis, 2004 route des Lucioles, B.P. 93, 06902, Sophia Antipolis Cedex, France (k.avrachenkov@sophia.inria.fr).

‡Department of Statistics, The Hebrew University, 91905 Jerusalem, Israel and Department of Econometrics, The University of Sydney, Sydney, NSW 2006, Australia (haviv@mscc.huji.ac.il).

§CIAM, School of Mathematics, The University of South Australia, The Levels, SA 5095, Australia (phil.howlett@unisa.edu.au).

$A^{-1}(z)$ can be given in terms of generalized Jordan chains. The generalized Jordan chains were initially developed in the context of matrix and operator polynomials (see [13, 26, 30] and numerous references therein). However, the concept can be easily generalized to the case of an analytic perturbation (1).

Following Gohberg and Sigal [15] and Gohberg and Rodman [14], we say that the vectors $\varphi_0, \ldots, \varphi_{r-1}$ form a Jordan chain of the perturbed matrix $A(z)$ at $z = 0$ if $\varphi_0 \neq 0$ and if

$$\sum_{i=0}^{k} A_i \varphi_{k-i} = 0$$

for each $0 \leq k \leq r-1$. Note that φ_0 is an eigenvector of the unperturbed matrix A_0 corresponding to the zero eigenvalue. The number r is called the length of the Jordan chain and φ_0 is the initial vector. Let $\{\varphi_0^{(j)}\}_{j=1}^{p}$ be a system of linearly independent eigenvectors, which span the null space of A_0. Then one can construct Jordan chains initializing at each of the eigenvectors $\varphi_0^{(j)}$. This generalized Jordan set plays a crucial role in the analysis of analytic matrix valued functions $A(z)$.

Gantmacher [11] analyzed the polynomial matrix (3) by using the canonical Smith form. Vishik and Lyusternik [37] studied the case of a linear perturbation $A(z) = A_0 + zA_1$ and showed that one can express $A^{-1}(z)$ as a Laurent series as long as $A(z)$ is invertible in some punctured neighborhood of the origin. In addition, an undetermined coefficient method for the calculation of Laurent series terms was given in [37]. Langenhop [25] showed that the coefficients of the regular part of the Laurent series for the inverse of a linear perturbation form a geometric sequence. The proof of this fact was refined later in Schweitzer [33, 34] and Schweitzer and Stewart [35]. In particular, [35] proposed a method for computing the Laurent series coefficients. However, the method of [35] cannot be applied (at least immediately) to the general case of an analytic perturbation. Many authors have obtained existence results for operator valued analytic and meromorphic functions [3, 15, 23, 27, 29, 36]. In particular, Gohberg and Sigal [15] used a local Smith form to elaborate on the structure of the principal part of the Laurent series in terms of generalized Jordan chains. Recently, Gohberg, Kaashoek, and Van Schagen [12] have refined the results of [15]. Furthermore, Bart, Kaashoek, and Lay [5] used their results on the stability of the null and range spaces [4] to prove the existence of meromorphic relative inverses of finite meromorphic operator valued functions. The ordinary inverse operator is a particular case of the relative inverse. For the applications of the inversion of analytic matrix functions, see, for example, [8, 9, 20, 23, 24, 28, 31, 32, 36].

Howlett [20] provided a computational procedure for the Laurent series coefficients based on a sequence of row and column operations on the coefficients of the original power series (1). Howlett used the rank test of Sain and Massey [32] to determine s, the order of the pole. He also showed that the coefficients of the Laurent series satisfy a finite linear recurrence relation in the case of a polynomial perturbation. The method of [20] can be considered as a starting point for our research. The algebraic reduction technique which is used in the present paper was introduced by Haviv and Ritov [17, 18] in the special case of stochastic matrices. Haviv, Ritov, and Rothblum [19] also applied this approach to the perturbation analysis of semisimple eigenvalues.

In this paper we provide three related methods for computing the coefficients of the Laurent series (2). The first method uses generalized inverse matrices to solve a set of linear equations and extends the work in [17] and [20]. The other two methods use results that appear in [2, 17, 18, 19] and are based on a reduction technique

[6, 10, 21, 23]. All three methods depend in a fundamental way on equating coefficients for various powers of z. By substituting the series (1) and (2) into the identity $A(z)A^{-1}(z) = I$ and collecting coefficients of the same power of z, one obtains the following system which we will refer to as the fundamental equations:

$$\begin{aligned}
(4.0) \qquad A_0X_0 &= 0,\\
(4.1) \qquad A_0X_1 + A_1X_0 &= 0,\\
&\vdots \quad \vdots\\
(4.s) \qquad A_0X_s + \cdots + A_sX_0 &= I,\\
(4.s+1) \qquad A_0X_{s+1} + \cdots + A_{s+1}X_0 &= 0,\\
&\vdots \quad \vdots\,.
\end{aligned}$$

A similar system can be written when considering the identity $A^{-1}(z)A(z) = I$, but of course the set of fundamental equations $(4.0), (4.1), \ldots$ is sufficient. Finally, for matrix operators, each infinite system of linear equations uniquely determines the coefficients of the Laurent series (2). This fact has been noted in [3, 20, 23, 37, 36].

2. Main results. Define the following augmented matrix $\mathcal{A}^{(t)} \in \mathbb{R}^{(t+1)n\times(t+1)n}$:

$$\mathcal{A}^{(t)} = \begin{bmatrix} A_0 & 0 & 0 & \cdots & 0\\ A_1 & A_0 & 0 & \cdots & 0\\ A_2 & A_1 & A_0 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ A_t & A_{t-1} & \cdots & A_1 & A_0 \end{bmatrix};$$

and let us prove the following basic lemma.

LEMMA 1. *Let s be the order of the pole at the origin for the inverse function $A^{-1}(z)$. Any eigenvector $\Phi \in \mathbb{R}^{(s+1)n}$ of $\mathcal{A}^{(s)}$ corresponding to the zero eigenvalue possesses the property that its first n elements are zero.*

Proof. Suppose on the contrary that there exists an eigenvector $\Phi \in \mathbb{R}^{(s+1)n}$ such that

$$\mathcal{A}^{(s)}\Phi = 0 \tag{5}$$

and not all of its first n entries are zero. Then, partition the vector Φ into $s+1$ blocks and rewrite (5) in the form

$$A_0\varphi_0 = 0,$$

$$A_0\varphi_1 + A_1\varphi_0 = 0,$$

$$\vdots$$

$$A_0\varphi_s + \cdots + A_s\varphi_0 = 0$$

with $\varphi_0 \neq 0$. This means that we have found a generalized Jordan chain of length $s + 1$. However, from the results of Gohberg and Sigal [15], we conclude that the

maximal length of a generalized Jordan chain of $A(z)$ at $z = 0$ is s. Hence, we came to a contradiction and, consequently, $\varphi_0 = 0$. ∎

REMARK 1. *A direct proof of Lemma* 1 *is given in Appendix* 1.

REMARK 2. *All vectors* $\Phi \in \mathbb{R}^{(s+j+1)n}$ *in the null space of the augmented matrix* $\mathcal{A}^{(s+j)}$, $j \geq 0$, *possess the property that the first* $(j+1)n$ *elements are zero.*

The following theorem provides a theoretical basis for the recursive solution of the infinite system of fundamental equations (4).

THEOREM 1. *Each coefficient* X_k, $k \geq 0$, *is uniquely determined by the previous coefficients* $X_0, \ldots, X_{k-1}$ *and the set of* s *fundamental equations* $(4.k)$–$(4.k+s)$.

Proof. It is obvious that the sequence of Laurent series coefficients $\{X_i\}_{i=0}^{\infty}$ is a solution to the fundamental equations (4). Suppose the coefficients X_i, $0 \leq i \leq k-1$, have been determined. Next we show that the set of fundamental equations $(4.k)$–$(4.k+s)$ uniquely determines the next coefficient X_k. Indeed, suppose there exists another solution $\tilde{X}_k$. Since X_k and $\tilde{X}_k$ are both solutions, we can write

$$\mathcal{A}^{(s)} \begin{bmatrix} \tilde{X}_k \\ \vdots \\ \tilde{X}_{k+s} \end{bmatrix} = \begin{bmatrix} J_k - \sum_{i=1}^{k} A_i X_{k-i} \\ \vdots \\ J_{k+s} - \sum_{i=1}^{k} A_{i+s} X_{k-i} \end{bmatrix} \tag{6}$$

and

$$\mathcal{A}^{(s)} \begin{bmatrix} X_k \\ \vdots \\ X_{k+s} \end{bmatrix} = \begin{bmatrix} J_k - \sum_{i=1}^{k} A_i X_{k-i} \\ \vdots \\ J_{k+s} - \sum_{i=1}^{k} A_{i+s} X_{k-i} \end{bmatrix}, \tag{7}$$

where the matrix J_i is defined as

$$J_i = \begin{cases} I, & i = s, \\ 0 & \text{otherwise} \end{cases}$$

and where $\tilde{X}_{k+1}, \ldots, \tilde{X}_{k+s}$ are any particular solutions of the nonhomogenous linear system $(4.k)$–$(4.k+s)$. Note that (6) and (7) have identical right-hand sides. Of course, the difference between these two right-hand sides, $[\tilde{X}_k - X_k \cdots \tilde{X}_{k+s} - X_{k+s}]^T$, is in the right null space of $\mathcal{A}^{(s)}$. Invoking Lemma 1, the first n rows of $[\tilde{X}_k - X_k, \ldots, \tilde{X}_{k+s} - X_{k+s}]^T$ are hence zero. In other words, $\tilde{X}_k - X_k = 0$, which proves the theorem. ∎

Using the above theoretical background, in the next section we provide three recursive computational schemes which are based on the generalized inverses and on a reduction technique. The reduction technique is based on the following result. A weaker version of this result was utilized in [17] and in [19].

THEOREM 2. *Let* $\{C_k\}_{k=0}^{t} \subseteq \mathbb{R}^{m \times m}$ *and* $\{R_k\}_{k=0}^{t} \subseteq \mathbb{R}^{m \times n}$, *with* $m \leq n$, *and suppose that the system of* $t+1$ *matrix equations*

$$\begin{aligned} C_0 V_0 &= R_0, & (8.0) \\ C_0 V_1 + C_1 V_0 &= R_1, & (8.1) \\ &\vdots \quad \vdots & \\ C_0 V_t + \cdots + C_t V_0 &= R_t & (8.t) \end{aligned}$$

is feasible. Then the general solution is given by

$$V_k = C_0^{\dagger}\left(R_k - \sum_{i=1}^{k} C_i V_{k-i}\right) + QW_k, \tag{9}$$

where $C_0^{\dagger}$ is the Moore–Penrose generalized inverse of C_0 and $Q \in \mathbb{R}^{m\times p}$ is any matrix whose columns form a basis for the right null space of C_0. Furthermore, the sequence of matrices W_k, $0 \le k \le t-1$, solves a reduced finite set of t matrix equations

$$\begin{aligned} D_0W_0 &= S_0, && (10.0)\\ D_0W_1 + D_1W_0 &= S_1, && (10.1)\\ &\vdots \quad\quad \vdots\\ D_0W_{t-1} + \cdots + D_tW_0 &= S_{t-1}, && (10.t-1) \end{aligned}$$

where the matrices $D_k \in \mathbb{R}^{p\times p}$ and $S_k \in \mathbb{R}^{p\times n}$, $0 \le k \le t-1$, are computed by the following recursion. Set $U_0 = C_1$ and calculate

$$U_k = C_{k+1} - \sum_{i=1}^{k} C_i C_0^{\dagger} U_{k-i}, \quad k = 1, \ldots, t-1. \tag{11}$$

Then,

$$D_k = MU_kQ \quad \textit{and} \quad S_k = M\left(R_{k+1} - \sum_{i=0}^{k} U_i C_0^{\dagger} R_{k-i}\right), \tag{12}$$

where $M \in \mathbb{R}^{p\times m}$ is any matrix whose rows form a basis for the left null space of C_0.

Proof. The general solution to the matrix equation (8.0) can be written in the form

$$V_0 = C_0^{\dagger}R_0 + QW_0, \tag{13}$$

where $W_0 \in \mathbb{R}^{p\times n}$ is some arbitrary matrix.

In order for the equation

$$C_0V_1 = R_1 - C_1V_0$$

to be feasible, we need that the right-hand side $R_1 - C_1V_0$ belongs to $R(C_0) = N^{\perp}(C_0^T)$, that is,

$$M(R_1 - A_1V_0) = 0,$$

where the rows of M form a basis for $N(C_0^T)$. Substituting expression (13) for the general solution V_0 into the above feasibility condition, one finds that W_0 satisfies the equation

$$M(R_1 - C_1(C_0^{\dagger}R_0 + QW_0)) = 0$$

which can be rewritten as

$$MC_1QW_0 = M(R_1 - C_1C_0^{\dagger}R_0).$$

Thus we have obtained the first reduced fundamental equation (10.0) with

$$D_0 := MU_0Q \quad \text{and} \quad S_0 := M(R_1 - U_0C_0^\dagger R_0),$$

where $U_0 = C_1$. Next we observe that the general solution of (8.1) is represented by the formula

$$V_1 = C_0^\dagger(R_1 - C_1V_0) + QW_1 \tag{14}$$

with $W_1 \in \mathbb{R}^{p\times n}$. Moving on and applying the feasibility condition to (8.2), we obtain

$$M(R_2 - (C_1V_1 + C_2V_0)) = 0$$

and again the substitution of expressions (13) and (14) into the above condition yields

$$MC_1(C_0^\dagger(R_1 - C_1[C_0^\dagger R_0 + QW_0]) + QW_1) + MC_2(C_0^\dagger R_0 + QW_0) = MR_2$$

which is rearranged to give

$$MC_1QW_1 + M(C_2 - C_1C_0^\dagger C_1)QW_0 = M(R_2 - C_1C_0^\dagger R_1 - (C_2 - C_1C_0^\dagger C_1)C_0^\dagger R_0).$$

The last equation is the reduced equation (10.1) with

$$D_1 := MU_1Q \quad \text{and} \quad S_1 := M(R_2 - U_0C_0^\dagger R_1 - U_1C_0^\dagger R_0),$$

where $U_1 = C_2 - C_1C_0^\dagger U_0$. Note that this equation imposes restrictions on W_1 as well as on W_0. By proceeding in the same way, we eventually obtain the complete system of equations (9) with coefficients given by formulas (11) and (12) each of which can be proved by induction in a straightforward way. □

REMARK 3. *In the above theorem it is important to observe that the reduced system has the same form as the original but the number of matrix equations is decreased by one and the coefficients are reduced in size to matrices in $\mathbb{R}^{p\times p}$, where p is the dimension of $N(C_0)$ or, equivalently, the number of redundant equations defined by the coefficient C_0.*

In the next section we use this reduction process to solve the system of fundamental equations. Note that the reduction process can be employed to solve any appropriate finite subset of the fundamental equations.

3. Solution methods. In this section we discuss three methods for solving the fundamental equations. The first method is based on the direct application of Moore–Penrose generalized inverses. The second method involves the replacement of the original system of the fundamental equations by a system of equations with a reduced dimension. In the third method we show that the reduction process can be applied recursively to reduce the problem to a nonsingular system. Since all methods depend to some extent on the prior knowledge of s, we begin by discussing a procedure for the determination of s. A special procedure for determining this order for the case where the matrices $A(z)$ are stochastic and the perturbation series is finite is given in [16]. It is based on combinatorial properties (actually, network representation) of the processes and hence it is a stable procedure. However, as will be seen in section 3.4, it is possible to use the third method without prior knowledge of s. Actually, the full reduction version of our procedure determines s as well. Of course, as in any computational method which is used to determine indices which have discrete values, using our procedures in order to compute the order of singularity might lack stability.

3.1. The determination of the order of the pole. The rank test on the matrix $\mathcal{A}^{(t)}$ proposed by Sain and Massey in [32] is likely to be the most effective procedure for determining the value of s. The calculation of rank is essentially equivalent to the reduction of $\mathcal{A}^{(t)}$ to a row echelon normal form and it can be argued that row operations can be used successively in order to calculate the rank of $\mathcal{A}^{(0)}$,$\mathcal{A}^{(1)}$,$\mathcal{A}^{(2)}$,... and find the minimum value of t for which $\text{rank}\mathcal{A}^{t} = \text{rank}\mathcal{A}^{(t-1)} + n$. This minimum value of t equals s, the order of the pole. Note that previous row operations for reducing $\mathcal{A}^{(t-1)}$ to row echelon form are replicated in the reduction of $\mathcal{A}^{(t)}$ and do not need to be repeated. For example, if a certain combination of row operations reduces A_0 to row echelon form, then the same operations are used again as part of the reduction of

$$\begin{bmatrix} A_0 & 0 \\ A_1 & A_0 \end{bmatrix}$$

to row echelon form.

3.2. Basic generalized inverse method. In this section we obtain a recursive formula for the Laurent series coefficients X_k, $k \geq 0$, by using the Moore–Penrose generalized inverse of the augmented matrix $\mathcal{A}^{(s)}$.

Let $\mathcal{G}^{(s)} \stackrel{def}{=} [\mathcal{A}^{(s)}]^{\dagger}$ be the Moore–Penrose generalized inverse of $\mathcal{A}^{(s)}$ and define the matrices $G_{ij}^{(s)} \in \mathbb{R}^{n\times n}$ for $0 \leq i, j \leq t$ by

$$\mathcal{G}^{(s)} = \begin{bmatrix} G_{00}^{(s)} & \cdots & G_{0s}^{(s)} \\ \vdots & \ddots & \vdots \\ G_{s0}^{(s)} & \cdots & G_{ss}^{(s)} \end{bmatrix}.$$

Furthermore, we would like to note that in fact we use only the first n rows of the generalized inverse $\mathcal{G}^{(s)}$, namely, $[G_{00}^{(s)} \ \cdots \ G_{0s}^{(s)}]$.

PROPOSITION 1. *The coefficients of the Laurent series* (2) *can be calculated by the recursive formula*

$$X_k = \sum_{j=0}^{s} G_{0j}^{(s)} \left(J_{j+k} - \sum_{i=1}^{k} A_{i+j} X_{k-i} \right) \quad , \quad k \geq 1, \tag{15}$$

where $X_0 = G_{0s}^{(s)}$ *and the matrix* J_i *is defined by*

$$J_i = \begin{cases} I, & i = s \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. According to Theorem 1, once the coefficients X_i, $0 \leq i \leq k-1$ are determined, the next coefficient X_k can be obtained from the $(4.k)$–$(4.k+s)$ fundamental equations.

$$\mathcal{A}^{(s)} \begin{bmatrix} X_k \\ \vdots \\ X_{k+s} \end{bmatrix} = \begin{bmatrix} J_k - \sum_{i=1}^{k} A_i X_{k-i} \\ \vdots \\ J_{k+s} - \sum_{i=1}^{k} A_{i+s} X_{k-i} \end{bmatrix}.$$

The general solution to the above system is given in the form

$$\begin{bmatrix} X_k \\ \tilde{X}_{k+1} \\ \vdots \\ \tilde{X}_{k+s} \end{bmatrix} = \begin{bmatrix} G_{00}^{(s)} & \cdots & G_{0s}^{(s)} \\ G_{10}^{(s)} & \cdots & G_{1s}^{(s)} \\ \vdots & \ddots & \vdots \\ G_{s0}^{(s)} & \cdots & G_{ss}^{(s)} \end{bmatrix} \begin{bmatrix} J_k - \sum_{i=1}^{k} A_i X_{k-i} \\ J_{k+1} - \sum_{i=1}^{k} A_{i+1} X_{k-i} \\ \vdots \\ J_{k+s} - \sum_{i=1}^{k} A_{i+s} X_{k-i} \end{bmatrix} + \begin{bmatrix} 0 \\ \Phi_1 \\ \vdots \\ \Phi_s \end{bmatrix},$$

where the first block of matrix Φ is equal to zero according to Lemma 1. Thus, we immediately obtain the recursive expression (15). In particular, applying the same arguments as above to the first $s+1$ fundamental equations, we obtain that $X_0 = G_{0s}^{(s)}$. □

Note that the matrices J_{j+k} in the expression (15) disappear when the regular coefficients are computed.

REMARK 4. *The formula* (15) *is a generalization of the recursive formula for the case where* A_0 *is invertible. In this case,*

$$X_k = -A_0^{-1} \sum_{i=1}^{k} A_i X_{k-i}, \quad k \geq 1,$$

while initializing with $X_0 = A_0^{-1}$.

REMARK 5. *Probably from the computational point of view it is better not to compute the generalized inverse* $\mathcal{G}^{(s)}$ *beforehand, but rather to find the SVD or LU decomposition of* $\mathcal{A}^{(s)}$ *and then use these decompositions for solving the fundamental equations* $(4.k)$–$(4.k+s)$. *This is the standard approach for solving linear systems with various right-hand sides.*

3.3. The one-step-reduction process. In this section we describe an alternative scheme that can be used in the case where it is relatively easy to compute the bases for the right and for the left null spaces of A_0. Specifically, let $p = n - r(A_0)$ be the dimension of the null space of A_0, let $Q \in \mathbb{R}^{n \times p}$ be a matrix whose p columns form a basis for the right null space of A_0, and let $M \in \mathbb{R}^{p \times n}$ be a matrix whose p rows form a basis for the left null space of A_0. Of course, although $p = 0$ and hence $s = 0$ is possible, we are interested in the singular case where $p \geq 1$.

Again, as before, we suppose that the coefficients X_i, $0 \leq i \leq k-1$, have already been determined. Then, by Theorem 1, the next coefficient X_k is the unique solution to the subsystem of fundamental equations

$$\begin{aligned} A_0 X_k &= J_k - \sum_{i=1}^{k} A_i X_{k-i}, \\ A_0 X_{k+1} + A_1 X_k &= J_{k+1} - \sum_{i=1}^{k} A_{i+1} X_{k-i}, \\ &\vdots \\ A_0 X_{k+s} + \cdots + A_s X_k &= J_{k+s} - \sum_{i=1}^{k} A_{i+s} X_{k-i}. \end{aligned} \tag{16}$$

The above system is like the one given in (8) with $C_i = A_i$, $0 \leq i \leq s$, and with $R_j = J_{k+j} - \sum_{i=1}^{k} A_{i+j} X_{k-i}$, $0 \leq j \leq s$. Therefore, we can apply the reduction

process described in Theorem 2. This results in the system

$$\begin{aligned} D_0W_0 &= S_0, \\ D_0W_1 + D_1W_0 &= S_1, \\ &\vdots \\ D_0W_{s-1} + \cdots + D_{s-1}W_0 &= S_{s-1}, \end{aligned} \tag{17}$$

where the coefficients D_i and S_i, $i = 0, \ldots, s-1$, can be calculated by the recursive formulae (11) and (12).

REMARK 6. *Note that in many practical applications p is much less than n and hence the above system* (17) *with $D_i \in \mathbb{R}^{p\times p}$ is much smaller than the original system* (16).

Now we have two options. We can either apply the reduction technique again (see the next subsection for more details) or we can solve the reduced system directly by using the generalized inverse approach. In the latter case, we define

$$\mathcal{D}^{(t)} \stackrel{def}{=} \begin{bmatrix} D_0 & 0 & 0 & \cdots & 0 \\ D_1 & D_0 & 0 & \cdots & 0 \\ D_2 & D_1 & D_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_t & D_{t-1} & \cdots & D_1 & D_0 \end{bmatrix}$$

and

$$\mathcal{H}^{(t)} = \begin{bmatrix} H_{00}^{(t)} & \cdots & H_{0t}^{(t)} \\ \vdots & \ddots & \vdots \\ H_{t0}^{(t)} & \cdots & H_{tt}^{(t)} \end{bmatrix} \stackrel{def}{=} [\mathcal{D}^{(t)}]^{\dagger}.$$

Then, by carrying out a similar computation to the one presented in the proof of Proposition 1, we obtain

$$W_0 = \sum_{i=0}^{s-1} H_{0i}^{(s-1)} S_i.$$

Once W_0 is determined it is possible to obtain X_k from the formula

$$X_k = A_0^{\dagger}R_0 + QW_0 = A_0^{\dagger}R_0 + Q\sum_{i=0}^{s-1} H_{0i}^{(s-1)} S_i.$$

Furthermore, substituting for S_i, $0 \le i \le s-1$, from (12) and changing the order of summation gives

$$X_k = \left(A_0^{\dagger} - \sum_{i=0}^{s-1} QH_{0i}^{(s-1)} MU_i A_0^{\dagger}\right) R_0 + \sum_{j=1}^{s} \left(QH_{0j-1}^{(s-1)} M - \sum_{i=j}^{s-1} QH_{0i}^{(s-1)} MU_{i-j} A_0^{\dagger}\right) R_j. \tag{18}$$

Note that by convention the sum disappears when the lower limit is greater than the upper limit. Now, substituting $R_j = J_{k+j} - \sum_{i=1}^{k} A_{i+j} X_{k-i}$, $0 \le j \le s$, into

the expression (18), we obtain the explicit recursive formula for the Laurent series coefficients

$$X_k = \left(A_0^\dagger - \sum_{i=0}^{s-1} Q H_{0i}^{(s-1)} M U_i A_0^\dagger \right) \left(J_k - \sum_{i=1}^{k} A_i X_{k-i} \right)$$

$$+ \sum_{j=1}^{s} \left(Q H_{0j-1}^{(s-1)} M - \sum_{i=j}^{s-1} Q H_{0i}^{(s-1)} M U_{i-j} A_0^\dagger \right) \left(J_{k+j} - \sum_{i=1}^{k} A_{i+j} X_{k-i} \right) \tag{19}$$

for all $k \geq 1$. In particular, the coefficient of the first singular term in (2) can be given by the formula

$$X_0 = Q H_{0s-1}^{(s-1)} M. \tag{20}$$

3.4. The complete reduction process. As was pointed out in the previous section, the reduced system has essentially the same structure as the original one and hence one can apply again the reduction step described in Theorem 2. Note that each time the reduction step is carried out, the number of matrix equations is reduced by one. Therefore one can perform up to s reduction steps. We now outline how these steps can be executed. We start by introducing the sequence of reduced systems. The fundamental matrix equations for the lth reduction step are

$$A_0^{(l)} X_0^{(l)} = R_0^{(l)}, \tag{21.0}$$

$$A_0^{(l)} X_1^{(l)} + A_1^{(l)} X_0^{(l)} = R_1^{(l)}, \tag{21.1}$$

$$\vdots \qquad \vdots$$

$$A_0^{(l)} X_{s-l}^{(l)} + \cdots + A_{s-l}^{(l)} X_0^{(l)} = R_{s-l}^{(l)}. \tag{21.$s-l$}$$

With $l = 0$, one gets the original system of fundamental equations and with $l = 1$ one gets the reduced system for the first reduction step described in the previous subsection. Initializing with $R_i^{(0)} = 0$, $0 \leq i \leq s-1$, and $R_s^{(0)} = I$ and with $A_i^{(0)} = A_i$, $0 \leq i \leq s$, the matrices $A_j^{(l)}$ and $R_j^{(l)}$, $0 \leq j \leq s-l$, for each reduction step $1 \leq l \leq s$, can be computed successively by a recursion similar to (11) and (12). In general we have

$$U_0^{(l)} = A_1^{(l-1)}, \quad U_j^{(l)} = A_{j+1}^{(l-1)} - \sum_{i=1}^{j} A_i^{(l-1)} A_0^{(l-1)\dagger} U_{j-i}^{(l)}, \quad j = 1, \ldots, s-l,$$

$$A_j^{(l)} = M^{(l)} U_j^{(l)} Q^{(l)}, \quad j = 0, \ldots, s-l,$$

$$R_j^{(l)} = M^{(l)} \left(- \sum_{i=0}^{j} U_{j-i}^{(l)} A_0^{(l-1)\dagger} R_i^{(l-1)} + R_{j+1}^{(l-1)} \right), \quad j = 0, \ldots, s-l,$$

where $Q^{(l)}$ and $M^{(l)}$ are the basis matrices for the right and left null spaces, respectively, of the matrix $A_0^{(l-1)}$ and where $A_0^{(l-1)\dagger}$ is the Moore–Penrose generalized inverse of $A_0^{(l-1)}$. After s reduction steps, one gets the final system of reduced equations

$$A_0^{(s)} X_0^{(s)} = R_0^{(s)}. \tag{22}$$

Since X_0 is a unique solution to the subsystem of fundamental equations (4.0)–(4.s) and Theorem 2 states the equivalence of the lth and $(l+1)$st systems of reduced equations, the system (22) possesses a unique solution, and hence matrix $A_0^{(s)}$ is invertible. Thus,

$$X_0^{(s)} = [A_0^{(s)}]^{-1} R_0^{(s)}. \tag{23}$$

The original solution $X_0 = X_0^{(0)}$ can now be retrieved by the backwards recursive relationship

$$X_0^{(l-1)} = A_0^{(l-1)\dagger} R_0^{(l-1)} + Q^{(l)} X_0^{(l)}, \quad l = s, \ldots, 1. \tag{24}$$

Now by taking $R_j^{(0)} = J_{k+j} - \sum_{i=1}^{k} A_{i+j} X_{k-i}$, $0 \le j \le s$, one gets the algorithm for computing the Laurent series coefficients $X_k, k \ge 1$. Of course, recursive formulae similar to (15) and (19) can be obtained, but they are quite complicated in the general case.

The order s of the pole can also be obtained from the reduction process by continuing the process until $A_0^{(l)}$ becomes nonsingular. The number of reduction steps equals the order of the pole. Note also that the sequence of matrices $A_0^{(l)}$, $l \ge 0$, can be computed irrespectively of the right hand sides. Once s is determined, one can compute $R_j^{(l)}$, $1 \le l \le s$, $0 \le j \le s - l$.

4. Computational complexity and comparison with symbolic algebra. In this section we compare the computational complexity of the one-step-reduction process when applied to compute X_0 with the complexity of symbolic algebra. In particular, we show that the former comes with a reduced complexity in the case where the pole has a relatively small order. The computational complexity of the other two procedures can be determined similarly.

To compute the coefficients D_i, $0 \le i \le s-1$, of the reduced fundamental system (17), one needs to perform $O(s^2n^3)$ operations. The total number of reduced equations is sp. (Recall that p is the dimension of the null space of A_0.) Hence, the computational complexity for determining X_0 by the one-step-reduction process is $O(\max\{s^2n^3, s^3p^3\})$. The Laurent series (2) in general, and the coefficient X_0 in particular, can also be computed by using symbolic algebra. This, for example, can be executed by MATLAB symbolic toolbox and is done as follows. Since X_0 is uniquely determined by the first $s+1$ fundamental equations (4.0), ..., (4.s), all one needs to do in order to compute X_0 is to invert symbolically the following matrix polynomial:

$$\hat{A}(z) = A_0 + zA_1 + \cdots + z^s A_s. \tag{25}$$

Symbolic computations here mean performing operations, such as multiplication and division, over the field of rational functions (and not over the field of the reals). In particular, if the degrees of numerators and of denominators of rational functions do not exceed q, then each operation (multiplication or division) which is performed in the field of rational functions translates into $q\log(q)$ operations in the field of real numbers [1]. Note that during the symbolic inversion of the polynomial matrix (25), the degree of rational functions does not exceed sn. The latter fact follows from Cramer's rule. Thus, the complexity of the symbolic inversion of (25) equals $O(n^3) \times O(sn\log(sn)) = O(sn^4\log(sn))$. As a result, one gets a matrix $\hat{A}^{-1}(z)$ whose elements are rational functions of z. The elements of the matrix X_0 can then be immediately calculated by

dividing the leading coefficients of the numerator and denominator. Finally, one can see that if $s << n$ and $p << n$, which is typically the case, then our method comes with a reduced computational burden.

5. Concluding remarks. In this paper we have shown that the Laurent series for the inversion of an analytic matrix valued function can be computed by solving a system of fundamental linear equations. Furthermore, we demonstrated that the system of fundamental equations can be solved recursively. In particular, the coefficient X_k is determined by the previous coefficients $X_0, \ldots, X_{k-1}$ and the next $s+1$ fundamental equations, where s is the order of the pole. We suggest three basic methods: one without any reduction (see (15)), one with a single reduction step (see (19) and (20)), and one using a complete reduction process with s steps (see (23) and (24)). Of course, an intermediate process with the number of reductions between 1 and s could be used too. We note that when the complete reduction process is used the order of the pole can be determined through the execution of the algorithm. When $s << n$ and $p << n$, the proposed algorithms far outperform the method based on symbolic algebra.

Appendix 1: Another proof of Lemma 1. A direct proof of Lemma 1 can be carried out using augmented matrices. Specifically, define

$$\mathcal{X}^{(t)} = \begin{bmatrix} X_0 & 0 & 0 & \cdots & 0 \\ X_1 & X_0 & 0 & \cdots & 0 \\ X_2 & X_1 & X_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_t & X_{t-1} & \cdots & X_1 & X_0 \end{bmatrix},$$

where X_k, $0 \le k \le t$, are the coefficients of the Laurent series (2). Then it follows from the fundamental systems (4) and (5) that the augmented matrices $\mathcal{A}^{(t)}$ and $\mathcal{X}^{(t)}$ satisfy the relationship

$$\mathcal{A}^{(t)}\mathcal{X}^{(t)} = \mathcal{X}^{(t)}\mathcal{A}^{(t)} = \mathcal{E}^{(t)}, \tag{26}$$

where the augmented matrix $\mathcal{E}^{(t)} \in \mathbb{R}^{(t+1)n \times (t+1)n}$ is defined by setting $\mathcal{E}^{(t)} = [E_{pq}]_{p,q=0}^{t}$, where $E_{pq} \in \mathbb{R}^{n \times n}$ and

$$E_{pq} = \begin{cases} I & \text{for} \quad p - q = s, \\ 0 & \text{for} \quad p - q \neq s. \end{cases}$$

Now, as before, let $\Phi \in \mathbb{R}^{(s+1)n}$ satisfy the equation

$$\mathcal{A}^{(s)}\Phi = 0. \tag{27}$$

If we multiply (27) from the left by $\mathcal{X}^{(s)}$, then it reduces to

$$\mathcal{E}^{(s)}\Phi = 0.$$

The vector $\mathcal{E}^{(s)}\Phi$ has φ_0 as the $(s+1)$st block, which gives the required result.

Appendix 2: A numerical example. Let us consider the matrix valued function

$$A(z) = A_0 + zA_1 = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} + z \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix},$$

where $\operatorname{rank}(A_0) = 2$. Construct the augmented matrices

$$\mathcal{A}^{(0)} = A_0 \quad \text{and} \quad \mathcal{A}^{(1)} = \begin{bmatrix} A_0 & 0 \\ A_1 & A_0 \end{bmatrix},$$

and note that $\operatorname{rank}(\mathcal{A}^{(1)}) - \operatorname{rank}(\mathcal{A}^{(0)}) = 5 - 2 = 3$, which is the dimension of the original coefficients A_0 and A_1. Therefore, according to the test of Sain and Massey [32], the Laurent expansion for $A^{-1}(z)$ has a simple pole. Alternatively, we can compute a basis for $N(\mathcal{A}^{(1)})$, which in this particular example consists of only one vector

$$q^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & -3 \end{bmatrix}^T .$$

The first three zero elements in $q^{(1)}$ confirm that the Laurent series has a simple pole. Next we compute the generalized inverse of $\mathcal{A}^{(1)}$ given by

$$\mathcal{A}^{(1)\dagger} = \mathcal{G}^{(1)} = \begin{bmatrix} G_{00}^{(1)} & G_{01}^{(1)} \\ G_{10}^{(1)} & G_{11}^{(1)} \end{bmatrix} = \begin{bmatrix} 1/3 & -5/12 & -1/12 & 1/8 & 1/8 & -1/8 \\ 0 & 1/4 & 1/4 & 1/8 & 1/8 & -1/8 \\ 1/3 & -5/12 & -1/12 & -3/8 & -3/8 & 3/8 \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}.$$

Consequently,

$$X_0 = G_{01}^{(1)} = \frac{1}{8} \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -3 & -3 & 3 \end{bmatrix}. \tag{28}$$

Alternatively, we know that X_0 is uniquely determined by the fundamental equations

$$\begin{aligned} A_0 X_0 &= 0, \\ A_0 X_1 + A_1 X_0 &= I. \end{aligned}$$

After one reduction step these equations reduce to

$$M A_1 Q W_0 = M,$$

where

$$M = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 1 & 1 & -3 \end{bmatrix}^T .$$

Hence,

$$W_0 = (M A_1 Q)^{-1} M = \frac{1}{8} \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$$

and

$$X_0 = Q W_0 = \begin{bmatrix} 1 \\ 1 \\ -3 \end{bmatrix} \frac{1}{8} \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -3 & -3 & 3 \end{bmatrix}.$$

The latter expression is identical with (28) and coincides with the one computed by expanding $A^{-1}(z)$ with the help of the MATLAB symbolic toolbox. Note that even for this three-dimensional example the direct symbolic calculation of the Laurent series takes a relatively long time.

Acknowledgment. The authors are grateful to Prof. Jerzy A. Filar for his helpful advice. Also the authors would like to thank the anonymous referees for their valuable suggestions and for directing us to some existing literature.

REFERENCES

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.
[2] K.E. Avrachenkov and J.B. Lasserre, *The fundamental matrix of singularly perturbed Markov chains*, Adv. in Appl. Prob., 31 (1999), pp. 679–697.
[3] H. Bart, *Meromorphic Operator Valued Functions*, Thesis, Vrije Universiteit, Amsterdam, Math. Center Tract 44, 1973.
[4] H. Bart, M.A. Kaashoek, and D.C. Lay, *Stability properties of finite meromorphic operator functions*, Nederl. Akad. Wetensch. Proc. Ser. A, 36 (1974), pp. 217–259.
[5] H. Bart, M.A. Kaashoek, and D.C. Lay, *Relative inverses of meromorphic operator functions and associated holomorphic projection functions*, Math. Ann., 218 (1975), pp. 199–210.
[6] H. Baumgärtel, *Analytic Perturbation Theory for Matrices and Operators*, Birkhäuser, Basel, 1985.
[7] S.L. Campbell and C.D. Meyer, *Generalized Inverses of Linear Transformation*, Pitman, London, 1979.
[8] S.L. Campbell, *Singular Systems of Differential Equations*, Pitman Res. Notes Math. Ser. 40, Longman, Harlow, UK, 1980.
[9] S.L. Campbell, *Singular Systems of Differential Equations,* Vol. II, Pitman Res. Notes Math. Ser. 61, Longman, Harlow, UK, 1982.
[10] F. Delebecque, *A reduction process for perturbed Markov chains*, SIAM J. Appl. Math., 43 (1983), pp. 325–350.
[11] F.R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1959.
[12] I. Gohberg, M.A. Kaashoek, and F. van Schagen, *On the local theory of regular analytic matrix functions,* Linear Algebra Appl., 182 (1993), pp. 9–25.
[13] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Computer Science and Applied Mathematics Series, Academic Press, New York, 1982.
[14] I. Gohberg and L. Rodman, *Analytic matrix functions with prescribed local data*, J. Analyse Math., 40 (1981), pp. 90–128.
[15] I.C. Gohberg and E.I. Sigal, *An operator generalization of the logarithmic residue theorem and the theorem of Rouché*, Math. USSR Sbornik, 13 (1971), pp. 603–625.
[16] R. Hassin and M. Haviv, *Mean passage times and nearly uncoupled Markov chains*, SIAM J. Discrete Math., 5 (1992), pp. 386–397.
[17] M. Haviv and Y. Ritov, *Series Expansions for Stochastic Matrices*, unpublished manuscript, 1989.
[18] M. Haviv and Y. Ritov, *On series expansions and stochastic matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 670–676.
[19] M. Haviv, Y. Ritov, and U.G. Rothblum, *Taylor expansions of eigenvalues of perturbed matrices with applications to spectal radii of nonnegative matrices,* Linear Algebra Appl., 168 (1992), pp. 159–188.
[20] P.G. Howlett, *Input retrieval in finite dimensional linear systems*, J. Austral. Math. Soc. Ser. B, 23 (1982), pp. 357–382.
[21] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, 1966.
[22] M.V. Keldysh, *On the characteristic values and characteristic functions of certain classes of non-selfadjoint equations*, Dokl. Akad. Nauk USSR, 77 (1951), pp. 11–14.
[23] V.S. Korolyuk and A.F. Turbin, *Mathematical Foundations of the State Lumping of Large Systems*, Naukova Dumka, Kiev, 1978.
[24] P. Lancaster, *Inversion of lambda-matrices and application to the theory of linear vibrations*, Arch. Ration. Mech. Anal., 6 (1960), pp. 105–114.
[25] C.E. Langenhop, *The Laurent expansion for a nearly singular matrix*, Linear Algebra Appl., 4 (1971), pp. 329–340.
[26] A.S. Markus, *Introduction to the Spectral Theory of Polynomial Operator Pencils*, Transl. Math. Monographs 71, AMS, Providence, RI, 1988.
[27] M.V. Pattabhiraman and P. Lancaster, *Spectral properties of a polynomial operator*, Numer. Math., 13 (1969), pp. 247–259.
[28] A.A. Pervozvanski and V.G. Gaitsgori, *Theory of Suboptimal Decisions*, Kluwer Academic

Publishers, Dordrecht, The Netherlands, 1988.
[29] M. Ribarič and I. Vidav, *Analytic properties of the inverse $A^{-1}(z)$ of an analytic operator valued function $A(z)$*, Arch. Rational Mech. Anal., 32 (1969), pp. 298–310.
[30] L. Rodman, *An Introduction to Operator Polynomials*, Oper. Theory Adv. Appl. 38, Birkhäuser, Boston, 1989.
[31] N.J. Rose, *The Laurent expansion of a generalized resolvent with some applications*, SIAM J. Math. Anal., 9 (1978), pp. 751–758.
[32] M.K. Sain and J.L. Massey, *Invertibility of linear time invariant dynamical systems*, IEEE Trans. Automat. Control, AC-14 (1969), pp. 141–149.
[33] P.J. Schweitzer, *The Laurent Expansion for a Nearly Singular Pencil*, Working Paper QM8413, Graduate School of Management, University of Rochester, Rochester, NY, 1984.
[34] P.J. Schweitzer, *Perturbation series expansions for nearly completely-decomposable Markov chains*, in Teletraffic Analysis and Computer Performance Evaluation, O.J. Boxma, J.W. Cohen, and H.C. Tijms, eds., Elsevier Science Publishers, North-Holland, Amsterdam, 1986, pp. 319–328.
[35] P.J. Schweitzer and G.W. Stewart, *The Laurent expansion of pencils that are singular at the origin*, Linear Algebra Appl., 183 (1993), pp. 237–254.
[36] M.M. Vainberg and V.A. Trenogin, *Theory of Branching of Solutions of Non-Linear Equations*, Noordhoff International Publishing, Leyden, 1969.
[37] M.I. Vishik and L.A. Lyusternik, *The solution of some perturbation problems in the case of matrices and self-adjoint and non-self-adjoint differential equations*, Uspechi Mat. Nauk, 15 (1960), pp. 3–80.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1190–1203

ON INFINITE PRODUCTS OF FUZZY MATRICES*

SY-MING GUU†, YUNG-YIH LUR‡, AND CHIN-TZONG PANG†

Abstract. In this paper, we study the convergence of infinite products of a finite number of fuzzy matrices, where the operations involved are max-min algebra. Two types of convergences in this context will be discussed: the weak convergence and strong convergence. Since any given fuzzy matrix can be "decomposed" of the sum of its associated Boolean matrices, we shall show that the weak convergence of infinite products of a finite number of fuzzy matrices is equivalent to the weak convergence of infinite products of a finite number of the associated Boolean matrices. Further characterizations regarding the strong convergence will be established. On the other hand, sufficient conditions for the weak convergence of infinite products of fuzzy matrices are proposed. A necessary condition for the weak convergence of infinite products of fuzzy matrices is presented as well.

Key words. Boolean matrices, fuzzy matrices, convergence of infinite products of fuzzy matrices

AMS subject classifications. 03G05, 15A57

PII. S0895479800366021

1. Introduction.

Unlike the convergence of infinite products of a finite number of matrices that has been studied quite extensively for several decades [1], [6], the same issue in the context of fuzzy matrices seems to be ignored. For a fuzzy matrix A, we mean $A = [a_{ij}]$ with $a_{ij} \in [0,1]$. Let $\mathbb{F}$ denote the unit interval, i.e., $\mathbb{F}= [0,1]$. We let $\mathbb{F}^{m\times n}$ denote the set of all the $m \times n$ fuzzy matrices. The algebraic operations of fuzzy matrices in this paper are max-min operations. Clearly, a Boolean matrix is a special fuzzy matrix. In practice, fuzzy matrices have been proposed to represent fuzzy relations in a system based on fuzzy sets theory [9], [15], [20].

The study of convergence of products of a finite number of fuzzy matrices arises from the field of time-invariant discrete-time fuzzy systems with nonfuzzy inputs (see chapter 2 in Dubois and Prade [2]). The time-invariant fuzzy systems can be studied within the same conceptual framework as classical dynamic systems. Precisely,[1] let u_t, y_t, and s_t denote the input, output, and state of a system Ω, respectively, at time t. Let U, Y, S be the sets of possible inputs, outputs, and states. And S_{t+1} and Y_t are fuzzy sets on S and Y. The dynamic fuzzy system may be represented by the *fuzzy transition relation* δ and *fuzzy output map* σ, where

$$s_{t+1} = \delta(u_t, s_t),\ y_t = \sigma(s_t, u_t),\ t \in N. \tag{1}$$

An initial state is denoted by s_0. Note that the δ and σ are fuzzy relations in $S\times U\times S$ and $Y \times S \times U$, respectively. If we assume further that all the state set S, input set U, and the output set Y are finite, then δ and σ have fuzzy matrix representations,

*Received by the editors February 1, 2000; accepted for publication (in revised form) by V. Mehrmann October 11, 2000; published electronically March 20, 2001.
http://www.siam.org/journals/simax/22-4/36602.html

†Department of Information Management, Yuan Ze University, Taoyuan, Taiwan, 320, People's Republic of China. (iesmguu@saturn.yzu.edu.tw, imctpang@saturn.yzu.edu.tw). The work of the first author was partially supported by National Science Council grant NSC-89-2213-E-155-018. The research of the third author was supported in part by National Science Council grant NSC 89-2115-M-155-001.

‡Department of Information Management, Fortune Institute of Technology, Kaohsiung, Taiwan, 842, People's Republic of China.

[1]For easy reference, we shall follow the settings in [2].

respectively. For instance, let $S = \{s^1, \ldots, s^m\}$ and $Y = \{y^1, \ldots, y^k\}$. Then for each input u_t we may define an $m \times m$ fuzzy matrix

$$M(u_t) = [M_{ij}(u_t)], \text{ where } M_{ij}(u_t) = \nu(s^i | s^j, u_t).$$

Here $\nu(\bullet)$ is an appropriate membership function. Similarly, we may define

$$M_Y(u_t) = [\mu(y^i | s^j, u_t)], \text{ where } \mu(\bullet) \text{ is a membership function.}$$

Let $\bar{s}_t = [\mu_s(s_t^1), \ldots, \mu_s(s_t^m)]^T$ and $\bar{y}_t = [\mu_y(y_t^1), \ldots, \mu_y(y_k^m)]^T$, where μ_s and μ_y are membership functions for S_t and Y_t, respectively. Then with these matrix representations, (1) may become (see chapter 9 in Mizumoto [14])

$$\bar{s}_{t+1} = M(u_t) \circ \bar{s}_t, \ \bar{y}_t = M_Y(u_t) \circ \bar{s}_t, \ t \in N, \tag{2}$$

where $\circ$ stands for the max-min operations. It follows from (2) that the behavior of the dynamic fuzzy system depends heavily on the products of fuzzy matrices $M(u_t)$s.

Theory of fuzzy sets has played an active role in the field of medicine (see chapter 6 in Klir and Folger [10]). The states of (2) may be interpreted as diseases. For instance, S may contain two elements: mild influenza and severe influenza. The input part of (2) may be interpreted as the appropriate therapeutic actions. For instance, during the treatment process, the physician may apply Panadol Cold & Flu or Sinutab to treat the patient. The output part of (2) may be interpreted as the clinical manifestations. For instance, the output set may contain elements such as fever and cough.

When the input set U contains only one element, all $M(u_t)$'s reduce to be a common fuzzy matrix. And the study of (2) depends on the understanding of the powers of this fuzzy matrix. In the literature, convergence of power of a fuzzy matrix does attract researchers to study. Thomason [19] proved that the powers of a fuzzy matrix are either convergent to an idempotent fuzzy matrix or oscillating with finite period, which is the same consequence of the powers of a Boolean matrix. Thomason proposed some sufficient conditions to establish convergence as well. The main concept employed in these sufficient conditions is to assume compactness for the given fuzzy matrix. Hashimoto [7] assumed the fuzzy matrix to be transitive to have convergence. Indeed, both compactness and transitivity induce convergence because of the monotonicity of its powers. Later, Fan and Liu [3] defined the notion of *maximum principle* and showed that if the fuzzy matrix satisfies the maximum principle, then its powers possess the monotonicity (starting, however, from the second power) and hence the sequence of its powers is convergent. Kolodziejczyk [11] showed that if the fuzzy matrix is strongly transitive (s-transitive), then its powers either converge or oscillate with period 2. Li [13] defined the notion of controllable fuzzy matrix and pointed out that all the nilpotent fuzzy matrices, symmetric fuzzy matrices, (max-min) transitive fuzzy matrices, and s-transitive fuzzy matrices are controllable fuzzy matrices but not vice versa in general. He showed that the powers of a controllable fuzzy matrix either converge or oscillate with period 2. Li [12] also studied the periodicity of powers of a fuzzy matrix. Fan and Liu [4] explored the oscillating property of the powers of a fuzzy matrix. They decomposed a fuzzy matrix to be the sum of a finite number of corresponding Boolean matrices. Through this decomposition, they can derive results for the oscillation index and period index of the fuzzy matrix by studying the respective index of the associated Boolean matrices. To understand the powers of a

Boolean matrix [8], [17], we refer to De Schutter and De Moor [18] where they completely characterized the ultimate behavior of the sequence of the consecutive powers of a matrix in Boolean algebra.

Guu, Chen, and Pang [5] generalized the study of convergence of powers of a fuzzy matrix to consider the products of a finite number of fuzzy matrices. They showed that the behavior of the infinite products is quite different from the powers of a fuzzy matrix:

- When a path (to be defined in section 2) converges to a fuzzy matrix, this "limiting" matrix is not necessarily idempotent.
- One cannot assert that in general an oscillating path should be with finite period.
- Each path generated from these fuzzy matrices may converge. However, a different path may converge to a different fuzzy matrix. This is the notion of weak convergence. When all the paths happen to converge to the same limit, the limit fuzzy matrix is idempotent. This is the notion of strong convergence. Since the powers of a given fuzzy matrix is the unique path, the convergence of the powers of a fuzzy matrix is exactly the strong convergence type.

Compactness and transitivity were extended as well to establish sufficient conditions for weak convergence of infinite products of a finite number of fuzzy matrices.

In this paper, we shall explore further the convergence of infinite products of a finite number of fuzzy matrices. Three main directions will be presented. First, similar to Fan and Liu's decomposition, we shall establish the weak convergence of infinite products of a finite number of fuzzy matrices in terms of certain (finite) Boolean matrices. Equivalence in strong convergence of infinite products of fuzzy matrices and infinite products of the associated Boolean matrices will be established as well. Further characterizations for infinite products of fuzzy matrices that converge strongly to zero are given. These constitute the work of section 3. Second, monotone properties such as compactness and transitivity are useful to establish the convergence of powers of a fuzzy matrix. By following the strategy of monotonicity, we shall construct suitable monotone conditions as sufficient conditions for the weak convergence. Third, a sufficient condition which somehow violates the spirit of monotonicity, where all the elements of infinite products are either increasing or decreasing, will be proposed to establish the weak convergence. Under this sufficient condition, we can show that the off-diagonal elements of infinite products are increasing, while the diagonal elements are decreasing. These will appear in section 4. Section 2 contains the preliminaries and some backgrounds in this study. Conclusions will be given in the final section.

2. Preliminaries. For any fuzzy matrices A and B of the same size, we have the sum of $A = [a_{ij}]$ and $B = [b_{ij}]$ as follows:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} := \max\{a_{ij}, b_{ij}\}.$$

If $A \in \mathbb{F}^{s\times n}$ and $B \in \mathbb{F}^{n\times s}$, then the product $A \otimes B$ of A and B is defined as follows:

$$[A \otimes B]_{ij} = \oplus_{k=1}^{n}\{a_{ik} \otimes b_{kj}\} := \max\{a_{ik} \otimes b_{kj} | k = 1, 2, \ldots, n\},$$

where $a_{ij} \otimes b_{ij} := \min\{a_{ij}, b_{ij}\}$. The product $A \otimes B$ is of size $s \times s$. For any fuzzy matrices A and B of the same size, we say $A \leq B$ if and only if $a_{ij} \leq b_{ij}$ for all i and j. For a fuzzy matrix $A = [a_{ij}]$, we may denote a_{ij} by $[A]_{ij}$.

Consider fuzzy matrices $A^{(1)}, A^{(2)}, \ldots, A^{(m)}$ with each $A^{(i)} \in \mathbb{F}^{n\times n}$. Let $\mathcal{F}$ denote the set of underlying fuzzy matrices, that is, $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\}$. Let $\mathcal{F}_k$ be

the set of all products of matrices in $\mathcal{F}$ of length k, that is,

$$\mathcal{F}_k = \{A_k \otimes A_{k-1} \otimes \cdots \otimes A_2 \otimes A_1 | A_i \in \mathcal{F}\ \forall i = 1, 2, \ldots, k\}.$$

For our convenience, we denote $F(k) = A_k \otimes A_{k-1} \otimes \cdots \otimes A_2 \otimes A_1$.

DEFINITION 2.1. *The sequence* $\{F(k)\}$ *is a* path *in set* $\cup_{k\geq 1}\mathcal{F}_k$ *if* $F(1) \in \mathcal{F}$ *and for each* $k \geq 1$, $F(k+1) = A_{k+1} \otimes F(k)$, *where* $A_{k+1} \in \mathcal{F}$.

Let $\mathcal{P}$ *denote the set containing all the paths in* $\cup_{k\geq 1}\mathcal{F}_k$. *We shall call* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *the* path system *generated by* $\mathcal{F}$. *We note that if* $\mathcal{F}$ *contains only one fuzzy matrix, say* $\mathcal{F} = \{B\}$, *then* $\mathcal{F}_k = \{B^k\}$, *the* k*th power of* B. *This is the only one path generated by* $\mathcal{F}$.

DEFINITION 2.2. *The fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is* weakly convergent *if each path* $\{F(k)\}$ *in* $\mathcal{P}$ *is convergent. Moreover, if all the paths* $\{F(k)\}$ *in* $\mathcal{P}$ *converges to the same fuzzy matrix, we say that the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is* strongly convergent.

For $\lambda \in [0, 1]$, *we follow* [4] *to define for any fuzzy matrix* A *the* λ*-level cut matrix* A_λ *by*

$$[A_\lambda]_{ij} = \begin{cases} 1 & \text{if } a_{ij} \geq \lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the cut matrix A_λ *is a Boolean matrix. We let* Φ_A *denote the set of all nonzero elements of* A. *It is easy to see that* $A = \oplus_{\lambda\in\Phi_A}\lambda \otimes A_\lambda$. *The* $\Phi_{\mathcal{F}}$ *is the union of* $\Phi_{A^{(i)}}$ *for all* $A^{(i)}$ *in* $\mathcal{F}$. *For any* λ, *we denote* $\mathcal{F}_\lambda = \{A_\lambda^{(1)}, A_\lambda^{(2)}, \ldots, A_\lambda^{(m)}\}$. *For our purposes, we let* $\underline{\lambda} := \min\{\lambda | \lambda \in \Phi_{\mathcal{F}}\}$.

Let A *be an* $n \times n$ *Boolean matrix. A nonzero element* $u \in \{0, 1\}^n$ *is called a Boolean eigenvector of* A *if there exists an* s *in* $\{0, 1\}$ *such that* $A \otimes u = s \otimes u$. *This* s *is a Boolean eigenvalue associated with an eigenvector* u. *It is a well-known result that all Boolean matrices have an eigenvalue. Define the Boolean spectral radius of* A *by* $\rho(A)$, *the largest Boolean eigenvalues of* A. *Note that* $\rho(A)$ *is* 0 *or* 1. *We refer to Robert* [16] *for further study in this area.*

3. Connection between fuzzy matrices and Boolean matrices. From the definitions of a fuzzy matrix and a Boolean matrix, we learn that a Boolean matrix is a special type of the fuzzy matrix. As noted in the introduction, Fan and Liu [4] have studied the oscillating property of the powers of a fuzzy matrix through the framework of the associated Boolean matrices. Their idea can be extended to study the convergence of the products of a finite number of fuzzy matrices, to which we now turn.

LEMMA 3.1. *Let* $A, B \in \mathbb{F}^{n\times n}$ *and* $\lambda \in [0, 1]$. *Then*

$$(A \otimes B)_\lambda = A_\lambda \otimes B_\lambda.$$

Proof. By the definition of cut matrices, for $1 \leq i, j \leq n$ we have

$$\begin{aligned}
[(A \otimes B)_\lambda]_{ij} = 1 &\Leftrightarrow [A \otimes B]_{ij} \geq \lambda \\
&\Leftrightarrow \oplus_{k=1}^n (a_{ik} \otimes b_{kj}) \geq \lambda \\
&\Leftrightarrow \text{ there exists a } 1 \leq k' \leq n \text{ such that } a_{ik'} \geq \lambda \text{ and } b_{k'j} \geq \lambda \\
&\Leftrightarrow \oplus_{k=1}^n \{[A_\lambda]_{ik} \otimes [B_\lambda]_{kj}\} = 1 \\
&\Leftrightarrow [A_\lambda \otimes B_\lambda]_{ij} = 1.
\end{aligned}$$

By similar arguments, we have $[(A \otimes B)_\lambda]_{ij} = 0$ if and only if $[A_\lambda \otimes B_\lambda]_{ij} = 0$. This completes the proof. ∎

3.1. Results of weak convergence. Let $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\}$ with each $A^{(i)} \in \mathbb{F}^{n\times n}$. Let $\hat{p}$ denote $(m \times n^2)^{n^2} + 1$. Define

$$V := \{[A^{(k)}]_{ij} | 1 \le i, j \le n; A^{(k)} \in \mathcal{F}\ \forall k = 1, 2, \ldots, m\}$$

and

$$T := \{G \in \mathbb{F}^{n\times n} | \text{ for } 1 \le i, j \le n, [G]_{ij} \in V\}.$$

LEMMA 3.2. *Let* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\}$. *Then*

$$|T| \le (m \times n^2)^{n^2} \text{ and } \cup_{k\ge 1} \mathcal{F}_k \subset T.$$

Proof. Since each entry of $G \in T$ has at most $(m \times n^2)$ choices, the cardinality of T is less than or equal to $(m \times n^2)^{n^2}$. The observation $\cup_{k\ge1}\mathcal{F}_k \subset T$ can be seen from the fact that the max and min operations can't yield an entry not in $\mathcal{F}$. ∎

LEMMA 3.3. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If the fuzzy path system* $(\cup_{k\ge1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent, then for each path* $\{F(k)\}$ *there exist* $1 \le i < j \le \hat{p}$ *such that* $F(i) = F(i + r)$ *for all* $1 \le r \le j - i$.

Proof. By Lemma 3.2, we have $F(k) \in T$ for all $k = 1, 2, \ldots$ and $|T| < \hat{p}$. There exist $1 \le i < j \le \hat{p}$ such that $F(i) = F(j)$. In other words,

$$A_i \otimes \cdots \otimes A_1 = A_j \otimes \cdots \otimes A_1.$$

Construct a new path $\{F'(k)\}$ by

$$\{F'(k)\} = \{F(1), F(2), \ldots, F(i), A_{i+1} \otimes F(i), \ldots, (A_j \otimes \cdots \otimes A_{i+1}) \otimes F(i), \ldots,$$
$$(A_j \otimes \cdots \otimes A_{i+1}) \otimes (A_j \otimes \cdots \otimes A_{i+1}) \otimes F(i), \ldots\}.$$

Since the path system is weakly convergent, this new path is convergent. This implies that

$$A_i \otimes \cdots \otimes A_1 = A_{i+r} \otimes \cdots \otimes A_1 \text{ for } 1 \le r \le j - i. \quad ∎$$

LEMMA 3.4. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If the fuzzy path system* $(\cup_{k\ge1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent, then there exists a positive integer* p *such that* $\mathcal{F}_p = \mathcal{F}_{p+1}$.

Proof. Choose $p = \hat{p}$. Let $A_p \otimes \cdots \otimes A_1 \in \mathcal{F}_p$. By Lemma 3.3, there exist $1 \le i < j \le p$ such that

$$A_i \otimes \cdots \otimes A_1 = A_{i+1} \otimes A_i \otimes \cdots \otimes A_1.$$

Thus,

$$A_p \otimes \cdots \otimes A_1 = A_p \otimes \cdots \otimes A_{i+1} \otimes A_{i+1} \otimes A_i \otimes \cdots \otimes A_1 \in \mathcal{F}_{p+1}.$$

Conversely, for any $A_{p+1} \otimes \cdots \otimes A_1 \in \mathcal{F}_{p+1}$, by Lemma 3.3, there exist $1 \le i < j \le p$ such that

$$A_{p+1} \otimes \cdots \otimes A_1 = A_{p+1} \otimes \cdots \otimes A_{i+2} \otimes A_i \otimes \cdots \otimes A_1 \in \mathcal{F}_p.$$

Thus $\mathcal{F}_{p+1} = \mathcal{F}_p$. This completes the proof. ∎

LEMMA 3.5. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *For* $\lambda \in \Phi_{\mathcal{F}}$, *let* $\{F_\lambda(k)\} \in \mathcal{P}_\lambda$. *Then either* $\{F_\lambda(k)\}$ *is convergent or there are subpaths* $\{F_\lambda(n_k)\}$ *and* $\{F_\lambda(m_k)\}$ *converging to* $\hat{F}_1$ *and* $\hat{F}_2$, *respectively. Here* $\hat{F}_1 \neq \hat{F}_2$.

Proof. It follows from the fact that the cardinality of $\cup_{k\geq 1}(\mathcal{F}_\lambda)_k$ is finite. ∎

THEOREM 3.6. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *The fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent if and only if for each* $\lambda \in \Phi_{\mathcal{F}}$, *the Boolean path system* $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ *is weakly convergent, where* $\mathcal{P}_\lambda$ *denotes the set of all paths in* $\cup_{k\geq 1}(\mathcal{F}_\lambda)_k$.

Proof. Assume that the system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ converges weakly. For $\lambda \in \Phi_{\mathcal{F}}$, we let $\{F_\lambda(k)\} \in \mathcal{P}_\lambda$, where $F_\lambda(k) = (A_k)_\lambda \otimes \cdots \otimes (A_2)_\lambda \otimes (A_1)_\lambda$ with each $A_i \in \mathcal{F}$. Denote $\hat{F}(k) = A_k \otimes \cdots \otimes A_2 \otimes A_1$. Then $\{\hat{F}(k)\}$ is a path in $\mathcal{P}$. Lemma 3.1 implies that

$$(\hat{F}(k))_\lambda = (A_k \otimes \cdots \otimes A_2 \otimes A_1)_\lambda = F_\lambda(k).$$

Since $\{\hat{F}(k)\}$ is weakly convergent, there exists an n_0 such that $\hat{F}(n_0 + j) = \hat{F}(n_0)$ for all $j = 1, 2, \ldots$. Therefore,

$$F_\lambda(n_0 + j) = (\hat{F}(n_0 + j))_\lambda = (\hat{F}(n_0))_\lambda = F_\lambda(n_0) \text{ for } j = 1, 2, \ldots.$$

And we have that $\{F_\lambda(k)\}$ is weakly convergent.

On the other hand, let $\{F(k)\} \in \mathcal{P}$. Suppose that for each $\lambda \in \Phi_{\mathcal{F}}$ the Boolean path system $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ converges weakly. There exists a positive N_λ such that $F_\lambda(N_\lambda + j) = F_\lambda(N_\lambda)$ for $j = 1, 2, \ldots$. Since the cardinal number of $\Phi_{\mathcal{F}}$ is finite, $N^* = \max_{\lambda\in\Phi_{\mathcal{F}}} N_\lambda$ is finite. We then have

$$F_\lambda(N^* + j) = F_\lambda(N^*), \quad \lambda \in \Phi_{\mathcal{F}}, \quad \text{and } j = 1, 2, \ldots.$$

The following relations now hold:

$$F(N^* + j) = \oplus_{\lambda\in\Phi_{\mathcal{F}}}\{\lambda \otimes F_\lambda(N^* + j)\} = \oplus_{\lambda\in\Phi_{\mathcal{F}}}\{\lambda \otimes F_\lambda(N^*)\} = F(N^*) \; \forall\, j = 1, 2, \ldots.$$

Therefore, $\{F(k)\}$ is weakly convergent. The fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is weakly convergent. ∎

For any $k \geq m$, we define the notation $\mathcal{F}^c := \cup_{k\geq 1}\mathcal{F}_k^c$, where

$$\mathcal{F}_k^c := \{M_k \otimes M_{k-1} \otimes \cdots \otimes M_2 \otimes M_1 \in \mathcal{F}_k | \text{ each fuzzy matrix in } \mathcal{F} \text{ should appear at least once in the product } M_k \otimes M_{k-1} \otimes \cdots \otimes M_2 \otimes M_1\}.$$

THEOREM 3.7. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *For* $\lambda \in \Phi_{\mathcal{F}}$, *the following statements* (a) *and* (b) *are mutually equivalent.*

(a) $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ *is weakly convergent.*

(b) (i) $(\cup_{k\geq 1}(\widetilde{\mathcal{F}}_\lambda)_k, \widetilde{\mathcal{P}}_\lambda)$ *is weakly convergent for all proper subset* $\widetilde{\mathcal{F}}_\lambda$ *of* $\mathcal{F}_\lambda$.

(ii) $(\cup_{k\geq 1}(\mathcal{F}_\lambda^c)_k, \mathcal{P}_\lambda^c)$ *is weakly convergent.*

Proof. "(a) ⇒ (b)" is obvious.

"(b) ⇒ (a)": Let $\{F_\lambda(k) = M_k \otimes \cdots \otimes M_1\}$ be a path in $\mathcal{P}_\lambda$. There are two cases to be discussed.

Case 1. There exist $A_i \in \mathcal{F}_\lambda$ and an index l_0 such that

$$A_i \in \{M_1, M_2, \ldots, M_{l_0-1}\} \text{ but } A_i \notin \{M_{l_0}, M_{l_0+1}, \ldots\}.$$

Let $\widetilde{\mathcal{F}}_\lambda = \mathcal{F}_\lambda \setminus \{A_i\}$. Consider the path

$$\{\widetilde{\mathcal{F}}_\lambda(k) | \widetilde{\mathcal{F}}_\lambda(1) = M_{l_0}, \ldots, \widetilde{\mathcal{F}}_\lambda(k) = M_{l_0+k-1} \otimes \cdots \otimes M_{l_0}, \ldots\}$$

in the system $(\cup_{k\geq 1}(\widetilde{\mathcal{F}}_\lambda)_k, \widetilde{\mathcal{P}}_\lambda)$. By assumption (i), the path $\{\widetilde{F}_\lambda(k)\}$ is convergent. Note that for k large enough

$$F_\lambda(k + l_0 - 1) = \widetilde{F}_\lambda(k) \otimes (M_{l_0-1} \otimes \cdots \otimes M_1).$$

Thus, by letting $k \to \infty$ and noting that $\widetilde{F}_\lambda(k)$ is convergent, the path $\{F_\lambda(k)\}$ is convergent.

Case 2. Suppose that each A_i in $\mathcal{F}_\lambda$ appears an infinite number of times in $\{M_1, M_2, \ldots\}$. Let $\{F_\lambda(n_k)\}$ and $\{F_\lambda(m_k)\}$ be two subpaths of $\{F_\lambda(k)\}$ which converge to F_1 and F_2, respectively. Construct two subpaths $\{F_\lambda(\hat{n}_k)\}$ and $\{F_\lambda(\hat{m}_k)\}$ of $\{F_\lambda(n_k)\}$ and $\{F_\lambda(m_k)\}$, respectively, such that

$$\begin{aligned}
F_\lambda(\hat{n}_1) &= M_{\hat{n}_1} \otimes \cdots \otimes M_1, \\
F_\lambda(\hat{m}_1) &= M_{\hat{m}_1} \otimes \cdots \otimes M_{\hat{n}_1+1} \otimes M_{\hat{n}_1} \otimes \cdots \otimes M_1, \\
F_\lambda(\hat{n}_2) &= M_{\hat{n}_2} \otimes \cdots \otimes M_{\hat{m}_1+1} \otimes M_{\hat{m}_1} \otimes \cdots \otimes M_1, \\
F_\lambda(\hat{m}_2) &= M_{\hat{m}_2} \otimes \cdots \otimes M_{\hat{n}_2+1} \otimes M_{\hat{n}_2} \otimes \cdots \otimes M_1 \\
&\vdots
\end{aligned}$$

where $M_{\hat{n}_1} \otimes \cdots \otimes M_1$, $M_{\hat{m}_1} \otimes \cdots \otimes M_{\hat{n}_1+1}$, $M_{\hat{n}_2} \otimes \cdots \otimes M_{\hat{m}_1+1}$, etc., are in $\mathcal{F}_\lambda^c$. Now we have constructed a path $\{F_\lambda(\hat{n}_1), F_\lambda(\hat{m}_1), F_\lambda(\hat{n}_2), F_\lambda(\hat{m}_2), \ldots\}$ in the system $(\cup_{k\geq 1}(\mathcal{F}_\lambda^c)_k, \mathcal{P}_\lambda^c)$. By the assumption (ii), the path is convergent, and hence, $F_1 = F_2$. By Lemma 3.5, we have that the path $\{F_\lambda(k)\}$ convergent. Since $\{F_\lambda(k)\}$ is an arbitrary path in $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$, the system $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ is weakly convergent. □

THEOREM 3.8. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *Then the fuzzy path system* $(\cup_{k\geq 1}(\mathcal{F})_k, \mathcal{P})$ *is weakly convergent if and only if*

(i) $(\cup_{k\geq 1}(\widetilde{\mathcal{F}})_k, \widetilde{\mathcal{P}})$ *is weakly convergent for all proper subset* $\widetilde{\mathcal{F}}$ *of* $\mathcal{F}$.

(ii) $(\cup_{k\geq 1}(\mathcal{F}^c)_k, \mathcal{P}^c)$ *is weakly convergent.*

Proof. It follows from Theorem 3.6 and Theorem 3.7. □

3.2. Results of strong convergence. We first note that, by Definition 2.2 and Lemma 3.4, if the fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is strongly convergent to C, then there exists a k_0 such that

$$\mathcal{F}_{k_0} = \mathcal{F}_{k_0+j} = \{C\} \ \forall j \geq 1.$$

THEOREM 3.9. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *Then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is strongly convergent if and only if for each* $\lambda \in \Phi_{\mathcal{F}}$, *the Boolean path system* $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ *is strongly convergent, where* $\mathcal{P}_\lambda$ *denotes the set of all paths in* $\cup_{k\geq 1}(\mathcal{F}_\lambda)_k$.

Proof. The proof is similar to that of Theorem 3.6. □

THEOREM 3.10. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *Then the following statements are equivalent.*

(i) *The fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is strongly convergent to zero.*
(ii) *The Boolean path systems* $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ *are strongly convergent to zero for all* $\lambda \in \Phi_{\mathcal{F}}$.
(iii) *The Boolean path system* $(\cup_{k\geq 1}(\mathcal{F}_{\underline{\lambda}})_k, \mathcal{P}_{\underline{\lambda}})$ *is strongly convergent to zero.*
(iv) $\lim_{k\to\infty}[\max_{M\in(\mathcal{F}_{\underline{\lambda}})_k} \rho(M)] = 0$.

Proof. (i)⇒ (ii) is similar to the proof in Theorem 3.6. (ii) ⇒ (iii) is obvious. We prove now (iii) ⇒ (iv). Since the system $(\cup_{k\geq 1}(\mathcal{F}_{\underline{\lambda}})_k, \mathcal{P}_{\underline{\lambda}})$ is strongly convergent to zero, there exists N_0 such that for all paths $\{F_{\underline{\lambda}}(k)\} \in \mathcal{P}_{\underline{\lambda}}$,

$$0 = F_{\underline{\lambda}}(N_0) = F_{\underline{\lambda}}(N_0 + j) \text{ for } j = 1, 2, \ldots.$$

Thus, $\rho(M) = 0$ for $M \in (\mathcal{F}_{\underline{\lambda}})_{N_0+j}$ $j = 1, 2, \ldots$. Therefore, $\lim_{k\to\infty}[\max_{M\in(\mathcal{F}_{\underline{\lambda}})_k} \rho(M)] = 0$.

(iv) ⇒ (i). Consider a path $\{F(k)\} \in \mathcal{P}$. Let $F(k) = A_k \otimes \cdots \otimes A_2 \otimes A_1$ with $A_i \in \mathcal{F}$. For each $\lambda \in \Phi_{\mathcal{F}}$, let

$$F_\lambda(k) = (A_k)_\lambda \otimes \cdots \otimes (A_2)_\lambda \otimes (A_1)_\lambda = (F(k))_\lambda \text{ for } k = 1, 2, \ldots.$$

Claim. Path $\{F_{\underline{\lambda}}(k)\}$ in $\mathcal{P}_{\underline{\lambda}}$ is strongly convergent to zero.
Assume to the contrary that for N large enough

$$F_{\underline{\lambda}}(N) = (A_N)_{\underline{\lambda}} \otimes \cdots \otimes (A_2)_{\underline{\lambda}} \otimes (A_1)_{\underline{\lambda}} \neq 0.$$

There exist i, j such that

$$[(A_N)_{\underline{\lambda}} \otimes \cdots \otimes (A_2)_{\underline{\lambda}} \otimes (A_1)_{\underline{\lambda}}]_{ij} = 1.$$

This implies that

$$[(A_N)_{\underline{\lambda}}]_{ik_N} = \cdots = [(A_2)_{\underline{\lambda}}]_{k_2k_1} = [(A_1)_{\underline{\lambda}}]_{k_1j} = 1$$

for some indices $k_1, k_2, \ldots, k_{N-1}$. Since k_is are in between 1 and n and N is large enough, there must be at least two equal indices k_l and k_m with $1 < l < m < N$ such that

$$[D]_{k_lk_l} = 1, \text{ where } D = (A_m)_{\underline{\lambda}} \otimes \cdots \otimes (A_{l+1})_{\underline{\lambda}}.$$

Define $B = [b_{ij}]$, where

$$b_{ij} = \begin{cases} 1 & \text{if } i = j = k_l, \\ 0 & \text{otherwise.} \end{cases}$$

Then we have $B \leq D$. Since $\rho(B) = 1$, we have $\rho(D) = 1$. Moreover, we have $\rho(D^k) = 1$ for $k = 1, 2, \ldots$. Now, since $D \in (\mathcal{F}_{\underline{\lambda}})_{m-l-1}$, we have

$$\max_{M\in(\mathcal{F}_{\underline{\lambda}})_{(m-l-1)k}} \rho(M) = 1 \ \forall\, k = 1, 2, \ldots.$$

This implies that

$$\lim_{k\to\infty}\left[\max_{M\in(\mathcal{F}_{\underline{\lambda}})_k} \rho(M)\right] \neq 0, \text{ a contradiction.}$$

This completes the proof of the claim. To finish the proof, we need to show that any other path converges strongly to zero as well. Precisely, let $\lambda \in \Phi_\lambda$. By the definitions of cut matrix and $\underline{\lambda}$, we have

$$F_\lambda(k) \leq F_{\underline{\lambda}}(k) \text{ for } k = 1, 2, \ldots.$$

Therefore, $F_\lambda(N + j) = F_\lambda(N) = 0$ for $j = 1, 2, \ldots$. Since

$$F(k) = \oplus_{\lambda \in \Phi_\lambda}\{\lambda \otimes F_\lambda(k)\},$$

we have

$$F(N + j) = F(N) = 0 \text{ for } j = 1, 2, \ldots.$$

This proves that any path $\{F(k)\}$ is strongly convergent to zero. Hence, the system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is strongly convergent to zero. □

4. Sufficient conditions for the weak convergence. In this section, we shall present several sufficient conditions for the weak convergence. Two kinds of strategies employed in those sufficient conditions for the weak convergence are considered. One strategy is to construct monotonicity for each path, where all the elements of products in each path are either nondecreasing or nonincreasing. The other strategy is to define the notion of row domination for $\mathcal{F}$, which enables the off-diagonal elements of infinite products in each path are nondecreasing, yet the diagonal elements are nonincreasing.

From the literature discussing the convergence of powers of a fuzzy matrix, we learn that the monotonicity involving two consecutive powers (such as compactness, $A \leq A^2$, transitivity, $A^2 \leq A$, or $A^2 \leq A^3$ in [3]) plays a key role in establishing the convergence. In [5], Guu, Chen, and Pang kept this spirit to generalize the concepts of compactness and transitivity to $\mathcal{F}$, where $\mathcal{F}$ contains a finite number of fuzzy matrices. In this section, one way to provide sufficient conditions for the weak convergence will be the assumption of a certain monotonicity for $\mathcal{F}$. Precisely, we shall generalize the idea of $A^2 \leq A^3$, to which we now turn.

DEFINITION 4.1. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *The* $\mathcal{F}$ *is second order increasing if for any* A, B, G *in* $\mathcal{F}$

$$A \otimes B \leq G \otimes (A \otimes B). \tag{3}$$

For instance, consider the 8 *inequalities involved in* (3) *for the fuzzy matrix set* $\mathcal{F} = \{A, B\}$:

$$\begin{array}{rcl@{\qquad}rcl} A\otimes B & \leq & A\otimes(A\otimes B) & & A\otimes A & \leq & A\otimes(A\otimes A) \\ A\otimes B & \leq & B\otimes(A\otimes B) & \textit{and} & A\otimes A & \leq & B\otimes(A\otimes A) \\ B\otimes A & \leq & A\otimes(B\otimes A) & & B\otimes B & \leq & B\otimes(B\otimes B) \\ B\otimes A & \leq & B\otimes(B\otimes A) & & B\otimes B & \leq & A\otimes(B\otimes B) \end{array} . \tag{4}$$

We note that when $A = B$, (4) *becomes* $A^2 \leq A^3$. *We are ready to present the following theorem.*

THEOREM 4.2. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If* $\mathcal{F}$ *is second order increasing, then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent.*

Proof. Consider a path $\{F(k)\}$ in $\mathcal{P}$. Since the underlying set $\mathcal{F}$ is second order increasing, for large k, we have

$$F(k) = (A_k \otimes A_{k-1}) \otimes F(k-2) \leq A_{k+1} \otimes (A_k \otimes A_{k-1}) \otimes F(k-2) = F(k+1).$$

The sequence $\{F(k)\}$ is monotone for $k \geq 3$. Hence, the path $\{F(k)\}$ is convergent. This implies that the fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is weakly convergent. □

Example 4.1. Consider the underlying fuzzy matrix set $\mathcal{F} = \{A, B\}$, where

$$A = \begin{bmatrix} 0.5 & 1 & 0 \\ 0 & 0.6 & 0.3 \\ 0.2 & 0 & 0.7 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0.5 & 0.8 & 0.1 \\ 0.3 & 0.6 & 0.2 \\ 0.2 & 0.5 & 0.7 \end{bmatrix}.$$

Direct computation shows that

$$A \otimes A = A \otimes A \otimes A = \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.2 & 0.6 & 0.3 \\ 0.2 & 0.2 & 0.7 \end{bmatrix}.$$

Comparing A and $A \otimes A$ shows that the monotonicity starts from the second power. Similarly

$$B \otimes B = B \otimes B \otimes B = \begin{bmatrix} 0.5 & 0.6 & 0.2 \\ 0.3 & 0.6 & 0.2 \\ 0.3 & 0.5 & 0.7 \end{bmatrix}.$$

Since

$$A \otimes B = \begin{bmatrix} 0.5 & 0.6 & 0.2 \\ 0.3 & 0.6 & 0.3 \\ 0.2 & 0.5 & 0.7 \end{bmatrix},$$

we have

$$A \otimes (A \otimes B) = \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.2 & 0.5 & 0.7 \end{bmatrix} \text{ and } B \otimes (A \otimes B) \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.3 & 0.5 & 0.7 \end{bmatrix}.$$

Both $A \otimes (A \otimes B)$ and $B \otimes (A \otimes B)$ are greater than $A \otimes B$. Similarly, since

$$B \otimes A = \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.2 & 0.5 & 0.7 \end{bmatrix},$$

we have

$$A \otimes (B \otimes A) = \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.2 & 0.5 & 0.7 \end{bmatrix} \text{ and } B \otimes (B \otimes A) = \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.3 & 0.5 & 0.7 \end{bmatrix}.$$

Both $A \otimes (B \otimes A)$ and $B \otimes (B \otimes A)$ are greater than $B \otimes A$. Computation shows that

$$A \otimes (B \otimes B) = \begin{bmatrix} 0.5 & 0.6 & 0.2 \\ 0.3 & 0.6 & 0.3 \\ 0.3 & 0.5 & 0.7 \end{bmatrix} \text{ and } B \otimes (A \otimes A) \begin{bmatrix} 0.5 & 0.6 & 0.3 \\ 0.3 & 0.6 & 0.3 \\ 0.2 & 0.5 & 0.7 \end{bmatrix}.$$

We have $A \otimes (B \otimes B) \geq (B \otimes B)$ and $B \otimes (A \otimes A) \geq (A \otimes A)$. Thus, the 8 inequalities in (4) hold for $\mathcal{F}$. We have that $\mathcal{F}$ is second order increasing and the fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is weakly convergent.

Remark. In Definition 4.1, we have defined the second order increasing property of $\mathcal{F}$. The main purpose is of course to have monotonicity of each path in $\mathcal{P}$. Indeed, we can define the *second order decreasing* of $\mathcal{F}$ to have weak convergence. This can be done by reversing the direction of each inequality in (3). On the other hand, one can define the kth order increasing (decreasing) of $\mathcal{F}$ to have weak convergence by involving more matrices in (3). For instance, one can define the third order increasing property for $\mathcal{F}$ if for any A, B, G, H in $\mathcal{F}$,

$$A \otimes B \otimes G \leq H \otimes (A \otimes B \otimes G).$$

Guu, Chen, and Pang generalized the compactness and transitivity of a fuzzy matrix to a finite number of fuzzy matrices. For our settings here, we say $\mathcal{F}$ is *transitive (compact)* if $A \otimes B \leq B$ ($A \otimes B \geq B$) for all A, B in $\mathcal{F}$. Hence, in the sense of Definition 4.1, the compactness and transitivity correspond to the first order increasing and the first order decreasing properties of $\mathcal{F}$, respectively.

THEOREM 4.3. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If* $\mathcal{F}_\lambda$ *is compact for each* $\lambda \in \Phi_\mathcal{F}$, *then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent.*

Proof. Let $\{F_\lambda(k)\}$ denote a path in $\mathcal{P}_\lambda$ for each $\lambda \in \Phi_\mathcal{F}$. Since compactness of $\mathcal{F}_\lambda$ implies that the Boolean path system $(\cup_{k\geq 1}(\mathcal{F}_\lambda)_k, \mathcal{P}_\lambda)$ converges weakly, we have by Theorem 3.1 that the fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is weakly convergent. □

THEOREM 4.4. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If* $\mathcal{F}_\lambda$ *is transitive for each* $\lambda \in \Phi_\lambda$, *then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent.*

Proof. The proof is similar to that of Theorem 4.3. □

From above, it is easy to see that monotonicity is a useful mechanism to establish the weak convergence. In the following theorem, we present a sufficient condition for the weak convergence but do not count on monotonicity.

DEFINITION 4.5. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *Define* $S = \max\{A | A \in \mathcal{F}\}$ *and* $D = \min\{A | A \in \mathcal{F}\}$, *where operations* max *and* min *are implemented elementwise.* $\mathcal{F}$ *is said to be* row dominated *if for each* $i \neq j$, $S_{ij} \leq D_{ii}$.

THEOREM 4.6. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *If* $\mathcal{F}$ *is row dominated, then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent.*

Proof. Let $\{F(k+1) = A_{k+1} \otimes A_k \otimes \cdots A_2 \otimes A_1\}$ be a path in $\mathcal{P}$. For $i \neq j$, we have

$$\begin{aligned} F(k+1)_{ij} &= (A_{k+1} \otimes A_k \otimes \cdots A_2 \otimes A_1)_{ij} \\ &= \oplus_{l=1}^{n} (A_{k+1})_{il} \otimes (A_k \otimes \cdots A_2 \otimes A_1)_{lj} \\ &\geq (A_{k+1})_{ii} \otimes (A_k \otimes \cdots A_2 \otimes A_1)_{ij}. \end{aligned}$$

Let $(A_k \otimes \cdots A_2 \otimes A_1)_{ij} = (A_k)_{il_1} \otimes \cdots \otimes (A_2)_{l_{k-2}l_{k-1}} \otimes (A_1)_{l_{k-1}j}$ for some $l_1, \ldots, l_{k-1}$. Since $i \neq j$, by the assumption of $S_{ij} \leq D_{ii}$ we can either select $t^* = \min\{x | l_x \neq i\}$ such that

$$(A_{k+1})_{ii} \geq (A_{k+1-t^*})_{il_{t^*}},$$

or when t^* is not well defined, we have $(A_{k+1})_{ii} \geq (A_1)_{ij}$. Thus,

$$(A_{k+1})_{ii} \otimes (A_k)_{il_1} \otimes \cdots \otimes (A_2)_{l_{k-2}l_{k-1}} \otimes (A_1)_{l_{k-1}j} \geq (A_k)_{il_1} \otimes \cdots \otimes (A_2)_{l_{k-2}l_{k-1}} \otimes (A_1)_{l_{k-1}j}.$$

Therefore, $F(k+1)_{ij} \geq F(k)_{ij}$. This implies that $\lim_{k\to\infty}(F(k))_{ij}$ converges.

For $i = j$, we let

$$F(k+1)_{ii} = (A_{k+1})_{il_1} \otimes (A_k)_{l_1 l_2} \otimes \cdots \otimes (A_2)_{l_{k-1}l_k} \otimes (A_1)_{l_k i}$$

for some $l_1, \ldots, l_k$. If $l_1 = i$, then we have

$$F(k+1)_{ii} = (A_{k+1})_{ii} \otimes (A_k)_{il_2} \otimes \cdots \otimes (A_2)_{l_{k-1}l_k} \otimes (A_1)_{l_k i} \leq (A_{k+1})_{ii} \otimes F(k)_{ii} \leq F(k)_{ii}.$$

If $l_1 \neq i$, then by assumption we have $(A_{k+1})_{il_1} \leq (A_t)_{ii}$ for all $t = 1, 2, \ldots, k+1$. We then have

$$F(k+1)_{ii} \leq (A_{k+1})_{il_1} \leq (A_k)_{ii} \otimes (A_{k-1})_{ii} \otimes \cdots (A_1)_{ii} \leq F(k)_{ii}.$$

This implies that $\lim_{k\to\infty}(F(k))_{ii}$ converges. The proof is completed. □

COROLLARY 4.7. *Let the set of underlying fuzzy matrices* $\mathcal{F} = \{A^{(1)}, A^{(2)}, \ldots, A^{(m)}\} \subset \mathbb{F}^{n\times n}$. *Consider* $D = \min\{A | A \in \mathcal{F}\}$. *If* $D_{ii} = 1$ *for all* $i = 1, 2, \ldots, n$, *then the fuzzy path system* $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ *is weakly convergent.*

Remark. We have shown that if $\mathcal{F}$ is row dominated, then the fuzzy path system $(\cup_{k\geq 1}\mathcal{F}_k, \mathcal{P})$ is weakly convergent. From the proof, one can see that the off-diagonal elements $F(k)_{ij}$ are increasing, while the diagonal elements $F(k)_{ii}$ are decreasing. Furthermore, if $\mathcal{F}$ contains only one matrix A, then by the fact that the diagonal elements of the powers of A are increasing, the diagonal elements remain constant in the powers of A. Since the off-diagonal elements of the powers are increasing, we have that the row domination implies the compactness of A. Example 4.2 illustrates these facts.

Example 4.2. Consider $\mathcal{F} = \{A, B\}$, where

$$A = \begin{bmatrix} 0.5 & 0.4 & 0.3 \\ 0 & 0.6 & 0 \\ 0.4 & 0.3 & 0.5 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0.4 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0 & 0.2 & 0.6 \end{bmatrix}.$$

$\mathcal{F}$ is row dominated for

$$S = \begin{bmatrix} 0.5 & 0.4 & 0.3 \\ 0.2 & 0.6 & 0.3 \\ 0.4 & 0.3 & 0.6 \end{bmatrix} \text{ and } D = \begin{bmatrix} 0.4 & 0.2 & 0.1 \\ 0 & 0.5 & 0 \\ 0 & 0.2 & 0.5 \end{bmatrix}.$$

We note that

$$A \otimes B = \begin{bmatrix} 0.4 & 0.4 & 0.3 \\ 0.2 & 0.5 & 0.3 \\ 0.4 & 0.3 & 0.5 \end{bmatrix}.$$

One can see that the off-diagonal elements (comparing with A or B) are increasing, yet the diagonal elements (comparing with A or B) are decreasing.

If $\mathcal{F} = \{A\}$, then direct computation shows that we have $A \leq A \otimes A = A \otimes A \otimes A$. Furthermore, the diagonal elements in the powers of A, $A \otimes A$, $A \otimes A \otimes A$ remain unchanged. □

5. Conclusions. Unlike the convergence of infinite products of a finite number of matrices that has been studied quite extensively for several decades, the same issue in the context of fuzzy matrices seems to be ignored. This paper concentrates on the convergent aspects of infinite products of a finite number of fuzzy matrices, which arise from the field of time-invariant discrete-time fuzzy systems with nonfuzzy inputs, by exploring the fuzzy path system generated by these underlying fuzzy matrices.

Three main directions have been presented. First, similar to Fan and Liu's decomposition, we established the weak convergence of infinite products of a finite number of fuzzy matrices in terms of certain (finite) Boolean matrices. Equivalence in strong convergence of infinite products of fuzzy matrices and infinite products of the associated Boolean matrices were established as well. Further characterizations for infinite products of fuzzy matrices to converge strongly to zero were given. Second, monotone properties such as compactness and transitivity are useful to establish the convergence of powers of a fuzzy matrix. By following the strategy of monotonicity, we constructed suitable monotone conditions as sufficient conditions for the weak convergence. Under the monotonicity, all the elements of infinite products are either nondecreasing or nonincreasing. Third, a sufficient condition based on the concept of row domination was proposed to establish the weak convergence. Under this sufficient condition, we showed that the off-diagonal elements of infinite products are increasing, yet the diagonal elements are decreasing.

REFERENCES

[1] I. Daubechies and J. C. Lagarias, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 161 (1992), pp. 227–263.

[2] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, San Diego, 1980.

[3] Z.-T. Fan and D.-F. Liu, *Convergence of the power sequence of a nearly monotone increasing fuzzy matrix*, Fuzzy Sets and Systems, 88 (1997), pp. 363–372.

[4] Z.-T. Fan and D.-F. Liu, *On the oscillating power sequence of a fuzzy matrix*, Fuzzy Sets and Systems, 93 (1998), pp. 75–85.

[5] S.-M. Guu, H.-H. Chen, and C.-T. Pang, *Convergence of products of fuzzy matrices*, Fuzzy Sets and Systems, to appear.

[6] D. J. Hartfiel, *On infinite products of nonnegative matrices*, SIAM J. Appl. Math, 26 (1974), pp. 297–301.

[7] H. Hashimoto, *Convergence of powers of a fuzzy transitive matrix*, Fuzzy Sets and Systems, 9 (1983), pp. 153–160.

[8] K. H. Kim, *Boolean Matrix Theory and Applications*, Marcel Dekker, New York, 1982.

[9] K. H. Kim and F. W. Roush, *Fuzzy matrix theory*, in Analysis of Fuzzy Information, Vol. 1, J. C. Bezdek, ed., CRC Press, Boca Raton, FL, 1987, pp. 107–129.

[10] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[11] W. Kolodziejczyk, *Convergence of powers of s-transitive fuzzy matrices*, Fuzzy Sets and Systems, 26 (1988), pp. 127–130.

[12] J.-X. Li, *Periodicity of powers of fuzzy matrices*, Fuzzy Sets and Systems, 48 (1992), pp. 365–369.

[13] J.-X. Li, *Convergence of powers of controllable fuzzy matrices*, Fuzzy Sets and Systems, 62 (1994), pp. 83–88.

[14] M. Mizumoto, *Fuzzy Theory and Its Applications*, Science Publications, 1988 (in Japanese).

[15] S. V. Ovchinnikov, *Structure of fuzzy relations*, Fuzzy Sets and Systems, 6 (1981), pp. 169–195.
[16] F. Robert, *Discrete Iterations: A Metric Study*, Springer-Verlag, Berlin, 1986.
[17] D. Rosenblatt, *On the graphs of finite idempotent Boolean relation matrices*, J. Res. Nat. Bur. Standards B, 67B (1963), pp. 249–259.
[18] B. De Schutter and B. De Moor, *On the sequence of consecutive powers of a matrix in a Boolean algebra*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 328–354.
[19] M. G. Thomason, *Convergence of powers of a fuzzy matrix*, J. Math. Anal. Appl., 57 (1977), pp. 476–480.
[20] L. A. Zadeh, *Similarity relations and fuzzy orderings*, Inform. Sci., 3 (1971), pp. 177–200.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1204–1221

CHOOSING REGULARIZATION PARAMETERS IN ITERATIVE METHODS FOR ILL-POSED PROBLEMS*

MISHA E. KILMER† AND DIANNE P. O'LEARY‡

Abstract. Numerical solution of ill-posed problems is often accomplished by discretization (projection onto a finite dimensional subspace) followed by regularization. If the discrete problem has high dimension, though, typically we compute an approximate solution by projecting the discrete problem onto an even smaller dimensional space, via iterative methods based on Krylov subspaces. In this work we present a common framework for efficient algorithms that regularize after this second projection rather than before it. We show that determining regularization parameters based on the final projected problem rather than on the original discretization has firmer justification and often involves less computational expense. We prove some results on the approximate equivalence of this approach to other forms of regularization, and we present numerical examples.

Key words. ill-posed problems, regularization, discrepancy principle, iterative methods, L-curve, Tikhonov, truncated singular value decomposition, projection, Krylov subspace

AMS subject classifications. 65F10, 65F22

PII. S0895479899345960

1. Introduction.

Linear, discrete ill-posed problems of the form

$$\min_x \|Ax - b\|_2 \tag{1.1}$$

arise, for example, from the discretization of first-kind Fredholm integral equations and occur in a variety of applications. We shall assume that the full-rank matrix A is $m \times n$ with $m \geq n$. In discrete ill-posed problems, A is ill-conditioned and there is often no gap in the singular value spectrum. Typically, the right-hand side b contains noise due to measurement and/or approximation error. This noise, in combination with the ill-conditioning of A, means that the exact solution of (1.1) has little relationship to the noise-free solution and is worthless.

Instead, we use a *regularization* method to determine a solution that approximates the noise-free solution. We replace the original operator by a better conditioned but related one in order to diminish the effects of noise in the data. Sometimes this regularized problem is too large to solve exactly. In that case, we typically project the problem onto an even smaller dimensional space, perhaps via iterative methods based on Krylov subspaces. Sometimes this projection provides enough regularization to produce a good approximate solution, but often (see, for example, [28, 15]) additional regularization is needed.

A fundamental decision to be made in such cases is whether to regularize before or after the projection. *One subtle issue is that the regularization parameter that is optimal for the discretized problem may not be optimal for the lower-dimensional problem actually solved by the iteration*, and this leads to the research discussed in this paper.

*Received by the editors December 3, 1999; accepted for publication (in revised form) by P. C. Hansen November 13, 2000; published electronically April 6, 2001. This work was supported by the National Science Foundation under grants CCR 95-03126 and CCR 97-32022 and by the Army Research Office, MURI grant DAAG55-97-1-0013.

http://www.siam.org/journals/simax/22-4/34596.html

†Department of Mathematics, Tufts University, Medford, MA 02155 (mkilme01@tufts.edu).

‡Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (oleary@cs.umd.edu).

At first glance, there can appear to be a lot of work associated with the selection of a good regularization parameter, and many algorithms proposed in the literature are needlessly complicated, repeating a Krylov iteration multiple times. By regularizing after projection by the iterative method, so that we are regularizing the lower dimensional problem that is actually being solved, this difficulty vanishes.

The purpose of this paper is to present a common framework for parameter selection techniques applied to the problem resulting from iterative methods such as Krylov subspace techniques. We show that by determining regularization parameters based on the final projected problem rather than on the original discretization, we can better approximate the optimal parameter and reduce the cost of solution.

Our paper is organized as follows. In section 2 we survey some methods for choosing the corresponding regularization parameters. In section 3, we show how any standard parameter selection technique for the original problem can be applied instead to a projected problem obtained from an iterative method, greatly reducing the cost without much degradation in the solution. We give experimental results in section 4 and conclusions in section 5.

In the following we shall assume that $b = b_{true} + e$, where b_{true} denotes the unperturbed data vector and e denotes zero-mean white noise. We will also assume that b_{true} satisfies the discrete Picard condition; that is, the spectral coefficients of b_{true} decay faster, on average, than the singular values.

Let $\hat{U}\Sigma\hat{V}^*$ denote the singular value decomposition (SVD) of A, where the columns of $\hat{U}$ and $\hat{V}$ are the singular vectors, and the singular values are ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Then the solution (1.1) is given by

$$x = \sum_{i=1}^{n} \frac{\hat{u}_i^* b}{\sigma_i} \hat{v}_i = \sum_{i=1}^{n} \left(\frac{\hat{u}_i^* b_{true}}{\sigma_i} + \frac{\hat{u}_i^* e}{\sigma_i} \right) \hat{v}_i. \tag{1.2}$$

As a consequence of the white noise assumption, $|\hat{u}_i^* e|$ is roughly constant for all i, while the discrete Picard condition guarantees that $|\hat{u}_i^* b_{true}|$ decreases with i faster than σ_i does. The matrix A is ill-conditioned, so small singular values magnify the corresponding coefficients $\hat{u}_i^* e$ in the second sum, and it is this large contribution of noise that renders the exact solution x defined in (1.2) worthless. The following four classes of regularization methods try in different ways to lessen the contribution of noise. For further information on these methods, see, for example, [19, 15].

In **Tikhonov regularization**, (1.1) is replaced by

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2, \tag{1.3}$$

where λ is a positive scalar regularization parameter, and we choose L to be the identity matrix I. Solving (1.3) is equivalent to solving

$$(A^*A + \lambda^2 I)x_\lambda = A^*b. \tag{1.4}$$

In analogy with (1.2) we have

$$x_\lambda = \sum_{i=1}^{n} \left(\frac{\sigma_i \, \hat{u}_i^* b_{true}}{\sigma_i^2 + \lambda^2} + \frac{\sigma_i \, \hat{u}_i^* e}{\sigma_i^2 + \lambda^2} \right) \hat{v}_i. \tag{1.5}$$

In **truncated SVD** we compute the regularized solution by truncating the expansion in (1.2) as

$$x_\ell = \sum_{i=1}^{\ell} \frac{\hat{u}_i^* b}{\sigma_i} \hat{v}_i. \tag{1.6}$$

Here the regularization parameter is ℓ, the number of terms retained in the sum. Rust [33] introduced a related truncation strategy, including in the sum (1.2) only those terms corresponding to a spectral coefficient $\hat{u}_i^* b$ whose magnitude is greater than or equal to some tolerance ρ, which can be regarded as the regularization parameter.

Solving (1.4) or (1.6) can be impractical if n is large, but fortunately, regularization can be achieved through **projection** onto a k-dimensional subspace; see, for example, [9]. The truncated SVD (TSVD) is one example, but projection is often achieved through the use of iterative methods such as conjugate gradients, GMRES, QMR, and other Krylov subspace methods [28, 1]. Krylov subspace algorithms tend to produce, at early iterations, solutions that resemble x_{true} more than later iterates. Therefore, the choice of the regularization parameter k, the stopping point for the iteration and the dimension of the subspace, is very important.

Another important family of regularization methods, termed **hybrid methods** [19, 15], was introduced by O'Leary and Simmons [28]. These methods combine a projection method with a direct regularization method such as TSVD or Tikhonov regularization. Since the dimension k is usually small relative to n, regularization of the restricted problem is much less expensive, but the end results can be very similar to those achieved by applying the same direct regularization technique to the original problem; see section 3.5.

2. Existing parameter selection methods. In this section, we discuss three parameter selection techniques that have been proposed in the literature. They differ in the amount of a priori information required as well as in the decision criteria.

The discrepancy principle [26] says that if δ is the expected value of $\|e\|_2$, then the regularization parameter should be chosen so that the norm of the residual corresponding to the regularized solution x_{reg} is $\tau\delta$; that is,

$$\|Ax_{reg} - b\|_2 = \tau\delta, \tag{2.1}$$

where $\tau > 1$ is some predetermined real number. Note that as $\delta \to 0$, $x_{reg} \to x_{true}$. Other methods based on knowledge of the variance are given, for example, in [3, 13, 7].

Generalized cross-validation (GCV) [11] does not depend on a priori knowledge about the noise variance. We find the parameter λ that minimizes the GCV functional

$$G(\lambda) = \frac{\|(I - AA_\lambda^\sharp)b\|_2^2}{(\mathrm{trace}(I - AA_\lambda^\sharp))^2}, \tag{2.2}$$

where $A_\lambda^\sharp$ denotes the matrix that maps the right-hand side b onto the regularized solution x_λ. In Tikhonov regularization, for example, $A_\lambda^\sharp$ is $(A^*A + \lambda^2 I)^{-1}A^*$.

The L-curve, the plot of the norm of the regularized solution versus the corresponding residual norm for each of a set of regularization parameter values, was introduced by Lawson and popularized by Hansen [17, 25]. Intuitively, the best regularization parameter should lie on the corner of the L-curve, since for values higher than this, the residual increases without reducing the norm of the solution much, while for values smaller than this, the norm of the solution increases rapidly without much decrease in residual. In practice, only a few points on the L-curve are computed and the corner is located by estimating the point of maximum curvature [20].

The appropriate choice of regularization parameter—especially for projection algorithms—is a difficult problem, and each method has severe flaws. The discrepancy principle is convergent as the noise goes to zero, but it relies on information that

TABLE 2.1

Summary of additional flops needed to compute the regularization parameter for each of four regularization methods with various parameter selection techniques. Notation: q is the cost of multiplication of a vector by A; p is the number of discrete parameters that must be tried; k is the dimension of the projection; m and n are problem dimensions.

	Basic cost	Added cost		
		Disc.	GCV	L-curve
Tikhonov	$O(mn^2)$	$O(p(m+n))$	$O(p(n+m))$	$O(p(m+n))$
TSVD	$O(mn^2)$	$O(m)$	$O(m)$	$O(m+n)$
Rust's TSVD	$O(mn^2)$	$O(m \log m)$	$O(m \log m)$	$O(m \log m)$
Projection	$O(qk)$	0	$O(q)$	$O(q)$

is often unavailable or erroneous. Even with a correct estimate of the variance, the solutions tend to be oversmoothed [21, p. 96]. (See also the discussion in section 6.1 of [17].) One noted difficulty with GCV is that G can have a very flat minimum, making it difficult to determine the optimal λ numerically [37]. The L-curve is usually more tractable numerically, but its limiting properties are nonideal. The solution estimates fail to converge to the true solution as $n \to \infty$ [38] or as the error norm goes to zero [8]. All methods that assume no knowledge of the error norm— including GCV—have this latter property [8].

For further discussion and references about parameter choice methods, see [7, 19]. The cost of these methods is tabulated in Table 2.1.

2.1. Previous work on parameter choice for hybrid methods. At first glance, it appears that for Tikhonov regularization, multiple systems of the form (1.4) must be solved in order to evaluate candidate values of λ for the discrepancy principle or the L-curve.

Chan and Ng [5] note that the systems involve matrices $C(\lambda) = A^*A + \lambda I$, which they solve using a Galerkin projection method on a sequence of "seed" systems. Although economical in storage, this is unnecessarily expensive in time because they do not exploit the fact that for each fixed k, the Krylov subspace $\mathcal{K}_k(A^*b, C(\lambda))$ is the same for all values of λ.

Frommer and Maass [10] propose two algorithms for approximating the λ that satisfies the discrepancy principle (2.1). The first is a "truncated conjugate gradient (CG)" approach, solving k systems of the form (1.4), truncating the iterative process early for large λ, and using previous solutions as starting guesses for later problems. Like Chan and Ng, this algorithm does not exploit the redundant Krylov subspaces. In the second method, however, they update the CG iterates for all k systems simultaneously, stopping their "shifted CG" algorithm when $\|Ax_\lambda - b\|_2 \leq \tau\delta$ for one of their λ values. The methods we propose in section 3 will usually require less work than the shifted CG algorithm because of less overhead.

Calvetti, Golub, and Reichel [4] use upper and lower bounds on the L-curve, generated by the matrices $C(\lambda)$ using a Lanczos bidiagonalization process, to approximate the best parameter for Tikhonov regularization before projection.

Kaufman and Neumaier [22] suggest an envelope guided conjugate gradient approach for the Tikhonov L-curve problem. Their method is necessarily somewhat more expensive than ours because they maintain nonnegativity constraints on the variables.

Substantial work has also been done on TSVD regularization of the projected problems. Björck, Grimme, and van Dooren [2] use GCV to determine the truncation point for the projected SVD. Their emphasis is on maintaining an accurate factor-

ization when many iterations are needed, using full reorthogonalization and implicit restart strategies. O'Leary and Simmons [28] take the viewpoint that the problem should be preconditioned appropriately so that a massive number of iterations is unnecessary. That viewpoint is echoed in this current work, so we implicitly assume that the problem has been preconditioned [28] so that $A = M^{-1}\hat{A}$ and $b = M^{-1}\hat{b}$, where $\hat{A}$ and $\hat{b}$ are the original data and M is a preconditioning matrix. See [16, 27, 24, 23] for preconditioners appropriate for certain types of ill-posed problems.

3. Regularizing the projected problem. In this section we categorize a dozen approaches to regularization of the projected problem that arise from using Krylov methods, giving enough detail to make the costs apparent and to show that the ideas are easy to program. Many Krylov methods have been proposed; for ease of exposition we focus on just one of these: the LSQR algorithm of Paige and Saunders [30].

LSQR iteratively computes a bidiagonalization related to that introduced by Golub and Kahan [12]. After k iterations, it has effectively computed three matrices: an upper-bidiagonal matrix B_k and two matrices $U_k \equiv [u_1, \ldots, u_k]$ and $V_k \equiv [v_1, \ldots, v_k]$, with orthonormal columns, related by

$$b = \beta_1 u_1 = \beta_1 U_{k+1} e_1 \,, \tag{3.1}$$

$$AV_k = U_{k+1} B_k \,, \tag{3.2}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T \,, \tag{3.3}$$

where e_i denotes the ith unit vector.

In numeric computations, the columns of U_k and V_k can fail to be orthonormal. This has never given us convergence difficulties, but if it becomes troublesome, there are well-known techniques to handle it [31, 32, 36, 6].

Now suppose we want to solve

$$\min_{x \in \mathcal{S}} \|b - Ax\|_2, \tag{3.4}$$

where $\mathcal{S}$ denotes the k-dimensional subspace spanned by the first k vectors v_i. The solution we seek is of the form $x^{(k)} = V_k y^{(k)}$ for some vector $y^{(k)}$ of length k. Define $r^{(k)} = b - Ax^{(k)}$ to be the corresponding residual and observe that

$$\begin{aligned} r^{(k)} &= \beta_1 u_1 - AV_k y^{(k)} \\ &= U_{k+1}(\beta_1 e_1 - B_k y^{(k)}). \end{aligned}$$

Since U_{k+1} has, in exact arithmetic, orthonormal columns, the projected problem we wish to solve is

$$\min_{y^{(k)}} \|\beta_1 e_1 - B_k y^{(k)}\|_2. \tag{3.5}$$

Solving this minimization problem is mathematically equivalent to solving the normal equations involving the bidiagonal matrix

$$B_k^* B_k y^{(k)} = \beta_1 B_k^* e_1, \tag{3.6}$$

although more stable means are used in practice. Typically k is small, so reorthogonalization to combat round-off error might or might not be necessary. The matrix B_k may be ill-conditioned because some of its singular values approximate some of the small singular values of A. Therefore, solving the projected problem might not yield

TABLE 3.1

Summary of flops for projection plus inner regularization with various parameter selection techniques, in addition to the O(qk) flops required for projection itself. Here k is the number of iterations (i.e., the size of the projection) taken and p is the number of discrete parameters that must be tried.

Projection plus –	Disc.	GCV	L-curve
Tikhonov	$O(pk)$	$O(k^3)$	$O(pk)$
TSVD	$O(k^3)$	$O(k^3)$	$O(k^3)$
Rust's	$O(k^3)$	$O(k^3)$	$O(k^3)$

TABLE 3.2

Summary of additional storage for each of four regularization methods under each of three parameter selection techniques. The original matrix is $m \times n$ with q nonzeros, p is the number of discrete parameters that must be tried, k iterations are used in projection, and the factorizations are assumed to take $\hat{q}$ storage.

	Basic cost	Added cost		
		Disc.	GCV	L-curve
Tikhonov	$O(\hat{q})$	$O(1)$	$O(p)$	$O(p)$
TSVD	$O(\hat{q})$	$O(1)$	$O(m)$	$O(m)$
Rust's TSVD	$O(\hat{q})$	$O(m)$	$O(m)$	$O(m)$
Projection	$O(kn)$	$O(1)$	$O(k)$	$O(k)$

TABLE 3.3

Summary of storage, not including storage for the matrix, for projection plus inner regularization approach and various parameter selection techniques. Here p denotes the number of discrete parameters tried. Each of these regularization methods also requires us to save the basis V or else regenerate it in order to reconstruct x.

Projection plus –	Disc.	GCV	L-curve
Tikhonov	$O(1)$	$O(p)$	$O(p)$
TSVD	$O(1)$	$O(k)$	$O(k)$
Rust's TSVD	$O(k)$	$O(k+p)$	$O(k+p)$

a good solution $y^{(k)}$, but we can use any of the methods of section 2 to regularize this projected problem; we discuss options in detail below.

If we used the algorithm GMRES [35] instead of LSQR, we would derive similar relations. Here, though, the U and V matrices are identical and the B matrix is upper Hessenberg rather than bidiagonal. Conjugate gradients would yield similar relationships.

For cost comparisons for these methods, see Tables 2.1 and 3.1. Storage comparisons are given in Tables 3.2 and 3.3.

3.1. Regularization by projection. As mentioned earlier, if we terminate the iteration after k steps, we have projected the solution onto a k-dimensional subspace and this has a regularizing effect that is sometimes sufficient. Determining the best value of k can be accomplished, for instance, by one of our three methods of parameter choice. Efficient implementation relies on LSQR recurrences for determining $\|r^{(k)}\|$ and $\|x^{(k)}\|$ cheaply, without computing either $r^{(k)}$ or $x^{(k)}$ [30, 34].

For the **discrepancy principle.** we stop the iteration for the smallest value of k for which $\|r_k\| \leq \tau\delta$.

To apply **GCV**, we note that in LSQR (see section 3.1), the operator $AA^\sharp$ is given by $U_{k+1}B_kB_k^\dagger U_{k+1}^*$, where $B_k^\dagger$ is the pseudoinverse of B_k. Thus from (2.2), the

GCV functional is [19]

$$G(k) = \frac{\|r^{(k)}\|_2^2}{(m-k)^2}.$$

We note that there are in fact two distinct definitions for $A^\sharp$ and hence two definitions for the denominator in $G(k)$; for small enough k, the two are comparable, and the definition we use here is less expensive to calculate [19, section 7.4].

To determine the **L-curve** associated with LSQR, values of $\|r^{(k)}\|_2$ and $\|x^{(k)}\|_2$ are needed for several values of k. In using this method or GCV, one must go a few iterations beyond the optimal k in order to verify the optimum [20].

3.2. Regularization by projection plus TSVD. If projection alone does not regularize, then we can compute the TSVD regularized solution to the projected problem (3.6). We need the SVD of the $(k+1) \times k$ matrix B_k. This requires $O(k^3)$ operations but can also be computed from the SVD of B_{k-1} in $O(k^2)$ operations [14].

Clearly, we still need to use some type of parameter selection technique to find a good value of $\ell(k)$. First, notice that it is easy to compute the norms of the residual and the solution resulting from retaining only the ℓ largest singular values. If ξ_{jk} is the component of e_1 in the direction of the jth left singular vector of B_k, and if γ_j is the jth singular value (ordered largest to smallest), then the residual and solution 2-norms are

$$\|r_\ell^{(k)}\| = \beta_1 \left(\sum_{j=\ell(k)+1}^{k+1} \xi_{jk}^2 \right)^{1/2} \quad \text{and} \quad \|x_\ell^{(k)}\| = \beta_1 \left(\sum_{j=1}^{\ell(k)} \left(\frac{\xi_{jk}}{\gamma_j} \right)^2 \right)^{1/2}. \tag{3.7}$$

Using this fact, we can use any of our three sample methods.

For the **discrepancy principle** we choose $\ell(k)$ to be the smallest value for which $\|r_\ell^{(k)}\| \le \tau\delta$, if such a value exists. As k increases, the number of neglected singular values will be monotonically nondecreasing (exact arithmetic).

The **GCV** functional for the kth projected problem is obtained by substituting B_k for A and $B_k^\sharp$ for $A^\sharp$, and substituting the expression of the residual in (3.7) for the numerator in (2.2):

$$G_k(\ell) = \frac{\beta_1^2 \sum_{j=\ell+1}^{k+1} \xi_{jk}^2}{(k-\ell+1)^2}.$$

We now have many **L-curves**, one for each value of k. The coordinate values in (3.7) form the discrete L-curve for a given k, from which the desired value of $\ell(k)$ can be chosen without forming the approximate solutions or residuals.

3.3. Regularization by projection plus Rust's TSVD. As in standard TSVD, to use Rust's version of TSVD for regularization of the projected problem requires computing the SVD of the $(k+1) \times k$ matrix B_k. Using the previous notation, Rust's strategy is to set

$$y_\rho^{(k)} = \sum_{j \in \mathcal{I}_\rho^{(k)}} \frac{\xi_{jk}}{\gamma_j} q_j^{(k)},$$

where $q_j^{(k)}$ are the right singular vectors of B_k and $\mathcal{I}_\rho^{(k)} = \{i < k+1 : |\xi_{ik}| > \rho\}$. We focus on three ways to determine ρ.

For the **discrepancy principle,** the norm of the residual of the regularized solution is given by $\|r_\rho^{(k)}\|_2 = \beta_1(\sum_{j\notin\mathcal{I}_\rho^{(k)}} \xi_{jk}^2)^{1/2}$. According to the discrepancy principle, we must choose ρ so that the residual is less than $\tau\delta$. In practice, this would require that the residual be evaluated by sorting the values $|\xi_{ik}|$ and adding terms in that order until the residual norm is less than $\tau\delta$.

For **GCV,** let $\mathrm{card}(\mathcal{I}_\rho^{(k)})$ denote the cardinality of the set $\mathcal{I}_\rho^{(k)}$. From (2.2), it is easy to show that the GCV functional corresponding to the projected problem for this regularization technique is given by

$$G_k(\rho) = \frac{\beta_1^2 \sum_{j\in\mathcal{I}_\rho^{(k)}} \xi_{jk}^2}{(k+1-\mathrm{card}(\mathcal{I}_\rho^{(k)}))^2}.$$

In practice, for each k we first sort the values $|\xi_{ik}|, i = 1,\dots,k$, from smallest to largest. Then we define k discrete values ρ_j to be equal to these values with ρ_1 being the smallest. We set $\rho_0 = 0$. Note that because the values of $\rho_j, j = 1,\dots,k$, are the sorted magnitudes of the SVD expansion coefficients, we have

$$G_k(\rho_j) = \frac{\beta_1^2(|\xi_{(k+1),k}|^2 + \sum_{i=1}^j \rho_i^2)}{(j+1)^2}, \quad j = 0,\dots,k.$$

Finally, we take the regularization parameter to be the ρ_j for which $G_k(\rho_j)$ is a minimum.

As with standard TSVD, we now have one **L-curve** for each value of k. For fixed k, if we define the $\rho_j, j = 0,\dots,k$, as we did for GCV above and we reorder the γ_i in the same way that the $|\xi_{ik}|$ were reordered when sorted, then we have

$$\|x_{\rho_j}^{(k)}\|_2^2 = \beta_1^2 \sum_{i=j+1}^{k} \left(\frac{\rho_i}{\gamma_i}\right)^2; \ \|r_{\rho_j}^{(k)}\|_2^2 = \beta_1^2\left(|\xi_{(k+1),k}|^2 + \sum_{i=1}^{j} \rho_i^2\right), \quad j = 0,\dots,k.$$

When these solution and residual norms are plotted against each other as functions of ρ, the value of ρ_j corresponding to the corner is selected as the regularization parameter.

3.4. Regularization by projection plus Tikhonov. Finally, let us consider using Tikhonov regularization to regularize the projected problem (3.5) for some integer k. Thus, for a given regularization parameter λ, we would like to solve

$$\min_y \|\beta_1 e_1 - B_k y\|_2^2 + \lambda^2\|y\|_2^2. \tag{3.8}$$

The solution $y_\lambda^{(k)}$ satisfies

$$(V_k^* A^* A V_k + \lambda^2 I) y_\lambda^{(k)} = V_k^* A^* b. \tag{3.9}$$

We need to address how to choose a suitable value of λ.

For the **discrepancy principle,** note that in exact arithmetic, we have

$$r_\lambda^{(k)} = b - A x_\lambda^{(k)} = U_{k+1}^*(\beta_1 e_1 - B_k y_\lambda^{(k)}). \tag{3.10}$$

Hence $\|B_k y_\lambda^{(k)} - \beta_1 e_1\|_2 = \|r_\lambda^{(k)}\|_2$. Therefore, to use the discrepancy principle requires that we choose λ so that $\|r_\lambda^{(k)}\|_2 \le \tau\delta$ with p discrete trial values λ_j. For a given k,

we take λ to be the largest value λ_j for which $\|r_\lambda^{(k)}\|_2 < \tau\delta$, if it exists; if not, we increase k and test again.

For **GCV,** let us define $(B_k)_\lambda^\dagger$ to be the operator mapping the right-hand side of the projected problem onto the regularized solution of the projected problem:

$$(B_k)_\lambda^\dagger = (B_k^* B_k + \lambda^2 I)^{-1} B_k^*.$$

Given the SVD of B_k as above, the denominator in the GCV functional defined for the projected problem (refer to (2.2)) is

$$\left(k + 1 - \sum_{j=1}^{k} \frac{\gamma_j^2}{\gamma_j^2 + \lambda^2} \right)^2.$$

The numerator is simply $\|r_\lambda^{(k)}\|_2^2$. For values of $k \ll n$, it is feasible to compute the singular values of B_k.

The **L-curve** is comprised of the points $(\|B_k y_\lambda^{(k)} - \beta_1 e_1\|_2, \|y_\lambda^{(k)}\|_2)$. But using (3.10) and the orthonormality of the columns of V_k, we see these points are precisely $(\|r_\lambda^{(k)}\|_2, \|x_\lambda^{(k)}\|_2)$. For p discrete values of λ, $\lambda_i, 1 \le i \le p$, the quantities $\|r_{\lambda_i}^{(k)}\|_2$ and $\|x_{\lambda_i}^{(k)}\|_2$ can be obtained by updating their respective estimates at the $(k-1)$st iteration.[1]

3.5. Correspondence between direct regularization and projection plus regularization. In this section, we demonstrate why the projection plus regularization approaches can be expected to yield regularized solutions nearly equivalent to the direct regularization counterpart. The following theorem, a simple corollary of the invariance of Krylov sequences under shifts, establishes the desired result for the case of Tikhonov vs. projection plus Tikhonov.

THEOREM 3.1. *Fix $\lambda > 0$ and define $x_\lambda^{(k)}$ to be the kth iterate of conjugate gradients applied to the Tikhonov problem*

$$(A^*A + \lambda^2 I)x = A^*b.$$

Let $y_\lambda^{(k)}$ be the exact solution to the regularized projected problem

$$(B_k^* B_k + \lambda^2 I)y = B_k^*(\beta e_1),$$

where B_k, V_k are derived from the original problem $A^*A = A^*b$, and set $z_\lambda^{(k)} = V_k y_\lambda^{(k)}$. Then $z_\lambda^{(k)} = x_\lambda^{(k)}$.

Proof. See [15, p. 301]. □

Let us compare TSVD regularization applied to the original problem to the projection plus TSVD approach. Direct computation convinces us that the two methods compute the same regularized solution if $k = n$ and arithmetic is exact. An approximate result holds in exact arithmetic when we take k iterations, with $\ell \le k \le n$. Let the SVD of B_k be denoted by $B_k = Z_k \Gamma_k Q_k^T$, and define the $s \times \ell$ matrix $W_{s,\ell}$ as

$$W_{s,\ell} = \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

[1] The technical details of the approach are found in [29, pp. 197–198], from which we obtain $\|r_\lambda^{(k)}\| = \sqrt{\|\bar{r}_\lambda^{(k)}\|^2 - \lambda^2 \|x_\lambda^{(k)}\|^2}$. The implementation details for estimating $\|x_\lambda^{(k)}\|$ and $\|\bar{r}_\lambda^{(k)}\|$ were taken from the Paige and Saunders algorithm at http://www.netlib.org/linalg/lsqr.

Then the regularized solution obtained from the TSVD regularization of the projected problem is

$$x_{reg}^{(k)} = V_k(Q_k W_{k,\ell} \Gamma_{k,1}^{-1} W_{k+1,\ell}^T Z_k^T U_k^T b),$$

where $\Gamma_{k,1}$ denotes the leading $\ell \times \ell$ principal submatrix of Γ_k. If k is taken to be sufficiently larger than ℓ so that $V_k Q_k W_{k,\ell} \approx \hat{V} W_{n,\ell}$, $W_{k+1,\ell}^T Z_k^T U_{k+1}^T \approx W_{n,\ell}^T \hat{U}^T$, and $\Gamma_{k,1} \approx \Sigma_1$ with Σ_1 the leading principal submatrix of Σ, then we expect $x_{reg}^{(k)}$ to be a good approximation to x_ℓ. This is made more precise in the following theorem.

THEOREM 3.2. *Let $k \geq \ell$ such that*

$$(V_k Q_k W_{k,\ell}) = \hat{V}_1 + E_1 \quad \text{with } \|E_1\| \leq \delta_1 \ll 1,$$

$$(U_{k+1} Z_k W_{k+1,\ell}) = \hat{U}_1 + E_2 \quad \text{with } \|E_2\| \leq \delta_2 \ll 1,$$

where $\hat{V}_1$ and $\hat{U}_1$ contain the first ℓ columns of $\hat{V}$ and $\hat{U}$, respectively. Let $D = diag(d_1, \ldots, d_\ell)$ satisfy

$$\Gamma_{k,1} = \Sigma_1 + D \quad \text{with } |d_i| \leq \delta_3 \ll 1.$$

Then

$$\|x_{reg}^{(k)} - x_\ell\| \leq \max_{1 \leq i \leq \ell} \frac{1}{\sigma_i + d_i} \left(\frac{\delta_3}{\sigma_\ell} + 3 \max(\delta_1, \delta_2) \right) \|b\|.$$

Proof. Using the representations $x_\ell = \hat{V}_1 \Sigma_1^{-1} \hat{U}_1^T b$ and $x_{reg}^{(k)} = (\hat{V}_1 + E_1)\Gamma_{k,1}^{-1}(\hat{U}_1^T + E_2^T)b$, we obtain

$$\|x_{reg}^{(k)} - x_\ell\| \leq (\|\Gamma_{k,1}^{-1} - \Sigma_1^{-1}\| + \|\Gamma_{k,1}^{-1}\| \|E_2\| + \|E_1\| \|\Gamma_{k,1}^{-1}\| + \|E_1\| \|\Gamma_{k,1}^{-1}\| \|E_2\|)\|b\|,$$

and the conclusion follows from bounding each term. ∎

Note that typically $\sigma_\ell \gg \sigma_n$ so that $1/\sigma_\ell$ is not too large. The bound says that the better LSQR captures the first ℓ singular values and vectors, the more we are assured the solution obtained by projection plus TSVD is close to the TSVD regularized solution to the original problem. For some results relating to the value of k necessary for the hypothesis of the theorem to hold, refer to the theory of Kaniel-Paige and Saad [31, section 12.4]. There is no universal recipe, but if k is large enough that the projected problem satisfies the discrete Picard condition, then this is some indication that the approximability property holds.

4. Numerical results. In this section, we present three numerical examples. All experiments were carried out using Matlab with IEEE double precision floating point arithmetic. Where noted, we made use of certain routines in Hansen's Regularization Tools [18]. Since the exact, noise-free solutions were known in these examples, we evaluated the methods using the relative, 2-norm difference between the regularized solutions and the exact solutions. When we applied Rust's method to the original problem, the ρ_i were taken to be the magnitudes of the spectral coefficients of b sorted in increasing order.

TABLE 4.1

Example 1: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for each of four regularization methods on the original problem, where the regularization method was chosen using methods indicated. The parameter values selected for each method are indicated in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.2E−2	(8.0E−2) 2.2E−2	(4.0E−2) 4.3E−2	(1.3E−1) 2.1E−2
TSVD	(6) 1.1E−1	(9) 1.6E−2	(10) 1.6E−2	(9) 1.6E−2
Rust's TSVD	(1.6E−2) 2.5E−2	(5.3E−5) 2.2E + 4	(1.6E−2) 2.5E−2	(1.6E−2) 2.5E−2
Projection	(5) 2.5E−2	(5) 2.5E−2	(10) 2.2E−2	(9) 2.2E−2

4.1. Example 1. The 200 × 200 matrix A and true solution x_{true} for this example were generated using the function `phillips` in Hansen's Regularization Tools. We generated $b_{true} = Ax_{true}$ and then computed the noisy vector b as $b + e$, where e was generated using the Matlab `randn` function and was scaled so that the noise level, $\frac{\|e\|}{\|b_{true}\|}$, was 5×10^{-3}. The condition number of A was on the order of 4×10^7.

Table 4.1 displays the values of the regularization parameters chosen when the original problem was solved using one of the three parameter selection techniques together with one of the four regularization methods. We set $\tau\delta$ for the discrepancy principle to be 8E−2, close to the value $\|e\|_2 = 7.65\text{E}{-2}$.

The last column in the table gives the value of the parameter that yielded a regularized solution with minimum relative error. Several values of λ were tested: $\log_{10}\lambda = -4, -3.9, \ldots, 0$. The relative error values for regularized solutions corresponding to the parameters are also presented in this table. The GCV and L-curve parameters for projection were determined after 15 iterations. Note that using GCV to determine a regularization parameter for Rust's TSVD resulted in an extremely noisy solution with huge error.

The corners of the L-curves for the Tikhonov, projection, and TSVD methods were determined using Hansen's `lcorner` function, with the modification that sometimes points not strictly on the portion of the curve that was L-shaped (that is, points with very large residual or very small residual) were not considered (otherwise, a false corner resulted); this was most often a concern with the TSVD method. Since the corner was so clearly defined for Rust's method but the function had trouble automatically finding the corner, the corner was picked manually.

Next, we projected using LSQR and then regularized the projected problem with one of the other three regularization methods together with one of the three parameter selection techniques. Results at iterations 10 and 25 are given in Tables 4.2 and 4.3, respectively. As before, the `lcorner` routine was used to determine the corners of the respective L-curves, with the modifications as mentioned above.

Comparing Tables 4.1 and 4.2, we observe that using either the discrepancy principle or the L-curve, 10 steps of projection plus Tikhonov gives results as good as or much better than if those techniques had been used with Tikhonov on the original problem. A similar statement can be made for projection plus Rust's TSVD when any of the 3 selection methods are used and for projection plus TSVD when the discrepancy principle is used. After 25 iterations, the errors for projection plus Tikhonov or Rust's TSVD closely resemble the errors in Table 4.1 with one exception. We note that at 25 iterations, the parameters chosen for projection plus Tikhonov by the discrepancy principle or the L-curve method and their corresponding errors are identical to those chosen for the original problem.

In fact, the L-curve, GCV, and discrepancy methods applied to the projected

TABLE 4.2
Example 1, *iteration* 10: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. The parameter values for each method are indicated in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.1E−2	(2.5E−2) 2.5E−2	(2.0E−4) 2.2E−2	(2.0E−2) 2.0E−2
TSVD	(7) 2.5E−2	(7) 2.5E−2	(10) 2.2E−2	(10) 2.2E−2
Rust's TSVD	(9.7E−3) 2.5E−2	(9.7E−3) 2.5E−2	(5.5E−4) 2.2E−2	(9.1E−3) 2.1E−2

TABLE 4.3
Example 1, *iteration* 25: *comparison of* $\|x_{true} - x_{reg}\|_2/\|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. The parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 2.2E−2	(2.0E−1) 2.3E−2	(4.0E−2) 4.3E−2	(1.3E−1) 2.1E−2
TSVD	(17) 2.5E−2	(17) 2.5E−2	(21) 2.4E−2	(19) 1.6E−2
Rust's TSVD	(2.0E−2) 2.5E−2	(2.0E−2) 2.5E−2	(1.5E−2) 2.5E−2	(1.5E−2) 2.5E−2

problem with Tikhonov regularization consistently chose the same parameter for future iterations (see Figure 4.1, for instance), and correspondingly the errors remain constant; however, the results at earlier iterations are actually better than after the parameter on the projected problem has converged to the L-curve parameter on the original. For the projection plus TSVD, both the discrepancy principle and GCV method yielded parameters for which the solutions had similar errors from one iteration to the next for at least the first 80 iterations (see the top of Figure 4.2); the L-curve behaved slightly less consistently for iterations beyond about 50. Discrepancy and GCV when applied to projection plus Rust's TSVD also gave consistent solutions for about 40 iterations, after which the GCV solutions began to grow very large in error, much like GCV applied to the original problem (refer to the bottom of Figure 4.2).

4.2. Example 2. The 3969×3969 matrix A for this example was a symmetric, block Toeplitz matrix with Toeplitz blocks formed according to $A = T \otimes T$. Here T is a symmetric, banded Toeplitz matrix with entries $T_{i,j} = t_{i-j}$; the nonzero entries in the first row were $t_k = (\sin(k/B)/(k/B))^2, 0 \le k \le 4, B = .8$. The singular values of this matrix range from 5.7 to 8.6×10^{-8} but do not decay very quickly, and the matrix has a condition number of about 7×10^7. x was obtained by stacking by columns the 63×63 image that was zero except for a rectangle with value 1 from rows 20 to 49, columns 4 to 24, and another rectangle with value .8 at rows 23 to 53, columns 29 to 52. We generated $b_{true} = Ax_{true}$ and then computed the noisy vector b as $b + e$, where e was generated using the Matlab `randn` function and was scaled so that the noise level, $\frac{\|e\|}{\|b_{true}\|}$, was 2×10^{-3}.

We generated our discrete λ_i using $\log_{10} \lambda = -4, -4.9, \ldots, 0$. The norm of the noise vector was 3.66E−1, so we took $\tau\delta = 4.00\text{E}{-1}$ for the discrepancy principle.

In this example, when no preconditioning was used, it took 90 iterations for LSQR to reach a minimum relative error of 7.93E−2. Likewise, the dimension k of the projected problem had to be at least 90 to obtain good results with the projection-plus-regularization approaches and even larger for the parameter selection techniques to work well on the projected problem. Therefore, for the projection based techniques, we chose to work with a left preconditioned system (refer to the discussion at the end of section 2.1). Our preconditioner was chosen as in [23] where the parameter defining the preconditioner was taken to be $m = 2080$. Results for right preconditioning were

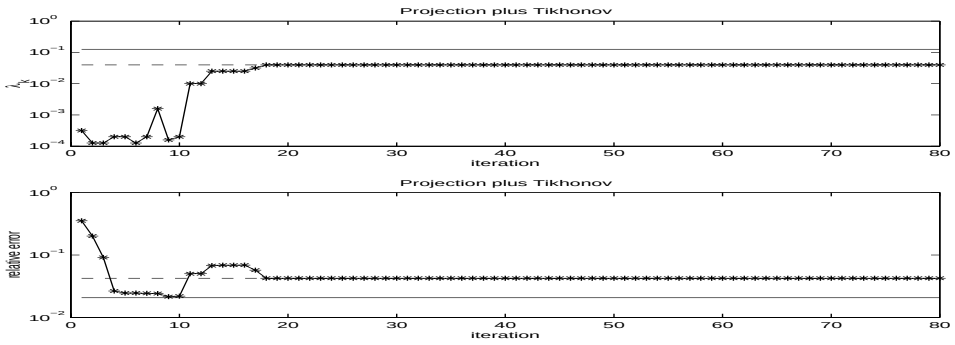


FIG. 4.1. *Example* 1. *Top:* λ_k *as selected by L-curve method; bottom: relative error for corresponding solution. The solid line indicates the optimal value on the original problem, and the dashed line indicates value selected by L-curve on the original problem.*

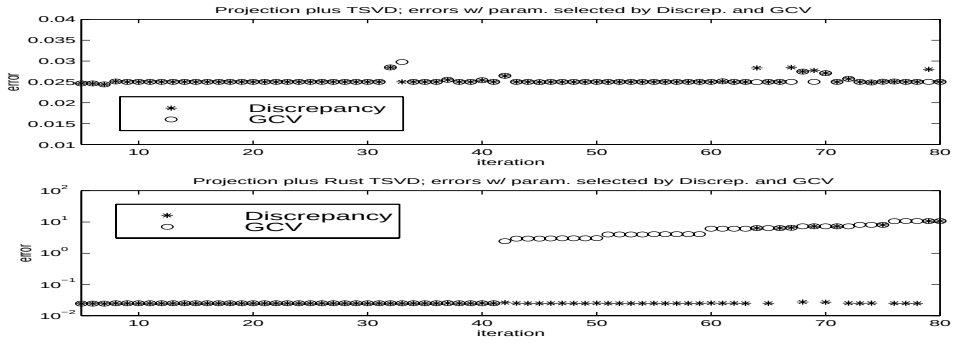


FIG. 4.2. *Example* 1. *Relative error between computed and exact solutions for projection plus TSVD (top) and projection plus Rust's TSVD (bottom) when the parameters for the projected problem are selected by either the discrepancy principle (*) or GCV method (o).*

similar, although the errors were not quite as small. On other examples, though, we found that right preconditioning by this type of preconditioner was only effective in certain instances, even when left preconditioning was effective.[2]

The results of the resulting regularization for the original problem parameters are given in Table 4.4. We note that GCV with Rust's TSVD was ineffective. Also, after 50 iterations on the left preconditioned system, the GCV functional for projection was still decreasing, so the value in Table 4.4 corresponds to the value after 50 iterations. The L-curve parameter in the table was determined after 20 iterations.

Although we projected using LSQR, we note that since the matrix and preconditioner were symmetric, we could have used MINRES as in [23]. The results in each case at iterations $10, 20$, and 40 are given in Tables 4.5, 4.6, and 4.7, respectively, and we summarize results up to 60 iterations in the discussion below.

Again, we used the `lcorner` routine to determine the corners of the respective L-curves, with the modification that for 20 iterations and beyond for TSVD, we first removed points on the curve with residual norm greater than 10 to avoid detecting a false corner.

[2]In the language of [23], right preconditioning worked well only when K was a very good approximation to C so that right preconditioning did not mix noise into early iterates; left preconditioning was not nearly as sensitive to the approximation on the transition and noise subspaces.

TABLE 4.4

Example 2: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for each of four regularization methods on the original problem. The parameter values are given in parentheses. The projection was performed on a left preconditioned system.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.6E−1) 8.5E−2	(5.0E−2) 8.0E−2	(3.2E−3) 5.3E−1	(6.3E−2) 7.8E−2
TSVD	(2073) 9.9E−2	(2534) 8.1E−2	(1509) 1.2E−1	(2521) 8.0E−2
Rust's TSVD	(2.1E−2) 7.6E−2	(9.2E−2) 4.0E + 3	(1.6E−2) 2.3E−1	(2.0E−2) 7.6E−2
Projection	(2) 9.7E−2	(50+) 2.7E−1	(13) 8.3E−2	(8) 7.9E−2

TABLE 4.5

Example 2, *iteration* 10: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.9E−2	(2.0E−4) 7.9E−2	(5.0E−2) 7.9E−2
TSVD	(6) 9.9E−2	(6) 7.9E−2	(8) 9.8E−2	(10) 7.9E−2
Rust's TSVD	(2.2E−1) 8.5E−2	(2.6E−1) 9.9E−2	(2.3E−1) 9.9E−2	(3.9E−4) 7.9E−2

TABLE 4.6

Example 2, *iteration* 20: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.8E−2	(2.0E−4) 1.1E−1	(6.3E−2) 7.8E−2
TSVD	(12) 9.9E−2	(12) 9.9E−2	(19) 8.3E−2	(19) 8.3E−2
Rust's TSVD	(1.6E−1) 9.5E−1	(7.9E−2) 1.1E−1	(4.6E−2) 1.1E−1	(1.3E−1) 8.3E−2

TABLE 4.7

Example 2, *iteration* 40: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(7.9E−2) 7.9E−2	(6.3E−2) 7.8E−2	(2.0E−1) 2.3E−1	(6.3E−2) 7.9E−2
TSVD	(24) 9.9E−2	(24) 9.9E−2	(28) 9.9E−2	(38) 8.3E−2
Rust's TSVD	(1.5E−1) 9.2E−2	(5.8E−2) 2.3E−1	(1.6E−1) 9.2E−2	(1.5E−1) 9.2E−2

Discrepancy and GCV consistently chose the same regularization parameter and hence gave the same error for projection plus Tikhonov for 10 to 60 iterations. From the tables, we see that these are not the same parameters as those chosen when applied to the original problem and that, in fact, the solutions for projection plus Tikhonov have smaller error. The errors for the solutions obtained using any of the 3 parameter selection methods applied to find ℓ for projection plus TSVD were also consistent for 10 to 60 iterations, as alluded to in the tables. Figure 4.3 shows the errors from iterations 5 to 60 for projection plus Tikhonov and projection plus TSVD when GCV is used. For Rust's TSVD, the L-curve and discrepancy rules are fairly consistent at picking parameters that give solutions with similar error from iteration to iteration. We note that GCV for Rust's TSVD picked parameters giving solutions with reasonably small errors, even though GCV for Rust's TSVD on the original problem failed, giving a solution with huge error. A similar statement can be made for the L-curve with projection plus Tikhonov.

Summarizing, we observe two phenomena. First, the parameters selected to regularize the projected problem can be different from those chosen on the original problem but still yield solutions of better or comparable error. Second, as this and the previous

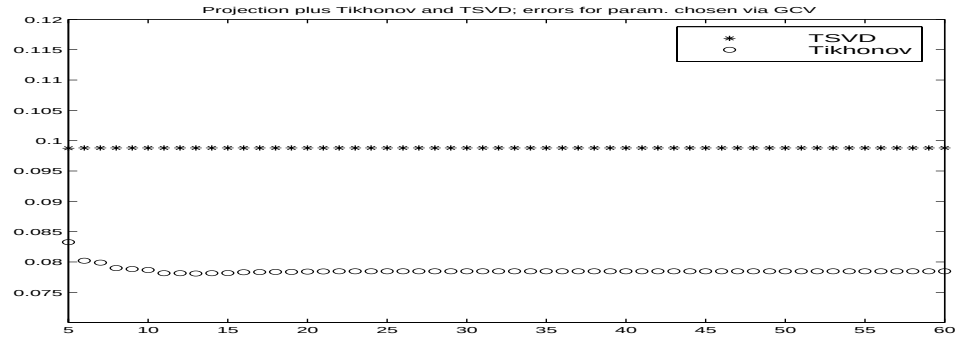


FIG. 4.3. *Example 2: Errors for projection plus Tikhonov (*) and projection plus TSVD (o) when the regularization parameter for the projected problem was given by GCV.*

TABLE 4.8

Example 3: comparison of $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for each of the 4 regularization methods on the original problem. Parameter values are given in parentheses. Those for GCV and the L-curve are those selected after 30 iterations.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 3.9E−1	(1.3) 4.0E−1	(5.0E−1) 4.1E−1	(7.9E−1) 3.9E−1
TSVD	(232) 4.2E−1	(400) 1.3E+4	(261) 4.0E−1	(241) 4.0E−1
Rust's TSVD	(3.0E−1) 7.4E+2	(1.8E−1) 1.2E+4	(3.7) 4.6E−1	(3.9E−1) 4.6E−1
Projection	(9) 4.0E−1	(23) 4.3E−1	(16) 4.0E−1	(12) 3.9E−1

example show, loss of orthogonality does not seem to hamper the parameter selection process, at least not for a reasonable number of iterations. This may be due to the fact that the parameter selection methods are applied directly to the projected problem: for example, the denominator of our GCV function for projection plus TSVD is different from the denominator of the GCV function given in [2, (3.8)].

4.3. Example 3. Our final example is from the field of computed tomography. In this example, the true vector x corresponded to the 20×20 image created with the `phantom.m` function. The matrix A was the corresponding 561×400 Radon transform matrix where it is understood that the data was taken at angles from 0 to 179 degrees in increments of 11 degrees. The matrix itself was computed (albeit naively) in Matlab column by column using successive applications of `radon.m` on images of point sources. The singular values fall off very slowly at first (the first 260 of the 400 singular values range between 18 and about 1) after which they fall off rapidly, resulting in a condition number for A of about 10^7.

Since the norm of the noise vector was about 3.44, we took the tolerance for the discrepancy principle to be 3. The discrete values λ_i used for Tikhonov regularization were 51 evenly log-spaced points between 10^{-4} and 10^1. The results computed using discrepancy, GCV, and L-curve methods for Tikhonov, TSVD, Rust's TSVD, and projection on the original problem are given in Table 4.8.

Table 4.9 gives the results after 10 iterations of LSQR. Notice that the errors for the projection plus Tikhonov solutions via GCV and L-curve are slightly better than the corresponding error for Tikhonov without projection at only 10 iterations. Also interesting is the fact that at 10 iterations the discrepancy and GCV methods for projection plus Rust's TSVD give solutions with reasonable errors, whereas these techniques give solutions with very large errors when applied to the original problem.

TABLE 4.9

Example 3, *iteration* 10: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 4.0E−1	(2.2) 4.0E−1	(1.6E−4) 3.9E−1	(4.0E−1) 4.0E−1
TSVD	(10) 3.9E−1	(1) 8.6E−1	(5) 8.3E−1	(10) 3.9E−1
Rust's TSVD	(1.0) 3.9E−1	(1.5) 4.0E−1	(2.2) 4.0E−1	(0.0) 3.9E−1

TABLE 4.10

Example 3, *iteration* 40: *comparison of* $\|x_{true} - x_{reg}\|_2 / \|x_{true}\|_2$ *for projection plus Tikhonov, TSVD, and Rust's TSVD. Parameter values are given in parentheses.*

	Disc.	GCV	L-curve	Optimal
Tikhonov	(1.0) 3.9E−1	(1.2) 4.1E−1	(5.0E−1) 4.1E−1	(7.9E−1) 3.9E−1
TSVD	(37) 4.0E−1	(15) 7.8E−1	(39) 4.1E−1	(38) 4.0E−1
Rust's TSVD	(6.0E−1) 4.2E−1	(1.2) 4.1E−1	(2.7E−1) 4.1E−1	(6.6E−1) 4.0E−1

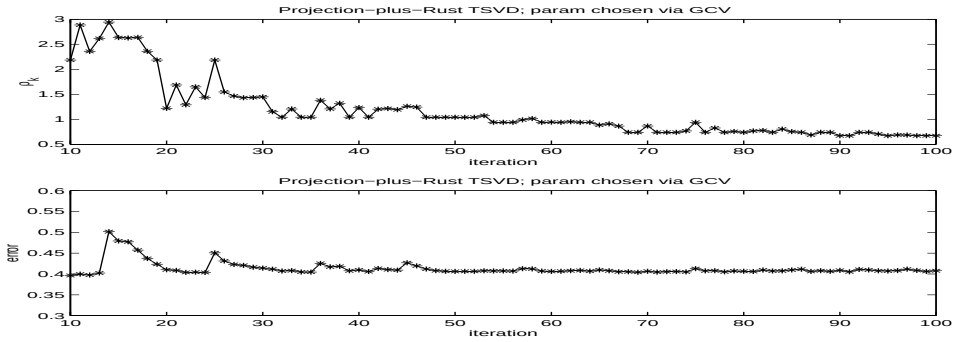


FIG. 4.4. *Example* 3. *Top: Value of* ρ_k *selected by GCV for projection plus Rust's TSVD; Bottom: Relative error of the corresponding solutions.*

Table 4.10 shows the parameters and the errors after 40 iterations. From these results, we see that the L-curve for projection plus Tikhonov eventually gives the same regularization parameter and same solution error as when applied to the larger problem, and we observed this to be true for several iterations beyond 40. Again, we see that discrepancy and GCV used with projection plus Rust's TSVD is effective, whereas they are ineffective when used on the original problem; we observed this behavior well beyond 40 iterations (see Figure 4.4).

5. Conclusions. In this work we have given a common framework for methods based on regularizing a projected problem. We have shown that determining regularization parameters based on the final projected problem rather than on the original discretization has firmer mathematical justification and often involves less computational expense. We presented results that in fact the regularized solution obtained by backprojecting the TSVD or Tikhonov solution to the projected problem is almost equivalent to applying TSVD or Tikhonov to the original problem, where "almost" depends on the size of k. The examples indicate the practicality of the method and illustrate that our regularized solutions are usually as good as those computed using the original system, and they can be computed in a fraction of the time, using a fraction of the storage. We note that similar approaches are valid using other Krylov subspace methods for computing the projected problem.

In this work, we did not address potential problems from loss of orthogonality as the iterations progress. In this discussion, we did, however, assume that either k was naturally very small compared to n or that preconditioning had been applied to enforce this condition. Possibly for this reason, we found that for modest k, round-off did not appear to degrade either the LSQR estimates of the residual and solution norms or the computed regularized solution in the following sense: the regularization parameters chosen via the projection-regularization and the corresponding regularized solutions were comparable to those chosen and generated for the original discretized problem. Another possible reason for the success of our approach is that we chose parameters for the projected problem directly, rather than for the backprojected, larger problem. In our experiments, we found that the parameters selected usually leveled out after a few iterations. The stagnation of the parameters themselves may suggest when k is large enough.

For the Tikhonov approach in this paper, we have assumed that the regularization operator L was the identity or was related to the preconditioning operator; this allowed us to efficiently compute $\|r_\lambda^{(k)}\|$ and $\|x_\lambda^{(k)}\|$ for multiple values of λ efficiently for each k. If L is not the identity but is invertible, we can first implicitly transform the problem to "standard form" [19]. With $\bar{A} = AL^{-1}$, $\bar{x} = Lx$, we can solve the equivalent system

$$\min_{\bar{x}} = \|\bar{A}\bar{x} - b\|_2^2 + \lambda^2\|\bar{x}\|_2^2.$$

Then the projection plus regularization schemes may be applied to this transformed problem. Clearly the projection based schemes will be useful as long as solving systems involving L can be done efficiently.

REFERENCES

[1] Å. Björck, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 28 (1988), pp. 659–670.

[2] A. Björck, E. Grimme, and P. V. Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems*, BIT, 34 (1994), pp. 510–534.

[3] P. Blomgren and T. F. Chan, *Modular Solvers for Constrained Image Restoration Problems*, Tech. Report, Mathematics Department, UCLA, Los Angeles, 1999.

[4] D. Calvetti, G. Golub, and L. Reichel, *Estimation of the L-curve via Lanczos bidiagonalization*, BIT, 39 (1999), pp. 603–619.

[5] T. Chan and M. Ng, *Galerkin projection method for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[6] J. Cullum and R. A. Willoughby, *Lanczos and the computation in specified intervals of the spectrum of large, sparse real symmetric matrices*, in Sparse Matrix Proceedings 1978, I. S. Duff and G. W. Stewart, eds., SIAM, Philadelphia, 1979, pp. 220–255.

[7] L. Desbat and D. Girard, *The "minimum reconstruction error" choice of regularization parameters: Some more efficient methods and their application to deconvolution problems*, SIAM J. Sci. Comput., 16 (1995), pp. 1387–1403.

[8] H. W. Engl and W. Grever, *Using the L-curve for determining optimal regularization parameters*, Numer. Math., 69 (1994), pp. 25–31.

[9] H. E. Fleming, *Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems*, Linear Algebra Appl., 130 (1990), pp. 133–150.

[10] A. Frommer and P. Maass, *Fast CG-based methods for Tikhonov-Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.

[11] G. Golub, M. Heath, and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.

[12] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[13] W. Groetsch, *Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, 1984.

[14] M. Gu and S. Eisenstat, *A Stable and Fast Algorithm for Updating the Singular Value Decomposition*, Tech. Report RR-939, Department of Computer Science, Yale University, New Haven, 1993.
[15] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Indust., 3 (1993), pp. 253–315.
[16] M. Hanke, J. Nagy, and R. Plemmons, *Preconditioned iterative regularization for ill-posed problems*, in Numerical Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan, and R. S. Varga, eds. 1993, pp. 141–163.
[17] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580.
[18] P. C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.
[19] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion,* SIAM Monogr. Math. Model Comput., SIAM, Philadelphia, 1998.
[20] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.
[21] B. Hofmann, *Regularization for Applied Inverse and Ill-Posed Problems*, Teubner-Texte Mathe. 85, Teubner, Leipzig, 1986.
[22] L. Kaufman and A. Neumaier, *Regularization of ill-posed problems by envelope guided conjugate gradients*, J. Comput. Graph. Statist., 6 (1997), pp. 451–463.
[23] M. Kilmer, *Regularization of ill-posed problems using (symmetric) Cauchy-like preconditioners*, in Proceedings of the SPIE Annual Meeting, Advanced Signal Processing Algorithms, Architectures, and Implementations VIII, 1998, SPIE, San Diego, CA, pp. 381–392.
[24] M. Kilmer and D. P. O'Leary, *Pivoted Cauchy-like preconditioners for regularized solution of ill-posed problems*, SIAM J. Sci. Stat. Comput., 21 (1999), pp. 88–110.
[25] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[26] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.
[27] J. Nagy, R. Plemmons, and T. Torgersen, *Iterative image restoration using approximate inverse preconditioning*, IEEE Trans. Image Process., 5 (96), pp. 1151–1163.
[28] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretization of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.
[29] C. C. Paige and M. A. Saunders, *Algorithm* 583*, LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Software, 8 (1982), pp. 195–209.
[30] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[31] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[32] B. N. Parlett and D. S. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[33] B. W. Rust, *Truncating the Singular Value Decomposition for Ill-Posed Problems*, Tech. Report NISTIR 6131, Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, MD, 1998.
[34] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[35] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[36] H. D. Simon, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–131.
[37] J. M. Varah, *Pitfalls in the numerical solution of linear ill-posed problems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 164–176.
[38] 1993, pp. 141–163. C. R. Vogel, *Non-convergence of the L-curve regularization parameter selection method*, Inverse Problems, 12 (1996), pp. 535–547.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1222–1244

MINIMAL SQUARE SPECTRAL FACTORS VIA TRIPLES*

M. A. PETERSEN† AND A. C. M. RAN‡

Abstract. We consider the problem of parametrizing the set of square minimal spectral factors of a rational matrix function taking positive semidefinite values on the imaginary axis in terms of invariant subspaces and of the minimal unitary left divisors of a certain unitary function. We shall use an approach which involves null-pole triples.

Key words. spectral factorization, minimal factorization, null-pole triples

AMS subject classifications. 47A68, 47A56, 15A24

PII. S0895479899357619

1. Introduction. In what follows, we will consider an $m \times m$ rational matrix function, $\Phi(\lambda)$, that has positive semidefinite values on the imaginary axis, $i\mathbb{R}$. Note that, in this case, it is possible that Φ may have poles or zeros on $i\mathbb{R}$. Furthermore, we shall assume that $\Phi(\infty) = I_m$, and we denote the McMillan degree of Φ by $2n$. We say that $W(\lambda)$ is a *minimal square spectral factor* of $\Phi(\lambda)$ if W is a rational $m \times m$ matrix function and

$$\Phi(\lambda) = W(\lambda)W(-\overline{\lambda})^* \tag{1.1}$$

is a minimal factorization. In other words, the McMillan degree of Φ is twice that of W. Here we denote the McMillan degree of W by $\delta(W)$, and we assume that $\delta(\Phi) = 2n$. We note that if $\Phi(\lambda) = W(\lambda)W(-\overline{\lambda})^*$, then Φ takes positive semidefinite values on the imaginary axis.

Various aspects of the problem of parametrizing all minimal square spectral factors of a given spectrum (i.e., a given positive semidefinite rational matrix function) were discussed in papers such as [CG], [C], and [FMP]. For instance, in the first of these, the problem was approached from a computational viewpoint. More recently, in [R2], the author considered a rational matrix function, Φ, that has real Hermitian, positive definite values on the imaginary axis and is invertible at infinity. Here, the constraint imposed in [FMP] that Φ should not have a pole which is also a zero was removed. In particular, it was proved that we may determine a parametrization of all minimal square spectral factors of a positive semidefinite rational matrix function in terms of invariant subspaces (see also [R1] and [RR3]). Also, it was proved that there is a one-to-one correspondence between the minimal unitary factorizations of some unitary matrix and the set of minimal square spectral factors. A strongly related parametrization was given in [F2]. Also, in [FG] and [LMP] minimal nonsquare spectral factors were studied, but with additional requirements on the behavior of the function $\Phi(\lambda)$ for pure imaginary values of λ. As in [R1], [R2], [F2], and [FMP] we shall restrict ourselves to the case of square spectral factors, but as in [R1] and [RR3] we allow for poles and zeros on the line. The problem of finding minimal square

*Received by the editors June 22, 1999; accepted for publication (in revised form) by U. Helmke August 23, 2000; published electronically April 6, 2001.
http://www.siam.org/journals/simax/22-4/35761.html
†Department of Mathematics and Applied Mathematics, University of Cape Town, Rondebosch 7700, Cape Town, South Africa (mpetersen@yebo.co.za).
‡Divisie Wiskunde en Informatica, Faculteit Exacte Wetenschappen, Vrije Universiteit, Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands (ran@cs.vu.nl).

spectral factors plays an important role in stochastic realization theory. For a detailed account we refer to the fundamental papers [LP1] and [LP2].

In our discussion, we show that results comparable to those in [R2] and [FMP] may be obtained in the positive semidefinite case. Here, the situation is more complicated, as we have to take into account the possible poles and zeros on the imaginary axis. However, a similar parametrization is possible, as we shall show. This parametrization is in terms of the zero and pole structure in the open right half plane and on the imaginary axis of the given positive semidefinite function.

Next, we define notions which are crucial in the development of the approach that will be adopted later. First, given a nonempty set σ in the complex plane $\mathbb{C}$, we denote by $\mathcal{R}_m(\sigma)$ the set of rational vector functions with values in $\mathbb{C}^m$ that are analytic outside σ. Now, let $\Phi(\lambda)$ be a regular $m \times m$ rational matrix function, taking the value I at infinity. Introduce the space of rational vector functions $\mathcal{S}(\Phi) = \{\Phi(\lambda)f(\lambda) \mid f(\lambda) \in \mathcal{R}_m(\sigma)\}$. (In case σ is the unit disc one can compare this space to the usual invariant subspace ΦH_2; in the study of rational matrix functions it plays a comparable role.) The space $\mathcal{S}(\Phi)$ gives information on the poles and zeros of Φ inside σ. More precisely, a collection of matrices

$$\omega = \{(C_1, A_1); (A_2, B_2); \Gamma : \mathcal{M} \to \mathcal{L}\}$$

is called a $\sigma-$*null-pole triple* of $\Phi(\lambda)$ if $A_1 : \mathcal{M} \to \mathcal{M}$ and $A_2 : \mathcal{L} \to \mathcal{L}$ both have spectra in σ, and for $C_1 : \mathcal{M} \to \mathbb{C}^m$ and $B_2 : \mathbb{C}^m \to \mathcal{L}$ we have that the pair (C_1, A_1) is observable and (A_2, B_2) is controllable, and finally, the set $\mathcal{S}(\Phi)$ is equal to

$$\left\{ C_1(\lambda - A_1)^{-1}x + h(\lambda) \mid x \in \mathcal{M},\ h \in \mathcal{R}_m(\sigma) \right.$$
$$\left. \text{such that } \sum_{\omega \in \sigma} \text{Res}_{\lambda=\omega}[(\lambda - A_2)^{-1}B_2 h(\lambda)] = \Gamma x \right\}.$$

Note that $\mathcal{M}$ and $\mathcal{L}$ are finite dimensional vector spaces here. As a consequence we have that the pair (C_1, A_1) is a right pole pair of Φ over σ; in particular, there is a matrix B_1 such that $\Phi(\lambda) - C_1(\lambda - A_1)^{-1}B_1$ is analytic on σ. Likewise, the pair (A_2, B_2) is a left null pair of Φ over σ, so there is a matrix C_2 such that $\Phi(\lambda)^{-1} - C_2(\lambda - A_2)^{-1}B_2$ is analytic on σ. Finally, it can be shown that the coupling operator Γ satisfies the Sylvester equation

$$\Gamma A_1 - A_2\Gamma = B_2C_1.$$

For a complete account of the theory connected to these notions, see, e.g., [BGR]. In what follows typical choices for σ will be $\mathbb{C}_+, \mathbb{C}_-$ or $i\mathbb{R}$, i.e., the open right half plane, the open left half plane, or the imaginary axis.

A null-pole triple is said to be *global* if Γ is invertible. In that case, the realization $I + C_1(\lambda - A_1)^{-1}\Gamma^{-1}B_2$ is minimal. It defines a rational matrix function $W(\lambda)$, say, the inverse of which is $W(\lambda)^{-1} = I + C_1\Gamma^{-1}(\lambda - A_2)^{-1}B_2$. Moreover, in that case, $W(\lambda)$ is a minimal divisor of $\Phi(\lambda)$ (see [GK] and also [BGR]).

The most important element of our strategy is that it must take pure imaginary poles and zeros of Φ into account. The idea is the following: let $\{(C, A); (A^\times, B); \Gamma\}$ be a global null-pole triple for Φ, i.e., $\Phi(\lambda) = I + C(\lambda - A)^{-1}\Gamma^{-1}B$ is a minimal realization of Φ, and assume that Φ takes positive semidefinite values on the imaginary axis. Let $H = -H^*$ be the unique invertible skew-Hermitian matrix such that $HA = -A^*H$

and $H\Gamma^{-1}B = C^*$. (Such an H exists by Kalman's state space isomorphism theorem.) We have a representation of all minimal square spectral factors in terms of invariant Lagrangian subspaces. As the sign condition holds (both for the pair (A, H) and the pair $(A^\times, H)$; see, e.g., [RR1], [RR2]) the part of the global triple of such a square spectral factor corresponding to its pure imaginary poles and zeros does not depend on the factor. In other words, this part of the global triple is the same for all factors, modulo, of course, similarity. A proof of this fact is given in Lemma 3.1. We also investigate the spectral factor W_+ (resp., W_-) which has the property that all its zeros and poles are in the closed right (resp., left) half plane. For these functions we consider a global triple, which is decomposed into the part corresponding to the open right (resp., left) half plane and the part corresponding to $i\mathbb{R}$. The strategy outlined above is discussed in more detail in section 3 of this paper. It should be noted that the main tools here come from the theory of null-pole triples [BGR]. Alternatively, parametrizations of all minimal square spectral factors can be based on the approach using [BGKvD] (see also [BGK] and [Sa]). The latter approach was taken in [R1] and [RR3].

In section 2, we consider a first parametrization in the spirit of [RR3] but using null-pole triples. The third section contains a statement and proof of the main result. Here we make use of the theory about invariant subspaces and null-pole triples. In section 4, we explain how a right minimal square spectral factor $W_-(\lambda)$ can be obtained from a left minimal square spectral factor $W_+(\lambda)$ by using the approach outlined here instead of the approach using [BGK] and [BR2]. Some comments on notation are in order here. In [FMP], [LP1], and [LP2] the notation W_+ is used for the stable spectral factor for which the zeros are in the right half plane (instead of as here, the antistable function for which all zeros are in the right half plane). This notation is different from ours, which is more in line with the notation used in the literature concerning Wiener–Hopf factorizations. That any choice here leads inevitably to confusion for at least some of the readers who are accustomed to a different convention is something we, as authors, regret, but have to live with. In section 5, we show that it is also possible to solve the problem of parametrizing the set of square minimal spectral factors in terms of the minimal unitary left divisors of the unitary function $U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$. Also, in section 6 we determine a parametrization in terms of the solutions of algebraic Riccati equations (compare [W], [FMP], [GLR], [R4], and [S]).

2. Parametrization in terms of invariant subspaces I. In this section, we discuss a first parametrization of all minimal square spectral factors of a positive semidefinite rational matrix function in terms of certain invariant subspaces. First, we provide an alternative formulation of the main results of [RR3] (compare also [R1]). Suppose that

$$\Phi(\lambda) = I + C(\lambda - A)^{-1}\Gamma^{-1}B \tag{2.1}$$

is a minimal realization of Φ, and its inverse is represented by

$$\Phi(\lambda)^{-1} = I - C\Gamma^{-1}(\lambda - Z)^{-1}B.$$

In other words, $\Theta = \{(C, A), (Z, B), \Gamma\}$ is a global null-pole triple for Φ. By Kalman's state space isomorphism theorem, there exist two unique invertible skew-Hermitian matrices H_p and H_z such that

$$H_pA = -A^*H_p\ , \quad H_p\Gamma^{-1}B = C^*, \tag{2.2}$$

and

$$H_z Z = -Z^* H_z \ , \quad H_z B = \Gamma^{*-1} C^*. \tag{2.3}$$

One easily shows (using the uniqueness of H_p and H_z) that

$$\Gamma^* H_z \Gamma = H_p. \tag{2.4}$$

Let $\mathcal{M}$ be an A-invariant H_p-Lagrangian subspace (i.e., $H_p\mathcal{M} = \mathcal{M}^\perp$), and let $\mathcal{M}^\times$ be a Z-invariant H_z-Lagrangian subspace. Let $P_{\mathcal{M}^\times}$ be the orthogonal projection along $\mathcal{M}^\times$, and denote by $\mathcal{L}$ its image. Then

$$\{(C|_{\mathcal{M}}, A|_{\mathcal{M}}), (P_{\mathcal{M}^\times} Z|_{\mathcal{L}}, P_{\mathcal{M}^\times} B), P_{\mathcal{M}^\times}\Gamma|_{\mathcal{M}}\}$$

is a corestriction of Θ in the sense of [GK].

LEMMA 2.1. *$P_{\mathcal{M}^\times}\Gamma|_{\mathcal{M}} : \mathcal{M} \to \mathcal{L}$ is invertible.*

Proof. Since $\dim \mathcal{M}^\times = \dim \mathcal{L} = \dim \mathcal{M}$, all we have to show is that $P_{\mathcal{M}^\times}\Gamma|_{\mathcal{M}}$ is injective. Suppose that $P_{\mathcal{M}^\times}\Gamma x = 0$ for some $x \in \mathcal{M}$. Then $x \in \mathcal{M} \cap \Gamma^{-1}\mathcal{M}^\times$. Put $A^\times = A - \Gamma^{-1}BC = \Gamma^{-1}Z\Gamma$. Then $\Gamma^{-1}\mathcal{M}^\times$ is $A^\times$-invariant and H_p-Lagrangian. It follows from Proposition 2.1.1 in [RR3] that $\mathcal{M} \cap \Gamma^{-1}\mathcal{M}^\times = (0)$. ∎

Next, we put

$$W(\lambda) = I + C|_{\mathcal{M}}(\lambda I - A|_{\mathcal{M}})^{-1}(P_{\mathcal{M}^\times}\Gamma|_{\mathcal{M}})^{-1}P_{\mathcal{M}^\times}B.$$

According to Theorem 6.1 in [GK] we have that $W(\lambda)$ is a minimal left divisor of $\Phi(\lambda)$. Moreover, according to the proof of Theorem 6.1 in [GK] it is the minimal left divisor corresponding to the supporting projection Π onto $\Gamma^{-1}\mathcal{M}^\times$ along $\mathcal{M}$, associated with the realization (2.1). Thus $W(\lambda) = I + C(\lambda - A)^{-1}\Pi\Gamma^{-1}B$. Then we may apply Theorem 2.1.2 in [RR3] to see that $W(\lambda)^{-1}\Phi(\lambda) = W(-\bar{\lambda})^*$. Also, all minimal square spectral factors are obtained in this way.

In fact, the remarks above prove the following result.

THEOREM 2.2. *Let $\Phi(\lambda)$ take positive semidefinite values on the imaginary axis, and assume that $\Phi(\infty) = I$. Let $\{(C, A), (Z, B), \Gamma\}$ be a global null-pole triple for Φ, and assume that H_p and H_z satisfy the requirements* (2.2), (2.3), *and* (2.4). *Then there is a one-to-one correspondence between all minimal square spectral factors $W(\lambda)$ of $\Phi(\lambda)$ and pairs of subspaces $\mathcal{M}$, $\mathcal{M}^\times$ that are A-invariant H_p-Lagrangian and Z-invariant H_z-Lagrangian, respectively. This one-to-one correspondence is given by*

$$W(\lambda) = I + C|_{\mathcal{M}}(\lambda - A|_{\mathcal{M}})^{-1}(P_{\mathcal{M}^\times}\Gamma|_{\mathcal{M}})^{-1}P_{\mathcal{M}^\times}B.$$

This result provides a parametrization of all minimal square spectral factors of a positive semidefinite rational matrix function in terms of invariant subspaces. In the next section, we will investigate a parametrization in terms of other invariant subspaces. For the case where $\Gamma = I$ the parametrization above is comparable to the one given in [R1].

3. Parametrization in terms of invariant subspaces II. First, we introduce some notation. We denote the left half plane by $\mathbb{C}_-$, and the right half plane by $\mathbb{C}_+$. Let us denote by W_+ the minimal square factor which is analytic in the open left half plane, $\mathbb{C}_-^0$, and has an analytic inverse there.

Next, we give a brief description of the pole and null pair structure for the (left) minimal square spectral factor, W_+. Let $\tau_+ = \{(C_+, A_+); (Z_+, B_+); \Gamma_+\}$ denote the null-pole triple of $W_+(\lambda)$ corresponding to the open right half plane. Suppose that

$$\tau_0 = \{(C_0, A_0); (Z_0, B_0); \Gamma_0\}$$

is an $i\mathbb{R}$-null-pole triple for W_+. A right pole pair for W_+ may be represented as

$$\left[\begin{pmatrix} C_+ & C_0 \end{pmatrix}, \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix}\right],$$

where $\sigma(A_+) \subset \mathbb{C}_+$ and $\sigma(A_0) \subset i\mathbb{R}$. Also, a left null pair for W_+ may be given by

$$\left[\begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix}, \begin{pmatrix} B_+ \\ B_0 \end{pmatrix}\right],$$

where $\sigma(Z_+) \subset \mathbb{C}_+$ and $\sigma(Z_0) \subset i\mathbb{R}$. Furthermore, we represent the coupling matrix associated with W_+ by

$$\Gamma = \begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}.$$

Note that the formula for $W_+(\lambda)$ may be expressed in realization form as follows (see, e.g., [BR1] and [BGR]):

$$W_+(\lambda) = I + \begin{pmatrix} C_+ & C_0 \end{pmatrix}\left[\lambda I - \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix}\right]^{-1}\begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1}\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}.$$

Here Γ_{12} and Γ_{21} are the unique solutions of the Lyapunov equations

$$\begin{aligned} \Gamma_{21}A_+ - Z_0\Gamma_{21} &= B_0C_+, \\ \Gamma_{12}A_0 - Z_+\Gamma_{12} &= B_+C_0. \end{aligned}$$

Also we have the following representation for the inverse of $W_+(\lambda)$:

$$W_+(\lambda)^{-1} = I - \begin{pmatrix} C_+ & C_0 \end{pmatrix}\begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1}\left[\lambda I - \begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix}\right]^{-1}\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}.$$

For the sake of computations in what follows, it is useful to write

$$\Gamma^{-1} = \begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1} = \Lambda = \begin{pmatrix} \Lambda_+ & \Lambda_{12} \\ \Lambda_{21} & \Lambda_0 \end{pmatrix}.$$

We know, from [R1] (see also [R3]) that if $\lambda_0 \in i\mathbb{R} \cap \sigma(A)$, then all partial multiplicities of A at λ_0 are even and all signs in the sign characteristics of (iA, iH) are $+1$ (see [GLR] for the definition of the sign characteristic). Likewise, if $\lambda_0 \in i\mathbb{R} \cap \sigma(A^\times)$, then all partial multiplicities of $A^\times$ at λ_0 are even and all signs in the sign characteristics of $(iA^\times, iH)$ are -1. The fact that A_0 and Z_0 is the same for all minimal square spectral factors is a direct consequence of this sign condition. In fact, we have the following result which characterizes the $i\mathbb{R}$ null-pole triple for all minimal square spectral factors. Our result is as follows.

LEMMA 3.1. *An $i\mathbb{R}$ null-pole triple for an arbitrary minimal square spectral factor of Φ (modulo similarity), is always given by*

$$\tau_0 = \{(C_0, A_0); (Z_0, B_0); \Gamma_0\}.$$

Proof. Suppose that W_1 and W_2 are any two minimal square spectral factors. Then $\widetilde{U}(\lambda) = W_1(\lambda)^{-1}W_2(\lambda)$ is a unitary matrix, and hence $W_1(\lambda)^{-1}W_2(\lambda)$ is analytic

on $i\mathbb{R}$ and does not have zeros there. Furthermore, by Theorem 4.5.8 of [BGR], the $i\mathbb{R}$ null-pole triples of $W_1(\lambda)$ and $W_2(\lambda)$ are the same. □

In the next theorem we give a parametrization of all minimal square spectral factors of Φ in terms of the triples τ_0 and τ_+. The idea behind the proof is to use τ_0 and τ_+ and the corresponding realization of $W_+(\lambda)$ to first find a minimal realization of $\Phi(\lambda)$ and its inverse $\Phi(\lambda)^{-1}$. Then, using the realization of $W_+(\lambda)^{-1}$ we obtain another minimal realization of $\Phi(\lambda)^{-1}$. We then have two minimal realizations of $\Phi(\lambda)^{-1}$ and may use Kalman's state space isomorphism theorem to obtain a global triple Θ of Φ. Next, we use a result of [GK], combined with Theorem 2.2, to obtain all minimal square spectral factors via certain corestrictions of Θ.

THEOREM 3.2. *The parametrization of all minimal square spectral factors of a positive semidefinite rational matrix function may be described as follows. Suppose that*

$$W_+(\lambda) = I + \begin{pmatrix} C_+ & C_0 \end{pmatrix} \left[\lambda I - \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix} \right]^{-1} \begin{pmatrix} \Lambda_+ & \Lambda_{12} \\ \Lambda_{21} & \Lambda_0 \end{pmatrix} \begin{pmatrix} B_+ \\ B_0 \end{pmatrix}$$

is a minimal realization of the left canonical spectral factor. Furthermore, assume that P and P_1 are the unique solutions of the Lyapunov equations

$$\begin{aligned} A_+P + PA_+^* &= (\Lambda_+B_+ + \Lambda_{12}B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^*, \\ A_0P_1 + P_1A_+^* &= (\Lambda_{21}B_+ + \Lambda_0B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^*, \end{aligned} \tag{3.1}$$

respectively. Also, let Q and Q_1 be the unique solutions of the Lyapunov equations

$$\begin{aligned} Z_+^*Q + QZ_+ &= -(C_+\Lambda_+ + C_0\Lambda_{21})^*(C_+\Lambda_+ + C_0\Lambda_{21}), \\ Z_0^*Q_1 + Q_1Z_+ &= -(C_+\Lambda_{12} + C_0\Lambda_0)^*(C_+\Lambda_+ + C_0\Lambda_{21}), \end{aligned} \tag{3.2}$$

respectively. Suppose that $\mathcal{N}$ is an A_+^*-invariant subspace and $\mathcal{N}^\times$ is a Z_+^*-invariant subspace. Furthermore, let $P_{\mathcal{N}^\times}$ denote the orthogonal projection onto $\mathcal{N}^\times$ and let $P_{\mathcal{N}}$ denote the orthogonal projection onto $\mathcal{N}$. Then every minimal square spectral factor may be expressed as

$$\begin{aligned} W(\lambda) = I + \begin{pmatrix} C_+|_{\mathcal{N}^\perp} & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*|_{\mathcal{N}} \end{pmatrix} \\ \cdot \left[\lambda I - \begin{pmatrix} A_+|_{\mathcal{N}^\perp} & 0 & 0 \\ 0 & A_0 & 0 \\ 0 & 0 & -A_+^*|_{\mathcal{N}} \end{pmatrix} \right]^{-1} \Gamma_W^{-1} \\ \cdot \begin{pmatrix} (I - P_{\mathcal{N}^\times})B_+ \\ B_0 \\ P_{\mathcal{N}^\times}(QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^*) \end{pmatrix}, \end{aligned} \tag{3.3}$$

where Γ_W is the matrix

$$\begin{pmatrix} (I - P_{\mathcal{N}^\times})\Gamma_+(I - P_{\mathcal{N}}) & (I - P_{\mathcal{N}^\times})\Gamma_{12} & (I - P_{\mathcal{N}^\times})(\Gamma_+P + \Gamma_{12}P_1)P_{\mathcal{N}} \\ \Gamma_{21}(I - P_{\mathcal{N}}) & \Gamma_0 & (\Gamma_{21}P + \Gamma_0P_1)P_{\mathcal{N}} \\ -P_{\mathcal{N}^\times}(Q\Gamma_+ + Q_1^*\Gamma_{21})(I - P_{\mathcal{N}}) & -P_{\mathcal{N}^\times}(Q\Gamma_{12} + Q_1^*\Gamma_0) & P_{\mathcal{N}^\times}(\Lambda_+^* + (Q\Gamma_+ + Q_1^*\Gamma_{21})P + (Q\Gamma_{12} + Q_1^*\Gamma_0)P_1)P_{\mathcal{N}} \end{pmatrix}.$$

Also, the inverse of $W(\lambda)$ may be expressed as

$$
\begin{aligned}
W(\lambda)^{-1} &= I - \begin{pmatrix} C_+|_{\mathcal{N}^\perp} & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*|_{\mathcal{N}} \end{pmatrix}\Gamma_W^{-1} \\
&\quad \cdot \left[\lambda I - \begin{pmatrix} (I - P_{\mathcal{N}^\times})Z_+|_{\mathcal{N}^{\times\perp}} & 0 & 0 \\ 0 & Z_0 & 0 \\ 0 & 0 & -P_{\mathcal{N}^\times}Z_+^*|_{\mathcal{N}^\times} \end{pmatrix}\right]^{-1} \\
&\quad \cdot \begin{pmatrix} (I - P_{\mathcal{N}^\times})B_+ \\ B_0 \\ P_{\mathcal{N}^\times}(QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^*) \end{pmatrix}.
\end{aligned} \tag{3.4}
$$

Proof. First, we compute $\Phi(\lambda) = W_+(\lambda)W_+(-\bar{\lambda})^*$ as

$$
\begin{aligned}
\Phi(\lambda) &= I + C(\lambda I - A)^{-1}B = I + \begin{pmatrix} C_+ & C_0 & -\begin{pmatrix} B_+^* & B_0^* \end{pmatrix}\Lambda^* \end{pmatrix} \\
&\quad \cdot \left[\lambda I - \begin{pmatrix} A_+ & 0 & -\Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}\begin{pmatrix} B_+^* & B_0^* \end{pmatrix}\Lambda^* \\ 0 & A_0 & \\ 0 & 0 & -A_+^* \quad 0 \\ 0 & 0 & 0 \quad -A_0^* \end{pmatrix}\right]^{-1} \begin{pmatrix} \Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix} \\ C_+^* \\ C_0^* \end{pmatrix}.
\end{aligned} \tag{3.5}
$$

Observe that with

$$
H = \begin{pmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{pmatrix}
$$

we have $HA = -A^*H$ and $HB = C^*$. Moreover, H is skew-Hermitian.

From an earlier observation, we recall that

$$
\begin{aligned}
W_+(\lambda)^{-1} &= I - \begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda\left[\lambda I - \begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix}\right]^{-1}\begin{pmatrix} B_+ \\ B_0 \end{pmatrix} \\
&= I - \begin{pmatrix} C_+ & C_0 \end{pmatrix}\left[\lambda I - \Lambda\begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix}\Gamma\right]^{-1}\Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}.
\end{aligned}
$$

It is clear that we are able to find an explicit formula for $\Phi(\lambda)^{-1}$ by using these expressions for $W_+(\lambda)^{-1}$. For the associated main operator appearing in the formula (3.5) for $\Phi(\lambda)^{-1}$, we have

$$
A^\times = \begin{pmatrix} A_+ & 0 & -\Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}\begin{pmatrix} B_+^* & B_0^* \end{pmatrix}\Lambda^* \\ 0 & A_0 & \\ 0 & 0 & -A_+^* \quad 0 \\ 0 & 0 & 0 \quad -A_0^* \end{pmatrix} - \begin{pmatrix} \Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix} \\ C_+^* \\ C_0^* \end{pmatrix}\begin{pmatrix} C_+ & C_0 & -\begin{pmatrix} B_+^* & B_0^* \end{pmatrix}\Lambda^* \end{pmatrix},
$$

which, after using

$$
\begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix}\Lambda - \Lambda\begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix} = \Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}\begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda,
$$

is seen to equal

$$
A^\times = \begin{pmatrix} \Lambda\begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix}\Gamma & 0 \\ -\begin{pmatrix} C_+^* \\ C_0^* \end{pmatrix}\begin{pmatrix} C_+ & C_0 \end{pmatrix} & -\Gamma^*\begin{pmatrix} Z_+^* & 0 \\ 0 & Z_0^* \end{pmatrix}\Lambda^* \end{pmatrix}.
$$

Hence, we may express $\Phi(\lambda)^{-1}$ as

$$\Phi(\lambda)^{-1} = I - \begin{pmatrix} C_+ & C_0 & -(\Lambda_+ B_+ + \Lambda_{12} B_0)^* & -(\Lambda_{21} B_+ + \Lambda_0 B_0)^* \end{pmatrix}$$

$$\left[\lambda I - \begin{pmatrix} \Lambda \begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix} \Gamma & 0 \\ -\begin{pmatrix} C_+^* \\ C_0^* \end{pmatrix} \begin{pmatrix} C_+ & C_0 \end{pmatrix} & -\Gamma^* \begin{pmatrix} Z_+^* & 0 \\ 0 & Z_0^* \end{pmatrix} \Lambda^* \end{pmatrix}\right]^{-1} \begin{pmatrix} \Lambda_+ B_+ + \Lambda_{12} B_0 \\ \Lambda_{21} B_+ + \Lambda_0 B_0 \\ C_+^* \\ C_0^* \end{pmatrix}. \tag{3.6}$$

Utilizing the relations above, we are able to determine a more transparent formula for Φ. Let

$$S = \begin{pmatrix} I & 0 & P & P_1^* \\ 0 & I & P_1 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}.$$

By using the equations in (3.1), we are able to compute that

$$S^{-1} A S = \begin{pmatrix} A_+ & 0 & 0 & 0 \\ 0 & A_0 & 0 & A_{24} \\ 0 & 0 & -A_+^* & 0 \\ 0 & 0 & 0 & -A_0^* \end{pmatrix},$$

where $A_{24} = -(\Lambda_{21} B_+ + \Lambda_0 B_0)(\Lambda_{21} B_+ + \Lambda_0 B_0)^*$. Thus, we may rewrite the expression for $\Phi(\lambda)$, appearing in (3.5), as

$$\Phi(\lambda) = I + \begin{pmatrix} C_+ & C_0 & C_+ P + C_0 P_1 - (\Lambda_+ B_+ + \Lambda_{12} B_0)^* & C_+ P_1^* - (\Lambda_{21} B_+ + \Lambda_0 B_0)^* \end{pmatrix}$$

$$\left[\lambda I - \begin{pmatrix} A_+ & 0 & 0 & 0 \\ 0 & A_0 & 0 & A_{24} \\ 0 & 0 & -A_+^* & 0 \\ 0 & 0 & 0 & -A_0^* \end{pmatrix}\right]^{-1} \begin{pmatrix} \Lambda_+ B_+ + \Lambda_{12} B_0 - P C_+^* - P_1 C_0^* \\ \Lambda_{21} B_+ + \Lambda_0 B_0 - P_1 C_+^* \\ C_+^* \\ C_0^* \end{pmatrix}. \tag{3.7}$$

In what follows, we put $H_p = S^* H S$. Note that H_p is skew-Hermitian.

We use Q and Q_1 to find a more suitable formula for Φ^{-1} given by (3.6). Put

$$T = \begin{pmatrix} \Lambda & 0 \\ \Gamma^* \begin{pmatrix} Q & Q_1^* \\ Q_1 & 0 \end{pmatrix} & \Gamma^* \end{pmatrix}.$$

Using the Lyapunov equations in (3.2), we have

$$T^{-1} A^{\times} T = \begin{pmatrix} Z_+ & 0 & 0 & 0 \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & -Z_+^* & 0 \\ 0 & Z_{42} & 0 & -Z_0^* \end{pmatrix},$$

where $Z_{42} = -(C_0 \Lambda_0 + C_+ \Lambda_{12})^* (C_0 \Lambda_0 + C_+ \Lambda_{12})$. Thus, we have

$$\Phi(\lambda)^{-1} = I - \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \end{pmatrix} \left[\lambda I - \begin{pmatrix} Z_+ & 0 & 0 & 0 \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & -Z_+^* & 0 \\ 0 & Z_{42} & 0 & -Z_0^* \end{pmatrix}\right]^{-1}$$

$$\begin{pmatrix} B_+ \\ B_0 \\ Q B_+ + Q_1^* B_0 + (C_+ \Lambda_+ + C_0 \Lambda_{21})^* \\ Q_1 B_+ + (C_+ \Lambda_{12} + C_0 \Lambda_0)^* \end{pmatrix}, \tag{3.8}$$

where $\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \end{pmatrix} = \begin{pmatrix} C_+ & C_0 & -\begin{pmatrix} B_+^* & B_0^* \end{pmatrix}\Lambda^* \end{pmatrix} T$.

We also introduce $H_z = T^* H T$. Obviously, H_z is skew-Hermitian.

By considering formulas (3.7) and (3.8) we may deduce a right pole pair and a left null pair for Φ. As a right pole pair for Φ, from (3.7), we have

$$\left[\begin{pmatrix} C_+ & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* & C_+P_1^* - (\Lambda_{21}B_+ + \Lambda_0B_0)^* \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} A_+ & 0 & 0 & 0 \\ 0 & A_0 & 0 & A_{24} \\ 0 & 0 & -A_+^* & 0 \\ 0 & 0 & 0 & -A_0^* \end{pmatrix} \right].$$

From (3.8), a left null pair for Φ may be given by

$$\left[\begin{pmatrix} Z_+ & 0 & 0 & 0 \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & -Z_+^* & 0 \\ 0 & Z_{42} & 0 & -Z_0^* \end{pmatrix}, \begin{pmatrix} B_+ \\ B_0 \\ QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ Q_1B_+ + (C_+\Lambda_{12} + C_0\Lambda_0)^* \end{pmatrix} \right].$$

Moreover, by considering (3.7), an alternative expression for (3.8) will be

$$\Phi(\lambda)^{-1} = I + \begin{pmatrix} C_+ & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* & C_+P_1^* - (\Lambda_{21}B_+ + \Lambda_0B_0)^* \end{pmatrix}$$
$$(\lambda I - \widetilde{A})^{-1} \begin{pmatrix} \Lambda_+B_+ + \Lambda_{12}B_0 - PC_+^* - P_1C_0^* \\ \Lambda_{21}B_+ + \Lambda_0B_0 - P_1C_+^* \\ C_+^* \\ C_0^* \end{pmatrix},$$

where the associate matrix is given by

$$\widetilde{A} = \begin{pmatrix} A_+ & 0 & 0 & 0 \\ 0 & A_0 & 0 & A_{24} \\ 0 & 0 & -A_+^* & 0 \\ 0 & 0 & 0 & -A_0^* \end{pmatrix} - \begin{pmatrix} \Lambda_+B_+ + \Lambda_{12}B_0 - PC_+^* - P_1C_0^* \\ \Lambda_{21}B_+ + \Lambda_0B_0 - P_1C_+^* \\ C_+^* \\ C_0^* \end{pmatrix}$$
$$\cdot \begin{pmatrix} C_+ & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* & C_+P_1^* - (\Lambda_{21}B_+ + \Lambda_0B_0)^* \end{pmatrix}.$$

It is important to note that both of these realizations for $\Phi(\lambda)^{-1}$ can be shown to be minimal. In this case, it is well known that they are similar. The matrix that gives the similarity is given by

$$\Gamma_\Phi = T^{-1}S = \begin{pmatrix} \Gamma & \Gamma \begin{pmatrix} P & P_1^* \\ P_1 & 0 \end{pmatrix} \\ -\begin{pmatrix} Q & Q_1^* \\ Q_1 & 0 \end{pmatrix}\Gamma & \Lambda^* - \begin{pmatrix} Q & Q_1^* \\ Q_1 & 0 \end{pmatrix}\Gamma\begin{pmatrix} P & P_1^* \\ P_1 & 0 \end{pmatrix} \end{pmatrix}.$$

Hence, a global null-pole triple for Φ may be represented by

$$\Theta = \left\{ \left[\begin{array}{cccc} \big(C_+ & C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* & C_+P_1^* - (\Lambda_{21}B_+ + \Lambda_0B_0)^*\big), \\ & & & \\ & & & \begin{pmatrix} A_+ & 0 & 0 & 0 \\ 0 & A_0 & 0 & A_{24} \\ 0 & 0 & -A_+^* & 0 \\ 0 & 0 & 0 & -A_0^* \end{pmatrix} \end{array} \right]; \right.$$
$$\left. \left[\begin{pmatrix} Z_+ & 0 & 0 & 0 \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & -Z_+^* & 0 \\ 0 & Z_{42} & 0 & Z_0^* \end{pmatrix}, \begin{pmatrix} B_+ \\ B_0 \\ QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ Q_1B_+ + (C_+\Lambda_{12} + C_0\Lambda_0)^* \end{pmatrix} \right]; \Gamma_\Phi \right\}.$$

Also, the matrices H_p and H_z satisfy the conditions (2.2), (2.3), and (2.4) for the realization of $\Phi(\lambda)$ connected to this global null-pole triple.

Now, let $\mathcal{N}$ be A_+^*-invariant, and let $\mathcal{N}^\times$ be Z_+^*-invariant. Denote by $P_\mathcal{N}$ and $P_{\mathcal{N}^\times}$ the orthogonal projections onto $\mathcal{N}$ and $\mathcal{N}^\times$, respectively. Put

$$\mathcal{M} = \left\{ \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \mid x \in \mathcal{N}^\perp, y \text{ arbitrary }, z \in \mathcal{N} \right\} \tag{3.9}$$

and $\widetilde{\mathcal{M}} = S^{-1}\mathcal{M}$. Then $\widetilde{\mathcal{M}}$ is A-invariant and H-Lagrangian. The latter assertion is readily checked from the fact that $\mathcal{M}$ is H_p-Lagrangian.

Also, we put

$$\mathcal{M}^\times = \left\{ \begin{bmatrix} x \\ 0 \\ z \\ w \end{bmatrix} \mid x \in \mathcal{N}^{\times\perp}, z \in \mathcal{N}^\times, w \text{ arbitrary} \right\} \tag{3.10}$$

and $\widetilde{\mathcal{M}}^\times = T^{-1}\mathcal{M}^\times$. Then $\widetilde{\mathcal{M}}^\times$ is $A^\times$-invariant and H-Lagrangian, as $\mathcal{M}^\times$ is H_z-Lagrangian.

By [RR3] we have that $\widetilde{\mathcal{M}} \oplus \widetilde{\mathcal{M}}^\times$ equals the whole state space. According to [BGK] and [RR3] (see also [R1]) the supporting projection $\widetilde{\Pi}$ along $\widetilde{\mathcal{M}}$ onto $\widetilde{\mathcal{M}}^\times$ gives rise to a minimal square spectral factorization. A formula for the corresponding factor $W(\lambda)$ may be obtained from the realization (3.7) and a formula for $\widetilde{\Pi}$, which may be derived in principle from the explicit representations of $\widetilde{\mathcal{M}}$ and $\widetilde{\mathcal{M}}^\times$.

However, we can also derive the formula for $W(\lambda)$ in another way, namely, by using [GK]. Indeed, a global null-pole triple for $W(\lambda)$ may be obtained as a corestriction of Θ. More specifically, the corestriction is connected to $\mathcal{M}$ and $\mathcal{M}^\times$ precisely in the way described in Theorem 2.2 in the previous section. The corestriction of Θ that is

involved is the following one:

$$\left\{\left[\begin{array}{c}\big(C_+|_{\mathcal{N}^\perp} \quad C_0 \quad (C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*)|_{\mathcal{N}}\big), \\ \begin{pmatrix} A_+|_{\mathcal{N}^\perp} & 0 & 0 \\ 0 & A_0 & 0 \\ 0 & 0 & -A_+^*|_{\mathcal{N}} \end{pmatrix}\end{array}\right], \\ \left[\begin{pmatrix} (I-P_{\mathcal{N}^\times})A_+|_{\mathcal{N}^{\times\perp}} & 0 & 0 \\ 0 & Z_0 & 0 \\ 0 & 0 & -P_{\mathcal{N}^\times}Z_+^*|_{\mathcal{N}^\times} \end{pmatrix}, \\ \left[\begin{pmatrix} (I-P_{\mathcal{N}^\times})B_+ \\ B_0 \\ P_{\mathcal{N}^\times}(QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^*) \end{pmatrix}\right], \Gamma_W\right\},$$

where Γ_W is given by

$$\Gamma_W = \begin{pmatrix} I-P_{\mathcal{N}^\times} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & P_{\mathcal{N}^\times} & 0 \end{pmatrix}\Gamma_\Phi \begin{pmatrix} I-P_{\mathcal{N}} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & P_{\mathcal{N}} \\ 0 & 0 & 0 \end{pmatrix},$$

which is equal to the Γ_W given in the statement of this theorem.

For the converse, if $W(\lambda)$ is a minimal square spectral factor, then there is a corresponding corestriction of Θ. This corestriction is connected to two subspaces $\mathcal{M}$ and $\mathcal{M}^\times$ as in Theorem 2.2, but it is now specialized to the null-pole triple Θ. Clearly, keeping Lemma 3.1 in mind, $\mathcal{M}$ must have the form (3.9) and $\mathcal{M}^\times$ must have the form (3.10) for some $\mathcal{N}$ that is A_+^*-invariant and some $\mathcal{N}^\times$ that is Z_+^*-invariant. It then follows that $W(\lambda)$ is given by (3.3). □

4. Left versus right minimal square spectral factors. In this section, it is our aim to solve the following problem: Suppose that $W_+(\lambda)$ is a rational matrix function that has all its zeros and poles in the closed right half plane. Here it is explicitly allowed that $W_+(\lambda)$ has zeros and poles on the imaginary axis. We form the rational matrix function $\Phi(\lambda) = W_+(\lambda)W_+(-\bar{\lambda})^*$. Then $\Phi(\lambda)$ has positive semidefinite values on the imaginary axis (except for possible poles). We wish to find a rational matrix function, $W_-(\lambda)$, having all its poles and zeros in the closed left half plane and satisfying $\Phi(\lambda) = W_-(\lambda)W_-(-\bar{\lambda})^*$. In other words, given a left spectral factor we wish to find a right spectral factor. That such a $W_-(\lambda)$ exists is an easy consequence of [R1]. The problem is solved in [BR2] for the case where $W_+(\lambda)$, and hence also $\Phi(\lambda)$, has no poles and zeros on the imaginary axis. Here we consider the problem for the case where there are zeros and poles in the imaginary axis.

As in section 3, let $\tau_0 = \{(C_0, A_0); (Z_0, B_0); \Gamma_0\}$ denote the $i\mathbb{R}$ null-pole triple of $W_+(\lambda)$ and $\tau_+ = \{(C_+, A_+); (Z_+, B_+); \Gamma_+\}$ denote the null-pole triple of $W_+(\lambda)$ corresponding to the open right half plane. As we have seen before, we may represent a left minimal square spectral factor by

$$W_+(\lambda) = I + \begin{pmatrix} C_+ & C_0 \end{pmatrix}\left[\lambda I - \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix}\right]^{-1}\Lambda\begin{pmatrix} B_+ \\ B_0 \end{pmatrix}.$$

Also, for the inverse, $W_+(\lambda)^{-1}$, we have the realization

$$W_+(\lambda)^{-1} = I - \begin{pmatrix} C_+ & C_0 \end{pmatrix} \Lambda \left[\lambda I - \begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix} \right]^{-1} \begin{pmatrix} B_+ \\ B_0 \end{pmatrix}.$$

To find $W_-(\lambda)$ and its inverse, in terms of all the matrices appearing in these realizations, obviously we may use the formulas from section 3, by taking $\mathcal{N}$ and $\mathcal{N}^\times$ in such a way that $P_{\mathcal{N}} = I$ and $P_{\mathcal{N}^\times} = I$. This yields

$$W_-(\lambda) = I + \begin{pmatrix} C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix} \left[\lambda I - \begin{pmatrix} A_0 & 0 \\ 0 & -A_+^* \end{pmatrix} \right]^{-1} \Gamma_{W_-}^{-1} \cdot \begin{pmatrix} B_0 \\ QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix},$$

where

$$\Gamma_{W_-} = \begin{pmatrix} \Gamma_0 & \Gamma_{21}P + \Gamma_0P_1 \\ Q\Gamma_{12} + Q_1^*\Gamma_0 & \Lambda_+^* + Q\Gamma_+P + Q_1\Gamma_{21}P + Q\Gamma_{12}P_1 + Q_1^*\Gamma_0P_1 \end{pmatrix}.$$

Also, for its inverse we have

$$W_-(\lambda)^{-1} = I - \begin{pmatrix} C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix} \Gamma_{W_-}^{-1} \left[\lambda I - \begin{pmatrix} Z_0 & 0 \\ 0 & -Z_+^* \end{pmatrix} \right]^{-1} \cdot \begin{pmatrix} B_0 \\ QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix}.$$

5. Parametrization in terms of unitary divisors. Throughout this discussion, we consider the rational matrix function

$$U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$$

which has unitary values on $i\mathbb{R}$. The function $U(\lambda)$ here is not the phase function because of our choice of $W_+(\lambda)$. It has poles in both the left half plane and the right half plane. So $U(\lambda)$ is not an inner function. The results in this section are close in spirit to [FMP], [F2], and [R2]. In this case, as in the previous sections, we take

$$W_+(\lambda)^{-1} = I - \begin{pmatrix} C_+ & C_0 \end{pmatrix} \Lambda \left[\lambda I - \begin{pmatrix} Z_+ & 0 \\ 0 & Z_0 \end{pmatrix} \right]^{-1} \begin{pmatrix} B_+ \\ B_0 \end{pmatrix}, \tag{5.1}$$

where

$$\Lambda = \begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1} = \begin{pmatrix} \Lambda_+ & \Lambda_{12} \\ \Lambda_{21} & \Lambda_0 \end{pmatrix}.$$

Also, we have

$$W_-(\lambda) = I + \begin{pmatrix} C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix} \left[\lambda I - \begin{pmatrix} A_0 & 0 \\ 0 & -A_+^* \end{pmatrix} \right]^{-1} \Gamma_{W_-}^{-1} \cdot \begin{pmatrix} B_0 \\ QB_+ + Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix}, \tag{5.2}$$

where

$$\Gamma_{W_-} = \begin{pmatrix} \Gamma_0 & \Gamma_{21}P + \Gamma_0P_1 \\ Q\Gamma_{12} + Q_1^*\Gamma_0 & \Lambda_+^* + Q\Gamma_+P + Q_1\Gamma_{21}P + Q\Gamma_{12}P_1 + Q_1^*\Gamma_0P_1 \end{pmatrix}.$$

Also, we recall from section 3 that the following Lyapunov equations hold:

$$B_+C_+ = \Gamma_+A_+ - Z_+\Gamma_+, \tag{5.3}$$

$$B_0C_0 = \Gamma_0A_0 - Z_0\Gamma_0, \tag{5.4}$$

$$B_+C_0 = \Gamma_{12}A_0 - Z_+\Gamma_{12}, \tag{5.5}$$

$$B_0C_+ = \Gamma_{21}A_+ - Z_0\Gamma_{21}, \tag{5.6}$$

$$A_0P_1 + P_1A_+^* = (\Lambda_{21}B_+ + \Lambda_0B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^*, \tag{5.7}$$

$$A_+P + PA_+^* = (\Lambda_+B_+ + \Lambda_{12}B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^*, \tag{5.8}$$

$$Z_+^*Q + QZ_+ = -(C_+\Lambda_+ + C_0\Lambda_{21})^*(C_+\Lambda_+ + C_0\Lambda_{21}), \tag{5.9}$$

$$Z_0^*Q_1 + Q_1Z_+ = -(C_+\Lambda_{12} + C_0\Lambda_0)^*(C_+\Lambda_+ + C_0\Lambda_{21}). \tag{5.10}$$

These Lyapunov equations will be used extensively in what follows. The main assertion that will be investigated in this section may be stated as follows.

THEOREM 5.1. *There is a one-to-one correspondence between the minimal factorizations of $U(\lambda)$ into two unitary factors and the set of minimal square spectral factors. This correspondence may be described as follows. If*

$$U(\lambda) = U_1(\lambda)U_2(\lambda)$$

is a minimal factorization with U_1 and U_2 being unitary factors, then

$$W(\lambda) = W_+(\lambda)U_1(\lambda)^{-1}$$

is a minimal square spectral factor. Moreover, any minimal square factor is obtained in this way. More precisely, let

$$W_+(\lambda) = I + \begin{pmatrix} C_+ & C_0 \end{pmatrix}\left(\lambda - \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix}\right)^{-1}\begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1}\begin{pmatrix} B_+ & B_0 \end{pmatrix}$$

be a minimal realization and put

$$Y = \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{pmatrix}.$$

Then there is also a one-to-one correspondence between the set of invariant subspaces of Y and the set of minimal square spectral factors.

These one-to-one correspondences may be given as follows. Let $\mathcal{N}^\times \oplus \mathcal{N}$ be a Y-invariant subspace, and let U_1 be given by

$$\begin{aligned} U_1(\lambda) = I - \begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})|_{\mathcal{N}^\times} & (\Lambda_{12}B_0 + \Lambda_+B_+)^*|_{\mathcal{N}} \end{pmatrix}\left[\lambda I - \begin{pmatrix} Z_+|_{\mathcal{N}^\times} & 0 \\ 0 & -A_+^*|_{\mathcal{N}} \end{pmatrix}\right]^{-1} \\ \cdot T^{-1}\begin{pmatrix} ((C_+\Lambda_+ + C_0\Lambda_{21})|_{\mathcal{N}^\times})^* \\ ((\Lambda_{12}B_0 + \Lambda_+B_+)^*|_{\mathcal{N}})^* \end{pmatrix}, \end{aligned} \tag{5.11}$$

where T is the unique solution of

$$T\begin{pmatrix} Z_+|_{\mathcal{N}^\times} & 0 \\ 0 & -A_+^*|_{\mathcal{N}} \end{pmatrix} + \begin{pmatrix} (Z_+|_{\mathcal{N}^\times})^* & 0 \\ 0 & (-A_+^*|_{\mathcal{N}})^* \end{pmatrix} T$$

(5.12)

$$= \begin{pmatrix} ((C_+\Lambda_+ + C_0\Lambda_{21})|_{\mathcal{N}^\times})^* \\ ((\Lambda_{12}B_0 + \Lambda_+B_+)^*|_{\mathcal{N}})^* \end{pmatrix} \big((C_+\Lambda_+ + C_0\Lambda_{21})|_{\mathcal{N}^\times} \quad (\Lambda_{12}B_0 + \Lambda_+B_+)^*|_{\mathcal{N}}\big).$$

Then $U_1(\lambda)$ is a minimal left unitary factor of $U(\lambda)$ and $W(\lambda) = W_+(\lambda)U_1(\lambda)$ is a minimal square spectral factor. In fact, any minimal square spectral factor is obtained in this way.

Proof.

Step 1. *An explicit formula for $U(\lambda)$.*

We use the formulas for $W_+(\lambda)^{-1}$ and $W_-(\lambda)$ given in (5.1) and (5.2), respectively, and Lyapunov equations (5.3)–(5.10) to give a formula for $U(\lambda)$. In what follows, we denote

$$\widetilde{A} = \begin{pmatrix} Z_+ & 0 & B_+C_0 & B_+(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*) \\ 0 & Z_0 & B_0C_0 & B_0(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*) \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix},$$

$$\widetilde{C} = \begin{pmatrix} -\big(C_+ & C_0\big)\Lambda & \big(C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*\big) \end{pmatrix},$$

$$\text{and } \widetilde{B} = \begin{pmatrix} B_+ \\ B_0 \\ B_0 \\ \Gamma_{W_-}^{-1}\begin{pmatrix} -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \end{pmatrix}.$$

Then, we see that

$$U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda) = I + \tilde{C}(\lambda I - \tilde{A})^{-1}\tilde{B}. \tag{5.13}$$

We have to ensure that the terms involving A_0 and Z_0 cancel as $U(\lambda)$ is unitary on the imaginary axis. We put

$$R = \begin{pmatrix} I & 0 & \Gamma_{12} & 0 \\ 0 & I & \Gamma_0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix},$$

$\widetilde{A}_{14} = B_+(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*)$, and $\widetilde{A}_{13} = B_0(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*)$. By using Lyapunov equations (5.4) and (5.5), we see that

$$R^{-1}\widetilde{A}R = \begin{pmatrix} I & 0 & -\Gamma_{12} & 0 \\ 0 & I & -\Gamma_0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} Z_+ & 0 & B_+C_0 & \widetilde{A}_{14} \\ 0 & Z_0 & B_0C_0 & \widetilde{A}_{13} \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix} \begin{pmatrix} I & 0 & \Gamma_{12} & 0 \\ 0 & I & \Gamma_0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$
$$= \begin{pmatrix} I & 0 & -\Gamma_{12} & 0 \\ 0 & I & -\Gamma_0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} Z_+ & 0 & B_+C_0 + Z_+\Gamma_{12} & \widetilde{A}_{14} \\ 0 & Z_0 & B_0C_0 + Z_0\Gamma_0 & \widetilde{A}_{13} \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix}$$
$$= \begin{pmatrix} Z_+ & 0 & 0 & \widetilde{A}_{14} \\ 0 & Z_0 & 0 & \widetilde{A}_{13} \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix} = \widehat{A}.$$

Also, it is clear that

$$\widehat{C} = \widetilde{C}R = \begin{pmatrix} -\begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda & \begin{pmatrix} C_0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix} \end{pmatrix} \cdot \begin{pmatrix} I & 0 & \Gamma_{12} & 0 \\ 0 & I & \Gamma_0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$
$$= \begin{pmatrix} -\begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda & 0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix}$$

and

$$\widehat{B} = R^{-1}\widetilde{B} = \begin{pmatrix} \begin{pmatrix} B_+ \\ B_0 \end{pmatrix} - \begin{pmatrix} \Gamma_{12} & 0 \\ \Gamma_0 & 0 \end{pmatrix}\Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \\ \Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \end{pmatrix}.$$

From $\widetilde{A}_{13} = B_0(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*)$ and Lyapunov equations (5.4), (5.6), (5.7), and (5.8) we see that

$$\begin{aligned} B_0C_+P + B_0C_0P_1 &= (\Gamma_{21}A_+ - Z_0\Gamma_{21})P + (\Gamma_0A_0 - Z_0\Gamma_0)P_1 \\ &= \Gamma_{21}A_+P - Z_0(\Gamma_{21}P + \Gamma_0P_1) + \Gamma_0A_0P_1 \\ &= -\Gamma_{21}PA_+^* + \Gamma_{21}(\Lambda_+B_+ + \Lambda_{12}B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &\quad -Z_0(\Lambda_{21}P + \Lambda_0P_1) - \Gamma_0P_1A_+^* \\ &\quad +\Gamma_0(\Lambda_{21}B_+ + \Lambda_0B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &= -(\Gamma_{21}P + \Gamma_0P_1)A_+^* - Z_0(\Gamma_{21}P + \Gamma_0P_1) \\ &\quad +\{(\Gamma_{21}\Lambda_+ + \Gamma_0\Lambda_{21})B_+ + (\Gamma_{21}\Lambda_{12} + \Gamma_0\Lambda_0)B_0\}(\Lambda_+B_+ + \Lambda_{12}B_0)^*. \end{aligned}$$

It is immediate that $\widetilde{A}_{13} = -(\Gamma_{21}P + \Gamma_0P_1)A_+^* - Z_0(\Gamma_{21}P + \Gamma_0P_1)$.

Next, we put

$$S = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & \Gamma_{21}P + \Gamma_0 P_1 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}.$$

Then it is clear that

$$\begin{aligned} S^{-1}\widehat{A}S &= \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & -(\Gamma_{21}P + \Gamma_0 P_1) \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} Z_+ & 0 & 0 & \widetilde{A}_{14} \\ 0 & Z_0 & 0 & \widetilde{A}_{13} + Z_0(\Gamma_{21}P + \Gamma_0 P_1) \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix} \\ &= \begin{pmatrix} Z_+ & 0 & 0 & \widetilde{A}_{14} \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & A_0 & 0 \\ 0 & 0 & 0 & -A_+^* \end{pmatrix}, \\ \widehat{C}S &= \begin{pmatrix} -\begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda & 0 & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* \end{pmatrix} \\ &\quad \cdot \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & \Gamma_{21}P + \Gamma_0 P_1 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \\ &= \begin{pmatrix} -\begin{pmatrix} C_+ & C_0 \end{pmatrix}\Lambda & 0 \\ & C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* - (C_+\Gamma_{12} + C_0\Gamma_0)(\Gamma_{21}P + \Gamma_0 P_1)) \end{pmatrix}, \end{aligned}$$

and

$$S^{-1}\widehat{B} = \begin{pmatrix} B_+ - \begin{pmatrix} \Gamma_{12} & 0 \end{pmatrix}\Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \\ B_0 - \begin{pmatrix} \Gamma_0 & \Gamma_{21}P + \Gamma_0 P_1 \end{pmatrix}\Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \\ \Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \end{pmatrix}.$$

Moreover, we see that

$$B_0 - \begin{pmatrix} \Gamma_0 & \Gamma_{21}P + \Gamma_0 P_1 \end{pmatrix}\Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} = 0,$$

as $\begin{pmatrix} \Gamma_0 & \Gamma_{21}P + \Gamma_0 P_1 \end{pmatrix}$ is the first row in Γ_{W_-}, so that

$$S^{-1}\widehat{B} = \begin{pmatrix} B_+ - \begin{pmatrix} \Gamma_{12} & 0 \end{pmatrix}\Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \\ 0 \\ \Gamma_{W_-}^{-1} \begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \end{pmatrix}.$$

So we may conclude, from the similarity transformations above, that (5.13) may be rewritten as

$$\begin{aligned} U(\lambda) &= W_+(\lambda)^{-1}W_-(\lambda) \\ &= I + (-C_+\Lambda_+ - C_0\Lambda_{21} \\ &\quad C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* - \ (C_+\Lambda_{12} + C_0\Lambda_0)(\Gamma_{21}P + \Gamma_0P_1)) \end{aligned}$$

$$\cdot \left[\lambda I - \begin{pmatrix} Z_+ & \widetilde{A}_{14} \\ 0 & -A_+^* \end{pmatrix}\right]^{-1} \begin{pmatrix} B_+ - \begin{pmatrix} \Gamma_{12} & 0 \end{pmatrix}\Gamma_{W_-}^{-1}\begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \\ \begin{pmatrix} 0 & I \end{pmatrix}\Gamma_{W_-}^{-1}\begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^* \end{pmatrix} \end{pmatrix}.$$

(5.14)

Here we note that $\sigma(Z_+) \cap \sigma(-A_+^*) = \emptyset$. So we are able to cancel the term $\widetilde{A}_{14}$ and end up with a block diagonal main operator in the following way. From $\widetilde{A}_{14} = B_+(C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^*)$ and Lyapunov equations (5.3), (5.5), (5.7), and (5.8) we see that

$$\begin{aligned} B_+C_+P + B_+C_0P_1 &= \Gamma_+A_+P + \Gamma_{12}A_0P_1 - Z_+(\Gamma_+P + \Gamma_{12}P_1) \\ &= -\Gamma_+PA_+^* + \Gamma_+(\Lambda_+B_+ + \Lambda_{12}B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &\quad -\Gamma_{12}P_1A_+^* + \Gamma_{12}(\Lambda_{21}B_+ + \Lambda_0B_0)(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &\quad -Z_+(\Gamma_+P + \Gamma_{12}P_1) \\ &= -(\Gamma_+P + \Gamma_{12}P_1)A_+^* - Z_+(\Gamma_+P + \Gamma_{12}P_1) \\ &\quad +(\Gamma_+\Lambda_+ + \Gamma_{12}\Lambda_{21})B_+(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &\quad +(\Gamma_+\Lambda_{12} + \Gamma_{12}\Lambda_0)B_0(\Lambda_+B_+ + \Lambda_{12}B_0)^* \\ &= -(\Gamma_+P + \Gamma_{12}P_1)A_+^* - Z_+(\Gamma_+P + \Gamma_{12}P_1) \\ &\quad +B_+(\Lambda_+B_+ + \Lambda_{12}B_0)^*. \end{aligned}$$

It is immediate that $\widetilde{A}_{14} = -(\Gamma_+P + \Gamma_{12}P_1)A_+^* - Z_+(\Gamma_+P + \Gamma_{12}P_1)$. Now we apply similarity with

$$V = \begin{pmatrix} I & \Gamma_+P + \Gamma_{12}P_1 \\ 0 & I \end{pmatrix}$$

to the formula for $U(\lambda)$ given in (5.14). First, we observe that

$$V^{-1}\begin{pmatrix} Z_+ & \widetilde{A}_{14} \\ 0 & -A_+^* \end{pmatrix}V = \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{pmatrix}.$$

Also, we have

$$\begin{aligned} &(-(C_+\Lambda_+ + C_0\Lambda_{21}) \\ &\qquad C_+P + C_0P_1 - (\Lambda_+B_+ + \Lambda_{12}B_0)^* - \ (C_+\Lambda_{12} + C_0\Lambda_0)(\Gamma_{21}P + \Gamma_0P_1))V \\ &\quad = \begin{pmatrix} -(C_+\Lambda_+ + C_0\Lambda_{21}) & -(\Lambda_{12}B_0 + \Lambda_+B_+)^* \end{pmatrix}. \end{aligned}$$

Instead of calculating

$$V^{-1}\begin{pmatrix} B_+ - \begin{pmatrix}\Gamma_{12} & 0\end{pmatrix}\Gamma_{W_-}^{-1}\begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^*\end{pmatrix} \\ \begin{pmatrix}0 & I\end{pmatrix}\Gamma_{W_-}^{-1}\begin{pmatrix} B_0 \\ -QB_+ - Q_1^*B_0 + (C_+\Lambda_+ + C_0\Lambda_{21})^*\end{pmatrix}\end{pmatrix}$$

in order to determine a suitable formula for U, we make use of Theorem 2.9 of [AG]. We note that $U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$ has unitary values and that a global pole pair for U is

$$\left(\begin{pmatrix}-(C_+\Lambda_+ + C_0\Lambda_{21}) & -(\Lambda_{12}B_0 + \Lambda_+B_+)^*\end{pmatrix}, \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^*\end{pmatrix}\right).$$

In order to apply [AG] we have to solve the equation

$$\begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^*\end{pmatrix}^*\begin{pmatrix} H_{11} & H_{21}^* \\ H_{21} & H_{22}\end{pmatrix} + \begin{pmatrix} H_{11} & H_{21}^* \\ H_{21} & H_{22}\end{pmatrix}\begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^*\end{pmatrix}$$
$$= \begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})^*(C_+\Lambda_+ + C_0\Lambda_{21}) & (C_+\Lambda_+ + C_0\Lambda_{21})^*(\Lambda_{12}B_0 + \Lambda_+B_+)^* \\ (\Lambda_{12}B_0 + \Lambda_+B_+)(C_+\Lambda_+ + C_0\Lambda_{21}) & (\Lambda_{12}B_0 + \Lambda_+B_+)(\Lambda_{12}B_0 + \Lambda_+B_+)^*\end{pmatrix}.$$

From Lyapunov equations (5.8) and (5.9) it is obvious that $H_{22} = -P$ and $H_{11} = -Q$, respectively. Next, we consider H_{21}. By using (5.3), (5.4), (5.5), and (5.6) we have

$$\begin{aligned} H_{21}Z_+ - A_+H_{21} &= (\Lambda_{12}B_0 + \Lambda_+B_+)(C_+\Lambda_+ + C_0\Lambda_{21}) \\ &= \Lambda_{12}(B_0C_0\Lambda_{21} + B_0C_+\Lambda_+) + \Lambda_+(B_+C_+\Lambda_+ + B_+C_0\Lambda_{21}) \\ &= \Lambda_{12}(\Gamma_0A_0\Lambda_{21} - Z_0\Gamma_0\Lambda_{21} + \Gamma_{21}A_+\Lambda_+ - Z_0\Gamma_{21}\Lambda_+) \\ &\quad +\Lambda_+(\Gamma_+A_+\Lambda_+ - Z_+\Gamma_+\Lambda_+ + \Gamma_{21}A_0\Lambda_{21} - Z_+\Gamma_{12}\Lambda_{21}) \\ &= (\Lambda_{12}\Gamma_0 + \Lambda_+\Gamma_{12})A_0\Lambda_{21} + (\Lambda_{12}\Gamma_{21} + \Lambda_+\Gamma_+)A_+\Lambda_+ - \Lambda_+Z_+ \\ &= A_+\Lambda_+ - \Lambda_+Z_+. \end{aligned}$$

So we can choose $H_{21} = -\Lambda_+$, and therefore

$$H = -\begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P\end{pmatrix}.$$

In this case, we have

$$\begin{aligned} U(\lambda) = I - \begin{pmatrix}C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+B_+)^*\end{pmatrix}\left[\lambda I - \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^*\end{pmatrix}\right]^{-1} \\ \begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P\end{pmatrix}^{-1}\begin{pmatrix}(C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+B_+\end{pmatrix}. \end{aligned} \tag{5.15}$$

Also, we see that (5.15) is a minimal realization of $U(\lambda)$. In what follows, we put

$$S = -\begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P\end{pmatrix} \text{ and } Y = \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^*\end{pmatrix}.$$

Then it follows that

$$SY + Y^*S = \begin{pmatrix}(C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+B_+\end{pmatrix}\begin{pmatrix}C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+B_+)^*\end{pmatrix}. \tag{5.16}$$

Step 2. *Factorizations of U.*

Next, we obtain a minimal factorization of $U(\lambda)$ into two unitary factors as

$$U(\lambda) = U_1(\lambda)U_2(\lambda).$$

The procedure is analogous to that in [R2]. Let S and Y be given as in the above. Any subspace that is invariant under Y is of the form $\mathcal{N}^\times \oplus \mathcal{N}$, where $\mathcal{N}^\times$ is Z_+-invariant and $\mathcal{N}$ is $-A_+^*$-invariant. This is an immediate consequence of the fact that the spectra of Z_+ and $-A_+^*$ are disjoint. Further, we are able to verify that any Y-invariant subspace is S-nondegenerate. In other words, we must show that if $a \in \mathcal{N}^\times \oplus \mathcal{N}$ and $\langle Sa, b\rangle = 0$ for all $b \in \mathcal{N}^\times \oplus \mathcal{N}$, then $a = 0$. Indeed, suppose that $\binom{x}{y} \in \mathcal{N}^\times \oplus \mathcal{N}$ is such that

$$\left\langle S\begin{pmatrix} x \\ y \end{pmatrix}, v \right\rangle = \left\langle \begin{pmatrix} -Qx - \Lambda_+^* y \\ -\Lambda_+ x - Py \end{pmatrix}, v \right\rangle = 0 \quad \text{for all } v \in \mathcal{N}^\times \oplus \mathcal{N}.$$

In particular, this holds for $v = \binom{x}{0}$ and for $v = \binom{0}{y}$. We deduce that

$$0 = \left\langle S\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix} \right\rangle = \langle -Qx, x\rangle - \langle \Lambda_+^* y, x\rangle$$

and

$$0 = \left\langle S\begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix} \right\rangle = \langle -Py, y\rangle - \langle \Lambda_+ x, y\rangle.$$

Consequently, as $Q < 0$ and $P > 0$, unless $x = 0$ we have $-\langle \Lambda_+ x, y\rangle = \langle Qx, x\rangle < 0$. Similarly, unless $y = 0$ we have $-\langle \Lambda_+ x, y\rangle = \langle Py, y\rangle > 0$. Hence it is clear that $x = y = 0$, and as a result $\mathcal{N}^\times \oplus \mathcal{N}$ is S-nondegenerate.

In [AG], it is asserted that there is a one-to-one correspondence between Z-invariant subspaces which are S-nondegenerate and minimal factorizations of U into two unitary factors. Since we have previously shown that any Z-invariant subspace is S-nondegenerate, this is equivalent to a one-to-one correspondence between Z-invariant subspaces and minimal factorizations of U into two unitary factors. Moreover, this one-to-one correspondence may be described as follows. Next, we suppose that $\mathcal{N}^\times \oplus \mathcal{N}$ is a $\left(\begin{smallmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{smallmatrix}\right)$-invariant subspace. Also, let π be the projection onto $\mathcal{N}^\times \oplus \mathcal{N}$ along $[S(\mathcal{N}^\times \oplus \mathcal{N})]^\perp$. For $U(\lambda) = U_1(\lambda)U_2(\lambda)$, we may express U_1 and U_2 and their inverses in terms of π in the following way:

$$\begin{aligned} U_1(\lambda) = I - \begin{pmatrix} C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+ B_+)^* \end{pmatrix} \pi \left[\lambda I - \pi \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{pmatrix} \pi\right]^{-1} \\ \cdot \pi \begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P \end{pmatrix}^{-1} \begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+ B_+ \end{pmatrix} \end{aligned} \tag{5.17}$$

and

$$\begin{aligned} U_1(\lambda)^{-1} = I + \begin{pmatrix} C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+ B_+)^* \end{pmatrix} \begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P \end{pmatrix}^{-1} \pi \\ \cdot \left[\lambda I - \pi \begin{pmatrix} -Z_+^* & 0 \\ 0 & A_+ \end{pmatrix} \pi\right]^{-1} \pi \begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+ B_+ \end{pmatrix}. \end{aligned}$$

Also, we have the minimal right unitary factor and its inverse given by

$$U_2(\lambda) = I - \begin{pmatrix} C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+B_+)^* \end{pmatrix}(I-\pi)$$
$$\cdot \left[\lambda I - (I-\pi)\begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{pmatrix}(I-\pi)\right]^{-1}(I-\pi)\begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P \end{pmatrix}^{-1}\begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+B_+ \end{pmatrix}$$

and

$$U_2(\lambda)^{-1} = I + \begin{pmatrix} C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+B_+)^* \end{pmatrix}\begin{pmatrix} Q & \Lambda_+^* \\ \Lambda_+ & P \end{pmatrix}^{-1}(I-\pi)$$
$$\cdot \left[\lambda I - (I-\pi)\begin{pmatrix} -Z_+^* & 0 \\ 0 & A_+ \end{pmatrix}(I-\pi)\right]^{-1}(I-\pi)\begin{pmatrix} (C_+\Lambda_+ + C_0\Lambda_{21})^* \\ \Lambda_{12}B_0 + \Lambda_+B_+ \end{pmatrix},$$

respectively.

Step 3. *The parametrization of* $W(\lambda)$.

Consider a minimal factorization of U into two unitary factors, $U(\lambda) = U_1(\lambda)U_2(\lambda)$, and put $W(\lambda) = W_+(\lambda)U_1(\lambda)$. Clearly, W is a square spectral factor, i.e., $\Phi(\lambda) = W(\lambda)W(-\overline{\lambda})^*$. It remains to show that this factorization is minimal. Let us suppose that U_1 is obtained as in (5.9). Let A_+ be an $n_+ \times n_+$ matrix. Because of analyticity of W_+ in $\mathbb{C}_-$ we have

$$\#\text{ poles of } W \text{ in } \mathbb{C}_- = \#\text{ poles of } U_1 \text{ in } \mathbb{C}_- = \dim \mathcal{N}.$$

(Of course, as usual, multiplicities are counted in the above number of poles.) As $U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$, we have for W also $W(\lambda) = W_-(\lambda)U_2(\lambda)^{-1}$. Thus, using the fact that W_- is analytic in $\mathbb{C}_+$, we have

$$\begin{aligned} \#\text{ poles of } W \text{ in } \mathbb{C}_+ &= \#\text{ poles of } U_2^{-1} \text{ in } \mathbb{C}_+ \\ &= n_+ - \dim \mathcal{N}, \end{aligned}$$

where we used the formulas above and the minimality of the factorization. Because of Lemma 3.1, # poles of W in $\mathbb{C}_0 = n_0$, where A_0 is an $n_0 \times n_0$ matrix. Furthermore, by (3.7) $\delta(\Phi) = 2n = 2n_+ + 2n_0$, and hence W is a minimal square spectral factor if and only if $\delta(W) = n_+ + n_0$. As the McMillan degree $\delta(W)$ of W is the total number of poles (multiplicities counted) of W, we see that

$$\delta(W) = \dim\mathcal{N} + (n_+ - \dim\mathcal{N}) + n_0 = n_+ + n_0.$$

Thus W is a minimal square spectral factor.

Conversely, let W be a minimal square spectral factor, and put $U_1(\lambda) = W_+(\lambda)^{-1}W(\lambda)$ and $U_2(\lambda) = W(\lambda)^{-1}W_-(\lambda)$. Then $U_1(\lambda)U_2(\lambda) = U(\lambda)$, and U_1 and U_2 are unitary valued for λ on the imaginary axis. Again it remains to show the minimality of this factorization. To see this note that

$$\begin{aligned} &\#\text{ poles of } U_1 \text{ in } \mathbb{C}_- = \#\text{ poles of } W \text{ in } \mathbb{C}_-, \\ &\#\text{ zeros of } U_1 \text{ in } \mathbb{C}_- = \#\text{ zeros of } W \text{ in } \mathbb{C}_-, \end{aligned}$$

and hence by unitarity of U_1, $\delta(U_1) = \#$ poles of W in $\mathbb{C}_- + \#$ poles of U_1 in $\mathbb{C}_+ = \#$ poles of W in $\mathbb{C}_- + \#$ zeros of U_1 in $\mathbb{C}_- = \#$ poles of W in $\mathbb{C}_- + \#$ zeros of W in $\mathbb{C}_-$. Likewise,

$$\begin{aligned} &\#\text{ poles of } U_2 \text{ in } \mathbb{C}_+ = \#\text{ zeros of } W \text{ in } \mathbb{C}_+, \\ &\#\text{ zeros of } U_2 \text{ in } \mathbb{C}_+ = \#\text{ poles of } W \text{ in } \mathbb{C}_+, \end{aligned}$$

and hence $\delta(U_2) = \#$ poles of W in $\mathbb{C}_+ + \#$ zeros of W in $\mathbb{C}_+$. So, we have $\delta(U_1) + \delta(U_2) = \#$ zeros of W not in $i\mathbb{R} + \#$ poles of W not in $i\mathbb{R} = 2\delta(W) - 2n_0 = 2n_+ = \delta(U)$, by (5.15). So, the factorization of U, given above, is minimal. □

6. Parametrization in terms of an algebraic Riccati equation. The connection between stable square spectral factors and the solutions of algebraic Riccati equations goes back to [W]. Also, because of the relationship between invariant Lagrangian subspaces and Hermitian solutions of Riccati equations (see, e.g., [LR], [S1], and [S2]) it is not surprising that the set of all square spectral factors can be parametrized in terms of Riccati equations. In this context we refer also to [FP], [F1], and [LMP]. The result closest in spirit to the one we determine below can be found in [L].

Our analysis will proceed via a procedure analogous to that of [R2], where the minimal square spectral factorization of a positive definite rational matrix function was discussed. In that paper it is shown that for unitary $U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$ there is a connection between the symmetric solution of a certain type of symmetric algebraic Riccati equation and minimal unitary left divisors $U_1(\lambda)$, of $U(\lambda)$ and hence with minimal square spectral factors. In what follows, we will show that an analogous result may be obtained in the positive semidefinite case. With notation as in the previous sections, the result is as follows.

THEOREM 6.1. *Let a minimal realization of the left canonical spectral factor be given by*

$$W_+(\lambda) = I + \begin{pmatrix} C_+ & C_0 \end{pmatrix} \left(\lambda - \begin{pmatrix} A_+ & 0 \\ 0 & A_0 \end{pmatrix} \right)^{-1} \begin{pmatrix} \Gamma_+ & \Gamma_{12} \\ \Gamma_{21} & \Gamma_0 \end{pmatrix}^{-1} \begin{pmatrix} B_+ \\ B_0 \end{pmatrix}$$

and put

$$Y = \begin{pmatrix} Z_+ & 0 \\ 0 & -A_+^* \end{pmatrix}.$$

Furthermore, let $U(\lambda) = W_+(\lambda)^{-1}W_-(\lambda)$, where $W_-(\lambda)$ is the right canonical spectral factor. Consider the algebraic Riccati equation

$$YK + KY^* = KL^*LK, \tag{6.1}$$

where $L = \begin{pmatrix} C_+\Lambda_+ + C_0\Lambda_{21} & (\Lambda_{12}B_0 + \Lambda_+B_+)^* \end{pmatrix}$. If K is a symmetric solution of (6.1), *then*

$$U_1(\lambda) = I - L(\lambda I - Y)^{-1}KL^* \tag{6.2}$$

is a minimal unitary left divisor of $U(\lambda)$, and consequently, $W(\lambda) = W_+(\lambda)U_1(\lambda)$ is a minimal square spectral factor. More precisely, if K solves (6.1), *then* im K *is Y-invariant, and hence it is of the form $\mathcal{N}^\times \oplus \mathcal{N}$ for some Z_+-invariant subspace $\mathcal{N}^\times$ and some $-A_+^*$-invariant subspace $\mathcal{N}$. Then $U_1(\lambda)$ given by* (6.2) *is the same as the unitary minimal left divisor given by* (5.17).

Conversely, any minimal left unitary divisor of $U(\lambda)$ is of the form (6.2) *for some solution K of* (6.1).

Proof. First, we prove the converse. Let $U_1(\lambda)$ be a minimal left unitary divisor of $U(\lambda)$. Then U_1 is given by (5.17) for some Z_+-invariant subspace $\mathcal{N}^\times$ and some $-A_+^*$-invariant subspace $\mathcal{N}$. We remember from the previous section that π projects

onto $\mathcal{N}^{\times} \oplus \mathcal{N}$ along $(S(\mathcal{N}^{\times} \oplus \mathcal{N}))^{\perp}$. We represent $Y : \operatorname{im} \pi \oplus \ker \pi \to \operatorname{im} \pi \oplus \ker \pi$ with respect to the decomposition $\operatorname{im} \pi \oplus \ker \pi$ in the form

$$Y = \begin{pmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{pmatrix}.$$

In addition, we write $L : \operatorname{im} \pi \oplus \ker \pi \to \mathbb{R}^m$ as

$$L = \begin{pmatrix} L_1 & L_2 \end{pmatrix}.$$

Finally, because $\operatorname{im} \pi$ and $\ker \pi$ are S-orthogonal we have $(I - \pi^*)S\pi = 0$. This enables us to present $S : \operatorname{im} \pi \oplus \ker \pi \to \operatorname{im} \pi \oplus \ker \pi$ in the form

$$S = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}.$$

Observe also that we can represent $Y^* : \operatorname{im} \pi^* \oplus \ker \pi^* \to \operatorname{im} \pi^* \oplus \ker \pi^*$ with respect to the decomposition $\operatorname{im} \pi^* \oplus \ker \pi^*$ as

$$Y^* = \begin{pmatrix} Y_{11}^* & 0 \\ Y_{12}^* & Y_{22}^* \end{pmatrix}.$$

We are able to rewrite (5.16) in the form

$$YS^{-1} + S^{-1}Y^* = S^{-1}L^*LS^{-1},$$

where, in particular, the (1,1)-entry is given by

$$Y_{11}S_1^{-1} + S_1^{-1}Y_{11}^* = S_1^{-1}L_1^*L_1S_1^{-1}.$$

If we put $K = \left(\begin{smallmatrix} S_1^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right)$, then K solves (6.1). In addition, (5.17) may be expressed in different ways as

$$U_1(\lambda) = I - L\pi(\lambda I - \pi Y \pi)^{-1}\pi S^{-1}L^*$$

and

$$U_1(\lambda) = I - L_1(\lambda I - Y_{11})^{-1}S_1^{-1}L_1^*,$$

where $\pi S^{-1} = \pi S^{-1}\pi^*$. Moreover, the latter expression may be written in an alternative form as

$$U_1(\lambda) = I - L(\lambda I - Y)^{-1}S^{-1}L^*.$$

The direct statement is obtained easily from the observation that if K solves (6.1), then $\operatorname{im} K$ is Y-invariant. ∎

REFERENCES

[AG] D. Alpay and I. Gohberg, *Unitary rational matrix functions*, in Topics in Interpolation Theory of Rational Matrix-Valued Functions, Oper. Theory Adv. Appl. 33, Birkhäuser-Verlag, Basel, 1988, pp. 175–222.

[BGR] J.A. Ball, I. Gohberg, and L. Rodman, *Interpolation of Rational Matrix Functions*, Oper. Theory Adv. Appl. 45, Birkhäuser-Verlag, Basel, 1990.

[BR1] J.A. Ball and A.C.M. Ran, *Global inverse spectral problems for rational matrix functions*, Linear Algebra Appl., 86 (1987), pp. 237–382.

[BR2] J.A. Ball and A.C.M. Ran, *Left versus right canonical Wiener–Hopf factorization*, in Constructive Methods of Wiener–Hopf Factorization, Oper. Theory Adv. Appl. 21, Birkhäuser-Verlag, Basel, 1986, pp. 9–38.

[BGK] H. Bart, I. Gohberg, and M.A. Kaashoek, *Minimal Factorization of Matrix and Operator Functions*, Oper. Theory Adv. Appl. 1, Birkhäuser-Verlag, Basel, 1979.

[BGKvD] H. Bart, I. Gohberg, M.A. Kaashoek, and P. van Dooren, *Factorization of transfer functions*, SIAM J. Control Optim., 18 (1980), pp. 675–696.

[C] D.J. Clements, *Rational spectral factorization using state-space methods*, Systems Control Lett., 20 (1993), pp. 335–343.

[CG] D.J. Clements and K. Glover, *Spectral factorization by Hermitian pencils*, Linear Algebra Appl., 122-124 (1989), pp. 797–846.

[FMP] A. Ferrante, G. Michaletzky, and M. Pavon, *Parametrization of all minimal square spectral factors*, Systems Control Lett., 21 (1993), pp. 249–254.

[FP] L. Finesso and G. Picci, *A characterization of minimal spectral factors*, IEEE Trans. Automat. Control, AC27 (1982), pp. 122–127.

[F1] P.A. Fuhrmann, *The algebraic Riccati equation–A polynomial approach*, Systems Control Lett., 5 (1985), pp. 369–376.

[F2] P.A. Fuhrmann, *On the characterization and parametrization of minimal spectral factors*, J. Math. Systems Estim. Control, 5 (1995), pp. 383–444.

[FG] P.A. Fuhrmann and A. Gombani, *On a Hardy space approach to the analysis of spectral factors*, Internat. J. Control, 71 (1998), pp. 277–357.

[GK] I. Gohberg and M.A. Kaashoek, *An inverse problem for rational matrix functions and minimal divisibility*, Integral Equations Operator Theory, 10 (1987), pp. 437–465.

[GLR] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Oper. Theory Adv. Appl. 8, Birkhäuser-Verlag, Basel, 1983.

[L] L. Lerer, *The matrix quadratic equation and factorization of matrix polynomials*, in The Gohberg Anniversary Collection, Vol. I, Oper. Theory Adv. Appl. 40, Birkhäuser-Verlag, Basel, 1989, pp. 279–324.

[LR] P. Lancaster and L. Rodman, *Existence and uniqueness theorems for algebraic Riccati equations*, Internat. J. Control, 32 (1980), pp. 285–309.

[LMP] A. Lindquist, G. Michaletzky, and G. Picci, *Zeros of spectral factors, the geometry of splitting subspaces, and the algebraic Riccati inequality*, SIAM J. Control Optim., 33 (1995), pp. 365–401.

[LP1] A. Lindquist and G. Picci, *A geometric approach to modelling and estimation of linear stochastic systems*, J. Math. Systems Estim. Control, 1 (1991), pp. 241–333.

[LP2] A. Lindquist and G. Picci, *Forward and backward semimartingale representations for stationary increment processes*, Stochastics, 15 (1985), pp. 1–50.

[R1] A.C.M. Ran, *Minimal factorizations of self-adjoint rational matrix functions*, Integral Equations Operator Theory, 5 (1982), pp. 850–869.

[R2] A.C.M. Ran, *Minimal square spectral factors*, Systems Control Lett., 26 (1994), pp. 621–634.

[R3] A.C.M. Ran, *Semidefinite Invariant Subspaces, Stability and Applications*, Brügemann, Den Burg-Texel, 1984.

[R4] A.C.M. Ran, *Unitary solutions of a class of algebraic Riccati equations and factorization*, Linear Algebra Appl., 162-164 (1992), pp. 521–540.

[RR1] A.C.M. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces* I, in Topics in Operator Theory, Oper. Theory Adv. Appl. 32, Birkhäuser-Verlag, Basel, 1988, pp. 181–218.

[RR2] A.C.M. Ran and L. Rodman, *Stability of invariant maximal semidefinite subspaces* I, Linear Algebra Appl., 62 (1984), pp. 51–86.

[RR3] A.C.M. Ran and L. Rodman, *Stable invariant Lagrangian subspaces: Factorization of symmetric rational matrix functions and applications*, Linear Algebra Appl., 137/138 (1990), pp. 576–620.

[Sa] L.A. Sahnovic, *On the factorization of an operator valued transfer function*, Soviet Math. Dokl., 17 (1976), pp. 203–207.

[S1] M.A. Shayman, *Geometry of the algebraic Riccati equation* I, SIAM J. Control Optim., 21 (1983), pp. 375–394.

[S2] M.A. Shayman, *Geometry of the algebraic Riccati equation* II, SIAM J. Control Optim., 21 (1983), pp. 395–409.

[W] J.C. Willems, *Least squares stationary optimal control and the algebraic Riccati equation*, IEEE Trans. Automat. Control, AC16 (1971), pp. 621–634.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1245–1262

STRUCTURE AND PERTURBATION ANALYSIS OF TRUNCATED SVDs FOR COLUMN-PARTITIONED MATRICES*

ZHENYUE ZHANG† AND HONGYUAN ZHA‡

Abstract. In this paper we study truncated SVDs for column-partitioned matrices. In particular, we analyze the relation between the truncated SVDs of a matrix and the truncated SVDs of its submatrices. We give necessary and sufficient conditions under which a truncated SVD of a matrix can be constructed from those of its submatrices. We then present perturbation analysis to show that an *approximate* truncated SVD can still be computed even if the given necessary and sufficient conditions are only approximately satisfied. We also apply our general results to a class of matrices with the so-called low-rank-plus-shift structure.

Key words. singular value and singular vector, singular value decomposition, perturbation analysis, block matrix

AMS subject classifications. 15A18, 65F15

PII. S0895479899357875

1. Introduction.

In many applications it is desirable to compute a low-rank approximation of a given matrix $A \in \mathcal{R}^{m\times n}$; see [6], for example, for a list of application areas. In many cases the matrix A is rather large and/or sparse, and therefore computational efficiency is of paramount importance. The theory of SVD provides the following characterization of the best low-rank approximation of A in terms of Frobenius norm $\|\cdot\|_F$ [4, Theorem 2.5.3]. (Similar results hold for general unitarily invariant norms such as the spectral norm as well.)

THEOREM 1.1. *Let the SVD of $A \in \mathcal{R}^{m\times n}$ be $A = U\Sigma V^T$ with*

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min\{m,n\}}), \quad \sigma_1 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0,$$

and U and V orthogonal. Then, for $1 \leq k \leq \min\{m,n\}$,

$$\sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2 = \min\{\ \|A-B\|_F^2 \mid \operatorname{rank}(B) \leq k\}.$$

And the minimum is achieved with $\operatorname{best}_k(A) \equiv U_k \operatorname{diag}(\sigma_1, \ldots, \sigma_k) V_k^T$, where U_k and V_k are the matrices formed by the first k columns of U and V, respectively. Furthermore, $\operatorname{best}_k(A)$ is unique if and only if $\sigma_k > \sigma_{k+1}$.

*Received by the editors June 23, 1999; accepted for publication (in revised form) by N. Higham December 8, 2000; published electronically April 6, 2001.

http://www.siam.org/journals/simax/22-4/35787.html

†Department of Mathematics, Zhejiang University, Hangzhou, 310027, People's Republic of China (zyzhang@math.zju.edu.cn). The work of this author was supported in part by NSFC (project 19771073), Foundation for University Key Teacher of the Ministry of Education of China, the Special Funds for Major State Basic Research Projects (G19990328), NSF grant CCR-9619452, the Director, Office of Science, Office of Laboratory Policy and Infrastructure Management of the U.S. Department of Energy under contract DE-AC-76SF00098. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.

‡Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 (zha@cse.psu.edu). The work of this author was supported in part by NSF grant CCR-9619452 and CCR-9901986.

In this paper, we call $\text{best}_k(A)$ a *truncated SVD* of A, which is obtained by truncating the finite sum expansion

$$A = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$

up to and including the kth term, where we have written $U = [u_1, \ldots, u_m]$ and $V = [v_1, \ldots, v_n]$. Algorithms for computing a (truncated) SVD, even in the case when A is large and/or sparse are well established [1, 3, 4]. In this paper we are concerned with an interesting issue which is motivated by some of the results developed in [13] where we dealt with the relation between truncated SVDs of the so-called term-document matrices and a special indexing method called *latent semantic indexing* (LSI) in information retrieval.[1] In this paper we will build on the results obtained in [13] and study truncated SVDs of column-partitioned matrices in greater generality.

We observed that in some applications, the matrix A is naturally partitioned into several block columns, i.e., A can be written as $A = [A_1, \ldots, A_s]$, where $A_i, i = 1, \ldots, s$, are block columns of A. In text categorization applications, for example, each column of A represents a document in a given text corpus, and A_i consists of all the documents in the text corpus that are about a particular topic i. For example, if we have three categories: science, entertainment and sports, then $A = [A_1, A_2, A_3]$, where A_1 contains all the documents about science, A_2 entertainment, and A_3 sports. In a similar situation when we consider dynamic information retrieval, A_1 will represent the documents from an old text corpus, and $A_2, \ldots, A_s$ are document collections added dynamically as new documents become available [11]. An important problem from those applications is the following: we have computed a truncated SVD of the first few of the A_i's, say, $\text{best}_k([A_1, \ldots, A_t])$ for $[A_1, \ldots, A_t]$ with some $t < s$, and the matrix $[A_1, \ldots, A_t]$ has been discarded, to save storage, for example, and is therefore no longer available. How can we construct a truncated SVD of A from $\text{best}_k([A_1, \ldots, A_t])$ and the remaining $[A_{t+1}, \ldots, A_s]$? To answer this question we need to study the relation between truncated SVDs of a matrix and those of its submatrices. It turns out that a general theory can be developed and the question we are interested in can be answered by certain special cases of the general theory. We should also mention that as far as the truncated SVD of a matrix A is concerned one may be interested in either $\text{best}_k(A)$ or the range space of $\text{best}_k(A)$, i.e., the subspace spanned by the first k left singular vectors of A. The latter case happens, for example, in signal processing, when one is interested in computing the signal subspace which is represented by the range space of $\text{best}_k(A)$ [8, 9]. In information retrieval applications one is interested in $\text{best}_k(A)$ itself in its factorized form $\text{best}_k(A) = U_k \Sigma_k V_k^T$: for a given query vector q, $q^T \text{best}_k(A)$ is used for ranking all the documents represented by the columns of A; see [2] for more details. As we will show later in section 3, perturbation results for the range space and row space of $\text{best}_k(A)$ can be easily obtained from those for $\text{best}_k(A)$, and therefore throughout the rest of the paper we will concentrate on $\text{best}_k(A)$ itself instead of the range space and row space of $\text{best}_k(A)$.

The rest of the paper is organized as follows. In section 2, we give necessary and sufficient conditions that guarantee a truncated SVD of a column-partitioned matrix A can be perfectly constructed from truncated SVDs of its submatrices. The orthogonality of certain submatrices of A plays an important role in specifying those conditions. We also relate the sufficient conditions to a class of matrices with the

[1] For a detailed discussion of LSI from the linear algebra point of view, see [2].

so-called low-rank-plus-shift structure [8, 9, 12]. In section 3, we expand the results in section 2 to the case where the necessary and sufficient conditions are only approximately satisfied by the given matrix A. We show that a truncated SVD of A can be approximately constructed from truncated SVDs of its submatrices. Along the way, we prove some novel perturbation bounds for the truncated SVD of a matrix that are of their own interest. The perturbation analysis for matrices with low-rank-plus-shift structure is carried out in some detail, and an improved perturbation bound is also derived.

2. Necessary and sufficient conditions. As mentioned in section 1, we are interested in finding conditions on a column-partitioned matrix $A = [A_1, \ldots, A_s]$ such that a truncated SVD of A can be constructed from those of the A_i's. It is not difficult to see that certain information about the original matrix A will be lost if we rely solely on using the truncated SVDs of the A_i's. Therefore, in general, we cannot expect to reconstruct a truncated SVD of A *perfectly* from those of the A_i's. The goal of this section is to find conditions under which this can be done. We first present a general result which gives the necessary and sufficient condition for a matrix and its perturbation to have the same truncated SVDs.

Note. Throughout the rest of the paper, we will use the following conventions: the singular values of a matrix are indexed in ascending order, i.e., for a matrix $B \in \mathcal{R}^{m\times n}$,

$$\sigma_1(B) \geq \sigma_2(B) \geq \cdots \geq \sigma_{\min\{m,n\}}(B)$$

whenever $\text{best}_k(B)$ is mentioned for a matrix B, it is implicitly assumed that $\sigma_k(B) > \sigma_{k+1}(B)$ so that $\text{best}_k(B)$ is uniquely defined. We consider only the spectral norm and write $\|\cdot\|$ for $\|\cdot\|_2$. We also use $\text{span}(A)$ to denote the linear subspace spanned by the columns of A.

THEOREM 2.1. *Let $A = B + C$. Then $\text{best}_k(A) = \text{best}_k(B)$ if and only if the following three conditions are satisfied:*

$$C^T\text{best}_k(B) = 0, \quad \text{best}_k(B)C^T = 0, \quad \sigma_k(B) > \sigma_{k+1}(A).$$

Proof. We first deal with the *only if* part of the proof which is rather straightforward. Since

$$(A - \text{best}_k(A))^T\text{best}_k(A) = 0,$$

it follows from $\text{best}_k(A) = \text{best}_k(B)$ that

$$(A - \text{best}_k(B))^T\text{best}_k(B) = 0.$$

Substituting A with $B + C$ and using the equality $(B - \text{best}_k(B))^T\text{best}_k(B) = 0$, we obtain

$$C^T\text{best}_k(B) = 0.$$

We can similarly show that $\text{best}_k(B)C^T = 0$. Furthermore, the inequality $\sigma_k(B) > \sigma_{k+1}(A)$ follows from $\sigma_k(A) = \sigma_k(B)$ and $\sigma_k(A) > \sigma_{k+1}(A)$.

Now we prove the *if* part. Let the SVD of B and C be

$$B = [U_1, U_2]\begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}[V_1, V_2]^T, \quad C = QDG^T,$$

respectively, where $\Sigma_1 \in \mathcal{R}^{k\times k}$ and the matrices are partitioned conformally. Then $\text{best}_k(B) = U_1\Sigma_1 V_1$. Now the two conditions $C^T\text{best}_k(B) = 0$ and $\text{best}_k(B)C^T = 0$ imply that

$$U_1^T C = 0, \quad CV_1 = 0.$$

Let the SVD of $U_2\Sigma_2 V_2^T + C$ be

$$U_2\Sigma_2 V_2^T + C = \tilde{U}_2\tilde{\Sigma}_2\tilde{V}_2^T.$$

It is readily verified that $\tilde{U}_2^T U_1 = 0$ and $\tilde{V}_2^T V_1 = 0$. Therefore,

$$A = B + C = [U_1, \tilde{U}_2]\begin{bmatrix} \Sigma_1 & \\ & \tilde{\Sigma}_2 \end{bmatrix}[V_1, \tilde{V}_2]^T$$

gives the SVD of A. Since $\sigma_{\min}(\Sigma_1) = \sigma_k(B) > \sigma_{k+1}(A)$, it follows that $\sigma_{\min}(\Sigma_1) > \sigma_{\max}(\tilde{\Sigma}_2)$, and therefore

$$\text{best}_k(A) = U_1\Sigma_1 V_1 = \text{best}_k(B),$$

completing the proof. ∎

Remark. We notice that the inequality $\sigma_k(B) > \sigma_{k+1}(A)$ together with the two orthogonality conditions $C^T\text{best}_k(B) = 0$ and $\text{best}_k(B)C^T = 0$ implies that $\sigma_{\min}(B) > \sigma_{\max}(C)$. Therefore, it is not necessary that C be zero in order for a perfect reconstruction of a truncated SVD of A to occur. To say it in a more interesting way (as will be demonstrated later in section 3), in order to have a good approximate reconstruction of a truncated SVD of A it is not necessary that C be *small* in norm.

Now we are ready to consider the case where A is partitioned in various block-column forms. We partition A as $A = [A_1, A_2]$, where $A_i \in \mathcal{R}^{m\times n_i}, i = 1, 2$. First, we look at the two truncated SVDs $\text{best}_k(A)$ and $\text{best}_k([A_1, 0])$.

COROLLARY 2.2. *Let* $A = [A_1, A_2]$. *Then*

$$\text{best}_k(A) = \text{best}_k([A_1, 0])$$

if and only if the following two conditions are satisfied:

$$A_2^T\text{best}_k(A_1) = 0, \quad \sigma_k(A_1) > \sigma_k(A).$$

Proof. Write $A = [A_1, 0] + [0, A_2]$. It is easy to see that $\text{best}_k([A_1, 0]) = [\text{best}_k(A_1), 0]$, and therefore, $\text{best}_k([A_1, 0])[0, A_2]^T = 0$. The result now follows from Theorem 2.1. ∎

Now we look at the case where we have computed $\text{best}_{k_1}(A_1)$, and A_1 has already been discarded. We then add A_2, and we want to reconstruct $\text{best}_k(A)$ based on $\text{best}_{k_1}(A_1)$ and A_2.

COROLLARY 2.3. *Let* $A = [A_1, A_2]$ *and* $k_1 \le n_1$. *Then*

$$\text{best}_k(A) = \text{best}_k([\text{best}_{k_1}(A_1), A_2])$$

if and only if the following two conditions are satisfied:

$$(A_1 - \text{best}_{k_1}(A_1))^T\text{best}_k([\text{best}_{k_1}(A_1), A_2]) = 0, \quad \sigma_k([\text{best}_{k_1}(A_1), A_2]) > \sigma_{k+1}(A).$$

Proof. We write

$$A = [A_1, A_2] = [\text{best}_{k_1}(A_1), A_2] + [A_1 - \text{best}_{k_1}(A_1), 0].$$

It is easy to see that

$$[\text{best}_{k_1}(A_1), A_2][A_1 - \text{best}_{k_1}(A_1), 0]^T = 0,$$

and therefore the second condition of Theorem 2.1 is automatically satisfied. The result then follows directly from Theorem 2.1 □

We finally consider the case that we have computed $\text{best}_{k_1}(A_1)$ and $\text{best}_{k_2}(A_2)$, and both A_1 and A_2 were discarded. We want to compute $\text{best}_k(A)$ using what we have, i.e., $\text{best}_{k_1}(A_1)$ and $\text{best}_{k_2}(A_2)$.

COROLLARY 2.4. *Let $A = [A_1, A_2]$, $k_1 \le n_1$, and $k_2 \le n_2$. Then*

$$\text{best}_k(A) = \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])$$

if and only if the following two conditions are satisfied:

$$[A_1 - \text{best}_{k_1}(A_1), A_2 - \text{best}_{k_2}(A_2)]^T \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) = 0,$$

$$\sigma_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) > \sigma_{k+1}(A).$$

Proof. The proof is similar to that of Corollary 2.3 and therefore is omitted. □

Remark. The conditions listed in both Corollary 2.3 and Corollary 2.4 seem to be rather complicated. In some situations, however, we may be able to verify some stronger but simpler conditions. For example,

$$(A_1 - \text{best}_{k_1}(A_1))^T A_2 = 0 \tag{2.1}$$

implies the condition

$$(A_1 - \text{best}_{k_1}(A_1))^T \text{best}_k([\text{best}_{k_1}(A_1), A_2]) = 0,$$

and the two equalities

$$(A_1 - \text{best}_{k_1}(A_1))^T A_2 = 0, \quad (A_2 - \text{best}_{k_2}(A_2))^T A_1 = 0 \tag{2.2}$$

imply the condition

$$[A_1 - \text{best}_{k_1}(A_1), A_2 - \text{best}_{k_1}(A_2)]^T \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) = 0.$$

A class of matrices which satisfies (2.2) will be given in Theorem 2.7. Roughly speaking, we can interpret the condition in (2.1) as follows: for a perfect reconstruction to occur, what is added, i.e., A_2, should be orthogonal to what is discarded, i.e., $A_1 - \text{best}_{k_1}(A_1)$. The conditions in (2.2) have a similar interpretation.

Now we show an interesting application of Corollary 2.4.

COROLLARY 2.5. *The equality* $\text{best}_k(A) = \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])$ *holds if and only if for any* $t_i \ge k_i, i = 1, 2,$

$$\text{best}_k(A) = \text{best}_k([\text{best}_{t_1}(A_1), \text{best}_{t_2}(A_2)]).$$

Proof. We just need to prove the *only if* part. Let $\tilde{A}_i = \text{best}_{t_i}(A_i), i = 1, 2$. It is easy to verify that $\text{best}_{k_i}(\tilde{A}_i) = \text{best}_{k_i}(A_i)$ since $t_i \geq k_i, i = 1, 2$. Now we need to show that

$$\text{best}_k([\tilde{A}_1, \tilde{A}_2]) = \text{best}_k([\text{best}_{k_1}(\tilde{A}_1), \text{best}_{k_2}(\tilde{A}_2)]).$$

Using Corollary 2.4, we need to first verify that

$$[\tilde{A}_1 - \text{best}_{k_1}(\tilde{A}_1), \tilde{A}_2 - \text{best}_{k_1}(\tilde{A}_2)]^T \text{best}_k([\text{best}_{k_1}(\tilde{A}_1), \text{best}_{k_2}(\tilde{A}_2)]) = 0.$$

Since $\text{span}\{\tilde{A}_i - \text{best}_{k_i}(\tilde{A}_i)\} \subset \text{span}\{A_i - \text{best}_{k_i}(A_i)\}$, the above equality follows from the given condition. Next the inequality

$$\sigma_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) \leq \sigma_k([\text{best}_{t_1}(A_1), \text{best}_{t_2}(A_2)])$$

follows from a general inequality about the monotonicity of singular values established in [10]. ∎

The results in Corollaries 2.4 and 2.5 can be generalized to the cases where $A = [A_1, \ldots, A_s]$. We just state the case for Corollary 2.4.

COROLLARY 2.6. *Let* $A = [A_1, \ldots, A_s]$ *with* $A_i \in \mathcal{R}^{m \times n_i}$, *and* $k_i \leq n_i, i = 1, \ldots, s$. *Then*

$$\text{best}_k(A) = \text{best}_k([\text{best}_{k_1}(A_1), \ldots, \text{best}_{k_s}(A_s)])$$

if and only if, for $i = 1, \ldots, s$, *we have*

$$(A_i - \text{best}_{k_i}(A_i))^T \text{best}_k([\text{best}_{k_1}(A_1), \ldots, \text{best}_{k_s}(A_s)]) = 0,$$

and

$$\sigma_k([\text{best}_{k_1}(A_1), \ldots, \text{best}_{k_s}(A_s)]) > \sigma_{k+1}(A).$$

As an application of the results established in the above corollaries, we consider a special class of matrices that possess the so-called *low-rank-plus-shift* structure. This class of matrices arises naturally in applications such as array signal processing and LSI in information retrieval [8, 9, 12]. Specifically, a matrix A has the low-rank-plus-shift structure if $A^T A$ is a low-rank perturbation of a positive multiple of the identity matrix (cf. (2.3)). We now show that matrices with low-rank-plus-shift structure satisfy the sufficient conditions of Corollary 2.4.[2]

THEOREM 2.7. *Let* $A = [A_1, A_2] \in \mathcal{R}^{m \times n}$ *with* $A_1 \in \mathcal{R}^{m \times n_1}$ *and* $A_2 \in \mathcal{R}^{m \times n_2}$. *Assume that*

$$A^T A = X + \sigma^2 I, \tag{2.3}$$

where X *is positive semidefinite with* $\text{rank}(X) = k$. *Partition* X *as* $X = (X_{ij})_{i,j=1}^2$ *with* $X_{ii} \in \mathcal{R}^{n_i \times n_i}$ *and let* $\text{rank}(X_{ii}) = k_i$, $i = 1, 2$. *Then*

$$(A_1 - \text{best}_{k_1}(A_1))^T A_2 = 0, \quad (A_2 - \text{best}_{k_2}(A_2))^T A_1 = 0. \tag{2.4}$$

Furthermore,

$$\text{best}_k(A) = \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]).$$

[2] A similar result was also proved in [13].

Proof. For $i = 1, 2$, we have $A_i^T A_i = X_{ii} + \sigma^2 I$ with X_{ii} positive semidefinite and $\operatorname{rank}(X_{ii}) = k_i$. In addition, we can write the SVD of A_i in the following form:

$$A_i = U_i \operatorname{diag}(\Sigma_i, \sigma I) V_i^T = [U_{i1}, U_{i2}] \operatorname{diag}(\Sigma_i, \sigma I)[V_{i1}, V_{i2}]^T,$$

where V_i is orthogonal, and

$$\Sigma_i = (D_i + \sigma^2 I)^{1/2}, \quad D_i = \operatorname{diag}(\mu_1^{(i)}, \ldots, \mu_{k_1}^{(i)})$$

with $\mu_1^{(i)} \geq \cdots \geq \mu_{k_1}^{(i)} > 0$. Hence

$$\operatorname{best}_{k_i}(A_i) = U_{i1} \Sigma_i V_{i1}^T, \quad A_i - \operatorname{best}_{k_i}(A_i) = \sigma U_{i2} V_{i2}^T,$$

and we now need only to show $U_{12}^T A_2 = 0$ and $U_{22}^T A_1 = 0$. To this end, consider the symmetric positive semidefinite matrix

$$\begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}^T (A^T A - \sigma^2 I) \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix} = \begin{bmatrix} D_1 & 0 & \Sigma_1 U_{11}^T U_{21} \Sigma_2 & \Sigma_1 U_{11}^T U_{22} \Sigma_2 \\ & 0 & \Sigma_1 U_{12}^T U_{21} \Sigma_2 & \Sigma_1 U_{12}^T U_{22} \Sigma_2 \\ & & D_2 & 0 \\ & & & 0 \end{bmatrix},$$

where for the last matrix in the above equation, *blank* denotes block matrix elements by symmetry. Since a principal submatrix of a positive semidefinite matrix is still positive semidefinite, we obtain

$$U_{12}^T U_{21} = 0, \quad U_{11}^T U_{22} = 0, \quad U_{12}^T U_{22} = 0,$$

and the rank of the matrix

$$\tilde{A} = \begin{bmatrix} D_1 & \Sigma_1 U_{11}^T U_{21} \Sigma_2 \\ (\Sigma_1 U_{11}^T U_{21} \Sigma_2)^T & D_2 \end{bmatrix}$$

equals k. Hence it follows from

$$B^T B = \operatorname{diag}(V_{11}, V_{21})(\tilde{A} + \sigma^2 I) \operatorname{diag}(V_{11}^T, V_{21}^T),$$

where $B = [\operatorname{best}_{k_1}(A_1), \operatorname{best}_{k_2}(A_2)]$, that

$$\sigma_k([\operatorname{best}_{k_1}(A_1), \operatorname{best}_{k_2}(A_2)]) > \sigma = \sigma_{k+1}(A).$$

The result of the theorem now follows from Corollary 2.4. ∎

Remark. By definition, the ranks $k_1 = \operatorname{rank}(X_{11})$ and $k_2 = \operatorname{rank}(X_{22})$ must satisfy $k_1 \leq k$, $k_2 \leq k$, and $k \leq k_1 + k_2 \leq 2k$. It is also easy to find examples for which $k_1 + k_2 = k$ or $k_1 + k_2 = 2k$. In some cases, it is possible to find a permutation P such that $AP \equiv [A_1, A_2]$ will have A_i with k_i, $i = 1, 2$, that are smaller than those of A. For example, let

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

It is easy to verify that $k = 2$. If we take the first two columns as A_1 and the last two columns as A_2, we have $k_1 = k_2 = 2$. However, if we take the middle two columns as A_1 and the first and last columns as A_2, we have $k_1 = k_2 = 1$. This example motivates the following question: Is it always possible to find a permutation P such that a partition of $AP = [A_1, A_2]$ with A_1 and A_2 having about the same column dimensions will give $k_1 + k_2 < 2k$? The answer turns out to be no. In the following we show that we can find a class of matrices A satisfying

$$A^T A - \sigma^2 I = X$$

with X positive semidefinite such that for any permutation $AP = [A_1, A_2]$ we will have $k_1 = k_2 = k$, provided the column dimensions of A_1 and A_2 are not smaller than k. To see this, let $Y \in \mathcal{R}^{k\times n}$ be a matrix any k columns of which are linearly independent. Let

$$C^T C = Y^T Y + \sigma^2 I$$

be the Cholesky decomposition of $Y^T Y + \sigma^2 I$. Set $A = QC$, where Q is an arbitrary orthogonal matrix. Then it is easy to see that for any permutation P, a partition of $AP = [A_1, A_2]$ with column dimensions of A_1 and A_2 at least k will have $k_1 = k_2 = k$.

3. Perturbation analysis. In the previous section we give necessary and sufficient conditions for perfectly reconstructing a truncated SVD of a matrix from those of its submatrices. In this section, we consider the case when these conditions are no longer satisfied and the reconstruction will not be perfect. We will perform perturbation analysis to bound the difference between a truncated SVD of a matrix, say, $\text{best}_k(A)$, and the reconstruction obtained from the truncated SVDs of its submatrices, say, $\text{best}_k(B)$. It was mentioned in section 1 that in some applications one may also be interested in comparing the range spaces of $\text{best}_k(A)$ and $\text{best}_k(B)$ instead of $\|\text{best}_k(A) - \text{best}_k(B)\|$. The following proposition shows that a bound on the angle between the two range spaces can be readily obtained once a bound on $\|\text{best}_k(A) - \text{best}_k(B)\|$ is available.

PROPOSITION 3.1. *Let* Θ *be the angle between the range spaces of* $\text{best}_k(A)$ *and* $\text{best}_k(B)$. *Then*

$$\|\sin\Theta\| \le \|\text{best}_k(A) - \text{best}_k(B)\|/\sigma_k(A).$$

Proof. Let $\text{best}_k(A) = U_1\Sigma_A V_1^T$, where $\Sigma_A = \text{diag}(\sigma_1(A), \ldots, \sigma_k(A))$, and let $\text{best}_k(B) = Q_1\Sigma_B G_1^T$. Writing $\text{best}_k(A) = \text{best}_k(B) + E$ gives

$$\begin{aligned} \|\sin\Theta\| &= \|(I - Q_1Q_1^T)U_1\| \\ &= \|(I - Q_1Q_1^T)\text{best}_k(A)V_1\Sigma_A^{-1}\| \\ &= \|(I - Q_1Q_1^T)EV_1\Sigma_A^{-1}\| \\ &\le \|E\|/\sigma_k(A). \end{aligned}$$

We can similarly give an upper bound on the sine of the angle between V_1 and G_1. ∎

In view of the result in Proposition 3.1, we will concentrate on deriving perturbation bounds for $\|\text{best}_k(A) - \text{best}_k(B)\|$ with the understanding that a bound on the range spaces can be obtained via Proposition 3.1. We first give a general result concerning perturbation bounds of truncated SVDs. The perturbation bound is so

derived that we get back the result of Theorem 2.1 when the necessary and sufficient conditions of Theorem 2.1 are satisfied.

THEOREM 3.2. *Let $A = B + C \in \mathcal{R}^{m\times n}$, and for some $k < \min\{m,n\}$ we have $\sigma_{k+1}(A) < \sigma_k(B)$. Then*

$$\|\mathrm{best}_k(A) - \mathrm{best}_k(B)\| \le \frac{\|A\|(\|\mathrm{best}_k(B)C^T\| + \|C^T\mathrm{best}_k(B)\|)}{\sigma_k^2(B) - \sigma_{k+1}^2(A)} + \|P_{\mathrm{best}_k(B^T)}C^T\|,$$

where $P_{\mathrm{best}_k(B^T)}$ is the orthogonal projector onto the subspace $\mathrm{span}\{\mathrm{best}_k(B^T)\}$.

Proof. Let the SVD of B be

$$B = U\Sigma V^T = [U_1, U_2]\,\mathrm{diag}(\Sigma_1, \Sigma_2)[V_1, V_2]^T$$

with $\Sigma_1 \in \mathcal{R}^{k\times k}$ and $\mathrm{best}_k(B) = U_1\Sigma_1 V_1^T$. Write

$$\tilde{C} \equiv U^T C V = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{3.1}$$

with $C_{11} \in \mathcal{R}^{k\times k}$, and let

$$U^T A V = \Sigma + \tilde{C} = QDG^T \equiv \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}^T \tag{3.2}$$

be the SVD of U^TAV with Q_{11}, D_1, and G_{11} all k-by-k matrices. Then we have

$$\begin{aligned} \|\Delta\| &= \|\mathrm{best}_k(A) - \mathrm{best}_k(B)\| \\ &= \left\| \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} D_1 \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix}^T - \begin{bmatrix} \Sigma_1 & \\ & 0 \end{bmatrix} \right\| \\ &= \left\| \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} [D_1, 0] - \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} [G_{11}, G_{12}] \right\| \\ &= \left\| \begin{bmatrix} Q_{11}D_1 - \Sigma_1 G_{11} & -\Sigma_1 G_{12} \\ Q_{21}D_1 & 0 \end{bmatrix} \right\|. \end{aligned}$$

From (3.2) we have

$$\Sigma_1 G_{11} + [C_{11}, C_{12}] \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} = Q_{11}D_1,$$

$$\Sigma_2 G_{21} + [C_{21}, C_{22}] \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} = Q_{21}D_1.$$

It follows that

$$\begin{aligned} \|\Delta\| &= \left\| \begin{bmatrix} C_{11}G_{11} + C_{12}G_{21} & -\Sigma_1 G_{12} \\ C_{21}G_{11} + (\Sigma_2 + C_{22})G_{21} & 0 \end{bmatrix} \right\| \\ &= \left\| UAV^T \begin{bmatrix} 0 & -G_{12} \\ G_{21} & 0 \end{bmatrix} + \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} [G_{11}, G_{12}] \right\| \qquad (3.3) \\ &\le \|A\|\|G_{12}\| + \left\| \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} \right\|, \end{aligned}$$

where we have used $\|G_{12}\| = \|G_{21}\|$. On the other hand from the equalities

$$(\Sigma + \tilde{C})G = QD, \quad (\Sigma + \tilde{C}^T)Q = GD$$

and those in (3.1) and (3.2), we obtain

$$\Sigma_1 G_{12} + [C_{11}, C_{12}] \begin{bmatrix} G_{12} \\ G_{22} \end{bmatrix} = Q_{12} D_2,$$

$$\Sigma_1 Q_{12} + [C_{11}^T, C_{21}^T] \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} = G_{12} D_2.$$

Therefore,

$$\Sigma_1^2 G_{12} + \Sigma_1 [C_{11}, C_{12}] \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} = \Sigma_1 Q_{12} D_2 = G_{12} D_2^2 - [C_{11}^T, C_{21}^T] \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix} D_2,$$

and we have

$$\begin{aligned} (\sigma_k^2(B) - \sigma_{k+1}^2(A))\|G_{12}\| &\leq \|\Sigma_1^2 G_{12} - G_{12} D_2^2\| \\ &\leq \|[C_{11}^T, C_{21}^T]\| \|D_2\| + \|\Sigma_1 [C_{11}, C_{12}]\|. \end{aligned}$$

Recall that $\|D_2\| = \sigma_{k+1}(A) < \sigma_k(B)$. Furthermore,

$$\|\Sigma_1 [C_{11}, C_{12}]\| = \|\Sigma_1 U_1^T C V\| = \|V_1^T (\mathrm{best}_k(B))^T C\| \leq \|C^T \mathrm{best}_k(B)\|$$

and

$$\|[C_{11}^T, C_{21}^T]\| = \|V_1^T C^T U\| = \|V_1^T C^T\| = \|P_{\mathrm{best}_k(B^T)} C^T\| \leq \|\mathrm{best}_k(B) C^T\| / \sigma_k(B).$$

We obtain

$$\|G_{12}\| \leq (\|\mathrm{best}_k(B) C^T\| + \|C^T \mathrm{best}_k(B)\|) / (\sigma_k^2(B) - \sigma_{k+1}^2(A)).$$

Substituting the above into (3.3) completes the proof. ∎

Remark. For standard perturbation results for the singular subspaces of matrices, the reader is referred to [7].

Parallel to the development in section 2, we consider the case where A is partitioned into two block columns $A = [A_1, A_2]$. Again we first consider dropping A_2 and comparing $\mathrm{best}_k(A)$ with $\mathrm{best}_k([A_1, 0])$.

COROLLARY 3.3. *Let $A = [A_1, A_2]$ and $\sigma_k(A_1) > \sigma_{k+1}(A)$. Then*

$$\|\mathrm{best}_k(A) - \mathrm{best}_k([A_1, 0])\| \leq \frac{\|A\|}{\sigma_k^2(A_1) - \sigma_{k+1}^2(A)} \|A_2^T \mathrm{best}_k(A_1)\|.$$

Proof. Write $A = [A_1, 0] + [0, A_2]$. It is easy to see that $\mathrm{best}_k([A_1, 0])[0, A_2]^T = 0$ and $P_{[A_1, 0]^T}[0, A_2]^T = 0$. The result now follows from Theorem 3.2. ∎

We now derive a bound on the difference between $\mathrm{best}_k(A)$ and the reconstruction based on $\mathrm{best}_{k_1}(A_1)$ and A_2.

COROLLARY 3.4. *Let $A = [A_1, A_2]$ with $\sigma_k([\mathrm{best}_{k_1}(A_1), A_2]) > \sigma_{k+1}(A)$. Then*

$$\|\mathrm{best}_k(A) - \mathrm{best}_k([\mathrm{best}_{k_1}(A_1), A_2])\| \leq \frac{\|A\|\eta}{\sigma_k^2([\mathrm{best}_{k_1}(A_1), A_2]) - \sigma_{k+1}^2(A)},$$

where

$$\eta = \|(A_1 - \text{best}_{k_1}(A_1))^T \text{best}_k([\text{best}_{k_1}(A_1), A_2])\| \le \|(A_1 - \text{best}_{k_1}(A_1))^T A_2\|.$$

Proof. Again we write

$$A = [A_1, A_2] = [\text{best}_{k_1}(A_1), A_2] + [A_1 - \text{best}_{k_1}(A_1), 0] \equiv B + C.$$

It is easy to see that

$$BC^T = 0, \quad P_{\text{best}_k(B^T)} C^T = 0.$$

The result now is a direct consequence of Theorem 3.2. ∎

We finally look at the most general case and bound the difference between $\text{best}_k(A)$ and $\text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])$.

COROLLARY 3.5. *Let* $A = [A_1, A_2]$. *If* $\sigma_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) > \sigma_k(A)$, *then* $\|\text{best}_k(A) - \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])\|$ *is no greater than*

$$\frac{\|A\|\eta}{\sigma_k^2([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]) - \sigma_{k+1}^2(A)},$$

where

$$\begin{aligned} \eta &= \|[A_1 - \text{best}_{k_1}(A_1), A_2 - \text{best}_{k_2}(A_2)]^T \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])\| \\ &\le \max\{\|(A_1 - \text{best}_{k_1}(A_1))^T \text{best}_{k_2}(A_2)\|, \|(A_2 - \text{best}_{k_2}(A_2))^T \text{best}_{k_1}(A_1)\|\}. \end{aligned}$$

Proof. The proof is similar to that of Corollary 3.4, and therefore the proof is omitted. ∎

Remark. It is easy to see that each of the corollaries following Theorem 2.1 is a direct consequence of the corresponding corollaries established above.

Now we return to matrices with low-rank-plus-shift structure, and we consider the case where the structural constraints imposed by the low-rank-plus-shift structure are only approximately satisfied. It turns out that the way this approximation is specified has direct impact on the perturbation bounds we can derive. In the following we prove two theorems, one giving an $O(\sqrt{\epsilon})$ perturbation bound and the other an $O(\epsilon)$ perturbation bound. The difference in the assumptions for the derivation of these two results is rather subtle, but it gives rise to qualitatively different results. We will elaborate on this later with some illustrative examples.

To derive the perturbation bounds, we first need two technical lemmas which were proved in [13]. The two results, especially the first one, are of their own interest as well.

LEMMA 3.6. *Assume the equality*

$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix} = X + E$$

holds for some symmetric matrix E *and symmetric positive semidefinite matrix* X. *Then we have*

$$\|B\| \le \sqrt{(\|A\| + \|E\|)(\|C\| + \|E\|)}.$$

LEMMA 3.7. *Let the symmetric matrix Z be partitioned as*

$$Z = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}.$$

Then $\|Z\| \leq \max\{\|A\|, \|C\|\} + \|B\|$.

THEOREM 3.8. *Let $A = [A_1, A_2] \in \mathcal{R}^{m\times n}$. Assume that for some integer $k < \min\{m, n\}$ there exists $\epsilon \geq 0$ satisfying*[3]

$$\begin{aligned} \lambda_j(A^TA - \sigma^2 I) &> 3\epsilon + \eta, \quad j \leq k, \\ |\lambda_j(A^TA - \sigma^2 I)| &\leq \epsilon, \qquad j > k, \end{aligned}$$

where $\eta = 2\sqrt{\|A^TA - \sigma^2 I\|\epsilon + \epsilon^2} = O(\sqrt{\epsilon})$. Define k_i such that

$$\begin{aligned} \lambda_j(A_i^TA_i - \sigma^2 I) &> \epsilon, \quad j \leq k_i, \\ |\lambda_j(A_i^TA_i - \sigma^2 I)| &\leq \epsilon, \quad j > k_i, \end{aligned}$$

for $i = 1, 2$. Then

$$\|\text{best}_k(A) - \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])\| \leq \frac{\|A\|\eta}{\sigma_k^2(A) - \sigma_{k+1}^2(A) - \eta}.$$

Proof. Define $X \equiv A^TA - \sigma^2 I$. By the eigendecomposition of A^TA and the assumptions on its eigenvalues, we can write $X = Y + E$, where $Y^TE = 0$, and Y is positive semidefinite with $\text{rank}(Y) = k$, $\|E\| \leq \epsilon$, and

$$\lambda_k(Y) = \lambda_k(X) > 3\epsilon + \eta.$$

On the other hand, using the column partition of A, we can write

$$X = \begin{bmatrix} A_1^TA_1 - \sigma^2 I & A_1^TA_2 \\ A_2^TA_1 & A_2^TA_2 - \sigma^2 I \end{bmatrix}.$$

Now for $i = 1, 2$, write the SVD for A_i as follows:

$$A_i = [U_{i1}, U_{i2}]\,\text{diag}(\Sigma_{i1}, \Sigma_{i2})[V_{i1}, V_{i2}]^T,$$

where $\Sigma_{i1} = \text{diag}(\sigma_{i1}, \ldots, \sigma_{i,k_i})$ and $\Sigma_{i2} = \text{diag}(\sigma_{i,k_i+1}, \ldots, \sigma_{i,m_i})$. By definition the integers k_i are chosen such that

$$\begin{aligned} \sigma_{ij}^2 - \sigma^2 &> \epsilon, \quad j \leq k_i, \\ |\sigma_{ij}^2 - \sigma^2| &\leq \epsilon, \quad j > k_i, \end{aligned}$$

for $i = 1, 2$, i.e., $\lambda_j(A_i^TA_i - \sigma^2 I) > \epsilon$ for $j \leq k_i$ and $|\lambda_j(A_i^TA_i - \sigma^2 I)| \leq \epsilon$ for $j > k_i$. It is easy to see that $k_i \leq k$ since $\sigma_{ij} \leq \sigma_j(A)$.

Next we write $A = BW^T \equiv [B_1, B_2]W^T$, where

$$B_1 = [U_{11}\Sigma_{11}, U_{21}\Sigma_{21}], \quad B_2 = [U_{12}\Sigma_{12}, U_{22}\Sigma_{22}],$$

[3]We assume that the eigenvalues of a matrix X are ordered in nonincreasing order $\lambda_1(X) \geq \cdots \geq \lambda_n(X)$.

and

$$W = \begin{bmatrix} V_{11} & 0 & V_{12} & 0 \\ 0 & V_{21} & 0 & V_{22} \end{bmatrix}.$$

Without loss of generality, we assume that W is orthogonal. (Otherwise replace W and B_2 by $[W, W^{\perp}]$ and $[B_2, 0]$, respectively.) Define

$$\Delta = \text{best}_k(A) - \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]).$$

It can be verified that

$$\text{best}_k[\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)] = \text{best}_k[B_1, 0]W^T, \quad \text{best}_k(A) = \text{best}_k(B)W^T,$$
$$\|\Delta\| = \|\text{best}_k(B) - \text{best}_k([B_1, 0])\|.$$

Now in order to apply Corollary 3.3, we need to verify that the condition $\sigma_k(B_1) > \sigma_{k+1}(B)$ holds, and we also need to derive a lower bound on $\sigma_k(B_1)^2 - \sigma_{k+1}^2(B)$ and an upper bound on $\|B_2^T B_1\|$. (Notice that $\|B_2^T \text{best}_k(B_1)\| \le \|B_2^T B_1\|$.) The derivation is done in the following three steps.

(1) We apply Lemma 3.6 twice to obtain an upper bound on $\|B_2^T B_1\|$. It is easy to see that both $B^T B - \sigma^2 I$ and $B_2^T B_2 - \sigma^2 I$ can be written as the sum of a symmetric positive semidefinite matrix and a symmetric matrix with norm no greater than ϵ. Applying Lemma 3.6 to

$$B^T B - \sigma^2 I = \begin{bmatrix} B_1^T B_1 - \sigma^2 I & B_1^T B_2 \\ B_2^T B_1 & B_2^T B_2 - \sigma^2 I \end{bmatrix}$$

gives

$$\begin{aligned} \|B_2^T B_1\| &\le \sqrt{(\|B_1^T B_1 - \sigma^2 I\| + \epsilon)(\|B_2^T B_2 - \sigma^2 I\| \epsilon)} \\ &\le \sqrt{(\|X\| + \epsilon)(\|B_2^T B_2 - \sigma^2 I\| \epsilon)}. \end{aligned}$$

Applying Lemma 3.6 to

$$B_2^T B_2 - \sigma^2 I = \begin{bmatrix} \Sigma_{12}^2 - \sigma^2 I & \Sigma_{12} U_{12}^T U_{22} \Sigma_{22} \\ \Sigma_{22} U_{22}^T U_{12} \Sigma_{12} & \Sigma_{22}^2 - \sigma^2 I \end{bmatrix}$$

yields

$$\|\Sigma_{12} U_{12}^T U_{22} \Sigma_{22}\| \le (\|\Sigma_{12}^2 - \sigma^2 I\| + \epsilon)(\|\Sigma_{22}^2 - \sigma^2 I\| + \epsilon) \le 4\epsilon^2,$$

where we have used $\|\Sigma_{i2}^2 - \sigma^2 I\| \le \epsilon$. By Lemma 3.7, we obtain $\|B_2^T B_2 - \sigma^2 I\| \le 3\epsilon$ and hence

$$\|B_2^T B_1\| \le 2\sqrt{\|X\|\epsilon + \epsilon^2} \equiv \eta.$$

(2) We now derive a lower bound on $\sigma_k(B_1)^2 - \sigma_{k+1}^2(B)$. Write

$$B^T B - \sigma^2 I = \begin{bmatrix} B_1^T B_1 - \sigma^2 I & \\ & B_2^T B_2 - \sigma^2 I \end{bmatrix} + \begin{bmatrix} & B_1^T B_2 \\ B_2^T B_1 & \end{bmatrix}.$$

Using perturbation bounds for eigenvalues, we have

$$\lambda_k(X) = \lambda_k(B^T B - \sigma^2 I) \le \lambda_k(\,\text{diag}(B_1^T B_1 - \sigma^2 I, B_2^T B_2 - \sigma^2 I)) + \eta.$$

The inequality $\lambda_k(X) > 3\epsilon + \eta$ implies that

$$\lambda_k(B_1^T B_1 - \sigma^2 I) > \|B_2^T B_2 - \sigma^2 I\|$$

because $\|B_2^T B_2 - \sigma^2 I\| \leq 3\epsilon$. Thus

$$|\sigma_k^2(B_1) - \sigma_k^2(A)| = |\lambda_k(B_1^T B_1 - \sigma^2 I) - \lambda_k(X)| \leq \|B_1^T B_2\| \leq \eta.$$

It follows that

$$\sigma_k^2(B_1) - \sigma_{k+1}^2(B) \geq \sigma_k^2(A) - \eta - \sigma_{k+1}^2(A) \geq \epsilon + \eta > 0.$$

(3) Finally, by Corollary 3.3, we have

$$\|\Delta\| \leq \frac{\|B\|\|B_1^T B_2\|}{\sigma_k^2(B_1) - \sigma_{k+1}^2(B)} \leq \frac{\|A\|\eta}{\sigma_k^2(A) - \sigma_{k+1}^2(A) - \eta},$$

completing the proof. □

A natural question to ask is whether the bound derived in Theorem 3.8 is tight or not. In the following, we provide an example for which the bound in Theorem 3.8 is achievable.

Example 1. *Let s be small, and for any $\sigma > s$, define*

$$c_1 = \sqrt{1 + \sigma^2 + s^2}, \quad c_2 = \sqrt{\sigma^2 + s^2}, \quad c_3 = \sqrt{\sigma^2 + s^2}, \quad \epsilon = c_1^2 s^2.$$

Let $A = [A_1, A_2]$ with

$$A_1 = \frac{1}{\sqrt{1+s^2}} \begin{bmatrix} D \\ sDJ \end{bmatrix}, \quad A_2 = \frac{1}{\sqrt{1+s^2}} \begin{bmatrix} -sDJ \\ D \end{bmatrix}, \quad \text{where } J = \begin{bmatrix} & & 1 \\ & 1 & \\ 1 & & \end{bmatrix},$$

and $D = \operatorname{diag}(c_1, c_2, c_3)$. It follows that

$$A = \frac{1}{\sqrt{1+s^2}} \begin{bmatrix} D & \\ & D \end{bmatrix} \begin{bmatrix} I & -sJ \\ sJ & I \end{bmatrix}.$$

It can be verified that

$$\begin{aligned} &\lambda_{1,2}(A^T A - \sigma^2 I) = 1 + s^2 > \epsilon > |\lambda_j(A^T A - \sigma^2 I)|, && j \geq 3, \\ &\lambda_1(A_i^T A_i - \sigma^2 I) = 1 + s^2(1 + \sigma^2 - s^2) > \epsilon \geq |\lambda_j(A_i^T A_i - \sigma^2 I)|, && j \geq 2, \end{aligned}$$

for $i = 1, 2$. Hence $k = 2$, and $k_1 = k_2 = 1$. A simple computation shows that

$$\operatorname{best}_k(A) = \frac{c_1}{\sqrt{1+s^2}}[e_1, e_3][e_1 - se_6, se_3 + e_4]^T,$$

$$\operatorname{best}_{k_1}(A_1) = \frac{1}{\sqrt{1+s^2}}[ce_1 + sc_3 e_6, 0, 0], \quad \operatorname{best}_{k_1}(A_1) = \frac{1}{\sqrt{1+s^2}}[-sc_3 e_3 + c_1 e_4, 0, 0],$$

where the e_i's are the canonical unit vectors. So we have

$$\operatorname{best}_k([\operatorname{best}_{k_1}(A_1), \operatorname{best}_{k_2}(A_2)]) = [\operatorname{best}_{k_1}(A_1), \operatorname{best}_{k_2}(A_2)]$$

and

$$\|\text{best}_k(A) - [\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)]\| = \frac{sc_1}{\sqrt{1+s^2}} = \sqrt{\frac{\epsilon}{1+s^2}} = O(\sqrt{\epsilon}).$$

We now impose another set of conditions on the perturbation E so that an $O(\epsilon)$ bound can be derived. Before we proceed, some motivations for imposing those conditions are in order: recall that we have always implicitly assumed that $\sigma_k(B) > \sigma_{k+1}(B)$ whenever we discuss $\text{best}_k(B)$ for a given matrix B. If the matrix B is perturbed by an amount of order $O(\epsilon)$, it makes sense to impose the constraint that $\sigma_k(B) - \sigma_{k+1}(B) > \epsilon$. This constraint is roughly the same as

$$\lambda_k(B^TB - \sigma^2 I) > \epsilon \geq |\lambda_j(B^TB - \sigma^2 I)|$$

for $j > k$ if B has the low-rank-plus-shift structure. If we impose this constraint on $A = [A_1, A_2]$, i.e., substituting B with $A = [A_1, A_2]$, we are no longer free to choose arbitrary $k_i \leq k$ for the rank of the truncated SVD of the submatrix A_i. The integers k_i will be automatically determined so that $\sigma_{k_i}(A_i)$ is much greater than those smallest singular values $\sigma_j(A_i)$ for $j > k_i$. As mentioned in the proof of Theorem 3.8, we always have $k_i \leq k$. In the example given above, $k_i < k$ and the perturbation bound $O(\sqrt{\epsilon})$ is shown to be achievable. In the following theorem we will show that an $O(\epsilon)$ perturbation bound is also possible if $k_1 = k_2 = k$.

THEOREM 3.9. *Let $A = [A_1, A_2]$. If there exists $\epsilon < \sigma^2$ and integer k such that*

$$\lambda_k(A^TA - \sigma^2 I) > \epsilon \geq |\lambda_j(A^TA - \sigma^2 I)|$$

for $j \geq k+1$, and $\lambda_k(A_i^TA_i - \sigma^2 I) > \epsilon, i = 1, 2$, then

$$\|(A_1 - \text{best}_k(A_1))^TA_2\| \leq \eta_1, \quad \|(A_2 - \text{best}_k(A_2))^TA_1\| \leq \eta_2, \tag{3.4}$$

and

$$\|\text{best}_k(A) - \text{best}_k([\text{best}_{k_1}(A_1), \text{best}_{k_2}(A_2)])\| \leq \frac{\|A\|\eta}{\lambda_{\max} - \epsilon}, \tag{3.5}$$

where

$$\eta_i = \left(\sigma + 2\|A\| + \frac{2\|A\|^3}{\lambda_k(A_i^TA_i - \sigma^2 I)}\right)\frac{\epsilon}{\sigma + \sqrt{\sigma^2 - \epsilon}}, \quad i = 1, 2,$$

$$\eta = \max\{\eta_1, \eta_2\} = \left(\sigma + 2\|A\| + \frac{2\|A\|^3}{\lambda_{\min}}\right)\frac{\epsilon}{\sigma + \sqrt{\sigma^2 - \epsilon}},$$

$$\lambda_{\max} = \max_{i=1,2}\lambda_k(A_i^TA_i - \sigma^2 I), \quad \lambda_{\min} = \min_{i=1,2}\lambda_k(A_i^TA_i - \sigma^2 I).$$

Proof. We will use Corollary 3.5 to prove the theorem. Since

$$\sigma_k([\text{best}_k(A_1), \text{best}_k(A_2)]) \geq \max_{i=1,2}\sigma_k(A_i),$$

it follows that (denoting $\Delta = \sigma_k^2([\text{best}_k(A_1), \text{best}_k(A_2)]) - \sigma_{k+1}^2(A)$)

$$\begin{aligned}\Delta &\geq \max_{i=1,2}(\sigma_k^2(A_i) - \sigma^2) - (\sigma_{k+1}^2(A) - \sigma^2) \\ &= \max_{i=1,2} \lambda_k(A_i^T A_i - \sigma^2 I) - \lambda_{k+1}(A^T A - \sigma^2 I) \\ &\geq \lambda_{\max} - \epsilon.\end{aligned}$$

Therefore the result (3.5) holds provided we can establish (3.4). To prove (3.4) we will construct a matrix $\tilde{A}$ which is close to A so that the inequality $\|\tilde{A} - A\| \leq \eta$ holds and its partitioned blocks $\tilde{A}_1$ and $\tilde{A}_2$ with $\tilde{A} = [\tilde{A}_1, \tilde{A}_2]$ satisfy

$$(\tilde{A}_1 - \text{best}_k(\tilde{A}_1))^T \tilde{A}_2 = 0, \quad (\tilde{A}_2 - \text{best}_k(\tilde{A}_2))^T \tilde{A}_1 = 0.$$

Theorem 3.2 will then be used to estimate $\|\text{best}_k(\tilde{A}_i) - \text{best}_k(A_i)\|$. To this end, denote $\lambda_j = \lambda_j(A^T A - \sigma^2 I)$ and define

$$\Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_k), \quad \Lambda_2 = \text{diag}(\lambda_{k+1}, \ldots, \lambda_n).$$

Then the eigendecomposition of $A^T A - \sigma^2 I$ and the SVD of A can be written as

$$A^T A - \sigma^2 I = V \,\text{diag}(\Lambda_1, \Lambda_2) V^T, \quad A = U \,\text{diag}(\sqrt{\Lambda_1 + \sigma^2 I}, \sqrt{\Lambda_2 + \sigma^2 I}) V^T,$$

respectively, for some orthogonal matrices U and V. Let

$$E = U \,\text{diag}(0, \sigma I - \sqrt{\Lambda_2 + \sigma^2 I}) V^T.$$

It can be verified that $\|E\| \leq \epsilon/(\sigma + \sqrt{\sigma^2 - \epsilon}) \equiv \tau$, and the matrix

$$\tilde{A} \equiv A + E = U \,\text{diag}(\sqrt{\Lambda_1 + \sigma^2 I}, \sigma I) V^T$$

has the low-rank-plus-shift structure. Now partition

$$E = [E_1, E_2], \quad \tilde{A} = [\tilde{A}_1, \tilde{A}_2]$$

conformally with the partition of A. Then $\|E_i\| \leq \tau$. Since

$$A^T A - \sigma^2 I = \tilde{A}^T \tilde{A} - \sigma^2 I + \tilde{E}, \quad \tilde{E} = V \,\text{diag}(0, \Lambda_2) V^T = (\tilde{E}_{i,j})_{i,j=1}^2,$$

it can be verified that $A_i^T A_i - \sigma^2 I = \tilde{A}_i^T \tilde{A}_i - \sigma^2 I + \tilde{E}_{ii}$, and we have

$$\begin{aligned}\lambda_k(\tilde{A}_i^T \tilde{A}_i - \sigma^2 I) &\geq \lambda_k(A_i^T A_i - \sigma^2 I) - \|\tilde{E}_{ii}\| \\ &\geq \lambda_k(A_i^T A_i - \sigma^2 I) - \epsilon > 0.\end{aligned}$$

It follows that $\text{rank}(\tilde{A}_i^T \tilde{A}_i - \sigma^2 I) = k$. By Theorem 2.7, we have

$$(\tilde{A}_1 - \text{best}_k(\tilde{A}_1))^T \tilde{A}_2 = 0, \quad (\tilde{A}_2 - \text{best}_k(\tilde{A}_2))^T \tilde{A}_1 = 0.$$

Let $\Delta_i = \text{best}_k(\tilde{A}_i) - \text{best}_k(A_i) - E_i$; then we have

$$\|\Delta_i\| \leq \|\text{best}_k(\tilde{A}_i) - \text{best}_k(A_i)\| + \|E_i\|.$$

It follows from Theorem 3.2 that

$$\begin{aligned}\|\text{best}_k(\tilde{A}_i) - \text{best}_k(A_i)\| &\leq \left(1 + \frac{2\|\tilde{A}_i\|\|A_i\|}{\sigma_k^2(A_i) - \sigma_{k+1}^2(\tilde{A}_i)}\right) \|E_i\| \\ &= \left(1 + \frac{2\|A\|^2}{\sigma_k^2(A_i) - \sigma^2}\right) \|E_i\|,\end{aligned}$$

and therefore

$$\|\Delta_i\| \le \left(2 + \frac{2\|A\|^2}{\lambda_k(A_i^T A_i - \sigma^2 I)}\right)\tau.$$

Here we have used $\|A_i\| \le \|A\|, \|\tilde{A}_i\| \le \|\tilde{A}\| \le \|A\|$, and $\sigma_{k+1}(\tilde{A}_i) = \sigma$. Since

$$A_i - \text{best}_k(A_i) = \tilde{A}_i - \text{best}_k(\tilde{A}_i) + \Delta_i,$$

we have

$$\begin{aligned}\|(A_1 - \text{best}_k(A_1))^T A_2\| &= \|\Delta_1^T A_2 - (\tilde{A}_1 - \text{best}_k(\tilde{A}_1))E_2\| \\ &\le \|A_2\|\|\Delta_1\| + \sigma_{k+1}(\tilde{A}_1)\|E_2\| \\ &\le (\sigma + 2\|A\| + 2\|A\|^3/\lambda_k(A_1^T A_1 - \sigma^2 I))\tau = \eta_1.\end{aligned}$$

We can similarly prove $\|(A_2 - \text{best}_k(A_2))^T A_1\| \le \eta_2$. Therefore (3.4) holds, completing the proof. □

Remark. Notice that the condition $\lambda_k(A_i^T A_i - \sigma^2 I) > \epsilon$ implies that $k_1 = k_2 = k$. In order for the perturbation bound to be of order $O(\epsilon)$, $\lambda_{\min}$ needs to be of order $O(1)$ provided $\lambda_{\max} \gg \epsilon$.

Example 2. *Now we construct a class of matrices that satisfy the conditions of Theorem* 3.9. *For any orthonormal matrices U_1 and V_1 with k columns, let $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_k)$, where $\lambda_i \gg \sigma^2 > 0$ for $i = 1, \ldots, k$. Let*

$$D_1 = (\Lambda\sqrt{\Lambda^2 + \epsilon I})^{-1}(\Lambda^2 - (\sigma^2 - \epsilon)I), \quad D_2 = \sqrt{I - D_1^2}, \quad U_2 = [U_1, U_1^{\perp}][D_1, D_2]^T,$$

where $U_1^{\perp}$ is any orthonormal matrix of k columns that is orthogonal to U_1. Define

$$A_1 = U_1 \Lambda V_1^T, \quad A_2 = U_2\sqrt{\Lambda^2 + \epsilon I}V_1^T.$$

It follows that

$$X = [A_1, A_2]^T[A_1, A_2] - \sigma^2 I = \begin{bmatrix} A_1^T A_1 - \sigma^2 I & A_1^T A_2 \\ A_2^T A_1 & A_2^T A_2 - \sigma^2 I \end{bmatrix}$$

$$= \begin{bmatrix} A_1^T A_1 - \sigma^2 I & A_1^T A_1 - \sigma^2 I + \epsilon \\ A_2^T A_1 & A_2^T A_1 \end{bmatrix} = \begin{bmatrix} A_2^T A_2 - \sigma^2 I & A_2^T A_2 - \sigma^2 I \\ A_2^T A_2 - \sigma^2 I & A_2^T A_2 - \sigma^2 I \end{bmatrix} - \begin{bmatrix} \epsilon & 0 \\ 0 & 0 \end{bmatrix}.$$

Hence,

$$\begin{aligned}\lambda_j(X) &\ge 2(\lambda_j - \sigma^2 + \epsilon) \gg \epsilon, \quad j \le k, \\ |\lambda_j(X)| &\le \epsilon, \qquad\qquad\qquad\quad j > k,\end{aligned}$$

and by definition $k_1 = k_2 = k$.

Remark. For the case where $k_1 < k$ and $k_2 < k$, if we replace $\text{best}_{k_1}(A_1)$ and $\text{best}_{k_2}(A_2)$ by $\text{best}_k(A_1)$ and $\text{best}_k(A_2)$, respectively, the error

$$\|\Delta\| = \|\text{best}_k(A) - \text{best}_k[\text{best}_k(A_1), \text{best}_k(A_2)]\|$$

may still be $O(\sqrt{\epsilon})$. For example, in Example 1, we have

$$\text{best}_2[\text{best}_2(A_1), \text{best}_2(A_2)] = \text{best}_2[\text{best}_1(A_1), \text{best}_1(A_2)].$$

Therefore,

$$\|\text{best}_2(A) - \text{best}_2[\text{best}_2(A_1), \text{best}_2(A_2)]\| = \sqrt{\frac{\epsilon}{1 + s^2}}.$$

Acknowledgments. Part of this work was done while both authors were visiting the National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory. The authors wish to thank Dr. Horst Simon for his hospitality and support. The authors also want to thank the anonymous referees for their comments and suggestions that greatly improved the presentation of the paper.

REFERENCES

[1] M. Berry, *Large scale singular value computations*, Internat. J. Supercomputer Appl., 6 (1992), pp. 13–49.

[2] M.W. Berry, S.T. Dumais, and G.W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Rev., 37 (1995), pp. 573–595.

[3] J. Cullum, R. A. Willoughby, and M. Lake, *A Lanczos algorithm for computing singular values and vectors of large matrices*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 197–215.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[5] B.N. Parlett, *The Symmetric Eigenvalue Problem,* Classics Appl. Math. 20, SIAM, Philadelphia, 1998.

[6] H. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[7] P. Wedin, *Perturbation bounds in connection with the singular value decomposition*, BIT, 12 (1972), pp. 99–111.

[8] G. Xu and T. Kailath, *Fast subspace decomposition*, IEEE Trans. Signal Process., 42 (1994), pp. 539–551.

[9] G. Xu, H. Zha, G. Golub, and T. Kailath, *Fast algorithms for updating signal subspaces*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 41 (1994), pp. 537–549.

[10] H. Zha and H.D. Simon, *On updating problems in latent semantic indexing*, SIAM J. Sci. Comput., 21 (1999), pp. 782–791.

[11] H. Zha and H. Simon, *A subspace-based model for latent semantic indexing in information retrieval*, in Proceedings of Interface '98, Berkeley, Springer-Verlag, New York, 1998, pp. 315–320.

[12] H. Zha, O. Marques, and H. Simon, *Large-scale SVD and subspace-based methods for information retrieval*, in Proceedings of Irregular '98, Lecture Notes in Comput. Sci. 1457, Springer-Verlag, New York, 1998, pp. 29–42.

[13] H. Zha and Z. Zhang, *Matrices with low-rank-plus-shift structure: Partial SVD and latent semantic indexing*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 522–536.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1263–1273

REAL HAMILTONIAN POLAR DECOMPOSITIONS*

CORNELIS V. M. VAN DER MEE†, ANDRÉ C. M. RAN‡, AND LEIBA RODMAN§

Abstract. For a given real invertible skew-symmetric matrix H, we characterize the real $2n \times 2n$ matrices X that allow an H-Hamiltonian polar decomposition of the type $X = UA$, where U is a real H-symplectic matrix ($U^T HU = H$) and A is a real H-Hamiltonian matrix ($HA = -A^T H$).

Key words. polar decomposition, Hamiltonian matrices, skew-Hamiltonian matrices, symplectic matrices

AMS subject classifications. 15A23, 47B50

PII. S0895479899362788

1. Introduction.

It is well known that every square matrix X allows a polar decomposition $X = UA$, where U is unitary and A is self-adjoint, and the proof of this fact is straightforward. When unitarity and self-adjointness are required to hold with respect to the indefinite scalar product $[x, y] = \langle Hx, y\rangle$ with H an invertible self-adjoint matrix, the theory of the H-polar decompositions $X = UA$, where U is H-unitary (i.e., $[Ux, Uy] = [x, y]$ for all vectors x, y) and A is H-self-adjoint (i.e., $[Ax, y] = [x, Ay]$ for all vectors x, y), is much more complicated and has been developed in [2, 3, 4]. Introducing the H-adjoint $X^{[*]}$ of X by $X^{[*]} = H^{-1}X^*H$ with X^* the usual adjoint (so that U is H-unitary if and only if U is invertible and $U^{-1} = U^{[*]}$, and A is H-self-adjoint if and only if $A^{[*]} = A$), an H-polar decomposition of a matrix X exists if and only if there exists an H-self-adjoint matrix A satisfying

$$X^{[*]}X = A^2, \quad \operatorname{Ker} X = \operatorname{Ker} A, \tag{1.1}$$

where the symbol Ker denotes the null space of a matrix. The H-unitary factor U is then constructed as an H-unitary extension (a so-called Witt extension) of the H-isometry $V : \operatorname{Im} A \to \operatorname{Im} X$ satisfying $Xy = VAy$ for every vector y. An H-polar decomposition of a given matrix X need not always exist, X may have many "nonequivalent" H-polar decompositions, and there exist various interesting subclasses of H-polar decompositions. Moreover, there exists a fairly complete stability theory for H-polar decompositions [6].

The situation is quite different for Hamiltonian polar decompositions, introduced below. Dealing exclusively with real matrices, we fix an invertible $2n \times 2n$ real matrix H such that $H = -H^T$. Without loss of generality we may assume that

$$H = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}.$$

*Received by the editors October 28, 1999; accepted for publication (in revised form) by P. Van Dooren October 9, 2000; published electronically April 6, 2001.
http://www.siam.org/journals/simax/22-4/36278.html

†Dipartimento di Matematica, Università di Cagliari, Via Ospedale 72, 09124 Cagliari, Italy (cornelis@krein.unica.it). The work of this author was partially supported by INDAM and MURST.

‡Divisie Wiskunde en Informatica, Faculteit Exacte Wetenschappen, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands (ran@cs.vu.nl).

§Department of Mathematics, The College of William and Mary, Williamsburg, VA 23187-8795 (lxrodm@math.wm.edu). The work of this author was partially supported by NSF grant DMS 9800704 and by a Faculty Research Assignment grant from the College of William and Mary.

A real $2n \times 2n$ matrix X is called *H-Hamiltonian* if $HX = -X^T H$, and *H-skew-Hamiltonian* if $HX = X^T H$. Denoting $X^{[*]} = H^{-1}X^T H$, we see that X is H-Hamiltonian if and only if $X^{[*]} = -X$, and H-skew-Hamiltonian if and only if $X^{[*]} = X$. Defining a real matrix U to be *H-symplectic* if $U^T HU = H$ or, equivalently, $U^{[*]} = U^{-1}$, and *H-antisymplectic* if $U^T HU = -H$ or, equivalently, $U^{[*]} = -U^{-1}$, we can in principle study four different polar decomposition problems for a given real $2n \times 2n$ matrix X, namely, we can study the problem of representing such X in the form $X = UA$, where U is H-symplectic (or H-antisymplectic) and A is H-Hamiltonian (or H-skew-Hamiltonian). In this article we will limit ourselves to *H-Hamiltonian polar decompositions* only, i.e., to representations of X of the type $X = UA$, where U is H-symplectic and A is H-Hamiltonian.

All matrices in sections 1 *and* 2 *are assumed to be real.*

The following result is immediate. For the sake of completeness we present a short proof.

THEOREM 1.1. *A real $2n \times 2n$ matrix X has an H-Hamiltonian polar decomposition if and only if there exists an H-Hamiltonian matrix A such that $A^2 = -X^{[*]}X$ and* $\operatorname{Ker} A = \operatorname{Ker} X$.

Proof. The necessity is clear: if $X = UA$ is an H-Hamiltonian polar decomposition, then

$$X^{[*]}X = A^{[*]}U^{[*]}UA = A^{[*]}A = -A^2,$$

and $\operatorname{Ker} A = \operatorname{Ker} X$ holds as well. Conversely, if an H-Hamiltonian matrix A exists with the properties as described in the theorem, then there exists an invertible map $U_0 : \operatorname{Im} A \to \operatorname{Im} X$ defined by the equality $U_0 Ay = Xy$ for every $y \in \mathbb{R}^{2n}$. Letting $[x, y] = \langle Hx, y\rangle$ be the skew-symmetric scalar product induced by H, we now have

$$\begin{aligned}[U_0 Ax, U_0 Ay] &= [Xx, Xy] = [X^{[*]}Xx, y] = [-A^2 x, y] \\ &= [A^{[*]}Ax, y] = [Ax, Ay], \quad x, y \in \mathbb{R}^{2n}.\end{aligned}$$

In other words, U_0 is an H-isometry. By a version of Witt's theorem (see Theorem 4.2 of [3]), U_0 can be extended to an H-symplectic linear transformation U on the whole of $\mathbb{R}^{2n}$. Thus, we obviously have an H-Hamiltonian polar decomposition $X = UA$. ∎

The following result recently proved in [1] greatly simplifies the problem of characterizing the real matrices X having an H-Hamiltonian polar decomposition.

THEOREM 1.2. *Every H-skew-Hamiltonian matrix is a square of an H-Hamiltonian matrix. Moreover, for every H-skew-Hamiltonian matrix A there exists an H-symplectic matrix U such that*

$$U^{-1}AU = \begin{bmatrix} B & 0 \\ 0 & B^T \end{bmatrix} \tag{1.2}$$

for some matrix B. Furthermore, B can be chosen in a real Jordan form.

Every matrix of the form $X^{[*]}X$ is obviously H-skew-Hamiltonian. The converse is also true, as stated in the following result. Proposition 1.3 is to be contrasted with the corresponding results for the symmetric (in the real case) or Hermitian (in the complex case) indefinite scalar products (see [6]): There the three classes of matrices A for which $A = A^{[*]}$, or $A = X^{[*]}X$ for some X, or $A = X^{[*]}X$ for some X such that $\operatorname{Ker} X = \operatorname{Ker} A$, are all different.

PROPOSITION 1.3. *Let A be H-skew-Hamiltonian. Then there exists a matrix X such that $A = X^{[*]}X$ and* $\operatorname{Ker} A = \operatorname{Ker} X$.

Proof. By Theorem 1.2 we may (and do) assume that

$$H = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \quad A = \begin{bmatrix} B & 0 \\ 0 & B^T \end{bmatrix}$$

for some matrix B. We then let

$$X = \begin{bmatrix} C & 0 \\ 0 & G \end{bmatrix},$$

where C and G are such that $B = G^T C$. Then the equality $A = X^{[*]}X$ is easily verified.

To ensure the condition $\operatorname{Ker} A = \operatorname{Ker} X$ we need $\operatorname{Ker} C = \operatorname{Ker} B$ and $\operatorname{Ker} G = \operatorname{Ker} B^T$. To this end, write the singular value decomposition $B = UDV$, where U and V are real orthogonal and D is diagonal with nonnegative entries, and put $C = \sqrt{D}V$, $G = (U\sqrt{D})^T$. ∎

Analogously one proves that for every H-skew-Hamiltonian matrix A there exists X such that $A = -X^{[*]}X$ and $\operatorname{Ker} A = \operatorname{Ker} X$.

PROPOSITION 1.4. *Every invertible $2n \times 2n$ matrix X has an H-Hamiltonian polar decomposition.*

Proof. By Theorem 1.2, there exists an H-Hamiltonian matrix A such that $A^2 = -X^{[*]}X$. Since X is invertible, the condition $\operatorname{Ker} A = \operatorname{Ker} X$ is trivially satisfied. By Theorem 1.1, we are done. As a matter of fact, the H-symplectic factor is given by $U = XA^{-1}$. ∎

There are examples of matrices that do not have any H-Hamiltonian polar decompositions.

Example 1.5. Let

$$H = \begin{bmatrix} 0 & I_2 \\ -I_2 & 0 \end{bmatrix}$$

be 4×4, where I_2 stands for the identity matrix of order 2. The matrix

$$W = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

is H-skew-Hamiltonian, so by Proposition 1.3 there exists X such that $W = -X^{[*]}X$ and $\operatorname{Ker} W = \operatorname{Ker} X$, for example,

$$X = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

On the other hand, a direct verification shows (see below) that there is no H-Hamiltonian square root V of W such that

$$\operatorname{Ker} V = \operatorname{Ker} W. \tag{1.3}$$

By Theorem 1.1, X has no H-Hamiltonian polar decomposition.

To verify that there is no H-Hamiltonian square root V of W with the property (1.3), assume that V is one. Being H-Hamiltonian, V must be of the form

$$V = \begin{bmatrix} E & F \\ G & -E^T \end{bmatrix},$$

where F and G are symmetric 2×2 matrices. Condition (1.3) implies that V must have a first and a last (fourth) column consisting of zeros. Thus, V must have the form

$$V = \begin{bmatrix} 0 & -p & q & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & r & p & 0 \end{bmatrix}$$

for some $p, q, r \in \mathbb{R}$. But then $V^2 = 0$, a contradiction with $V^2 = W$.

The main result in [1], i.e., Theorem 1.2, allows us to write a short paper. The main result on the characterization of the real matrices X allowing an H-Hamiltonian polar decomposition is stated and proved in section 2. The final section 3 is devoted to a comparison of the main result to existing results on K-polar decomposition for the invertible self-adjoint matrix $K = iH$.

2. Main result. To formulate and prove our main result, we need canonical forms for H-Hamiltonian and H-skew-Hamiltonian matrices, where H is a fixed real invertible skew-symmetric matrix. We will state these forms only to the extent in which they are needed in our proofs. For the complete canonical forms for H-Hamiltonian and H-skew-Hamiltonian matrices, as well as for pairs of matrices with related symmetries, see, e.g., [5, 7].

LEMMA 2.1.

(a) *Let A be H-skew-Hamiltonian. Then there exists an invertible real matrix S such that $\tilde{A} = S^{-1}AS$ and $\tilde{H} = S^THS$ have the following form:*

$$\tilde{A} = \bigoplus_{j=1}^{k} \left(J_j \oplus (J_j)^T\right),$$
$$\tilde{H} = \bigoplus_{j=1}^{k} \begin{bmatrix} 0 & I_{p_j} \\ -I_{p_j} & 0 \end{bmatrix},$$

where J_j is a real Jordan block of size $p_j \times p_j$ corresponding either to a real eigenvalue or to a pair of nonreal complex eigenvalues.

(b) *Let A be H-Hamiltonian. Then there exists an invertible real matrix S such that $\tilde{A} = S^{-1}AS$ and $\tilde{H} = S^THS$ have the following form:*

$$\tilde{A} = \tilde{A}_0 \oplus \left[\bigoplus_{j=1}^{k} J_{2p_j}(0)\right] \oplus \left[\bigoplus_{j=k+1}^{\ell} \left(J_{2p_j+1}(0) \oplus -J_{2p_j+1}(0)^T\right)\right],$$
$$\tilde{H} = \tilde{H}_0 \oplus \left[\bigoplus_{j=1}^{k} \varepsilon_j F_{2p_j}\right] \oplus \left[\bigoplus_{j=k+1}^{\ell} \begin{bmatrix} 0 & I_{2p_j+1} \\ -I_{2p_j+1} & 0 \end{bmatrix}\right],$$

where $\tilde{A}_0$ *is invertible,* $J_q(0)$ *stands for the* $q \times q$ *nilpotent (upper triangular) Jordan block,* ε_j *are signs* ± 1*, and*

$$F_{2p} = \begin{bmatrix} & & & & 1 \\ & & & -1 & \\ & & \iddots & & \\ & 1 & & & \\ -1 & & & & \end{bmatrix}$$

is the $2p \times 2p$ *skew-symmetric matrix with zeros off the trailing diagonal.*

We now state our main result.

THEOREM 2.2. *Let* X *be a real matrix and* H *a real skew-symmetric invertible matrix. Then there exists an* H*-Hamiltonian polar decomposition of* X *if and only if the part of the canonical form of* $(X^{[*]}X, H)$*, as presented in Lemma* 2.1(a)*, corresponding to the zero eigenvalue of* $X^{[*]}X$*, can be represented in the block diagonal form*

$$(\operatorname{diag}(B_r)_{r=0}^{m}, \operatorname{diag}(G_r)_{r=0}^{m}),$$

where

(i) B_0 *is the zero matrix of order* $2k_0$ *and*

$$G_0 = \begin{bmatrix} 0 & I_{k_0} \\ -I_{k_0} & 0 \end{bmatrix},$$

(ii) $m = m_1 + m_2$*, and for each* $r = 1, \dots, m_1$ *we have*

$$B_r = \begin{bmatrix} J_{k_r}(0) & 0 \\ 0 & J_{k_r}(0)^T \end{bmatrix}, \qquad G_r = \begin{bmatrix} 0 & I_{k_r} \\ -I_{k_r} & 0 \end{bmatrix}, \tag{2.1}$$

while for $r = m_1 + 1, \dots, m_1 + m_2$ *we have*

$$B_r = \begin{bmatrix} J_{k_r}(0) & 0 & 0 & 0 \\ 0 & J_{k_r-1}(0) & 0 & 0 \\ 0 & 0 & J_{k_r}(0)^T & 0 \\ 0 & 0 & 0 & J_{k_r-1}(0)^T \end{bmatrix},$$

$$G_r = \begin{bmatrix} 0 & I_{2k_r-1} \\ -I_{2k_r-1} & 0 \end{bmatrix}, \tag{2.2}$$

(iii) *and, denoting the corresponding basis in* $\operatorname{Ker}(X^{[*]}X)^{2n} \subseteq \mathbb{R}^{2n}$ *in which the form* (i)*,* (2.1)*,* (2.2) *is achieved by* $\{e_{r,j}\}_{r=0,j=1}^{m,\ell_r}$*, where* $\ell_0 = 2k_0$*,* $\ell_r = 2k_r$ *for* $r = 1, \dots, m_1$*, and* $\ell_r = 4k_r - 2$ *for* $r = m_1 + 1, \dots, m_2$*, we have*

$$\begin{aligned} \operatorname{Ker} X = {} & \operatorname{span}\{e_{r,1} + \varepsilon_r e_{r,2k_r} \mid r = 1, \dots, m_1\} \\ & + \operatorname{span}\{e_{r,1}, e_{r,4k_r-2} \mid r = m_1 + 1, \dots, m_2\} \\ & + \operatorname{span}\{e_{0,j}\}_{j=1}^{2k_0} \end{aligned} \tag{2.3}$$

for some numbers $\varepsilon_r = \pm 1$.

Note that there may be more than one way to divide the part of the canonical form of $(X^{[*]}X, H)$ corresponding to the zero eigenvalue of $X^{[*]}X$ into blocks of the

form (i), of the form (2.1), and of the form (2.2). Also, for some bases in which the form (i), (2.1), (2.2) is achieved the formula (2.3) may be valid, and for some other bases in which the form (i), (2.1), (2.2) is achieved the formula (2.3) may not be valid. Theorem 2.2 says that a necessary and sufficient condition for existence of an H-Hamiltonian polar decomposition of X is that *there is* a suitable division of the part of the canonical form of $(X^{[*]}X, H)$ corresponding to the zero eigenvalue of $X^{[*]}X$ into the blocks of the forms (i), (2.1), and (2.2), and *there is* a suitable basis in which this division is achieved so that (2.3) is valid.

Proof. First of all we show that the proof can be reduced to the case when $X^{[*]}X$ is nilpotent.

By Lemma 2.1 we may let

$$X^{[*]}X = S^{-1}\begin{bmatrix} Z_1 & 0 \\ 0 & Z_0 \end{bmatrix} S, \qquad H = S^T \begin{bmatrix} H_1 & 0 \\ 0 & H_0 \end{bmatrix} S,$$

where Z_1 is invertible and Z_0 is nilpotent. Replacing X by $S^{-1}XS$ and H by S^THS, we see that we may assume without loss of generality that

$$X^{[*]}X = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_0 \end{bmatrix}, \quad H = \begin{bmatrix} H_1 & 0 \\ 0 & H_0 \end{bmatrix},$$

with Z_1 invertible and Z_0 nilpotent and Z_i being H_i-skew Hamiltonian for $i = 0, 1$. So, for the sake of the present argument, we shall assume that this is the case. Then

$$(X^{[*]}X)^n = \begin{bmatrix} Z_1^n & 0 \\ 0 & 0 \end{bmatrix}.$$

We see that

$$\mathbb{R}^{2n} = \operatorname{Im}(X^{[*]}X)^n \oplus \operatorname{Ker}(X^{[*]}X)^n. \tag{2.4}$$

Note also $\operatorname{Ker} X \subseteq \operatorname{Ker}(X^{[*]}X)^n$. Define $\tilde{X}$ as follows: $\tilde{X}x = 0$ for $x \in \operatorname{Im}(X^{[*]}X)^n$, while $\tilde{X}x = Xx$ for $x \in \operatorname{Ker}(X^{[*]}X)^n$. It follows that

$$\operatorname{Ker}\tilde{X} = \operatorname{Im}(X^{[*]}X)^n \oplus \operatorname{Ker} X \tag{2.5}$$

with respect to decomposition (2.4). Indeed, the inclusion $\supseteq$ in (2.5) is obvious in view of the definition of $\tilde{X}$. For the opposite inclusion, let $x + y \in \operatorname{Ker}\tilde{X}$, where $x \in \operatorname{Im}(X^{[*]}X)^n$, $y \in \operatorname{Ker}(X^{[*]}X)^n$. Then clearly $y \in \operatorname{Ker}\tilde{X}$, and hence $y \in \operatorname{Ker} X$ by the definition of $\tilde{X}$. Note also the equality

$$\tilde{X}^{[*]}\tilde{X} = \begin{bmatrix} 0 & 0 \\ 0 & Z_0 \end{bmatrix}. \tag{2.6}$$

To verify (2.6), first note that because of (2.5), $\tilde{X}^{[*]}\tilde{X}$ has the form

$$\begin{bmatrix} 0 & ? \\ 0 & ? \end{bmatrix},$$

and the H-skew-Hamiltonian property of $\tilde{X}^{[*]}\tilde{X}$ implies that in fact

$$\tilde{X}^{[*]}\tilde{X} = \begin{bmatrix} 0 & 0 \\ 0 & ? \end{bmatrix},$$

the question marks denoting irrelevant parts of matrices. For every $x, y \in \operatorname{Ker}(X^{[*]}X)^n$ we have

$$\langle \tilde{X}^{[*]}\tilde{X}x, y\rangle = \langle H^{-1}\tilde{X}^T H\tilde{X}x, y\rangle,$$

which in view of the definition of $\tilde{X}$ is equal to

$$-\langle HXx, \tilde{X}H^{-1}y\rangle = -\langle HXx, XH^{-1}y\rangle = \langle X^{[*]}Xx, y\rangle.$$

Therefore, the lower right block of $\tilde{X}^{[*]}\tilde{X}$ must be Z_0, as claimed by equality (2.6).

Now assume that X has an H-Hamiltonian polar decomposition $X = UA$. Then A commutes with $X^{[*]}X = -A^2$, and since Z_1 and Z_0 have disjoint spectra it follows that A is block diagonal:

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_0 \end{bmatrix}.$$

Hence $Z_0 = -A_0^2$. Put

$$\tilde{A} = \begin{bmatrix} 0 & 0 \\ 0 & A_0 \end{bmatrix};$$

then it follows that $-\tilde{A}^2 = \tilde{X}^{[*]}\tilde{X}$, while $\operatorname{Ker}\tilde{A} = \operatorname{Ker}\tilde{X}$, since $\operatorname{Ker} A = \operatorname{Ker} X$. We conclude that if X admits an H-Hamiltonian polar decomposition, then so does $\tilde{X}$. Conversely, assume that $\tilde{X}$ has an H-Hamiltonian polar decomposition. Then

$$\tilde{X}^{[*]}\tilde{X} = \begin{bmatrix} 0 & 0 \\ 0 & Z_0 \end{bmatrix} = -B^2$$

for some H-Hamiltonian B such that $\operatorname{Ker} B = \operatorname{Ker}\tilde{X}$. The latter property ensures that with respect to the H-orthogonal decomposition (2.4), B has the form

$$B = \begin{bmatrix} 0 & ? \\ 0 & ? \end{bmatrix},$$

and the H-Hamiltonian property of B ensures that $B = O \oplus B_0$ for some B_0. The matrix B_0 is actually such that $H_0B_0 = -B_0^T H_0$ and $\{0\} \oplus \operatorname{Ker} B_0 = \operatorname{Ker} X$. Let B_1 be any H_1-Hamiltonian matrix such that $Z_1 = -B_1^2$, which exists by [1]. Put $A = B_1 \oplus B_0$; then A is H-Hamiltonian, $A^2 = -X^{[*]}X$, and $\operatorname{Ker} A = \operatorname{Ker} X$. Thus, if $\tilde{X}$ admits an H-Hamiltonian polar decomposition, then so does X. We have shown that X admits an H-Hamiltonian polar decomposition if and only if $\tilde{X}$ does, and $\tilde{X}^{[*]}\tilde{X}$ is nilpotent. Moreover, the conditions of Theorem 2.2 are satisfied for X if and only if they are satisfied for $\tilde{X}$. So we may indeed assume in the remainder of the proof that $X^{[*]}X$ is nilpotent.

To prove necessity, assume that $X = UA$ for an H-symplectic U and an H-Hamiltonian A. Bringing the pair (A, H) into canonical form (see Lemma 2.1(b)) and considering separately each orthogonal summand corresponding to the eigenvalue zero of A, we can assume that either

$$A = J_{2p}(0), \qquad H = \varepsilon F_{2p}$$

with respect to some basis $e_1, \ldots, e_{2p}$ or

$$A = \begin{bmatrix} J_{2p-1}(0) & 0 \\ 0 & -J_{2p-1}(0)^T \end{bmatrix}, \qquad H = \begin{bmatrix} 0 & I_{2p-1} \\ -I_{2p-1} & 0 \end{bmatrix}$$

with respect to some basis $e_1, \dots, e_{4p-2}$.

In the former case take the square of A. Consider the vectors

$$f_i = (-1)^{i-1} \frac{1}{\sqrt{2}}(e_{2i-1} + e_{2i})$$

together with the vectors

$$g_i = (-1)^{i-1} \frac{\varepsilon}{\sqrt{2}}(e_{2i-1} - e_{2i})$$

for $i = 1, \dots, p$. Observe that these are real vectors and that $-A^2$ with respect to the basis given by $\{f_1, \dots, f_p; g_p, \dots, g_1\}$ has the form $J_p(0) \oplus J_p(0)^T$. Moreover, the matrix H with respect to this basis has the form

$$\begin{bmatrix} 0 & I_{2p-1} \\ -I_{2p-1} & 0 \end{bmatrix}.$$

Finally, the kernel of A, and hence of X, is given by $\operatorname{Ker} A = \operatorname{span}\{e_1\} = \operatorname{span}\{f_1 + \varepsilon g_1\}$.

In the latter case, take as a basis

$$\begin{aligned} &e_1, -e_3, \dots, (-1)^{p-1} e_{2p-1}; && e_2, -e_4, \dots, (-1)^p e_{2p-2}; \\ &e_{2p}, -e_{2p+2}, \dots, (-1)^{p-1} e_{4p-2}; && e_{2p+1}, -e_{2p+3}, \dots, (-1)^p e_{4p-3}, \end{aligned}$$

in this order. With respect to this basis $(-A^2, H)$ has the form (2.2), and, moreover, $\operatorname{Ker} A = \operatorname{span}\{e_1, e_{4p-2}\}$, which proves necessity.

To prove sufficiency, we argue as in the proof of Theorem 4.4 in [2]. As observed above, we may assume that X is such that $X^{[*]}X$ is nilpotent. Let us assume that the pair $(X^{[*]}X, H)$ is in the form as described in this theorem with respect to some basis $\{e_{r,j}\}_{r=0,j=1}^{m,\ell_r}$, where $l_0 = 2k_0$, $l_r = 2k_r$ $(r = 1, \dots, m_1)$, and $l_r = 4k_r - 2$ $(r = m_1 + 1, \dots, m)$. We shall produce for each block (B_r, G_r) a matrix A_r such that $G_r A_r = -A_r^T G_r$ and $-A_r^2 = B_r$, and finally $\operatorname{Ker} A_r = \operatorname{Ker} X \cap \operatorname{span}\{e_{r,j}\}_{j=1}^{\ell_r}$, where $\operatorname{Ker} X$ is given by (2.3).

For the block (B_0, H_0) this is trivial: take $A_0 = B_0 = 0_{2k_0 \times 2k_0}$, the zero matrix of order k_0. Thus we have only to consider the blocks (B_r, H_r) with $r \geq 1$. First consider such a block of type (2.2). Let S be a matrix with the vectors

$$\begin{aligned} &e_{r,1}, e_{r,k_r+1}, -e_{r,2}, -e_{r,k_r+2}, \dots, (-1)^{k_r} e_{r,k_r-1}, (-1)^{k_r} e_{r,2k_r-1}, (-1)^{k_r-1} e_{r,k_r}; \\ &e_{r,2k_r}, e_{r,3k_r}, -e_{r,2k_r+1}, e_{r,3k_r+1}, \dots, (-1)^{k_r} e_{r,3k_r-2}, (-1)^{k_r} e_{r,4k_r-2}, (-1)^{k_r-1} e_{r,3k_r-1} \end{aligned} \tag{2.7}$$

as its columns, in this order. Then

$$S^{-1} B_r S = - \begin{bmatrix} J_{2k_r-1}(0) & 0 \\ 0 & -J_{2k_r-1}(0)^T \end{bmatrix}^2. \quad S^T G_r S = G_r.$$

Put

$$A_r = S \begin{bmatrix} J_{2k_r-1}(0) & 0 \\ 0 & -J_{2k_r-1}(0)^T \end{bmatrix} S^{-1}.$$

Then $-A_r^2 = B_r$ and

$$\operatorname{Ker} A_r = \operatorname{span}\{e_{r,1}, e_{r,4k_r-2}\} = \operatorname{Ker} X \cap \operatorname{span}\{e_{r,j}\}_{j=1}^{\ell_r}.$$

Next, consider a block B_r of type (2.1). Let S be the matrix with the following vectors as its columns:

$$\begin{aligned}&\frac{1}{\sqrt{2}}(e_{r,1}+\varepsilon_r e_{r,2k_r}),\ \frac{1}{\sqrt{2}}(e_{r,1}-\varepsilon_r e_{r,2k_r}),\ -\frac{1}{\sqrt{2}}(e_{r,2}+\varepsilon_r e_{r,2k_r-1}),\\ &\quad -\frac{1}{\sqrt{2}}(e_{r,2}-\varepsilon_r e_{r,2k_r-1}),\dots,(-1)^{k_r-1}\frac{1}{\sqrt{2}}(e_{r,k_r}+\varepsilon_r e_{r,k_r+1}),\\ &\qquad\qquad (-1)^{k_r-1}\frac{1}{\sqrt{2}}(e_{r,k_r}-\varepsilon_r e_{r,k_r+1}).\end{aligned} \tag{2.8}$$

It is assumed that the vectors appear in S in the same order. Then

$$S^{-1}B_rS=-J_{2k_r}(0)^2\ ,\ S^TG_rS=\varepsilon_rF_{2k_r}.$$

Let $A_r=SJ_{2k_r}(0)S^{-1}$. Then $A_r^2=-B_r$ and

$$\operatorname{Ker}A_r=\operatorname{span}\{e_{r,1}+\varepsilon_re_{r,2k_r}\}=\operatorname{Ker}X\cap\operatorname{span}\{e_{r,j}\}_{j=1}^{\ell_r},$$

as desired. □

The proof of Theorem 2.2 shows that the signs ε_r coincide with the signs ε_j in the canonical form of (A,H) corresponding to the blocks $(J_{2p_j}(0),\varepsilon_jF_{2p_j})$ as in Lemma 2.1(b); here A is the H-Hamiltonian matrix in an H-Hamiltonian polar decomposition $X=UA$ of X.

3. Comparison with existing polar decompositions. Hamiltonian polar decompositions can be compared with the polar decompositions studied in [2, 3, 4]. To do so, note that an H-Hamiltonian polar decomposition $X=UA$ gives rise to the iH-polar decomposition (in the terminology of [2]) $iX=U(iA)$. Here iA is iH-self-adjoint and U is H-symplectic, therefore also iH-unitary. Denoting by $^{[*]}$ the iH-adjoint operation we have $(iX)^{[*]}(iX)=X^{[*]}X$ (note that the definition of $^{[*]}$ given in section 1 coincides with iH-adjoint operation for real matrices:

$$H^{-1}X^TH=(iH)^{-1}X^*(iH)$$

for real X). Since by Theorem 1.2, $X^{[*]}X$ can be put in the form (1.2), it is clear that the partial multiplicities of $X^{[*]}X$ occur only in pairs and that the signs in the iH-sign characteristic of $(iX)^{[*]}(iX)=X^{[*]}X$ corresponding to each pair of multiplicities associated with a real eigenvalue of $(iX)^{[*]}(iX)=X^{[*]}X$ are opposite. Compare the canonical form of H-skew-Hamiltonian matrices; see Lemma 2.1(a).

In what follows we shall denote by Q_k the $k\times k$ matrix with zeros everywhere except on the south-west/north-east diagonal, where there are ones.

Combining the observation above with the necessary and sufficient conditions (obtained in [2]) for the existence of an iH-polar decomposition of iX, we obtain the following result.

THEOREM 3.1. *Let X be a real $2n\times 2n$ matrix and H a real skew-symmetric invertible $2n\times 2n$ matrix. Then there exists an iH-polar decomposition of iX if and only if the part of the canonical form of $(X^{[*]}X,iH)$ corresponding to the zero eigenvalue of $X^{[*]}X$ can be represented in the form*

$$(\operatorname{diag}(B_j)_{j=0}^m,\operatorname{diag}(G_j)_{j=0}^m),$$

where

(i) *B_0 is the zero matrix of order $2k_0$ and $G_0 = I_{k_0} \oplus -I_{k_0}$,*
(ii) *$m = m_1 + m_2$, and for each $j = 1, \dots, m_1$ we have*

$$B_j = \begin{bmatrix} J_{k_j}(0) & 0 \\ 0 & J_{k_j}(0) \end{bmatrix}, \qquad G_j = \begin{bmatrix} Q_{k_j} & 0 \\ 0 & -Q_{k_j} \end{bmatrix}, \tag{3.1}$$

while for $j = m_1 + 1, \dots, m_1 + m_2$ we have

$$B_j = \begin{bmatrix} J_{k_j}(0) & 0 & 0 & 0 \\ 0 & J_{k_j-1}(0) & 0 & 0 \\ 0 & 0 & J_{k_j}(0) & 0 \\ 0 & 0 & 0 & J_{k_j-1}(0) \end{bmatrix},$$

$$G_j = \begin{bmatrix} Q_{k_j} & 0 & 0 & 0 \\ 0 & Q_{k_j-1} & 0 & 0 \\ 0 & 0 & -Q_{k_j} & 0 \\ 0 & 0 & 0 & -Q_{k_j-1} \end{bmatrix}, \tag{3.2}$$

(iii) *and, denoting the corresponding basis in $\operatorname{Ker}(X^{[*]}X)^{2n} \subseteq \mathbb{C}^{2n}$ in which the form* (i), (3.1), (3.2) *is achieved, by $\{e_{r,j}\}_{r=0,j=1}^{m,\ell_r}$, where $\ell_0 = 2k_0$, $\ell_r = 2k_r$ for $r = 1, \dots, m_1$, and $\ell_r = 4k_r - 2$ for $r = m_1 + 1, \dots, m_2$ we have*

$$\begin{aligned} \operatorname{Ker} X = {} & \operatorname{span}\{e_{r,1} + e_{r,k_r+1} \mid r = 1, \dots, m_1\} \\ & + \operatorname{span}\{e_{r,1}, e_{r,2k_r} \mid r = m_1 + 1, \dots, m_2\} \\ & + \operatorname{span}\{e_{0,j}\}_{j=1}^{2k_0}. \end{aligned}$$

The clarifications made after Theorem 2.2 apply to Theorem 3.1 as well.

It is easily verified that the conditions of Theorem 3.1 are necessary for the existence of an H-Hamiltonian polar decomposition of X. Theorem 2.2 shows that they are also sufficient. We indicate (omitting many details) how one can derive Theorem 2.2 directly from Theorem 3.1. First write $(X^{[*]}X, iH)$ in canonical form as in Theorem 2.1 of [2] for $F = \mathbb{C}$ and take the complex conjugate of (2.2) and (2.3) of [2]. This leads to real Jordan blocks with opposite signs, and hence they can be arranged in pairs having opposite signs. Further, the condition in Theorem 4.4 of [2] on the negative eigenvalues of $X^{[*]}X$ to guarantee the existence of an (iH)-polar decomposition of X turns out to be superfluous. Next, writing $(X^{[*]}X, iH)$ in canonical form as in Theorem 2.1 of [2] for $F = \mathbb{C}$ with consecutive real Jordan blocks of equal size and opposite sign and letting $\sigma_1^j, \dots, \sigma_{k_j}^j$ $(j = 1, \dots, \alpha)$ stand for the first $k_1 + \cdots + k_\alpha$ columns of the complex matrix S that transforms $(X^{[*]}X, iH)$ to the canonical form, we obtain the part of the canonical form of $(X^{[*]}X, H)$ according to Lemma 2.1(a) if we let the columns of the new S be the vectors

$$\begin{aligned} &\rho_j^1 - \varepsilon_j \tau_{k_j}^j, \rho_j^2 - \varepsilon_j \tau_{k_j-1}^j, \dots, \rho_j^{k_j} - \varepsilon_j \tau_1^j, \\ &\rho_j^1 + \varepsilon_j \tau_{k_j}^j, \rho_j^2 + \varepsilon_j \tau_{k_j-1}^j, \dots, \rho_j^{k_j} + \varepsilon_j \tau_1^j, \end{aligned}$$

where $j = 1, \dots, \alpha$ and ρ_r^j and τ_r^j are the real and imaginary parts of σ_r^j, and we arrive at a direct derivation of Theorem 2.2 from Theorem 3.1. We have omitted in the above discussion consideration of Jordan blocks corresponding to nonreal eigenvalues of $X^{[*]}X$.

COROLLARY 3.2. *A real matrix X has an H-Hamiltonian polar decomposition with respect to a real invertible skew-symmetric matrix H if and only if iX has an iH-polar decomposition (over the field of complex numbers).*

REFERENCES

[1] H. FASSBENDER, D. S. MACKEY, N. MACKEY, AND H. XU, *Hamiltonian square roots of skew-Hamiltonian matrices*, Linear Algebra Appl., 287 (1999), pp. 125–159.

[2] YU. BOLSHAKOV, C. V. M. VAN DER MEE, A. C. M. RAN, B. REICHSTEIN, AND L. RODMAN, *Polar decompositions in finite dimensional indefinite scalar product spaces: General theory*, Linear Algebra Appl., 261 (1997), pp. 91–141.

[3] YU. BOLSHAKOV, C. V. M. VAN DER MEE, A. C. M. RAN, B. REICHSTEIN, AND L. RODMAN, *Extensions of isometries in finite-dimensional indefinite scalar product spaces and polar decompositions*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 752–774.

[4] YU. BOLSHAKOV, C. V. M. VAN DER MEE, A. C. M. RAN, B. REICHSTEIN, AND L. RODMAN, *Polar decompositions in finite dimensional indefinite scalar product spaces: Special cases and applications*, in Recent Developments in Operator Theory and Its Applications, Oper. Theory Adv. Appl. 87, I. Gohberg, P. Lancaster, and P. N. Shivakumar, eds., Birkhäuser, Basel, 1996, pp. 61–94. Erratum in Integral Equations Operator Theory, 27 (1997), pp. 497–501.

[5] D. Ž. DJOKOVIĆ, J. PATERA, P. WINTERNITZ, AND H. ZASSENHAUS, *Normal forms of elements of classical real and complex Lie and Jordan algebras*, J. Math. Phys., 24 (1983), pp. 1363–1374.

[6] C. V. M. VAN DER MEE, A. C. M. RAN, AND L. RODMAN, *Stability of self-adjoint square roots and polar decompositions in indefinite scalar product spaces*, Linear Algebra Appl., 302/303 (1999), pp. 77–104.

[7] R. C. THOMPSON, *Pencils of complex and real symmetric and skew matrices*, Linear Algebra Appl., 147 (1991), pp. 323–371.

SIAM J. MATRIX ANAL. APPL.
Vol. 22, No. 4, pp. 1274–1293

DATA FITTING PROBLEMS WITH BOUNDED UNCERTAINTIES IN THE DATA*

G. A. WATSON†

Abstract. An analysis of a class of data fitting problems, where the data uncertainties are subject to known bounds, is given in a very general setting. It is shown how such problems can be posed in a computationally convenient form, and the connection with other more conventional data fitting problems is examined. The problems have attracted interest so far in the special case when the underlying norm is the least squares norm. Here the special structure can be exploited to computational advantage, and we include some observations which contribute to algorithmic development for this particular case. We also consider some variants of the main problems and show how these too can be posed in a form which facilitates their numerical solution.

Key words. data fitting, bounded uncertainties, robustness, minimum norm problems, separable matrix norms

AMS subject classifications. 15A06, 15A60, 65F30, 65K05, 41A65

PII. S0895479899356596

1. Introduction.

Let $A \in R^{m\times n},\ b \in R^m$ arise from observed data, and for given $x \in R^n$, define

$$r = Ax - b.$$

Then a conventional fitting problem is to minimize $\|r\|$ over $x \in R^n$, where the norm is some norm on R^m. This involves an assumption that A is exact, and all the errors are in b, which may not be the case in many practical situations; the effect of errors in A as well as b has been recognized and studied for many years, mainly in the statistics literature. One way to take the more general case into account is to solve the problem

$$\text{minimize } \|E :\ d\| \text{ subject to } (A+E)x = b+d, \tag{1.1}$$

where the matrix norm is one on $(m\times(n+1))$ matrices. This problem, when the matrix norm is the Frobenius norm, was first analyzed by Golub and Van Loan [10], who used the term total least squares and developed an algorithm based on the singular value decomposition of $[A : b]$. Since then, the problem has attracted considerable attention: see, for example, [18], [19].

While the formulation (1.1) is often satisfactory, it can lead to a solution in which the perturbations E or d are quite large. However, it may be the case that, for example, A is known to be nearly exact, and the resulting correction to A may therefore be excessive. In particular, if bounds are known for the size of the perturbations, then it makes sense to incorporate these into the problem formulation, and this means that the equality constraints in (1.1) should be relaxed and satisfied only approximately. These observations have motivated new parameter estimation formulations where both A and b are subject to errors, but in addition, the quantities E and d are bounded, having known bounds. This idea gives rise to a number of different, but

*Received by the editors May 25, 1999; accepted for publication (in revised form) by L. El Ghaoui December 5, 2000; published electronically April 6, 2001.
http://www.siam.org/journals/simax/22-4/35659.html
† Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland (gawatson@maths.dundee.ac.uk).

closely related, problems and algorithms and analysis for problems of this type based on least squares norms are given, for example, in [1], [2], [3], [4], [5], [8], [9], [15], [17].

The general problem (1.1) is amenable to analysis and algorithmic development for a wide class of matrix norms, known as separable norms, a concept introduced by Osborne and Watson [13]. The main purpose of this paper is to show how problems with bounded uncertainties also can be considered in this more general setting. In particular, it is shown how such problems can be posed in a more computationally convenient form. As well as facilitating their numerical solution, this enables connections with conventional data fitting problems to be readily established. Motivation for extending these ideas beyond the familiar least squares setting is provided by the important role which other norms can play in more conventional data fitting contexts.

We continue this introductory section by defining separable norms and by introducing some other necessary notation and tools. We first introduce the concept of the dual norm. Let $\|.\|$ be a norm on R^m. Then for any $v \in R^m$, the *dual* norm is the norm on R^m defined by

$$\|v\|^* = \max_{\|r\| \leq 1} r^T v. \tag{1.2}$$

The relationship between a norm on R^m and its dual is symmetric, so that for any $r \in R^m$,

$$\|r\| = \max_{\|v\|^* \leq 1} r^T v. \tag{1.3}$$

DEFINITION 1.1. *A matrix norm on $m \times n$ matrices is said to be* separable *if given vectors $u \in R^m$ and $v \in R^n$, there are vector norms $\|.\|_A$ on R^m and $\|.\|_B$ on R^n such that*

$$\begin{aligned} \|uv^T\| &= \|u\|_A \|v\|_B^*, \\ \|uv^T\|^* &= \|u\|_A^* \|v\|_B. \end{aligned}$$

Most commonly occurring matrix norms (operator norms, orthogonally invariant norms, norms based on an l_p vector norm on the elements of the matrix treated as an extended vector in $R^{m \times n}$) are separable. A result which holds for separable norms and will be subsequently useful is that

$$\|Mv\|_A \leq \|M\| \|v\|_B; \tag{1.4}$$

see, for example, [13] or [20].

Another valuable tool is the subdifferential of a vector norm, which extends the idea of the derivative to the nondifferentiable case. A useful characterization of the subdifferential (for $\|.\|_A$) is as follows.

DEFINITION 1.2. *The* subdifferential *or set of subgradients of $\|r\|_A$ is the set*

$$\partial \|r\|_A = \{v \in R^m : \|r\|_A = r^T v, \ \|v\|_A^* \leq 1\}. \tag{1.5}$$

If the norm is differentiable at r, then the subdifferential is just the unique vector of partial derivatives of the norm with respect to the components of r.

The main emphasis of this paper is on problems which address the effects of worst case perturbations. This gives rise to problems of min-max type. In section 2, we consider problems where separate bounds on $\|E\|$ and $\|d\|_A$ are assumed known, and in section 3, we consider a similar problem except that a single bound on $\|E : d\|$ is given. In both cases, the matrix norm is assumed to be separable. In section 4, some variants of the original problems are considered, and finally, in section 5 we consider a related class of problems which are of min-min rather than min-max type.

2. Known bounds on $\|E\|$ and $\|d\|_A$. Suppose that the underlying problem is such that we know bounds on the uncertainties in A and b so that

$$\|E\| \leq \rho, \quad \|d\|_A \leq \rho_d,$$

where the matrix norm is a separable norm, as in Definition 1.1. Then instead of forcing the equality constraints of (1.1) to be satisfied, we wish to satisfy them approximately by minimizing the A-norm of the difference between the left- and right-hand sides, over all perturbations satisfying the bounds. This leads to the problem

$$\min_x \max_{\|E\| \leq \rho,\ \|d\|_A \leq \rho_d} \|(A+E)x - (b+d)\|_A. \tag{2.1}$$

Therefore x minimizes the worst case residual, and this can be interpreted as permitting a more robust solution to be obtained to the underlying data fitting problem: for an explanation of the significance of the term robustness in this context, in the least squares case, see, for example, [9], where a minimizing x is referred to as a robust least squares solution. Another interpretation of the problem being solved is that it guarantees that the effect of the uncertainties in the data will never be overestimated, beyond the assumptions made by knowledge of the bounds.

We now show that (2.1) can be restated in a much simpler form as an unconstrained problem in x alone.

THEOREM 2.1. *For any x, the maximum in* (2.1) *is attained when*

$$E = \rho u w^T, \quad w \in \partial \|x\|_B,$$

$$d = -\rho_d u,$$

where $u = \frac{r}{\|r\|_A}$ if $r \neq 0$, otherwise u is arbitrary but $\|u\|_A = 1$. The maximum value is

$$\|r\|_A + \rho \|x\|_B + \rho_d.$$

Proof. We have for any E, d such that $\|E\| \leq \rho$, $\|d\|_A \leq \rho_d$,

$$\begin{aligned} \|(A+E)x - (b+d)\|_A &= \|r + Ex - d\|_A \\ &\leq \|r\|_A + \rho \|x\|_B + \rho_d. \end{aligned}$$

Now let E and d be as in the statement of the theorem. Then

$$\|E\| = \rho, \quad \|d\|_A = \rho_d,$$

and further

$$\begin{aligned} \|(A+E)x - (b+d)\|_A &= \|r + \rho \|x\|_B u + \rho_d u\|_A \\ &= \|r\|_A + \rho \|x\|_B + \rho_d. \end{aligned}$$

The result follows. ∎

An immediate consequence of this result is that the problem (2.1) is solved by minimizing with respect to x

$$\|Ax - b\|_A + \rho \|x\|_B, \tag{2.2}$$

and it is therefore appropriate to analyze this problem. In particular, we give conditions for x to be a solution and also conditions for that solution to be $x = 0$. Both

results are a consequence of standard convex analysis, as is found, for example, in [14].

THEOREM 2.2. *The function* (2.2) *is minimized at* x *if and only if there exists* $v \in \partial\|Ax-b\|_A$, $w \in \partial\|x\|_B$ *such that*

$$A^T v \;+\; \rho w \;=\; 0. \tag{2.3}$$

THEOREM 2.3. *Let there exist* $v \in \partial\|b\|_A$ *so that*

$$\|A^T v\|_B^* \le \rho.$$

Then $x = 0$ *minimizes* (2.2).

Proof. For $x = 0$ to give a minimum we must have $v \in \partial\|b\|_A$ so that (2.3) is satisfied with $\|w\|_B^* \le 1$. The result follows. ∎

2.1. Connections with least norm problems. We will next establish some connections between solutions to (2.2) and solutions to traditional minimum norm data fitting problems. In [9], coincidence of solutions in the least squares case is said to mean that the usual least squares solution may be considered to be robust.

Consider the least norm problem

$$\text{minimize } \|Ax-b\|_A. \tag{2.4}$$

Then x is a solution if and only if there exists $v \in \partial\|Ax-b\|_A$ such that

$$A^T v \;=\; 0.$$

If $x = 0$ solves (2.4), then clearly it also solves (2.2). (Note that we can take $w = 0$ in (2.3).) Otherwise, if $\|.\|_A$ is smooth, solutions to (2.2) and to (2.4) can coincide only if $b \in \text{range}(A)$ (since otherwise v is unique). In this case, let $x = A^+ b$ be any solution to $Ax = b$ and let $y = (A^T)^+ c$ denote the minimum A-norm solution to $A^T y = c$. Then if x minimizes (2.2), (2.3) is satisfied with $w \in \partial\|x\|_B$ and $\|v\|_A^* \le 1$, otherwise v is unrestricted. Because

$$v \;=\; -\rho(A^T)^+ w,$$

it follows that we must have

$$\rho \le \frac{1}{\|(A^T)^+ w\|_A^*}.$$

In other words, $A^+ b \neq 0$ is also a solution to (2.2) only if $b \in \text{range}(A)$ and

$$\rho \le \max_{w \in \partial\|A^+ b\|_B} \frac{1}{\|(A^T)^+ w\|_A^*}.$$

For example, if both norms are least squares norms, then this condition is

$$\rho \le \frac{\|A^+ b\|_2}{\|(AA^T)^+ b\|_2}.$$

Note that if $x = A^+ b$ is the minimum B-norm solution to $Ax = b$, then it immediately solves (2.2), and so there must exist w such that this inequality is satisfied independently of ρ.

The case when $\|.\|_A$ is nonsmooth is more complicated.

EXAMPLE 2.1. *Let* $A = [1,1]^T$, $b = (1,2)^T$, $\|.\|_A = \|.\|_1$, $\|.\|_B = \|.\|_\infty$. *(This corresponds to the separable norm being the sum of moduli of the components.) Then* (2.4) *is solved by any* x, $1 \le x \le 2$. *Further* $x = 1$ *is a solution to* (2.2) *provided that* $0 < \rho < 2$.

To summarize, we can augment Theorem 2.3 by the following, which can be interpreted as a generalization of a result of [9].

THEOREM 2.4. *If* $b \in \text{range}(A)$ *and* $x = A^+b$ *is any solution to* $Ax = b$, *then provided that*

$$\rho \le \max_{w \in \partial \|A^+b\|_B} \frac{1}{\|(A^T)^+ w\|_A^*},$$

A^+b *also minimizes* (2.2).

We can also prove the following, which connects Theorems 2.3 and 2.4.

THEOREM 2.5. *Let* $b \in \text{range}(A)$, *and* $x = A^+b$ *satisfy* $Ax = b$. *Then*

$$\max_{w \in \partial \|A^+b\|_B} \frac{1}{\|(A^T)^+ w\|_A^*} \le \min_{v \in \partial \|b\|_A} \|A^T v\|_B^*.$$

Proof. Let $v \in \partial\|b\|_A$, $w \in \partial\|A^+b\|_B$ be otherwise arbitrary. It follows by definition of A^+ and $(A^T)^+$ that

$$AA^+b \;=\; b,$$

$$A^T(A^T)^+w \;=\; w.$$

Thus

$$b^T(A^+)^T A^T v = b^T v = \|b\|_A, \tag{2.5}$$

and

$$b^T(A^+)^T A^T (A^T)^+ w = b^T (A^+)^T w = \|A^+b\|_B. \tag{2.6}$$

Now

$$\begin{aligned} \|A^T v\|_B^* \|(A^T)^+ w\|_A^* &= \max_{\|c\|_B \le 1} c^T A^T v . \max_{\|d\|_A \le 1} d^T (A^T)^+ w \\ &\ge \frac{(A^+b)^T A^T v}{\|A^+b\|_B} . \frac{b^T (A^+)^T A^T (A^T)^+ w}{\|AA^+b\|_A} \\ &= 1, \end{aligned}$$

using (2.5) and (2.6). The result follows. ∎

A consequence of the above results is that if $b \in \text{range}(A)$ and

$$\max_{w \in \partial \|A^+b\|_B} \frac{1}{\|(A^T)^+ w\|_A^*} \min_{v \in \partial \|b\|_A} \|A^T v\|_B^*,$$

then any point in the convex hull of $\{0,\ A^+b\}$ is a solution.

2.2. Methods of solution. From a practical point of view, it is obviously of importance to efficiently solve (2.1) (or, equivalently, (2.2)) in appropriate cases.

Let

$$f(x) \;=\; \|Ax-b\|_A \;+\; \rho\|x\|_B.$$

Most commonly occurring norms are either smooth (typified by l_p norms, $1<p<\infty$) or polyhedral (typified by the l_1 and l_∞ norms). If the norms in the definition of f are smooth, then derivative methods are natural ones to use. A reasonable assumption in most practical situations is that $b \notin \text{range}(A)$, so that $x=0$ would then give the only derivative discontinuity. If $x=0$ is not a solution, then f is differentiable in a neighborhood of the solution, and once inside that neighborhood, derivative methods can be implemented in a straightforward manner. Theorem 2.3 tells us when $x=0$ is a solution; the following theorem gives a way of identifying a descent direction at that point in the event that it is not. It applies to arbitrary norms.

THEOREM 2.6. *Assume that*

$$\|A^Tv\|_B^* > \rho \quad \text{for all } v \in \partial\|b\|_A,$$

and let $\hat{v} \in \partial\|b\|_A$, $\hat{g} \in \partial\|A^T\hat{v}\|_B^*$ *be such that*

$$\hat{g}^TA^T\hat{v} \;=\; \min\{g^TA^Tv:\; v \in \partial\|b\|_A\}.$$

Let $d=-\hat{g}$. *Then* d *is a descent direction for* f *at* $x=0$.

Proof. Let the stated conditions be satisfied, and let d be defined as in the statement of the theorem. By theorem 2.3, $x=0$ is not a solution. For d to be a descent direction at $x=0$, the directional derivative of f at $x=0$ in the direction d must be negative, that is,

$$\max_{v\in\partial\|b\|_A,\|w\|_B^*\le 1} d^TA^Tv+\rho d^Tw<0.$$

Let $v \in \partial\|b\|_A, \|w\|_B^* \le 1$ be otherwise arbitrary. Then

$$\begin{aligned} d^TA^Tv+\rho d^Tw &= -\hat{g}^TA^Tv+\rho d^Tw \\ &\le -\hat{g}^TA^T\hat{v} \;+\; \rho, \quad \text{since } \|d\|_B \le 1 \\ &= -\|A^T\hat{v}\|_B^* \;+\; \rho \\ &< 0. \end{aligned}$$

The result follows. ∎

If $\|.\|_A$ is smooth, then $\hat{v}$ is unique, and the construction of d using this result is straightforward. If the norm on E is the norm given by

$$\|E\| = \left(\sum_{i,j}|E_{ij}|^p\right)^{1/p}$$

or

$$\|E\| = \max_{\|x\|_q=1} \|Ex\|_p,$$

then f becomes

$$f(x) = \|Ax-b\|_p \;+\; \rho\|x\|_q,$$

where $1/p + 1/q = 1$. In fact, the minimization of f for any p, q satisfying $1 < p, q < \infty$ can be readily achieved by derivative methods, using Theorem 2.6 to get started. Indeed, it is normally the case that second derivatives exist and can easily be calculated so that Newton's method (damped if necessary) can be used following a step based on Theorem 2.6. The Hessian matrix of f is positive definite (because f is convex), so that the Newton direction is a descent direction away from a minimum. Some numerical results are given in [21].

For polyhedral norms (typified by l_1 and l_∞ norms), the convex objective function (2.2) is a piecewise linear function. Therefore, it may be posed as a linear programming problem, and solved by appropriate methods.

Arguably, the most interesting case from a practical point of view is the special case when both $\|.\|_A$ and $\|.\|_B$ are the least squares norm, so that

$$f(x) \;=\; \|Ax - b\|_2 \;+\; \rho\|x\|_2. \tag{2.7}$$

This case has particular features which greatly facilitate computation, and Chandrasekaran et al. [1], [3] exploit these in a numerical method. In contrast to the problems considered above, which involve a minimization problem in R^n, special features of the l_2 case can be exploited so that the problem reduces to one in R. When (2.7) is differentiable, (2.3) becomes

$$\|r\|_2^{-1} A^T r \;+\; \rho\|x\|_2^{-1} x \;=\; 0$$

or

$$(A^T A + \alpha I)x \;=\; A^T b,$$

where

$$\alpha \;=\; \frac{\rho\|r\|_2}{\|x\|_2}.$$

Let the singular value decomposition of A be

$$A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T,$$

where $U \in R^{m\times m}$ and $V \in R^{n\times n}$ are orthogonal and $\Sigma = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\}$ is the matrix of singular values in descending order of magnitude. Let

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \;=\; U^T b,$$

where $b_1 \in R^n$ and $b_2 \in R^{m-n}$. It will be assumed in what follows that A has rank n, and further that $x = 0$ is not a solution (which means, in particular, that $b_1 \neq 0$) and $b \notin \text{range}(A)$ (which means that $b_2 \neq 0$). From Theorem 2.3, we require that

$$\rho < \frac{\|A^T b\|_2}{\|b\|_2} \;=\; \frac{\|\Sigma b_1\|_2}{\|b\|_2}. \tag{2.8}$$

Then it is shown in [3] that α satisfies the equation

$$\alpha \;=\; g(\alpha), \tag{2.9}$$

where

$$g(\alpha) = \frac{\rho\sqrt{\|b_2\|_2^2 + \alpha^2\|(\Sigma^2 + \alpha I)^{-1}b_1\|_2^2}}{\|\Sigma(\Sigma^2 + \alpha I)^{-1}b_1\|_2}.$$

This can be rearranged as

$$G(\alpha) = 0, \tag{2.10}$$

where

$$G(\alpha) = b_1^T(\Sigma^2 - \rho^2 I)(\Sigma^2 + \alpha I)^{-2}b_1 - \frac{\rho^2\|b_2\|_2^2}{\alpha^2}.$$

It is also shown in [3] that (2.8) is both necessary and sufficient for (2.10) to have exactly one positive root α^*. In addition, $G'(\alpha^*) > 0$. Different methods can be used for finding α^* in this case. One possibility which is suggested by (2.9) is the simple iteration process

$$\alpha_k = g(\alpha_{k-1}), \quad k = 1, \ldots, \tag{2.11}$$

and it is of interest to investigate whether or not this is likely to be useful. It turns out that this method is always locally convergent, as the following result shows.

THEOREM 2.7. *Let*

$$\rho \ < \ \frac{\|\Sigma b_1\|_2}{\|b\|_2}. \tag{2.12}$$

Then (2.10) *has exactly one positive root* α^* *and* (2.11) *is locally convergent to* α^*.

Proof. Let ρ satisfy (2.12). Then (2.10) has a unique positive root α^*. Differentiating $G(\alpha)$ gives

$$G'(\alpha) = -2b_1^T(\Sigma^2 - \rho^2 I)(\Sigma^2 + \alpha I)^{-3}b_1 + 2\frac{\rho^2\|b_2\|_2^2}{\alpha^3},$$

and so

$$\alpha^* G'(\alpha^*) = 2b_1^T\Sigma^2(\Sigma^2 - \rho^2 I)(\Sigma^2 + \alpha^* I)^{-3}b_1, \tag{2.13}$$

using $G(\alpha^*) = 0$. Now $g(\alpha)$ and $G(\alpha)$ are related by

$$G(\alpha) \ = \ \left(1 - \left(\frac{g(\alpha)}{\alpha}\right)^2\right)\|\Sigma(\Sigma^2 + \alpha I)^{-1}b_1\|_2^2,$$

and so

$$G'(\alpha^*) = \frac{2(1 - g'(\alpha^*))}{\alpha^*}\|\Sigma(\Sigma^2 + \alpha^* I)^{-1}b_1\|_2^2, \tag{2.14}$$

using $g(\alpha^*) = \alpha^*$. Thus

$$g'(\alpha^*) = 1 - \frac{\alpha^* G'(\alpha^*)}{2\|\Sigma(\Sigma^2 + \alpha^* I)^{-1}b_1\|_2^2}. \tag{2.15}$$

TABLE 1
Simple iteration: stack loss data.

	$\rho = 0.0001$	$\rho = 0.01$	$\rho = 0.05$	$\rho = 0.5$	$\rho = 1.0$	$\rho = 2.0$
n	α_n	α_n	α_n	α_n	α_n	α_n
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.000033	0.003348	0.016738	0.167379	0.334759	0.669517
3	0.000033	0.003500	0.020728	0.620634	2.169373	7.781285
4		0.003507	0.021731	1.826895	8.554021	22.998133
5		0.003508	0.021985	3.935491	11.606395	25.307605
6			0.022050	5.192151	11.922507	25.549150
7			0.022066	5.466537	11.949293	25.573871
8			0.022070	5.509046	11.951529	25.576395
9			0.022071	5.515242	11.951715	25.576653
10				5.516138	11.951731	25.576679
11				5.516267	11.951732	25.576682
12				5.516285		
13				5.516288		

Substituting from (2.13) gives

$$\begin{aligned} g'(\alpha^*) &= 1 - \frac{b_1^T\Sigma^2(\Sigma^2 - \rho^2 I)(\Sigma^2 + \alpha^* I)^{-3}b_1}{b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-2}b_1} \\ &= 1 - \frac{b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-2}b_1 - (\alpha^* + \rho^2)b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-3}b_1}{b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-2}b_1} \\ &= \frac{(\alpha^* + \rho^2)b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-3}b_1}{b_1^T\Sigma^2(\Sigma^2 + \alpha^* I)^{-2}b_1} \\ &> 0. \end{aligned}$$

It follows using (2.15) and $G'(\alpha^*) > 0$ that

$$0 < g'(\alpha^*) < 1,$$

and the result is proved. □

Indeed, simple iteration seems to be remarkably effective, and in problems tried, it converged in a satisfactory way from $\alpha = 0$ and other less obvious starting points. For example, for the stack loss data set of Daniel and Wood [6] ($m = 21,\ n = 4$), performance for different values of ρ is shown in Table 1, where the iteration is terminated when the new value of α differs from the previous one by less than 10^{-6}.

Another example is given by using the Iowa wheat data from Draper and Smith [7] ($m = 33,\ n = 9$). The performance of simple iteration in this case is shown in Table 2.

Although simple iteration is in some ways suggested by the above formulation, of course higher order methods can readily be implemented, such as the secant method or Newton's method. Actual performance will depend largely on factors such as the nature and size of the problem and the relative goodness of starting points.

3. A known bound on $\|E : d\|$. Suppose now that the underlying problem is such that we know upper bounds on the uncertainties in A and b, in the form

$$\|E :\ d\| \le \rho,$$

where ρ and the (separable) matrix norm are given. Consider the problem of determining

$$\min_x \max_{\|E:\ d\| \le \rho} \|(A + E)x - (b + d)\|_A, \tag{3.1}$$

TABLE 2
Simple iteration: Iowa wheat data.

	$\rho = 0.0001$	$\rho = 0.01$	$\rho = 0.05$	$\rho = 0.5$	$\rho = 1.0$	$\rho = 2.0$
n	α_n	α_n	α_n	α_n	α_n	α_n
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.001084	0.108417	0.542085	5.420855	10.841710	21.683420
3		0.108602	0.546694	5.872607	12.606823	28.410850
4			0.546733	5.909318	12.878252	30.238701
5				5.912295	12.919596	30.716282
6				5.912536	12.925885	30.839779
7				5.912556	12.926841	30.871628
8				5.912557	12.926986	30.879837
9					12.927008	30.881951
10					12.927012	30.882496
11						30.882637
12						30.882673
13						30.882682
14						30.882685

where the A-norm on R^m is defined by the particular choice of separable norm (or vice versa). This problem and variants have been considered, for example, by El Ghaoui and Lebret [8], [9], where the matrix norm is the Frobenius norm, so that both the A- and B-norms are least squares norms. Arguing as in Theorem 2.1 gives the following result.

THEOREM 3.1. *For any x, the maximum in* (3.1) *is attained when*

$$[E : d] = \rho u w^T, \quad w \in \partial \|[x^T : -1]^T\|_B, \tag{3.2}$$

where $u = \frac{r}{\|r\|_A}$ if $r \neq 0$, otherwise any vector with $\|u\|_A = 1$. The maximum value is

$$\|r\|_A + \rho \|[x^T : -1]^T\|_B. \tag{3.3}$$

The problem (3.1) is therefore equivalent to the problem of minimizing with respect to x

$$\|Ax - b\|_A + \rho \|[x^T : -1]^T\|_B. \tag{3.4}$$

Standard convex analysis then gives the following result.

THEOREM 3.2. *The function* (3.4) *is minimized at x if and only if there exists $v \in \partial \|Ax - b\|_A$, $u \in \partial \|[x^T : -1]^T\|_B$ such that*

$$A^T v + \rho u_1 = 0, \tag{3.5}$$

where u_1 denotes the first n components of u.

3.1. Connection with least norm problems. As before, it is of interest to establish connections with the corresponding least norm problems. If $x = 0$ solves (2.4), then it will also minimize (3.4) for monotonic norms $\|.\|_B$. ($\|.\|_B$ is a monotonic norm on R^{n+1} if $\|c\|_B \leq \|d\|_B$ whenever $|c_i| \leq |d_i|$, $i = 1, \ldots, n$.) If $x = 0$ does not solve (2.4), then just as before when $\|.\|_A$ is smooth, solutions to this problem and (3.1) cannot coincide unless $b \in \text{range}(A)$. In that case, as in section 2.1 let $x = A^+ b$ denote a solution to $Ax = b$, and let $y = (A^T)^+ c$ denote the minimum A-norm solution to $A^T y = c$. For a solution to (3.4), there must exist v, $\|v\|_A^* \leq 1$ (otherwise unrestricted) so that

$$A^T v + \rho u_1 = 0,$$

where u_1 consists of the first n components of $u \in \partial\|[(A^+b)^T, -1]^T\|_B$. Therefore,

$$v \ + \ \rho(A^T)^+u_1 \ = \ 0,$$

and so

$$\rho \leq \frac{1}{\|(A^T)^+u_1\|_A^*}.$$

In other words the solutions will coincide if $b \in \text{range}(A)$ and

$$\rho \leq \max_{u \in \partial\|[(A^+b)^T,-1]^T\|_B} \frac{1}{\|(A^T)^+u_1\|_A^*}. \tag{3.6}$$

Note that if $\|.\|_B$ is smooth, then u is unique. For example, when both norms are least squares norms, this gives

$$\rho \leq \frac{\sqrt{\|A^+b\|_2^2 + 1}}{\|(AA^T)^+b\|_2},$$

as given in [9]. The situation when $\|.\|_A$ is not smooth is, of course, once again more complicated. Consider again Example 2.1 where $\rho > 0$ is arbitrary. Recall that (2.4) is solved by any $x, \ 1 \leq x \leq 2$: the unique solution to the problem of minimizing (3.4) is $x = 1$.

3.2. Connection with total approximation problems. The nature of the bound in (3.1) means that there is a connection to be made with the total approximation problem (1.1). It is known [13], [20] that a minimum value of (1.1) coincides with the minimum of the problem

$$\text{minimize } \|[A : \ b]z\|_A \quad \text{subject to } \|z\|_B = 1, \tag{3.7}$$

the smallest β-generalized singular value of the matrix $[A : \ b]$. In particular, if the vector norms are least squares norms, then this is just the smallest singular value of $[A : \ b]$. An $x = x_T$ at a minimum of (1.1) is obtained from a $z = z_T$ at a minimum of (3.7) by scaling so that

$$z_T^T \ = \ \alpha[x_T^T, -1],$$

whenever $(z_T)_{n+1} \neq 0$. (No z_T with $(z_T)_{n+1} \neq 0$ corresponds to nonexistence of a solution to (1.1).) It is known also that a minimizing pair $E, \ d$ is given by

$$[E_T : \ d_T] \ = \ -[A : \ b]z_T w_T^T, \ \text{ where } w_T \in \partial\|z_T\|_B.$$

Define

$$A_T = A + E_T, \ \ b_T = b + d_T,$$

and consider the problem

$$\min_x \max_{\|E: \ d\| \leq \rho} \|(A_T + E)x - (b_T + d)\|_A, \tag{3.8}$$

or equivalently,

$$\min_x \ \|A_T x - b_T\|_A \ + \ \rho\|[x^T, -1]^T\|_B.$$

Then $b_T \in \text{range}(A_T)$, with $A_T x_T = b_T$, and so if $\|.\|_A$ is smooth, x_T is a solution to this problem provided that

$$\rho \leq \max_{u \in \partial\|[x_T^T, -1]^T\|_B} \frac{1}{\|(A_T^T)^+ u_1\|_A^*}, \tag{3.9}$$

as a consequence of the previous analysis. For example, when both norms are least squares norms, this gives (see also [9])

$$\rho \leq \frac{\sqrt{\|x_T\|_2^2 + 1}}{\|(A_T^T)^+ x_T\|_2}.$$

For the least squares case, El Ghaoui and Lebret [8] suggest using robust methods in conjunction with total approximation to identify an appropriate value of ρ. The idea is first to solve the total approximation problem. Then (3.8) is constructed from the total approximation solution and solved with ρ set to ρ_T, the minimum value in (3.7), that is,

$$\rho_T \ = \ \frac{\|Ax_T - b\|_A}{\|[x_T^T, \ -1]^T\|_B}.$$

Of course if ρ_T does not exceed the right-hand side of (3.9), there is nothing to solve.

3.3. Methods of solution. For the special case of (3.4) when the norms $\|.\|_A$ and $\|.\|_B$ are (possibly different) l_p norms, we have

$$f(x) \ = \ \|Ax - b\|_p \ + \ \rho\|[x^T : -1]^T\|_q. \tag{3.10}$$

When $1 < p, \ q < \infty$, then derivative methods may again be used. Let us again make the (reasonable) assumption that there is no x which makes $\|Ax - b\|_p = 0$, so that $\|Ax - b\|_p$ is differentiable for all x. Then in contrast to the earlier problem, since the second term cannot be identically zero, f is differentiable for all x. We can easily compute first and second derivatives of f, and so Newton's method, for example, can be implemented. A line search in the direction of the Newton step will always guarantee descent, because f is convex, so eventually we must be able to take full steps and get a second order convergence rate. Some numerical results are given in [21]. For polyhedral norms occurring in (3.4), linear programming techniques may be used.

Now consider the special case when $p = q = 2$. An analysis similar to that given in section 2.2 can be given in this case, leading to a similar numerical method. This particular problem is considered by El Ghaoui and Lebret [8], [9]. The main emphasis of those papers is on structured perturbations, which is a harder problem, and an exact solution to that problem is obtained. For the present case, the method suggested is similar to that given for the problem of section 2 by Chandrasekaran et al. in [1], [3].

Let A have singular value decomposition as before and have full rank. Assume also that $b \notin \text{range}(A)$. Then optimality conditions are

$$\|r\|_2^{-1} A^T r \ + \ \rho\|[x^T : -1]^T\|_2^{-1} x \ = \ 0$$

or

$$(A^TA + \alpha I)x \;=\; A^Tb,$$

where

$$\alpha \;=\; \frac{\rho\|r\|_2}{\|[x^T : -1]^T\|_2}.$$

It can be shown as before that α satisfies the equation

$$\alpha \;=\; h(\alpha),$$

where

$$h(\alpha) \;=\; \frac{\rho\sqrt{\|b_2\|_2^2 + \alpha^2\|(\Sigma^2 + \alpha I)^{-1}b_1\|_2^2}}{\sqrt{\|\Sigma(\Sigma^2 + \alpha I)^{-1}b_1\|_2^2 + 1}}.$$

This can as before be rearranged as

$$H(\alpha) \;=\; 0,$$

where

$$H(\alpha) = G(\alpha) + 1$$

with $G(\alpha)$ defined as in (2.10). It is easily seen that $H(\alpha)$ has at least one positive root for any $\rho > 0$. As in [3], it may be shown that $H(\alpha)$ in fact has exactly one positive root, $\hat{\alpha}$, with

$$H(\hat{\alpha}) > 0.$$

Note that here there is no restriction on ρ except that it should be positive. Consider the simple iteration process

$$\alpha_k \;=\; h(\alpha_{k-1}), \quad k = 1, \ldots. \tag{3.11}$$

THEOREM 3.3. *The iteration scheme* (3.11) *is locally convergent to* $\hat{\alpha}$.

Proof. We can first show that

$$\hat{\alpha}H'(\hat{\alpha}) \;=\; 2b_1^T\Sigma^2(\Sigma^2 - \rho^2)(\Sigma^2 + \hat{\alpha})^{-3}b_1 + 2. \tag{3.12}$$

We can then show that $h(\alpha)$ and $H(\alpha)$ are related by

$$H(\alpha) \;=\; \left(1 - \left(\frac{h(\alpha)}{\alpha}\right)^2\right)(1 + C(\alpha)^2),$$

where

$$C(\alpha) \;=\; \|\Sigma(\Sigma^2 + \alpha I)^{-1}b_1\|_2.$$

Thus

$$H'(\hat{\alpha}) \;=\; \frac{2(1 - h'(\hat{\alpha}))}{\hat{\alpha}}(1 + C(\hat{\alpha})^2),$$

using $h(\hat{\alpha}) = \hat{\alpha}$. Thus

$$h'(\hat{\alpha}) \;=\; 1 - \frac{\hat{\alpha}H'(\hat{\alpha})}{2(1 + C(\hat{\alpha})^2)}. \tag{3.13}$$

Substituting from (3.12) gives

$$\begin{aligned} h'(\hat{\alpha}) &= 1 - \frac{b_1^T \Sigma^2(\Sigma^2 - \rho^2)(\Sigma^2 + \hat{\alpha})^{-3} b_1 \;+\; 1}{1 + C(\hat{\alpha})^2} \\ &= \frac{b_1^T(\Sigma^2(\Sigma^2 + \hat{\alpha}I)^{-2} - \Sigma^2(\Sigma^2 - \rho^2)(\Sigma^2 + \hat{\alpha})^{-3})b_1}{1 + C(\hat{\alpha})^2} \\ &= (\rho^2 + \hat{\alpha})\frac{b_1^T \Sigma^2(\Sigma^2 + \hat{\alpha}I)^{-3} b_1}{1 + C(\hat{\alpha})^2}. \end{aligned}$$

It follows using (3.13) and $H'(\hat{\alpha}) > 0$ that

$$0 < h'(\hat{\alpha}) < 1,$$

and the result is proved. □

The performance of simple iteration in this case is, of course, similar to the same method applied in the previous situation. Other methods like the secant method, or Newton's method, are more complicated but can give potentially better performance.

4. Some modifications. There are different ways in which additional information may be incorporated into the problems of the last two sections, resulting in appropriate modifications of these problems. For example, some components of A or b may be exact, in which case the corresponding components of E or d will be zero. The bounds may take different forms and may be on submatrices of E rather than E itself. Also the perturbation matrices may have known structure, which we want to preserve. Examples of all these possibilities are considered in this section.

4.1. Exact columns and rows. Some problems are such that some of the columns and possibly rows of A are known to be exact (see, for example, [3]). A treatment can be given for both the problems of sections 2 and 3, and we will demonstrate only for those of section 2; the appropriate requirements for the problems of section 3 are obvious. We begin by considering the case when certain columns only of A are known to be exact. In that case (following suitable reordering of columns if necessary) the general problem is to minimize

$$\min_x \max_{\|E\| \le \rho,\ \|d\|_A \le \rho_d} \|(A_1 :\ A_2 + E)x - (b + d)\|_A, \tag{4.1}$$

where $A_1 \in R^{m \times (n-t)}$, $A_2 \in R^{m \times t}$, and the (separable) matrix norm is one defined on $m \times t$ matrices. We can partition x as $x^T = (x_1^T,\ x_2^T)^T$, with $x_2 \in R^t$. Then arguing as in Theorem 2.1, we have the following.

THEOREM 4.1. *For any x, the maximum in* (4.1) *is attained when*

$$E \;=\; \rho u w^T, \quad w \in \partial\|x_2\|_B,$$

$$d \;=\; -\rho_d u,$$

where $u = \frac{r}{\|r\|_A}$ if $r \neq 0$; otherwise u is arbitrary, but $\|u\|_A = 1$. The maximum value is

$$\|r\|_A \;+\; \rho\|x_2\|_B \;+\; \rho_d.$$

Therefore, the problem is solved by minimizing with respect to x

$$\|Ax - b\|_A \ + \ \rho\|x_2\|_B. \tag{4.2}$$

Now consider the case when some columns and rows of A are exact. This corresponds to the requirement to perturb only a submatrix of A. Assume this to be the lower right-hand $s \times t$ submatrix. An appropriate problem is then to minimize

$$\min_x \max_{\|E\| \le \rho, \ \|d\|_A \le \rho_d} \left\| \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 + E \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - (b + d) \right\|_A, \tag{4.3}$$

where A_2 and A_4 have t columns, A_3 and A_4 have s rows, and the matrix norm is a separable norm on $s \times t$ matrices. Unfortunately, the separable norm is defined in terms of two vector norms $\|.\|_A$ on R^s and $\|.\|_B$ on R^t, and $\|.\|_A$ as used in (4.3) is on R^m. We get around this potential conflict by assuming that $\|.\|_A$ is defined for any length of vector; we will also assume that the introduction of additional zero components does not change the value of the norm.

The attainment of the maximum in (4.3) is not quite so straightforward as before. However, we can prove the following result.

THEOREM 4.2. *Let $r = Ax - b$, let r_1 denote the first $m - s$ components of r, and let r_2 denote the last s components. Let x solve the problem*

$$\text{minimize} \ \ \|r_2\|_A \ + \ \rho\|x_2\|_B \ \ \text{subject to} \ r_1 = 0. \tag{4.4}$$

Then x solves (4.3).

Proof. Arguing as in previous results, an upper bound for the maximum (subject to the constraints) in (4.3) is

$$\|r\|_A \ + \ \rho\|x_2\|_B + \rho_d.$$

Now define the set

$$X \ = \ \{x \in R^n : \ r_1 = 0\}.$$

For any $x \in X$, define

$$E \ = \ \rho u_2 w^T, \ \ w \in \partial\|x_2\|_B,$$

$$d \ = \ -\rho_d u,$$

where $u \in R^m$ has first $(m - s)$ components zero, and last s components forming the vector u_2 with $u_2 = \frac{r_2}{\|r\|_A}$ if $r \neq 0$; otherwise $u_2 \in R^s$ is arbitrary except that $\|u_2\|_A = 1$.

Then $\|E\| = \rho$, $\|d\| = \rho_d$, and

$$\begin{aligned}
\left\| \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 + E \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - (b + d) \right\|_A &= \left\| r \ + \ \begin{bmatrix} 0 \\ E \end{bmatrix} x_2 - d \right\|_A \\
&= \left\| r \ + \ \begin{bmatrix} 0 \\ \rho u_2 w^T \end{bmatrix} x_2 + \rho_d u \right\|_A \\
&= \left\| r \ + \ \rho \begin{bmatrix} 0 \\ u_2 \end{bmatrix} \right\| x\|_B + \rho_d u\|_A \\
&= \|r \ + \ \rho u\|x\|_B + \rho_d u\|_A \\
&= \|r\|_A \ + \ \rho\|x_2\|_B \ + \ \rho_d.
\end{aligned}$$

The result is proved. □

Of course the set X may be empty. In that case, while the problem (4.3) is still well defined, it is not clear that a matrix E and a vector d can be defined such that the maximum in the problem is attained. That being the case, there is no obvious equivalent simpler problem.

4.2. Bounded columns of *E*. Suppose that the columns of E are individually bounded so that

$$\|Ee_i\|_A \leq \rho_i, \quad i = 1, \ldots, n,$$

where e_i is the ith unit vector, and consider the problem of finding

$$\min_x \max_{\|Ee_i\|_A \leq \rho_i, i=1,\ldots,n} \|(A+E)x - b\|_A. \tag{4.5}$$

As for Theorem 2.1, we can prove the following result.

THEOREM 4.3. *For any x, the maximum in* (4.5) *is attained when*

$$Ee_i = \rho_i \theta_i u, \quad i = 1, \ldots, n,$$

where $\theta_i = \operatorname{sign}(x_i)$, and where $u = \frac{r}{\|r\|_A}$ if $r \neq 0$; otherwise u is arbitrary but $\|u\|_A = 1$. The maximum value is

$$\|Ax - b\|_A \; + \; \sum_{i=1}^{n} \rho_i |x_i|.$$

Even in the least squares case, this objective function is not normally differentiable, being a combination of a least squares norm and a weighted l_1 norm. It can be reposed as a smooth constrained optimization problem, and solved by standard techniques.

4.3. Structured problems. In some applications, the perturbation matrices have known structure, as in the following problem considered by El Ghaoui and Lebret [9]. Given $A_0, \ldots, A_p \in R^{m \times n}$, $b_0, \ldots, b_p \in R^m$, determine

$$\min_{x \in R^n} \max_{\|\delta\| \leq \rho} \left\| \left(A_0 + \sum_{i=1}^{p} \delta_i A_i \right) x - \left(b_0 + \sum_{i=1}^{p} \delta_i b_i \right) \right\|_A, \tag{4.6}$$

where $\|.\|_A$ is a given norm on R^m and $\|.\|$ is a given norm on R^p. Define for any $x \in R^n$,

$$r_i \; = \; A_i x - b_i, \quad i = 0, \ldots, p,$$

$$M \; = \; [r_1, \ldots, r_p] \; \in R^{m \times p},$$

$$F = M^T M, \quad g = M^T r_0, \quad \text{and} \quad h = r_0^T r_0.$$

Consider the maximum in (4.6), which will be attained at the solution to the problem

$$\text{maximize } \|r_0 + M\delta\|_A \quad \text{subject to } \|\delta\| = \rho,$$

assuming that δ maximizing $\|r_0 + M\delta\|_A$ exceeds ρ in norm. Because the functions involved are convex, necessary conditions for a solution can readily be given: these are that there exists $v \in \partial\|r_0 + M\delta\|_A$, $w \in \partial\|\delta\|$, $\lambda \in R$ such that

$$M^T v - \lambda w \; = \; 0, \quad \|\delta\| = \rho.$$

Using these conditions, it is easily seen that

$$\|r_0 + M\delta\|_A \;=\; v^T r_0 \;+\; \lambda\rho.$$

Therefore, an equivalent (in a sense dual) problem is

$$\text{maximize } v^T r_0 \;+\; \lambda\rho \;\text{ subject to}$$

$$M^T v - \lambda w \;=\; 0, \quad \|\delta\| = \rho,$$

$$v \in \partial\|r_0 + M\delta\|_A,$$

$$w \in \partial\|\delta\|.$$

Consider the special case when both norms are least squares norms. Then

$$M^T v - \lambda w \;=\; \frac{M^T(r_0 + M\delta)}{\|r_0 + M\delta\|_2}$$

and so the necessary conditions can be written

$$(\tau I - F)\delta \;=\; g, \quad \|\delta\|_2 = \rho,$$

where $\tau = \lambda\|r_0 + M\delta\|_2/\rho$. Further,

$$\begin{aligned}\|r_0 + M\delta\|_2 &= \lambda\rho \;+\; v^T r_0 \\ &= \frac{\rho^2\tau}{\|r_0 + M\delta\|_2} \;+\; \frac{(r_0 + M\delta)^T r_0}{\|r_0 + M\delta\|_2}.\end{aligned}$$

Thus

$$\begin{aligned}\|r_0 + M\delta\|_2^2 &= \rho^2\tau + h + \delta^T g \\ &= \rho^2\tau + h + g^T(\tau I - F)^{-1} g,\end{aligned}$$

provided that $\tau I - F$ is nonsingular. A way of solving this problem based on those results is given by El Ghaoui and Lebret [9]. They also consider the problem when $\|.\|$ is the Chebyshev norm. Extending the ideas to more general norms, however, does not look straightforward.

5. A min-min problem. The problems (2.1) and (3.1) are examples of min-max problems: minimization is carried out with respect to x over all allowed perturbations in the data. This is justified if the emphasis is on robustness. However, from other considerations it may be sufficient to minimize with respect to x while simultaneously minimizing with respect to the perturbations. This gives rise to a min-min problem, as considered (least squares case) in [2], [3], [5]. In this final section, we will briefly consider this problem. Again there are two versions, consistent with those treated in sections 2 and 3. To illustrate the ideas involved, we will consider finding

$$\min_x \min_{\|E:d\| \le \rho} \|(A + E)x - (b + d)\|_A. \tag{5.1}$$

In contrast to the min-max case, here we are seeking to find a solution x which gives the smallest possible error over allowable perturbations.

Again the problem can be replaced by an equivalent unconstrained optimization problem.

THEOREM 5.1. *Let ρ be small enough that*

$$\rho\|[x^T : \ -1]^T\|_B \le \|r\|_A \quad \text{for all } x \in R^n. \tag{5.2}$$

Then (5.1) *is equivalent to the problem of minimizing with respect to x*

$$\|Ax - b\|_A \ - \ \rho\|[x^T : \ -1]^T\|_B. \tag{5.3}$$

Proof. Let (5.2) be satisfied and let x be arbitrary. Let $\|E : d\| \le \rho$ with E, d otherwise arbitrary. Then

$$\begin{aligned}\|(A+E)x - (b+d)\|_A &= \|r + (Ex - d)\|_A \\ &\ge \|r\|_A - \|E : d\|\|[x^T : -1]^T\|_B \\ &\ge \|r\|_A - \rho\|[x^T : \ -1]^T\|_B.\end{aligned}$$

Now fix

$$[E : d] \ = \ -\rho u w^T, \quad w \in \partial\|[x^T : -1]^T\|_B,$$

where $u = \frac{r}{\|r\|_A}$, $r \ne 0$; otherwise u is arbitrary with $\|u\|_A = 1$. Then $\|E : d\| = \rho$, and further

$$\begin{aligned}\|(A+E)x - (b+d)\| &= \|r - \rho u\|[x^T : 1]^T\|_B\|_A \\ &= \|r\|_A - \rho\|[x^T : \ -1]^T\|_B,\end{aligned}$$

using (5.2). The result follows. ∎

There are two important differences between (5.3) and (3.4): first, the relationship leading to (5.3) requires a condition on ρ, and second, the resulting problem is not a convex problem. The nonconvexity of (5.3) is interpreted in [2] as being equivalent to using an "indefinite" metric, in the spirit of recent work on robust estimation and filtering: see, for example, [11], [12], [16].

The condition (5.2) is satisfied if

$$\rho \le \frac{\|[A : b]z\|_A}{\|z\|_B},$$

that is, if ρ does not exceed ρ_T (see section 3.2). If $\rho = \rho_T$, then

$$\min_x\{\|Ax - b\|_A - \rho_T\|[x^T : \ -1]^T\|_B\} \ = \ 0,$$

attained at $x = x_T$. Indeed if ρ is set to any *local* minimum of (3.7), with value ρ_T, then the corresponding point x_T generated from the local minimizer z_T is a stationary point of (5.1), as the following argument shows.

Necessary conditions for x to solve (3.7) are that there exist $v \in \partial\|[A : \ b]z_T\|_A$, $w \in \partial\|z_T\|_B$, and a Lagrange multiplier λ such that

$$[A : \ b]^T v - \lambda w \ = \ 0.$$

Multiplying through by z_T^T shows that $\lambda = \rho_T$, and so

$$[A : \ b]^T v - \rho_T w \ = \ 0.$$

Now the relationship $z_T = \alpha[x_T^T, \ -1]$ implies that $\text{sign}(\alpha)v \in \partial\|Ax_T - b\|_A$ and $\text{sign}(\alpha)w \in \partial\|[x^T, -1]^T\|_B$. In other words, there exist $v \in \partial\|Ax_T - b\|_A$, $w \in \partial\|[x^T, -1]^T\|_B$ such that

$$A^T v - \rho w_1 \ = \ 0,$$

where w_1 denotes the first n components of w. It follows from standard convex analysis that x_T is a stationary point of the problem of minimizing

$$\|Ax - b\|_A - \rho_T\|[x^T : \ -1]^T\|_B.$$

A similar treatment can be given if (5.1) is replaced by the related problem of finding

$$\min_x \min_{\|E\| \le \rho} \ \|(A + E)x - b\|_A.$$

Provided that ρ is small enough that

$$\rho\|x\|_B \le \|Ax - b\|_A \ \text{ for all } x \in R^n,$$

then this is equivalent to the problem of finding

$$\min_x\{\|Ax - b\|_A \ - \ \rho\|x\|_B\}.$$

An algorithm is given in [2] for solving the least squares case of this problem. It has similarities to the algorithms given before, involving the solution of a nonlinear equation for α and a linear system for x. Indeed it is clear that many of the ideas which apply to min-max problems carry over to problems of the present type. However, we do not consider that further here.

6. Conclusions. We have given an analysis in a very general setting of a range of data fitting problems, which have attracted interest so far in the special case when least squares norms are involved. While this case is likely to be most useful in practice, consideration of other possibilities can be motivated by the valuable role that other norms play in a general data fitting context. The main thrust of the analysis has been to show how the original problems may be posed in a simpler form. This permits the numerical treatment of a wide range of problems involving other norms, for example, l_p norms. We have also included some observations which contribute to algorithmic development for the important least squares case.

Acknowledgment. I am grateful to the referees for helpful comments which have improved the presentation.

REFERENCES

[1] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *Efficient algorithms for least squares type problems with bounded uncertainties*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 171–180.

[2] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *Parameter estimation in the presence of bounded modeling errors*, IEEE Signal Process. Lett., 4 (1997), pp. 195–197.
[3] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *Parameter estimation in the presence of bounded data uncertainties*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 235–252.
[4] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *An efficient algorithm for a bounded errors-in-variables model*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 839–859.
[5] S. Chandrasekaran, M. Gu, A. H. Sayed, and K. E. Schubert, *The degenerate bounded errors-in-variables model*, SIAM J. Matrix Anal. Appl., to appear.
[6] C. Daniel and F. S. Wood, *Fitting Equations to Data*, Wiley, New York, 1971.
[7] N. R. Draper and H. Smith, *Applied Regression Analysis*, Wiley, New York, 1966.
[8] L. El Ghaoui and H. Lebret, *Robust solutions to least squares problems with uncertain data*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 161–170.
[9] L. El Ghaoui and H. Lebret, *Robust solutions to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064.
[10] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
[11] B. Hassibi, A. H. Sayed, and T. Kailath, *Recursive linear estimation in Krein spaces–Part I: Theory*, IEEE Trans. Automat. Control, AC-41 (1996), pp. 18–33.
[12] P. Khargonekar, and K. M. Nagpal, *Filtering and smoothing in an H^∞ setting*, IEEE Trans. Automat. Control, AC-36 (1991), pp. 151–166.
[13] M. R. Osborne and G. A. Watson, *An analysis of the total approximation problem in separable norms, and an algorithm for the total l_1 problem*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 410–424.
[14] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.
[15] A. H. Sayed and S. Chandrasekaran, *Estimation in the presence of multiple sources of uncertainties with applications*, in Proceedings of the Asilomar Conference, Pacific Grove, CA, 1998, pp. 1811–1815.
[16] A. H. Sayed, B. Hassibi, and T. Kailath, *Inertia conditions for the minimization of quadratic forms in indefinite metric spaces*, in Operator Theory: Advances and Applications, I. Gohberg, P. Lancaster, and P. N. Shivakumar, eds., Birkhauser, Basel, 1996, pp. 309–347.
[17] A. H. Sayed, V. H. Nascimento, and S. Chandrasekaran, *Estimation and control in the presence of bounded data uncertainties*, Linear Algebra Appl., 284 (1998), pp. 259–306.
[18] S. Van Huffel, ed., *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, in Proceedings of the 2nd International Workshop on Total Least Squares and Errors-in-Variables Modeling, Leuven, 1996, SIAM, Philadelphia, 1997.
[19] S. Van Huffel and J. Vandevalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.
[20] G. A. Watson, *Choice of norms for data fitting and function approximation*, Acta Numer. 7, 1998, pp. 337–377.
[21] G. A. Watson, *Solving data fitting problems in l_p norms with bounded uncertainties in the data*, in Proceedings of the Dundee Conference, Numerical Analysis 1999, D. F. Griffiths and G. A. Watson, eds., Chapman and Hall/CRC Res. Notes Math. 420, Boca Raton, FL, 2000, pp. 249–265.