SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 1–14
© 2001 Society for Industrial and Applied Mathematics

VALUES OF MINORS OF AN INFINITE FAMILY OF *D*-OPTIMAL DESIGNS AND THEIR APPLICATION TO THE GROWTH PROBLEM*

C. KOUKOUVINOS†, M. MITROULI‡, AND JENNIFER SEBERRY§

Abstract. We obtain explicit formulae for the values of the $2v-j$ minors, $j=0,1,2$, of D-optimal designs of order $2v=x^2+y^2$, v odd, where the design is constructed using two circulant or type 1 incidence matrices of $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ supplementary difference sets (sds). This allows us to obtain information on the growth problem for families of matrices with moderate growth. Some of our theoretical formulae imply growth greater than $2(2s^2+2s+1)$ but experimentation has not yet supported this result. An open problem remains to establish whether the $(1,-1)$ completely pivoted (CP) incidence matrices of $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ sds which yield D-optimal designs can have growth greater than $2v$.

Key words. D-optimal designs, symmetric balanced incomplete block designs, supplementary difference sets, Gaussian elimination, growth, complete pivoting

AMS subject classifications. 05B20, 15A15, 65F05, 65G05

PII. S0895479800373644

1. Introduction.

A D-optimal design of order n is an $n\times n$ matrix with entries ± 1 having maximum determinant. In the present paper we evaluate the $2v-j$, $j=0,1,2$, minors for $(1,-1)$ incidence matrices of certain symmetric balanced incomplete block designs (SBIBDs) which yield D-optimal designs. For the purpose of this paper we will define a SBIBD(v,k,λ) to be a $v\times v$ matrix, B, with entries 0 or 1, which has exactly k entries $+1$ and $v-k$ entries 0 in each row and column and for which the inner product of any distinct pairs of rows and columns is λ. The $(1,-1)$ incidence matrix of B is obtained by letting $A=2B-J$, where J is the $v\times v$ matrix with entries all $+1$. We write I for the identity matrix of order v.

Then we have

$$BB^T=(k-\lambda)I+\lambda J \tag{1.1}$$

and

$$AA^T=4(k-\lambda)I+(v-4(k-\lambda))J. \tag{1.2}$$

It can be easily shown that

$$\det B=(k-\lambda)^{\frac{v-1}{2}}\sqrt{k+(v-1)\lambda}$$

and since $\lambda(v-1)=k^2-k$

$$\det A=2^{v-1}(k-\lambda)^{\frac{v-1}{2}}|v-2k|. \tag{1.3}$$

*Received by the editors June 13, 2000; accepted for publication (in revised form) by R. Brualdi December 26, 2000; published electronically April 18, 2001.
http://www.siam.org/journals/simax/23-1/37364.html

†Department of Mathematics, National Technical University of Athens, Zografou 15773, Athens, Greece (ckoukouv@math.ntua.gr).

‡Department of Mathematics, University of Athens, Panepistemiopolis 15784, Athens, Greece (mmitroul@cc.uoa.gr).

§School of Information Technology and Computer Science, University of Wollongong, Wollongong, NSW, 2522, Australia (jennie@uow.edu.au).

In this paper we also study the application of the computed values of the minors to the growth problem for SBIBD $(2s^2+2s+1, s^2, \frac{1}{2}s(s-1))$, which is Brouwer's design and which yields a D-optimal design.

Let $A=[a_{ij}] \in \mathcal{R}^{n\times n}$. We reduce A to upper triangular form by using Gaussian elimination with complete pivoting (GECP) [19]. Let $A^{(k)}=[a_{ij}^{(k)}]$ denote the matrix obtained after the first k pivoting operations, so $A^{(n-1)}$ is the final upper triangular matrix. A diagonal entry of that final matrix will be called a pivot. Matrices with the property that no exchanges are actually needed during GECP are called completely pivoted (CP). Let $g(n,A)=\max_{i,j,k}|a_{ij}^{(k)}|/|a_{11}^{(0)}|$ denote the growth associated with GECP on A and $g(n)=\sup\{\, g(n,A)/A \in \mathcal{R}^{n\times n}\,\}$. The problem of determining $g(n)$ for various values of n is called the growth problem.

The determination of $g(n)$ remains one of the major unsolved problems in numerical analysis. See [9] for a detailed description of the problem. One of the curious frustrations of the growth problem is that it is quite difficult to construct any examples of $n\times n$ matrices A other than Hadamard matrices for which $g(n,A)$ is even close to n. The equality $g(n,A)=n$ has been proved for a certain class of $n\times n$ Hadamard matrices [4]. It has also been observed that weighing matrices of order n can give $g(n,A)=n-1$ [12]. In [11] the pivot structure of $(1,-1)$ incidence matrices of SBIBD(v,k,λ) is studied. In the present paper we get values for the pivots of $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ supplementary difference sets (sds), and D-optimal designs made from them. Calculations have given moderate values of growth for D-optimal designs. An open problem concerning the possibility of finding $(1,-1)$ $2v\times 2v$ CP D-optimal designs having growth greater than $2v$ is posed.

Notation. Write A for a matrix of order n whose initial pivots are derived from matrices with CP structure. Write $A(j)$ for the absolute value of the determinant of the $j\times j$ principal submatrix in the upper left-hand corner of the matrix A and $A[j]$ for the absolute value of the determinant of the $(n-j)\times(n-j)$ principal submatrix in the bottom right-hand corner of the matrix A. Throughout this paper when we have used i pivots we then find all possible values of the $A(n-i)$ minors. Hence, if any minor is CP, it must have one of these values. The magnitude of the pivots appearing after the application of GE (Gaussian elimination) operations on a CP matrix W is given by

$$p_j = W(j)/W(j-1), \quad j=1,2,\ldots,n, \quad W(0)=1. \tag{1.4}$$

In particular, for a CP SBIBD(v,k,λ), A,

$$p_v = A(v)/A(v-1), \quad p_{v-1}=A(v-1)/A(v-2). \tag{1.5}$$

We use the notation M_j to denote the $j\times j$ minor of A.

For completeness we give the determinant simplification theorem in the appendix as we use it extensively in this paper.

2. *D*-optimal designs of order $2v \equiv 2 \pmod 4$ from SBIBDs. Let d_n denote the maximum determinant of all $n\times n$ matrices with elements ± 1. It follows from Hadamard's inequality that $d_n \le n^{\frac{n}{2}}$ and it is easily shown that equality can only hold if $n=1$ or 2 or if $n\equiv 0 \,(\mathrm{mod}\, 4)$. We shall here be concerned with the case $n\equiv 2\,(\mathrm{mod}\,4)$, $n\neq 2$. Ehlich [6] showed that

$$d_n \le (2n-2)(n-2)^{\frac{n}{2}-1}$$

and equality can hold only if $2n-2=x^2+y^2$, where x and y are integers.

Recently two infinite series of $n\times n$ ($n\equiv 2(\mathrm{mod}\,4)$) matrices with elements ± 1 and maximum determinant were discovered. The first series (Koukouvinos–Kounias–Seberry or KKS) [10], exists for $n=2(q^2+q+1)$ where q is a prime power. The second series (Whiteman–Brouwer or WB) [18], exists for $n=2(2q^2+2q+1)$ where q is an odd prime power.

For the purpose of this paper we will define two sds $2-\{v;k_1,k_2;\lambda\}$ to be two circulant (or type 1) $v\times v$ matrices B_1 and B_2, with entries 0 or 1, which have exactly k_i entries $+1$ and $v-k_i$ entries 0, $i=1,2$, respectively, in each row and column and for which the inner product of any pair of rows is λ. The $(1,-1)$ incidence matrices of B_i, are obtained by letting $A_i=2B_i-J,\quad i=1,2$.

The family of SBIBD$(2s^2+2s+1,s^2,\frac{1}{2}s(s-1))$, for s is an odd prime power, has been found by Brouwer [3]. For $s=2$, the SBIBD$(13,4,1)$ comes from the projective plane. The case for $s=4$, the SBIBD$(41,16,6)$ is given by Bridges, Hall, and Hayden [2] and independently by van Trung [16]. The case for $s=6$, the SBIBD$(85,36,15)$ is given as unknown by van Trung [17, p. 84] and Beth, Jungnickel, and Lenz [1, p. 625]. However, for $s=6$, Gysin [8] gives the first $2-\{85;36,36;30\}$ sds. For $s=8$, Djokovic [5] gives the first $2-\{145;64,64;56\}$ sds. Georgiou and Koukouvinos [7] give further results for $s=6$ and $s=8$. Examples of $2-\{25;9,9;6\}$ sds, $2-\{41;16,16;12\}$ sds, $2-\{61;25,25;20\}$ sds, $2-\{113;49,49;42\}$ sds, and $2-\{181;81,81;72\}$ sds corresponding to the cases $s=3,4,5,7,9$, respectively, are given in [13]. In addition, for $s=3$, i.e., $2-\{25;9,9;6\}$, there is a type 1 solution in the group $Z_5\times Z_5$.

These $2-\{2s^2+2s+1;s^2,s^2;s(s-1)\}$ sds have $(1,-1)$ incidence matrices which satisfy

$$A_1A_1^T+A_2A_2^T=(4s^2+4s)I+2J.$$

Let R and S be permutation matrices of order v. Then A given by

$$\begin{bmatrix} P & P \\ RPS & -RPS \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} A_1 & A_2 \\ A_2^T & -A_1^T \end{bmatrix}$$

are D-optimal designs of order $2v\equiv 2(\mathrm{mod}4)$ of the WB family. We can say the WB family is constructed from $2-\{2s^2+2s+1;s^2,s^2;s(s-1)\}$ sds. Note $A_1=A_2$ for the WB family.

We can write

$$AA^T=(2v-2)I_{2v}+2I_2\times J_v.$$

It is easy to use the determinant simplification theorem to see that

$$\det A=2^v(2v-1)(v-1)^{v-1}.$$

Since A has been constructed using the $2-\{2s^2+2s+1;s^2,s^2;s(s-1)\}$ sds,

$$\det A=M_{2v}=2^{(2s+1)^2}(2s+1)^2(s(s+1))^{2s(s+1)}. \tag{2.1}$$

2.1. Minors of size $(2v-1)$. To find the $(2v-1)\times(2v-1)$ minors we remove the first row and column of A to get B. We denote by $\Delta(h,i,j,k,m)$ the following matrix of order $2v$.

$$\Delta(h,i,j,k,m) =$$

$$\left[\begin{array}{cccc|cccc|cccc|cccc}
\overbrace{m}^{h} & 1 & \cdots & 1 & \overbrace{3}^{i} & 3 & \cdots & 3 & \overbrace{-}^{j} & - & \cdots & - & \overbrace{1}^{k} & 1 & \cdots & 1\\
1 & m & \cdots & 1 & 3 & 3 & \cdots & 3 & - & - & \cdots & - & 1 & 1 & \cdots & 1\\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\
1 & 1 & \cdots & m & 3 & 3 & \cdots & 3 & - & - & \cdots & - & 1 & 1 & \cdots & 1\\
\hline
3 & 3 & \cdots & 3 & m & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & - & - & \cdots & -\\
3 & 3 & \cdots & 3 & 1 & m & \cdots & 1 & 1 & 1 & \cdots & 1 & - & - & \cdots & -\\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\
3 & 3 & \cdots & 3 & 1 & 1 & \cdots & m & 1 & 1 & \cdots & 1 & - & - & \cdots & -\\
\hline
- & - & \cdots & - & 1 & 1 & \cdots & 1 & m & 1 & \cdots & 1 & 3 & 3 & \cdots & 3\\
- & - & \cdots & - & 1 & 1 & \cdots & 1 & 1 & m & \cdots & 1 & 3 & 3 & \cdots & 3\\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\
- & - & \cdots & - & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & m & 3 & 3 & \cdots & 3\\
\hline
1 & 1 & \cdots & 1 & - & - & \cdots & - & 3 & 3 & \cdots & 3 & m & 1 & \cdots & 1\\
1 & 1 & \cdots & 1 & - & - & \cdots & - & 3 & 3 & \cdots & 3 & 1 & m & \cdots & 1\\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\
1 & 1 & \cdots & 1 & - & - & \cdots & - & 3 & 3 & \cdots & 3 & 1 & 1 & \cdots & m
\end{array}\right].$$

$m = 2v = h + i + j + k$. Then by the determinant simplification theorem

$$\det\Delta(h,i,j,k,m) = (m-1)^{m-4}\begin{vmatrix} m-1+h & 3h & -h & h\\ 3i & m-1+i & i & -i\\ -j & j & m-1+j & 3j\\ k & -k & 3k & m-1+k\end{vmatrix}$$

and $\det\,\Delta(h,i,j,k,m) = (m-1)^{(m-4)}[(m-1)^4 + (m-1)^3(i+j+h+k) - 8(m-1)^2(jk+ih) - 16(m-1)(jk(i+h)+ih(j+k))]$.

Now $\det\,BB^T$ is obtained from $\Delta(h,i,j,k,m)$ by removing a row and the corresponding column. Thus $\det\,BB^T$ is $\det\,\Delta(h-1,i,j,k,m-1)$ or $\det\,\Delta(h,i-1,j,k,m-1)$ or $\det\,\Delta(h,i,j-1,k,m-1)$ or $\det\,\Delta(h,i,j,k-1,m-1)$.

LEMMA 2.1. *The $(2v-1)\times(2v-1)$ minors of the D-optimal designs of the WB series are*

$$M_{2v-1} = 2^{4s(s+1)}\,(2s+1)\,s^{2s^2+2s-1}\,(s+1)^{2s^2+2s},\ \ 2^{4s(s+1)}\,(2s+1)\,s^{2s^2+2s}\,(s+1)^{2s^2+2s-1},$$

where s is an odd prime power, $s = 2, 4, 6,$ or 8.

Proof. Here we use the $(1,-1)$ incidence matrices of the $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ sds. By the reasoning above, with $v = 2s^2+2s+1$, $h = j = s^2$, $i = k = s^2+2s+1$, $m = 4s^2+4s+2$, substituted into $\det\,\Delta(h-1,i,j,k,m-1) = \det\,\Delta(h,i,j-1,k,m-1)$, we obtain the result.

Specifically, the determinant is the square root of the determinant given by

$$(4s^2+4s)^{4s^2+4s-3}\begin{vmatrix} 5s^2+4s-1 & 3s^2-3 & -s^2+1 & s^2-1\\ 3s^2+6s+3 & 5s^2+6s+1 & s^2+2s+1 & -s^2-2s-1\\ -s^2 & s^2 & 5s^2+4s & 3s^2\\ s^2+2s+1 & -s^2-2s-1 & 3s^2+6s+3 & 5s^2+6s+1\end{vmatrix}$$

$$= 2^{2(4s^2+4s)}(s^2+s)^{4s^2+4s-2}(s+1)^2(2s+1)^2 = \left[2^{4s^2+4s}s^{2s^2+2s-1}(s+1)^{2s^2+2s}(2s+1)\right]^2,$$

$$\begin{aligned}\det\Delta(h,i-1,j,k,m-1) &= \det\Delta(h,i,j,k-1,m-1)\\ &= \left[2^{4s^2+4s}s^{2s^2+2s}(s+1)^{2s^2+2s-1}(2s+1)\right]^2\end{aligned}$$

which gives the second result. □

2.2. Minors of size $(2v-2)$. As the partitioned matrix A of the D-optimal design is composed from 2-$\{v;k_1,k_2;\lambda\}$ sds, these are in fact 2-$\{2s^2+2s+1;s^2,s^2;s^2-s\}$ sds. We will use $k=k_1=k_2$ for all our calculations. Using the formula for the inner product of the rows of the $(1,-1)$ incidence matrix formed from these sds we see that the inner product is $2v-4(k_1+k_2-\lambda)=2$.

We now return to A with two rows and columns removed to find the generic matrix. We have not included this in expanded form except for one case but moved straight to the determinant after it has been simplified using the determinant simplification theorem of the matrix D given by

$$\begin{bmatrix}
2v-2 & 2u_2 & 2u_3 & 4u_4 & -2u_5 & 0 & 0 & 2u_8\\
2u_1 & 2v-2 & 4u_3 & 2u_4 & 0 & -2u_6 & 2u_7 & 0\\
2u_1 & 4u_2 & 2v-2 & 2u_4 & 0 & 2u_6 & -2u_7 & 0\\
4u_1 & 2u_2 & 2u_3 & 2v-2 & 2u_5 & 0 & 0 & -2u_8\\
-2u_1 & 0 & 0 & 2u_4 & 2v-2 & 2u_6 & 2u_7 & 4u_8\\
0 & -2u_2 & 2u_3 & 0 & 2u_5 & 2v-2 & 4u_7 & 2u_8\\
0 & 2u_2 & -2u_3 & 0 & 2u_5 & 4u_6 & 2v-2 & 2u_8\\
2u_1 & 0 & 0 & -2u_4 & 4u_5 & 2u_6 & 2u_7 & 2v-2
\end{bmatrix}.$$

This gives the determinant of A with two rows and columns removed, as $(4s^2+4s)^{2s^2+2s-4}\sqrt{\det D}$.

To calculate the minors of size $(2v-2)$ we distinguish two major cases: Case I, where the two rows removed to form the minor came from the same part of the D-optimal design that is they have inner product 2; Case II, where the two rows removed to form the minor came from different parts of the D-optimal design, that is, they have inner product zero. This leads to the following four subcases.

Case Ia. $\left[\begin{smallmatrix} x & y\\ x & \bar{y}\end{smallmatrix}\right]$, where the (1,1) and (2,1) elements have the same sign, the (1,2) element and the (2,2) element have opposite signs, and the inner product of row one and two with each other is 2. The inner product of the first two rows with the next $(v-i)$ rows is $+i$ and the inner product of row one and two with the $v+3-i$ to $2v$th rows is $2-i$, where $i=2$ or 0.

Case Ib. $\left[\begin{smallmatrix} x & y\\ x & \bar{y}\end{smallmatrix}\right]$, where the (1,1) and (2,1) elements have the same sign, the (1,2) element and the (2,2) element have opposite signs, and the inner product of rows one and two with each other is $+2$. The inner product of rows one and two with next $v-i$ rows is $+i$ and the inner product of the first two rows with rows $v+3-i$ to $2v$ is $2-i$, where $i=2$ or 0.

Case IIa. $\left[\begin{smallmatrix} x & y\\ x & \bar{y}\end{smallmatrix}\right]$, where the (1,1) element and the (2,1) element have the same signs, the (1,2) element and the (2,2) element have different signs, and the inner product of rows one and two with each other is zero. Rows 3 to $v+1$ have inner product $+2$ with row one and zero with row two. Rows $v+2$ to $2v$ have

inner product zero with row one and +2 with row two.

Case IIb. $\left[\begin{smallmatrix} x & y \\ x & y \end{smallmatrix}\right]$, where the (1,1) element and the (2,1) element have the same signs, the (1,2) element and the (2,2) element also have the same sign, and the inner product of row one and two with each other is zero. Rows 3 to $v+1$ have inner product +2 with row one and zero with row two. Rows $v+2$ to $2v$ have inner product zero with row one and +2 with row two.
A careful study of cases leads to only the cases now listed as Case III not being permutation equivalent to one of Cases I and II.

Case III. $\left[\begin{smallmatrix} x & y \\ x & \bar{y} \end{smallmatrix}\right]$, where one of the columns in the submatrix has two identical elements and the other has two different elements. The inner product of rows one and two with each other is zero. Each of row one and row two have inner product i with $v-1$ other rows and $2-i$ with the remainder of the rows, $i = 2$ or 0.

Case Ia. We have the possible 2×2 submatrices:

(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
$1\ \ y$	$-\ \ y$	$y\ \ 1$	$y\ \ -$	$1\ \ \bar{y}$	$-\ \ \bar{y}$	$\bar{y}\ \ 1$	$\bar{y}\ \ -$
$1\ \ \bar{y}$	$-\ \ \bar{y}$	$\bar{y}\ \ 1$	$\bar{y}\ \ -$	$1\ \ y$	$1\ \ y$	$y\ \ 1$	$y\ \ -$.

Since permutation of columns 1 and 2 has no effect on M_{2v-2}, Cases (i) and (iii), (v) and (vii), (ii) and (iv), and (vi) and (viii) give the same values. Cases (i) and (v) give the same values depending on whether $y = 1$ or -1. This leaves the following submatrices for Case Ia:

(i)	(ii)	(vi)
$1\ \ y$	$-\ \ y$	$-\ \ \bar{y}$
$1\ \ \bar{y}$	$-\ \ \bar{y}$	$1\ \ y$.

However, in Cases I rows 1 and 2 may be permuted without altering the value of M_{2v-2}, so, without loss of generality we may consider $y = 1$. Also, without loss of generality, we may permute rows three to $2v$ of the matrix so rows three to v have inner product +2 with rows one and two. The inner product of rows one and two with the next $v-2$ rows is 2, and the inner product of rows 1 and 2 with rows $v+1$ to $2v$ is zero. This yields the following cases for Case Ia.

TABLE 1

2×2 submatrix	Number of rows of each type Ia							
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
$1\ \ 1$ $1\ \ -$	$\lambda-1$	$k-\lambda-1$	$k-\lambda$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$-\ \ 1$ $-\ \ -$	λ	$k-\lambda$	$k-\lambda-1$	$v-2k+\lambda-1$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$-\ \ -$ $1\ \ 1$	λ	$k-\lambda-1$	$k-\lambda-1$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$

To illustrate the derivation of the tables such as Table 1 we give Case Ia as an example.

1	y		$-$	y		Inner product
1	$\bar{y}$		$-$	$\bar{y}$		of rows is 2
1	1		1	1		
$\vdots$	$\vdots$	$\lambda-1$	$\vdots$	$\vdots$	λ	
1	1		1	1		
1	$-$		1	$-$		$v-2$ rows
$\vdots$	$\vdots$	$k-\lambda-1$	$\vdots$	$\vdots$	$k-\lambda$	which have
1	$-$		1	$-$		inner
$-$	1		$-$	1		product 2
$\vdots$	$\vdots$	$k-\lambda$	$\vdots$	$\vdots$	$k-\lambda-1$	with rows
$-$	1		$-$	1		one and two
$-$	$-$		$-$	$-$		
$\vdots$	$\vdots$	$v-2k+\lambda$	$\vdots$	$\vdots$	$v-2k+\lambda-1$	
$-$	$-$		$-$	$-$		
1	1		1	1		
$\vdots$	$\vdots$	λ	$\vdots$	$\vdots$	λ	
1	1		1	1		
1	$-$		1	$-$		v rows
$\vdots$	$\vdots$	$k-\lambda$	$\vdots$	$\vdots$	$k-\lambda$	which have
1	$-$		1	$-$		inner
$-$	1		$-$	1		product 0
$\vdots$	$\vdots$	$k-\lambda$	$\vdots$	$\vdots$	$k-\lambda$	with rows
$-$	1		$-$	1		one and two
$-$	$-$		$-$	$-$		
$\vdots$	$\vdots$	$v-2k+\lambda$	$\vdots$	$\vdots$	$v-2k+\lambda$	
$-$	$-$		$-$	$-$		

Case Ib. A similar argument to that for Case Ia shows that using the permutations of columns 1 and 2 we have only to consider the submatrices

(i) (ii)

$$\begin{matrix} 1 & y \\ 1 & y \end{matrix} \qquad \begin{matrix} - & y \\ - & y \end{matrix}$$

for $y = 1$ and $y = -1$. These give the results for Table 2. We make the inner product with rows three to v with the first two rows equal $+2$ and the product of rows $v+1$ to $2v$ with the first two rows equal 0.

These reduce to three cases to test as the cases for

$$\begin{matrix} - & 1 \\ - & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} 1 & - \\ 1 & - \end{matrix}$$

give permutations only of the terms to be evaluated.

Case IIa. A similar argument to that for Case Ia shows that using the permutations of columns one and two we have only to consider the submatrices

(i) (ii)

$$\begin{matrix} 1 & y \\ 1 & \bar{y} \end{matrix} \qquad \begin{matrix} - & y \\ - & \bar{y} \end{matrix}$$

for $y = 1$ and $y = -1$. These give the results for Table 3.

These reduce to three cases to test as the cases for

$$\begin{matrix} - & 1 \\ - & - \end{matrix} \quad \text{and} \quad \begin{matrix} 1 & - \\ 1 & 1 \end{matrix}$$

TABLE 2

2×2 subsquare	Number of rows of each type Ib							
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
$\begin{matrix}1 & 1\\ 1 & 1\end{matrix}$	$\lambda-2$	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$\begin{matrix}- & 1\\ - & 1\end{matrix}$	λ	$k-\lambda-2$	$k-\lambda$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$\begin{matrix}1 & -\\ 1 & -\end{matrix}$	λ	$k-\lambda$	$k-\lambda-2$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$\begin{matrix}- & -\\ - & -\end{matrix}$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda-2$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$

TABLE 3

2×2 subsquare	Number of rows of each type IIa							
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
$\begin{matrix}1 & 1\\ 1 & -\end{matrix}$	$\lambda-1$	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$	λ	$k-\lambda-1$	$k-\lambda$	$v-2k+\lambda$
$\begin{matrix}- & 1\\ - & -\end{matrix}$	λ	$k-\lambda$	$k-\lambda-1$	$v-2k+\lambda$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda-1$
$\begin{matrix}1 & -\\ 1 & 1\end{matrix}$	λ	$k-\lambda-1$	$k-\lambda$	$v-2k+\lambda$	$\lambda-1$	$k-\lambda$	$k-\lambda$	$v-2k+\lambda$
$\begin{matrix}- & -\\ - & 1\end{matrix}$	λ	$k-\lambda$	$k-\lambda$	$v-2k+\lambda-1$	λ	$k-\lambda$	$k-\lambda-1$	$v-2k+\lambda$

give permutations only of the terms to be evaluated.

Case IIb. A similar argument to that for Case Ia shows that using the permutations of columns 1 and 2 we have only to consider the submatrices

$$\begin{matrix}\text{(i)} & \\ 1 & y\\ 1 & y\end{matrix} \qquad \begin{matrix}\text{(ii)} & \\ - & y\\ - & y\end{matrix}$$

for $y = 1$ and $y = -1$. These give the results for Table 4.

These reduce to three cases to test as the cases for

$$\begin{matrix}- & 1\\ - & 1\end{matrix} \quad \text{and} \quad \begin{matrix}1 & -\\ 1 & -\end{matrix}$$

give permutations, only of the terms to be evaluated.

Case III. We have the following 2×2 submatrices:

$$\begin{matrix}\text{(i)} & \\ x & y\\ x & \bar{y}\end{matrix} \qquad \begin{matrix}\text{(ii)} & \\ y & x\\ \bar{y} & x.\end{matrix}$$

One column removed comes from the columns with $2k$ ones per column and the other from the columns with v ones per column in the original design. This means the generic form of these two columns is the following.

TABLE 4

2×2 subsquare	Number of rows of each type IIb							
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
1 1 1 1	$\lambda - 1$	$k - \lambda$	$k - \lambda$	$v - 2k + \lambda$	$\lambda - 1$	$k - \lambda$	$k - \lambda$	$v - 2k + \lambda$
$-$ 1 $-$ 1	λ	$k - \lambda$	$k - \lambda - 1$	$v - 2k + \lambda$	λ	$k - \lambda$	$k - \lambda - 1$	$v - 2k + \lambda$
1 $-$ 1 $-$	λ	$k - \lambda - 1$	$k - \lambda$	$v - 2k + \lambda$	λ	$k - \lambda - 1$	$k - \lambda$	$v - 2k + \lambda$
$-$ $-$ $-$ $-$	λ	$k - \lambda$	$k - \lambda$	$v - 2k + \lambda - 1$	λ	$k - \lambda$	$k - \lambda$	$v - 2k + \lambda - 1$

$$
\begin{array}{lll}
1 & & 1 \\
 & & \vdots \\
1 & & \vdots \quad \rho \\
1 & & 1 \\
1 & k & - \\
\vdots & & \vdots \quad k - \rho \\
1 & & - \\
- & & 1 \\
\vdots & & \vdots \quad k - \rho \\
- & v - k & 1 \\
- & & - \\
\vdots & & \vdots \quad v - 2k + \rho \\
- & & - \\
\hline
1 & & 1 \\
 & & \vdots \\
1 & & \vdots \quad k - \rho \\
1 & & 1 \\
1 & v - k & - \\
\vdots & & \vdots \quad v - 2k + \rho \\
1 & & - \\
- & & 1 \\
\vdots & & \vdots \quad \rho \\
- & k & 1 \\
- & & - \\
\vdots & & \vdots \quad k - \rho \\
- & & -
\end{array}
$$

Note they have inner product zero. Also note $0 \leq \rho \leq k$. We have not proceeded to eliminate cases for ρ except where $\rho - 1 < 0$. Table 5 lists the possible cases that arise. In the case of

$$
\begin{matrix} 1 & - \\ 1 & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} - & - \\ - & 1 \end{matrix}
$$

the $u_i, i = 1, \ldots, 8$ were permutations of each other. This also occurred for

$$
\begin{matrix} 1 & 1 \\ 1 & - \end{matrix} \quad \text{and} \quad \begin{matrix} - & - \\ - & 1 \end{matrix} \quad \text{and for} \quad \begin{matrix} - & 1 \\ 1 & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} 1 & - \\ - & - \end{matrix}.
$$

Thus we have five theoretical values for Case III. Another two cases can arise by removing a pair of orthogonal rows and columns from the original design.

LEMMA 2.2. *The* $(2v-2) \times (2v-2)$ *minors of the* D*-optimal design of the* WB *series are*

TABLE 5

2×2 subsquare	Number of rows of each type III							
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
1 1 1 −	$\rho-1$	$k-\rho$	$k-\rho$	$v-2k+\rho$	$k-\rho$	$v-2k+\rho-1$	ρ	$k-\rho$
1 − 1 1	ρ	$k-\rho-1$	$k-\rho$	$v-2k+\rho$	$k-\rho-1$	$v-2k+\rho$	ρ	$k-\rho$
1 1 − 1	$\rho-1$	$k-\rho$	$k-\rho$	$v-2k+\rho$	$k-\rho$	$v-2k+\rho$	$\rho-1$	$k-\rho$
− 1 1 1	ρ	$k-\rho$	$k-\rho-1$	$v-2k+\rho$	$k-\rho-1$	$v-2k+\rho$	ρ	$k-\rho$
− − 1 −	ρ	$k-\rho$	$k-\rho$	$v-2k+\rho-1$	$k-\rho$	$v-2k+\rho-1$	ρ	$k-\rho$

$\hat{0}, (2s+1)(s+1)s\mathcal{T}, \quad (2s+1)s^2\mathcal{T}, \quad 2s^2(s+1)\mathcal{T}, \quad s(2s^2+2s+1)\mathcal{T}, \quad 2s(s+1)^2\mathcal{T},$
$2s^3\mathcal{T}, s(2s+1)\mathcal{T}, s(s+1)\mathcal{T}, s^2\mathcal{T},$

where s is an odd prime power, $s = 2,\ 4,\ 6,$ or 8 and $\mathcal{T} = 2^{4s^2+4s-1}s^{2s^2+2s-3}(s+1)^{2s^2+2s-2}$.

Proof. Here $\lambda = \frac{1}{2}s(s-1), \quad k = s^2, \quad$ and $\quad v = 2s^2+2s+1$. The expressions for $u_i, i = 1, \ldots, 8$, were calculated in each case. Maple was then used to evaluate the determinant for D giving the required result. Ia and IIa give the values $2^7s^2(2s+1)(s+1)^3$, $2^7s^3(2s+1)(s+1)^2$, and $2^7s^2(2s+1)(s+1)^2$.

Cases Ib and IIb give the value zero for the determinant.

Case III give the values $2^7s^3(s+1)^3$, $2^7(2s^2+2s+1)(s+1)^2s^2$, $2^8s^2(s+1)^4$, $2^8s^4(s+1)^2$, $2^7s^2(s+1)^3$, and $2^7s^3(s+1)^2$.

Multiplying by $(4s^2+4s)^{2s^2+2s-4}$ gives the required result. □

3. Pivot structure for WB family of *D*-optimal designs.

Conjecture (the growth conjecture for WB family). *Let A be an $2v\times 2v$ CP D-optimal design of WB family which is constructed from $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ sds. Reduce A by GE. Then we conjecture*

(i) $g(v, A) = 2s(2s+1), \quad or \quad 2(s+1)(2s+1)$;
(ii) *the last pivot is equal to* $2s(2s+1)$ *or* $2(s+1)(2s+1)$;
(iii) *the second last pivot can take the values* $2s(s+1) = \frac{2v-1}{2}$, $2s^2$, $2(s+1)^2$, $(s+1)(2s+1)$, $\frac{(2s+1)2s(s+1)^2}{(2s^2+2s+1)}$, $s(2s+1)$, $\frac{(s+1)^2(2s+1)}{s}$, $\frac{(s)^2(2s+1)}{(s+1)}$, *and* $\frac{(2s+1)2s^2(s+1)}{(2s^2+2s+1)}$;
(iv) *every pivot before the last has magnitude at most* $2v$;
(v) *the first four pivots are equal to* $1, 2, 2, 4$;
(vi) *the fifth pivot may be* 2 *or* 3.

We prove (ii) *and* (iii) *in this paper.* (v) *and* (vi) *were proved for Brouwer's SBIBD* $(2s^2+2s+1, s^2, \frac{1}{2}s(s-1))$ *in* [11] *and we also show they hold for the WB family.*

THEOREM 3.1. *Let A be the $2v\times 2v$ D-optimal design of the WB family. Reduce A by GECP. Then the last pivots are $2s(2s+1)$ and $2(s+1)(2s+1)$. The second last pivots are $2s(s+1)$, $2s^2$, $2(s+1)^2$, $(s+1)(2s+1)$, $\frac{(2s+1)2s(s+1)^2}{(2s^2+2s+1)}$, $s(2s+1)$, $\frac{(s+1)^2(2s+1)}{s}$, $\frac{(s)^2(2s+1)}{(s+1)}$, and $\frac{(2s+1)2s^2(s+1)}{(2s^2+2s+1)}$.*

Proof. From (1.4) and Lemmas 2.1 and 2.2 we have for the D-optimal design made using $2-\{2s^2+2s+1; s^2, s^2; s(s-1)\}$ sds, the results in Table 6, where the first row gives the values of M_{2v-1}, the first column gives the values of M_{2v-2} appearing in GECP and the entries are $p_{2v} = \frac{M_{2v}}{M_{2v-1}}$, $p_{2v-1} = \frac{M_{2v-1}}{M_{2v-2}}$. □

TABLE 6

M_{2v-1}	$2s^2(s+1)^2(2s+1)\mathcal{T}$	$2s^3(s+1)(2s+1)\mathcal{T}$
M_{2v-2}		
$(2s+1)(s+1)\mathcal{T}$	$2s(s+1)^*$	$2s^2$
$(2s+1)s^2\mathcal{T}$	$2(s+1)^2$	$2s(s+1)$
$2s^2(s+1)\mathcal{T}$	$(s+1)(2s+1)$	$s(2s+1)$
$s(2s^2+2s+1)\mathcal{T}$	$\frac{2s(s+1)^2(2s+1)}{(2s^2+2s+1)}^*$	$\frac{2s^2(s+1)(2s+1)}{(2s^2+2s+1)}$
$2s(s+1)^2\mathcal{T}$	$s(2s+1)^*$	$\frac{s^2(2s+1)}{(s+1)}$
$2s^3\mathcal{T}$	$\frac{(s+1)^2(2s+1)}{s}$	$(s+1)(2s+1)$

TABLE 7

$2v$	s	p_{2v}		p_{2v-1}			
		$2s(2s+1)$	$2(s+1)(2s+1)$	$2s(s+1)$	$2(s+1)^2$	$2s^2$	$(s+1)(2s+1)$
26	2	20	30	12	18	8	15
50	3	42	56	24	32	18	28
82	4	72	90	40	50	32	45

TABLE 7 (continued)

$2v$	s	p_{2v-1}				
		$s(2s+1)$	$\frac{2s(s+1)^2(2s+1)}{(2s^2+2s+1)}$	$\frac{2s^2(s+1)(2s+1)}{(2s^2+2s+1)}$	$\frac{s^2(2s+1)}{(s+1)}$	$\frac{(s+1)^2(2s+1)}{s}$
26	2	10	$\frac{180}{13}$	$\frac{120}{13}$	$\frac{20}{3}$	$\frac{45}{2}$
50	3	21	$\frac{672}{25}$	$\frac{704}{25}$	$\frac{63}{4}$	$\frac{112}{3}$
82	4	36	$\frac{1800}{41}$	$\frac{1440}{41}$	$\frac{144}{5}$	$\frac{225}{4}$

The entries marked $*$ are these obtained in experiments.

In Table 7 we give some values for the family WB.

Remark. We experimented with $2v = 26$. The theoretical values for M_{2v-1} are $2^{35}\cdot 5\cdot 3^{12}$ and $2^{36}\cdot 5\cdot 3^{11}$. In our calculations we found always $p_{2v} = 20$ and $p_{2v-1} = 12$ or 10 or $\frac{180}{13}$. This leaves as an open problem the existence of a 26×26 matrix having growth equal to 30.

The next result is easy to prove using a counting argument and noting the inner product of every pair of rows is $+1$ to see that the design always contains a 4×4 Hadamard matrix.

PROPOSITION 3.2. *Let A be the $2v \times 2v$ $(1,-1)$ incidence matrix of an SBIBD of the WB family. Reduce A by GECP, then the magnitudes of the first four pivots are $1, 2, 2,$ and 4; the magnitude of $|a_{55}^{(4)}|$ is 2 or 3.*

Proof. Since the design always contains a 4×4 Hadamard matrix, this can be moved to be the 4×4 principal minor without changing the CP property. Thus the first four pivots will be $1, 2, 2,$ and 4 [4]. Because every entry in $A^{(3)}$ is of magnitude $0, 2,$ or 4, pivoting on $a_{44}^{(3)}$ will involve only adding ± 1 or $\pm 1/2$ times the fourth row of $A^{(3)}$ to the rows below and this will create only integer entries in $A^{(4)}$. It is known (see Payne [14]) that if $v \equiv 1(\mathrm{mod}4), \quad v \neq 1, \quad d_v \leq (v-1)^{\frac{v-1}{2}}\sqrt{2v-1}$, and equality can hold only if $v = 2s^2 + 2s + 1, \quad s = 1, 2, 3, \ldots$. Thus $|a_{55}^{(4)}|$ must be an integer satisfying the relation

$$A(1\,2\,3\,4\,5) = 16|a_{55}^{(4)}| \leq 4^{4/2}\sqrt{10-1} \Rightarrow |a_{55}^{(4)}| \leq 3,$$

TABLE 8
Growth factors and pivots patterns for small CP WB designs.

s	$2v$	growth	pivot pattern
2	26	20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,\frac{18}{4},\ldots,12,20)$
2	26	20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,\frac{44}{9},\ldots,,\frac{180}{13},20)$
2	26	20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,4,\ldots,10,20)$
3	50	42	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,5,\ldots,24,42)$
3	50	42	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,4,\ldots,21,42)$
4	82	72	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,4,\ldots,40,72)$
5	122	110	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,4,\ldots,60,110)$

where $A(1\,2\,3\,4\,5)$ denotes the determinant of the 5×5 principal submatrix of A. Thus $|a_{55}^{(4)}|$ must be $1,2$, or 3. To see that it cannot be 1 is to show that one could not have

$$A^{(4)}\begin{bmatrix} 1 & & & & \\ & 2 & & & \\ & & 2 & & \\ & & & 4 & \\ & & & & B \end{bmatrix},$$

where every entry of B is zero or ± 1, for, if that were true, then B would be a normalized $(v-4)\times(v-4)$ matrix, and so

$$|\det B| \le (v-4)^{\frac{v-4}{2}}.$$

But $|\det B| = \frac{(v-1)^{\frac{v-1}{2}}\sqrt{2v-1}}{16}$ and it is easily checked that these cannot both hold when $v > 4$. □

By detecting the pivot structure of WB, Table 8 was computed. The first nine pivots and the last two are presented. All the other intermediate pivots take a variety of values. At least 837 different pivot structures were detected for $2v = 26$ and 500 for $2v = 50$.

4. Appendix: The determinant simplification theorem. We use the notation

$$CC^T = (k - a_{ii})I_{b_1,b_2,\ldots,b_z} + a_{ij}J_{b_1,b_2,\ldots,b_z}$$

for a matrix of blocks with integer multiples. For example, consider the matrix

$$CC^T = (k - a_{ii})I_{u,v,w,x} + a_{ij}J_{u,v,w,x}, \tag{4.1}$$

where

$$(a_{ij}) = \begin{bmatrix} a & b & c & d \\ b & a & e & f \\ c & e & a & g \\ d & f & g & a \end{bmatrix}$$

is the $(u+v+w+x)\times(u+v+w+x)$ matrix

$$CC^T = \begin{bmatrix} \overbrace{k\ a\ \cdots\ a}^{u} & \overbrace{b\ b\ \cdots\ b}^{v} & \overbrace{c\ c\ \cdots\ c}^{w} & \overbrace{d\ d\ \cdots\ d}^{x} \\ a\ k\ \cdots\ a & b\ b\ \cdots\ b & c\ c\ \cdots\ c & d\ d\ \cdots\ d \\ \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots \\ a\ a\ \cdots\ k & b\ b\ \cdots\ b & c\ c\ \cdots\ c & d\ d\ \cdots\ d \\ \\ b\ b\ \cdots\ b & k\ a\ \cdots\ a & e\ e\ \cdots\ e & f\ f\ \cdots\ f \\ b\ b\ \cdots\ b & a\ k\ \cdots\ a & e\ e\ \cdots\ e & f\ f\ \cdots\ f \\ \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots \\ b\ b\ \cdots\ b & a\ a\ \cdots\ k & e\ e\ \cdots\ e & f\ f\ \cdots\ f \\ \\ c\ c\ \cdots\ c & e\ e\ \cdots\ e & k\ a\ \cdots\ a & g\ g\ \cdots\ g \\ c\ c\ \cdots\ c & e\ e\ \cdots\ e & a\ k\ \cdots\ a & g\ g\ \cdots\ g \\ \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots \\ c\ c\ \cdots\ c & e\ e\ \cdots\ e & a\ a\ \cdots\ k & g\ g\ \cdots\ g \\ \\ d\ d\ \cdots\ d & f\ f\ \cdots\ f & g\ g\ \cdots\ g & k\ a\ \cdots\ a \\ d\ d\ \cdots\ d & f\ f\ \cdots\ f & g\ g\ \cdots\ g & a\ k\ \cdots\ a \\ \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots & \vdots\ \ \ \vdots \\ d\ d\ \cdots\ d & f\ f\ \cdots\ f & g\ g\ \cdots\ g & a\ a\ \cdots\ k \end{bmatrix}.$$

We now give a theorem proved similarly to the proof for finding the determinant of an SBIBD in [15, Thm. 3, p. 32].

THEOREM 4.1 (determinant simplification theorem). *Let*

$$CC^T = (k - a_{ii})I_{b_1,b_2,\ldots,b_z} + a_{ij}J_{b_1,b_2,\ldots,b_z},$$

then

$$\det\, CC^T = \Pi_{i=1}^{z}(k - a_{ii})^{b_i - 1}\det\, D, \tag{4.2}$$

where

$$D = \begin{bmatrix} k + (b_1 - 1)a_{11} & b_2a_{12} & b_3a_{13} & \cdots & b_za_{1z} \\ b_1a_{21} & k + (b_2 - 1)a_{22} & b_3a_{23} & \cdots & b_za_{2z} \\ \vdots & \vdots & \vdots & & \vdots \\ b_1a_{z1} & b_2a_{z2} & b_3a_{z2} & \cdots & k + (b_z - 1)a_{zz} \end{bmatrix}.$$

COROLLARY 4.2. *Suppose C is the matrix of order $(u + v + w + x) \times (u + v + w + x)$, where $n = u + v + w + x$, for which CC^T is given above, satisfying $CC^T = (k - a_{ii})I_{u,v,w,x} + a_{ij}J_{u,v,w,x}$. Then*

$$\det\, CC^T = (k - a)^{n-4}\det\, D,$$

where

$$D = \begin{bmatrix} k + (u - 1)a & vb & wc & xd \\ ub & k + (v - 1)a & we & xf \\ uc & ve & k + (w - 1)a & xg \\ ud & vf & wg & k + (x - 1)a \end{bmatrix}. \tag{4.3}$$

REFERENCES

[1] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, UK, 1985.

[2] W.G. Bridges, M. Hall Jr., and J.L. Hayden, *Codes and designs*, J. Combin. Theory Ser. A, 31 (1981), pp. 155–174.

[3] A.E. Brouwer, *An Infinite Series of Symmetric Designs*, Afdeling Zuivere Wiskunde 202, Mathematisch Centrum, Amsterdam, 1983.

[4] J. Day and B. Peterson, *Growth in Gaussian elimination*, Amer. Math. Monthly, 95 (1988), pp. 489–513.

[5] D.Z. Djokovic, *Some new D-optimal designs*, Australas. J. Combin., 15 (1997), pp. 221–231.

[6] H. Ehlich, *Determinantenabschätzungen für binäre matrizen*, Math. Z., 83 (1964), pp. 123–132.

[7] S. Georgiou and C. Koukouvinos, *On multipliers of supplementary difference sets and D-optimal designs for* $n \equiv 2\text{mod}4$, Util. Math., 56 (1999), pp. 127–136.

[8] M. Gysin, *New D-optimal designs via cyclotomy and generalised cyclotomy*, Australas. J. Combin., 15 (1997), pp. 247–255.

[9] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[10] C. Koukouvinos, S. Kounias, and J. Seberry, *Supplementary difference sets and optimal designs*, Discrete Math., 88 (1991), pp. 49–58.

[11] C. Koukouvinos, M. Mitrouli, and J. Seberry, *Values of minors of* $(1, -1)$ *incidence matrices of SBIBDs and their application to the growth problem*, Des. Codes Cryptogr., to appear.

[12] C. Koukouvinos, M. Mitrouli, and J. Seberry, *Growth in Gaussian elimination for weighing matrices* $W(n, n-1)$, Linear Algebra Appl., 306 (2000), pp. 189–202.

[13] C. Koukouvinos, J. Seberry, A.L. Whiteman, and M.Y. Xia, *Optimal designs, supplementary difference sets and multipliers*, J. Statist. Plann. Inference, 62 (1997), pp. 81–90.

[14] S.E. Payne, *On maximizing* $\det(AA^T)$, Discrete Math., 10 (1974), pp. 145–158.

[15] A.P. Street and D.J. Street, *Combinatorics of Experimental Design*, Oxford University Press, Oxford, UK, 1987.

[16] T. van Trung, *The existence of symmetric block designs with parameters* $(41, 16, 6)$ *and* $(66, 26, 10)$, J. Combin. Theory Ser. A, 33 (1982), pp. 201–204.

[17] T. van Trung, *Symmetric designs*, in CRC Handbook of Combinatorial Designs, C.J. Colbourn and J.H. Dinitz, eds., CRC Press, Boca Raton, FL, 1996, pp. 75–87.

[18] A.L. Whiteman, *A family of D-optimal designs*, Ars Combin., 30 (1990), pp. 23–26.

[19] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1988.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 15–41
© 2001 Society for Industrial and Applied Mathematics

A FULLY ASYNCHRONOUS MULTIFRONTAL SOLVER USING DISTRIBUTED DYNAMIC SCHEDULING*

PATRICK R. AMESTOY†, IAIN S. DUFF‡, JEAN-YVES L'EXCELLENT§, AND JACKO KOSTER¶

Abstract. In this paper, we analyze the main features and discuss the tuning of the algorithms for the direct solution of sparse linear systems on distributed memory computers developed in the context of a long term European research project. The algorithms use a multifrontal approach and are especially designed to cover a large class of problems. The problems can be symmetric positive definite, general symmetric, or unsymmetric matrices, both possibly rank deficient, and they can be provided by the user in several formats. The algorithms achieve high performance by exploiting parallelism coming from the sparsity in the problem and that available for dense matrices. The algorithms use a dynamic distributed task scheduling technique to accommodate numerical pivoting and to allow the migration of computational tasks to lightly loaded processors. Large computational tasks are divided into subtasks to enhance parallelism. Asynchronous communication is used throughout the solution process to efficiently overlap communication with computation.

We illustrate our design choices by experimental results obtained on an SGI Origin 2000 and an IBM SP2 for test matrices provided by industrial partners in the PARASOL project.

Key words. sparse linear equations, Gaussian elimination, multifrontal methods, asynchronous parallelism, distributed memory computation, dynamic scheduling

AMS subject classifications. 65F05, 65F50

PII. S0895479899358194

1. Introduction.

We consider the direct solution of large sparse linear systems on distributed memory computers. The systems are of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an $n \times n$ symmetric positive definite, general symmetric, or unsymmetric sparse matrix that is possibly rank deficient, $\mathbf{b}$ is the right-hand side vector, and $\mathbf{x}$ is the solution vector to be computed.

Most of the work presented in this article was performed as part of Work Package 2.1 within the PARASOL project [6]. PARASOL was an ESPRIT IV Long Term Research project (no. 20160) for "An Integrated Environment for Parallel Sparse Matrix Solvers." The main goal of this project, which started in January 1996 and finished in June 1999, was to build and test a portable library for solving large sparse systems of equations on distributed memory systems. The library is now in the public domain (www.pallas.de/parasol) and contains routines for both direct and iterative solutions of symmetric positive definite, symmetric general, and unsymmetric systems.

The direct solver that we produced for the PARASOL project is a `MU`ltifrontal `M`assively `P`arallel `S`olver for which we use the acronym `MUMPS`. We continue to develop `MUMPS` and updated versions of the code are available.[1] Several aspects of the algo-

*Received by the editors July 7, 1999; accepted for publication (in revised form) by E. Ng November 19, 2000; published electronically April 18, 2001. This work was partially supported by the PARASOL Project (EU ESPRIT IV LTR Project 20160).

http://www.siam.org/journals/simax/23-1/35819.html

†ENSEEIHT-IRIT, 2 rue Camichel, 31071 Toulouse cedex, France (amestoy@enseeiht.fr).

‡Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX England, and CERFACS, Toulouse, France (I.Duff@rl.ac.uk).

§NAG Ltd., Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR England (excelle@enseeiht.fr).

¶Parallab, University of Bergen, 5020 Bergen, Norway (jak@ii.uib.no).

[1]Information on how to obtain updated copies of `MUMPS` can be obtained from the web page http://www.enseeiht.fr/apo/MUMPS/ or by sending email to mumps@cerfacs.fr.

rithms used in MUMPS combine to give us an approach which is unique among sparse direct solvers. These include

- partial threshold pivoting during numerical factorization requiring the use of dynamic data structures,
- the ability to automatically adapt to computer load variations during the numerical phase,
- high performance, by exploiting the independence of computations due to sparsity and that available for dense matrices, and
- the capability of solving a wide range of problems, including symmetric positive definite, symmetric general, unsymmetric, and rank-deficient systems using either $\mathbf{LU}$ or $\mathbf{LDL}^T$ factorization.

To address all these factors, we have designed a fully asynchronous algorithm based on a multifrontal approach with distributed dynamic scheduling of tasks. The current version of our package provides a large range of options, including the possibility of inputting the matrix in assembled format either on a single processor or distributed over the processors. Additionally, the matrix can be input in elemental format (currently only on one processor). MUMPS can also determine the rank and a null-space basis for rank-deficient matrices, and can return a Schur complement matrix. It contains classical pre- and postprocessing facilities; for example, matrix scaling, iterative refinement, and error analysis.

To control the growth in the factors, partial threshold pivoting has been used to handle both unsymmetric and general symmetric matrices. To preserve the symmetry of general symmetric matrices, pivoting is restricted to the diagonal in that case. For symmetric positive definite matrices, there is no need for numerical pivoting and a quite different approach based on a static mapping of the tasks and the data could have been used for the factorization phase (see, for example, [8, 28, 29, 30]). We have not chosen this option in our algorithms. Instead, we use the same (dynamic) approach for all classes of matrices. When the user indicates that the matrix is symmetric positive definite, the main differences with respect to the general symmetric case lie in the numerical pivoting and the (parallel) dense matrix kernels that are involved. These differences will be further discussed in sections 4.3 and 4.5, once the main features of our parallel algorithms have been introduced.

Among the other work on distributed memory sparse direct solvers of which we are aware [8, 11, 15, 16, 26, 28, 29, 30, 32, 36, 37, 40], we do not know of any with the same capabilities as the MUMPS solver. To our knowledge, only SPOOLES[2] [9] handles numerical pivoting and offers comparable functionalities in a distributed memory environment. Because of the difficulty of handling dynamic data structures efficiently, most distributed memory approaches do not perform numerical pivoting during the factorization phase. Instead, they are based on a static mapping of the tasks and data and do not allow task migration during numerical factorization. Numerical pivoting can clearly be avoided for symmetric positive definite matrices. For unsymmetric matrices, Duff and Koster [22, 23] have designed algorithms to permute large entries onto the diagonal and have shown that this can significantly reduce numerical pivoting. Demmel and Li [16] have shown that, if one preprocesses the matrix using the code of Duff and Koster, static pivoting (with possibly modified diagonal values) followed by iterative refinement can normally provide reasonably accurate solutions.

[2]Developed by the Mathematics and Engineering Analysis Unit of Boeing Phantom Works and supported by DARPA contract DABT63-95-C-0122 and the DoD High Performance Computing Modernization Program Common HPC Software Support Initiative.

They have observed that this preprocessing, in combination with an appropriate scaling of the input matrix, is a key issue for the numerical stability of their approach.

The rest of this paper is organized as follows. We first introduce some of the main terms used in a multifrontal approach in section 2. Throughout this paper, we study the performance obtained on the set of test problems that we describe in section 3. We discuss, in section 4, the main parallel features of our approach. In section 5, we give initial performance figures and show the influence of the ordering of the variables on the performance of `MUMPS`. In section 6, we describe our work on the input of matrices in elemental form. Section 7 then briefly describes the main properties of the algorithms used for distributed assembled matrices. In section 8, we comment on memory scalability issues. In section 9, we describe and analyze the distributed dynamic scheduling strategies that will be further analyzed in section 10 where we examine how we can modify the assembly tree to introduce more parallelism. We present a summary of our results in section 11.

The majority of the results presented in this paper have been obtained on the 35 processor IBM SP2 located at GMD (National Research Center for Information Technology in Bonn, Germany). Each node of this computer is a 66 MHertz processor with 128 MBytes of physical memory and 512 MBytes of virtual memory. The SGI Origin 2000 from Parallab (University of Bergen, Norway) has also been used to run some of our largest test problems. The Parallab computer consists of 64 nodes sharing 24 GBytes of physically distributed memory. Each node has two R10000 MIPS RISC 64-bit processors sharing 384 MBytes of local memory. Each processor runs at a frequency of 195 MHertz and has a peak performance of a little under 400 Mflops per second.

All experiments reported in this paper use Version 4 of `MUMPS`. The software is written in Fortran 90. It requires MPI for message passing and makes use of BLAS [18, 19], LAPACK [7], BLACS [17], and ScaLAPACK [10] subroutines. On the IBM SP2, we used a nonoptimized portable local installation of ScaLAPACK, because the IBM optimized library PESSL V2 was not available.

2. Multifrontal methods. It is not our intention to describe the details of a multifrontal method. Rather, we just define terms used later in the paper and refer the reader to earlier publications for a more detailed description, for example, [3, 20, 24].

In the multifrontal method, all elimination operations take place within dense submatrices, called *frontal matrices*. A frontal matrix can be partitioned as shown in Figure 2.1. In this matrix, pivots can be chosen from within the block $\mathbf{F}_{11}$ only. The Schur complement matrix $\mathbf{F}_{22} - \mathbf{F}_{21}\mathbf{F}_{11}^{-1}\mathbf{F}_{12}$ is computed and used to update later rows and columns of the overall matrix. We call this update matrix the *contribution block*.

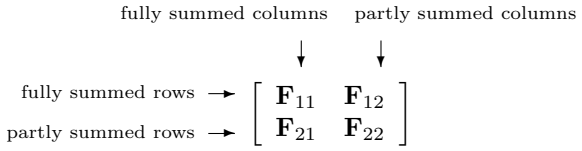


FIG. 2.1. *Partitioning of a frontal matrix.*

The overall factorization of the sparse matrix using a multifrontal scheme can be described by an *assembly tree*, where each node corresponds to the computation

of a Schur complement as just described, and each edge represents the transfer of the contribution block from the son node to the parent node (or father) in the tree. This parent node assembles (or sums) the contribution blocks from all its son nodes with entries from the original matrix. If the original matrix is given in assembled format, complete rows and columns of the input matrix are assembled at once, and, to facilitate this, the input matrix is ordered according to the pivot order and stored as a collection of arrowheads. For example, if the permuted matrix has entries in columns $\{j_1, j_2, j_3\}$ of row i, $i < j_1, j_2, j_3$, and in rows $\{k_1, k_2\}$ of column i, $i < k_1, k_2$, then the arrowhead list associated with variable i is $\{a_{ii}, a_{ij_1}, a_{ij_2}, a_{ij_3}, a_{k_1 i}, a_{k_2 i}\}$. In the symmetric case, only entries from the lower triangular part of the matrix are stored. We say that we are storing the matrix in *arrowhead form* or by *arrowheads*. For unassembled matrices, complete element matrices are assembled into the frontal matrices and the input matrix need not be preprocessed.

In our implementation, the assembly tree is constructed from the symmetrized pattern of the matrix and a given sparsity ordering. By symmetrized pattern, we mean the pattern of the matrix $\mathbf{A} + \mathbf{A}^T$ where the summation is symbolic. Note that this allows the matrix to be unsymmetric.

Because of numerical pivoting, it is possible that some variables cannot be eliminated from a frontal matrix. The fully summed rows and columns that correspond to such variables are added to the contribution block that is sent to the parent node resulting in a larger than predicted frontal matrix. The assembly of fully summed rows and columns into the frontal matrix of the parent node means that the corresponding elimination operations have been *delayed*. The delay of eliminations corresponds to an a posteriori change to the data passed from son to father in the assembly tree and in the sizes of the frontal matrices. In general, this introduces additional (numerical) fill-in in the factors.

An important aspect of the assembly tree is that operations at a pair of nodes where neither is an ancestor of the other are independent. This makes it possible to obtain parallelism from the tree (so-called *tree parallelism*). For example, work can commence in parallel on all the leaf nodes of the tree. Fortunately, near the root node of the tree, where the tree parallelism is very poor, the frontal matrices are usually much larger and so techniques for exploiting parallelism in dense factorizations can be used (for example, blocking and use of higher level BLAS). We call this *node parallelism*. We discuss further aspects of the parallelism of the multifrontal method in later sections of this paper. Our work is based on our experience with designing and implementing a multifrontal scheme on shared and virtual shared memory computers (for example, [2, 3, 4]) and on an initial prototype distributed memory multifrontal version [25]. We describe the design of our resulting distributed memory multifrontal algorithm in the rest of this paper.

3. Test problems. Throughout this paper, we will use a set of test problems to illustrate the performance of our algorithms. We describe the set in this section.

In Tables 3.1 and 3.2, we list, respectively, our unassembled and assembled test problems. All except one come from the industrial partners of the PARASOL project and are available at Parallab (Bergen, Norway).[3] The remaining matrix, BBMAT, is from the forthcoming Rutherford-Boeing Sparse Matrix Collection [21]. All the symmetric matrices in our test set were stated by their source to be positive definite and, in our experiments, we treated them as if they were. In fact, we have subse-

[3]They can be found by following links from the web page http://www.parallab.uib.no/parasol/.

quently discovered that the factorization of BMW3_2 has 11 negative pivots. Typical PARASOL test cases are from the following major application areas: computational fluid dynamics (CFD), structural mechanics, modelling compound devices, modelling ships and mobile offshore platforms, industrial processing of complex non-Newtonian liquids, and modelling car bodies and engine components. All the test problems in elemental format are also provided in assembled format. The suffix (RSE or RSA) is used to differentiate them. For those in elemental format, the original matrix is represented as a sum of element matrices

$$\mathbf{A} = \sum \mathbf{A}_i \ ,$$

where each $\mathbf{A}_i$ has nonzero entries only in those rows and columns that correspond to variables in the ith element. Because element matrices may overlap, the number of entries of a matrix in elemental format is usually larger than for the same matrix when assembled (compare the matrices from Det Norske Veritas in Norway in Tables 3.1 and 3.2). Typically there are about twice the number of entries in the unassembled elemental format.

TABLE 3.1
Unassembled symmetric test matrices from PARASOL partner (in elemental format).

Real Symmetric Positive Definite Elemental (RSE)				
Matrix name	Order	No. of elements	No. of entries	Origin
THREAD.RSE	29736	2176	3718704	Det Norske Veritas
SHIP_001.RSE	34920	3431	3686133	Det Norske Veritas
M_T1.RSE	97578	5328	6882780	Det Norske Veritas
X104.RSE	108384	6019	7065546	Det Norske Veritas
SHIPSEC8.RSE	114919	35280	7431867	Det Norske Veritas
SHIP_003.RSE	121728	45464	9729631	Det Norske Veritas
SHIPSEC1.RSE	140874	41037	8618328	Det Norske Veritas
SHIPSEC5.RSE	179860	52272	11118602	Det Norske Veritas

In Tables 3.3, 3.4, and 3.5, we present statistics on the factorizations of the various test problems using `MUMPS`. The tables show the number of entries in the factors and the number of floating-point operations (flops) for the elimination. For symmetric matrices, we give the number of entries in the lower triangular part of the matrix. For unsymmetric matrices, we show both the estimated number, assuming no additional pivoting, and the actual number when numerical pivoting is used. All our unsymmetric matrices are row and column scaled (each row/column is divided by its maximum value). This is a default option of `MUMPS` for unsymmetric matrices. Scaling significantly reduces the number of delayed pivots. The average number (over the three unsymmetric matrices) is equal to 19257 before scaling and 1527 after. Permuting large entries onto the diagonal [22] can significantly reduce the amount of numerical pivoting but can also deteriorate the structural symmetry of the matrices. On our set of relatively large unsymmetric matrices with a fairly symmetric pattern (structural symmetry larger than 0.54), row and column scaling was enough to reduce numerical pivoting (compare the columns "estim." and "actual" in Table 3.5).

The statistics clearly depend on the ordering used. Two classes of ordering are considered in this paper. The first is an approximate minimum degree ordering (referred to as AMD; see [1]). The second class is based on a hybrid nested dissection and minimum degree technique (referred to as ND). These hybrid orderings were generated using ONMETIS [35] or a combination of the graph partitioning tool SCOTCH

TABLE 3.2
Assembled test matrices from PARASOL partners (except the matrix BBMAT*).* (*) *StrSym is the number of nonzeros matched by nonzeros in symmetric locations divided by the total number of entries.*

Real Unsymmetric Assembled (RUA)				
Matrix name	Order	No. of entries	StrSym(*)	Origin
MIXTANK	29957	1995041	1.00	Polyflow S.A.
INVEXTR1	30412	1793881	0.97	Polyflow S.A.
BBMAT	38744	1771722	0.54	Rutherford-Boeing (CFD)
Real Symmetric Assembled (RSA)				
Matrix name	Order	No. of entries		Origin
OILPAN	73752	1835470		INPRO
SMDOOR	162610	4036144		INPRO
LDOOR	952203	23737339		INPRO
CRANKSG1	52804	5333507		MSC.Software
CRANKSG2	63838	7106348		MSC.Software
BMW7ST_1	141347	3740507		MSC.Software
BMWCRA_1	148770	5396386		MSC.Software
BMW3_2	227362	5757996		MSC.Software
INLINE_1	503712	18660027		MSC.Software
THREAD.RSA	29736	2249892		Det Norske Veritas
SHIP_001.RSA	34920	2339575		Det Norske Veritas
M_T1.RSA	97578	4925574		Det Norske Veritas
X104.RSA	108384	5138004		Det Norske Veritas
SHIPSEC8.RSA	114919	3384159		Det Norske Veritas
SHIP_003.RSA	121728	4103881		Det Norske Veritas
SHIPSEC1.RSA	140874	3977139		Det Norske Veritas
SHIPSEC5.RSA	179860	5146478		Det Norske Veritas

TABLE 3.3
*Statistics for symmetric test problems, available in both assembled (*RSA*) and unassembled (*RSE*) formats (MFR ordering from Det Norske Veritas).*

Matrix	Entries in factors ($\times 10^6$)	Flops ($\times 10^9$)
THREAD	24.3	38.8
SHIP_001	14.5	9.4
M_T1	29.5	16.8
X104	24.1	9.8
SHIPSEC8	34.3	33.7
SHIP_003	57.1	73.0
SHIPSEC1	36.6	32.1
SHIPSEC5	50.8	51.7

TABLE 3.4
Statistics for symmetric test problems on the SGI Origin 2000.

	AMD ordering			ND ordering	
Matrix	Entries in factors ($\times 10^6$)	Flops ($\times 10^9$)	Time for analysis (seconds)	Entries in factors ($\times 10^6$)	Flops ($\times 10^9$)
OILPAN	10.2	3.8	1.9	9.5	3.2
SMDOOR	26.0	13.2	4.7	23.5	11.5
LDOOR	156.0	110.5	39.4	146.3	74.5
CRANKSG1	40.1	50.2	4.7	31.8	29.7
CRANKSG2	60.7	101.9	7.5	40.9	41.6
BMW7ST_1	27.1	15.4	4.2	25.0	11.3
BMW3_2	51.1	44.9	8.3	44.9	28.6
BMWCRA_1	97.2	127.9	8.5	70.3	61.0
INLINE_1	222.1	242.0	36.1	174.9	143.2

TABLE 3.5
Statistics for unsymmetric test problems. Time for analysis in seconds on the SGI Origin 2000.

	AMD ordering					ND ordering			
Matrix	Entries in factors ($\times 10^6$)		Flops ($\times 10^9$)		Time	Entries in factors ($\times 10^6$)		Flops ($\times 10^9$)	
	estim.	actual	estim.	actual		estim.	actual	estim.	actual
MIXTANK	38.5	39.1	64.1	64.4	2.6	18.9	19.6	13.0	13.2
INVEXTR1	30.3	31.2	34.3	35.8	2.3	15.7	16.1	7.7	8.1
BBMAT	46.0	46.2	41.3	41.6	3.9	35.7	35.8	25.5	25.7

[38] with a variant of AMD (Halo-AMD; see [39]). For matrices available in both assembled and unassembled format, we used nested dissection based orderings provided by Det Norske Veritas and denote these by MFR.

Note that, in this paper, it is not our intention to compare the packages that we used to obtain the orderings; we will only discuss the influence of the type of ordering on the performance of `MUMPS` (in section 5).

4. Parallel implementation issues. In this paper, we assume a one-to-one mapping between processes and processors in our distributed memory environment. A process will thus implicitly refer to a unique processor and, when we say, for example, that a task is allocated to a process, we mean that the task is also mapped onto the corresponding processor.

As we did before in a shared memory environment [4], we exploit both tree parallelism (arising from sparsity) and node parallelism (arising from dense factorization kernels). To avoid the limitations due to centralized scheduling (where a host process is in charge of scheduling the work of the other processes), we have chosen a distributed scheduling strategy. In our implementation, a pool of work tasks is distributed among the processes that participate in the numerical factorization. A host process is still used to perform the analysis phase (and identify the pool of work tasks), distribute the right-hand side vector, and collect the solution. Our implementation allows this host process to participate in the computations during the factorization and solution phases. This allows the user to run the code on a single processor and avoids one processor being idle during the factorization and solution phases.

The code solves the system $\mathbf{Ax} = \mathbf{b}$ in three main steps:

1. **Analysis.** The host performs an approximate minimum degree ordering based on the symmetrized matrix pattern $\mathbf{A} + \mathbf{A}^T$ or accepts an ordering provided by the user. It also performs the subsequent symbolic factorization phase. The host then computes a mapping of the nodes of the assembly tree to the processors. The mapping, fully described in [5], is such that it keeps communication costs during factorization and solution to a minimum and it balances the memory and computation required by the processes. Using the top-down (starting from the root of the tree) algorithm described in [27], we first identify subtrees and perform a subtree-to-process mapping to balance the computational work of the subtrees between the processes (see Figure 4.1). Memory balancing criteria are then used to map the top of the tree; that is, the nodes of the tree that are not in any of the subtrees. After computing the mapping, the host sends symbolic information to the other processes. Using this information, each process estimates the work space required for its part of the factorization and solution. The estimated work space should be large enough to handle the computational tasks that were assigned to the process

at analysis time plus possible tasks that it may receive dynamically during the factorization, assuming that no excessive amount of unexpected fill-in occurs due to numerical pivoting.

2. **Factorization.** The original matrix is first preprocessed (for example, converted to arrowhead format in the case when the matrix is assembled) and distributed to the processes that will participate in the numerical factorization. Each process allocates an array for contribution blocks and factors. The numerical factorization on each frontal matrix is performed by a process determined by the analysis phase and potentially one or more other processes that are determined dynamically. The factors must be kept for the solution phase.
3. **Solution.** The right-hand side vector **b** is broadcast from the host to the other processes. They compute the solution vector **x** using the distributed factors computed during the factorization phase. The solution vector is then assembled on the host.

4.1. Sources of parallelism. We consider the condensed assembly tree of Figure 4.1, where the leaves represent subtrees of the assembly tree.

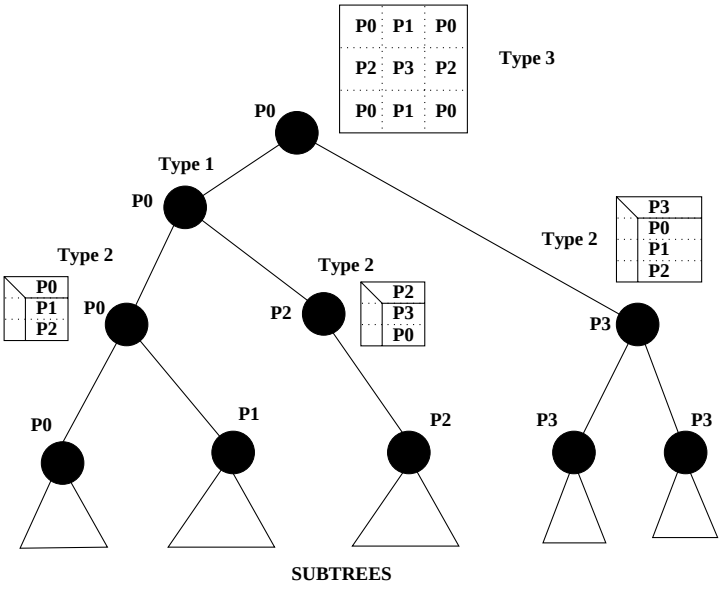


FIG. 4.1. *Distribution of the computations of a multifrontal assembly tree over the four processes* **P0**, **P1**, **P2**, *and* **P3**.

If we consider only tree parallelism, then the transfer of the contribution block from a node in the assembly tree to its parent node requires only local data movement when the nodes are assigned to the same process. Communication is required when the nodes are assigned to different processes. To reduce the amount of communication and balance the computation during the factorization and solution phases, the mapping computed during the analysis phase first assigns a subtree of the assembly tree to a single process in a way similar to the approach described in [27]. The computational cost of a subtree is approximated by the number of floating-point operations, assuming no additional numerical pivoting is performed. In general, the mapping algorithm

chooses more leaf subtrees than there are processes and, by mapping these subtrees carefully onto the processes, we achieve a good overall load balance of the computation at the bottom of the tree. However, if we exploit only tree parallelism, the speedups are very disappointing. Obviously it depends on the problem, but typically the maximum speedup is no more than 3 to 5 as illustrated in Table 5.1. This poor performance is caused by the fact that the tree parallelism decreases while going towards the root of the tree. Moreover, it has been observed (see, for example, [4]) that often more than 75% of the computations are performed in the top three levels of the assembly tree. It is thus necessary to obtain further parallelism within the large nodes near the root of the tree. The additional parallelism will be based on parallel blocked versions of the algorithms used during the factorization of the frontal matrices.

Nodes of the assembly tree that are treated by only one process will be referred to as nodes of *type* 1 and the parallelism of the assembly tree will be referred to as *type* 1 *parallelism*. Further parallelism is obtained by a one-dimensional (1D) block partitioning of the rows of the frontal matrix for nodes with a large contribution block (see Figure 4.1). Such nodes will be referred to as nodes of *type* 2 and the corresponding parallelism as *type* 2 *parallelism*. Finally, if the frontal matrix of the root node is large enough, we partition it in a two-dimensional (2D) block cyclic way. The parallel root node will be referred to as a node of *type* 3 and the corresponding parallelism as *type* 3 *parallelism*.

During the analysis, the type 2 and type 3 nodes are first selected and then the top of the tree is mapped onto the processes. This mapping tries to balance the memory used by the processes assuming that all processes will contribute to a type 2 node. The storage cost is approximated by the number of entries in the factors. In our implementation, we combine the main features of static and dynamic approaches; we use the storage estimates obtained during analysis to map the main computational tasks; the other tasks are dynamically scheduled at execution time. The dynamic scheduling is performed using the statically generated mapping as a basis. We explain this in more detail in sections 4.2 and 4.3.

4.2. Type 2 parallelism. During the analysis phase, a node is determined to be of type 2 if the number of rows in its contribution block is sufficiently large. (For nodes where the number of fully summed variables is very large, we discuss node splitting in section 10.) If a node is of type 2, one process (called the *master*) holds all the fully summed rows and performs the pivoting and the factorization on this block while other processes (called *slaves*) perform the updates on the partly summed rows. The slave processes are in charge of all the computation, both assembly and factorization steps, associated with the blocks $\mathbf{F}_{21}$ and $\mathbf{F}_{22}$ in Figure 2.1.

The slaves are determined dynamically during factorization and any process may be selected. We discuss strategies for this selection process in section 9. To be able to assemble the original matrix entries quickly into the frontal matrix of a type 2 node, we duplicate the corresponding original matrix entries (stored as arrowheads or element matrices) on all the processes before the factorization. This means that the master and slave processes of a type 2 node have immediate access to the entries that need to be assembled in the local part of the frontal matrix. This duplication of original data enables efficient dynamic scheduling of computational tasks, but requires some extra storage. This is studied in more detail in section 8. (Note that for a type 1 node, the original matrix entries need only be present on the process handling this node.)

At execution time, the master of a type 2 node first receives symbolic information

describing the structure of the contribution blocks of the son nodes in the tree. This information is sent by the (master) processes handling the sons. Based on this information, the master determines the exact structure of its frontal matrix and decides which slave processes will participate in the factorization of the node. It then sends information to the processes handling the sons to enable them to send the entries in their contribution blocks directly to the appropriate processes involved in the type 2 node. The assemblies for this node are subsequently performed in parallel. The master and slave processes then perform the elimination operations on the frontal matrix in parallel. Macro-pipelining based on a blocked factorization of the fully summed rows is used to overlap communication with computation. The efficiency of the algorithm thus depends on both the block size used to factor the fully summed rows and on the number of rows allocated to a slave process. Further details and differences between the implementations for symmetric positive definite, general symmetric, and unsymmetric matrices are described in [5]. Numerical issues will be further discussed in section 4.5.

4.3. Type 3 parallelism. At the root node, we must factorize a dense matrix and we can use standard codes for this. For scalability reasons, we use a 2D block cyclic distribution of the root node and ScaLAPACK [10] or the vendor equivalent implementation for the actual factorization.

Currently, a maximum of one root node, chosen during the analysis, is processed in parallel. The node chosen will be the largest root provided its size is larger than a computer dependent parameter (otherwise it is factorized on only one processor). One process (also called master) holds all the indices describing the structure of the root frontal matrix.

We call this root node, as determined by the analysis phase, the *estimated root node*. Before factorization, the structure of the frontal matrix of the estimated root node is statically mapped onto a 2D grid of processes. This mapping fully determines to which process an entry of the estimated root node is assigned. Hence, for the assembly of original matrix entries and contribution blocks, the processes holding this information can determine exactly the processes to which they must send data.

In the factorization phase, the original matrix entries and the part of the contribution blocks from the sons corresponding to the estimated root can be assembled as soon as they are available. The master of the root node then collects the index information for all the delayed variables (due to numerical pivoting) of its sons and builds the final structure of the root frontal matrix. This symbolic information is broadcast to all processes that participate in the factorization. The contributions corresponding to delayed variables can then be sent by the sons to the appropriate processes in the 2D grid for assembly (or the contributions can directly be assembled locally if the destination is the same process). Note that, because of the requirements of ScaLAPACK, the local part of the root node is copied since the leading dimension will change if there are delayed pivots. Furthermore, because of the lack of a parallel ScaLAPACK code for $\mathbf{LDL}^T$ factorization, we differentiate between symmetric positive definite and general symmetric matrices. On symmetric positive definite matrices, the ScaLAPACK code PDPOTRF for the $\mathbf{LL}^T$ factorization is used whereas on general symmetric matrices, the parallel factorization of the type 3 full matrix is based on the ScaLAPACK **LU** factorization routine PDGETRF. Note that it is the only place in the code where, because of numerical pivoting, we do not fully exploit the symmetry of a general symmetric matrix.

4.4. Parallel triangular solution. The solution phase is also performed in parallel and uses asynchronous communications both for the forward elimination and the back substitution. In the case of the forward elimination, the tree is processed from the leaves to the root, in a similar way to the factorization, while the back substitution requires a different algorithm that processes the tree from the root to the leaves. A pool of ready-to-be-activated tasks is used. We do not change the distribution of the factors as generated in the factorization phase. Hence, type 2 and 3 parallelism are also used in the solution phase. At the root node, we use the ScaLAPACK routine PDGETRS for general matrices and the routine PDPOTRS for symmetric positive definite matrices.

4.5. Numerical pivoting and related issues. To handle a large class of matrices, including symmetric positive definite, symmetric, unsymmetric, and rank-deficient matrices, one of the main features of our distributed memory sparse code is its ability to postpone the elimination of a numerically unstable pivot. As a result, the main data structures manipulated during factorization must be dynamic. For all classes of matrices, numerical pivoting is based on partial threshold pivoting. Pivots are selected from entries in the block $\mathbf{F}_{11}$ of Figure 2.1 which are not too small (threshold parameter) with respect to the maximum value in the corresponding fully summed row. Numerically unstable fully summed rows and columns are then sent to the father node together with the contribution block.

For unsymmetric matrices, this strategy is applied to type 1 and type 2 nodes since the fully summed rows (blocks $\mathbf{F}_{11}$ and $\mathbf{F}_{12}$) are located on the master process. On the type 3 node, the ScaLAPACK routine PDGETRF, which uses partial pivoting, is applied.

For symmetric matrices, partial threshold pivoting is applied to type 1 nodes with the additional restriction that, to preserve symmetry, numerically stable pivots must be found on the diagonal. Note that for type 1 nodes, one could also implement a two-by-two pivoting strategy [12, 24] that would reduce the number of delayed variables. This is not available in the current version of the package. For type 2 nodes, only the master holds block $\mathbf{F}_{11}$ of Figure 2.1. The eligibility of a diagonal entry as a pivot in a type 2 node is based on two numerical tests. We first check that the diagonal entry is not too small with respect to the not yet factored fully summed entries in its row. A diagonal entry that satisfies this criterion is called an *acceptable* diagonal pivot. Second, if an acceptable pivot a_{ii} is such that $|a_{ii}|$ is larger than $\sqrt{\epsilon}\,\|\mathbf{A}\|$, where ϵ is the machine precision and $\mathbf{A}$ is the original matrix, then a_{ii} is retained as a pivot and permuted symmetrically immediately after the block of already eliminated variables in $\mathbf{F}_{11}$. Otherwise the next acceptable diagonal entry is considered. The previous tests are repeated until we are left with a block of numerically unsatisfactory diagonal pivots for which the elimination is delayed to the father node.

When a rank-revealing factorization is requested (for rank detection and a null-space basis), we implicitly assume that partial threshold pivoting will have postponed all numerical problems to the dense root frontal matrix. In this context, both rank-revealing **QR** and **LU** based algorithms have been developed for matrices with relatively small rank deficiency. This ongoing work also involves Miroslav Tůma (Czech Academy of Sciences, Praha, Czech Republic) and uses work in [13, 14, 33, 34].

To conclude this section, we mention that taking into account numerical pivoting on distributed memory computers adds a significant level of complexity and many constraints to the algorithms (also discussed in [9]). This could result in somewhat less efficient and less scalable software with respect to approaches based on static

pivoting.

5. Basic performance and influence of ordering. From earlier studies (for example, [31]), we know that the ordering may significantly impact both the number of floating-point operations for the factorization and hence the uniprocessor time and also the parallel behavior of the method. Table 5.1 shows some performance results obtained using *only type* 1 parallelism. The results show that using only type 1 parallelism does not produce good speedups. The results also show (see columns "Speedup") that we usually get better parallelism with nested dissection based orderings than with minimum degree based orderings. We thus gain by using nested dissection because of a reduction in the number of floating-point operations (see Tables 3.4 and 3.5) and a better balanced assembly tree.

TABLE 5.1
Influence of the ordering on the time (in seconds) and speedup for the factorization phase, using only type 1 *parallelism, on* 32 *processors of the IBM SP*2.

Matrix	Time		Speedup	
	AMD	ND	AMD	ND
BBMAT	78.4	49.4	4.08	4.00
OILPAN	12.6	7.3	2.91	4.45
SMDOOR	33.4	25.5	3.47	4.22
BMW7ST_1	55.6	21.3	2.55	4.87

We now discuss the performance obtained by `MUMPS` on matrices in assembled format that will be used as a reference for this paper. The performance obtained on matrices provided in elemental format is discussed in section 6. In Tables 5.2 and 5.3, we show the performance of `MUMPS` using nested dissection and minimum degree orderings on the IBM SP2 and the SGI Origin 2000, respectively.

TABLE 5.2
*Impact of the ordering on the time (in seconds) for factorization on the IBM SP*2. $^{(*)}$ *estimated CPU time on one processor; — means not enough memory.*

Matrix	Ordering	Number of processors					
		$1^{(*)}$	4	8	16	24	32
BBMAT	AMD	320	276.4	68.3	47.8	44.0	39.8
	ND	198	106.4	76.7	35.2	34.6	30.9
INVEXTR1	AMD	279	—	67.9	63.2	56.5	56.0
	ND	70	25.7	17.5	16.0	13.1	12.4
MIXTANK	AMD	495	—	288.5	70.7	64.5	61.3
	ND	104	32.8	26.1	17.4	14.4	14.8
OILPAN	AMD	37	13.6	9.0	6.8	5.9	5.8
	ND	33	10.8	7.1	5.7	4.6	4.6
SMDOOR	AMD	116	155.5	24.1	16.8	16.1	13.1
	ND	108	55.7	21.6	16.8	14.7	10.5
CRANKSG1	AMD	456	—	508.3	162.4	78.4	63.3
	ND	270	228.2	102.0	42.4	39.1	31.9
CRANKSG2	AMD	939	—	—	819.6	308.5	179.7
	ND	362	—	316.6	79.7	41.7	35.7
BMW7ST_1	AMD	142	153.4	46.5	21.3	18.4	16.7
	ND	104	105.7	36.7	20.2	12.9	11.7
BMW3_2	AMD	421	—	309.8	74.2	51.0	34.2
	ND	251	—	145.3	52.4	26.2	23.0

Note that, for large problems, speedups are difficult to compute on the IBM SP2 because the data incur memory paging when run on a small number of processors.

Hence, the better performance with nested dissection orderings on a small number of processors of the IBM SP2 is due, in part, to the reduction in the memory required by each processor (since there are fewer entries in the factors). To get a better idea of the true algorithmic speedups (without memory paging effects), we give, in Table 5.2, the uniprocessor CPU time for one processor, instead of the elapsed time. The speedup over the elapsed time on one processor (not listed) can be considerable. When the memory was not large enough to run on one processor, an estimate of the Megaflop rate was used to compute the uniprocessor CPU time. The ratio of the actual performance obtained by one processor of the SGI Origin 2000 over that obtained on the IBM SP2 is used to deduce the Megaflop rate on the IBM SP2. (This estimate was also used, when necessary, to compute the speedups in Table 5.1.) On a small number of processors, there can still be a memory paging effect that may significantly increase the elapsed time.

Table 5.3 also shows the elapsed time for the solution phase; we observe that the speedups for this phase are quite respectable if one considers the relatively higher ratio of communication over computation for the parallel triangular solution.

TABLE 5.3
Impact of the ordering on the time (in seconds) for factorization and solve phases on the SGI Origin 2000.

Factorization phase							
Matrix	Ordering	Number of processors					
		1	2	4	8	16	32
LDOOR	AMD	635.0	428.5	211.4	112.7	69.2	54.8
	ND	411.9	219.2	120.9	69.6	44.7	31.6
CRANKSG2	AMD	566.1	392.2	220.0	115.9	86.4	77.4
	ND	216.9	115.9	72.0	60.3	46.9	38.9
BMW7ST_1	AMD	85.7	56.0	28.2	18.5	15.1	14.2
	ND	62.4	38.5	27.9	19.5	21.1	11.5
BMWCRA_1	AMD	663.0	396.5	238.7	141.6	110.3	76.9
	ND	306.6	182.7	80.9	52.9	41.2	35.5
BMW3_2	AMD	252.7	153.4	81.8	49.4	34.0	27.3
	ND	151.1	93.8	52.5	33.0	22.1	17.0
INLINE_1	AMD	1329.0	856.0	449.7	275.5	161.4	125.3
	ND	751.8	400.8	223.5	129.3	80.7	64.0
Solution phase							
Matrix	Ordering	Number of processors					
		1	2	4	8	16	32
LDOOR	AMD	22.1	14.9	13.4	8.6	7.4	7.9
	ND	22.3	13.9	9.9	6.7	6.2	9.1
CRANKSG2	AMD	6.8	5.8	4.4	2.9	2.4	2.3
	ND	4.3	2.7	1.8	1.5	1.1	1.8
BMW7ST_1	AMD	4.2	2.4	2.3	1.9	1.4	1.6
	ND	3.3	2.1	1.7	1.4	1.6	1.5
BMWCRA_1	AMD	11.4	7.2	6.8	3.9	2.8	2.4
	ND	8.3	4.7	2.7	2.1	1.8	2.0
BMW3_2	AMD	6.7	4.1	3.6	2.4	2.1	1.9
	ND	6.3	3.8	2.9	2.4	2.0	2.4
INLINE_1	AMD	27.1	16.5	15.7	14.2	9.7	7.9
	ND	21.2	12.0	8.9	5.8	5.0	5.4

In the remainder of this paper, we will use nested dissection based orderings, unless stated otherwise.

6. Elemental input matrix format. In this section, we discuss the main algorithmic changes to efficiently handle problems that are provided in elemental format.

We assume that the original matrix can be represented as a sum of element matrices

$$\mathbf{A} = \sum \mathbf{A}_i \ ,$$

where each $\mathbf{A}_i$ has nonzero entries only in those rows and columns that correspond to variables in the ith element. $\mathbf{A}_i$ is usually held as a dense matrix, but if the matrix $\mathbf{A}$ is symmetric, only the lower triangular part of each $\mathbf{A}_i$ is stored.

The main modifications that we had to make to our algorithms for assembled matrices to accommodate unassembled matrices lie in the analysis, the distribution of the matrix, and the assembly process. We describe them in more detail below.

In the analysis phase, we exploit the elemental format of the matrix to detect supervariables. We define a *supervariable* as a set of variables having the same list of adjacent elements. Table 6.1 shows the impact of using supervariables on the size of the graph processed by the ordering phase (AMD ordering). *Graph_size* is the length of the adjacency lists of variables/supervariables given as input to the ordering phase. Without supervariable detection, *Graph_size* is twice the number of off-diagonal entries in the corresponding assembled matrix. The work space required by the analysis phase using the AMD ordering is dominated by the space required by the ordering phase and is *Graph_size* plus an overhead that is a small multiple of the order of matrix. Table 6.1 shows that, on large graphs, compression can reduce the memory requirements of the analysis phase dramatically.

TABLE 6.1

Impact of supervariable detection on the length of the adjacency lists given to the ordering phase and on the analysis time (in seconds) (SGI Origin 2000). The time spent in the AMD ordering is in parentheses.

Matrix	*Graph_size* with supervariable detection		Time for analysis supervariable detection	
	OFF	ON	OFF	ON
THREAD.RSE	4440312	397410	2.6 (0.9)	1.2 (0.2)
M_T1.RSE	9655992	299194	4.6 (1.8)	1.5 (0.3)
X104.RSE	10059240	246950	6.4 (3.5)	1.5 (0.3)
SHIPSEC8.RSE	6538480	171428	5.7 (2.0)	2.6 (0.5)
SHIP_003.RSE	7964306	204324	7.4 (2.8)	3.2 (0.7)
SHIPSEC1.RSE	7672530	193560	6.0 (2.2)	2.6 (0.6)
SHIPSEC5.RSE	9933236	256976	10.1 (4.6)	3.9 (0.8)

Table 6.1 also shows the impact of using supervariables on the time for the complete analysis phase (including graph compression and ordering). We see that the reduction in time is not only due to the reduced time for ordering; significantly less time is also needed for building the much smaller adjacency graph of the supervariables.

The overall time spent in the assembly process for matrices in elemental format will differ from the overall time spent in the assembly process for the equivalent assembled matrix. Obviously, for the matrices in elemental format there is often significantly more data to assemble (usually about twice the number of entries than for the same matrix in assembled format). However, the assembly process of matrices in elemental format might be performed more efficiently than the assembly process of assembled matrices. First, because we potentially assemble at once a larger and more regular structure (a full matrix). Second, because most input data will be assembled at or near leaf nodes in the assembly tree. This has two consequences. The assemblies are performed in a more distributed way and most assemblies of original element matrices

are done at type 1 nodes. (Hence, less duplication of original matrix data is necessary.) A more detailed analysis of the duplication issues linked to matrices in elemental format will be addressed in section 8. In experiments (not presented here), we have observed that, despite the differences in the assembly process, the performance of MUMPS for assembled and unassembled problems is very similar, provided the same ordering is used. The reason for this is that the extra amount of assemblies of original data for unassembled problems is relatively small compared to the total number of flops.

The experimental results in Table 6.2, obtained on the SGI Origin 2000, show the good behavior of the code for the factorization phase on our set of unassembled matrices.

TABLE 6.2
Time (in seconds) for factorization of the unassembled matrices on the SGI Origin 2000. *MFR ordering from Det Norske Veritas is used.*

Matrix	Number of processors					
	1	2	4	8	16	32
THREAD.RSE	186	120	69	46	37	32
M_T1.RSE	92	56	30	18	17	13
X104.RSE	56	34	20	16	16	13
SHIPSEC8.RSE	187	127	68	36	30	23
SHIP_003.RSE	392	242	156	120	92	73
SHIPSEC1.RSE	174	128	65	36	27	24
SHIPSEC5.RSE	281	176	114	63	43	36

7. Distributed assembled matrix. The distribution of the input matrix over the available processors is the main preprocessing step in the numerical factorization phase. During this step, the input matrix is organized into arrowhead format and is distributed according to the mapping provided by the analysis phase. In the symmetric case, the first arrowhead of each frontal matrix is also sorted to enable efficient assembly [5]. If the assembled matrix is initially held centrally on the host, we have observed that the time to distribute the real entries of the original matrix can sometimes be comparable to the time to perform the actual factorization. For example, for the matrix OILPAN, the time to distribute the input matrix on 16 processors of the IBM SP2 is on average 6 seconds whereas the time to factorize the matrix is 6.8 seconds (using the AMD ordering; see Table 5.2). Clearly, for larger problems where more arithmetic is required for the actual factorization, the time for factorization will dominate the time for redistribution.

With a distributed input matrix format, we can expect to reduce the time for the redistribution phase, because we can parallelize the reformatting and sorting tasks and we can use asynchronous all-to-all (instead of one-to-all) communications. Furthermore, we can expect to solve larger problems since storing the complete matrix on one processor limits the size of the problem that can be solved on a distributed memory computer. Thus, to improve both the memory and the time scalability of our approach, we should allow the input matrix to be distributed.

Based on the static mapping of tasks to processes that is computed during the analysis phase, one can a priori distribute the input data so that no further remapping is required at the beginning of the factorization. This distribution, referred to as the MUMPS *distribution*, will limit the communication to duplications of the original matrix corresponding to type 2 nodes (further studied in section 8).

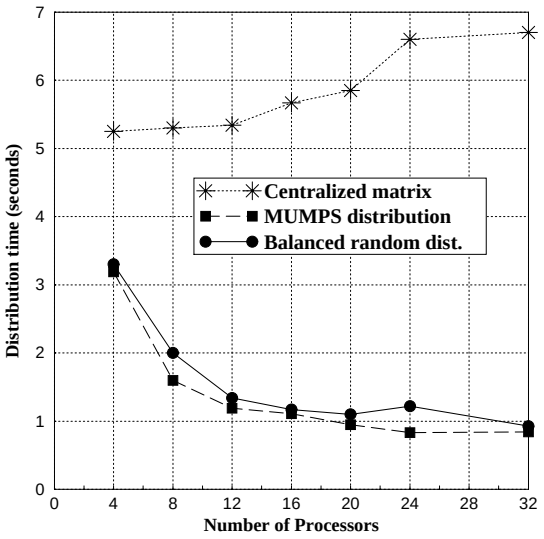


FIG. 7.1. *Impact of the initial distribution for matrix* OILPAN *on the time for redistribution on the IBM SP*2.

To show the influence of the initial matrix distribution on the time for redistribution, we compare, in Figure 7.1, three ways for providing the input matrix:

1. Centralized mapping: the input matrix is held on one process (the host).
2. MUMPS distribution: the input matrix is distributed over the processes according to the static mapping that is computed during the analysis phase.
3. Balanced random distribution: the input matrix is uniformly distributed over the processes in a random manner that has no correlation to the mapping computed during the analysis phase.

The fastest approach should clearly be the MUMPS distribution because the amount of communication is minimized and all-to-all communications can be used. This approach is, however, too restrictive for the user who has to build the input matrix according to the static mapping returned by MUMPS in the analysis phase. In this context, we want, with the balanced random distribution, to analyze the extra cost of not respecting the MUMPS mapping while still having the input matrix uniformly distributed.

Figure 7.1 clearly shows the benefit of using asynchronous all-to-all communications (required by the MUMPS and random distributions) compared to using one-to-all communications (for the centralized mapping). It is even more interesting to observe that distributing the input matrix according to the MUMPS distribution does not significantly reduce the time for the redistribution. We attribute this to the good overlapping of communication with computation (mainly data reformatting and sorting) in our redistribution algorithm.

8. Memory scalability issues. In this section, we study the memory requirements and memory scalability of our algorithms.

Figure 8.1 illustrates how MUMPS balances the memory load over the processors.

The figure shows, for two matrices, the maximum memory required on a processor and the average over all processors, as a function of the number of processors. We observe that, for varying numbers of processors, these values are quite similar.

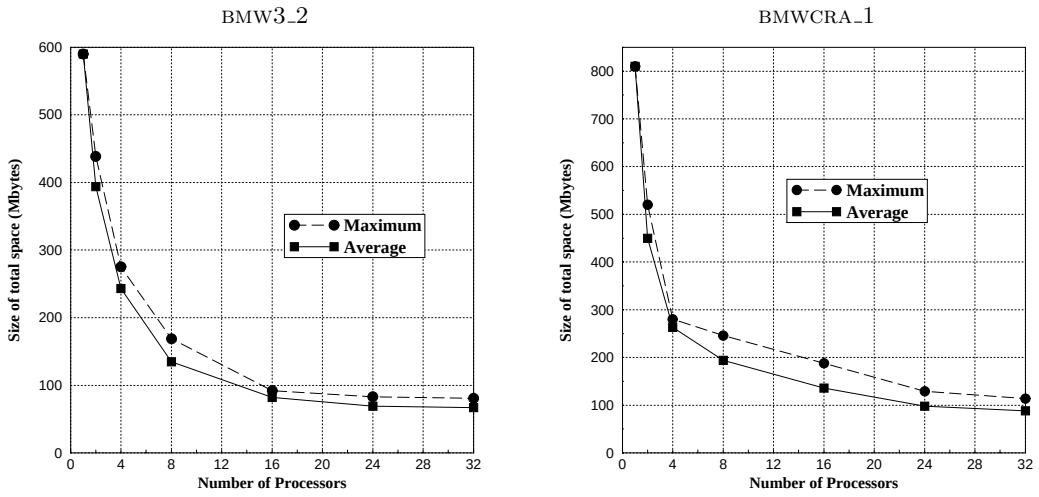


FIG. 8.1. *Total memory (maximum and average) requirement per processor during factorization (ND ordering).*

Table 8.1 shows the *average* size per processor of the main components of the work space used during the factorization of the matrix BMW3_2. These components are

- FACTORS: the space reserved for the factors; a processor does not know after the analysis phase in which type 2 nodes it will participate, and therefore it reserves enough space to be able to participate in all type 2 nodes.
- STORE_AREA: the space used to store both the contribution blocks and the factors.
- INITIAL MATRIX: the space required to store the initial matrix in arrowhead format.
- COMMUNICATION BUFFERS: the space allocated for both send and receive buffers.
- OTHER: the size of all the remaining work space allocated per processor.
- TOTAL: the total memory required per processor.

The lines *ideal* in Table 8.1 are obtained by dividing the memory requirement on one processor by the number of processors. By comparing the actual and ideal numbers, we get an idea of how `MUMPS` scales in terms of memory for some of the components.

We see that, even if the total memory (sum of all the local work spaces) increases, the average memory required per processor significantly decreases up to 16 processors. We also see that the size for the FACTORS and the STORE_AREA are much larger than ideal. Part of this difference is due to parallelism and is unavoidable. Another part, however, is due to an overestimation of the space required. The main reason for this is that the mapping of the type 2 nodes on the processors is not known at analysis and each processor can potentially participate in the elimination of any type 2 node. Therefore, each processor allocates enough space to be able to participate in all type 2 nodes. The work space that is actually used is smaller and, on a large number of processors, we could reduce the estimate for both the FACTORS and the STORE_AREA.

TABLE 8.1
Analysis of the memory used during factorization of matrix BMW3_2 *(ND ordering). All sizes are in MBytes per processor.*

Number of processors	1	2	4	8	16	24	32
FACTORS	423	211	107	58	35	31	31
ideal	—	211	106	53	26	18	13
STORE_AREA	502	294	172	92	51	39	38
ideal	—	251	126	63	31	21	16
INITIAL MATRIX	69	34.5	17.3	8.9	5.0	4.0	3.5
ideal	—	34.5	17.3	8.6	4.3	2.9	2.2
COMMUNICATION BUFFERS	0	45	34	14	6	6	5
OTHER	20	20	20	20	20	20	20
TOTAL	590	394	243	135	82	69	67
ideal	—	295	147	74	37	25	18

For example, we have successfully factorized matrix BMW3_2 on 32 processors with a STORE_AREA that is 20% smaller than reported in Table 8.1.

The average work space used by the communication buffers also significantly decreases up to 16 processors. This is mainly due to type 2 node parallelism where contribution blocks are split among processors until a minimum granularity is reached. Therefore, when we increase the number of processors, we decrease (until reaching this minimum granularity) the size of the contribution blocks sent between processors. Note that on larger problems, the average size per processor of the communication buffers will continue to decrease for a larger number of processors. The line OTHER does not scale at all since it corresponds to data arrays of size $O(n)$ that are allocated on each process. We see that this space significantly affects the difference between TOTAL and *ideal*, especially for larger numbers of processors. Note that the relative influence of this fixed size area will be smaller on large matrices from three-dimensional simulations. Actually, this $O(n)$ working area per process could be reduced to a working area of size $O(|Subtree|)$, where $|Subtree|$ denotes the number of nodes in the subtrees assigned to the process. This has not been considered in our package but could be considered to improve the memory scalability of the approach.

The imperfect scalability of the initial matrix storage comes from the duplication of the original matrix data that is linked to type 2 nodes in the assembly tree. We will study this in more detail in the remainder of this section. We want to stress, however, that from a user point of view, all numbers reported in this context should be related to the total memory used by the `MUMPS` package which is usually dominated, on large problems, by the size of the STORE_AREA.

An alternative to the duplication of data related to type 2 nodes would be to allocate the original data associated with a frontal matrix to only the master process responsible for the type 2 node. During the assembly process, the master process would then be in charge of redistributing the original data to the slave processes. This strategy introduces extra communication costs during the assembly of a type 2 node and thus has not been chosen. With the approach based on duplication, the process responsible for a type 2 node has all the flexibility to choose collaborating processes dynamically since this will not involve any data migration of the original matrix. However, the extra cost of this strategy is that it requires partial duplication of the original matrix.

All the nodes that do not belong to a subtree are candidates for type 2 selection. Since the mapping algorithm naturally computes more (but smaller) subtrees for

TABLE 8.2
Amount of duplication due to type 2 nodes. "Total entries" is the sum of the number of original matrix entries over all processors ($\times 10^3$). The number of type 2 nodes is also given.

Matrix		Number of processors					
		1	2	4	8	12	16
OILPAN	Type 2 nodes	0	4	7	10	17	22
	Total entries	1835	1845	1888	2011	2235	2521
BMW7ST_1	Type 2 nodes	0	4	7	9	13	21
	Total entries	3740	3759	3844	4031	4308	4793
BMW3_2	Type 2 nodes	0	1	3	13	14	21
	Total entries	5758	5767	5832	6239	6548	7120
THREAD.RSA	Type 2 nodes	0	3	8	12	23	25
	Total entries	2250	2342	2901	4237	6561	8343
THREAD.RSE	Type 2 nodes	0	2	8	12	15	25
	Total entries	3719	3719	3719	3719	3719	3719
SHIPSEC1.RSA	Type 2 nodes	0	0	4	11	19	21
	Total entries	3977	3977	4058	4400	4936	5337
SHIPSEC1.RSE	Type 2 nodes	0	1	4	13	19	27
	Total entries	8618	8618	8618	8627	8636	8655

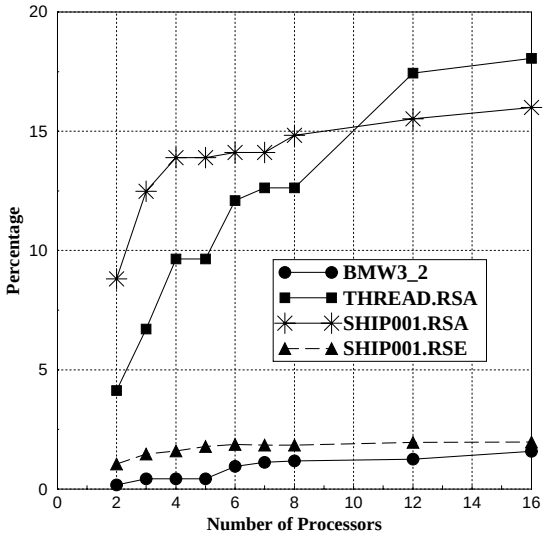


FIG. 8.2. *Percentage of entries in the original matrix that are duplicated on all processors due to type 2 nodes.*

larger numbers of processors, the potential (and actual) number of type 2 nodes also increases. This implies that more data of the original matrix will be duplicated when the number of processors increases. The influence of the number of processors on the amount of duplication is shown in Table 8.2. On a representative subset of our test problems, we show the total number of type 2 nodes and the sum over all processors of the number of original matrix entries and duplicates. If there is only one processor,

type 2 nodes are not used and no data is duplicated. Figure 8.2 shows, for four of our matrices, the number of original matrix entries that are duplicated on all processors, relative to the total number of entries in the original matrix.

Since the original data for unassembled matrices are in general assembled earlier in the assembly tree than the data for the same matrix in assembled format, the number of duplications is often relatively much smaller with unassembled matrices than with assembled matrices. Matrix THREAD.RSE (in elemental format) is an extreme example since, even on 16 processors, type 2 node parallelism does not require any duplication (see Table 8.2).

To conclude this section, we point out that, apart from the $O(n)$ working arrays, the code scales well in terms of memory usage. On (virtual) shared memory computers, the total memory (sum of local workspaces of all the processors) required by MUMPS can sometimes be prohibitive. Therefore, we are currently investigating how we can reduce the current overestimates of the local STORE_AREAs to reduce the total memory required. Limiting the dynamic scheduling of a type 2 node (and corresponding data duplication) to a subset of processors might be a solution for this.

9. Dynamic scheduling strategies. To avoid the drawback of centralized scheduling on distributed memory computers, we have chosen to implement distributed dynamic scheduling strategies. Recall that type 1 nodes are statically mapped to processes at analysis time and that only type 2 tasks, which represent a large part of the computations and of the parallelism of the method, are involved in the dynamic scheduling strategy.

To be able to choose dynamically the processes that will collaborate in the processing of a type 2 node, we have designed a two-phase assembly process. Let $Inode$ be a node of type 2 and let $Pmaster$ be the process to which $Inode$ is initially mapped. In the first phase, the (master) processes, to which the sons of $Inode$ are mapped, send symbolic data (integer lists) to $Pmaster$. When the structure of the frontal matrix is determined, $Pmaster$ decides a partitioning of the frontal matrix and chooses the slave processes. It is during this phase that $Pmaster$ will collect information concerning the load of the other processors to help in its decision process. The slave processes are informed that a new task has been allocated to them. $Pmaster$ then sends the description of the distribution of the frontal matrix to all collaborative processes of all sons of $Inode$ so that they can send their contribution blocks (numerical values) in pieces directly to the correct processes involved in the computation of $Inode$. The assembly process is thus fully parallelized and the maximum size of a message sent between processes is reduced (see section 8).

A pool of tasks private to each process is used to implement dynamic scheduling. All tasks ready to be activated on a given process are stored in the pool of tasks local to the process. Each pool of tasks is initialized with the assembly process associated with the local leaf nodes. Each process executes the following algorithm:

ALGORITHM 1.

while (*not all nodes processed*)
  if *local pool empty* **then**
    blocking receive for a message; process the message
  elseif *message available* **then**
    receive and process message

else
extract work from the pool, and process it
endif
end while

Note that the task corresponding to the processing of a message is immediately executed without going to the pool. A task ready to be activated can, however, result from the processing of a message and will be added at the head of the pool. The algorithm gives priority to message reception. The main reasons for this choice are first that the message received might be a source of additional work and parallelism and second, the sending process might be blocked because its send buffer is full (see [5]). In the actual implementation, we use the routine MPI_IPROBE to check whether a message is available.

TABLE 9.1
Comparison of cyclic and flops-based schedulings. Time (in seconds) for factorization on the IBM SP2 (ND ordering).

Matrix & scheduling	Number of processors 16	20	24	28	32
CRANKSG2					
cyclic	79.7	47.9	41.7	41.3	35.7
flops-based	61.1	45.6	41.9	41.7	40.4
BMW3_2					
cyclic	52.4	31.8	26.2	29.2	23.0
flops-based	29.4	27.8	25.1	25.3	22.6

We implemented two scheduling strategies. In the first strategy, referred to as *cyclic scheduling*, the master of a type 2 node does not take into account the load on the other processors and performs a simple cyclic mapping of the tasks to the processors. In the second strategy, referred to as *(dynamic) flops-based scheduling*, the master process uses information on the load of the other processors to allocate type 2 tasks to the least loaded processes. The load of a processor is defined here as the amount of work (flops) associated with all the active or ready-to-be-activated tasks. Each process is in charge of maintaining local information associated with its current load. With a simple remote memory access procedure, using, for example, the one-sided communication routine MPI_GET included in MPI-2, each process has access to the load of all other processes when necessary. However, MPI-2 is not available on our target computers. To overcome this, we have designed a module based only on symmetric communication tools (MPI asynchronous send and receive). Each process is in charge of both updating and broadcasting its local load. To control the frequency of these broadcasts, an updated load is broadcast only if it is significantly different from the last load broadcast.

When the initial static mapping does not balance the work well, we might expect that the dynamic flops-based scheduling will improve the performance with respect to cyclic scheduling. Tables 9.1 and 9.2 show that significant performance gains can be obtained by using dynamic flops-based scheduling. On more than 24 processors, the gains are less significant because the number of parallel tasks is not enough to keep all the processes busy. We also expect that this feature will improve the behavior of the parallel algorithm on a multiuser distributed memory computer.

Another possible use of dynamic scheduling is to improve the memory usage. We have seen, in section 8, that the size of the STORE_AREA is overestimated. Dynamic scheduling based on memory load, instead of computational load, could be used to

TABLE 9.2
Comparison of cyclic and flops-based schedulings. Time (in seconds) for factorization on the SGI Origin 2000 *(MFR ordering).*

Matrix & scheduling	Number of processors 4	8	16
SHIPSEC8.RSE			
cyclic	68.3	36.3	29.9
flops-based	65.0	35.0	25.1
SHIP_003.RSE			
cyclic	156.1	119.9	91.9
flops-based	140.3	110.2	83.8
SHIPSEC5.RSE			
cyclic	113.5	63.1	42.8
flops-based	99.9	61.3	37.0

address this issue. Type 2 tasks can then be mapped to the least loaded processor (in terms of memory used in the STORE_AREA).

10. Splitting nodes of the assembly tree. During the processing of a parallel type 2 node, both in the symmetric and the unsymmetric case, the factorization of the pivot rows is performed by a single processor. Other processors can then help in the update of the rows of the contribution block using a 1D decomposition (as shown in section 4). The elimination of the fully summed rows can represent a potential bottleneck for scalability, especially for frontal matrices with a large fully summed block near the root of the tree. To overcome this problem, we subdivide nodes with large fully summed blocks, as illustrated in Figure 10.1.

In this figure, we consider an initial node of size NFRONT with NPIV pivots. We replace this node by a son node of size NFRONT with NPIV_{son} pivots, and a father node of size $\text{NFRONT} - \text{NPIV}_{son}$, with $\text{NPIV}_{father} = \text{NPIV} - \text{NPIV}_{son}$ pivots. Note that by splitting a node, we increase the number of operations for factorization, because we add assembly operations. Nevertheless, we expect to benefit from splitting because we increase parallelism.

We experimented with a simple algorithm that postprocesses the tree after the symbolic factorization. The algorithm considers only nodes near the root of the tree. Splitting large nodes far from the root, where sufficient tree parallelism can already be exploited, would only lead to additional assembly and communication costs. A node is considered for splitting only if its distance to the root, that is, the number of edges between the root and the node, is less than or equal to $d_{max} = \log_2(\text{NPROCS} - 1)$.

Let $Inode$ be a node in the tree, and $d(Inode)$ the distance of $Inode$ to the root. For all nodes $Inode$ such that $d(Inode) \leq d_{max}$, we apply the Algorithm 2.

ALGORITHM 2. *Splitting of a node*
if $\text{NFRONT} - \text{NPIV}/2$ *is large enough* **then**
1. *Compute* W_{master} = *number of flops performed by the master of Inode.*
2. *Compute* W_{slave} = *number of flops performed by a slave, assuming that* $\text{NPROCS} - 1$ *slaves can participate.*
3. **if** $W_{master} > W_{slave} \cdot (1 + \frac{p \cdot max(1, d(Inode) - 1)}{100})$ **then**
   3.1. *Split Inode in nodes son and father so that* $\text{NPIV}_{son} = \text{NPIV}_{father} = \text{NPIV}/2$.
   3.2. *Apply Algorithm* 2 *recursively to nodes son and father.*
   endif
endif

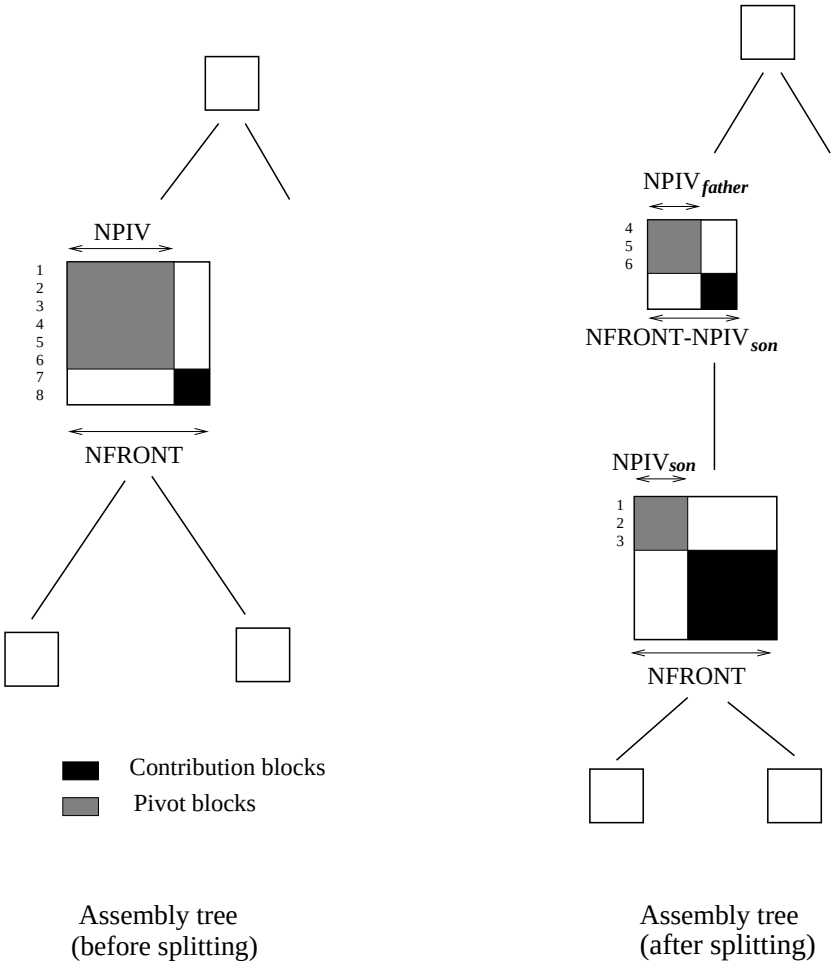


FIG. 10.1. *Tree before and after the subdivision of a frontal matrix with a large pivot block.*

Algorithm 2 is applied to a node only when NFRONT – NPIV/2 is large enough because we want to make sure that the son of the split node is of type 2. (The size of the contribution block of the son will be NFRONT – NPIV_{son}.) A node is split only when the amount of work for the master (W_{master}) is large relative to the amount of work for a slave (W_{slave}). To reduce the amount of splitting further away from the root, we add, at step 3 of the algorithm, a relative factor to W_{slave}. This factor depends on a machine dependent parameter p, $p > 0$, and increases with the distance of the node to the root. Parameter p allows us to control the general amount of splitting. Finally, because the algorithm is recursive, we may divide the initial node into more than two new nodes.

The effect of splitting is illustrated in Table 10.1 on both the symmetric matrix CRANKSG2 and the unsymmetric matrix INVEXTR1. *Ncut* corresponds to the number of type 2 nodes cut. A value $p = \infty$ is used to indicate no splitting. Flops-based dynamic scheduling is used for all runs in this section. The best time obtained for a given number of processors is indicated in boldface. We see that significant performance improvements (of up to 40% reduction in time) can be obtained by using

TABLE 10.1

Time (in seconds) for factorization and number of nodes cut (Ncut) for different values of parameter p on the IBM SP2. Nested dissection ordering and flops-based dynamic scheduling are used.

CRANKSG2						
p		Number of processors				
		16	20	24	28	32
∞	Time	61.1	45.6	41.9	41.7	40.4
	Ncut	0	0	0	0	0
200	Time	37.9	31.4	30.4	29.5	**25.4**
	Ncut	6	7	9	9	12
150	Time	41.8	**31.3**	31.0	28.9	27.2
	Ncut	7	9	10	12	13
100	Time	39.8	32.3	**28.4**	**28.6**	26.7
	Ncut	9	11	13	14	15
50	Time	**36.7**	33.6	31.4	29.6	27.4
	Ncut	12	13	16	17	21
10	Time	40.8	32.5	29.5	29.8	26.0
	Ncut	16	17	21	28	32

INVEXTR1						
p		Number of processors				
		4	8	16	24	32
∞	Time	25.9	16.7	14.6	13.5	14.6
	Ncut	0	0	0	0	0
200	Time	25.5	16.7	**13.4**	**12.1**	**12.4**
	Ncut	0	1	3	6	12
150	Time	**24.9**	16.3	13.5	13.4	12.4
	Ncut	1	1	4	11	9
100	Time	24.9	**16.2**	13.7	13.1	13.6
	Ncut	1	2	6	19	24
50	Time	24.9	17.0	13.5	13.6	16.6
	Ncut	1	3	14	25	35
10	Time	24.9	17.5	13.4	14.5	15.8
	Ncut	2	6	17	27	33

node splitting. The best timings are generally obtained for relatively large values of p. More splitting occurs for smaller values of p, but the corresponding times increase only slightly.

11. Summary. Tables 11.1 and 11.2 show results obtained with Version 4 of MUMPS using both flops-based dynamic scheduling and node splitting. Default values for the parameters controlling the efficiency of the package have been used and therefore the timings do not always correspond to the fastest possible execution time. A comparison with results presented in Tables 5.2, 5.3, and 6.2 summarizes well the benefits coming from the work presented in sections 9 and 10.

The largest problem we have solved to date is an automotive crankshaft model from MSC.Software. The problem matrix is symmetric positive definite and has order 943695 with more than 39 million entries. The number of entries in the factors is 1.4×10^9 and the number of operations during factorization is 5.9×10^{12}. On one processor of the SGI Origin 2000, the factorization phase required 8.9 hours and on two (nondedicated) processors 6.2 hours were required. Because of the total amount of memory estimated and reserved by MUMPS, we could not solve it on more than two processors. This issue will need to be addressed to improve the scalability on globally addressable memory computers and further analysis will be performed on purely distributed memory computers with a larger number of processors. Possible

TABLE 11.1
Time (in seconds) for factorization using Version 4 *of* MUMPS *with default options on the IBM SP*2*. ND ordering is used.* $^{(*)}$*: uniprocessor CPU or estimated CPU time; — means excessive swapping or not enough memory.*

Matrix	Number of processors					
	$1^{(*)}$	4	8	16	24	32
BBMAT	198	106	85	35	33	31
INVEXTR1	70	25	16	14	13	12
MIXTANK	104	31	22	16	15	15
OILPAN	33	11	8	5	5	5
SMDOOR	108	56	22	13	13	11
CRANKSG1	270	185	92	27	26	21
CRANKSG2	362	—	—	42	31	27
BMW7ST_1	104	—	30	14	12	11
BMW3_2	251	—	—	24	24	20

TABLE 11.2
Time (in seconds) for factorization using Version 4 *of* MUMPS *with default options on the SGI Origin* 2000*. ND ordering is used.*

Matrix	Number of processors					
	1	2	4	8	16	32
LDOOR	412	228	121	68	39	31
CRANKSG2	217	112	66	46	29	23
BMW7ST_1	62	36	25	12	10	8
BMWCRA_1	307	178	82	58	36	27
BMW3_2	151	96	53	33	18	15
INLINE_1	752	406	225	127	76	55
THREAD.RSE	186	125	70	38	24	22
M_T1.RSE	92	56	31	19	13	9
X104.RSE	56	34	19	12	11	10
SHIPSEC8.RSE	187	119	64	35	27	23
SHIP_003.RSE	392	237	124	108	51	43
SHIPSEC1.RSE	174	125	63	39	25	20
SHIPSEC5.RSE	281	181	103	62	37	29

solutions to this have been mentioned in the paper (limited dynamic scheduling and/or memory based dynamic scheduling) and will be developed in the future.

Acknowledgments. We are grateful to Jennifer Scott and John Reid for their comments on an early version of this paper. We want to thank the referees for their helpful comments and suggestions which have improved the presentation of the material.

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, M. J. DAYDÉ, I. S. DUFF, AND P. MORÈRE. *Linear algebra calculations on a virtual shared memory computer*, Int. J. High Speed Computing, 7 (1995), pp. 21–43.
[3] P. R. AMESTOY AND I. S. DUFF, *Vectorization of a multiprocessor multifrontal code*, Int. J. Supercomputer Appl., 3 (1989), pp. 41–59.
[4] P. R. AMESTOY AND I. S. DUFF, *Memory management issues in sparse multifrontal methods on multiprocessors*, Int. J. Supercomputer Appl., 7 (1993), pp. 64–82.
[5] P. R. AMESTOY, I. S. DUFF, AND J.-Y. L'EXCELLENT, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 501–520.

[6] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and P. Plecháč, *PARASOL. An integrated programming environment for parallel sparse matrix solvers*, in High-Performance Computing, R. J. Allan, M. F. Guest, A. D. Simpson, D. S. Henty, and D. A. Nicole, eds., Kluwer Academic/Plenum Publishers, New York, 1999, pp. 79–90.

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

[8] C. Ashcraft, *The fan-both family of column-based distributed Cholesky factorisation algorithms*, in Graph Theory and Sparse Matrix Computations, J. R. Gilbert and J. W. H. Liu, eds., Springer-Verlag, New York, 1993, pp. 159–190.

[9] 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.

[10] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, 1997.

[11] T. Blank, R. Lucas, and J. Tiemann, *A parallel solution method for large sparse systems of equations*, IEEE Trans. Comput., 6 (1989), pp. 981–991.

[12] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comput., 31 (1977), pp. 162–179.

[13] T. F. Chan, *Rank revealing QR-factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[14] T. F. Chan and P. C. Hansen, *Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 519–530.

[15] J. M. Conroy, S. G. Kratzer, and R. F. Lucas, *Data-parallel sparse matrix factorization*, in Proceedings Fifth SIAM Conference on Applied Linear Algebra, Snowbird, UT, 1994, J. G. Lewis, ed., SIAM, Philadelphia, 1994, pp. 377–381.

[16] J. W. Demmel and X. S. Li, *Making sparse Gaussian elimination scalable by static pivoting*, in Proceedings of SC98, Orlando, FL, IEEE/ACM, 1998.

[17] J. Dongarra and R. C. Whaley, *LAPACK Working Note* 94: *A Users' Guide to the BLACS* v1.0, Technical Report UT-CS-95-281, University of Tennessee, Knoxville, TN, 1995; updated 1997.

[18] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, *Algorithm* 679. *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[19] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, *Algorithm* 679. *A set of level* 3 *basic linear algebra subprograms: Model implementation and test programs*, ACM Trans. Math. Software, 16 (1990), pp. 18–28.

[20] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1986.

[21] I. S. Duff, R. G. Grimes, and J. G. Lewis, *The Rutherford-Boeing Sparse Matrix Collection*, Technical Report RAL-TR-97-031, Rutherford Appleton Laboratory, Oxfordshire, England, 1997. Also Technical Report ISSTECH-97-017 from Boeing Information & Support Services and Report TR/PA/97/36 from CERFACS, Toulouse, France.

[22] 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.

[23] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.

[24] 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.

[25] V. Espirat, *Développement d'une approche multifrontale pour machines a mémoire distribuée et réseau hétérogène de stations de travail*, Rapport de stage 3ieme Année, Trainee graduate report, ENSEEIHT-IRIT, 1996.

[26] C. Fu, X. Jiao, and T. Yang, *Efficient sparse LU factorization with partial pivoting on distributed memory architectures*, IEEE Trans. Parallel Distributed Systems, 9 (1998), pp. 109–125.

[27] A. Geist and E. Ng, *Task scheduling for parallel sparse Cholesky factorization*, Internat. J. Parallel Programming, 18 (1989), pp. 291–314.

[28] A. George, M. T. Heath, J. Liu, and E. Ng, *Sparse Cholesky factorization on a local-memory multiprocessor*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 327–340.

[29] A. Gupta, G. Karypis, and V. Kumar, *Highly scalable parallel algorithms for sparse matrix factorization*, IEEE Trans. Parallel Distributed Systems, 8 (1997), pp. 502–520.

[30] M. T. Heath, E. Ng, and B. W. Peyton, *Parallel algorithms for sparse linear systems*, SIAM

Rev., 33 (1991), pp. 420–460.
[31] B. Hendrickson and E. Rothberg, *Improving the run time and quality of nested dissection ordering*, SIAM J. Sci. Comput., 20 (1998), pp. 468–489.
[32] P. Henon, P. Ramet, and J. Roman, *A mapping and scheduling algorithm for parallel sparse fan-in numerical factorization*, in EuroPar'99 Parallel Processing, Lecture Notes in Comput. Sci. 1685, Springer, Berlin, 1999, pp. 1059–1067.
[33] T.-M. Hwang, W.-W. Lin, and D. Pierce, *Improved bounds for rank revealing LU factorizations*, Linear Algebra Appl., 261 (1997), pp. 172–186.
[34] T.-M. Hwang, W.-W. Lin, and E. Yang, *Rank revealing LU factorizations*, Linear Algebra Appl., 175 (1992), pp. 213–232.
[35] G. Karypis and V. Kumar, MeTiS*, A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version* 4.0, University of Minnesota, Minneapolis, MN, 1998.
[36] J. Koster, *On the Parallel Solution and the Reordering of Unsymmetric Sparse Linear Systems*, Ph.D. thesis, Institut National Polytechnique de Toulouse, France, 1997. Available as CERFACS report TH/PA/97/51.
[37] 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, TX, 1999.
[38] F. Pellegrini, Scotch 3.1 *User's Guide*, Technical Report 1137-96, LaBRI, Université Bordeaux I, Bordeaux, France, 1996.
[39] F. Pellegrini, J. Roman, and P. R. Amestoy, *Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering*, in Proceedings of Irregular'99, San Juan, PR, Lecture Notes in Comput. Sci. 1586, Springer, Berlin, 1999, pp. 986–995.
[40] E. Rothberg, *Efficient sparse Cholesky factorization on distributed-memory multiprocessors*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, UT, 1994, J. G. Lewis, ed., SIAM, Philadelphia, 1994, p. 141.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 42–56
© 2001 Society for Industrial and Applied Mathematics

ON WEIGHTED LINEAR LEAST-SQUARES PROBLEMS RELATED TO INTERIOR METHODS FOR CONVEX QUADRATIC PROGRAMMING*

ANDERS FORSGREN† AND GÖRAN SPORRE†

Abstract. It is known that the norm of the solution to a weighted linear least-squares problem is uniformly bounded for the set of diagonally dominant symmetric positive definite weight matrices. This result is extended to weight matrices that are nonnegative linear combinations of symmetric positive semidefinite matrices. Further, results are given concerning the strong connection between the boundedness of weighted projection onto a subspace and the projection onto its complementary subspace using the inverse weight matrix. In particular, explicit bounds are given for the Euclidean norm of the projections. These results are applied to the Newton equations arising in a primal-dual interior method for convex quadratic programming and boundedness is shown for the corresponding projection operator.

Key words. unconstrained linear least-squares problem, weighted least-squares problem, quadratic programming, interior method

AMS subject classifications. 65F20, 65F35, 65K05

PII. S0895479800372298

1. Introduction.

In this paper we study certain properties of the weighted linear least-squares problem

$$\underset{\pi\in\mathbb{R}^m}{\text{minimize}} \qquad \|W^{1/2}(A^T\pi - g)\|_2^2, \tag{1.1}$$

where A is an $m \times n$ matrix of full row rank and W is a positive definite symmetric $n \times n$ matrix whose matrix square root is denoted by $W^{1/2}$. (See, e.g., Golub and Van Loan [14, p. 149] for a discussion on matrix square roots.) Linear least-squares problems are fundamental within linear algebra; see, e.g., Lawson and Hanson [20], Golub and Van Loan [14, Chapter 5] and Gill, Murray, and Wright [12, Chapter 6]. An individual problem of the form (1.1) can be converted to an unweighted problem by substituting $\widetilde{A} = AW^{1/2}$ and $\widetilde{g} = W^{1/2}g$. However, our interest is in *sequences* of weighted problems, where the weight matrix W changes and A is constant. The present paper is a continuation of the paper by Forsgren [10], in which W is assumed to be diagonally dominant. Our concern is when the weight matrix is of the form

$$W = (H + D)^{-1}, \tag{1.2}$$

where H is a constant positive semidefinite symmetric matrix and D is an arbitrary positive definite diagonal matrix. Such matrices arise in interior methods for convex quadratic programming. See section 1.1 below for a brief motivation.

The solution of (1.1) is given by the *normal equations*

$$AWA^T\pi = AWg \tag{1.3}$$

*Received by the editors May 18, 2000; accepted for publication (in revised form) by M. Overton November 29, 2000; published electronically April 18, 2001. This research was supported by the Swedish Natural Science Research Council (NFR).

http://www.siam.org/journals/simax/23-1/37229.html

†Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, SE-100 44 Stockholm, Sweden (anders.forsgren@math.kth.se, goran.sporre@math.kth.se).

or alternatively as the solution to the *augmented system* (or *Karush–Kuhn–Tucker* (*KKT*) *system*)

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r \\ \pi \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{1.4}$$

where $M = W^{-1}$. In some situations, we will prefer the KKT form (1.4), since we are interested in the case when M is a positive semidefinite symmetric and singular matrix. In this situation, W^{-1} and (1.3) are not defined, but (1.4) is well defined. This would be the case, for example, in an equality-constrained weighted linear least-squares problem; see, e.g., Lawson and Hanson [20, Chapter 22]. For convenience, we will mainly use the form (1.3).

If $M = W^{-1}$, then, mathematically, (1.3) and (1.4) are equivalent. From a computational point of view, this need not be the case. There is a large number of papers giving reasons for solving systems of one type or the other, starting with Bartels, Golub, and Saunders [1], followed by, e.g., Duff et al. [9], Björck [4], Gulliksson and Wedin [17], Wright [29, 31], Björck and Paige [5], Vavasis [26], Forsgren, Gill, and Shinnerl [11], and Gill, Saunders, and Shinnerl [13]. The focus of the present paper is linear algebra, and we will not discuss these important computational aspects.

If A has full row rank and if $\mathcal{W}_+$ is defined as the set of $n \times n$ positive definite symmetric matrices, then for any $W \in \mathcal{W}_+$, the unique solution of (1.1) is given by

$$\pi = (AWA^T)^{-1}AWg. \tag{1.5}$$

In a number of applications, it is of interest to know if the solution remains in a compact set as the weight matrix changes, i.e., the question is whether

$$\sup_{W \in \mathcal{W}} \|(AWA^T)^{-1}AW\|$$

remains bounded for a particular subset $\mathcal{W}$ of $\mathcal{W}_+$. It should be noted that boundedness does not hold for an arbitrary subset $\mathcal{W}$ of $\mathcal{W}_+$. Take for example $A = (0\ 1)$ and let

$$W(\epsilon) = \begin{pmatrix} \frac{2}{\epsilon} & 1 \\ 1 & \epsilon \end{pmatrix}$$

for $\epsilon > 0$. Then $W(\epsilon) \in \mathcal{W}_+$ for $\epsilon > 0$, and

$$(AW(\epsilon)A^T)^{-1}AW(\epsilon) = \begin{pmatrix} \frac{1}{\epsilon} & 1 \end{pmatrix}.$$

This implies that $\|(AWA^T)^{-1}AW\|$ is not bounded when W is allowed to vary in $\mathcal{W}_+$. See Stewart [24] for another example of unboundedness and related discussion. For the case where $\mathcal{W}$ is the set of positive definite diagonal matrices, Dikin [8] gives an explicit formula for the optimal π in (1.1) as a convex combination of the *basic solutions* formed by satisfying m linearly independent equations. From this result, the boundedness is obvious. If A does not have full row rank, it is still possible to show boundedness; see Ben-Israel [2, p. 108]. Later, Wei [28] also studied boundedness in absence of a full row rank assumption on A and has furthermore given some stability results. Bobrovnikova and Vavasis [6] have given boundedness results for complex diagonal weight matrices. The geometry of the set $(AWA^T)^{-1}AWg$ when W varies over the set of positive

definite diagonal matrices has been studied by Hanke and Neumann [18]. Based on the formula derived by Dikin [8], Forsgren [10] has given boundedness results when $\mathcal{W}$ is the set of positive definite diagonally dominant matrices.

We show boundedness for the set of weight matrices that are arbitrary nonnegative combinations of a set of fixed positive semidefinite symmetric matrices and the set of inverses of such matrices. As a special case, we then obtain the set of weight matrices of the form (1.2), which was our original interest. The boundedness is shown in the following way. In section 2, we review results for the characterization of π as W varies over the set of symmetric matrices such that AWA^T is nonsingular. Section 3 establishes the boundedness when W is allowed to vary over a set of matrices that are nonnegative linear combinations of a number of fixed positive semidefinite matrices such that AWA^T is positive definite. In section 4, results that are needed to handle the projection using the inverse weight matrix are given. In section 5, we combine results from the previous two sections to show boundedness for the π that solves (1.4) when M is allowed to vary over the nonnegative linear combinations of a set of fixed positive semidefinite symmetric matrices.

The research was initiated by a paper by Gonzaga and Lara [15]. The link to that paper has subsequently been superseded, but we include a discussion relating our results to the result of Gonzaga and Lara in the appendix.

1.1. Motivation. Our interest in weighted linear least-squares problems is from interior methods for optimization and in particular for convex quadratic programming. There is a vast number of papers on interior methods, and here we give only a brief motivation for the weighted linear least-squares problems that arise. Any convex quadratic programming problem can be transformed to the form

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & \frac{1}{2}x^THx + c^Tx \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array} \tag{1.6}$$

where H is a positive semidefinite symmetric $n \times n$ matrix and A is an $m \times n$ matrix of full row rank. For $x \in \mathbb{R}^n$, $\pi \in \mathbb{R}^m$, and $s \in \mathbb{R}^n$ such that $x > 0$ and $s > 0$, an iteration of a primal-dual path-following interior method for solving (1.6) typically takes a Newton step towards the solution of the equations

$$Hx + c - A^T\pi - s = 0, \tag{1.7a}$$

$$Ax - b = 0, \tag{1.7b}$$

$$Xs - \mu e = 0, \tag{1.7c}$$

where μ is a positive barrier parameter; see, e.g., Monteiro and Adler [21, p. 46]. Here, $X = \operatorname{diag}(x)$ and similarly below $S = \operatorname{diag}(s)$. Strict positivity of x and s is implicitly required and typically maintained by limiting the step length. If μ is set equal to zero in (1.7) and the implicit requirements $x > 0$ and $s > 0$ are replaced by $x \geq 0$ and $s \geq 0$, the optimality conditions for (1.6) are obtained. Consequently, (1.7) and the implicit positivity of x and s may be viewed as a perturbation of the optimality conditions for (1.6). In a primal-dual path-following interior method, the perturbation is driven to zero to make the method converge to an optimal solution. The equations (1.7) are often referred to as the primal-dual equations. Forming the Newton equations associated with (1.7) for the corrections Δx, $\Delta\pi$, Δs and eliminating Δs gives

$$\begin{pmatrix} H + X^{-1}S & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} -\Delta x \\ \Delta\pi \end{pmatrix} = \begin{pmatrix} Hx + c - \mu X^{-1}e - A^T\pi \\ Ax - b \end{pmatrix}. \tag{1.8}$$

If x and s are strictly feasible, i.e., x and s are strictly positive and x satisfies $Ax = b$, then a comparison of (1.4) and (1.8) shows that the Newton equations (1.8) can be associated with a weighted linear least-squares problem with a positive definite weight matrix $(H + X^{-1}S)^{-1}$. A sequence of strictly feasible iterates $\{x_k\}_{k=0}^{\infty}$ gives rise to a sequence of weighted linear least-squares problems, where the weight matrix changes but A is constant.

In a number of convergence proofs for linear programming, a crucial step is to ensure boundedness of the step $(\Delta x, \Delta s)$; see, e.g., Vavasis and Ye [27, Lemma 4] and Wright [30, Lemmas 7.2 and A.4]. Since linear programming is the special case of convex quadratic programming where $H = 0$, we are interested in extending this boundedness result to convex quadratic programming. Therefore, the boundedness of

$$\|(A(H + X^{-1}S)^{-1}A^T)^{-1}A(H + X^{-1}S)^{-1}\| \tag{1.9}$$

as $X^{-1}S$ varies over the set of diagonal positive definite matrices is of interest. This boundedness property of (1.9) is shown in section 5.

1.2. Notation. When we refer to matrix norms and make no explicit reference to what type of norm is considered, it can be any matrix norm that is induced from a vector norm such that $\|(x^T\ 0)^T\| = \|x\|$ holds for any vector x. To denote the ith eigenvalue and the ith singular value, we use λ_i and σ_i, respectively. For symmetric matrices A and B of equal dimension, $A \succeq B$ means that $A - B$ is positive semidefinite. Similarly, $A \succ B$ means that $A - B$ is positive definite.

The remainder of this section is given in Forsgren [10]. It is restated here for completeness. For an $m \times n$ matrix A of full row rank, we shall denote by $\mathcal{J}(A)$ the collection of sets of column indices associated with the nonsingular $m \times m$ submatrices of A. For $J \in \mathcal{J}(A)$, we denote by A_J the $m \times m$ nonsingular submatrix formed by the columns of A with indices in J. Associated with $J \in \mathcal{J}(A)$, for a diagonal $n \times n$ matrix D, we denote by D_J the $m \times m$ diagonal matrix formed by the elements of D that have row and column indices in J. Similarly, for a vector g of dimension n, we denote by g_J the vector of dimension m with the components of g that have indices in J. The slightly different meanings of A_J, D_J, and g_J are used in order not to make the notation more complicated than necessary. For an example clarifying the concepts, see Forsgren [10, p. 766].

The analogous notation is used for an $m \times n$ matrix A of full row rank and an $n \times r$ matrix U of full row rank in that we associate $\mathcal{J}(AU)$ with the collection of sets of column indices corresponding to nonsingular $m \times m$ submatrices of AU. Associated with $J \in \mathcal{J}(AU)$, for a diagonal $r \times r$ matrix D, we denote by D_J the $m \times m$ diagonal matrix formed by the elements of D that have row and column indices in J. Similarly, for a vector g of dimension r, we denote by g_J the vector of dimension m with the components of g that have indices in J. Since column indices of AU are also column indices of U, for $J \in \mathcal{J}(AU)$, we denote by U_J the $n \times m$ submatrix of full column rank formed by the columns of U with indices in J. Note that each element of $\mathcal{J}(A)$ as well as each element of $\mathcal{J}(AU)$ is a collection of m indices.

2. Background. In this section, we review some fundamental results. The following theorem, which states that the solution of a diagonally weighted linear least-squares problem can be expressed as a certain convex combination, is the basis for our results. As far as we know, it was originally given by Dikin [8] who used it in the convergence analysis of the interior point method for linear programming he proposed [7]. The proof of the theorem is based on the Cauchy–Binet formula and Cramer's rule.

THEOREM 2.1 (Dikin [8]). *Let A be an $m \times n$ matrix of full row rank, let g be a vector of dimension n, and let D be a positive definite diagonal $n \times n$ matrix. Then,*

$$(ADA^T)^{-1}ADg = \sum_{J \in \mathcal{J}(A)} \left(\frac{\det(D_J)\det(A_J)^2}{\sum_{K \in \mathcal{J}(A)} \det(D_K)\det(A_K)^2} \right) A_J^{-T} g_J,$$

where $\mathcal{J}(A)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of A.

Proof. See, e.g., Ben-Tal and Teboulle [3, Corollary 2.1]. ∎

Theorem 2.1 implies that if the weight matrix is diagonal and positive definite, then the solution to the weighted least-squares problem (1.1) lies in the convex hull of the *basic solutions* formed by satisfying m linearly independent equations. Hence, this theorem provides an expression on the supremum of $\|(ADA^T)^{-1}AD\|$ for D diagonal and positive definite, as the following corollary shows.

COROLLARY 2.2. *Let A be an $m \times n$ matrix of full row rank, and let $\mathcal{D}_+$ denote the set of positive definite diagonal $n \times n$ matrices. Then,*

$$\sup_{D \in \mathcal{D}_+} \|(ADA^T)^{-1}AD\| = \max_{J \in \mathcal{J}(A)} \|A_J^{-T}\|,$$

where $\mathcal{J}(A)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of A.

Proof. See, e.g., Forsgren [10, Corollary 2.2]. ∎

The boundedness has been discussed by a number of authors over the years; see, e.g., Ben-Tal and Teboulle [3], O'Leary [22], Stewart [24], and Todd [25]. Theorem 2.1 can be generalized to the case where the weight matrix is an arbitrary symmetric, not necessarily diagonal, matrix such that AWA^T is nonsingular. The details are given in the following theorem.

THEOREM 2.3 (Forsgren [10]). *Let A be an $m \times n$ matrix of full row rank, and let W be a symmetric $n \times n$ matrix such that AWA^T is nonsingular. Suppose $W = UDU^T$, where D is diagonal. Then,*

$$(AWA^T)^{-1}AW = \sum_{J \in \mathcal{J}(AU)} \left(\frac{\det(D_J)\det(AU_J)^2}{\sum_{K \in \mathcal{J}(AU)} \det(D_K)\det(AU_K)^2} \right) (AU_J)^{-T} U_J^T,$$

where $\mathcal{J}(AU)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of AU.

Proof. See Forsgren [10, Theorem 3.1]. ∎

3. Nonnegative combinations of positive semidefinite matrices. Let A be an $m \times n$ matrix of full row rank and assume that we are given an $n \times n$ symmetric weight matrix $W(\alpha)$, which depends on a vector $\alpha \in \mathbb{R}^t$ for some t. If $W(\alpha)$ can be decomposed as $W(\alpha) = UD(\alpha)U^T$, where U does *not* depend on α and $D(\alpha)$ is diagonal, Theorem 2.3 can be applied, provided $AW(\alpha)A^T$ is nonsingular and the matrices $(AU_J)^{-T}U_J^T$ involved do not depend on α. If, in addition $D(\alpha) \succeq 0$, then the linear combination of Theorem 2.3 is a convex combination. Consequently, the norm remains bounded as long as the supremum is taken over a set of values of α for which $AW(\alpha)A^T \succ 0$ and $D(\alpha) \succeq 0$. In particular, we are interested in the case where a set of positive semidefinite and symmetric matrices, W_i, $i = 1, \ldots, t$, are given and $W(\alpha)$ is defined as $W(\alpha) = \sum_{i=1}^t \alpha_i W_i$. The following two lemmas and associated corollary concern the decomposition of $W(\alpha)$. The first lemma concerns the set of all possible

decompositions of a positive semidefinite matrix W as $W = UU^T$ and the relation between different decompositions of this type.

LEMMA 3.1. *Let W be a symmetric positive semidefinite $n \times n$ matrix of rank r, and let $\bar{\mathcal{U}} = \{U \in \mathbb{R}^{n\times r} : UU^T = W\}$. Then, $\bar{\mathcal{U}}$ is nonempty and compact. Further, if U and $\widetilde{U}$ belong to $\bar{\mathcal{U}}$, then there is an $r \times r$ orthogonal matrix Q such that $U = \widetilde{U}Q$.*

Proof. It is possible to decompose W as $W = UU^T$, where U is an $n \times r$ matrix of full column rank, for example, using a Cholesky factorization with symmetric interchanges; see, e.g., Golub and Van Loan [14, section 4.2.9]. Therefore, $\bar{\mathcal{U}}$ is nonempty. If U and $\widetilde{U}^T$ both belong to $\bar{\mathcal{U}}$, then

$$U^T x = 0 \;\Leftrightarrow\; UU^T x = 0 \;\Leftrightarrow\; \widetilde{U}\widetilde{U}^T x = 0 \;\Leftrightarrow\; \widetilde{U}^T x = 0.$$

Hence, U^T and $\widetilde{U}^T$ have the same null space, which implies that the range spaces of U and $\widetilde{U}$ are the same. Therefore, there is a nonsingular $r \times r$ matrix M such that $U = \widetilde{U}M$, from which it follows that $\widetilde{U}\widetilde{U}^T = \widetilde{U}MM^T\widetilde{U}^T$. Premultiplying this equation by $\widetilde{U}^T$ and postmultiplying it by $\widetilde{U}$ gives

$$\widetilde{U}^T\widetilde{U}\widetilde{U}^T\widetilde{U} = \widetilde{U}^T\widetilde{U}MM^T\widetilde{U}^T\widetilde{U}. \tag{3.1}$$

Since $\widetilde{U}^T\widetilde{U}$ is nonsingular, (3.1) gives $MM^T = I$. Compactness is established by proving boundedness and closedness. Boundedness holds because $\|U^T e_i\|_2^2 = W_{ii}$, $i = 1, \ldots, n$, where e_i is the ith unit vector. Let $\{U^{(i)}\}_{i=1}^{\infty}$ be a sequence converging to U^* such that $U^{(i)} \in \bar{\mathcal{U}}$ for all i. From the continuity of matrix multiplication, U^* belongs to $\bar{\mathcal{U}}$, and the closedness of $\bar{\mathcal{U}}$ follows. ∎

A consequence of this lemma is that we can decompose each W_i, $i = 1, \ldots, t$, as stated in the following corollary.

COROLLARY 3.2. *For $i = 1, \ldots, t$, let W_i be an $n \times n$ symmetric positive semidefinite matrix of rank r_i. Let $r = \sum_{i=1}^{t} r_i$. Then*

$$\mathcal{U} = \left\{U \in \mathbb{R}^{n\times r} : U = \begin{pmatrix} U_1 & U_2 & \cdots & U_t \end{pmatrix}, U_i \in \mathbb{R}^{n\times r_i}, U_i U_i^T = W_i, i = 1, \ldots, t\right\}$$

is a well-defined compact subset of $\mathbb{R}^{n\times r}$. Furthermore, if U and $\widetilde{U}$ belong to $\mathcal{U}$, then, for $i = 1, \ldots, t$, there are orthogonal $r_i \times r_i$ matrices Q_i such that $U_i = \widetilde{U}_i Q_i$.

Proof. The result follows by applying Lemma 3.1 to each W_i. ∎

It should be noted that $\mathcal{U}$ depends on the matrices W_i. This dependence will be suppressed in order to not make the notation more complicated than necessary. From Corollary 3.2, we get a decomposition result for matrices that are nonnegative linear combinations of symmetric positive semidefinite matrices, as is stated in the following lemma. It shows that if we are given a set of positive semidefinite and symmetric matrices, W_i, $i = 1, \ldots, t$, and $W(\alpha)$ is defined as $W(\alpha) = \sum_{i=1}^{t} \alpha_i W_i$, then we can decompose $W(\alpha)$ into the form $W(\alpha) = UD(\alpha)U^T$, where U does not depend on α and $D(\alpha)$ is diagonal.

LEMMA 3.3. *For $\alpha \in \mathbb{R}^t$, let $W(\alpha) = \sum_{i=1}^{t} \alpha_i W_i$, where W_i, $i = 1, \ldots, t$, are symmetric positive semidefinite $n \times n$ matrices. Further, let $\mathcal{U}$ be associated with $W_i, i = 1, \ldots, t$, according to Corollary 3.2, and for each i, let r_i denote* $\operatorname{rank}(W_i)$ *and let I_i be an identity matrix of dimension r_i. Then $W(\alpha)$ may be decomposed as*

$$W(\alpha) = UD(\alpha)U^T,$$

where U is any matrix in $\mathcal{U}$ and $D(\alpha) = \operatorname{diag}(\alpha_1 I_1, \alpha_2 I_2, \ldots, \alpha_t I_t)$.

Proof. Corollary 3.2 shows that we may write

$$W(\alpha) = \sum_{i=1}^{t} \alpha_i W_i = \sum_{i=1}^{t} \alpha_i U_i U_i^T = U D(\alpha) U^T,$$

where U is an arbitrary matrix in $\mathcal{U}$ and $D(\alpha) = \operatorname{diag}(\alpha_1 I_1, \alpha_2 I_2, \ldots, \alpha_t I_t)$. ∎

Note that $D(\alpha)$ is positive semidefinite if $\alpha \geq 0$. An application of Theorem 2.3 to the decomposition of Lemma 3.3 now gives the boundedness result for nonnegative combinations of positive semidefinite matrices, as stated in the following proposition.

PROPOSITION 3.4. *Let A be an $m \times n$ matrix of full row rank. For $\alpha \in \mathbb{R}^t$, $\alpha \geq 0$, let $W(\alpha) = \sum_{i=1}^{t} \alpha_i W_i$, where W_i, $i = 1, \ldots, t$, are symmetric positive semidefinite $n \times n$ matrices. If $W(\alpha)$ is decomposed as $W(\alpha) = U D(\alpha) U^T$, according to Lemma 3.3, then for $\alpha \geq 0$ and $AW(\alpha)A^T \succ 0$,*

$$(AW(\alpha)A^T)^{-1} AW(\alpha) = \sum_{J \in \mathcal{J}(AU)} \left(\frac{\det(D_J(\alpha)) \det(AU_J)^2}{\sum_{K \in \mathcal{J}(AU)} \det(D_K(\alpha)) \det(AU_K)^2} \right) (AU_J)^{-T} U_J^T.$$

Furthermore,

$$\sup_{\substack{\alpha \geq 0: \\ AW(\alpha)A^T \succ 0}} \|(AW(\alpha)A^T)^{-1} AW(\alpha)\| \leq \min_{U \in \mathcal{U}} \max_{J \in \mathcal{J}(AU)} \|(AU_J)^{-T} U_J^T\|, \tag{3.2}$$

where $\mathcal{J}(AU)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of AU, and $\mathcal{U}$ is associated with W_i, $i = 1, \ldots, t$, according to Corollary 3.2.

Proof. If $AW(\alpha)A^T \succ 0$, Theorem 2.3 immediately gives

$$(AW(\alpha)A^T)^{-1} AW(\alpha) = \sum_{J \in \mathcal{J}(AU)} \left(\frac{\det(D_J(\alpha)) \det(AU_J)^2}{\sum_{K \in \mathcal{J}(AU)} \det(D_K(\alpha)) \det(AU_K)^2} \right) (AU_J)^{-T} U_J^T.$$

Since $\alpha \geq 0$, it follows that $D(\alpha) \succeq 0$. Consequently, $\det(D_J(\alpha)) \geq 0$ for all $J \in \mathcal{J}(AU)$. Thus, the above expression gives

$$\sup_{\alpha \geq 0: AW(\alpha)A^T \succ 0} \|(AW(\alpha)A^T)^{-1} AW(\alpha)\| \leq \max_{J \in \mathcal{J}(AU)} \|(AU_J)^{-T} U_J^T\|.$$

Since this result holds for all $U \in \mathcal{U}$, it holds when taking the infimum over $U \in \mathcal{U}$. To show that the infimum is attained, let

$$f_J(U) = \begin{cases} \|(AU_J)^{-T} U_J^T\| & \text{if } \det(AU_J) \neq 0, \\ 0 & \text{otherwise} \end{cases}$$

for every J that is a subset of $\{1, \ldots, n\}$ such that $|J| = m$. For a fixed J, f_J is continuous at every $\widetilde{U}$ such that $\det(A\widetilde{U}_J) \neq 0$. Further, at $\widetilde{U}$ such that $A\widetilde{U}_J$ is singular, f_J is a lower semicontinuous function; see, e.g., Royden [23, p. 51]. Hence, f_J is lower semicontinuous everywhere. Due to the construction of $f_J(U)$,

$$\max_{J \in \mathcal{J}(AU)} \|(AU_J)^{-T} U_J^T\| = \max_{J: |J| = m} f_J(U).$$

The maximum of a finite collection of lower semicontinuous functions is lower semicontinuous; see, e.g., Royden [23, p. 51], and the set $\mathcal{U}$ is compact by Corollary 3.2. Therefore, the infimum is attained (see, e.g., Royden [23, p. 195]) and the proof is complete. ∎

Note that Proposition 3.4 as special cases includes two known cases: (i) the diagonal matrices, where $W(\alpha) = \sum_{i=1}^{n} \alpha_i e_i e_i^T$; and (ii) the diagonally dominant matrices, where

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i e_i e_i^T + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\alpha_{ij}^{+} (e_i + e_j)(e_i + e_j)^T + \alpha_{ij}^{-} (e_i - e_j)(e_i - e_j)^T \right).$$

In both these cases, the supremum bound of (3.2) is sharp. This is because all the matrices whose nonnegative linear combinations form the weight matrices are of rank one. In that case, the minimum over U in (3.2) is not necessary since it follows from Corollary 3.2 that the columns of U are unique up to multiplication by ± 1. Hence, $D(\alpha)$ may be adjusted so as to give weight one to the submatrix AU_J for which the maximum of the right-hand side of (3.2) is achieved and to give negligible weight to the other submatrices. In general, when not all matrices whose nonnegative linear combinations form the weight matrix have rank one, it is an open question if the supremum bound is sharp.

4. Inversion of the weight matrix. For a constant positive semidefinite matrix H, our goal is to obtain a bound on $\|(A(H+D)^{-1}A^T)^{-1}A(H+D)^{-1}\|$ when D is an arbitrary positive definite diagonal matrix. One major obstacle in applying Theorem 2.3 is the inverse in the weight matrix $(H+D)^{-1}$. The following proposition and its subsequent corollary and lemma provide a solution to this problem.

PROPOSITION 4.1. *Suppose that an $n \times n$ orthogonal matrix Q is partitioned as $Q = (Z\ Y)$, where Z is an $n \times s$ matrix and $2s \le n$. Further, let W be a symmetric nonsingular $n \times n$ matrix such that $Z^T W^{-1} Z$ and $Y^T W Y$ are nonsingular. Then*

$$(Y^T W Y)^{-1} Y^T W Z = -((Z^T W^{-1} Z)^{-1} Z^T W^{-1} Y)^T$$

and

$$\begin{aligned} \sigma_i^2((Y^T W Y)^{-1} Y^T W) &= \sigma_i^2((Z^T W^{-1} Z)^{-1} Z^T W^{-1}) \\ &= 1 + \sigma_i^2((Z^T W^{-1} Z)^{-1} Z^T W^{-1} Y) \\ &= 1 + \sigma_i^2((Y^T W Y)^{-1} Y^T W Z), \quad i = 1, \ldots, s, \\ \sigma_i((Y^T W Y)^{-1} Y^T W) &= 1, \quad i = s+1, \ldots, n-s. \end{aligned}$$

Proof. The orthogonality of Q ensures that $Y^T Z = 0$ and $ZZ^T + YY^T = I$. This gives

$$0 = Y^T Z = Y^T W (ZZ^T + YY^T) W^{-1} Z = Y^T W Z Z^T W^{-1} Z + Y^T W Y Y^T W^{-1} Z,$$

and hence

$$(Y^T W Y)^{-1} Y^T W Z = -((Z^T W^{-1} Z)^{-1} Z^T W^{-1} Y)^T, \tag{4.1}$$

proving the first part of the proposition.

Since $Z^T W^{-1} Z$ and $Y^T W Y$ are nonsingular, we may write

$$(Z^T W^{-1} Z)^{-1} Z^T W^{-1} \begin{pmatrix} Z & Y \end{pmatrix} = \begin{pmatrix} I & (Z^T W^{-1} Z)^{-1} Z^T W^{-1} Y \end{pmatrix}, \tag{4.2a}$$

$$(Y^T W Y)^{-1} Y^T W \begin{pmatrix} Z & Y \end{pmatrix} = \begin{pmatrix} (Y^T W Y)^{-1} Y^T W Z & I \end{pmatrix}. \tag{4.2b}$$

The orthogonality of Q ensures that

$$\sigma_i((Z^TW^{-1}Z)^{-1}Z^TW^{-1}Q) = \sigma_i((Z^TW^{-1}Z)^{-1}Z^TW^{-1}), \quad i = 1, \dots, s. \tag{4.3}$$

We also have

$$\sigma_i^2 \begin{pmatrix} I & (Z^TW^{-1}Z)^{-1}Z^TW^{-1}Y \end{pmatrix} = 1 + \sigma_i^2 \begin{pmatrix} (Z^TW^{-1}Z)^{-1}Z^TW^{-1}Y \end{pmatrix}, \tag{4.4}$$

$i = 1, \dots, s$. A combination of (4.2a), (4.3), and (4.4) gives

$$\sigma_i^2((Z^TW^{-1}Z)^{-1}Z^TW^{-1}) = 1 + \sigma_i^2((Z^TW^{-1}Z)^{-1}Z^TW^{-1}Y), \quad i = 1, \dots, s. \tag{4.5}$$

An analogous argument applied to (4.2b), taking into account that $2s \le n$, gives

$$\sigma_i^2((Y^TWY)^{-1}Y^TW) = 1 + \sigma_i^2((Y^TWY)^{-1}Y^TWZ), \quad i = 1, \dots, s, \tag{4.6a}$$
$$\sigma_i^2((Y^TWY)^{-1}Y^TW) = 1, \quad i = s+1, \dots, n-s. \tag{4.6b}$$

The second part of the proposition follows by a combination of (4.1), (4.5), and (4.6). ∎

In particular, Proposition 4.1 gives the equivalence between the Euclidean norms of a projection and the projection onto the complementary space using the inverse weight matrix, given that the matrices used to represent the spaces are orthogonal. This is shown in the following corollary.

COROLLARY 4.2. *Suppose that an $n \times n$ orthogonal matrix Q is partitioned as $Q = (Z\ Y)$, where Y is an $n \times m$ matrix. Further, let W be a symmetric nonsingular $n \times n$ matrix such that $Z^TW^{-1}Z$ and Y^TWY are nonsingular. Then*

$$\|(Y^TWY)^{-1}Y^TW\|_2 = \|(Z^TW^{-1}Z)^{-1}Z^TW^{-1}\|_2.$$

Further, let $\mathcal{W}_+$ denote the set of $n \times n$ positive definite symmetric matrices, and let $\mathcal{W} \subseteq \mathcal{W}_+$. Then,

$$\sup_{W \in \mathcal{W}} \|(Y^TWY)^{-1}Y^TW\|_2 = \sup_{W \in \mathcal{W}} \|(Z^TW^{-1}Z)^{-1}Z^TW^{-1}\|_2.$$

Proof. If $m \ge n/2$, the first statement follows by letting $i = 1$ in Proposition 4.1. The second statement is a direct consequence of the first one. If $m < n/2$, we may similarly apply Proposition 4.1 after interchanging the roles of Y and Z, and W and W^{-1}. ∎

As noted above, Corollary 4.2 states the equality between the Euclidean norms of two projections, given that the matrices describing the spaces onto which we project are orthogonal. The following lemma relates the Euclidean norms of the projections when the matrices are not orthogonal.

LEMMA 4.3. *Let A be an $m \times n$ matrix of full row rank, and let N be a matrix whose columns form a basis for the null space of A. Further, let W be a symmetric nonsingular $n \times n$ matrix such that $N^TW^{-1}N$ and A^TWA are nonsingular. Then*

$$\frac{\sigma_{n-m}(N)}{\sigma_1(A)} \le \frac{\|(AWA^T)^{-1}AW\|_2}{\|(N^TW^{-1}N)^{-1}N^TW^{-1}\|_2} \le \frac{\sigma_1(N)}{\sigma_m(A)}.$$

Proof. Let $Q = (Z\ Y)$ be an orthogonal matrix such that the columns of Z form a basis for the null space of A. Then, there are nonsingular matrices R_Z and R_Y such that $N = ZR_Z$ and $A^T = YR_Y$. Since a matrix norm which is induced from a vector

norm is submultiplicative (see, e.g., Horn and Johnson [19, Theorem 5.6.2]) this gives

$$\frac{1}{\|R_Z\|} \le \frac{\|(N^TW^{-1}N)^{-1}N^TW^{-1}\|}{\|(Z^TW^{-1}Z)^{-1}Z^TW^{-1}\|} \le \|R_Z^{-1}\|, \tag{4.7a}$$

$$\frac{1}{\|R_Y\|} \le \frac{\|(AWA^T)^{-1}AW\|}{\|(Y^TWY)^{-1}Y^TW\|} \le \|R_Y^{-1}\|. \tag{4.7b}$$

If the Euclidean norm is used, the bounds in (4.7) can be expressed in terms of singular values of A and N since Y and Z are orthogonal matrices, i.e.,

$$\|R_Z\|_2 = \sigma_1(N), \quad \|R_Z^{-1}\|_2 = 1/\sigma_{n-m}(N), \tag{4.8a}$$

$$\|R_Y\|_2 = \sigma_1(A), \quad \|R_Y^{-1}\|_2 = 1/\sigma_m(A). \tag{4.8b}$$

A combination of Corollary 4.2, (4.7), and (4.8) gives the stated result. ∎

If the weight matrix is allowed to vary over some subset of the positive definite symmetric matrices, it follows from Lemma 4.3 that the norm of the projection onto a subspace is bounded if and only if the norm of the projection onto the orthogonal complement is bounded when using inverses of the weight matrices. This is made precise in the following corollary.

COROLLARY 4.4. *Let $\mathcal{W}_+$ denote the set of $n \times n$ positive definite symmetric matrices, and let $\mathcal{W} \subseteq \mathcal{W}_+$. Let A be an $m \times n$ matrix of full row rank, and let N be a matrix whose columns form a basis for the null space of A. Then*

$$\sup_{W\in\mathcal{W}} \|(AWA^T)^{-1}AW\| < \infty \quad \textit{if and only if} \quad \sup_{W\in\mathcal{W}} \|(N^TW^{-1}N)^{-1}N^TW^{-1}\| < \infty.$$

In particular,

$$\frac{\sigma_{n-m}(N)}{\sigma_1(A)} \sup_{W\in\mathcal{W}} \|(N^TW^{-1}N)^{-1}N^TW^{-1}\|_2 \le \sup_{W\in\mathcal{W}} \|(AWA^T)^{-1}AW\|_2,$$

$$\sup_{W\in\mathcal{W}} \|(AWA^T)^{-1}AW\|_2 \le \frac{\sigma_1(N)}{\sigma_m(A)} \sup_{W\in\mathcal{W}} \|(N^TW^{-1}N)^{-1}N^TW^{-1}\|_2.$$

Proof. The second statement follows by multiplying the inequalities in Lemma 4.3 by $\|(N^TW^{-1}N)^{-1}N^TW^{-1}\|_2$ and then taking the supremum of the three expressions. The first statement of the corollary then follows from the equivalence of matrix norms that are induced from vector norms; see, e.g., Horn and Johnson [19, Theorem 5.6.18]. ∎

5. Inversion and nonnegative combination. Let A be an $m \times n$ matrix of full row rank, and let Z be a matrix whose columns form an orthonormal basis for the null space of A. Further, let $M(\alpha) = \sum_{i=1}^t \alpha_i M_i$, where M_i, $i = 1, \ldots, t$, are given symmetric positive semidefinite $n \times n$ matrices. In section 3 the weight matrix was assumed to be the nonnegative combination of symmetric positive semidefinite matrices. This section concerns weight matrices that are the inverse of such combinations, i.e., where the weight matrix is the inverse of $M(\alpha)$. Further, if the problem is originally posed as the KKT system, cf. (1.4),

$$\begin{pmatrix} M(\alpha) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r(\alpha) \\ \pi(\alpha) \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{5.1}$$

it makes sense to study the problem under the assumption that $Z^TM(\alpha)Z \succ 0$ since in our situation, $Z^TM(\alpha)Z \succ 0$ if and only if the matrix of (5.1) is nonsingular; see Gould [16, Lemma 3.4]. Note that $Z^TM(\alpha)Z \succ 0$ is a weaker assumption than $M(\alpha) \succ 0$, which is necessary if the least-squares formulation is to be valid. A combination of Proposition 3.4 and Lemma 4.3 shows that $\pi(\alpha)$ remains bounded under the above-mentioned assumptions. This is stated in the following theorem, which is the main result of this paper.

THEOREM 5.1. *Let A be an $m \times n$ matrix of full row rank and let g be an n-vector. Further, let Z be a matrix whose columns form an orthonormal basis for the null space of A. For $\alpha \in \mathbb{R}^t$, $\alpha \geq 0$, let $M(\alpha) = \sum_{i=1}^t \alpha_i M_i$, where M_i, $i = 1, \ldots, t$, are symmetric positive semidefinite $n \times n$ matrices. Further, let $r(\alpha)$ and $\pi(\alpha)$ satisfy*

$$\begin{pmatrix} M(\alpha) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r(\alpha) \\ \pi(\alpha) \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}.$$

Then,

$$\sup_{\substack{\alpha \geq 0: \\ Z^TM(\alpha)Z \succ 0}} \|\pi(\alpha)\| < \infty. \tag{5.2}$$

In particular, if $Z^TM(\alpha)Z \succ 0$, then

$$\|\pi(\alpha)\|_2 \leq \frac{1}{\sigma_m(A)} \|(Z^TM(\alpha)Z)^{-1}Z^TM(\alpha)\|_2 \|g\|_2. \tag{5.3}$$

Finally, if $M(\alpha)$ is decomposed according to Lemma 3.3, then

$$\sup_{\substack{\alpha \geq 0: \\ Z^TM(\alpha)Z \succ 0}} \|\pi(\alpha)\|_2 \leq \frac{1}{\sigma_m(A)} \min_{U \in \mathcal{U}} \max_{J \in \mathcal{J}(Z^TU)} \|(Z^TU_J)^{-T}U_J^T\|_2 \|g\|_2, \tag{5.4}$$

where $\mathcal{J}(Z^TU)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of Z^TU, and $\mathcal{U}$ is associated with M_i, $i = 1, \ldots, t$, according to Corollary 3.2.

Proof. For $\alpha \geq 0$ and $\epsilon > 0$, $M(\alpha) + \epsilon I \succ 0$. Therefore,

$$\pi(\alpha, \epsilon) = (A(M(\alpha) + \epsilon I)^{-1}A^T)^{-1}A(M(\alpha) + \epsilon I)^{-1}g$$

is well defined. By Lemma 4.3 it follows that

$$\begin{aligned} \|\pi(\alpha, \epsilon)\|_2 &\leq \|(A(M(\alpha) + \epsilon I)^{-1}A^T)^{-1}A(M(\alpha) + \epsilon I)^{-1}\|_2 \|g\|_2 \\ &\leq \frac{1}{\sigma_m(A)} \|(Z^T(M(\alpha) + \epsilon I)Z)^{-1}Z^T(M(\alpha) + \epsilon I)\|_2 \|g\|_2. \end{aligned} \tag{5.5}$$

For α such that $Z^TM(\alpha)Z \succ 0$, the matrix in the system of equations defining $\pi(\alpha)$ and $r(\alpha)$ is nonsingular; see Gould [16, Lemma 3.4]. Then, the implicit function theorem implies that $\lim_{\epsilon \to 0^+} \pi(\alpha, \epsilon) = \pi(\alpha)$. Therefore, letting $\epsilon \to 0^+$ in (5.5) gives (5.3). Taking the supremum over α such that $\alpha \geq 0$ and $Z^TM(\alpha)Z \succ 0$ and using Proposition 3.4 gives (5.4), from which (5.2) follows upon observing that all norms on a real finite-dimensional vector space are equivalent; see, e.g., Horn and Johnson [19, Corollary 5.4.5]. ∎

As a consequence of Theorem 5.1, we are now able to prove the boundedness of the projection operator for the application of primal-dual interior methods to convex quadratic programming described in section 1.1.

COROLLARY 5.2. *Let H be a positive semidefinite symmetric $n \times n$ matrix, let A be an $m \times n$ matrix of full row rank, and let $\mathcal{D}_+$ denote the space of positive definite diagonal $n \times n$ matrices. Then,*

$$\sup_{D \in \mathcal{D}_+} \|(A(H+D)^{-1}A^T)^{-1}A(H+D)^{-1}\| < \infty.$$

Proof. If $M(\alpha) \succ 0$, then $\pi(\alpha)$ of Theorem 5.1 satisfies

$$\pi(\alpha) = (AM(\alpha)^{-1}A^T)^{-1}AM(\alpha)^{-1}g.$$

Since $\{\alpha \geq 0 : M(\alpha) \succ 0\} \subseteq \{\alpha \geq 0 : Z^T M(\alpha) Z \succ 0\}$, Theorem 5.1 implies that $\pi(\alpha)$ is bounded. This holds for any vector g, and hence

$$\sup_{\alpha \geq 0 : M(\alpha) \succ 0} \|(AM(\alpha)^{-1}A^T)^{-1}AM(\alpha)^{-1}\| < \infty. \tag{5.6}$$

The stated result follows by applying (5.6) with $M_i = e_i e_i^T$, $i = 1, \ldots, m$, $M_{m+1} = H$, and letting $\alpha_{m+1} = 1$. ∎

For convenience in notation, it has been assumed that all variables of the convex quadratic program are subject to bounds. It can be observed that the analogous results hold when some variables are not subject to bounds. In this situation, M of (1.4) may be partitioned as

$$M = \begin{pmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{pmatrix} + \begin{pmatrix} D_{11} & 0 \\ 0 & 0 \end{pmatrix},$$

where H is symmetric and positive semidefinite and D_{11} is diagonal and positive definite. Let A be partitioned conformally with M as $A = (A_1 \quad A_2)$. Then, (1.4) has a unique solution as long as there is no nonzero p_2 such that $A_2 p_2 = 0$ and $p_2^T H_{22} p_2 = 0$; see Gould [16, Lemma 3.4]. Hence, under this additional assumption, Theorem 5.1 can be applied to bound $\|\pi(\alpha)\|$ as D_{11} varies over the set of positive definite diagonal matrices.

6. Summary. It has been shown that results concerning the boundedness of $(AWA^T)^{-1}AW$ for A of full row rank and W diagonal, or diagonally dominant, and symmetric positive definite can be extended to a more general case where W is a nonnegative linear combination of a set of symmetric positive semidefinite matrices such that $AWA^T \succ 0$. Further, boundedness has been shown for the projection onto the null space of A using as the weight matrix the inverse of a nonnegative linear combination of a number of symmetric positive semidefinite matrices. This result has been used to show boundedness of a projection operator arising in a primal-dual interior method for convex quadratic programming.

The main tools for deriving these results have been the explicit formula for the solution of a weighted linear least-squares problem given by Dikin [8] and the relation between a projection onto a subspace with a certain weight matrix and the projection onto the orthogonal complement using the inverse weight matrix.

An interesting question that is left open is whether or not the explicit bounds that are given are sharp. In the case where all the matrices whose nonnegative linear

combination form the weight matrix are of rank one, the bounds are sharp. In the general case, this is an open question. On a higher level, an interesting question is whether the results of this paper can be utilized to give new complexity bounds for quadratic programming, analogous to the case of linear programming; see, e.g., Vavasis and Ye [27, section 9].

Appendix. Relationship to partitioned orthogonal matrices. In this appendix we review a result by Gonzaga and Lara [15] concerning diagonally weighted projections onto orthogonally complementary subspaces and combine this result with a result concerning singular values of submatrices of orthogonal matrices. It was these results in fact which lead to the more general results relating weighted projection onto a subspace and the projection onto its complementary subspace using the inverse weight matrix, as described in section 4.

Gonzaga and Lara [15] state that if Y is an $n \times m$ orthogonal matrix and Z is a matrix whose columns form an orthonormal basis for the null space of Y^T, then

$$\sup_{D \in \mathcal{D}_+} \|(Y^T D Y)^{-1} Y^T D\| = \sup_{D \in \mathcal{D}_+} \|(Z^T D Z)^{-1} Z^T D\|,$$

where $\mathcal{D}_+$ is the set of positive definite diagonal $n \times n$ matrices. They use a geometric approach to prove this result. We note that Corollary 4.2, specialized to the case of diagonal positive definite weight matrices, allows us to state the same result. Furthermore, we obtain an explicit expression for the supremum by Corollary 2.2. The following corollary summarizes this result.

COROLLARY A.1. *Suppose that an $n \times n$ orthogonal matrix Q is partitioned as $Q = (Z\ Y)$, where Y is an $n \times m$ matrix. Let $\mathcal{D}_+$ denote the set of diagonal positive definite $n \times n$ matrices. Then,*

$$\sup_{D \in \mathcal{D}_+} \|(Z^T D Z)^{-1} Z^T D\|_2 = \max_{J \in \mathcal{J}(Z^T)} \frac{1}{\sigma_{\min}(Z_J)}$$

$$= \sup_{D \in \mathcal{D}_+} \|(Y^T D Y)^{-1} Y^T D\|_2 = \max_{\tilde{J} \in \mathcal{J}(Y^T)} \frac{1}{\sigma_{\min}(Y_{\tilde{J}})},$$

where $\mathcal{J}(Z^T)$ is the collection of sets of column indices associated with nonsingular $(n-m) \times (n-m)$ submatrices of Z^T and $\mathcal{J}(Y^T)$ is the collection of sets of column indices associated with nonsingular $m \times m$ submatrices of Y^T.

Proof. Since $D \in \mathcal{D}_+$ if and only if $D^{-1} \in \mathcal{D}_+$, Corollary 4.2 shows that

$$\sup_{D \in \mathcal{D}_+} \|(Z^T D Z)^{-1} Z^T D\|_2 = \sup_{D \in \mathcal{D}_+} \|(Y^T D Y)^{-1} Y^T D\|_2.$$

The explicit expressions for the two suprema follow from Corollary 2.2. ∎

Hence, in our setting, we would rather state the result of Gonzaga and Lara [15] in the equivalent form

$$\sup_{D \in \mathcal{D}_+} \|(Y^T D Y)^{-1} Y^T D\| = \sup_{D \in \mathcal{D}_+} \|(Z^T D^{-1} Z)^{-1} Z^T D^{-1}\|$$

with the expressions for the suprema stated in Corollary A.1.

Note that an implication of Corollary A.1 is that if an $n \times n$ orthogonal matrix Q is partitioned as $Q = (Z\ \ Y)$ where Y has m columns, there is a certain relationship between the smallest singular value of all nonsingular $(n-m) \times (n-m)$ submatrices of Z and the smallest singular value of all nonsingular $m \times m$ submatrices of Y. This is

in fact a consequence of a more general result, namely, that if Q is partitioned further as

$$Q = \begin{pmatrix} Z_1 & Y_1 \\ Z_2 & Y_2 \end{pmatrix}, \tag{A.1}$$

where Z_1 is $(n-m) \times (n-m)$, then all singular values of Z_1 and Y_2 that are less than one are identical. This in turn is a consequence of properties of singular values of submatrices of orthogonal matrices that can be obtained by the CS-decomposition of an orthogonal matrix; see, e.g., Golub and Van Loan [14, section 2.6.4].

This result relating the singular values of Z_1 and Y_2 of (A.1) implies the existence of J and $\tilde{J}$, which are complementary subsets of $\{1, \ldots, n\}$, for which the maxima in Corollary A.1 are achieved. This observation lead us to the result that

$$\|(Y^T D Y)^{-1} Y^T D\|_2 = \|(Z^T D^{-1} Z)^{-1} Z^T D^{-1}\|_2$$

for any positive definite diagonal D. Subsequently, this result was superseded by the more general analysis presented in section 4.

Acknowledgments. We thank the two anonymous referees for their constructive and insightful comments, which significantly improved the presentation.

REFERENCES

[1] R. H. Bartels, G. H. Golub, and M. A. Saunders, *Numerical techniques in mathematical programming*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 123–176.

[2] A. Ben-Israel, *A volume associated with $m \times n$ matrices*, Linear Algebra Appl., 167 (1992), pp. 87–111.

[3] A. Ben-Tal and M. Teboulle, *A geometric property of the least squares solution of linear equations*, Linear Algebra Appl., 139 (1990), pp. 165–170.

[4] Å. Björck, *Pivoting and stability in the augmented system method*, in Numerical Analysis 1991, Proceedings of the 14th Dundee Conference, University of Dundee, UK, D. F. Griffiths and G. A. Watson, eds., Longman, Wiley, Harlow, UK, New York, 1991, pp. 1–16.

[5] Å. Björck and C. C. Paige, *Solution of augmented linear systems using orthogonal factorizations*, BIT, 34 (1994), pp. 1–24.

[6] E. Y. Bobrovnikova and S. A. Vavasis, *A norm bound for projections with complex weights*, Linear Algebra Appl., 307 (2000), pp. 69–75.

[7] I. I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[8] I. I. Dikin, *On the speed of an iterative process*, Upravlyaemye Sistemi, 12 (1974), pp. 54–60.

[9] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner, *The Factorization of Sparse Symmetric Indefinite Matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.

[10] A. Forsgren, *On linear least-squares problems with diagonally dominant weight matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 763–788.

[11] A. Forsgren, P. E. Gill, and J. R. Shinnerl, *Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 187–211.

[12] P. E. Gill, W. Murray, and M. H. Wright, *Numerical Linear Algebra and Optimization*, Vol. 1, Addison-Wesley, Redwood City, CA, 1991.

[13] P. E. Gill, M. A. Saunders, and J. R. Shinnerl, *On the stability of the Cholesky factorization for symmetric quasidefinite systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 35–46.

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

[15] C. C. Gonzaga and H. L. Lara, *A note on properties of condition numbers*, Linear Algebra Appl., 261 (1997), pp. 269–273.

[16] N. I. M. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*, Math. Program., 32 (1985), pp. 90–99.

[17] M. Gulliksson and P.-Å. Wedin, *Modifying the QR-decomposition to constrained and weighted linear least squares*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1298–1313.
[18] M. Hanke and M. Neumann, *The geometry of the set of scaled projections*, Linear Algebra Appl., 190 (1993), pp. 137–148.
[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[20] C. L. Lawson and R. J. Hanson, *Solving Least-Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[21] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part* II: *Convex quadratic programming*, Math. Program., 44 (1989), pp. 43–66.
[22] D. P. O'Leary, *On bounds for scaled projections and pseudoinverses*, Linear Algebra Appl., 132 (1990), pp. 115–117.
[23] H. L. Royden, *Real Analysis*, 3rd ed., Macmillan, New York, 1988.
[24] G. W. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra Appl., 112 (1989), pp. 189–193.
[25] M. J. Todd, *A Dantzig-Wolfe-like variant of Karmarkar's interior-point linear programming algorithm*, Oper. Res., 38 (1990), pp. 1006–1018.
[26] S. A. Vavasis, *Stable numerical algorithms for equilibrium systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1108–1131.
[27] S. A. Vavasis and Y. Ye, *A primal-dual interior point method whose running time depends only on the constraint matrix*, Math. Program., 74 (1996), pp. 79–120.
[28] M. Wei, *Upper bound and stability of scaled pseudoinverses*, Numer. Math., 72 (1995), pp. 285–293.
[29] S. J. Wright, *Stability of linear equations solvers in interior-point methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1287–1307.
[30] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1997.
[31] S. J. Wright, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 191–222.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 57–76
© 2001 Society for Industrial and Applied Mathematics

COMPUTING THE SOBOLEV REGULARITY OF REFINABLE FUNCTIONS BY THE ARNOLDI METHOD*

AMOS RON†, ZUOWEI SHEN‡, AND KIM-CHUAN TOH‡

Abstract. The recent paper [*J. Approx. Theory,* 106 (2000), pp. 185–225] provides a complete characterization of the L_2-smoothness of a refinable function in terms of the spectrum of an associated operator. Based on this theory, we devise in this paper a numerically stable algorithm for calculating that smoothness parameter, employing the deflated Arnoldi method to this end. The algorithm is coded in `Matlab`, and details of the numerical implementation are discussed, together with some of the numerical experiments. The algorithm is designed to handle large masks, as well as masks of refinable functions with unstable shifts. This latter case is particularly important, in view of the recent developments in the area of wavelet frames.

Key words. refinable functions, wavelets, smoothness, regularity, transition operators, transfer operators, Arnoldi's method

AMS subject classifications. Primary, 42C15; Secondary, 39B99, 46E35

PII. S0895479899363010

1. Introduction.

We are interested in the computation of the smoothness parameter of refinable functions. *Refinable functions* (known also as "scaling functions") are solutions of special functional equations that are known as *refinement equations*. The refinement equation expresses a dilate of the solution as the convolution product of that solution with a discrete kernel, the latter being known as the *mask* (cf. (2.2) for the precise definition). The smoothness of refinable functions is important in two subareas of analysis. In the area of *subdivision algorithms*, it determines the smoothness of the limit curve/surface of the subdivision process; in the area of *wavelets*, the smoothness of the refinable function is passed on to all wavelet systems that are derived from it (via the *multiresolution analysis* vehicle). In most practical cases, the refinable function is not known explicitly, and the available information consists, primarily, of the mask. Therefore, the determination of the smoothness of the solution from properties of the mask is one of the key problems in the above-mentioned areas.

Our efforts in this paper are focused on the study of the above problem via the *transfer/transition operator* approach. The analysis of the regularity of refinable functions in terms of the transfer operator was developed by several authors (cf., e.g., [D], [DD], [E], and [V] for the univariate case, [RiS], [CGV], [J], [LMW], [RS1], and [R2] for the multivariate case). In the L_2-case, the regularity estimates are in terms of a specific eigenpair of an associated transfer operator; hence, they seem to be computationally feasible. However, while the smoothness parameter of some examples was successfully computed by some authors (see, e.g., [HJ] and [RS1]), there has not

*Received by the editors October 29, 1999; accepted for publication (in revised form) by M. Hanke-Bourgeois September 5, 2000; published electronically May 3, 2001. This work was supported by the National Science Foundation under grants DMS-9626319 and DMS-9872890, by the U.S. Army Research Office under contracts DAAH04-95-1-0089 and DAAG55-98-1-0443, by the National Institute of Health, and by the Strategic Wavelet Program grant from the National University of Singapore.

http://www.siam.org/journals/simax/23-1/36301.html

†Computer Science Department, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, WI 53706 (amos@cs.wisc.edu).

‡Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260 (matzuows@math.nus.edu.sg, mattohkc@math.nus.edu.sg).

been (to best of our knowledge) a reliable (i.e., robust) numerical algorithm that works without significant restrictions on the mask.

Our method is based on the characterizations of the L_2-smoothness parameter given in [RS1], a detailed account of which is given in section 2. For the discussion here, it suffices to note that the characterization is given in terms of the restriction of a certain linear operator (the *transfer operator*) to a finite-dimensional invariant subspace H (the elements of H are trigonometric polynomials). In order to compute the smoothness using this approach, one has to overcome four different obstacles, two of which are of theoretical nature and the other two of numerical nature. First, one needs a characterization of the space H, a characterization that applies to a wide range of refinement equations; specifically, one should avoid restrictions on the refinement equations that either cannot be verified numerically or exclude examples of interest. Second, the characterization of the invariant space H must be computationally verifiable; we found that in most practical cases it is not feasible to compute a basis for H, hence one must have an alternative method for checking whether a given function belongs to that invariant space. That alternative method employs a superspace H_0 of H which is also an invariant subspace of the transfer operator, and which has an easily computable basis. The algorithm then finds in H_0 eigenvectors of the transfer operator, and uses a subtle criterion to determine whether the eigenvector found also lies in H. The success of this approach relies on the ability to recover accurately many eigenvectors, and not only few dominant ones. Thus, our third obstacle is the necessity of choosing and implementing carefully the eigensolver. Fourth, a direct implementation of the theory converts "small" problems (measured, say, in terms of the support of the mask) to a huge numerical mess, unless properly approached. For example, the matrix involved in computing one of the bivariate interpolatory refinable functions constructed in [RiS] has an order of about 4×10^3, leading, thereby, to a numerically prohibitive eigenproblem.

We present our algorithm and its implementation in four stages. In the first (section 2), we survey the results of [RS1] on the regularity of refinable functions, results that serve as the main stimulus for the present endeavor. As is seen there, the characterization of [RS1] can be implemented in many different ways, and we carefully devise in the second stage (section 3) what we consider to be the "winning algorithm" (designed to be fast for the average problem and robust for other cases). The algorithm requires a supplementary stable method for computing eigenvalues of linear operators. In the third stage of the presentation (section 4), we describe a variation of the Arnoldi method [A] that is used to that end, and provide a rough sketch of our `Matlab` code. We document in section 5 a sample of the numerical experiments. Finally, proofs of some results in section 3 are given in section 6.

One must keep in mind that it is rather hard to devise a good universal numerical algorithm for this problem since the numerical challenge in computing the smoothness has many, conflicting, faces. For example, in the construction of compactly supported bivariate interpolatory subdivision schemes, as well as in the related construction of certain orthogonal and biorthogonal refinable functions (see, e.g., [DGL], [DDD], [CD], [CS], [RiS], [JRS], [HL], [HJ], [KS], [BW]), one expects to have a relatively large mask, hence one has to cope with the sheer size of the problem. In contrast, in the theory of *wavelet frames*, and in the subsequent constructions of tight wavelet frames and bi-frames, (cf. [RS2], [RS3], [RS4], [RS5] and in particular [GR]), good wavelet systems (e.g., tight frames), are derived from a multiresolution analysis based on a refinable function with unstable shifts. While that refinable function may be very at-

tractive for applications (having many alternative properties such as high smoothness, good approximation order, and small support), the problem of finding its smoothness without the stability assumption is a theoretical challenge (which was overcome for the first time in [RS1]) and is also a computational challenge.

2. The Sobolev regularity of refinable functions. Since the main objective of this paper is to convert (some of) the results in [RS1] from theory to practice, we naturally review first the pertinent results of that paper. The presentation here is confined to the setup of the present paper. We consider here only *scalar* refinable functions (PSI case) in one or two variables whose refinement masks are finitely supported. (The characterizations of [RS1] apply to the vector (FSI) case, to any number of dimensions, and do not assume the mask to be finitely supported.) A complete list of the assumptions made in this paper is provided in what follows.

Let s be a $d \times d$ integer matrix that satisfies

$$s^* s = \lambda^2 I \tag{2.1}$$

for some $\lambda > 1$. We refer to such a matrix s as a *dilation matrix* or, more precisely, as an *isotropic dilation matrix*. Let ϕ be a compactly supported L_2-function in d variables (or, more generally, a compactly supported distribution). We say that ϕ is *refinable with respect to the dilation matrix* s if there exists a finitely supported sequence a such that

$$\phi(x) = |\det s| \sum_{j \in \mathbb{Z}^d} a(j)\phi(sx - j), \quad x \in \mathbb{R}^d. \tag{2.2}$$

The equivalent formulation of this condition on the Fourier domain is

$$\widehat{\phi}(s^* \cdot) = \widehat{a}\widehat{\phi}, \tag{2.3}$$

with $\widehat{a}$ the symbol of the sequence a, i.e.,

$$\widehat{a}(\omega) = \sum_{j \in \mathbb{Z}^d} a(j) \exp(-ij\omega).$$

The sequence a (as well as its symbol $\widehat{a}$) is called the *refinement mask* of ϕ. The *L_2-regularity parameter* $\alpha(\phi)$ of ϕ is defined by

$$\alpha(\phi) := \sup\{\alpha \in \mathbb{R} : \ \phi \in W_2^\alpha(\mathbb{R}^d)\}.$$

Here, $W_2^\alpha(\mathbb{R}^d)$ is the usual Sobolev space. For the more general nonisotropic dilation, the analysis in [RS1] provides only upper and lower bounds on the regularity parameter. Moreover, most of the interesting refinable functions correspond to isotropic dilation matrices, whence our decision to consider only isotopic dilations.

As it turns out, the regularity of ϕ is determined by properties of a related function known as *the autocorrelation function* $\phi^\#$ of ϕ, and which is defined as follows:

$$\phi^\# : t \mapsto \int_{\mathbb{R}^d} \phi(x)\phi(x - t)\, dx.$$

It is easy to see that the Fourier transform of $\phi^\#$ is $|\widehat{\phi}|^2$. Hence, $\phi^\#$ is refinable with mask

$$\widehat{b} := |\widehat{a}|^2.$$

The 2π-periodization of the Fourier transform of $\phi^\#$, i.e., the $L_1(\mathbb{T}^d)$-function

$$\Phi := \sum_{j\in\mathbb{Z}^d} |\widehat{\phi}(\cdot + 2\pi j)|^2, \tag{2.4}$$

plays a pivotal role in our discussion. Since $\phi^\#$ is compactly supported (by the fact that ϕ is), the Poisson summation formula implies that Φ is a trigonometric polynomial whose spectrum (i.e., frequencies) in the set

$$(\operatorname{supp}\phi^\#)\cap\mathbb{Z}^d = \{x-y\in\mathbb{Z}^d : x,y\in\operatorname{supp}\phi\}. \tag{2.5}$$

Next, we define the transfer operator. Let Γ be any representer set of the quotient group $2\pi(s^{*-1}\mathbb{Z}^d/\mathbb{Z}^d)$. The *transfer* or *transition operator* T is defined as

$$T : L_2(\mathbb{T}^d)\mapsto L_2(\mathbb{T}^d) : f\mapsto \sum_{\gamma\in\Gamma}(\widehat{b}f)(s^{*-1}\cdot+\gamma). \tag{2.6}$$

For example, if the spatial dimension is 1, and the dilation is dyadic (i.e., $s=2$), Γ can be chosen as $\{0,\pi\}$, and T becomes

$$(Tf)(\omega) = (\widehat{b}f)\left(\frac{\omega}{2}\right) + (\widehat{b}f)\left(\frac{\omega}{2}+\pi\right).$$

As was already alluded to in the introduction, the L_2-smoothness of ϕ is characterized by the spectral radius of the restriction of T to a certain invariant space H (of T), with H finite dimensional and consisting of trigonometric polynomials. In general, the space H does not have a simple structure. As a first step, we would like to construct a finite dimensional superspace of H (made also of trigonometric polynomials) which on one hand will be T-invariant, while, on the other hand, will have a simple structure.

To this end, let

$$\mathcal{Z}_\phi \;:=\; \{j\in\mathbb{Z}^d \,:\, \|j\|_2\le r\},$$

where r is any (fixed) number larger than or equal to

$$\frac{1}{\lambda-1}\,\max\{\|j\|_2 \,:\, b_j\neq 0\}$$

with λ defined in (2.1) and with (b_j) being the mask coefficients of the autocorrelation function. Then, since $\|s^{*-1}x\|_2 = \frac{1}{\lambda}\|x\|_2$, the space

$$H_\phi \tag{2.7}$$

of all trigonometric polynomials whose band lies in that set (i.e., the space spanned by the exponentials $\exp(ij\cdot)$, $j\in\mathcal{Z}_\phi$) is a T-invariant subspace, and all eigenvectors of T that are trigonometric polynomials must lie in H_ϕ. Moreover, given *any* trigonometric polynomial f, we have that $T^k f\in H_\phi$ for all sufficiently large k (see [LLS1]). This last property implies that H_ϕ must contain each eigenvector f of T, provided that f is a trigonometric polynomial, and that its associated eigenvalue is nonzero. We use these basic facts in what follows without further notice.

Theorem 2.2 of [RS1] states that the regularity parameter $\alpha(\phi)$ of ϕ is

$$\alpha(\phi) = -\frac{\log_\lambda\rho}{2},$$

where λ is given by (2.1), and $\rho = |\mu|$ with μ an eigenvalue of the transfer operator (and with the associated eigenvector being a trigonometric polynomial). Hence, the key to the numerical computation of the regularity parameter $\alpha(\phi)$ is to compute this eigenpair (μ, f_μ) of T. We will describe in this paper a reliable and numerically stable algorithm that computes this eigenpair of T, and which thereby finds $\alpha(\phi)$. The algorithm is based on the characterization of ρ as the spectral radius of the restriction of T to H, with H a certain T-invariant subspace (that is defined below) of H_ϕ. One should note that H, as any subspace of H_ϕ, consists of trigonometric polynomials, each of which can be finitely represented in terms of its Fourier coefficients. However, in order to compute ρ directly from the above description, we also need a robust method for constructing a basis for H; since the methods we could find for constructing a basis for H are highly unstable, we will study the action of T on the larger space H_ϕ, and we will actually find ρ by other means. But, first, we recall the description of the space H from [RS1].

The space H is defined as $H := H_\phi \cap I_\phi$, with I_ϕ an ideal of trigonometric polynomials defined below. To this end, we set Π for the space of all d-variate (algebraic) polynomials, and Π_ϕ for the following subspace of it:

$$\Pi_\phi := \left\{ p \in \Pi : \sum_{j \in \mathbb{Z}^d} p(j)\phi^\#(\cdot - j) \in \Pi \right\}.$$

DEFINITION 2.1 (the ideal I_ϕ). *Let ϕ be a compactly supported L_2-function with $\widehat{\phi}(0) \neq 0$. Let $\phi^\#$ be the autocorrelation function of ϕ and let Φ be the 2π-periodization of the Fourier transform of $\phi^\#$ as given in* (2.4). *The ideal I_ϕ is the collection of all trigonometric polynomials (in $L_2(\mathbb{T}^d)$) f that satisfy*

(i) *$f/\Phi \in L_\infty(\mathbb{T}^d)$;*

(ii) *f is annihilated by Π_ϕ in the sense that $p(-iD)f(0) = 0$ for all $p \in \Pi_\phi$. Here $D = \frac{\partial}{\partial\omega_1 \ldots \partial\omega_d}$, i.e., $p(D)$ is the constant coefficient differential operator associated with the polynomial p.*

With the definition of I_ϕ, the results of [RS1] that are used in the present paper for computing the regularity parameter $\alpha(\phi)$ are summarized as follows.

RESULT 2.2. *Let ϕ be a compactly supported refinable function corresponding to the isotropic dilation matrix s with $\widehat{\phi}(0) \neq 0$ and let T be its associated transfer operator. Further, let the space H_ϕ and the ideal I_ϕ be given as in* (2.7) *and Definition* 2.1, *respectively. Then*

(i) *I_ϕ is T-invariant;*

(ii) *the regularity parameter $\alpha(\phi)$ is*

$$-(\log_\lambda \rho)/2,$$

where ρ is the maximal modulus of the eigenvalues of the restriction of T to I_ϕ;

(iii) *for ρ in* (ii), *there is an eigenpair (μ, f) of T such that $\rho = |\mu|$ and $f \in H_\phi \cap I_\phi$.* □

Indeed, the T-invariance of I_ϕ is proved in Theorem 2.4 of [RS1]. That theorem also shows that the regularity parameter $\alpha(\phi)$ is determined by any dominant eigenpair (μ, f) of T restricted on $H_\phi \cap I_\phi$, in the sense that $\alpha(\phi) = -(\log_\lambda |\mu|)/2$. This gives (ii). Recalling that all the T-eigenvectors in I_ϕ are either in H_ϕ or in $\ker T$, we get (iii).

3. An algorithm for computing the regularity parameter. Result 2.2 suggests that in order to compute the regularity parameter of the refinable function, we "merely" need to find the spectral radius of the restriction of T to H, where $H = H_\phi \cap I_\phi$. However, the result cannot be implemented directly, due to the fact that there is no "good" method for constructing a basis for H.

Before we advance the discussion any further, we seek the following "terminological relief": from now on, given any linear space S, and any linear bounded operator T from S into a superspace of it, the notion of *the spectral radius of* T is meant as *the spectral radius of the restriction of* T *to the largest* T*-invariant subspace of* S.

Result 2.2 suggests the following "direct algorithm": Given the transfer operator T associated with the compactly supported refinable ϕ, a simple method for computing $\alpha(\phi)$ is as follows: (i) Choose a T-invariant superspace H_0 of $H_\phi \cap I_\phi$ (one which is convenient for computations). (ii) Find all eigenvalues ν of $T_{|_{H_0}}$. (iii) For each eigenvalue ν, find the corresponding eigenspace V_ν, then check whether $V_\nu \cap I_\phi \neq 0$. (iv) The desired ρ is $\max\{|\nu| : \ V_\nu \cap I_\phi \neq 0\}$.

Various improvements of this direct algorithm are possible. The most obvious one is to avoid finding all the eigenvalues ((ii) above), and instead finding them one by one in decreasing the modulus of the eigenvalue; stop when the first eigenvector in I_ϕ is found. That approach suits the Arnoldi method of computing eigenvalues and eigenvectors. However, even with that improvement, the above "direct method" suffers from the following drawbacks: (a) If the critical eigenpair (μ, f_μ) is preceded by many other eigenpairs (whose eigenvalues have greater magnitudes), the approximation provided by the Arnoldi method for the critical eigenvector f_μ may be crude, and it may be hard to determine numerically whether $f_\mu \in I_\phi$. (b) The necessity to compute a bulk of eigenpairs makes the process relatively slow. (c) Even if the eigenvector is computed with high accuracy, it may still be hard to determine whether it belongs to I_ϕ. This problem (which exists in other approaches too, but to a lesser extent) is particularly troubling in the case of a multiple eigenvalue, since then we must check whether I_ϕ has a nonzero intersection with the eigenspace, a task which is almost always a numerical challenge (unless the eigenspace lies entirely in I_ϕ).

The above discussion reveals the following three different aspects that a successful algorithm has to deal with:

Aspect I: The eigenproblem aspect. We need to recover an eigenpair of a linear operator. The eigenpair that we look for may be dominated by many other pairs; nonetheless, we need a fast and accurate recovery of the eigenpair. It would be best if all/many/most of the eigenpairs that dominate the critical one can somehow be avoided. Standard variations of the power method (such as the shifted inverse power method) require an estimate of the critical eigenvalue, an estimate that is not available here. A fast implementation also requires a savvy conversion of the problem to matrix computations.

Aspect II: Φ and Π_ϕ. One of the key steps in any algorithm that computes the regularity parameter is to determine whether a given trigonometric polynomial f is in I_ϕ. For this, one needs to (1) find the polynomial Φ, and (2) find the space Π_ϕ (see the definition of the ideal I_ϕ). The first task is relatively modest: once we adopt a mild assumption (the *E-condition*; see below), it becomes truly simple to compute Φ accurately. As to the second task, viz., computing a basis for Π_ϕ, it is hampered by the fact that Π_ϕ, in general, does not have a simple structure (e.g., may not have a monomial basis), which makes it "unpleasant" even under some additional conditions (e.g., stability). To overcome this difficulty, we use subtle theoretical facts

that allow us to get away with only partial computation of Π_ϕ. Moreover, under "favorable conditions" (which are far less demanding than stability), the approach yields a substantial shortcut in the search of the critical eigenvalue.

Aspect III: Testing a given eigenvector. In order to check whether a given eigenvector f is in I_ϕ, one needs to check whether both (i) and (ii) in the definition of I_ϕ are satisfied. As we will see, the algorithm used here frees us from checking the second condition in the definition of I_ϕ. Furthermore, when the trigonometric polynomial Φ is positive *everywhere* (a condition which is known as "the stability of the shifts of ϕ"), the first condition in the definition of the ideal I_ϕ is automatically satisfied. Hence, under this stability assumption, the process of checking whether the eigenvector in hand is in I_ϕ is fast and very robust. Without the stability assumption, we have to check whether f/Φ is bounded or not. This problem is on par with the classical NA problem: determining whether a small number is 0 or not. As said, this problem is particularly acute for multiple eigenvalues.

The first and third aspects above are problems that belong to the area of numerical algebra, and we will discuss them in the next section as a part of the discussion on the implementation and the code. To have an optimal treatment of the second aspect, we need some additional discussion concerning the regularity of refinable functions (beyond the general discussion of the previous section).

The discussion is divided into two parts: the first is about the computation of Φ and the second deals with Π_ϕ.

Computing the trigonometric polynomial Φ. We start with a finitely supported mask a. For a given mask, we want to know whether there exists a compactly supported solution to the corresponding refinement equation, and if there is a solution, whether the solution is unique and whether the solution is in L_2. The following result provides satisfactory answers.

RESULT 3.1. *Let a be a finitely supported mask, and let T be the associated transfer operator.*

(i) *If $\sum_{\alpha\in\mathbb{Z}^d} a(\alpha) = 1$ (i.e., $\widehat{a}(0) = 1$), there exists a compactly supported distribution ϕ that solves the refinement equation. It is the unique solution that satisfies $\widehat{\phi}(0) = 1$.*

(ii) *If the restriction of T to H_ϕ has spectral radius 1, and if all the eigenvalues (of that restriction) that lie on the unit circle are nondefective, then the corresponding solution of the refinement equation must lie in L_2.*

(iii) *If the solution ϕ of the corresponding refinement equation is in L_2, then $(1, \Phi)$ is an eigenpair of T.*

The first statement is proved by showing that the infinite expansion $\prod_{j=1}^{\infty} \widehat{a}(s^{*-j}\omega)$ converges, uniformly on compact sets, to a tempered distribution. The last assertion is a straightforward exercise. The proof of the second assertion can be found in [LLS2] as well as in [R2].

COROLLARY 3.2. *Let a be a finite mask satisfying $\widehat{a}(0) = 1$. Assume that the corresponding transfer operator T satisfies* (ii) *of Result* 3.1. *Then T must have an eigenpair $(1, f)$, with f a nonnegative trigonometric polynomial.*

The condition that appears in part (ii) of Result 3.1 is not necessary for the solution ϕ to be in L_2 (cf. [RS1]), but refinable functions whose transfer operator violates this condition are quite "pathological." In our algorithm, we assume a bit more, namely that the eigenvalue 1 is *simple*.

Definition: The weak E-condition. Let a be a given finite mask with $\widehat{a}(0) = 1$ and let ϕ be the corresponding compactly supported solution. Let T be the transfer

operator associated with ϕ. We say that a (or ϕ, or T) satisfies the weak E-condition if the restriction of T to H_ϕ has spectral radius 1, all the eigenvalues on the unit circle are nondefective, and 1 (which is then necessarily an eigenvalue) is a simple eigenvalue.

Remark. The previous discussion implies that, under the weak E-condition, the refinement equation has a unique compactly supported solution, ϕ, that lies in L_2 and satisfies $\widehat{\phi}(0) = 1$. Further, Φ (i.e., the 2π-periodization of the Fourier transform of the autocorrelation of the solution) is the unique eigenvector (up to a constant) of the eigenvalue 1 of the transfer operator.

Remark. If we add to the weak E-condition the additional assumption that $T_{|_{H_\phi}}$ has a *unique* dominant eigenvalue, we obtain a condition known as the E-condition (which is useful in the analysis of various problems; for example, [LLS2] proves that the E-condition characterizes the L_2-convergence of the cascade algorithm; see also section 3.1 of [R2]). This explains our usage of "weak E-condition." Finally, we point out that it is not difficult to show that if T satisfies the weak E-condition on H_ϕ, then it satisfies weak E-condition on any T-invariant superspace H_0 of H_ϕ that consists of trigonometric polynomial (see [LLS2]).

In the first step of the algorithm, we select a convenient T-invariant superspace H_0 of H_ϕ. Then, the algorithm checks whether T satisfies the weak E-condition. If the weak E-condition is satisfied, it computes the eigenvector associated with the eigenvalue 1. The (normalized) symbol of that eigenvector is the function Φ.

Doing without $\boldsymbol{\Pi_\phi}$. We now elaborate on the second condition in the definition of I_ϕ. Let $\mathcal{Z}$ be some finite, fixed, subset of $\mathbb{Z}^d$, and let $H_\mathcal{Z}$ be the span of the exponentials $\omega \mapsto \exp(ij \cdot \omega)$, $j \in \mathcal{Z}$, endowed with the $L_2(\mathbb{T}^d)$-inner product $\langle \cdot, \cdot \rangle$. Given any (algebraic) polynomial p and any $f \in H_\mathcal{Z}$, one observes that $p(-iD)f(0) = \sum_{j\in\mathcal{Z}} p(j)f_j$, with $(f_j)_j$ the Fourier coefficients of f. Hence, the linear functional (in $H_\mathcal{Z}^*$) $f \mapsto p(-iD)f(0)$ is represented by the trigonometric polynomial $t_p(\omega) := \sum_{j\in\mathcal{Z}} p(j)\exp(ij\cdot\omega)$, i.e.,

$$p(-iD)f(0) = \langle t_p, f\rangle = \sum_{j\in H_\mathcal{Z}} p(j)f_j \quad \forall f \in H_\mathcal{Z}. \tag{3.1}$$

We now connect the above abstract discussion to our concrete problem. In this discussion, we use, for a given subspace $Q \subset \Pi$ of algebraic polynomials, the notation

$$P_Q$$

for the orthogonal projector from $H_\mathcal{Z}$ onto $\{t_p : p \in Q\}$.

When $\mathcal{Z}$ above is $\mathcal{Z}_\phi$ (see (2.5)), the space $H_\mathcal{Z}$ becomes H_ϕ (of (2.7)). Furthermore, by choosing Q above to be Π_ϕ, the second condition in the definition of I_ϕ simply says that the critical eigenvector lies in the orthogonal complement (in H_ϕ) of $\{t_p : \ p \in \Pi_\phi\}$. Thus, if we set

$$P_\phi := P_{\Pi_\phi}$$

for the orthogonal projection of H_ϕ onto $\{t_p : \ p \in \Pi_\phi\}$, condition (ii) in the definition of I_ϕ will be automatically satisfied if we iterate (in the search for the critical eigenvector) with the operator $(1 - P_\phi)T$, instead of iterating with the transfer operator itself. This allows us to restate Result 2.2 in the following equivalent (yet more practical) way.

Restatement of Result 2.2. In the notations and assumptions of Result 2.2, let I'_ϕ be the ideal of all trigonometric polynomials of the form $t\Phi$, $t \in L_\infty(\mathbb{T}^d)$ (i.e., those that satisfy the first condition in the definition of I_ϕ). Then the spectral radius ρ in Result 2.2 is the same as the spectral radius of the restriction of $(1-P_\phi)T$ to $H_\phi \cap I'_\phi$.

The discussion still leaves us with the need of finding a basis for Π_ϕ (in order to compute the projector P_ϕ). As we alluded to before, this can be partially circumvented: suppose that Q is some subspace of Π_ϕ, and suppose that we replace the operator $(1-P_\phi)T$ by the operator $(1-P_Q)T$. The latter one will fail to suppress some of the eigenvalues that the former one does; however, that apparent fault is harmless if we know that all these "unsuppressed" eigenvalues are smaller than the critical one. But do we have such a space Q, which, in addition, has a simple basis?

In order to answer the above question, we define

$$m_\phi := \max\{m \in \mathbb{N} : \Pi_m \subset \Pi_\phi\}, \tag{3.2}$$

where Π_m is the space of d-variate algebraic polynomials with degree $\le m$. We will show that we can replace the space Π_ϕ by the space Π_{m_ϕ}, and, moreover, we can sometimes do with Π_m for $m < m_\phi$. In addition, we show a way to compute m_ϕ from the given data, viz., the mask a and the trigonometric polynomial Φ. We begin with that latter issue.

PROPOSITION 3.3. *Let ϕ be a refinable compactly supported L_2-function with mask a. Let $\Gamma = 2\pi(s^{*-1}\mathbb{Z}^d/\mathbb{Z}^d)$. Then $\Pi_m \subset \Pi_\phi$ if and only if $\widehat{b}\Phi$ $(= |\widehat{a}|^2\Phi)$ has a zero of order $m+1$ at each of the points in $\Gamma\backslash 0$.*

The spaces Π_m, $m \le m_\phi$, are certainly subspaces of Π_ϕ and have a simple structure. The next result studies the suitability of the choice $Q := \Pi_m$. For notational convenience we set, for any nonnegative integer m,

$$P_m := P_{\Pi_m}.$$

PROPOSITION 3.4. *Let ϕ be a refinable function with corresponding mask a and transfer operator T. Let ρ_m be the spectral radius of the restriction of $(1-P_m)T$ to $H_\phi \cap I'_\phi$. Then*

(a) $\rho_{m_\phi} = \rho$;

(b) *for an odd $m \le m_\phi$, we still have $\rho_m = \rho$, unless $\rho_m \le \lambda^{-m-1}$.*

We prove the above propositions in the last section; hence, we are ready to present here our algorithm.

Algorithm: Step I. Compute the T-invariant space H_ϕ. Then, check whether T satisfies (on H_ϕ) the weak E-condition. If 1 is not an eigenvalue of T, return the message ``There is no L_2-solution to the refinement equation" and quit. If, otherwise, the weak E-condition is still violated, give another appropriate rejection message (that indicates that the solution may still not be in L_2) and quit. If the weak E-condition is satisfied, compute the eigenvector associated with the eigenvalue 1. Check (for consistency only) that the eigenvector is nonnegative (or nonpositive). The (normalized) symbol of that eigenvector is the function Φ.

Algorithm: Step II. Set $m_{\phi,\gamma}+1$ to be the order of the zero that $|\widehat{a}|^2\Phi$ has at γ, and set

$$m_\phi := \min\{m_{\phi,\gamma} : \gamma \in \Gamma\backslash 0\}.$$

Algorithm: Step III. Find the eigenpairs (in H_ϕ) of $(1-P_{\Pi_{m_\phi}})T$, one by one, ordered according to the eigenvalue modulus. Stop when finding the first eigenpair (μ, f_μ) for which f_μ/Φ is bounded. The L_2-regularity of $\alpha(\phi)$ is then $-\frac{\log_\lambda(|\mu|)}{2}$.

Remark. We note that no differentiation is really conducted in Step II. Instead, one uses the fact that

$$p(-iD)f(\gamma) = \langle p\,\exp(i\gamma\cdot), f\rangle$$

(compare with 3.1). Further, since the maximal order of zeros of $\widehat{b}\Phi$ is even, m_ϕ is odd. □

Remark. We note that if Φ does not vanish at $\Gamma\backslash 0$, then the space $(1-P_{\Pi_{m_\phi}})H_\phi$ is T-invariant. In contrast, if Φ vanishes at a point of $\Gamma\backslash 0$, $(1-P_{\Pi_{m_\phi}})H_\phi$ may not be T-invariant anymore. Nonetheless, Proposition 3.4 always holds. Its proof relies on the fact that the subspace $(1-P_{\Pi_{m_\phi}})(H_\phi\cap I_\phi)$ is *always* T-invariant. □

The algorithm checks for possible shortcuts: Stability. In many cases of interest, the shifts of the refinable function are *stable*. A convenient way to define the stability here (which is entirely equivalent to the more standard definitions) is that $\Phi > 0$ (everywhere). Since our algorithm computes Φ in any event, it checks whether Φ is everywhere positive. In that event, it performs two shortcuts. The major one is that the first condition in the definition of I_ϕ becomes *superfluous*, and hence the iterations with $(1-P_{\Pi_{m_\phi}})T$ search for a dominant eigenvalue. This not only accelerates the algorithm, but also results in a dramatic improvement of its numerical stability. Indeed, in this case we do not need to determine whether a large value of f/Φ should be interpreted as finite or infinite. Note that, since Φ is the dominant eigenvector, we are able to compute Φ with great accuracy. Hence, it is possible to have a stable numerical algorithm to check whether $\Phi > 0$.

In the case of stability, another, less important, shortcut occurs: in the computation of m_ϕ, we look in general for the order of the zeros of $\widehat{b}\Phi$ on $\Gamma\backslash 0$. If Φ vanishes nowhere, these zeros coincide with those of $\widehat{b}$, and we do not need to compute $\widehat{b}\Phi$ (i.e., to convolve their Fourier coefficients.) For that shortcut, we need only Φ to be nonzero on $\Gamma\backslash 0$ (and indeed we implement that shortcut under that mere latter condition).

4. Numerical implementation details. In the actual numerical implementation, we treat the transfer operator as acting on *sequences*, i.e., we use the operator $\mathcal{T}$ defined by

$$\mathcal{T}c := (T\widehat{c})^\vee,$$

where $f^\vee$ is the inverse Fourier transform of f. The sequence c is always defined on $\mathbb{Z}^d$ and has finite support. We use the pairing

$$\langle\theta, c\rangle := \sum_{j\in\mathbb{Z}^d}\theta(j)\overline{c(j)}, \tag{4.1}$$

in which c is finitely supported and θ is any sequence defined on $\mathbb{Z}^d$, or more generally, a function in $C(\mathbb{R}^d)$.

Next, we provide some details about the steps in the algorithm given in the previous section.

For the first step, we find the set $\mathcal{Z}_\phi$ as in section 2. We then compute, via the deflated Arnoldi method, a basis for the dominant eigenspace of the transfer operator $\mathcal{T}: \ell_2(\mathcal{Z}_\phi) \longrightarrow \ell_2(\mathcal{Z}_\phi)$. Then, we check whether the transfer operator $\mathcal{T}$ satisfies the weak E-condition. If the weak E-condition is satisfied, we compute the eigenvector

corresponding to the eigenvalue 1: its Fourier series is the function Φ; else, the weak E-condition is violated, and we quit.

For the second step, we first check whether Φ vanishes on $\Gamma\backslash\{0\}$. If it does not, we find the largest integer m such that

$$\langle \exp(i\gamma\cdot)\, p, a\rangle = 0 \qquad \forall\, p \in \Pi_m,\ \gamma \in \Gamma\backslash\{0\}, \tag{4.2}$$

where a is the refinement mask of ϕ. We then set $m_\phi := 2m+1$. If Φ vanishes on $\Gamma\backslash\{0\}$, then we find the largest integer m such that

$$\langle \exp(i\gamma\cdot)\, p, c\rangle = 0 \qquad \forall\, p \in \Pi_m,\ \gamma \in \Gamma\backslash\{0\}, \tag{4.3}$$

where $c = b * h$, b is the mask of the autocorrelation function $\phi^\#$, and h is the Fourier coefficients of Φ, i.e., $\widehat{h} = \Phi$. We then set $m_\phi := m$.

For these, it is sufficient to check that (4.2) or (4.3) holds for a basis of Π_m. However, it is important to choose a well-conditioned basis. The usual monomial basis of Π_m is very ill conditioned, and therefore is inappropriate for our purpose. We choose here instead a suitable orthonormal basis. That orthonormal basis is described in what follows.

For the third step, if Φ vanishes nowhere, we compute the dominant eigenvalue μ of $(I - P_{\Pi m_\phi})T$ via the deflated Arnoldi method as detailed below. Then, set $\alpha(\phi) = -\frac{\log_\lambda |\mu|}{2}$.

If Φ vanishes anywhere (in $[-\pi,\pi]^d$ as this function is 2π-periodic), then we proceed as follows.

(i) We compute the next group of the distinct dominant eigenvalues of $(I - P_{\Pi m_\phi})T$ via the deflated Arnoldi method. Then we order the eigenvalues according to decreasing magnitudes of their values as

$$|\mu_1| \ge |\mu_2| \ge \cdots.$$

(ii) We compute a basis for the eigenspace associated with each of the eigenvalues computed in (i) via the deflated Arnoldi method. Denote them as $\{f_1, \ldots, f_L\}$.

(iii) If there exists scalars $t_1, \ldots, t_L$ not all zero such that

$$\sum_{i=1}^{L} t_i \widehat{f_i}/\Phi$$

is bounded, then set $\alpha(\phi) = -\frac{\log_\lambda |\mu_k|}{2}$; stop. Otherwise, go back to step (i).

We discuss now the following numerical methods used to implement the algorithm.

The action of $\mathcal{T}$ on a vector. Let c be an arbitrary sequence in $\ell_2(\mathcal{Z}_\phi)$. The action of the transfer operator $\mathcal{T}$ on c is as follows. First, generate a new sequence $b*c$ by convolution, then reparameterize the sequence $(b*c)_{j\in\mathbb{Z}^d}$ to a sequence defined on $s^{-1}\mathbb{Z}^d$. Finally, the image $\mathcal{T}c$ is the restriction to $\mathbb{Z}^d$ of the sequence $(b*c)_{j\in s^{-1}\mathbb{Z}^d}$. The resulting sequence $\mathcal{T}c$ is still supported in $\mathcal{Z}_\phi$. Once $\mathcal{T}c \in \ell_2(\mathcal{Z}_\phi)$ is obtained, it is relatively easy to compute orthogonal projections of it onto various subspaces, provided that we also have an orthonormal basis for these subspaces.

Construction of an orthonormal basis for Π_n. The standard construction of an orthonormal basis (ON) for Π_m is done by applying the Gram–Schmidt process to the monomial basis $\{(j^\beta)_{j\in\mathcal{Z}_\phi} : |\beta| \le m\}$. However, this standard construction is numerically unstable. A more stable process (known as the modified Gram–Schmidt)

can be devised by modifying the Gram–Schmidt process, which we describe now in the bivariate case. Set $N := \#\mathcal{Z}_\phi$.

Modified Gram–Schmidt:

Let $v^{(0,0)} = \frac{1}{\sqrt{N}}(1)_{j\in\mathcal{Z}_\phi}$.
for $k = 1, 2, \dots, m$
  for $\beta_1 = 0, 1, \dots, k$
    if $\beta_1 = 0$
      $w = (j(2)\, v^{(0,k-1)}(j))_{j\in\mathcal{Z}_\phi}$.
    else
      $w = (j(1)\, v^{(\beta_1-1,k-\beta_1)}(j))_{j\in\mathcal{Z}_\phi}$.
    Orthogonalize w against all previously generated ON vectors v to get w'.
    Set $v^{(\beta_1,k-\beta_1)} = w'/\|w'\|_2$.

Let

$$B_m := \{v^{(\beta)} : |\beta| \le m\}. \tag{4.4}$$

Now, we describe here how to apply the deflated Arnoldi method [S] to our case. The method may not be as robust as other more sophisticated methods for the same purpose, such as the implicitly restarted Arnoldi [LSVY], [LS], the Jacobi–Davidson method [SV], and the truncated RQ iteration [SY]. Nonetheless, as our examples in the next section show, even with this relatively simple method, our proposed algorithm works well. Of course, for a more robust implementation, one should replace the deflated Arnoldi method by one of the more robust dominant eigenspace solvers just mentioned.

The deflated Arnoldi method. We first note that the operator $(I - P_\phi)\mathcal{T}$ can be viewed as an operator on $\mathbb{R}^N$ with $N = |\mathcal{Z}_\phi|$; we just need to order the points in $\mathcal{Z}_\phi$, and identify $\ell_2(\mathcal{Z}_\phi)$ with $\mathbb{R}^N$, the latter equipped with the standard inner product on $\mathbb{R}^N$. Let A be an arbitrary linear endomorphism of $\mathbb{R}^N$. The deflated Arnoldi method is described in the following steps:

(1) Choose an initial vector $v_1 \in \mathbb{R}^N$ with $\|v_1\|_2 = 1$. Set $k = 1$. Select the number m of Arnoldi iterations to be performed in each pass.

(2) Arnoldi iteration:
  for $j = k, k+1, \dots, m$
  compute $w = Av_j$
  for $i = 1, 2, \dots, j$
    $h_{ij} = \langle w, v_i\rangle$
    $w = w - h_{ij}v_i$
  $h_{j+1,j} = \|w\|_2$
  $v_{j+1} = w/h_{j+1,j}$.

Let V_m be the matrix whose kth column is the vector v_k and $H_m = (h_{ij})$ be the $m \times m$ upper Hessenberg matrix constructed above. The vectors v_j generated by the Arnoldi iteration satisfy the following relation:

$$AV_m = V_m H_m + h_{m+1,m}\, v_{m+1} e_m^T.$$

Suppose (μ, y) is an eigenpair of H_m. Then $(\mu, V_m y)$ is an approximate eigenpair of A.

(3) Compute approximate eigenvectors $y_1, y_2, \dots, y_t$, associated with the dominant eigenvalues $\mu_1, \mu_2, \dots, \mu_t$ of H_m. Compute the residual norms $\rho_k =$

$\|AV_m y_k - \mu V_m y_k\|_2$ for $k = 1, \ldots, t$. If $y_{i_1}, y_{i_2}, \ldots, y_{i_r}$ (where $r \le t$) are the vectors such that the corresponding residual norms are small enough, then $u_{i_1} = V_m y_{i_1}, u_{i_2} = V_m y_{i_2}, \ldots, u_{i_r} = V_m y_{i_r}$ are converged approximate eigenvectors of A associated with the dominant eigenvalues $\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_r}$.

(4) *Deflation*: Suppose $y_{i_1}, y_{i_2}, \ldots, y_{i_r}$ are eigenvectors of H_m corresponding to converged eigenvectors $u_{i_1}, u_{i_2}, \ldots, u_{i_r}$ of A associated with the dominant eigenvalues $\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_r}$. This step is to deflate these converged eigenvectors from the Arnoldi iteration so that additional eigenvectors of A associated with these dominant eigenvalues can be found, whenever they exist.

(i) Compute the QR factorization of the matrix $(y_{i_1}, \ldots, y_{i_r})$ using Householder matrices:

$$(y_{i_1} \quad \cdots \quad y_{i_r}) = Q \begin{pmatrix} R_r \\ 0 \end{pmatrix},$$

where Q is an $m \times m$ orthogonal matrix and R_r is an $r \times r$ upper triangular matrix.

(ii) Update the factorization

$$H_m \leftarrow Q^T H_m Q,$$

$$V_m \leftarrow V_m Q.$$

It can be shown that the matrices V_m and H_m satisfy the relation

$$AV_m = V_m H_m + h_{m+1,m}\, v_{m+1} e_m^T + h_{m+1,m}\, v_{m+1} w^T$$

for some vector w such that $\|w\|_2$ is close to the machine epsilon if the condition number of R_r is modest. Furthermore, the columns of V_m together with v_{m+1} form an orthonormal set, and the first r vectors of V_m lie in the eigenspace of A associated with μ. That is, the first r vectors of V_m are Schur vectors for the eigenspace of A associated with μ. The $r \times r$ principal minor of H_m is the upper triangular matrix R_r.

(iii) Exit Step 4. Discard the vectors $v_{r+1}, \ldots, v_m$ in V_m. Set $k = r + 1$ and $v_{r+1} = v_{m+1}$, repeat step 2 through step 4; stop if a basis for the eigenspace of A associated with μ has been found. Note that this process is equivalent to applying a new deflated Arnoldi iteration with initial vector v_{r+1} to the operator $(I - P_r)A$, where P_r is the orthogonal projector onto the subspace spanned by the Schur vectors $\{v_1, \ldots, v_r\}$ of A.

Remark. For simplicity, our discussion above focused on finding the dominant eigenspace of A, but this restriction is not necessary. In practice, one can find the eigenspaces associated with several dominant eigenvalues simultaneously.

Checking the boundedness of f/Φ. The major and the most difficult substep of Step III is to check the boundedness of f/Φ, with f a given trigonometric polynomial. When $\Phi > 0$ (i.e., when the shifts of ϕ are stable), f/Φ is always bounded, and this substep is omitted. Thus, under the stability assumption, our algorithm (and code) is very robust for both univariate and bivariate cases. As a proof of evidence, our code successfully computed the regularity of the $(8, 8, 8)$ bivariate interpolatory mask of [RiS], whose autocorrelation mask is support on the square $[-31, 31] \times [-31, 31]$. The matrix representation of the associated transfer operation has an order of about 4000, and a brute force calculation of the regularity using the transfer operator would require one to find hundreds of eigenvalues of a huge matrix and decide later which of

the eigenvalues is the critical one. In contrast, since our algorithm does not use the matrix representation of the transfer operator explicitly, the size of the memory we need is only a small fraction of that required by a direction calculation. Also, by suppressing a priori hundreds of eigenvectors corresponding to polynomial reproduction, we need only to calculate the dominant eigenvalue of the operator $(I - P_\phi)\mathcal{T}$ instead of a multitude of eigenvalues of $\mathcal{T}$.

For the univariate case, since the function Φ has only finitely many isolated zeros and the multiplicity of each zero is relatively easy to find, the boundedness of f/Φ can be completely settled. Consequently, the algorithm and the derived code provide in this case the exact regularity parameter.

For the bivariate case, it is much more difficult to compute numerically the multiplicity space of the zeros of Φ. The current version of the code can only handle the case when Φ has finitely many zeros (which we find as an acceptable assumption: refinable functions with unstable shifts may be very useful in the construction of *framelets* with "customized" properties (cf. [R1]). It is very unlikely that any of these constructs will violate the "finitely many zeros" condition). Already for this case, the reliability of our code depends on (i) the number of the zeros and their distribution, (ii) the "degree" of the multiplicity space of each zero. However, for all of the interesting examples we tested, we did obtain reliable smoothness parameters. Even for an "extremely bad" refinable function (i.e., whose Φ vanishes at many points and to high degrees) the code is able to provide "good" lower bounds on the regularity, much better than the lower bound obtained by ignoring the dependence relation effect.

Given a trigonometric polynomial f, in order to check whether f/Φ is bounded in $[-\pi,\pi]^d$, one needs only to check whether it is bounded in local neighborhoods of the zeros of Φ.

Let ξ be a zero of Φ in $[-\pi,\pi]^2$ of exact order m. (The number m can be computed numerically.) Thus, all the derivatives of Φ up to order $m-1$ vanish at ξ, but some derivatives of order m do not. The Taylor expansion of Φ at ξ has then the form

$$\Phi(\xi+\eta) \;=\; \sum_{|\beta|=m} \frac{D^\beta \Phi(\xi)}{\beta!}\,\eta^\beta \;+\; \mathcal{O}(\|\eta\|^{m+1}).$$

Now, if f/Φ were to be bounded in a local neighborhood of ξ, then it would be necessary for f to satisfy the condition

$$D^\beta f(\xi) \;=\; 0 \quad \forall\, |\beta| \le m-1. \tag{4.5}$$

Hence, we can reject those eigenvalues whose eigenspace contains no eigenvectors that satisfy (4.5) (for all ξ).

Next we discuss how the condition (4.5) can be checked numerically in our algorithm.

Suppose $\{g_1,\dots,g_L\}$ is a basis for the eigenspace associated with an eigenvalue μ. Consider the eigenvector $g = \sum_{i=1}^L c_i g_i$, where not all the coefficients are zero. The condition that $D^\beta \widehat{g}(\xi) = 0$ for all $|\beta| \le m-1$ is equivalent to

$$\langle g, S_\xi \rangle = 0,$$

where

$$S_\xi \;=\; \{p\,\exp(i\xi\cdot) \,:\, p \in \Pi_{m-1}\}\,.$$

Since g is supported on $\mathcal{Z}_\phi$, the functions in S_ξ may be regarded as sequences defined on that domain too. Thus, we may interpret the above condition as saying that g lies in the null space of B_ξ^* for a suitable matrix B_ξ (whose columns span S_ξ). Thus, if G is a corresponding matrix representation for the basis $\{g_1, \ldots, g_L\}$, we need to find the null space of $B_\xi^* G$.

If Φ has more than one zero, say $\xi^{(1)}, \ldots, \xi^{(K)}$, then in order to find g such that $\widehat{g}/\Phi$ is bounded on $[-\pi, \pi]^d$, we seek a nontrivial null space for

$$\mathcal{G} \;:=\; \left[B_{\xi^{(1)}} \;\cdots\; B_{\xi^{(K)}}\right]^* G.$$

In our implementation, we find c in the null space of $\mathcal{G}$ by computing the SVD (singular value decomposition) of $\mathcal{G}$. If the minimal singular value $\sigma_{\min}(\mathcal{G})$ of $\mathcal{G}$ is sufficiently small, then we conclude that the null space of $\mathcal{G}$ is nontrivial and take c to be a minimal singular vector of $\mathcal{G}$.

Suppose that f satisfies (4.5) and that $d = 2$. If, in addition, the following two polynomials

$$t \mapsto \sum_{|\beta|=q} \frac{D^\beta \Phi(\xi)}{\beta!}\, t^{\beta_2}, \qquad t \mapsto \sum_{|\beta|=q} \frac{D^\beta \Phi(\xi)}{\beta!}\, t^{\beta_1}, \tag{4.6}$$

are strictly positive on the interval $[-1, 1]$, then f/Φ is bounded. To see this, we analyze the ratio

$$\frac{f(\xi+\eta)}{\Phi(\xi+\eta)} \;=\; \frac{\sum_{|\beta|=q} \frac{D^\beta f(\xi)}{\beta!}\, \eta^\beta + \mathcal{O}(\|\eta\|^{q+1})}{\sum_{|\beta|=q} \frac{D^\beta \Phi(\xi)}{\beta!}\, \eta^\beta + \mathcal{O}(\|\eta\|^{q+1})} \tag{4.7}$$

for sufficiently small nonzero vector η. Suppose $|\eta_2| \le |\eta_1|$. Then $\eta_2 = t\,\eta_1$ for $t \in [-1, 1]$, and substituting this into (4.7) would lead to

$$\frac{f(\xi+\eta)}{\Phi(\xi+\eta)} \;=\; \frac{\sum_{|\beta|=q} \frac{D^\beta f(\xi)}{\beta!}\, t^{\beta_2} + \mathcal{O}(\eta_1)}{\sum_{|\beta|=q} \frac{D^\beta \Phi(\xi)}{\beta!}\, t^{\beta_2} + \mathcal{O}(\eta_1)}.$$

Hence, whenever the polynomials in (4.6) are strictly positive on $[-1, 1]$, f/Φ is bounded in a neighborhood of ξ (the above argument applies to the case $|\eta_2| \le |\eta_1|$, and the complementary case is obtained by symmetry.) Finally, we remark that whether the polynomials in (4.6) are strictly positive can be checked numerically.

It must be emphasized that the multiplicity of the zero of Φ at a given point ξ, while necessarily of finite-dimension (since the zero is isolated), is not always of a total degree form. The present version of our code, however, computes only the total degree subspace of that multiplicity space, and hence provides in such cases *lower bounds* on the smoothness parameter.

5. Examples. We record some of our numerical experiments that we conducted as a test for the code.

The first class of examples are taken from the bivariate interpolatory refinable functions that were constructed by [RiS] ("interpolatory" means that $\phi(j) = \delta_j$, $j \in \mathbb{Z}^d$, and is a stronger property than stability). These examples demonstrate that the code can handle very large masks of stable refinable functions.

Example 5.1. The mask a_r of an interpolatory refinable function ϕ_r in [RiS] is obtained by convoluting the mask m_r of a centered three directional box spline with mask q_r of a carefully chosen distribution. The symbol of m_r (for an even r) is

$$\widehat{m}_r(\omega) = \Big(\cos\Big(\frac{\omega_1}{2}\Big) \cos\Big(\frac{\omega_2}{2}\Big) \cos\Big(\frac{\omega_1+\omega_2}{2}\Big)\Big)^r.$$

The mask m_r is of a box spline that lies in $C^{2r-2}(\mathbb{R}^2)$. The smoothness of ϕ_r also increases with r, but not at the same rate as its box spline factor. The distribution factor q_r, while having a negative effect on the smoothness, is necessary in order to achieve the interpolatory property of ϕ_r. For $r = 2$, the corresponding q_2 is

$$\widehat{q_2}(\omega) = \Big(5 - \cos(\omega_1) - \cos(\omega_2) - \cos(\omega_1+\omega_2)\Big)/2.$$

The L_2-regularity of ϕ_2 is 2.440765. We computed the smoothness α_r of the other interpolatory refinable functions ϕ_r, $r = 3, 4, \ldots, 8$. They are as follows:

r	3	4	5	6	7	8
α_r	3.175132	3.793134	4.344014	4.862018	5.362768	5.852746

. □

As a second test class, we tested four directional box splines. It is well known (cf. [BHR]) that the shifts of the four directional box spline are not stable. At the same time, their smoothness is explicitly known.

Example 5.2. The symbols of the masks of the four direction box splines considered here are

$$\widehat{m}_r(\omega) = \Big(\cos\Big(\frac{\omega_1}{2}\Big) \cos\Big(\frac{\omega_2}{2}\Big) \cos\Big(\frac{\omega_1+\omega_2}{2}\Big) \cos\Big(\frac{\omega_1-\omega_2}{2}\Big)\Big)^r.$$

Our code computed, for $r = 1, 2, 3, 4$, the corresponding smoothness of 2.5, 5.5, 8.5, 11.5. These are, indeed, the exact smoothness parameters of these splines.

The third set of examples is taken from [JS]. The pertinent refinable functions are univariate, interpolatory, and correspond to dilation $s = 3, 4$. The shifts of these functions form an orthonormal system.

Example 5.3. The mask a_n of the interpolatory refinable function ϕ_n whose shifts form an orthonormal basis is obtained by convoluting a B-spline of order n with the mask q_n of some distribution. The smoothness of the examples in [JS] with dilation $s = 3$, and with B-spline factor of order 2 and 3 are 0.963825 and 1.098068, respectively. The smoothness of the examples in [JS] with dilation $s = 4$ and a B-spline factor of order $2, 3, 4$ are 0.890339, 1.21178, and 1.303449. □

Example 5.4. The next example is a univariate refinable function whose shifts are unstable, with mask given by

$$\widehat{m}(\omega) = \cos^j\Big(\frac{\omega}{2}\Big)\,\big(2\cos(\omega) - 1\big)^k.$$

For $(j, k) = (4, 3)$ and $(j, k) = (4, 2)$, the computed smoothness of the refinable functions is 3.5. This agrees with the fact that both functions are cubic splines. We note that for this examples the lower bound estimates (that ignore the first condition in the definition of I_ϕ) fail to yield the correct smoothness. □

The last example shows the difficulties in getting the exact regularity of refinable functions, in the case where the corresponding dominant eigenvector Φ of T has many zeros. However, a good lower bound of the regularity is still possible to obtain.

Example 5.5. The mask is

$$\widehat{m}_r(\omega) = \cos\Big(\frac{\omega_1}{2}\Big)\cos\Big(\frac{\omega_2}{2}\Big)\cos\Big(\frac{\omega_1+\omega_2}{2}\Big)\cos\Big(\frac{\omega_1-\omega_2}{2}\Big)\Big(\frac{1+e^{i(6\omega_1+5\omega_2)}}{2}\Big)\Big(\frac{1+e^{i(-3\omega_1+5\omega_2)}}{2}\Big).$$

The operator $(I-P_\phi)T$ has the following dominant eigenvalues:

$$\mu = 2^{-6} \text{ with the dimension of eigenspace} = 6;$$
$$\mu = 2^{-7} \text{ with the dimension of eigenspace} = 12;$$
$$\mu = 2^{-8} \text{ where dimension of eigenspace} = 52.$$

Thus, a straightforward lower bound on the smoothness is 3. The function Φ has about 79 zeros in $[-\pi,\pi)\times[-\pi,\pi)$. In our computations, we were able to compute accurately the following zeros:

$$(-\pi,-\pi) \quad \pm(2\pi/3,0) \quad \pm(0,0.8\pi) \quad \pm(0,0.4\pi) \quad \pm(2\pi/3,0.8\pi).$$

Each is verified to have total order 4. Based on these zeros, we were able to reject the eigenvalues 2^{-6} and 2^{-7} as "false" eigenvalues. Thus a lower bound on the regularity is 4. The refinable function is this case is a box spline whose exact L_2-smoothness is $\alpha = 4.5$. □

6. Proofs of Propositions 3.3 and 3.4.

Proof of Proposition 3.3. Approximation theory basics (cf., e.g., [BDR] and [BR]) imply that $\Pi_m \subset \Pi_\phi$ if and only if $|\widehat{\phi}|^2$ has a zero of order $m+1$ at each $j \in 2\pi\mathbb{Z}^d\backslash 0$. Set $\mathcal{L} := 2\pi(\mathbb{Z}^d\backslash(s^*\mathbb{Z}^d))$ (to get a feeling for that set: in one dimension, dyadic dilations, this is the set of 2π-odd integers). Given a nonzero 2π-integer j, we write it as $j = s^{*k}j'$, $j' \in \mathcal{L}$, and use k times the refinement equation to conclude that

$$\widehat{\phi}(\omega+j) = \widehat{\phi}(s^{*-k}\omega + j')\prod_{n=1}^{k}\widehat{a}(s^{*-n}(\omega+j)).$$

This means that $|\widehat{\phi}|^2$ has a zero of order $m+1$ as each point of $2\pi\mathbb{Z}^d\backslash 0$ if and only if it has such a zero at each point of (the smaller set) $\mathcal{L}$.

We proceed by stating the following lemma, whose proof is postponed until after the proposition is proved.

LEMMA. *Let ϕ be a compactly supported L_2-function. Let $\gamma \in \mathbb{R}^d$. Then $|\widehat{\phi}|^2$ vanishes to order m at each $j \in \gamma + 2\pi\mathbb{Z}^d$ if and only if its 2π-periodization Φ has such zero at γ.*

In order to complete the proof of the proposition, note that $\mathcal{L}$ is the disjoint union of the cosets $s^*(\gamma + 2\pi\mathbb{Z}^d)$, $\gamma \in \Gamma\backslash 0$. For $j \in 2\pi\mathbb{Z}^d$, the 2π-periodicity of $\widehat{b}$ implies that $|\widehat{\phi}(s^*(\gamma+j))|^2 = \widehat{b}(\gamma)|\widehat{\phi}(\gamma+j)|^2$. The 2π-periodization of the right-hand side is $\widehat{b}(\gamma)\Phi(\gamma)$; thus the lemma applies to show that $|\widehat{\phi}|^2$ has a zero of order $m+1$ at each of $s^*(\gamma+2\pi\mathbb{Z}^d)$ if and only if $\widehat{b}\Phi$ has such a zero at γ. Varying that conclusion over all $\gamma \in \Gamma\backslash 0$, we obtain the desired result.

It remains now to prove the lemma. One implication here is trivial: since $|\widehat{\phi}|^2$ is nonnegative, its 2π-periodization can have a zero of a certain order at γ only if each of the summands has a corresponding zero.

Assume conversely that $|\widehat{\phi}|^2$ has a zero of order m at each $\gamma+j$, $j\in 2\pi\mathbb{Z}^d$, and note that (since $\widehat{\phi}$ is smooth) m must be even. Let Ω be a small neighborhood of γ. Since ϕ is compactly supported, we have that $\widehat{\phi}\in W_2^\rho(\mathbb{R}^d)$ for any ρ. Now, since $\widehat{\phi}$ has a zero of order $m/2$ at $\gamma+j$, we have (with D^β, $\beta\in\mathbb{Z}^d$, the usual partial differentiation)

$$|\widehat{\phi}(\omega+\gamma+j)|\le c|\omega|^{m/2}\max_{|\beta|=m/2}\|D^\beta\widehat{\phi}\|_{L_\infty(\Omega+j)}\quad\text{for }\omega\in\Omega. \tag{6.1}$$

Choosing $\rho>m/2+d/2$, the Sobolev embedding theorem implies that $W_2^\rho(\Omega+j)$ is continuously embedded in the Sobolev space $W_\infty^{m/2}(\Omega+j)$. Thus,

$$\max_{0\le|\beta|\le m/2}\|D^\beta\widehat{\phi}\|_{L_\infty(\Omega+j)}\le c_1\|\widehat{\phi}\|_{W_2^\rho(\Omega+j)},$$

with c_1 *independent of* j (since all the $\Omega+j$ sets are translates of each other). Substituting this into (6.1) we obtain that

$$|\widehat{\phi}(\omega+\gamma+j)|\le c_2|\omega|^{m/2}\|\widehat{\phi}\|_{W_2^\rho(\Omega+j)},\quad \omega\in\Omega, j\in 2\pi\mathbb{Z}^d.$$

Squaring the last inequality and summing over $j\in 2\pi\mathbb{Z}^d$ (and assuming, for simplicity and without loss, that ρ is an integer) we obtain that

$$\Phi(\omega+\gamma)\le c_3|\omega|^m\|\widehat{\phi}\|^2_{W_2^\rho(\mathbb{R}^d)}.\qquad\square$$

Proof of Proposition 3.4. Statement (a) follows from (b): choosing in (b) m to be (the odd number) m_ϕ, we get (a) unless $\rho_{m_\phi}\le\lambda^{-m_\phi-1}$. However, in the event that this latter inequality holds, we get that $\rho\le\rho_m\le\lambda^{-m_\phi-1}$, implying thereby that $\alpha(\phi)\ge\frac{m_\phi+1}{2}$. This implies (cf. [R1]) that the shifts of ϕ span all polynomials of degree $\frac{m_\phi+1}{2}$, and hence that the shifts of $\phi^\#$ span all polynomials of degree $m_\phi+1$, in contradiction to the very definition of m_ϕ.

In order to prove (b), let $f\in H_\phi\cap I'_\phi$ be an eigenvector of the operator $(1-P_m)T$, with an associated eigenvalue μ. Assume also that $|\mu|>\lambda^{-m-1}$.

We first prove that f is actually an eigenvector of T. For that, we first observe that $\widehat{b}f$ has a zero of order $m+1$ at each of the points of Γ: for $\gamma\in\Gamma\backslash 0$, this follows from the fact that $\widehat{b}f=\widehat{b}\Phi t$, for a bounded t (since $f\in I'_\phi$), together with Proposition 3.3. For $\gamma=0$, this follows from the fact that, by assumption, $(1-P_m)Tf=\mu f$, hence f lies in the range of $(1-P_m)$ (and every function in that range vanishes to order $m+1$ at the origin). Thus, indeed, $\widehat{b}f$ vanishes to order $m+1$ on Γ. We conclude from the definition of T that Tf vanishes to order $m+1$ at the origin, and hence that $\mu f=(1-P_m)Tf=Tf$.

We will now establish (b) by a chain of (in)equalities. First, by our assumptions on μ,

$$m+1<\log_\lambda(|\mu|).$$

Second, once we know that (μ,f) is an eigenpair of T, we can write

$$\log_\lambda(|\mu|)=\lim_{k\to\infty}\frac{\log_\lambda\|T^k(f)\|_{L_1(\mathbb{T}^d)}}{k}. \tag{6.2}$$

Since $|Tf| \le T|f|$ (regardless of the nature of f), we get that

$$\lim_{k\to\infty} \frac{\log_\lambda \|T^k(f)\|_{L_1(\mathbb{T}^d)}}{k} \le \limsup_{k\to\infty} \frac{\log_\lambda \|T^k|f|\|_{L_1(\mathbb{T}^d)}}{k}. \tag{6.3}$$

Let

$$u : \omega \mapsto \left(\sum_{j=1}^{d} \sin^2(\omega_j/2)\right)^{(m+1)/2}.$$

Then, our assumptions here imply that the function $g := |f|/u$ is bounded and that, moreover, g/Φ is also bounded. Invoking (b) of Corollary 2.10 of [RS1] (with g there being our g here, and with ℓ there being $(m+1)/2$ here; the corollary requires that the right-hand side of (6.2) is greater than $m+1$, something that we have already proved), we get that

$$\limsup_{k\to\infty} \frac{\log_\lambda \|T^k(|f|)\|_{L_1(\mathbb{T}^d)}}{k} \le -2\alpha(\phi).$$

Finally, $-2\alpha(\phi) = \log_\lambda \rho$. We thus conclude that $\log_\lambda |\mu| \le \log_\lambda \rho$ and hence that $\rho_m \le \rho$. The converse inequality is trivial. □

REFERENCES

[A] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[BDR] C. de Boor, R. DeVore, and A. Ron, *Approximation from shift-invariant subspaces of $L_2(\mathbb{R}^d)$*, Trans. Amer. Math. Soc., 341 (1994), pp. 787–806. Also available at ftp://ftp.cs.wisc.edu/Approx file: l2shift.ps.

[BR] C. de Boor and A. Ron, *The exponentials in the span of the integer translates of a compactly supported function: Approximation orders and quasi-interpolation*, J. London Math. Soc., 45 (1992), pp. 519–535. Also available at ftp://ftp.cs.wisc.edu/Approx file: quasi.ps.

[BW] E. Belogay and Yang Wang, *Arbitrarily smooth orthogonal nonseparable wavelets in $\mathbb{R}^2$*, SIAM J. Math. Anal., 30 (1999), pp. 678–697.

[CD] A. Cohen and I. Daubechies, *Nonseparable bidimensional wavelet bases*, Rev. Mat. Iberoamericana, 9 (1993), pp. 51–137.

[CGV] A. Cohen, K. Gröchenig, and L. Villemoes, *Regularity of multivariate refinable functions*, Constr. Approx., 15 (1999), pp. 241–255.

[CS] A. Cohen and J.-M. Schlenker, *Compactly supported bidimensional wavelet bases with hexagonal symmetry*, Constr. Approx., 9 (1993), pp. 209–236.

[D] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conf. Ser. Appl. Math. 61, SIAM, Philadelphia, 1992.

[DD] G. Deslaurier and S. Dubuc, *Symmetric iterative interpolation processes*, Constr. Approx., 5 (1989), pp. 49–68.

[DDD] G. Deslaurier, J. Dubois, and S. Dubuc, *Multidimensional iterative interpolation*, Canad. J. Math, 43 (1991), pp. 297–312.

[DGL] N. Dyn, J. A. Gregory, and D. Levin, *A butterfly subdivision scheme for surface interpolation with tension control*, ACM Trans. on Graphics, 9 (1990), pp. 160–169.

[E] T. Eirola, *Sobolev characterization of solutions of dilation equations*, SIAM J. Math. Anal., 23 (1992), pp. 1015–1030.

[GR] K. Gröchenig and A. Ron, *Tight compactly supported wavelet frames of arbitrary high smoothness*, Proc. Amer. Math. Soc., 126 (1998), pp. 1101–1107. Also available at ftp://ftp.cs.wisc.edu/Approx file: cg.ps.

[HL] W. He and M.-J. Lai, *Construction of bivariate compactly supported biorthogonal box spline wavelets with arbitrarily high regularities*, Appl. Comput. Harmon. Anal., 6 (1999), pp. 53–74.

[HJ] B. Han and R.-Q. Jia, *Optimal interpolatory subdivision schemes in multidimensional spaces*, SIAM J. Numer. Anal., 36 (1999), pp. 105–124.

[J] R.-Q. Jia, *Characterization of smoothness of multivariate refinable functions in Sobolev spaces*, Trans. Amer. Math. Soc., 351 (1999), pp. 4089–4112.

[JRS] H. Ji, S.D. Riemenschneider, and Z. Shen, *Multivariate compactly supported fundamental refinable functions, duals and biorthogonal wavelets*, Stud. Appl. Math., 102 (1999), pp. 173–204.

[JS] H. Ji and Z. Shen, *Compactly supported (bi)orthogonal wavelets generated by interpolatory refinable functions*, Adv. Comput. Math., 11 (1999), pp. 81–104.

[KS] J. Kovacević and W. Sweldens, *Wavelet families increasing order in arbitrary dimensions*, IEEE Trans. Image Process, 9 (2000), pp. 480–496.

[LMW] K.S. Lau, M.F. Ma, and J. Wang, *On some sharp regularity estimations of L^2-scaling functions*, SIAM J. Math. Anal., 27 (1996), pp. 835–864.

[LLS1] W. Lawton, S.L. Lee, and Z. Shen, *Stability and orthonormality of multivariate refinable functions*, SIAM J. Math. Anal., 28 (1997), pp. 999–1014.

[LLS2] W. Lawton, S.L. Lee, and Z. Shen, *Convergence of multidimensional cascade algorithm*, Numer. Math., 78 (1998), pp. 427–438.

[LS] R.B. Lehoucq and D.C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[LSVY] R.B. Lehoucq, D.C. Sorensen, P. Vu, and C. Yang, *ARPACK: An Implementation of the Implicitly Restarted Arnoldi Iteration that Computes Some of the Eigenvalues and Eigenvectors of a Large Sparse Matrix*, 1995. Also available at ftp.caam.rice.edu from the directory pub/software/ARPACK.

[R1] A. Ron, *Smooth refinable functions provide good approximation orders*, SIAM J. Math. Anal., 28 (1997), pp. 731–748.

[R2] A. Ron, *Wavelets and their associated operators*, in Approximation Theory IX Vol. II, C. K. Chui and L.L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, 1998, pp. 283–317. Also available at ftp://ftp.cs.wisc.edu/Approx File: texas9.ps.

[RiS] S.D. Riemenschneider and Z. Shen, *Multidimensional interpolatory subdivision schemes*, SIAM J. Numer. Anal., 34 (1997), pp. 2357–2381.

[RS1] A. Ron and Z. Shen, *The Sobolev regularity of refinable functions*, J. Approx. Theory, 106 (2000), pp. 185–225. Also available at ftp://ftp.cs.wisc.edu/Approx file reg.ps.

[RS2] A. Ron and Z. Shen, *Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator*, J. Funct. Anal., 148 (1997), pp. 408–447. Also available at ftp://ftp.cs.wisc.edu/Approx file affine.ps.

[RS3] A. Ron and Z. Shen, *Affine systems in $L_2(\mathbb{R}^d)$* II: *Dual system*, J. Fourier Anal. Appl., 3 (1997), pp. 617–637. Also available at ftp://ftp.cs.wisc.edu/Approx file dframe.ps.

[RS4] A. Ron and Z. Shen, *Compactly supported tight affine spline frames in $L_2(\mathbb{R}^d)$*, Math. Comp., 67 (1998), pp. 191–207. Also available at ftp://ftp.cs.wisc.edu/Approx file tight.ps.

[RS5] A. Ron and Z. Shen, *Construction of compactly supported affine frames in $L_2(R^d)$*, in Advances in Wavelets, Ka-Sing Lau, ed., Springer-Verlag, New York, 1998, pp. 27–49.

[S] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[SV] 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.

[SY] D.C. Sorensen and C. Yang, *A truncated RQ iteration for large scale eigenvalue calculations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1045–1073.

[V] L.F. Villemoes, *Wavelet analysis of refinement equations*, SIAM J. Math. Anal., 25 (1994), pp. 1433–1460.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 77–102
© 2001 Society for Industrial and Applied Mathematics

ROBUST EIGENSTRUCTURE ASSIGNMENT IN QUADRATIC MATRIX POLYNOMIALS: NONSINGULAR CASE*

N. K. NICHOLS† AND J. KAUTSKY‡

Abstract. Feedback design for a second-order control system leads to an eigenstructure assignment problem for a quadratic matrix polynomial. It is desirable that the feedback controller not only assigns specified eigenvalues to the second-order closed loop system but also that the system is *robust*, or insensitive to perturbations. We derive here new sensitivity measures, or condition numbers, for the eigenvalues of the quadratic matrix polynomial and define a measure of the robustness of the corresponding system. We then show that the robustness of the quadratic inverse eigenvalue problem can be achieved by solving a generalized linear eigenvalue assignment problem subject to *structured* perturbations. Numerically reliable methods for solving the structured generalized linear problem are developed that take advantage of the special properties of the system in order to minimize the computational work required. In this part of the work we treat the case where the leading coefficient matrix in the quadratic polynomial is nonsingular, which ensures that the polynomial is regular. In a second part, we will examine the case where the open loop matrix polynomial is not necessarily regular.

Key words. second-order control systems, quadratic inverse eigenvalue problem, feedback design, robust eigenstructure assignment, structured perturbations

AMS subject classifications. 65F18, 65F35, 93B32, 93B52, 93B55

PII. S0895479899362867

1. Introduction.

The time-invariant second-order control system

$$J\ddot{\mathbf{z}} - D\dot{\mathbf{z}} - C\mathbf{z} = B\mathbf{u}, \quad \mathbf{z}(0), \dot{\mathbf{z}}(0) \text{ given}, \tag{1}$$

where $\mathbf{z}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $J, D, C \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$, arises naturally in a wide variety of applications, including, for example, the control of large flexible space structures, earthquake engineering, the control of mechanical multibody systems, stabilization of damped gyroscopic systems, robotics, and vibration control in structural dynamics [1], [2], [14], [15], [22], [6], [25], [12], [13], [3], [26], [32], [23], [4], [7]. The control problem is to design a proportional and derivative state feedback controller of the form

$$\mathbf{u} = K_1\mathbf{z} + K_2\dot{\mathbf{z}} + \mathbf{r}, \tag{2}$$

where $K_1, K_2 \in \mathbb{R}^{m\times n}$ and $\mathbf{r}(t) \in \mathbb{R}^m$, such that the closed loop system

$$J\ddot{\mathbf{z}} - (D + BK_2)\dot{\mathbf{z}} - (C + BK_1)\mathbf{z} = B\mathbf{r} \tag{3}$$

has desired properties. The behavior of the closed loop system (3) is governed by the eigenstructure of its associated quadratic matrix polynomial

$$P_{cl}(\lambda) \equiv \lambda^2 J - \lambda(D + BK_2) - (C + BK_1). \tag{4}$$

*Received by the editors October 28, 1999; accepted for publication (in revised form) by D. Boley September 13, 2000; published electronically May 3, 2001.
http://www.siam.org/journals/simax/23-1/36286.html

†Department of Mathematics, The University of Reading, Box 220, Reading, RG6 6AX, United Kingdom (n.k.nichols@reading.ac.uk).

‡School of Mathematical Sciences, Flinders University of South Australia, Bedford Park, S.A. 5042, Australia (jarka@ist.flinders.edu.au).

The response of the system can, therefore, be shaped by selecting the feedback gain matrices K_1 and K_2 to assign the eigenstructure of the quadratic polynomial (4). The control design problem is thus formulated as an inverse quadratic eigenvalue problem. In practice (if $m > 1$), there is additional freedom in the solution to the problem and it is desirable to choose the feedback to ensure that the eigenstructure of the closed loop system is as *robust*, or insensitive to perturbations in the system matrices $J,\ D + BK_2,\ C + BK_1$, as possible.

Few computational techniques are available for treating the multi-input quadratic eigenstructure assignment problem directly. In [2], [14], [22], methods based on modal decompositions, which require the simultaneous diagonalization of the system matrices, are proposed. This approach is not generally applicable since the open loop system matrices may not always be diagonalizable. In any case, the technique is not numerically reliable because modal decompositions can be highly sensitive to computational errors. Two methods that are numerically reliable are described in [5]. The first of these is a modification of a technique proposed in [15], and the second is a generalization of a feedback stabilization procedure given in [8]. Both of these techniques aim to ensure that the (augmented) matrix of eigenvectors is well conditioned for inversion, which is a desirable property of the design. These procedures do not, however, ensure the *robustness* of the closed loop system.

In the majority of methods that have been proposed for solving the robust quadratic eigenvalue assignment problem, the second-order control system (1) is rewritten as a first-order system and techniques for treating the generalized linear feedback design problem are applied. There are two difficulties in using this approach. The first is that the measure of robustness for the linear problem is not the same as for the quadratic problem, since the allowable perturbations in the linear system are more general than in the quadratic problem. The second difficulty arises because the linear system has double the dimensions of the original quadratic system and, hence, the computational work used to solve the problem is greater than necessary.

In [16] we have developed numerical techniques for maximizing the robustness of the feedback design in linear systems that are subject to *structured* perturbations. We show here that the sensitivity of the eigenvalue problem for the quadratic polynomial is equivalent to that for a generalized linear pencil subject to a specific class of structured perturbations. The robustness of the second-order closed loop system can thus be ensured by solving a generalized linear eigenvalue assignment problem subject to this class of perturbations. We extend the methods derived in [16] to generalized linear systems and show how the special structure of the linear pencil derived from the quadratic polynomial can be exploited to reduce the computational work needed to solve the problem.

We consider here the case where the system matrix J is *nonsingular* and the quadratic polynomial is thus guaranteed to be regular. In a second paper we will consider the case where the system matrix J may be singular and the quadratic polynomial associated with the open loop system may not be regular. The aim of the feedback design is then to guarantee the regularity of the closed loop system as well as to assign the finite eigenvalues of the system robustly.

In the case where J is nonsingular, the quadratic matrix polynomial (4) can be reduced to a monic polynomial by applying the inverse of J from the left. In practice, the inversion of J should be avoided to ensure numerical reliability. The nonsingularity of J is assumed here in the theoretical derivation of the robustness measures, but the computational methods derived here do not use this inverse and

rely only on numerically stable procedures. We begin by presenting the background and sensitivity theory for the quadratic eigenvalue problem. In section 3 we establish the relation between the quadratic problem and the linear eigenvalue problem subject to structured perturbations. The robust eigenstructure assignment problem is defined and analyzed in section 4, and a numerical method for constructing the feedback controller is described in section 5. The results are summarized in the final section.

2. Quadratic eigenvalue problem.

2.1. Preliminary theory.

The quadratic matrix polynomial

$$P(\lambda) \equiv \lambda^2 J - \lambda D - C \tag{5}$$

and the corresponding second-order system (1) are said to be *regular* if

$$\det(P(\lambda)) \neq 0 \quad \text{for some } \lambda \in \mathbb{C}. \tag{6}$$

We assume throughout that the matrix J is nonsingular. The polynomial $P(\lambda)$ is thus regular and the system (1) is *solvable* in the sense that it admits a classical twice-differentiable solution $\mathbf{z}(t)$ for all continuous controls $\mathbf{u}(t)$ and *any* initial conditions $\mathbf{z}(0),\ \dot{\mathbf{z}}(0) \in \mathbb{R}^n$. This solution can be characterized in terms of the eigenstructure of the quadratic polynomial $P(\lambda)$.

For J nonsingular, the generalized eigenvalues of the quadratic polynomial are given by the $2n$ values of $\lambda \in \mathbb{C}$ for which $\det(\lambda^2 J - \lambda D - C) = 0$. The corresponding right and left eigenvectors are defined, respectively, to be nonzero vectors $\mathbf{v}$ and $\mathbf{w}$ satisfying

$$\begin{aligned} (\lambda^2 J - \lambda D - C)\mathbf{v} &= 0, \\ \mathbf{w}^H(\lambda^2 J - \lambda D - C) &= 0. \end{aligned} \tag{7}$$

Regularity of the polynomial ensures that there exist full rank matrices $V, W \in \mathbb{C}^{n\times 2n}$ that simultaneously satisfy

$$\begin{aligned} JV\Lambda^2 - DV\Lambda - CV &= 0, \\ \Lambda^2 W^H J - \Lambda W^H D - W^H C &= 0, \end{aligned} \tag{8}$$

and

$$VW^H J = 0, \qquad V\Lambda W^H J = I, \tag{9}$$

where $\Lambda \in \mathbb{C}^{2n\times 2n}$ is in Jordan canonical form with the eigenvalues of $P(\lambda)$ on the diagonal. The columns of V and W comprise, respectively, the right and left eigenvectors and principle vectors of the quadratic polynomial. The relations (9) define a specific normalization of these vectors.

We assume that the modal matrix V satisfying the first equation of (8) is such that $\tilde{V} = [V^T, (V\Lambda)^T]^T$ is nonsingular. Then, in the notation of [9], the matrix V and the Jordan matrix Λ together form a Jordan pair of the polynomial $P(\lambda)$. The matrix $W^H = \tilde{V}^{-1}[0, I]^T J^{-1}$ then satisfies the second of (8) and the relations (9) also hold. The matrices V, Λ, W are known as a Jordan triple of the quadratic polynomial. Conversely, we find that if V, Λ, W satisfy (8) and (9), where Λ is in Jordan form, then V, Λ, W form a Jordan triple and we can establish the following lemma. The lemma gives an explicit form for the inverse of $\tilde{V}$ that we use subsequently.

LEMMA 1. *Let V, W be full rank matrices satisfying* (8)–(9), *where Λ is in Jordan canonical form. Then the matrix $\tilde{V} = [V^T, (V\Lambda)^T]^T$ is nonsingular and its inverse is given by*

$$\tilde{V}^{-1} = [\Lambda W^H J - W^H D, W^H J]. \tag{10}$$

Proof. If (8) and (9) hold, then the conditions

$$\begin{aligned} V\Lambda W^H J - VW^H D &= I, \\ V(\Lambda^2 W^H J - \Lambda W^H D) = VW^H C &= 0 \end{aligned} \tag{11}$$

also hold. Therefore,

$$\begin{bmatrix} V \\ V\Lambda \end{bmatrix} [\Lambda W^H J - W^H D, W^H J] = I_{2n}, \tag{12}$$

which proves the result. ∎

The solution to the second-order system (1) can be written in terms of the Jordan triple V, Λ, W as follows.

THEOREM 2. *Let V, W, Λ satisfy* (8)–(9) *and let $\mathbf{u}(t)$ be a continuous function on the interval $t \in [0, T]$. Then, the solution to the second-order system of differential equations* (1) *is given explicitly for all $t \in [0, T]$ by*

$$\begin{aligned} \mathbf{z}(t) = {} & V \exp(\Lambda t)(\Lambda W^H J - W^H D)\mathbf{z}(0) + V \exp(\Lambda t) W^H J \dot{\mathbf{z}}(0) \\ & + \int_0^t V \exp(\Lambda(t-s)) W^H B \mathbf{u}(s) ds. \end{aligned} \tag{13}$$

Proof. The proof is by differentiation and direct verification. We let $\mathbf{z}(t)$ be defined by (13) and assume that (9) holds. Then, by Leibniz's rule, the continuity of $\mathbf{u}(s)$ and $\exp(\Lambda(t-s))$ for $s, t \in [0, T]$ implies that the first and second derivatives of $\mathbf{z}(t)$ are given by

$$\begin{aligned} \dot{\mathbf{z}} = {} & V\Lambda \exp(\Lambda t)(\Lambda W^H J - W^H D)\mathbf{z}(0) + V\Lambda \exp(\Lambda t) W^H J \dot{\mathbf{z}}(0) \\ & + \int_0^t V\Lambda \exp(\Lambda(t-s)) W^H B \mathbf{u}(s) ds, \\ \ddot{\mathbf{z}} = {} & V\Lambda^2 \exp(\Lambda t)(\Lambda W^H J - W^H D)\mathbf{z}(0) + V\Lambda^2 \exp(\Lambda t) W^H J \dot{\mathbf{z}}(0) \\ & + \int_0^t V\Lambda^2 \exp(\Lambda(t-s)) W^H B \mathbf{u}(s) ds + J^{-1} B \mathbf{u}(t). \end{aligned} \tag{14}$$

The relations (9) imply also that the initial conditions on $\mathbf{z}$ and $\dot{\mathbf{z}}$ at $t = 0$ are both satisfied. The proof then follows from (8) by direct substitution of (14) into (1). (See also [9], [19].) ∎

The response of the control system is therefore shaped by the eigenstructure of its corresponding quadratic polynomial, and the robustness of the system design depends on the sensitivity of the eigenstructure to perturbations in the system matrices. In the next sections, measures of the sensitivity and robustness of the system are derived.

2.2. Sensitivity and robustness. In order to measure the sensitivity of an eigenvalue of the quadratic polynomial $P(\lambda)$ to perturbations in its coefficient matrices, we follow the approach of Wilkinson [30]. Without loss of generality (since J is

nonsingular), we let $J\delta J, J\delta D, J\delta C \in \mathbb{R}^{n\times n}$ denote the perturbations in the coefficient matrices J, D, C, respectively. We assume that λ is a simple eigenvalue of $P(\lambda)$ with corresponding right and left eigenvectors $\mathbf{v}$ and $\mathbf{w}$ satisfying (7). The *condition number* of λ is then defined to be

$$c(\lambda) = \lim_{\epsilon\to 0} \sup \left(|\delta\lambda|/\epsilon\right), \tag{15}$$

where

$$\left((\lambda+\delta\lambda)^2(J+J\delta J) - (\lambda+\delta\lambda)(D+J\delta D) - (C+J\delta C)\right)(\mathbf{v}+\delta\mathbf{v}) = 0 \tag{16}$$

and

$$\|[\delta J, \delta D, \delta C]\|_2 \le \epsilon. \tag{17}$$

It is assumed that ϵ is sufficiently small to ensure that $J(I+\delta J)$ is nonsingular and the perturbed polynomial thus remains regular. (It is assumed implicitly in the definition that the perturbations $\delta\lambda, \delta\mathbf{v} \to 0$ as $\epsilon \to 0$. See also [11].) From this definition we have that

$$|\delta\lambda| \le c(\lambda)\epsilon + O(\epsilon^2), \tag{18}$$

and the condition number $c(\lambda)$ therefore gives a measure of the sensitivity of λ to perturbations of order ϵ in the coefficients of $P(\lambda)$. An explicit form for $c(\lambda)$ can be derived as follows.

THEOREM 3. *Let λ be a simple eigenvalue of the quadratic polynomial* (5). *Then, the condition number $c(\lambda)$ is given by*

$$c(\lambda) = \frac{\alpha \left\|\mathbf{w}^H J\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda J - D)\mathbf{v}|}, \tag{19}$$

where $\alpha = (|\lambda|^4 + |\lambda|^2 + 1)^{\frac{1}{2}}$.

Proof. By expanding (16), premultiplying by $\mathbf{w}^H$, and applying (7) we obtain

$$\begin{aligned}\delta\lambda\mathbf{w}^H(2\lambda J - D)\mathbf{v} &= -\mathbf{w}^H J(\lambda^2\delta J - \lambda\delta D - \delta C)\mathbf{v} + O(\epsilon^2)\\ &= -(\mathbf{w}^H J)[\delta J, \delta D, \delta C]\begin{bmatrix}\lambda^2\mathbf{v}\\ -\lambda\mathbf{v}\\ -\mathbf{v}\end{bmatrix} + O(\epsilon^2).\end{aligned} \tag{20}$$

The assumption that λ is a simple eigenvalue implies that $\mathbf{w}^H(2\lambda J - D)\mathbf{v} \neq 0$, and hence an upper bound on the first-order perturbation in λ is given by

$$|\delta\lambda| \le \frac{\alpha \left\|\mathbf{w}^H J\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda J - D)\mathbf{v}|} \|[\delta J, \delta D, \delta C]\|_2 + O(\epsilon^2). \tag{21}$$

To show that this upper bound is attained we let $T = (\epsilon/\alpha)J^T\mathbf{w}\mathbf{v}^H / \left\|\mathbf{w}^H J\right\|_2 \|\mathbf{v}\|_2$ and take $\delta J = \bar{\lambda}^2 T$, $\delta D = -\bar{\lambda}T$, and $\delta C = -T$. Then

$$\|[\delta J, \delta D, \delta C]\|_2 = \epsilon \tag{22}$$

and, since

$$|\mathbf{w}^H J(\lambda^2\delta J - \lambda\delta D - \delta C)\mathbf{v}| = \epsilon\alpha \left\|\mathbf{w}^H J\right\|_2 \|\mathbf{v}\|_2, \tag{23}$$

we obtain equality in (21) for these choices of the perturbations. Dividing (21) by ϵ and taking the limit as $\epsilon \to 0$ then completes the proof. □

The condition number $c(\lambda)$ given by (19) measures the sensitivity of the eigenvalue λ to perturbations in $P(\lambda)$ in an absolute sense. For a nonzero eigenvalue, a measure of the relative sensitivity is given by the condition number $\kappa(\lambda)$ defined, as in [11], [29], to be

$$\kappa(\lambda) = \lim_{\epsilon \to 0} \sup \left(|\delta\lambda| / (\epsilon |\lambda|) \right). \tag{24}$$

With this definition we find that

$$\kappa(\lambda) = c(\lambda)/|\lambda| = \frac{\alpha \left\| \mathbf{w}^H J \right\|_2 \|\mathbf{v}\|_2}{|\lambda| |\mathbf{w}^H (2\lambda J - D)\mathbf{v}|}. \tag{25}$$

This expression is similar to the result derived in [29]. The difference is due to the definition of the perturbations and the form of the bound on $\delta J, \delta D, \delta C$. The perturbations chosen here represent errors relative to the components in the leading coefficient matrix J. This particular form ensures that the same condition number is derived if the polynomial is first reduced to monic form. It also allows the relations between the quadratic and linear cases to be established directly, as shown in section 3. More importantly, this formulation leads to a numerical procedure for solving the robust eigenstructure assignment problem that does not require the inversion of the matrix J.

To measure the robustness of the second-order system (1), we need an indicator of the overall sensitivity of the eigenvalues of the corresponding quadratic polynomial (5). The condition number (19) gives a proportional measure of the sensitivity of a simple eigenvalue to perturbations of order ϵ in the coefficient matrices. For a nondefective eigenvalue λ of multiplicity p, the condition numbers (19) are also well defined for a particular choice of the basis eigenvectors $\{\mathbf{v}_j\}_1^p, \{\mathbf{w}_j\}_1^p$ spanning the corresponding right and left invariant subspaces. Provided that these bases are biorthogonal with respect to the matrix $2\lambda J - D$, then an equivalent proportional measure of the sensitivity of the eigenvalue is given by the square root of the sum of the squares of all the associated condition numbers. If the system has a defective multiple eigenvalue, then the sensitivity of *some* eigenvalue to perturbations of order ϵ is expected to be larger by at least an order of magnitude in ϵ [30], [20], [24]. Therefore, systems that have defective eigenvalues are necessarily less robust than nondefective systems.

As a global measure of robustness we thus take

$$\nu^2 = \sum_{j=1}^{2n} \omega_j^2 c(\lambda_j)^2, \tag{26}$$

where the eigenvalues $\{\lambda_j\}_1^n$ of the system are assumed to be nondefective and the positive weights $\omega_j,\ j = 1, \ldots, 2n$, satisfy $\sum_{j=1}^{2n} \omega_j^2 = 1$ with $\omega_j = \omega_k$ if $\lambda_j = \lambda_k$. (See also [18], [17].) If the right and left eigenvectors $\mathbf{v}_j,\ \mathbf{w}_j$, corresponding to λ_j, are normalized such that

$$(|\lambda_j|^4 + |\lambda_j|^2 + 1)^{\frac{1}{2}} \|\mathbf{v}_j\|_2 = 1, \quad |\mathbf{w}_j^H (2\lambda_j J - D)\mathbf{v}_j| = 1, \quad j = 1, \ldots, 2n, \tag{27}$$

then the robustness measure ν^2 can be written

$$\nu^2 = \sum_{j=1}^{2n} \omega_j^2 \left\| \mathbf{w}_j^H J \right\|_2^2 = \left\| D_\omega W^H J \right\|_F^2, \tag{28}$$

where $D_\omega = \mathrm{diag}\{\omega_1, \ldots, \omega_{2n}\}$. (Here $\|(\cdot)\|_F$ denotes the Frobenius matrix norm.) The normalization (27) is consistent with (9) and is selected to enable the relationships with the linear eigenvalue problem to be established.

2.3. Monic polynomial. Of practical interest is the case where $P(\lambda)$ is a monic polynomial with leading coefficient matrix $J = I$. It is assumed that this leading coefficient matrix is not subject to perturbations. We consider specifically the monic quadratic polynomial

$$\hat{P}(\lambda) \equiv \lambda^2 I - \lambda A_2 - A_1, \tag{29}$$

corresponding to the second-order system (1), where $J = I,\ D = A_2,\ C = A_1$.

A measure of the sensitivity of a simple eigenvalue λ of the monic polynomial (29) to perturbations $\delta A_1, \delta A_2$ in its coefficient matrices A_1, A_2, respectively, is given by the condition number $c(\lambda)$ defined in (15). The right and left eigenvectors corresponding to λ are again denoted by $\mathbf{v}, \mathbf{w}$, and the first-order perturbation $\delta\lambda$ now satisfies

$$\left((\lambda + \delta\lambda)^2 I - (\lambda + \delta\lambda)(A_2 + \delta A_2) - (A_1 + \delta A_1)\right)(\mathbf{v} + \delta\mathbf{v}) = 0, \tag{30}$$

where

$$\|[\delta A_1, \delta A_2]\|_2 \leq \epsilon. \tag{31}$$

An explicit form for $c(\lambda)$ in this case can be derived as follows.

THEOREM 4. *Let λ be a simple eigenvalue of the monic polynomial* (29). *Then, the condition number $c(\lambda)$ is given by*

$$c(\lambda) = \frac{\hat{\alpha}\left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda I - A_2)\mathbf{v}|}, \tag{32}$$

where $\hat{\alpha} = (|\lambda|^2 + 1)^{\frac{1}{2}}$.

Proof. By expanding (30), premultiplying by $\mathbf{w}^H$, and using the assumption that λ is a simple eigenvalue, we can show, by similar arguments to those in Theorem 3, that an upper bound on the first-order perturbation in λ is given by

$$|\delta\lambda| \leq \frac{\hat{\alpha}\left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda I - A_2)\mathbf{v}|} \|[\delta A_1, \delta A_2]\|_2 + O(\epsilon^2). \tag{33}$$

This upper bound is attained for the perturbations $\delta A_1 = T$ and $\delta A_2 = \bar{\lambda} T$, where $T = (\epsilon/\hat{\alpha})\mathbf{w}\mathbf{v}^H / \left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2$, since this choice ensures that

$$\|[\delta A_1, \delta A_2]\|_2 = \epsilon \tag{34}$$

and

$$|\mathbf{w}^H(\lambda\delta A_2 + \delta A_1)\mathbf{v}| = \epsilon\hat{\alpha}\left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2. \tag{35}$$

Dividing (33) by ϵ and taking the limit as $\epsilon \to 0$ then completes the proof. ∎

The form of the condition number in the monic case is thus the same as in the generalized case up to a constant factor. The difference is due to the different assumptions on the allowable perturbations.

The condition number (32) gives an absolute measure of the sensitivity of the eigenvalue λ in the monic case. For a nonzero eigenvalue, a measure of the relative sensitivity is given by the condition number $\kappa(\lambda)$ defined in (24). We find now that

$$\kappa(\lambda) = c(\lambda)/|\lambda| = \frac{\hat{\alpha}\left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2}{|\lambda||\mathbf{w}^H(2\lambda I - A_2)\mathbf{v}|}. \tag{36}$$

The global measure of robustness in the monic case is also taken to be ν^2, defined as in (26). Normalizing the eigenvectors of the polynomial such that

$$(1+|\lambda_j|^2)^{\frac{1}{2}} \|\mathbf{v}_j\|_2 = 1, \quad |\mathbf{w}_j^H(2\lambda_j I - A_2)\mathbf{v}_j| = 1, \qquad j = 1, \ldots, 2n, \tag{37}$$

then gives

$$\nu^2 = \sum_{j=1}^{2n} \omega_j^2 \left\|\mathbf{w}_j^H\right\|_2^2 = \left\|D_\omega W^H\right\|_F^2. \tag{38}$$

In both the generalized and the monic quadratic polynomial cases, the control design problem is to select the feedback gains to assign a given set of $2n$ nondefective eigenvalues to the second-order closed loop system and to minimize its robustness measure ν^2. In section 3 we show that this problem can be solved by minimizing the robustness of a generalized linear system subject to a restricted set of perturbations.

3. Generalized linear problem.

3.1. Transformation of the system. The inverse quadratic eigenvalue problem is commonly treated by transforming the second-order control system (1) into a generalized linear state-space, or *descriptor*, system of the form

$$E\dot{\mathbf{x}} = A\mathbf{x} + \tilde{B}\mathbf{u}, \quad \mathbf{x}(0) \text{ given}, \tag{39}$$

where $E, A \in \mathbb{R}^{2n\times 2n}, \tilde{B} \in \mathbb{R}^{2n\times m}$, and $\mathbf{x} = [\ \mathbf{z}^T,\ \dot{\mathbf{z}}^T\ ]^T$. Various transformations can be used to embed the second-order equations into the linear form. We consider the generalized linear system where

$$E = \begin{bmatrix} I & 0 \\ 0 & J \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & I \\ C & D \end{bmatrix}, \qquad \tilde{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}. \tag{40}$$

This form is suitable for treating the feedback design problem. Different transformations may be desirable for other purposes (see [29]).

The response of the system (39) is governed by the eigenstructure of the generalized linear matrix pencil

$$L(\lambda) \equiv \lambda E - A. \tag{41}$$

Since J is nonsingular, the linear pencil $L(\lambda)$ is regular in the case where E, A are defined by (40). The system (39) is then uniquely solvable for any continuous control $\mathbf{u}(t)$ and the solution is equivalent to that of the second-order system (1). The solutions to (1) can therefore also be characterized in terms of the eigenstructure of $L(\lambda)$.

The generalized eigenvalues of the linear pencil (41) are given by the $2n$ values of $\lambda \in \mathbb{C}$ for which $\det(\lambda E - A) = 0$. The corresponding right and left eigenvectors are defined, respectively, to be nonzero vectors $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{w}}$ satisfying

$$\begin{aligned} (\lambda E - A)\tilde{\mathbf{v}} &= 0, \\ \tilde{\mathbf{w}}^H(\lambda E - A) &= 0. \end{aligned} \tag{42}$$

Regularity of the pencil ensures that there exist nonsingular matrices $\tilde{V}, \tilde{W} \in \mathbb{C}^{2n\times 2n}$ that simultaneously satisfy

$$\begin{aligned} E\tilde{V}\Lambda - A\tilde{V} &= 0, \\ \Lambda\tilde{W}^H E - \tilde{W}^H A &= 0, \end{aligned} \tag{43}$$

and

$$\tilde{W}^H E\tilde{V} = I, \tag{44}$$

where $\Lambda \in \mathbb{C}^{2n\times 2n}$ is in Jordan canonical form. The columns of $\tilde{V}$ and $\tilde{W}$ comprise, respectively, the right and left eigenvectors and principle vectors of the linear matrix pencil. The relation (44) defines a normalization of these vectors.

The equivalence between the eigenstructure of the linear matrix pencil (41) with coefficients given by (40) and that of the quadratic matrix polynomial (5) can now be established.

THEOREM 5. *Let E, A be given by* (40). *If $\tilde{V}, \tilde{W}$ are nonsingular matrices satisfying* (43)–(44), *where Λ is in Jordan canonical form, then*

$$\tilde{V} = \begin{bmatrix} V \\ V\Lambda \end{bmatrix}, \qquad \tilde{W}^H = [\Lambda W^H J - W^H D, W^H], \tag{45}$$

where V, W are full rank matrices satisfying (8)–(9). *Conversely, if V, W are full rank matrices satisfying* (8)–(9), *then $\tilde{V}, \tilde{W}$ given by* (45) *are nonsingular and satisfy* (43)–(44).

Proof. We let $\tilde{V} = [\tilde{V}_1^H, \tilde{V}_2^H]^H$. If $\tilde{V}$ satisfies the first equation of (43), where E, A are defined as in (40), then

$$\begin{aligned} \tilde{V}_2 &= \tilde{V}_1\Lambda, \\ C\tilde{V}_1 + D\tilde{V}_2 &= J\tilde{V}_2\Lambda. \end{aligned} \tag{46}$$

It follows that $\tilde{V}_1 = V$ satisfies the first equation of (8) and $\tilde{V}_2 = V\Lambda$. Conversely, if V satisfies the first equation of (8), then

$$A\begin{bmatrix} V \\ V\Lambda \end{bmatrix} = \begin{bmatrix} V\Lambda \\ CV + DV\Lambda \end{bmatrix} = \begin{bmatrix} V\Lambda \\ JV\Lambda^2 \end{bmatrix} = E\begin{bmatrix} V \\ V\Lambda \end{bmatrix}\Lambda, \tag{47}$$

and the first equation of (43) is satisfied. The relation between $\tilde{W}$ and W is shown similarly. The invertibility of E together with (44) then implies that $E^{-1} = \tilde{V}\tilde{W}^H$ and hence, from (45), V, W must satisfy $VW^H = 0$ and $V\Lambda W^H = J^{-1}$ and (9) must hold. Conversely, if conditions (8)–(9) are satisfied, then, by Lemma 1, $\tilde{V}, \tilde{W}$ are invertible and $\tilde{V}\tilde{W}^H = E^{-1}$, which implies that (44) holds. ∎

We next relate the sensitivity of the eigenstructure of the quadratic matrix polynomial to that of the linear matrix pencil.

3.2. Sensitivity to structured perturbations. The sensitivity of a simple eigenvalue λ of the linear matrix pencil (41) to arbitrary perturbations in the pencil is known to be directly proportional to the condition number

$$c^L(\lambda) = \left\|\tilde{\mathbf{w}}^H E\right\|_2 \|\tilde{\mathbf{v}}\|_2 / |\tilde{\mathbf{w}}^H E\tilde{\mathbf{v}}|, \tag{48}$$

where $\tilde{\mathbf{v}}, \tilde{\mathbf{w}}$ are the right and left eigenvectors of the pencil corresponding to λ. (See [30], [11], [27], [28].) In the case where the coefficient matrices of the pencil are given

by (40), this condition number is *not* equivalent to the condition number $c(\lambda)$ of the embedded quadratic matrix polynomial derived in section 2.2. The condition number $c^L(\lambda)$ measures the sensitivity of λ to arbitrary perturbations in all components of the coefficient matrices E, A of the pencil, whereas the condition number $c(\lambda)$ measures the sensitivity of λ only to perturbations in the coefficient matrices J, D, C of the quadratic polynomial.

In order to establish a condition number for the generalized linear eigenproblem that is equivalent to that of the quadratic eigenproblem, we extend the theory of [16] to find a measure of the sensitivity of an eigenvalue of the generalized pencil (41) to a specific class of *structured* perturbations. We assume again that λ is a simple eigenvalue of $L(\lambda)$ with corresponding right and left eigenvectors $\tilde{\mathbf{v}}, \tilde{\mathbf{w}}$, respectively. We consider perturbations $\delta E, \delta A$ to the coefficient matrices E, A of $L(\lambda)$ of the form

$$\delta E = EF\Delta_E G_1^T, \qquad \delta A = EF\Delta_A G_2^T, \tag{49}$$

where Δ_E, Δ_A are arbitrary (unknown) disturbance matrices and F, G_1, G_2 are specified matrices that define the structure of the perturbations. The sensitivity of λ to perturbations of the form (49) can then be measured by the condition number $\tilde{c}(\lambda)$, defined as in (15), where the first-order perturbation $\delta\lambda$ now satisfies

$$\left((\lambda+\delta\lambda)(E+EF\Delta_E G_1^T) - (A+EF\Delta_A G_2^T)\right)(\tilde{\mathbf{v}}+\delta\tilde{\mathbf{v}}) = 0 \tag{50}$$

and

$$\|[\Delta_E, \Delta_A]\|_2 \le \epsilon. \tag{51}$$

It is assumed that ϵ is sufficiently small to ensure that $E(I + F\Delta_E G_1^T)$ is nonsingular and the perturbed linear pencil therefore remains regular. An explicit form for $\tilde{c}(\lambda)$ is given as follows.

THEOREM 6. *Let λ be a simple eigenvalue of the linear matrix pencil* (41). *Then, the condition number $\tilde{c}(\lambda)$ is given by*

$$\tilde{c}(\lambda) = \frac{\left\|\tilde{\mathbf{w}}^H EF\right\|_2 \left\|G_\lambda^T \tilde{\mathbf{v}}\right\|_2}{|\tilde{\mathbf{w}}^H E\tilde{\mathbf{v}}|}, \tag{52}$$

where $G_\lambda = [\lambda G_1, \ -G_2]$.

Proof. Applying arguments analogous to those in the proofs of Theorems 3 and 4, we find from (50) that

$$\begin{aligned} |\delta\lambda||\tilde{\mathbf{w}}^H E\tilde{\mathbf{v}}| &= |\tilde{\mathbf{w}}^H[\lambda EF\Delta_E G_1^T - EF\Delta_A G_2^T]\tilde{\mathbf{v}}| + O(\epsilon^2) \\ &= \left|\tilde{\mathbf{w}}^H EF[\Delta_E, \Delta_A]\begin{bmatrix} \lambda G_1^T \\ -G_2^T \end{bmatrix}\tilde{\mathbf{v}}\right| + O(\epsilon^2) \\ &\le \left\|\tilde{\mathbf{w}}^H EF\right\|_2 \left\|G_\lambda^T\tilde{\mathbf{v}}\right\|_2 \epsilon + O(\epsilon^2). \end{aligned} \tag{53}$$

Regularity of the pencil ensures that $\tilde{\mathbf{w}}^H E\tilde{\mathbf{v}} \neq 0$, and hence an upper bound on the first-order perturbation in λ is given by

$$|\delta\lambda| \le \frac{\left\|\tilde{\mathbf{w}}^H EF\right\|_2 \left\|G_\lambda^T\tilde{\mathbf{v}}\right\|_2}{|\tilde{\mathbf{w}}^H E\tilde{\mathbf{v}}|} \|[\Delta_E, \Delta_A]\|_2 + O(\epsilon^2). \tag{54}$$

Equality in (54) is achieved for the perturbations

$$\Delta_E = \bar{\lambda} E^T F^T \tilde{\mathbf{w}}\tilde{\mathbf{v}}^H G_1 \epsilon/\tau, \qquad \Delta_A = -E^T F^T \tilde{\mathbf{w}}\tilde{\mathbf{v}}^H G_2 \epsilon/\tau, \tag{55}$$

where $\tau = \left\|\tilde{\mathbf{w}}^H EF\right\|_2 \left\|G_\lambda^T \tilde{\mathbf{v}}\right\|_2$. Then $\|[\Delta_E, \Delta_A]\|_2 = \epsilon$ and

$$|\tilde{\mathbf{w}}^H E[\lambda F \Delta_E G_1^T - F\Delta_A G_2^T]\tilde{\mathbf{v}}| = \epsilon \left\|\tilde{\mathbf{w}}^H EF\right\|_2 \left\|G_\lambda^T \tilde{\mathbf{v}}\right\|_2, \tag{56}$$

and the upper bound on $|\delta\lambda|$ is attained. Dividing (54) by ϵ and taking the limit as $\epsilon \to 0$ then proves the result. ∎

In the case where the quadratic polynomial is embedded in the linear pencil (41) and the coefficients of the pencil are given by (40), the arbitrary perturbations may be taken to be

$$\Delta_E = \delta J, \qquad \Delta_A = [\delta D, \delta C], \tag{57}$$

and the matrices F, G_λ that structure the perturbations may be defined by

$$F = \begin{bmatrix} 0 \\ I_n \end{bmatrix}, \qquad G_1^T = [0, I_n], \qquad G_2^T = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}. \tag{58}$$

The admissible structured perturbations (49) then have the forms

$$\delta E = \begin{bmatrix} 0 & 0 \\ 0 & J\delta J \end{bmatrix}, \qquad \delta A = \begin{bmatrix} 0 & 0 \\ J\delta C & J\delta D \end{bmatrix}, \tag{59}$$

where

$$\|[\delta J, \delta D, \delta C]\|_2 = \|[\Delta_E, \Delta_A]\|_2 \leq \epsilon. \tag{60}$$

The condition number $\tilde{c}(\lambda)$ of the linear pencil, subject to the structured perturbations, can now be shown to equal the condition number $c(\lambda)$ of the quadratic polynomial.

COROLLARY 7. *Let E, A be defined by* (40) *and let Δ_E, Δ_A and F, G_λ be defined by* (57)–(58). *Then, the condition number $\tilde{c}(\lambda)$ satisfies*

$$\tilde{c}(\lambda) = \frac{\alpha \left\|\mathbf{w}^H J\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda J - D)\mathbf{v}|} \equiv c(\lambda), \tag{61}$$

where $\alpha = (|\lambda|^4 + |\lambda|^2 + 1)^{\frac{1}{2}}$ and $\mathbf{v}, \mathbf{w}$ are the right and left eigenvectors of the quadratic polynomial (5) *corresponding to the eigenvalue λ.*

Proof. From Theorem 5 it follows that

$$\left\|\tilde{\mathbf{w}}^H EF\right\|_2 = \left\|\mathbf{w}^H J\right\|_2, \quad \left\|G_\lambda^T \tilde{\mathbf{v}}\right\|_2 = \left\| \begin{bmatrix} \lambda G_1^T \tilde{\mathbf{v}} \\ G_2^T \tilde{\mathbf{v}} \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \lambda^2 \mathbf{v} \\ \lambda \mathbf{v} \\ \mathbf{v} \end{bmatrix} \right\|_2 = \alpha \|\mathbf{v}\|, \tag{62}$$

and

$$|\tilde{\mathbf{w}}^H E \tilde{\mathbf{v}}| = |2\lambda \mathbf{w}^H J\mathbf{v} - \mathbf{w}^H D\mathbf{v}|. \tag{63}$$

Substitution into the definitions of the condition numbers then establishes the result. ∎

An analogous result can be obtained in the case where the embedded quadratic polynomial is monic. In this case the linear pencil is also monic with coefficient matrices

$$E = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & I \\ A_1 & A_2 \end{bmatrix}. \tag{64}$$

It is assumed that the matrix E remains unperturbed. The arbitrary perturbations are now taken to be $\Delta_E = 0$ and $\Delta_A = [\delta A_1, \delta A_2]$. The structure of the perturbations is defined by $F = [0, I_n]^T$ and $G_\lambda^T \equiv G_2^T = I_{2n}$. The admissible perturbations then satisfy $\|[\delta A_1, \delta A_2]\|_2 \leq \epsilon$ and the condition number $\tilde{c}(\lambda)$, given by (52), can be shown to equal the condition number $c(\lambda)$, given by (32).

COROLLARY 8. *In the case of a monic pencil, with coefficients E, A defined by* (64), *the condition number $\tilde{c}(\lambda)$ satisfies*

$$\tilde{c}(\lambda) = \frac{\hat{\alpha} \left\|\mathbf{w}^H\right\|_2 \|\mathbf{v}\|_2}{|\mathbf{w}^H(2\lambda I - A_2)\mathbf{v}|} \equiv c(\lambda), \tag{65}$$

where $\hat{\alpha} = (|\lambda|^2 + 1)^{\frac{1}{2}}$ and $\mathbf{v}, \mathbf{w}$ are the right and left eigenvectors of the monic quadratic polynomial (29) *corresponding to the eigenvalue λ.*

Proof. From Theorem 5 we have

$$\left\|G_\lambda^T \tilde{\mathbf{v}}\right\|_2 = \left\| I_{2n} \begin{bmatrix} \mathbf{v} \\ \lambda \mathbf{v} \end{bmatrix} \right\|_2 = \hat{\alpha} \|\mathbf{v}\|, \quad |\tilde{\mathbf{w}}^H E \tilde{\mathbf{v}}| = |\mathbf{w}^H(2\lambda I - A_2)\mathbf{v}|. \tag{66}$$

The proof then follows as in Corollary 7 with $J = I$, $D = A_2$, and $C = A_1$. ∎

3.3. Robustness. As an overall measure of the sensitivity of the linear matrix pencil (41) to structured perturbations of the form (49), we take a weighted sum of the squares of the condition numbers $\tilde{c}(\lambda_j)$, $j = 1, \ldots, 2n$ (see [16], [18], [17]). We assume that the pencil is nondefective, since if any eigenvalue is defective, the order of the perturbation is expected to be magnified in some eigenvalue. In the case of a nondefective multiple eigenvalue, the condition numbers are defined with respect to a particular choice of the basis eigenvectors spanning the corresponding invariant subspaces and biorthogonal with respect to the matrix E. The right and left eigenvectors $\tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_j$ associated with each eigenvalue λ_j may also be normalized such that

$$\left\|G_{\lambda_j}^T \tilde{\mathbf{v}}_j\right\|_2 = 1, \qquad |\tilde{\mathbf{w}}_j^H E \tilde{\mathbf{v}}_j| = 1 \qquad \text{for all } j = 1, \ldots, 2n. \tag{67}$$

Then (44) holds and the global robustness measure is given by

$$\tilde{\nu}^2 \equiv \sum_{j=1}^{2n} \omega_j^2 \tilde{c}(\lambda_j)^2 = \sum_{j=1}^{2n} \omega_j^2 \left\|\tilde{\mathbf{w}}_j^H E F\right\|_2^2 = \left\|D_\omega \tilde{W}^H E F\right\|_F^2 = \left\|D_\omega \tilde{V}^{-1} F\right\|_F^2, \tag{68}$$

where $D_\omega = \operatorname{diag}\{\omega_1, \ldots, \omega_{2n}\}$ and ω_j, $j = 1, \ldots, 2n$, are positive weights satisfying $\sum_{j=1}^{2n} \omega_j^2 = 1$ with $\omega_j = \omega_k$ if $\lambda_j = \lambda_k$.

In the case where the coefficients of the linear pencil are given by (40), we can show, using Theorem 5, that the robustness measure (68) is equal to the robustness measure (28) of the embedded quadratic polynomial. As demonstrated in the proof of Corollary 7, the normalizations (67) and (27) are equivalent and, since $\tilde{c}(\lambda) = c(\lambda)$, it follows that

$$\tilde{\nu}^2 = \left\|D_\omega \tilde{W}^H E F\right\|_F^2 = \left\|D_\omega \tilde{V}^{-1} [0, I]^T\right\|_F^2 = \left\|D_\omega W^H J\right\|_F^2 = \nu^2, \tag{69}$$

which proves the result.

The equivalence of the robustness measures can also be established in the case where the quadratic polynomial embedded in the linear pencil is monic and the coef-

ficient matrices of the linear pencil are given by (64). Using the definitions of F, G_λ applicable to the monic case, we find that the normalizations (67) and (37) are equivalent. The equality between the robustness measures (68) and (38) of the linear pencil and the monic quadratic polynomial, respectively, then follows immediately from Corollary 8.

The robust eigenstructure assignment problem for the second-order control system (1) can now be formulated as an equivalent problem for a linear pencil. Numerical methods previously developed in [16] can then be applied directly to find the desired feedback gain matrices. In the next section we reformulate the control problem and establish the theory needed for eigenstructure assignment. In section 5 we derive a modified numerical procedure for solving the design problem that takes advantage of the special structure of the generalized pencil.

4. Robust eigenstructure assignment.

4.1. Quadratic control problem. The control design problem for the second-order system (1) is to select feedback matrices K_1, K_2 to ensure that the closed loop system (3) has a desired modal response. As demonstrated in section 2, the modal behavior of the closed loop system is characterized by the eigenstructure of its corresponding quadratic matrix polynomial $P_{cl}(\lambda) = \lambda^2 J - \lambda(D + BK_2) - (C + BK_1)$. The primary aim of the controller is therefore to determine feedback gains that assign a given set of eigenvalues to the quadratic polynomial. The inverse quadratic eigenvalue problem is stated explicitly as follows.

PROBLEM 1. *Given real matrices $J, D, C \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and a set of $2n$ complex numbers $\mathcal{L} = \{\lambda_1, \ldots, \lambda_{2n}\}$, closed under complex conjugation, find real matrices $K_1, K_2 \in \mathbb{R}^{m\times n}$ such that the eigenvalues of $P_{cl}(\lambda)$ are equal to λ_j, $j = 1, \ldots, 2n$.*

Conditions for the existence of solutions to Problem 1 are known and the following theorem is easily established.

THEOREM 9. *Solutions K_1, K_2 to Problem* 1 *exist for* every *set $\mathcal{L}$ of self-conjugate complex numbers if and only if the system* (1) *is* completely controllable*, that is,*

$$\operatorname{rank}[\lambda^2 J - \lambda D - C, B] = n \quad \text{for all } \lambda \in \mathbb{C}. \tag{70}$$

If the system is not completely controllable, then solutions exist if and only if the set $\mathcal{L} = \{\mathcal{L}_u, \mathcal{L}_c\}$ contains $\mathcal{L}_u$, the set of all values of λ for which the system (1) *is uncontrollable (that is, the set of values of λ for which* (70) *is not satisfied).*

Proof. The proof follows directly from the standard theory for the equivalent generalized linear system (39), characterized by the matix triple $(E, A, \tilde{B})$ defined as in (40). (See also [5], for example.) □

In the single input case ($m = 1$), the solution to Problem 1 is unique and the robustness of the closed loop system cannot be controlled. In the multi-input case ($m > 1$), there are extra degrees of freedom in the design that can be specified so as to optimize a measure of the robustness of the system. The feedback gains can be parameterized in terms of the eigenvectors of the closed loop system and the eigenvectors corresponding to the desired eigenvalues can then be selected to minimize the sensitivity measure ν^2, defined by (26). The degrees of freedom in the feedback matrices are reflected precisely by the degrees of freedom available for assigning the eigenvectors. The robust eigenstructure assignment problem is formulated explicitly as follows.

PROBLEM 2. *Given real matrices J, D, C, B and a set $\mathcal{L}$ as in Problem* 1, *find real matrices $K_1, K_2 \in \mathbb{R}^{m\times n}$ and matrix $V \in \mathbb{C}^{n\times 2n}$ such that $\tilde{V} = [V^T, (V\Lambda)^T]^T$ is nonsingular,*

$$JV\Lambda^2 - (D + BK_2)V\Lambda - (C + BK_1)V = 0, \quad \Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{2n}\}, \tag{71}$$

and the measure ν^2, defined as in (26), *is minimized.*

We remark that the requirement that the matrix Λ is diagonal, together with the invertibility of $\tilde{V}$, ensures that the closed loop system is nondefective. This requirement imposes certain simple restrictions on the multiplicity of the eigenvalues that may be assigned. The condition that $\tilde{V}$ is nonsingular is also needed for a well-posed parameterization of the feedback gains in terms of V. In the next section we derive conditions for the solution of Problem 2.

4.2. Eigenstructure assignment. The objective of the design problem now is to select the modal matrix V of right eigenvectors of the closed loop polynomial $P_{cl}(\lambda)$ to satisfy condition (71) of Problem 2 for some choice of K_1, K_2. We let W denote the corresponding modal matrix of left eigenvectors of the polynomial. We assume without loss of generality that B is of full column rank. No restriction is made on the controllability of the open loop system, but it is assumed that the set of prescribed eigenvalues $\mathcal{L}$ contains each uncontrollable eigenvalue with its full multiplicity. We remark that although the values of the uncontrollable eigenvalues of the system are not affected by the feedback, the corresponding eigenvectors may be modified and the conditioning of these eigenvalues may be improved.

The next theorem provides necessary and sufficient conditions under which a given set of nondefective eigenvalues and corresponding eigenvectors can be assigned.

THEOREM 10. *Let $V \in \mathbb{C}^{n\times 2n}$ be such that $\tilde{V} = [V^T, (V\Lambda)^T]^T$ is nonsingular, where $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{2n}\}$. Then, there exist real matrices K_1, K_2, satisfying condition* (71) *of Problem* 2 *if and only if*

$$U_1^T(JV\Lambda^2 - DV\Lambda - CV) = 0, \tag{72}$$

where

$$B = [U_0, U_1]\begin{bmatrix} Z \\ 0 \end{bmatrix} \tag{73}$$

with $U = [U_0, U_1]$ orthogonal and Z nonsingular. The matrices K_1, K_2 are given explicitly by

$$[K_1, K_2] = Z^{-1}U_0^T(JV\Lambda^2 - DV\Lambda - CV)\tilde{V}^{-1}. \tag{74}$$

Proof. The assumption that B is full rank implies the existence of the decomposition (73). Condition (71) then holds if and only if the feedback matrices K_1, K_2 satisfy

$$B[K_1, K_2]\begin{bmatrix} V \\ V\Lambda \end{bmatrix} = (JV\Lambda^2 - DV\Lambda - CV). \tag{75}$$

Premultiplication by U^T gives

$$\begin{aligned} Z[K_1, K_2]\tilde{V} &= U_0^T(JV\Lambda^2 - DV\Lambda - CV), \\ 0 &= U_1^T(JV\Lambda^2 - DV\Lambda - CV). \end{aligned} \tag{76}$$

If condition (71) is satisfied, then (72) and (74) follow directly, since $\tilde{V}$ is invertible by assumption. Conversely, if (72) is satisfied and $\tilde{V}$ is nonsingular, then K_1, K_2 given by (74) exist and satisfy (71). (See also [15], [5].) ∎

An immediate consequence of Theorem 10 is the following.

COROLLARY 11. *The right eigenvector* $\mathbf{v}_j$ *of* $P_{cl}(\lambda)$ *corresponding to the prescribed eigenvalue* $\lambda_j \in \mathcal{L}$ *must belong to the space*

$$\mathcal{S}_j = \mathcal{N}\{U_1^T(\lambda_j^2 J - \lambda_j D - C)\}, \tag{77}$$

where $\mathcal{N}\{\cdot\}$ *denotes right nullspace. The dimension of* $\mathcal{S}_j$ *is given by*

$$\dim(\mathcal{S}_j) = m + k_{\lambda_j}, \tag{78}$$

where $k_{\lambda_j} = \dim(\mathcal{N}\{[B, \lambda_j^2 J - \lambda_j D - C]^T\})$.

Proof. From (72) we obtain immediately that

$$U_1^T(\lambda_j^2 J - \lambda_j D - C)\mathbf{v}_j = 0, \tag{79}$$

and therefore $\mathbf{v_j} \in \mathcal{S}_j,\ j = 1, \ldots, 2n$, is necessary. Using (72)and (73) we find that

$$U^T[B, \lambda_j^2 J - \lambda_j D - C] = \begin{bmatrix} Z & U_0^T(\lambda_j^2 J - \lambda_j D - C) \\ 0 & U_1^T(\lambda_j^2 J - \lambda_j D - C) \end{bmatrix}. \tag{80}$$

From the definition of k_{λ_j}, we find also that $\operatorname{rank}(U^T[B, \lambda_j^2 J - \lambda_j D - C]) = n - k_{\lambda_j}$. Since matrix Z is square $(m \times m)$ and invertible, we then have $\operatorname{rank}(U_1^T(\lambda_j^2 J - \lambda_j D - C)) = n - m - k_{\lambda_j}$, from which (78) readily follows. ∎

From Corollary 11 we can now deduce restrictions on the set $\mathcal{L}$ of eigenvalues that can be assigned. If the system (1) is completely controllable, then the dimension k_λ is zero for all λ. For the closed loop polynomial to be nondefective, the maximum multiplicity of any eigenvalue λ_j that can be assigned is then equal to $\dim(\mathcal{S}_j) = m$. If the system is not completely controllable and $\lambda_j \in \mathcal{S}$ is an uncontrollable eigenvalue, then there exists a set of at least k_{λ_j} independent (left) eigenvectors of the polynomial $P_{cl}(\lambda)$ for every choice of K_1, K_2. The eigenvalue λ_j must, therefore, be assigned with multiplicity at least k_{λ_j} and at most $\dim(\mathcal{S}_j) = m + k_{\lambda_j}$.

As a consequence of Theorem 10 we can also derive explicit expressions for the feedback matrices directly in terms of the right and left modal matrices V, W of the closed loop polynomial.

COROLLARY 12. *Let* V *be such that* $\tilde{V} = [V^T, (V\Lambda)^T]^T$ *is nonsingular and condition* (72) *of Theorem* 10 *is satisfied and let* $W^H J = \tilde{V}^{-1}[0, I]^T$. *Then the feedback matrices* K_1, K_2 *satisfying condition* (71) *of Problem* 2 *are given explicitly by*

$$\begin{aligned} K_1 &= Z^{-1}U_0^T(JV\Lambda^3 W^H J - J(V\Lambda^2 W^H J)^2 - C), \\ K_2 &= Z^{-1}U_0^T(JV\Lambda^2 W^H J - D). \end{aligned} \tag{81}$$

Proof. By definition, the matrices V, Λ, W form a Jordan triple of the closed loop polynomial $P_{cl}(\lambda)$ for some K_1, K_2. Then conditions (9) hold and W^H satisfies

$$\Lambda^2 W^H J - \Lambda W^H D - W^H C = \Lambda W^H B K_2 + W^H B K_1. \tag{82}$$

Premultiplying by $Z^{-1}U_0^T JV$, using $Z^{-1}U_0^T B = I$, and applying (9) then gives the result for K_2. Similarly, premultiplying (82) by $Z^{-1}U_0^T JV\Lambda$, applying (9), and substituting for K_2 then gives the result for K_1. (Alternatively, (81) can be established using the definition of $\tilde{V}^{-1}$ given by Lemma 1 in the closed loop case.) ∎

The solution to Problem 2 can now be found by selecting the columns $\mathbf{v}_j$ of V from the subspaces $\mathcal{S}_j,\ j = 1, \ldots, 2n$, such that the matrix $\tilde{V} = [V^T, (V\Lambda)^T]^T$ is nonsingular and the robustness measure ν^2 is minimized. The required feedback matrices K_1, K_2 can then be constructed directly from (74) or (81). In the next section we show that this solution can also be obtained by solving the eigenstructure assignment problem for the corresponding generalized linear control system. Methods previously developed for optimizing the robustness of the linear system subject to structured perturbations are then adapted to solve the quadratic control design problem.

4.3. Reformulation of the control problem. In order to solve the control design problem, Problem 1, it is common practice to transform the second-order control system (1) into a generalized linear state-space (descriptor) system of the form (39), where the coefficients $E, A, \tilde{B}$ are given by (40). The matrix $\tilde{B}$ is assumed, without loss of generality, to be of full column rank.

The control problem is now to synthesize a proportional state feedback controller of the form

$$\mathbf{u} = \tilde{K}\mathbf{x} + \mathbf{r}, \tag{83}$$

where $\tilde{K} \in \mathbb{R}^{m\times 2n}$, such that the closed loop system

$$E\dot{\mathbf{x}} = (A + \tilde{B}\tilde{K})\mathbf{x} + \tilde{B}\mathbf{r} \tag{84}$$

has desired properties. Specifically, the aim is to select real matrix $\tilde{K}$ such that the $2n$ eigenvalues of the linear matrix pencil

$$L_{cl}(\lambda) \equiv \lambda E - (A + \tilde{B}\tilde{K}) \tag{85}$$

corresponding to the closed loop system (84) are equal to $\lambda_j \in \mathcal{L}$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_{2n}\}$ is a specified self-conjugate set of complex numbers. In the case where the system coefficients are given by (40) and $\tilde{K} = [K_1, K_2]$, the closed loop pencil has the form

$$L_{cl}(\lambda) = \lambda \begin{bmatrix} I & 0 \\ 0 & J \end{bmatrix} - \begin{bmatrix} 0 & I \\ C + BK_1 & D + BK_2 \end{bmatrix} \tag{86}$$

and the solution to the generalized linear inverse eigenvalue problem gives the solution to Problem 1 immediately.

The linear inverse eigenvalue problem has been studied widely and conditions for the existence of solutions are well known [31]. The eigenvalues of the closed loop pencil $L_{cl}(\lambda)$, given by (85), can be assigned arbitrarily if and only if the system (39) is completely controllable, that is, if and only if $\operatorname{rank}([\tilde{B}, \lambda E - A]) = 2n$ for all $\lambda \in \mathbb{C}$. If the system is not completely controllable, then the prescribed set $\mathcal{L}$ of eigenvalues must contain each value of λ for which the system is uncontrollable with its full multiplicity. In the case where the coefficients $E, A, \tilde{B}$ of the system are given by (40), the conditions for the existence of solutions are precisely equivalent to those of Theorem 9 for the embedded quadratic polynomial.

The robust eigenstructure assignment problem for the generalized linear system (39) has also been investigated thoroughly [18], [17], [16]. The objective is to find a nonsingular matrix $\tilde{V}$ comprising the right eigenvectors of the closed loop pencil $L_{cl}(\lambda)$ for some feedback $\tilde{K}$ such that the robustness of the closed loop system is optimized. Specifically the aim now is to minimize the sensitivity of the assigned

eigenvalues to *structured* perturbations of the form (49). The robustness measure is thus given by $\tilde{\nu}^2$, defined as in (68). For the system (39)–(40), this measure is equal to the robustness measure ν^2 of the embedded second-order system, as shown in section 3. The solution to the linear robust eigenstructure problem therefore gives the solution $V = [I, 0]\tilde{V}$ and $[K_1, K_2] = \tilde{K}$ to Problem 2 directly.

We remark that the robustness measure for the linear system is commonly taken to be the sum of the squares of the condition numbers $c^L(\lambda)$, defined as in (48). This measure gives the sensitivity of the closed loop eigenvalues to perturbations in all elements of $E, A + \tilde{B}\tilde{K}$. Its minimal value varies with the form of linear embedding used and it is not a true measure of the robustness of the quadratic polynomial. In order for the linear and quadratic inverse problems to be equivalent, it is necessary to apply the measure of robustness for the linear system with respect to the structured perturbations. The generalized linear eigenstructure problem is thus formulated explicitly as follows.

PROBLEM 3. *Given real matrices $E, A \in \mathbb{R}^{2n\times 2n}, \tilde{B} \in \mathbb{R}^{2n\times m}$, a set of $2n$ complex numbers $\mathcal{L} = \{\lambda_1, \ldots, \lambda_{2n}\}$, closed under complex conjugation, and real matrices $F \in \mathbb{R}^{2n\times m_F}, G_{\lambda_j} \in \mathbb{R}^{2n\times m_G}$, $j = 1, \ldots, 2n$, find real matrix $\tilde{K} \in \mathbb{R}^{m\times 2n}$ and nonsingular matrix $\tilde{V} \in \mathbb{C}^{2n\times 2n}$ such that*

$$E\tilde{V}\Lambda - (A + \tilde{B}\tilde{K})\tilde{V} = 0, \quad \Lambda = \operatorname{diag}\{\lambda_1, \ldots, \lambda_{2n}\}, \tag{87}$$

and $\tilde{\nu}^2 \equiv \left\| D_\omega \tilde{V}^{-1} F \right\|_F^2$ is minimized, subject to $\left\| G_{\lambda_j}^T \tilde{V} \mathbf{e}_j \right\|_2 = 1$, $j = 1, \ldots, 2n$. (Here $\mathbf{e}_j$ denotes the jth unit vector.)

Conditions under which a given set of nondefective eigenvalues and eigenvectors can be assigned to the linear system are given in the monic case in [18]. These conditions can be extended to the generalized (nonsingular) case with minor modifications and the following results can be established by similar arguments to those used in [18] and in the proof of Theorem 10.

THEOREM 13. *Let $\tilde{V} \in \mathbb{C}^{2n\times 2n}$ be nonsingular. Then, there exists real matrix $\tilde{K}$ satisfying condition* (87) *of Problem* 3 *if and only if*

$$\tilde{U}_1^T (E\tilde{V}\Lambda - A\tilde{V}) = 0, \tag{88}$$

where

$$\tilde{B} = [\tilde{U}_0, \tilde{U}_1] \begin{bmatrix} \tilde{Z} \\ 0 \end{bmatrix} \tag{89}$$

with $\tilde{U} = [\tilde{U}_0, \tilde{U}_1]$ orthogonal and $\tilde{Z}$ nonsingular. The matrix $\tilde{K}$ is given explicitly by

$$\tilde{K} = \tilde{Z}^{-1} \tilde{U}_0^T (E\tilde{V}\Lambda - A\tilde{V})\tilde{V}^{-1}. \tag{90}$$

Proof. See [18]. □

In the case where the coefficients of the control system are defined by (40), the decomposition of $\tilde{B}$ can be written in terms of the decomposition (73) of B. Using the orthogonal matrix $U = [U_0, U_1]$ from (73), we find that the decomposition (89) is given by

$$\tilde{U}_0 = \begin{bmatrix} 0 \\ U_0 \end{bmatrix}, \quad \tilde{U}_1 = \begin{bmatrix} 0 & I \\ U_1 & 0 \end{bmatrix}, \quad \tilde{Z} = Z. \tag{91}$$

The following corollary is then a direct consequence of Theorem 13.

COROLLARY 14. *Let $E, A, \tilde{B}$ be defined by* (40). *Then the right eigenvector $\tilde{\mathbf{v}}_j$ of $L_{cl}(\lambda)$ corresponding to the prescribed eigenvalue $\lambda_j \in \mathcal{L}$ must belong to the space*

$$\tilde{\mathcal{S}}_j = \mathcal{N}\{\tilde{U}_1^T(\lambda_j E - A)\} \tag{92}$$

and must satisfy $\tilde{\mathbf{v}}_j = [I, \lambda_j I]^T \mathbf{v}_j$ with $\mathbf{v}_j \in \mathcal{S}_j$, where $\mathcal{S}_j$ is defined by (77).

Proof. From (88) we immediately obtain

$$\tilde{U}_1^T(\lambda_j E - A)\tilde{\mathbf{v}}_j = 0, \tag{93}$$

and therefore $\tilde{\mathbf{v}}_j \in \tilde{\mathcal{S}}_j, j = 1, \ldots, 2n$, is necessary. Using (40) and (91) in (93) then gives

$$\begin{bmatrix} -U_1^T C & U_1^T(\lambda_j J - D) \\ \lambda_j I & -I \end{bmatrix} \tilde{\mathbf{v}}_j = 0. \tag{94}$$

The second of these relations implies that $\tilde{\mathbf{v}}_j = [\mathbf{v}_j^T, (\lambda_j \mathbf{v}_j)^T]^T$ and the first establishes that $\mathbf{v}_j \in \mathcal{S}_j$ is necessary. ∎

Finally, from the result (90) of Theorem 13 and from (91) we may establish a direct relation between the solutions to the linear and quadratic feedback design problems.

COROLLARY 15. *Let $E, A, \tilde{B}$ be defined by* (40). *Let $\tilde{V}$ be a nonsingular matrix satisfying condition* (88) *of Theorem* 13 *and let $V = [I, 0]\tilde{V}$, $W^H J = \tilde{V}^{-1}[0, I]^T$. Then, the feedback matrix $\tilde{K}$ satisfying condition* (87) *of Problem* 3 *is equal to $\tilde{K} = [K_1, K_2]$, where K_1, K_2 are defined by* (74), *or equivalently, by* (81).

Proof. Substituting (40) and (91) into (90) gives the result immediately. ∎

In summary, the solution to Problem 3 can then be found by selecting the columns $\tilde{\mathbf{v}}_j$ of $\tilde{V}$ from the subspaces $\tilde{\mathcal{S}}_j$, $j = 1, \ldots, 2n$, such that the matrix $\tilde{V}$ is nonsingular and the robustness measure $\tilde{\nu}^2 \equiv \nu^2$ is minimized, subject to the constraints $\|G_{\lambda_j}^T \tilde{\mathbf{v}}_j\|_2 = 1$. The required feedback matrix $\tilde{K}$ can then be constructed from (90). If $E, A, \tilde{B}$ are given by (40) and $F, G_{\lambda_j} = [\lambda_j G_1, G_2]$, $j = 1, \ldots, 2n$, are determined by (58), then the solution to Problem 3 immediately gives the solution $V = [I, 0]\tilde{V}$, $[K_1, K_2] = \tilde{K}$ to the quadratic robust eigenstructure assignment problem, Problem 2.

5. Numerical algorithm. Previously, in [16], we have developed a numerical algorithm for solving the linear robust eigenstructure assignment problem subject to structured perturbations. In the monic case this method can be applied directly to solve Problem 3. The algorithm is easily adapted to treat the generalized case. The method does not, however, take direct advantage of the special structure of the linear pencil in the case where the linear system represents an embedded quadratic system.

We now present a modified form of the algorithm that can be applied to solve the robust quadratic eigenstructure problem, Problem 2.

5.1. Basic steps. The basic steps of the algorithm are first described. Details of the implementation are then discussed.

ALGORITHM 1.

Input: *Real matrices $J, D, C \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, a set of $2n$ complex numbers $\mathcal{L} = \{\lambda_1, \ldots, \lambda_{2n}\}$, and a diagonal matrix $D_\omega = \mathrm{diag}\{\omega_1, \ldots, \omega_{2n}\}$, where ω_j, $j = 1, \ldots, 2n$, are real positive weights satisfying $\sum_{j=1}^{2n} \omega_j^2 = 1$ with $\omega_j = \omega_k$ if $\lambda_j = \lambda_k$.*

Step 1. *Find the decomposition* (73) *of B and an orthonormal basis, comprised by the columns of the matrix S_j, for the subspaces $\mathcal{S}_j$, $j = 1, \ldots, 2n$, defined in* (77).

Step 2. *Select an initial matrix* $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{2n}]$ *such that* $\mathbf{v}_j \in \mathcal{S}_j$, $\alpha_j \|\mathbf{v}_j\|_2 = 1$, *and* $\tilde{V} = [V^T, (V\Lambda)^T]^T$ *is nonsingular, where* $\Lambda = \text{diag}\{\lambda_j,\ j = 1, \ldots, 2n\}$ *and* $\alpha_j = (|\lambda_j|^4 + |\lambda_j|^2 + 1)^{\frac{1}{2}}$.

Step 3. *For* $j = 1, 2, \ldots, 2n$ *do*

Step 3.1. *Find vector* $\hat{\mathbf{v}}_j$ *that minimizes*

$$\nu^2 = \|D_\omega W^H J\|_F^2 \equiv \left\| D_\omega \begin{bmatrix} V \\ V\Lambda \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \right\|_F^2$$

over all $\mathbf{v}_j \in \mathcal{S}_j$ *subject to* $\alpha_j \|\mathbf{v}_j\|_2 = 1$ *and* $\mathbf{v}_i$ *fixed for all* $i \neq j$.

Step 3.2. *Form updated matrices* $V = [\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}, \hat{\mathbf{v}}_j, \mathbf{v}_{j+1}, \ldots, \mathbf{v}_{2n}]$ *and* $\tilde{V} = [V^T, (V\Lambda)^T]^T$ *and CONTINUE.*

Step 4. *Repeat Step* 3 *until* ν^2 *has "converged."*

Step 5. *Construct feedback matrices* K_1, K_2 *by solving*

$$[K_1, K_2]\tilde{V} = Z^{-1}U_0^T(JV\Lambda^2 - DV\Lambda - CV). \qquad \square$$

We remark that the decomposition of B in Step 1 can be found either by the QR or the SVD method (see [10]). The matrix S_j can be found from the QR decomposition of $(U_1^T(\lambda_j^2 J - \lambda_j D - C))^T$.

If the system (1) is completely controllable and the prescribed eigenvalues are distinct, then an initial matrix V satisfying the requirements in Step 2 can always be selected. (Under mild restrictions, this also holds for uncontrollable systems and/or for prescribed multiple eigenvalues.) To obtain the initial matrix V it is generally sufficient to select *random* vectors from each subspace $\mathcal{S}_j$. The conditioning of the initial matrix $\tilde{V}$ is not significant and it may be very close to singular without detriment.

The key step of the algorithm is Step 3. Details of the procedure used for updating the eigenvectors in Step 3.1 are discussed in the next section. If the initial matrix $\tilde{V}$ is nonsingular, then each subsequent matrix $\tilde{V}$ generated in this step is guaranteed also to be nonsingular. The update is selected to minimize the robustness measure ν^2 over all vectors in the finite dimensional subspace $\mathcal{S}_j$. The sequence of values of ν^2 generated by the iteration process is therefore nonincreasing and bounded from below, and hence the iteration must converge.

The problem of computing the feedback matrices in Step 5 from the constructed matrix $\tilde{V}$ is well conditioned if $\tilde{V}$ is well conditioned for inversion. Since the aim of the procedure is essentially to orthogonalize $\tilde{V}$ with respect to $[0, I]^T$, $\tilde{V}$ is expected to be reasonably well-conditioned. Additional degrees of freedom in $\tilde{V}$ may exist, however, and these are then selected *explicitly* in Step 3.1 to make $\tilde{V}$ as well conditioned as possible. If the constructed matrix $\tilde{V}$ is, nevertheless, very badly conditioned, then the closed loop system will necessarily be very sensitive to perturbations, regardless of the accuracy of the computed feedback gains. It is then recommended that the set of prescribed eigenvalues should be altered, allowing a less sensitive closed loop system to be derived.

As an alternative to the procedure in Step 5, the matrices K_1, K_2 could be determined by solving for $W^H J$ from $\tilde{V}(W^H J) = [0, I]^T$ and then substituting directly into (81). Analysis suggests, however, that this procedure will be less efficient and less accurate than that proposed in Step 5. The solution for $W^H J$ requires the inversion of $\tilde{V}$ into n right-hand-side vectors, whereas the solution in Step 5 requires the inversion of $\tilde{V}$ into only $m \leq n$ right-hand sides. Moreover, forming the product

of the computed $W^H J$ with the other factors in (81), which already contain numerical errors, is likely to magnify the computational errors introduced into the feedback matrices and hence to give less accurate solutions.

5.2. Updating the eigenvectors. The computation of the update to the vector $\mathbf{v}_j$ in Step 3.1 of the algorithm is accomplished explicitly. In essence, this step aims to orthogonalize the vectors $\tilde{\mathbf{v}}_j = [I, \lambda_j I]^T \mathbf{v}_j$, $j = 1, \ldots, 2n$, with respect to the matrix $[0, I]^T$, subject to the constraints. In the first phase, orthogonal bases Q and $\mathbf{q}$ are found for the space spanned by the fixed vectors $\tilde{\mathbf{v}}_i$, $i \neq j$, and its orthogonal complement, respectively, and the measure ν^2 is expressed in terms of these bases. Next, the required vector is scaled to have a fixed normalization and the direction of the minimizing vector in the required subspace is found by solving a least squares problem. The optimal normalization is then determined to satisfy the constraint. These steps follow the algorithm of [16], but are modified to produce the vector $\mathbf{v}_j$ as efficiently as possible. The technical details are as follows.

We denote $\tilde{V}_j = [\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_{j-1}, \tilde{\mathbf{v}}_{j+1}, \ldots, \tilde{\mathbf{v}}_{2n}]$ and let the (complex) QR decomposition of $\tilde{V}_j$ be given by

$$\tilde{V}_j = [Q, \mathbf{q}] \begin{bmatrix} R \\ \mathbf{0}^T \end{bmatrix}, \tag{95}$$

where $[Q, \mathbf{q}]$ is orthogonal and R is upper triangular and nonsingular. We write $\mathbf{v}_j = S_j \boldsymbol{\eta} \in \mathcal{S}_j$. Then we obtain $\nu^2 = \left\| D_\omega \tilde{V}^{-1} [0, I]^T \right\|_F^2 = \|Y\|_F^2$, where

$$\begin{aligned} Y &= \begin{bmatrix} D_{\hat{\omega}} & 0 \\ 0 & \omega_j \end{bmatrix} \left[\tilde{V}_j, \begin{bmatrix} I \\ \lambda_j I \end{bmatrix} S_j \boldsymbol{\eta} \right]^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \\ &= \begin{bmatrix} D_{\hat{\omega}} & 0 \\ 0 & \omega_j \end{bmatrix} \begin{bmatrix} R^{-1} & -\rho R^{-1} Q^H [I, \lambda_j I]^T S_j \boldsymbol{\eta} \\ 0 & \rho \end{bmatrix} \begin{bmatrix} Q_2^H \\ \mathbf{q}_2^H \end{bmatrix} \\ &= \begin{bmatrix} D_{\hat{\omega}} & 0 \\ 0 & \omega_j \end{bmatrix} \begin{bmatrix} R^{-1} Q_2^H - R^{-1} (Q_1^H + \lambda_j Q_2^H) S_j \boldsymbol{\eta} \rho \mathbf{q}_2^H \\ \rho \mathbf{q}_2^H \end{bmatrix}, \end{aligned} \tag{96}$$

and $D_{\hat{\omega}} = \operatorname{diag}\{\omega_1, \ldots, \omega_{j-1}, \omega_{j+1}, \ldots, \omega_{2n}\}$, $\rho = \left(\mathbf{q}^H [I, \lambda_j I]^T S_j \boldsymbol{\eta}\right)^{-1}$, $Q^H = [Q_1^H, Q_2^H]$, $\mathbf{q}^H = [\mathbf{q}_1^H, \mathbf{q}_2^H]$.

If $\mathbf{q}_2 \neq 0$, then using $\alpha_j \|\mathbf{v}_j\|_2 = \|\alpha_j \boldsymbol{\eta}\|_2 = 1$ and applying Lemma 2 of [16], for example, we can show that

$$\delta^2 \|Y\|_F^2 = \left\| \delta^2 \begin{bmatrix} D_{\hat{\omega}} R^{-1} (Q_1^H + \lambda_j Q_2^H) S_j \\ \alpha_j I_m \end{bmatrix} (\rho \boldsymbol{\eta}) - \begin{bmatrix} D_{\hat{\omega}} R^{-1} Q_2^H \mathbf{q}_2 \\ 0 \end{bmatrix} \right\|_F^2 + c, \tag{97}$$

where $\delta^2 = \mathbf{q}_2^H \mathbf{q}_2$ and c is a constant independent of $\boldsymbol{\eta}$.

The problem now is to minimize $\|Y\|_F^2$ over all $\boldsymbol{\eta} \in \mathbb{C}^m$. In order to reduce this nonlinear minimization problem to a linear least-squares problem, we fix the normalization of the vector $\rho \boldsymbol{\eta}$. We find the Householder transformation P such that

$$(\mathbf{q}_1^H + \lambda_j \mathbf{q}_2^H) S_j P = \sigma \mathbf{e}_m^T, \tag{98}$$

where $\mathbf{e}_m$ is the mth unit vector. From the definition of ρ, we then have

$$1 = \mathbf{q}^H [I, \lambda_j I]^T S_j \rho \boldsymbol{\eta} = (\mathbf{q}_1^H + \lambda_j \mathbf{q}_2^H) S_j P P^H \rho \boldsymbol{\eta} = \sigma \mathbf{e}_m^T P^H \rho \boldsymbol{\eta}. \tag{99}$$

We may therefore define $\hat{\boldsymbol{\eta}}$ to be such that $[\hat{\boldsymbol{\eta}}^H, 1]^H = \sigma P^H \rho \boldsymbol{\eta}$.

Writing $P = [P_1, \mathbf{p}]$ then gives $\sigma P P^H \rho \boldsymbol{\eta} = P[\hat{\boldsymbol{\eta}}^H, 1]^H = P_1\hat{\boldsymbol{\eta}} + \mathbf{p}$ and the minimization problem becomes

$$\min_{\hat{\boldsymbol{\eta}}} \left\| \delta^2 \begin{bmatrix} D_{\hat{\omega}} H \\ \alpha_j I \end{bmatrix} P_1 \hat{\boldsymbol{\eta}} + \begin{bmatrix} D_{\hat{\omega}}(\delta^2 H\mathbf{p} - \sigma \mathbf{h}) \\ \delta^2 \alpha_j \mathbf{p} \end{bmatrix} \right\|_F^2, \tag{100}$$

where $H = R^{-1}(Q_1^H + \lambda_j Q_2^H)S_j$, $\mathbf{h} = R^{-1}Q_2^H \mathbf{q}_2$. This is a standard linear least-squares problem that can be solved by the QR (or SVD) method.

Finally, we restore the scaling of the optimal vector to satisfy the constraints. Since P is orthogonal and the columns of S_j form a set of orthonormal vectors, the required update is given by

$$\hat{\mathbf{v}}_j = S_j P \begin{bmatrix} \hat{\boldsymbol{\eta}} \\ 1 \end{bmatrix} \Big/ \left\| \alpha_j \begin{bmatrix} \hat{\boldsymbol{\eta}} \\ 1 \end{bmatrix} \right\|_2. \tag{101}$$

In the special case where $\mathbf{q}_2 = 0$ (or is very small), then $\|Y\|_F$ is constant (almost), independent of $\boldsymbol{\eta}$. In this case the new vector $\hat{\mathbf{v}}_j$ could be selected to be *any* vector in $\mathcal{S}_j$. In order to maximize the orthogonality of $\tilde{V}$, however, the new vector is chosen such that $[I, \lambda_j I]^T \hat{\mathbf{v}}_j$ equals the closest vector to $\mathbf{q}$ in the allowable subspace, given by the projection of $\mathbf{q}$ into $[I, \lambda_j I]^T S_j$. The required update is then

$$\hat{\mathbf{v}}_j = S_j S_j^H (\mathbf{q}_1 + \bar{\lambda_j}\mathbf{q}_2) / \left\| \alpha_j S_j^H (\mathbf{q}_1 + \bar{\lambda_j}\mathbf{q}_2) \right\|_2. \tag{102}$$

The new updated matrix $\tilde{V}$ generated by this procedure must be nonsingular. Since the original matrix was nonsingular, the definition of $\mathbf{q}$ implies that $\mathbf{q}^H[I, \lambda_j I]^T S_j \neq 0$ and $\sigma \neq 0$. Hence $\mathbf{q}^H[I, \lambda_j I]^T \hat{\mathbf{v}}_j \neq 0$ and the vector $[I, \lambda_j I]^T \hat{\mathbf{v}}_j$ has a component in the direction orthogonal to all the other columns of $\tilde{V}$. The columns of the updated matrix $\tilde{V}$ must therefore all be linearly independent, which establishes the result.

We may summarize the update step of the algorithm as follows.

ALGORITHM 1, STEP 3.1.

Input: tol

Step 3.1.1. *Form matrix* $\tilde{V}_j$ *and find its QR decomposition* (95) *to determine* $Q = [Q_1^H, Q_2^H]^H, \mathbf{q} = [q_1^H, q_2^H]^H$, *and* R. *Form* $\delta^2 = \mathbf{q}_2^H \mathbf{q}_2$.

Step 3.1.2. *If* $|\delta^2| >$ tol, *form* $(\mathbf{q}_1^H + \lambda_j \mathbf{q}_2^H)S_j$ *and find the Householder matrix* P *satisfying* (98). *Solve* $R[H, \mathbf{h}] = [(Q_1^H + \lambda_j Q_2^H)S_j, Q_2^H \mathbf{q}_2]$ *for* $H, \mathbf{h}$ *by back-substitution and solve the least-squares problem* (100) *for* $\hat{\boldsymbol{\eta}}$.

Step 3.1.3. *If* $|\delta^2| >$ tol, *define the update* $\mathbf{v}_j$ *by* (101); *else define* $\mathbf{v}_j$ *by* (102). ∎

In the case where $\hat{\mathbf{v}}_j$ corresponds to a real eigenvalue λ_j, the method generates a real update. In the case where λ_j is complex, a complex eigenvector is generated and, in order to ensure that the computed feedback matrices are real, the updated eigenvector corresponding to the conjugate eigenvalue $\bar{\lambda_j}$ must be taken to be the conjugate vector $\bar{\hat{\mathbf{v}}}_j$. In practice, complex arithmetic can be avoided by generating the real and imaginary parts of $\hat{\mathbf{v}}_j$ independently. The optimization is no longer precise, however, and a reduction in ν^2 cannot be guaranteed at every iteration step. Experience indicates that this is not a drawback and rapid overall convergence is obtained in practice.

We remark that the QR decomposition of $\tilde{V}_j$ can be found by inexpensive updating techniques from the QR decomposition of $\tilde{V}_{j-1}$. The solution of the least-squares

problem (100) requires the decomposition of a matrix of order $m-1$, which may be small even where the order $2n$ of the full system is large. The procedure is then relatively efficient. Each update requires $\mathcal{O}(6n^2m)+\mathcal{O}(2nm^2)$ operations. Practical experiments have shown that the reduction of the minimization problem to a sequence of linear least-squares problems is generally more efficient than global nonlinear optimization techniques for objective functions of this form [21]. Further work on the procedure for maximizing robustness would, however, be useful.

Overall, the algorithm is considerably more efficient than the method of [16] for treating the $2n \times 2n$ linearized eigenstructure problem directly. The primary advantage results from operating with the $n \times m$ subspaces $\mathcal{S}_j$ instead of the $2n \times m$ subspaces $\tilde{\mathcal{S}}_j$. In addition to reductions in the work required in the first step of the procedure, further savings are achieved in the update step. The number of operations saved is of the order of $\mathcal{O}(16n^3m)+\mathcal{O}(6n^2m^2)$ *per iteration*. The new algorithm therefore provides a significant improvement in the solution of the robust quadratic eigenstructure assignment problem.

5.3. Examples. The application of the algorithm is demonstrated with two examples.

Example 1. The first example is a third-order system with two inputs, given in [5], and is defined by

$$J = 10I_3, \quad D = 0, \quad C = \begin{bmatrix} -40 & 40 & 0 \\ 40 & -80 & 40 \\ 0 & 40 & -40 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 3 & 2 \\ 3 & 4 \end{bmatrix}. \tag{103}$$

The system is undamped and the open loop eigenvalues (to five figures) are

$$\{\pm 3.6039i, \ \pm 2.4940i, \ \pm 0.89008i\}. \tag{104}$$

The desired closed loop eigenvalues $\lambda_j,\ j = 1, \ldots, 2n$, are given by

$$\mathcal{L} = \{-1, \ -2, \ -3, \ -4, \ -5, \ -6\}. \tag{105}$$

The initial matrix V is generated by a random selection of vectors from each subspace. The corresponding matrix $\tilde{V}$ has condition number $\kappa = 3438$ and the value of the sensitivity measure is $\nu = 2808$. After one iteration of the algorithm the sensitivity measure is reduced to $\nu = 137.6$ and after three iterations the value becomes $\nu = 136.6$. The conditioning of $\tilde{V}$ is then $\kappa = 497.2$ and the computed feedback matrices are given (to five figures) by

$$\begin{aligned} K_1 &= \begin{bmatrix} -1.2566 & -44.622 & 120.23 \\ 56.184 & 42.276 & -227.72 \end{bmatrix}, \\ K_2 &= \begin{bmatrix} 86.175 & -27.228 & -16.522 \\ -85.494 & 13.016 & -4.9924 \end{bmatrix}. \end{aligned} \tag{106}$$

Further iterations give no significant improvement in the robustness measure and the algorithm is stopped. For the feedback matrices obtained by the procedure, the computed eigenvalues of the closed loop system equal the desired eigenvalues to within an error of $\pm 1 \times 10^{-13}$. The results compare favorably with those obtained in [5] for this problem. Not only are the eigenvalues more accurately assigned and the matrix of eigenvectors more well-conditioned, but the components of the feedback matrices are

significantly smaller in magnitude, and therefore the required control effort is much reduced.

The computed condition numbers $c(\lambda_j),\ j = 1, \ldots, 2n$, of the assigned eigenvalues are given (to four figures) by

$$\{11.44,\ 68.21,\ 77.68,\ 49.22,\ 22.21,\ 70.22\}. \tag{107}$$

Introducing perturbations of the form $J\delta D,\ J\delta C$ into the closed loop coefficient matrices $D + BK_2,\ C + BK_1$, where $\delta D,\ \delta C$ are random matrices bounded such that $\|[\delta D, \delta C]\|_2 \leq \epsilon = 0.002$, produces perturbations in the assigned closed loop eigenvalues of order $\mathcal{O}(0.01)$. The largest errors occur in the third eigenvalue, which is expected to be the most sensitive to system perturbations. The perturbations in the eigenvalues are well within the theoretical error estimates given by the product of the condition numbers with ϵ. The absolute errors introduced into the coefficient matrices are of order $\mathcal{O}(0.01)$. The perturbations in the eigenvalues are thus of the same order of magnitude as the perturbations in the system matrices and the solution is therefore very *robust*.

Example 2. In the second example we examine a case from [4] where the matrix J is very ill-conditioned. The system matrices $D,\ C$, and B are the same as in Example 1, and the matrix J is defined by

$$J = \begin{bmatrix} 5000 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1.00001 \end{bmatrix}. \tag{108}$$

The open loop spectrum becomes (to five figures)

$$\{\pm 4899.0i,\ \pm 4.4726i,\ \pm 0.051635i\}. \tag{109}$$

The desired closed loop eigenvalues $\lambda_j,\ j = 1, \ldots, 2n$, are again prescribed to be

$$\mathcal{L} = \{-1,\ -2,\ -3,\ -4,\ -5,\ -6\}. \tag{110}$$

The condition number of J is 1×10^9. Since the inverse of the matrix J is not required by the algorithm, the numerical stability of the procedure is not affected by this ill-conditioning.

The initial matrix V is again generated by a random selection of vectors from each subspace. The corresponding matrix $\tilde{V}$ has condition number $\kappa = 5.24 \times 10^5$ and the sensitivity measure is $\nu = 4.47 \times 10^5$. After one iteration of the method the sensitivity becomes $\nu = 3.298 \times 10^4$ and after three iterations it is reduced to $\nu = 3.156 \times 10^4$. The conditioning of $\tilde{V}$ is then $\kappa = 7.622 \times 10^4$ and the computed feedback matrices (to five figures) are given by

$$\begin{aligned} K_1 &= \begin{bmatrix} 2668.9 & 59.004 & -58.413 \\ 19.996 & -60.000 & 60.000 \end{bmatrix}, \\ K_2 &= \begin{bmatrix} 1291.4 & -3.2463 & -3.1778 \\ -0.018680 & 0.45542 \times 10^{-5} & -0.53043 \times 10^{-4} \end{bmatrix}. \end{aligned} \tag{111}$$

With these feedback matrices, the computed eigenvalues of the closed loop system are equal to the desired eigenvalues to at least 8 figures of accuracy. The feedbacks are comparable to those obtained in [4], the large gains reflecting the amount of control

effort needed to move the eigenvalues to the prescribed positions. Minor improvements are achieved by further iterations of the method, but the sensitivity of the problem remains of the same order of magnitude even after 300 iterations.

The condition numbers $c(\lambda_j),\ j = 1, \ldots, 2n$, of the assigned eigenvalues are given (to four figures) by

$$\{0.1722,\ 0.9689,\ 2.191,\ 0.3722,\ 0.8464,\ 1.827\} \times 10^4. \tag{112}$$

Perturbations of the form $J\delta D,\ J\delta C$ in the closed loop coefficient matrices $D + BK_2,\ C + BK_1$, where $\delta D,\ \delta C$ are random matrices bounded such that $\|[\delta D, \delta C]\|_2 \leq \epsilon = 2\times 10^{-6}$, now lead to perturbations in the closed loop eigenvalues of order $\mathcal{O}(0.001)$ with the largest again occuring in the third eigenvalue as expected. The perturbations in the eigenvalues are well within the errors estimated by the product of the condition numbers with ϵ. The absolute perturbations in the coefficient matrices are now of order $\mathcal{O}(0.01)$, however, because of the large size of the matrix J. The perturbations in the eigenvalues are actually less in magnitude than the errors in the system matrices and the closed loop system is robust.

The effects of relative perturbations in the ill-conditioned matrix J are interesting to note. Perturbations $J\delta J,\ J\delta D,\ J\delta C$ in the closed loop system matrices $J,\ D + BK_2,\ C + BK_1$, where $\delta J,\ \delta D,\ \delta C$ are random matrices bounded such that $\|[\delta J, \delta D, \delta C]\|_2 \leq \epsilon = 3 \times 10^{-6}$, now lead to perturbations in the assigned eigenvalues of order $\mathcal{O}(0.01)$. The perturbations in the eigenvalues therefore remain within the theoretical error estimate given by the product of the condition number with ϵ. The absolute errors introduced into the coefficient matrices correspond to perturbations in the sixth figure in the components of J and are thus of order $\mathcal{O}(0.01)$. The perturbations in the eigenvalues are therefore of the same size as the errors in the system matrices and, despite the ill-conditioning of the matrix J, the closed loop system is robust with respect to these "relative" perturbations.

6. Conclusions. We have investigated here the problem of robust eigenstructure assignment by state feedback in a second-order control system. The response of the system is determined by the eigenstructure of the associated quadratic matrix polynomial and the aim of the controller design is to assign specified eigenvalues to the closed loop system polynomial.

In the first sections of the paper we derive sensitivity measures, or condition numbers, for the eigenvalues of the quadratic matrix polynomial and define a measure of the robustness of the corresponding system. In practice the second-order system is commonly embedded in a generalized linear first-order control system. The standard measure of sensitivity, or robustness, of the corresponding generalized linear matrix pencil is not equivalent to that of the embedded quadratic polynomial. We show, however, that an equivalent robustness measure for the linear pencil can be established by considering its sensitivity to *structured* perturbations. We derive condition numbers for the eigenvalues of the generalized linear pencil subject to perturbations with specified structure and show that these condition numbers are equal to the sensitivity measures for the embedded quadratic polynomial. We show also that the robustness measures based on these condition numbers are equal.

In the remaining sections of the paper we review and extend the theory of eigenstructure assignment in second-order control systems. We show that the solution of the robust eigenstructure assignment problem for the second-order system can be achieved by solving the generalized linear problem subject to structured perturbations. Reliable and efficient numerical methods for determining the required feedback

matrices are then developed, based on methods previously devised for solving the structured linear problem.

REFERENCES

[1] M.J. Balas, *Trends in large space structure control theory: Fondest hopes wildest dreams*, IEEE Trans. Automat. Control, 27 (1982), pp. 522–535.

[2] A. Bhaya and C. Desoer, *On the design of large flexible space structures*, IEEE Trans. Automat. Control, 30 (1985), pp. 1118–1120.

[3] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N.K. Nichols, *Feedback design for regularizing descriptor systems*, Linear Algebra Appl., 299 (1999), pp. 119–151.

[4] H.C. Chan, J. Lam, and D.W.C. Ho, *Robust eigenvalue assignment in second-order systems: A gradient flow approach*, Optim. Control Appl. Methods, 18 (1997), pp. 283–296.

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

[6] R.W. Clough and S. Mojtahedi, *Earthquake response analysis considering non-proportional damping*, Earthquake Engrg. Structural Dynam., 4 (1976), pp. 489–496.

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

[8] B.N. Datta and F. Rincón, *Feedback stabilization of a second-order system: A nonmodal approach*, Linear Algebra Appl., 188–189 (1993), pp. 135–161.

[9] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, London, 1982.

[10] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[11] 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.

[12] D.W. Ho and H.C. Chan, *Feedback Stabilization of Damped-Gyroscopic Second-Order Systems*, Research report, Report No. MA–93–14, Faculty of Science and Technology, City University, Hong Kong, 1993.

[13] M. Hou and P.C. Mueller, *LQ and Tracking Control of Descriptor Systems with Application to Constrained Manipulator*, Technical report, Sicherheitstechnische Regelungs-und Messtechnik, Universitaet Wuppertal, Wuppertal, 1994.

[14] S.M. Joshi, *Control of Large Flexible Space Structures*, Lecture Notes in Control and Inform. Sci. 131, Springer-Verlag, Berlin, 1989.

[15] J. Juang and P.G. Maghami, *Robust eigensystem assignment for second-order dynamics systems*, in Mechanics and Control of Large Flexible Structures, Progress in Astronautics and Aeronautics, Vol. 129, AIAA, Washington, DC, 1990, pp. 373–388.

[16] J. Kautsky and N.K. Nichols, *Robust pole assignment in systems subject to structured perturbations*, Systems Control Lett., 15 (1990), pp. 373–380.

[17] J. Kautsky, N.K. Nichols, and E.K.-W. Chu, *Robust pole assignment in singular control systems*, Linear Algebra Appl., 21 (1989), pp. 9–37.

[18] J. Kautsky, N.K. Nichols, and P. Van Dooren, *Robust pole assignment in linear state feedback*, Internat. J. Control, 41 (1985), pp. 1129–1155.

[19] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966.

[20] H. Langer, B. Najman, and K. Veselić, *Perturbation of the eigenvalues of quadratic matrix polynomials*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 474–489.

[21] D.M. Littleboy and N.K. Nichols, *Modal coupling in linear control systems using robust eigenstructure assignment*, Linear Algebra Appl., 275–276 (1998), pp. 359–379.

[22] L. Meirovitch, H. Baruh, and H. Oz, *A comparison of control techniques for large flexible systems*, J. Guidance, 6 (1983), pp. 302–310.

[23] P.C. Mueller, *Linear quadratic control of mechanical descriptor systems*, in Systems and Networks: Mathematical Theory and Applications, Vol. II, U. Helmke, R. Mennicken, and J. Saurer, eds., Akademie-Verlag, Birkhaeuser, Berlin, 1994, pp. 361–366.

[24] M. Radjabalipour and A. Salem, *On eigenvalues of quadratic matrix polynomials and their perturbations*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 563–569.

[25] B. Simeon, F. Grupp, C. Fuehrer, and P. Rentrop, *A Nonlinear Truck Model and its Treatment as a Multibody System*, Technical report, Mathematisches Institut, Technische Universitaet Muenchen, 1992.

[26] H.A. Smith, R.K. Singh, and D.C. Sorensen, *Formulation and solution of the non-linear, damped eigenvalue problem for skeletal systems*, Internat. J. Numer. Methods Engrg., 38 (1995), pp. 3071–3085.

[27] G.W. Stewart, *Gerschgorin theory for the generalized eigenvalue problem* $Ax = \lambda Bx$, Math. Comp., 29 (1975), pp. 600–606.
[28] G.W. Stewart, *On the sensitivity of the eigenvalue problem* $Ax = \lambda Bx$, SIAM J. Numer. Anal., 9 (1972), pp. 669–686.
[29] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[30] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.
[31] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed., Springer, New York, 1979.
[32] Z.C. Zheng, G.X. Ren, and W.J. Wang, *A reduction method for large scale unsymmetric eigenvalue problems in structural dynamics*, J. Sound Vibration, 199 (1997), pp. 253–268.

SIAM J. Matrix Anal. Appl.
Vol. 23, No. 1, pp. 103–125
© 2001 Society for Industrial and Applied Mathematics

STABILITY OF STRUCTURED HAMILTONIAN EIGENSOLVERS*

FRANÇOISE TISSEUR†

Abstract. Various applications give rise to eigenvalue problems for which the matrices are Hamiltonian or skew-Hamiltonian and also symmetric or skew-symmetric. We define structured backward errors that are useful for testing the stability of numerical methods for the solution of these four classes of structured eigenproblems. We introduce the symplectic quasi-QR factorization and show that for three of the classes it enables the structured backward error to be efficiently computed. We also give a detailed rounding error analysis of some recently developed Jacobi-like algorithms of Faßbender, Mackey, and Mackey [*Linear Algebra Appl.*, to appear] for these eigenproblems. Based on the direct solution of 4×4, and in one case 8×8, structured subproblems these algorithms produce a complete basis of symplectic orthogonal eigenvectors for the two symmetric cases and a symplectic orthogonal basis for all the real invariant subspaces for the two skew-symmetric cases. We prove that, when the rotations are implemented using suitable formulae, the algorithms are strongly backward stable and we show that the QR algorithm does not have this desirable property.

Key words. Hamiltonian, skew-Hamiltonian, symmetric, skew-symmetric, symplectic, backward error, structure-preserving, rounding error, Jacobi algorithm, quaternion rotation

AMS subject classifications. 65F15, 65G05

PII. S0895479800368007

1. Introduction.

This work concerns real structured Hamiltonian and skew-Hamiltonian eigenvalue problems where the matrices are either symmetric or skew-symmetric. We are interested in algorithms that are strongly backward stable for these problems. In general, a numerical algorithm is called *backward stable* if the computed solution is the true solution for slightly perturbed initial data. If, in addition, this perturbed initial problem has the same structure as the given problem, then the algorithm is said to be *strongly backward stable*.

There are three reasons for our interest in strongly backward stable algorithms. First, such algorithms preserve the algebraic structure of the problem and hence force the eigenvalues to lie in a certain region of the complex plane or to occur in particular kinds of pairings. Because of rounding errors, algorithms that do not respect the structure of the problem can cause eigenvalues to leave the required region [26]. Second, by taking advantage of the structure, storage and computation can be lowered. Finally, structure-preserving algorithms may compute eigenpairs that are more accurate than the ones provided by a general algorithm.

Structured Hamiltonian eigenvalue problems appear in many scientific and engineering applications. For instance, symmetric skew-Hamiltonian eigenproblems arise in quantum mechanical problems with time reversal symmetry [9], [23]. In response theory, the study of closed shell Hartree–Fock wave functions yields a linear response eigenvalue equation with a symmetric Hamiltonian [21]. Also, total least squares problems with symmetric constraints lead to the solution of a symmetric Hamiltonian problem [17].

*Received by the editors February 23, 2000; accepted for publication (in revised form) by V. Mehrmann November 24, 2000; published electronically May 3, 2001.
http://www.siam.org/journals/simax/23-1/36800.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.

The motivation for this work comes from recently developed Jacobi algorithms for structured Hamiltonian eigenproblems [10]. These algorithms are structure-preserving, inherently parallelizable, and hence attractive for solving large-scale eigenvalue problems. Our first contribution is to define and show how to compute structured backward errors for structured Hamiltonian eigenproblems. These backward errors are useful for testing the stability of numerical algorithms. Our second contribution concerns the stability of these new Jacobi-like algorithms. We give a unified description of the algorithms for the four classes of structured Hamiltonian eigenproblems. This provides a framework for a detailed rounding error analysis and enables us to show that the algorithms are strongly backward stable when the rotations are implemented using suitable formulae.

The organization of the paper is as follows. In section 2 we recap the necessary background concerning structured Hamiltonians. In section 3 we derive computable structured backward errors for structured Hamiltonian eigenproblems. In section 4, we describe the structure-preserving QR-like algorithms proposed in [5] for structured Hamiltonian eigenproblems. We give a unified description of the new Jacobi-like algorithms and detail the Jacobi-like update for each of the four classes of structured Hamiltonian. In section 5 we give the rounding error analysis and in section 6 we use our computable backward errors to confirm empirically the strong stability of the Jacobi algorithms.

2. Preliminaries. A matrix $P \in \mathbb{R}^{2n\times 2n}$ is *symplectic* if $P^TJP = J$, where $J = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$ and I_n is the $n \times n$ identity matrix.

A matrix $H \in \mathbb{R}^{2n\times 2n}$ is *Hamiltonian* if $JH = (JH)^T$ is symmetric. Hamiltonian matrices have the form

$$H = \begin{bmatrix} E & F \\ G & -E^T \end{bmatrix},$$

where $E, F, G \in \mathbb{R}^{n\times n}$ and $F^T = F,\ G^T = G$. We denote the set of real Hamiltonian matrices by $\mathcal{H}_{2n}$.

A matrix $S \in \mathbb{R}^{2n\times 2n}$ is *skew-Hamiltonian* if $JS = -(JS)^T$ is skew-symmetric. Skew-Hamiltonian matrices have the form

$$S = \begin{bmatrix} E & F \\ G & E^T \end{bmatrix},$$

where $E, F, G \in \mathbb{R}^{n\times n}$ and $F^T = -F,\ G = -G^T$ are skew-symmetric. We denote the set of real skew-Hamiltonian matrices by $\mathcal{SH}_{2n}$.

Note that if $H \in \mathcal{H}_{2n}$, then $P^{-1}HP \in \mathcal{H}_{2n}$ and if $S \in \mathcal{SH}_{2n}$, then $P^{-1}SP \in \mathcal{SH}_{2n}$, where P is an arbitrary symplectic matrix. Thus symplectic similarities preserve Hamiltonian and skew-Hamiltonian structure. Also, symmetric and skew-symmetric structures are preserved by orthogonal similarity transformations. Therefore structure-preserving algorithms for symmetric or skew-symmetric Hamiltonian or skew-Hamiltonian eigenproblems have to use real symplectic orthogonal transformations, that is, matrices $U \in \mathbb{R}^{2n\times 2n}$ satisfying $U^TJU = J, \quad U^TU = I$. As in [10], we denote by $SpO(2n)$ the group of real symplectic orthogonal matrices. Any $U \in SpO(2n)$ can be written as $U = \left[\begin{smallmatrix} U_1 & -U_2 \\ U_2 & U_1 \end{smallmatrix}\right]$, where $U_1^TU_1 + U_2^TU_2 = I$ and $U_1^TU_2 = U_2^TU_1$.

In Tables 2.1 and 2.2, we summarize the structure of Hamiltonian and skew-Hamiltonian matrices that are either symmetric or skew-symmetric, their eigenvalue

TABLE 2.1
Properties of structured Hamiltonian matrices $H \in \mathcal{H}_{2n}$.

$JH = (JH)^T$	Structure	Eigenvalues	Canonical form
Symmetric $H = H^T$	$\begin{bmatrix} E & F \\ F & -E \end{bmatrix}$, $E = E^T$ $F = F^T$	real, pairs $\lambda, -\lambda$	$\begin{bmatrix} D & 0 \\ 0 & -D \end{bmatrix}$
Skew-symmetric $H = -H^T$	$\begin{bmatrix} E & F \\ -F & E \end{bmatrix}$, $E = -E^T$ $F = F^T$	pure imaginary, pairs $\lambda, \bar{\lambda}$	$\begin{bmatrix} 0 & -D \\ D & 0 \end{bmatrix}$

TABLE 2.2
Properties of structured skew-Hamiltonian matrices $S \in \mathcal{SH}_{2n}$.

$JS = -(JS)^T$	Structure	Eigenvalues	Canonical form
Symmetric $S = S^T$	$\begin{bmatrix} E & F \\ -F & E \end{bmatrix}$, $E = E^T$ $F = -F^T$	real, double	$\begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}$
Skew-symmetric $S = -S^T$	$\begin{bmatrix} E & F \\ F & -E \end{bmatrix}$, $E = -E^T$ $F = -F^T$	pure imaginary, double, pairs $\lambda, \bar{\lambda}$	$\begin{bmatrix} B & 0 \\ 0 & -B \end{bmatrix}$

properties, and their symplectic orthogonal canonical form. We use $D \in \mathbb{R}^{n\times n}$ to denote a diagonal matrix and $B \in \mathbb{R}^{n\times n}$ to denote a block-diagonal matrix that is the direct sum of 1×1 zero blocks and 2×2 blocks of the form $\left[\begin{smallmatrix} 0 & b \\ -b & 0 \end{smallmatrix}\right]$. These canonical forms are consequences of results in [19].

Next, we show that the eigenvectors of skew-symmetric Hamiltonian matrices can be chosen to have structure. This property is important when defining and deriving structured backward errors.

LEMMA 2.1. *The eigenvectors of a skew-symmetric Hamiltonian matrix H can be chosen to have the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$ with $z \in \mathbb{C}^n$.*

Proof. Let $\left[\begin{smallmatrix} 0 & -D \\ D & 0 \end{smallmatrix}\right] = U^T H U$ be the canonical form of H with $U = \left[\begin{smallmatrix} U_1 & -U_2 \\ U_2 & U_1 \end{smallmatrix}\right]$ symplectic orthogonal. The matrix $X = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} I & I \\ -iI & iI \end{smallmatrix}\right]$ is unitary and diagonalizes the canonical form of H:

$$X^* \begin{bmatrix} 0 & -D \\ D & 0 \end{bmatrix} X = \begin{bmatrix} iD & 0 \\ 0 & -iD \end{bmatrix}.$$

Hence

$$UX = \frac{1}{\sqrt{2}} \begin{bmatrix} U_1 + iU_2 & U_1 - iU_2 \\ U_2 - iU_1 & U_2 + iU_1 \end{bmatrix}$$

is an eigenvector basis for H and this shows that the eigenvectors can be taken to have the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$ with $z \in \mathbb{C}^n$. ∎

Note that an eigenvector of a skew-symmetric Hamiltonian matrix does not necessarily have the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$. For instance, consider $H = \left[\begin{smallmatrix} 0 & -D \\ D & 0 \end{smallmatrix}\right]$ with $D = \operatorname{diag}(-d, d)$.

TABLE 3.1
t: Number of parameters defining H.

	Hamiltonian		Skew-Hamiltonian	
	$H = H^T$	$H = -H^T$	$H = H^T$	$H = -H^T$
t	$n^2 + n$	n^2	n^2	$n^2 - n$

Then $x^T = [i, 1, 1, i]$ is an eigenvector of H, corresponding to the eigenvalue $-id$, that is not of the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$.

3. Structured backward error. We begin by developing structured backward errors that can be used to test the strong stability of algorithms for our classes of Hamiltonian eigenproblems.

3.1. Definition. For notational convenience, the symbol H denotes from now on both Hamiltonian and skew-Hamiltonian matrices. Let $(\widetilde{x}, \widetilde{\lambda})$ be an approximate eigenpair for the structured Hamiltonian eigenvalue problem $Hx = \lambda x$, where $H \in \mathbb{R}^{2n\times 2n}$. A natural definition of the normwise backward error of an approximate eigenpair is

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \min\Big\{\epsilon : (H + \Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x},\ \|\Delta H\| \le \epsilon\|H\|\Big\},$$

where we measure the perturbation in a relative sense and $\|\cdot\|$ denotes any vector norm and the corresponding subordinate matrix norm. Deif [8] derived the explicit expression for the 2-norm

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \frac{\|r\|_2}{\|H\|_2\|\widetilde{x}\|_2},$$

where $r = \widetilde{\lambda}\widetilde{x} - H\widetilde{x}$ is the residual. This shows that the normwise relative backward error is a scaled residual. The componentwise backward error is a more stringent measure of the backward error in which the components of the perturbation ΔH are measured individually:

$$\omega(\widetilde{x}, \widetilde{\lambda}) = \min\Big\{\epsilon : (H + \Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x},\ |\Delta H| \le \epsilon|H|\Big\}.$$

Here inequalities between matrices hold componentwise. Geurts [12] showed that

$$\omega(\widetilde{x}, \widetilde{\lambda}) = \max_{1\le i\le 2n} \frac{|r|_i}{(|H||\widetilde{x}|)_i}.$$

The componentwise backward error provides a more meaningful measure of the stability than the normwise version when the elements in H vary widely in magnitude. However, this measure is not entirely appropriate for our problems as it does not respect any structure (other than sparsity) in H. Bunch [2] and Van Dooren [25] have also discussed other situations when it is desirable to preserve structure in definitions of backward errors.

The four classes of structured Hamiltonian matrices we are dealing with are defined by $t \le n^2 + n$ real parameters that make up E and F (see Table 3.1). We write this dependence as $H = H[p]$ with $p \in \mathbb{R}^t$. Higham and Higham [13], [14] extend the notion of componentwise backward error to allow dependence of the perturbations on

a set of parameters and they define structured componentwise backward errors. Following their idea and notation we define the structured relative normwise backward error by

$$\mu(\widetilde{x},\widetilde{\lambda}) = \min\Big\{\epsilon : (H+\Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x},\ H+\Delta H = H[p+\Delta p], \quad \|\Delta H\|_F \le \epsilon\|H\|_F\Big\}, \tag{3.1}$$

where $H+\Delta H = H[p+\Delta p]$ implies that ΔH has the same structure as H. The structured relative componentwise backward error $\nu(\widetilde{x},\widetilde{\lambda})$ is defined as in (3.1) but with the constraint $\|\Delta H\|_F \le \epsilon\|H\|_F$ replaced by $|\Delta H| \le \epsilon|H|$.

In our case, the dependence of the data on the t parameters is linear. We naturally require $(\widetilde{x},\widetilde{\lambda})$ to have any properties forced upon the exact eigenpairs, otherwise the backward error will be infinite. In the next subsections, we give algorithms for computing these backward errors. We start by describing a general approach that was used in [13] in the context of structured linear systems and extend it to the case where the approximate solution lies in the complex plane.

3.2. A general approach for the computation of $\mu(\widetilde{x},\widetilde{\lambda})$. Let $\widetilde{x} = \widetilde{u} + i\widetilde{v}$ and $\widetilde{\lambda} = \widetilde{\mu} + i\widetilde{\nu}$. By equating real and imaginary parts, the constraint $(H+\Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$ in (3.1) becomes

$$\begin{bmatrix} \Delta H & 0 \\ 0 & \Delta H \end{bmatrix}\begin{bmatrix} \widetilde{u} \\ \widetilde{v} \end{bmatrix} = \begin{bmatrix} (\widetilde{\mu}I - H)\widetilde{u} - \widetilde{\nu}\widetilde{v} \\ \widetilde{\nu}\widetilde{u} + (\widetilde{\mu}I - H)\widetilde{v} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \tag{3.2}$$

or equivalently $\Delta H\,[\,\widetilde{u} \quad \widetilde{v}\,] = [\,s_1 \quad s_2\,]$. Applying the vec operator (which stacks the columns of a matrix into one long vector), we obtain

$$\Big([\,\widetilde{u} \quad \widetilde{v}\,]^T \otimes I_{2n}\Big)\,\mathrm{vec}(\Delta H) = s, \quad s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \tag{3.3}$$

where $\otimes$ denotes the Kronecker product. We refer to Lancaster and Tismenetsky [18, Chap. 12] for properties of the vec operator and the Kronecker product. By linearity we have

$$\mathrm{vec}(\Delta H) = B\Delta p \tag{3.4}$$

for $B \in \mathbb{R}^{4n^2\times t}$ of full rank and where Δp is the t-vector of parameters defining ΔH. There exists a diagonal matrix D_1 depending on the structure of H (symmetric/skew-symmetric Hamiltonian/skew-Hamiltonian) such that

$$\|\Delta H\|_F = \|D_1\Delta p\|_2. \tag{3.5}$$

Let $y = D_1\Delta p$ and $Y = ([\,\widetilde{u} \quad \widetilde{v}\,]^T \otimes I_{2n}) \in \mathbb{R}^{4n\times 4n^2}$. Using (3.4) we can rewrite (3.3) as $YBD_1^{-1}y = s$ with $YBD^{-1} \in \mathbb{R}^{4n\times t}$. Then, using (3.5),

$$\mu(\widetilde{x},\widetilde{\lambda}) = \min_{y\,\in\,\mathbb{R}^t}\Big\{\|y\|_2/\|H\|_F \ :\ YBD_1^{-1}y = s\Big\}. \tag{3.6}$$

This shows that the structured backward error is given in terms of the minimal 2-norm

solution to an underdetermined system. If the underdetermined system is consistent, then the minimal 2-norm solution is given in terms of the pseudo-inverse by $y = (YBD_1^{-1})^+ s$. In this case

$$\mu(\widetilde{x},\widetilde{\lambda}) = \|(YBD_1^{-1})^+ s\|_2/\|H\|_F. \tag{3.7}$$

When H is a symmetric structured Hamiltonian, we can assume that $\widetilde{\lambda}$ and $\widetilde{x}$ are real. Therefore $\widetilde{\nu} = 0$ and $\widetilde{v} = 0$ and from (3.2) we have $[\, s_1 \quad s_2\,] = (\widetilde{\mu} I - H)\,[\,\widetilde{u} \quad \widetilde{v}\,]$. Applying the vec operation gives

$$s = \left([\,\widetilde{u} \quad \widetilde{v}\,]^T \otimes I_{2n}\right) \mathrm{vec}(\widetilde{\mu} I - H) = Y\,\mathrm{vec}(\widetilde{\mu} I - H).$$

As $\widetilde{\mu} I - H$ is also a symmetric structured Hamiltonian, we have by linearity that $\mathrm{vec}(\widetilde{\mu} I - H) = B p_{\widetilde{\mu}}$, where $p_{\widetilde{\mu}}$ is the t-vector of parameters defining $\widetilde{\mu} I - H$. Then $s = YBD_1^{-1} D_1 p_{\widetilde{\mu}}$ lies in the range of YBD_1^{-1}. Therefore, the underdetermined system in (3.6) is consistent for symmetric Hamiltonians and for symmetric skew-Hamiltonians. For a skew-symmetric Hamiltonian, we can again prove consistency for pure imaginary approximate eigenvalues and approximate eigenvectors of the form in Lemma 2.1. We have not been able to prove that the underdetermined system is consistent for the skew-symmetric skew-Hamiltonian case.

As the dependence on the parameters is linear, in the definition of the structured relative componentwise backward error $\nu(\widetilde{x},\widetilde{\lambda})$, we have the equivalence

$$|\Delta H| \le \epsilon |H| \quad \Longleftrightarrow \quad |\Delta p| \le \epsilon |p|.$$

Let $D_2 = \mathrm{diag}(p_i)$ and $\Delta p = D_2 q$. Then the smallest ϵ satisfying $|\Delta p| \le \epsilon |p|$ is $\epsilon = \|q\|_\infty$. The minimal ∞-norm solution of $YBD_2\, q = s$ can be approximated by minimizing in the 2-norm. We have

$$\nu(\widetilde{x},\widetilde{\lambda}) \le \|(YBD_2)^+ s\|_2 \le \sqrt{t+n}\; \nu(\widetilde{x},\widetilde{\lambda}).$$

By looking at each problem individually, it is possible to reduce the size of the underdetermined system. Nevertheless, solution of the system by standard techniques still takes $O(n^3)$ operations. In the next section, we show that by using a symplectic quasi-QR factorization of the approximate eigenvector and residual (or some appropriate parts) we can derive expressions for $\mu(\widetilde{x},\widetilde{\lambda})$ that are cheaper to compute for all the structured Hamiltonians of interest except for skew-symmetric skew-Hamiltonians. First, we define a symplectic quasi-QR factorization.

3.3. Symplectic quasi-QR factorization. We define the symplectic quasi-QR factorization of an $2n \times m$ matrix A by

$$A = QT, \quad T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, \tag{3.8}$$

where Q is real symplectic orthogonal, $T_1 \in \mathbb{R}^{n\times m}$ is upper trapezoidal, and $T_2 \in \mathbb{R}^{n\times m}$ is strictly upper trapezoidal. Such a symplectic quasi-QR factorization has also been discussed by Bunse-Gerstner [3, Cor. 4.5(ii)].Before giving an algorithm to

compute this symplectic quasi-QR factorization, we need to describe two types of elementary orthogonal symplectic matrices that can be used to zero selected components of a vector.

A *symplectic Householder* matrix $H \in \mathbb{R}^{2n\times 2n}$ is a direct sum of $n\times n$ Householder matrices:

$$H(k,v) = \begin{bmatrix} P(k,v) & 0 \\ 0 & P(k,v) \end{bmatrix},$$

where

$$P(k,v) = \begin{cases} \operatorname{diag}\left(I_{k-1}, I_{n-k+1} - \dfrac{2}{v^T v} vv^T\right) & \text{if} \quad v \neq 0, \\ I_n & \text{otherwise,} \end{cases}$$

and v is determined such that for a given $x \in \mathbb{R}^n$, $P(k,v)x = y$ with $y(k+1\colon n) = 0$.

A *symplectic Givens rotation* $G(k,\theta) \in \mathbb{R}^{2n\times 2n}$ is a Givens rotation where the rotation is performed in the plane $(k, k+n)$, $1 \le k \le n$, that is, $G(k,\theta)$ has the form

$$G(k,\theta) = \begin{bmatrix} C & S \\ -S & C \end{bmatrix}, \quad \text{where} \quad \begin{array}{l} C = \operatorname{diag}(I_{k-1}, \cos\theta, I_{n-k}), \\ S = \operatorname{diag}(0_{k-1}, \sin\theta, 0_{n-k}), \end{array} \tag{3.9}$$

where θ is chosen such that for a given $x \in \mathbb{R}^{2n}$, $G(k,\theta)x = y$ with $y_{n+k} = 0$.

We use a combination of these orthogonal transformations to compute our symplectic quasi-QR factorization: symplectic Householder matrices are used to zero large portions of a vector and symplectic Givens are used to zero single entries.

ALGORITHM 3.1 (symplectic quasi-QR factorization). *Given a matrix* $A = \left[\begin{smallmatrix} A_1 \\ A_2 \end{smallmatrix}\right]$ *with* $A_1, A_2 \in \mathbb{R}^{n\times m}$, *this algorithm computes the symplectic quasi-QR factorization* (3.8).

$Q = I_{2n}$
For $k = 1\colon \min(n-1, m)$
  Determine $H_k^1 = H(k,v)$ with $x = A_2 e_k$; $\quad A \leftarrow H_k^1 A$
  Determine $G_k = G(k,\theta)$ with $x = Ae_k$; $\quad A \leftarrow G_k A$
  Determine $H_k^2 = H(k,v)$ with $x = A_1 e_k$; $\quad A \leftarrow H_k^2 A$
  $Q \leftarrow Q H_k^1 G_k^T H_k^2$
End
If $m \ge n$
  Determine $G_n = G(n,\theta)$ with $x = Ae_n$; $\quad A \leftarrow G_n A$, $Q \leftarrow Q G_n^T$
End

We illustrate the procedure for a generic 6×4 matrix:

$$\left[\begin{array}{cccc}\times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\\ \hline \times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\end{array}\right] \overset{H_1^1}{\longrightarrow} \left[\begin{array}{cccc}\times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\\ \hline \times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\end{array}\right] \overset{G_1}{\longrightarrow} \left[\begin{array}{cccc}\times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\\ \hline 0&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\end{array}\right] \overset{H_1^2}{\longrightarrow}$$

$$\left[\begin{array}{cccc}\times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\\ \hline 0&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\end{array}\right] \overset{H_2^1}{\longrightarrow} \left[\begin{array}{cccc}\times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\\ \hline 0&\times&\times&\times\\ 0&\times&\times&\times\\ 0&0&\times&\times\end{array}\right] \overset{G_2}{\longrightarrow} \left[\begin{array}{cccc}\times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\\ \hline 0&\times&\times&\times\\ 0&0&\times&\times\\ 0&0&\times&\times\end{array}\right] \overset{H_2^2}{\longrightarrow}$$

$$\left[\begin{array}{cccc}\times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&0&\times&\times\\ \hline 0&\times&\times&\times\\ 0&0&\times&\times\\ 0&0&\times&\times\end{array}\right] \overset{G_3}{\longrightarrow} \left[\begin{array}{cccc}\times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&0&\times&\times\\ \hline 0&\times&\times&\times\\ 0&0&\times&\times\\ 0&0&0&\times\end{array}\right].$$

3.4. Symmetric Hamiltonian eigenproblems. Let $r = (\widetilde{\lambda} I - H)\widetilde{x} = \Delta H \widetilde{x}$ be the residual vector and $QR = [\widetilde{x}\ r]$ be the symplectic quasi-QR factorization (3.8) with Q symplectic orthogonal and

$$R = \begin{bmatrix} e_{11} & e_{12} \\ 0 & e_{22} \\ \vdots & 0 \\ & \vdots \\ & e_{n+1,2} \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2n\times 2}.$$

We have $Q^T \Delta H Q Q^T \widetilde{x} = Q^T r$, which is equivalent to

$$\Delta\widetilde{H} \begin{bmatrix} e_{11} \\ 0 \\ \vdots \\ \\ \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} e_{12} \\ e_{22} \\ 0 \\ \vdots \\ e_{n+1,2} \\ \vdots \\ 0 \end{bmatrix}, \tag{3.10}$$

where $\Delta\widetilde{H} = Q^T \Delta H Q$ is still a symmetric Hamiltonian matrix. Equation (3.10) defines the first column of $\Delta\widetilde{H}$. As $|e_{11}| = \|\widetilde{x}\| \neq 0$, we have $\Delta\tilde{h}_{11} = e_{12}/e_{11}$, $\Delta\tilde{h}_{21} = e_{22}/e_{11}$, $\Delta\tilde{h}_{n+1,1} = e_{n+1,2}/e_{11}$, and $\Delta\tilde{h}_{k,1} = 0$ for $k \neq 1, 2, n+1$. Let $\Delta\widetilde{E} = (\Delta\widetilde{E})^T$

and $\Delta\widetilde{F} = (\Delta\widetilde{F})^T$ be such that

$$\Delta\widetilde{E} = \frac{1}{e_{11}}\begin{bmatrix} e_{12} & e_{22} & 0 & \cdots & 0 \\ e_{22} & \times & \cdots & \cdots & \times \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & \times & \cdots & \cdots & \times \end{bmatrix}, \quad \Delta\widetilde{F} = \frac{1}{e_{11}}\begin{bmatrix} e_{n+1,2} & 0 & 0 & \cdots & 0 \\ 0 & \times & \cdots & \cdots & \times \\ 0 & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & \times & \cdots & \cdots & \times \end{bmatrix},$$

where the $\times$'s are arbitrary real coefficients. Then, any symmetric Hamiltonian of the form

$$\Delta H = Q\begin{bmatrix} \Delta\widetilde{E} & \Delta\widetilde{F} \\ \Delta\widetilde{F} & -\Delta\widetilde{E} \end{bmatrix} Q^T$$

satisfies $(H + \Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$. The Frobenius norm of ΔH is minimized by setting the $\times$'s to zero in the definition of $\Delta\widetilde{E}$ and $\Delta\widetilde{F}$. We obtain the following lemma.

LEMMA 3.2. *The backward error of an approximate eigenpair of a symmetric Hamiltonian eigenproblem is given by*

$$\mu(\widetilde{x}, \widetilde{\lambda}) = \frac{2}{|e_{11}|}\sqrt{\frac{e_{12}^2}{2} + e_{22}^2 + \frac{e_{n+1,2}^2}{2}}\Bigg/ \|H\|_F,$$

where $R = (e_{ij}) = Q^T[\widetilde{x}\ r]$ is the quasi-triangular factor in the symplectic quasi-QR factorization of $[\widetilde{x}\ r]$ with $r = (\widetilde{\lambda}I - H)\widetilde{x}$. We also have

$$\mu(\widetilde{x}, \widetilde{\lambda}) = \frac{\sqrt{2\|Q^T r\|^2 + 2(e_2^T Q^T r)^2}}{\|Q^T\widetilde{x}\|}\Bigg/ \|H\|_F,$$

where e_2 is the second unit vector.

3.5. Skew-symmetric Hamiltonian eigenproblems. For skew-symmetric Hamiltonian eigenproblems the technique developed in section 3.4 needs to be modified as in this case $r, \widetilde{x}$ are complex vectors and we want to define a real skew-symmetric Hamiltonian perturbation

$$\Delta H = \begin{bmatrix} \Delta E & \Delta F \\ -\Delta F & \Delta E \end{bmatrix}, \quad \Delta E = -\Delta E^T,\ \Delta F = \Delta F^T$$

so that $(H + \Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$.

In the definition of the structured backward error (3.1), we now assume that $\widetilde{\lambda}$ is pure imaginary and that $\widetilde{x}$ has the form $[\,\widetilde{z}^T \quad \pm i\widetilde{z}^T\,]^T$ (see Lemma 2.1). Taking the plus sign in $\widetilde{x}$, the equation $(H + \Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$ can be written as

$$\Delta E\widetilde{z} + i\Delta F\widetilde{z} = (\widetilde{\lambda}I - E)\widetilde{z} - iF\widetilde{z}, \tag{3.11}$$

$$-\Delta F\widetilde{z} + i\Delta E\widetilde{z} = F\widetilde{z} - i(\widetilde{\lambda}I - E)\widetilde{z}. \tag{3.12}$$

Multiplying (3.12) by $-i$ gives (3.11). Hence, we carry out the analysis with (3.11) only. Setting $\widetilde{\lambda} = i\widetilde{\mu},\ \widetilde{\mu} \in \mathbb{R}$, $\widetilde{z} = \widetilde{u} + i\widetilde{v}$ in (3.11) and equating real and imaginary parts yields

$$\Delta E\widetilde{u} - \Delta F\widetilde{v} = -\widetilde{\mu}\widetilde{v} + F\widetilde{v} - E\widetilde{u},$$
$$\Delta E\widetilde{v} + \Delta F\widetilde{u} = \widetilde{\mu}\widetilde{u} - E\widetilde{v} - F\widetilde{u},$$

which is equivalent to $\Delta H w = s$ with $w = [\,\widetilde{u}^T \quad -\widetilde{v}^T\,]^T$ and $s = (\widetilde{\mu}J - H)w$. Using $x^T E x = 0$ and $F^T = F$, we show that w and s are orthogonal:

$$\begin{aligned} w^T s &= [\,\widetilde{u}^T \quad -\widetilde{v}^T\,]\,(\widetilde{\mu}J - H)\begin{bmatrix} \widetilde{u} \\ -\widetilde{v} \end{bmatrix} \\ &= -\widetilde{u}^T E\widetilde{u} - \widetilde{\mu}\widetilde{u}^T\widetilde{v} + \widetilde{u}^T F\widetilde{v} + \widetilde{\mu}\widetilde{v}^T\widetilde{u} - \widetilde{v}^T E\widetilde{v} - \widetilde{v}^T F\widetilde{u} = 0. \end{aligned}$$

For the other choice of sign with $\widetilde{x} = [\,\widetilde{z}^T \quad -i\widetilde{z}^T\,]^T$, the equation $(H+\Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$ is equivalent to $\Delta H w = s$ with $w = [\,\widetilde{u}^T \quad \widetilde{v}^T\,]^T$ and $s = -(\widetilde{\mu}J + H)w$. Here again, we can show that $w^T s = 0$.

We can now carry on the analysis as in section 3.4. Let $QR = [w\ s]$ be the symplectic quasi-QR factorization of $[w\ s]$. As $w^T s = 0$, we have that $e_{12} = 0$. We obtain ΔH by solving the underdetermined system

$$\Delta\widetilde{H}\begin{bmatrix} e_{11} \\ 0 \\ \vdots \\ \\ \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ e_{22} \\ 0 \\ \vdots \\ e_{n+1,2} \\ \vdots \\ 0 \end{bmatrix}, \quad \Delta H = Q\Delta\widetilde{H}Q^T. \tag{3.13}$$

LEMMA 3.3. *The backward error of an approximate eigenpair $(\widetilde{x}, \widetilde{\lambda})$ of a skew-symmetric Hamiltonian eigenproblem with $\widetilde{\lambda}$ pure imaginary and $\widetilde{x}$ of the form $\widetilde{x} = [\,\widetilde{z}^T \quad \pm i\widetilde{z}^T\,]^T$ is given by*

$$\mu(\widetilde{x}, \widetilde{\lambda}) = \left(\frac{2}{|e_{11}|}\sqrt{e_{22}^2 + \frac{e_{n+1,2}^2}{2}}\right) \Big/ \|H\|_F,$$

where $R = Q^T[w\ s]$ is the quasi-triangular factor in the symplectic quasi-QR factorization of $[w\ s]$ with $\widetilde{z} = \widetilde{u} + i\widetilde{v}$ and

$$w = \begin{cases} [\,\widetilde{u}^T \quad -\widetilde{v}^T\,]^T & \text{if } \widetilde{x} = [\,\widetilde{z}^T \quad i\widetilde{z}^T\,]^T, \\ [\,\widetilde{u}^T \quad \widetilde{v}^T\,]^T & \text{otherwise,} \end{cases} \quad s = \begin{cases} (\widetilde{\mu}J - H)w & \text{if } \widetilde{x} = [\,\widetilde{z}^T \quad i\widetilde{z}^T\,]^T, \\ -(\widetilde{\mu}J + H)w & \text{otherwise.} \end{cases}$$

We also have

$$\mu(\widetilde{x}, \widetilde{\lambda}) = \left(\frac{\sqrt{2\|Q^T s\|_2^2 + 2|e_2^T Q^T s|^2}}{\|Q^T\widetilde{x}\|_2}\right) \Big/ \|H\|_F,$$

where e_2 is the second unit vector.

3.6. Symmetric skew-Hamiltonian eigenproblems. The analysis for symmetric skew-Hamiltonian eigenproblems is similar to that in section 3.4. The only difference comes from noting that

$$\begin{aligned} (J\widetilde{x})^T r &= [\,\widetilde{x}_2^T \quad -\widetilde{x}_1^T\,]\begin{bmatrix} \widetilde{\lambda}I - E & -F \\ F & \widetilde{\lambda}I - E \end{bmatrix}\begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{bmatrix} \\ &= \widetilde{x}_2^T(\widetilde{\lambda}I - E)\widetilde{x}_1 - \widetilde{x}_2^T F\widetilde{x}_2 - \widetilde{x}_1^T F\widetilde{x}_1 - \widetilde{x}_1^T(\widetilde{\lambda}I - E)\widetilde{x}_2 = 0 \end{aligned}$$

using $v^TFv = 0$ and $\widetilde{x}_1^T(\widetilde{\lambda}I-E)\widetilde{x}_2 = \widetilde{x}_2^T(\widetilde{\lambda}I-E)\widetilde{x}_1$. Instead of computing a symplectic quasi-QR factorization of $[\widetilde{x}\ r]$, we compute a symplectic quasi-QR factorization of $[J\widetilde{x}\ r]$ in order to introduce one more zero in the triangular factor R. We summarize the result in the next lemma.

LEMMA 3.4. *The backward error of an approximate eigenpair of a symmetric skew-Hamiltonian eigenproblem is given by*

$$\mu(\widetilde{x},\widetilde{\lambda}) = \frac{2}{|e_{11}|}\sqrt{e_{22}^2 + \frac{e_{n+1,2}^2}{2}}\Bigg/\|H\|_F, \tag{3.14}$$

where $R = Q^T[J\widetilde{x}\ r]$ is the quasi-triangular factor in the symplectic quasi-QR factorization of $[J\widetilde{x}\ r]$ with $r = (\widetilde{\lambda}I - H)\widetilde{x}$. We also have

$$\mu(\widetilde{x},\widetilde{\lambda}) = \left(\frac{\sqrt{2\|Q^Tr\|_2^2 + 2|e_2^TQ^Tr|^2}}{\|Q^TJ\widetilde{x}\|_2}\right)\Bigg/\|H\|_F.$$

3.7. Comments. Lemmas 3.2–3.4 provide an explicit formula for the backward error that can be computed in $O(n^2)$ operations.

For skew-symmetric skew-Hamiltonian matrices H, the eigenvectors are complex with no particular structure. The constraint $(H+\Delta H)\widetilde{x} = \widetilde{\lambda}\widetilde{x}$ in (3.1) can be written in the form $\Delta H[\Re(\widetilde{x}),\Im(\widetilde{x})] = [\Re(r),\Im(r)]$, where $r = (H - \widetilde{\lambda}I)\widetilde{x}$ is the residual. We were unable to explicitly construct matrices ΔH satisfying this constraint via a symplectic QR factorization of $[\Re(\widetilde{x}),\Im(\widetilde{x}),\Re(r),\Im(r)]$. Thus, in this case, we have to use the approach described in section 3.2 to compute $\mu(\widetilde{x},\widetilde{\lambda})$, which has the drawback that it requires $O(n^3)$ operations.

4. Algorithms for Hamiltonian eigenproblems. A simple but inefficient approach to solve structured Hamiltonian eigenproblems is to use the (symmetric or unsymmetric as appropriate) QR algorithm on the $2n \times 2n$ structured Hamiltonian matrix. This approach is computationally expensive and uses $4n^2$ storage locations. Moreover, the QR algorithm does not use symplectic orthogonal transformations and is therefore not structure-preserving.

Benner, Merhmann, and Xu's method [1] for computing the eigenvalues and invariant subspaces of a real Hamiltonian matrix uses the relationship between the eigenvalues and invariant subspaces of H and an extended $4n \times 4n$ Hamiltonian matrix. Their algorithm is structure-preserving for the extended Hamiltonian matrix but is not structure-preserving for H. Therefore, it is not strongly backward stable in the sense of this paper.

4.1. QR-like algorithms. Bunse-Gerstner, Byers, and Mehrmann [5] provide a chart of numerical methods for structured eigenvalue problems, most of them based on QR-like algorithms. In this section, we describe their recommended algorithms for our structured Hamiltonian eigenproblems. In the limited case where $\operatorname{rank}(F) = 1$, Byer's Hamiltonian QR algorithm [6] based on symplectic orthogonal transformations yields a strongly backward stable algorithm.

For symmetric Hamiltonian eigenproblems, the quaternion QR algorithm [4] is suggested. The quaternion QR algorithm is an extension of the Francis QR algorithm for complex or real matrices to quaternion matrices. This algorithm uses exclusively quaternion unitary similarity transformations so that it is backward stable. Compared with the standard QR algorithm for symmetric matrices, this algorithm cuts the

storage and work requirements approximately in half. However, its implementation requires quaternion arithmetic and it is not clear whether it is strongly backward stable.

A skew-symmetric Hamiltonian H is first reduced via symplectic orthogonal transformations to block antidiagonal form $\left[\begin{smallmatrix} 0 & T \\ -T & 0 \end{smallmatrix}\right]$, where the blocks are symmetric tridiagonal. The complete solution is obtained via the symmetric QR algorithm applied to T. The whole algorithm is strongly backward stable as it uses only real symplectic orthogonal transformations that are known to be backward stable.

For symmetric skew-Hamiltonian problems, the use of the "X-trick" is suggested:

$$X^T H X = \begin{bmatrix} E - iF & 0 \\ 0 & E + iF \end{bmatrix} \quad \text{with} \quad X = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ -iI & iI \end{bmatrix}. \tag{4.1}$$

The eigenvalues of H are computed from the eigenvalue of the Hermitian matrices $E - iF$ or $E + iF$, using the Hermitian QR algorithm for instance. One drawback of this approach is that it uses complex arithmetic and does not provide a real symplectic orthogonal eigenvector basis. Hence the algorithm does not preserve the "realness" of the original matrix.

Finally, for the skew-symmetric skew-Hamiltonian case, H is reduced to block-diagonal form via a finite sequence of symplectic orthogonal transformations. The blocks are themselves tridiagonal and skew-symmetric. Then Paardekooper's Jacobi algorithm [22] or the algorithm in [11] for skew-symmetric tridiagonal matrices can be used to obtain the complete solution. The whole algorithm is strongly backward stable.

4.2. Jacobi-like algorithms. Byers [7] adapted the nonsymmetric Jacobi algorithm [24] to the special structure of Hamiltonian matrices. The Hamiltonian Jacobi algorithm based on symplectic Givens rotations and symplectic double Jacobi rotations of the form $J \otimes I_{2n}$, where J is a 2×2 Jacobi rotation, preserves the Hamiltonian structure. This Jacobi algorithm, when it converges, builds a Hamiltonian Schur decomposition [7, Thm. 1]. For symmetric H, this Jacobi algorithm converges to the canonical form $\left[\begin{smallmatrix} D & 0 \\ 0 & -D \end{smallmatrix}\right]$ and is strongly backward stable. For skew-symmetric Hamiltonian H, this Jacobi algorithm does not converge as the symplectic orthogonal canonical form for H is not Hamiltonian triangular.

Recently, Faßbender, Mackey, and Mackey [10] developed Jacobi algorithms for structured Hamiltonian eigenproblems that preserve the structure and produce a complete basis of symplectic orthogonal eigenvectors for the two symmetric cases and a symplectic orthogonal basis for all the real invariant subspaces for the two skew-symmetric cases. These Jacobi algorithms are based on the direct solution of 4×4, and in one case 8×8, subproblems using appropriate transformations. The algorithms work entirely in real arithmetic. Note that "realness" of the initial matrix can be viewed as additional structure that these Jacobi algorithms preserve. We give a unified description of these Jacobi-like algorithms for the four classes of structured Hamiltonian eigenproblems under consideration.

Let $H \in \mathbb{R}^{2n \times 2n}$ be a structured Hamiltonian matrix (see Table 2.1 and 2.2). These Jacobi methods attempt to reduce the quantity (off-diagonal norm)

$$\operatorname{off}(H) = \sqrt{\sum_{i=1}^{2n} \sum_{j \in \mathcal{S}} |h_{ij}|^2},$$

where $\mathcal{S}$ is a set of indices depending on the structure of the problem using a sequence of symplectic orthogonal transformations $H \leftarrow SHS^T$ with $S \in \mathbb{R}^{2n\times 2n}$. The aim is that H converges to its canonical form. In the following, we note $A_{i,j,i+n,j+n}$ the restriction to the $(i, j, i + n, j + n)$ plane of A.

ALGORITHM 4.1. *Given a structured Hamiltonian matrix $H \in \mathbb{R}^{2n\times 2n}$ and a tolerance* $\mathrm{tol} > 0$*, this algorithm overwrites H with its approximate canonical form PHP^T, where P is symplectic orthogonal and* $\mathrm{off}(PHP^T) \leq \mathrm{tol}\,\|H\|_F$.

$P = I_{2n}$
$\epsilon = \mathrm{tol}\,\|H\|_F$
while $\mathrm{off}(H) > \epsilon$
  Choose (i, j)
  Compute a symplectic orthogonal S
    such that $(SHS^T)_{i,j,i+n,j+n}$ is in canonical form.
  $H \leftarrow SHS^T$ preserving structure
  $P \leftarrow SP$ preserving structure
end

Note that the pair (i, j) uniquely determines a 4×4 principal submatrix

$$H_{i,j,n+i,n+j} = \begin{bmatrix} h_{ii} & h_{ij} & h_{i,i+n} & h_{i,j+n} \\ h_{ji} & h_{jj} & h_{j,i+n} & h_{j,j+n} \\ h_{i+n,i} & h_{i+n,j} & h_{i+n,i+n} & h_{i+n,j+n} \\ h_{j+n,i} & h_{j+n,j} & h_{j+n,i+n} & h_{j+n,j+n} \end{bmatrix} \tag{4.2}$$

that also inherits the Hamiltonian or skew-Hamiltonian structure together with the symmetry or skew-symmetry property. There are many ways of choosing the indices (i, j) but this choice does not affect the rest of the analysis. We refer to $n(n - 1)/2$ updates as a sweep. Each sweep must be complete, that is, every part of the matrix must be reached. We see immediately that any complete sweep of the $(1, 1)$ block of H consisting of 2×2 principal submatrices generates a corresponding complete sweep of H.

For each 4×4 target submatrix, a symplectic orthogonal matrix that directly computes the corresponding canonical form is constructed and embedded into the $2n \times 2n$ identity matrix in the same way that the 4×4 target has been extracted.

For skew-symmetric skew-Hamiltonians, the 4×4 based Jacobi algorithm does not converge. The aim of these Jacobi algorithms is to move the weight to the diagonal of either the diagonal blocks or off-diagonal blocks. That cannot be done for a skew-symmetric skew-Hamiltonian because these diagonals are zero. There is no safe place where the norm of the target submatrix can be kept. However, if an 8×8 skew-symmetric skew-Hamiltonian problem is solved instead, the 2×2 diagonal blocks of H become a safe place for the norm of target submatrices and the resulting 8×8 based Jacobi algorithm is expected to converge. The complete sweep is defined by partitioning E in 2×2 blocks, leaving 2×1 and 1×2 blocks along the rightmost and lower edges when n is odd. Hence, in this case we must also be able to directly solve 6×6 subproblems.

Immediately, we see that the difficult part in deriving these algorithms is to define the appropriate symplectic orthogonal transformation S that computes the canonical form of the restriction to the $(i, j, i + n, j + n)$ plane of H. Faßbender, Mackey, and Mackey [10] show that by using a quaternion representation of the 4×4 symplectic orthogonal group, as well as 4×4 Hamiltonian and skew-Hamiltonian matrices in the tensor square of the quaternion algebra, we can define and construct 4×4 symplectic

orthogonal matrices R that do the job. These transformations are based on rotations of $\mathbb{P} \cong \mathbb{R}^3$, the subspace of pure quaternions.

We need to give all the required transformations in a form suitable for rounding error analysis and also to facilitate the description of the structure preserving Jacobi algorithms. We start by defining two types of quaternion rotations. This enables us to encode the formulas in [10] into one. Let $e_s \neq e_1$ be a standard basis vector of $\mathbb{R}^4$ and $p \in \mathbb{R}^4$ such that $p \neq 0$, $e_1^T p = 0$ (p is a pure quaternion), and $p/\|p\|_2 \neq e_s$. Let

$$x^T = \|p\|_2 e_1^T + e_s^T \begin{bmatrix} 0 & -p_2 & -p_3 & -p_4 \\ p_2 & 0 & p_4 & -p_3 \\ p_3 & -p_4 & 0 & p_2 \\ p_4 & p_3 & -p_2 & 0 \end{bmatrix}. \tag{4.3}$$

We define the left quaternion rotation by

$$Q_L(p,s) = \frac{1}{\|x\|_2} \begin{bmatrix} x_1 & -x_2 & -x_3 & -x_4 \\ x_2 & x_1 & -x_4 & x_3 \\ x_3 & x_4 & x_1 & -x_2 \\ x_4 & -x_3 & x_2 & x_1 \end{bmatrix}. \tag{4.4}$$

Q_L is symplectic orthogonal and not difficult to compute. We have $x_1 = \|p\|_2 + p_s$ and the other components of x are just permutations of the coordinates of p.

We define the right quaternion rotation by

$$Q_R(p,s) = \frac{1}{\|x\|_2} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2 & x_1 & -x_4 & x_3 \\ -x_3 & x_4 & x_1 & -x_2 \\ -x_4 & -x_3 & x_2 & x_1 \end{bmatrix}. \tag{4.5}$$

The matrix Q_R is orthogonal. It is symplectic when $s \neq 3$ and $x_2 = x_4 = 0$.

Let $p = [\, p_1 \quad p_2 \quad p_3 \quad p_4 \,]^T \in \mathbb{R}^4$ be nonzero. Following [10], we define the 4×4 symplectic orthogonal Givens rotation associated with p by

$$G(p) = \frac{1}{\|p\|_2} \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ -p_2 & p_1 & p_4 & -p_3 \\ -p_3 & -p_4 & p_1 & p_2 \\ -p_4 & p_3 & -p_2 & p_1 \end{bmatrix}. \tag{4.6}$$

We now have all the tools needed to define the symplectic orthogonal transformations that directly compute the canonical form for each of the 4×4 structured Hamiltonian eigenproblems of interest. We refer to [10] for more details about how these transformations have been derived.

4.2.1. Symmetric Hamiltonian. Let $H \in \mathbb{R}^{4\times 4}$ be a symmetric Hamiltonian matrix. The canonical form of H is obtained in two steps: first H is reduced to 2×2 block diagonal form and then the complete diagonalization is obtained by using a double Jacobi rotation.

For the first step we consider the singular value decomposition of the 3×3 matrix

$$A = \begin{bmatrix} \frac{1}{2}(h_{11} + h_{22}) & 0 & \frac{1}{2}(h_{13} + h_{24}) \\ h_{14} & 0 & -h_{12} \\ \frac{1}{2}(h_{24} - h_{13}) & 0 & \frac{1}{2}(h_{11} - h_{22}) \end{bmatrix} = U \Sigma V^T.$$

Let u_1 and v_1 be the left and right singular vectors corresponding to the largest singular value σ_1 and let $u = [\, {0 \atop u_1} \,]$, $v = [\, {0 \atop v_1} \,]$. We have $A^T u_1 = \sigma_1 v_1$ and $e_2^T A^T u_1 = 0$

so that $e_2^T v_1 = 0$. Hence, for $s = 2$ and $p = v$, the vector x in (4.3) is such that $x_2 = x_4 = 0$, which implies that the right quaternion rotation $Q_R(v, 2)$ is symplectic orthogonal. As shown in [10], the product $Q = Q_L(u, 2)Q_R(v, 2)$ block diagonalizes H, that is, $QHQ^T = \operatorname{diag}(\widetilde{E}, -\widetilde{E})$. Complete diagonalization is obtained by using a double Jacobi rotation $J(\theta) \otimes I_2$, where θ is chosen such that $J(\theta) = \left[\begin{smallmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{smallmatrix}\right]$ diagonalizes $\widetilde{E}$.

In summary, the symplectic orthogonal transformation S used in Algorithm 4.1 is equal to the identity matrix except in the $(i, j, n+i, n+j)$ plane, where the $(i, j, n+i, n+j)$-restriction matrix is given by

$$S_{i,j,n+i,n+j} = \big(J(\theta) \otimes I_2\big) Q_L(u, 2) Q_R(v, 2).$$

4.2.2. Skew-symmetric Hamiltonian. Let $H \in \mathbb{R}^{4\times 4}$ be a skew-symmetric Hamiltonian matrix and let $p \in \mathbb{R}^4$ be defined from the elements of H by

$$p = [\, 0, \quad h_{21}, \quad \tfrac{1}{2}(h_{31} - h_{42}), \quad h_{41} \,]^T.$$

It is easy to verify that for $S = Q_L(p, 3)$,

$$SHS^T = \begin{bmatrix} 0 & 0 & -\|p\|_2 + b & 0 \\ 0 & 0 & 0 & \|p\|_2 + b \\ \|p\|_2 - b & 0 & 0 & 0 \\ 0 & -\|p\|_2 - b & 0 & 0 \end{bmatrix},$$

where $b = \frac{1}{2}(h_{13} + h_{24})$.

4.2.3. Symmetric skew-Hamiltonian. Let $H \in \mathbb{R}^{4\times 4}$ be a symmetric skew-Hamiltonian matrix and let $p \in \mathbb{R}^4$ be defined from the elements of H by

$$p = [\, 0, \quad -h_{14}, \quad \tfrac{1}{2}(h_{11} - h_{22}), \quad h_{12} \,]^T.$$

Then, $S = Q_L(p, 3)$ diagonalizes H and

$$SHS^T = \begin{bmatrix} b + \|p\|_2 & 0 & 0 & 0 \\ 0 & b - \|p\|_2 & 0 & 0 \\ 0 & 0 & b + \|p\|_2 & 0 \\ 0 & 0 & 0 & b - \|p\|_2 \end{bmatrix},$$

where $b = \frac{1}{2}(h_{11} + h_{22})$.

4.2.4. Skew-symmetric skew-Hamiltonian. For the convergence of the Jacobi algorithm to be possible we need to solve an 8×8 subproblem. The matrix $H \in \mathbb{R}^{8\times 8}$ is block diagonalized with three 4×4 symplectic Givens rotations of the form (4.6) and one symplectic Givens rotation of the form (3.9). Let G be the product of these rotations. We have

$$GHG^T = \begin{bmatrix} \widetilde{E} & 0 \\ 0 & -\widetilde{E} \end{bmatrix}, \tag{4.7}$$

where $\widetilde{E} \in \mathbb{R}^{4\times 4}$ is tridiagonal and skew-symmetric. The complete 2×2 block-diagonalization is obtained by directly transforming $\widetilde{E}$ into its real Schur form as follows. In [20], Mackey showed that the transformation $Q = Q_L(q_1, 2)Q_R(q_2, 2)$ with

$$q_1 = [\, 0, -\tfrac{1}{2}(\tilde{e}_{12} + \tilde{e}_{34}), 0, -\tfrac{1}{2}\tilde{e}_{23} \,]^T, \qquad q_2 = [\, 0, \tfrac{1}{2}(\tilde{e}_{12} - \tilde{e}_{34}), 0, -\tfrac{1}{2}\tilde{e}_{23} \,]^T$$

directly computes the real Schur form of $\widetilde{E}$, that is,

$$Q\widetilde{E}Q^T = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \text{ with } \quad B_1 = \begin{bmatrix} 0 & s_2 - s_1 \\ s_1 - s_2 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & -s_1 - s_2 \\ s_1 + s_2 & 0 \end{bmatrix},$$

where $s_1 = \|q_1\|_2$ and $s_2 = \|q_2\|_2$. Then $S = (Q \otimes I_2)G$ is the symplectic orthogonal transformation that computes the real Schur form of the 8×8 skew-symmetric skew-Hamiltonian H:

$$SHS^T = \begin{bmatrix} B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & -B_1 & 0 \\ 0 & 0 & 0 & -B_2 \end{bmatrix}.$$

When n is odd, we have to solve a 6×6 subproblem for each complete sweep of the Jacobi algorithm. As for the 8×8 case, the 6×6 skew-symmetric skew-Hamiltonian H is first reduced to the form (4.7), where $\widetilde{E} \in \mathbb{R}^{3\times3}$ is tridiagonal and skew-symmetric. This is done by using just one 4×4 symplectic Givens rotation followed by one 2×2 symplectic Givens rotation. Let

$$\widetilde{E}_{aug} = \begin{bmatrix} 0 & 0 \\ 0 & \widetilde{E} \end{bmatrix} \in \mathbb{R}^{4\times4}$$

and $q = [\,0 \quad -\frac{1}{2}\tilde{e}_{23}, 0, -\frac{1}{2}\tilde{e}_{12}\,]^T$. Then $Q = Q_L(q,4)Q_R(q,4)$ computes directly the real Schur form of $\widetilde{E}_{aug}$. Moreover, we have $e_1^T Q e_1 = 1$, so that $Q = \mathrm{diag}(1, \widetilde{Q})$ and $\widetilde{Q}\widetilde{E}\widetilde{Q}^T = B$, where

$$B = \begin{bmatrix} 0 & -b & 0 \\ b & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{with} \quad b = 2\|q\|_2.$$

5. Error analysis of the Jacobi algorithms. In floating point arithmetic, Algorithm 4.1 computes an approximate canonical form $\widehat{T}$ such that

$$\widehat{T} =: P(H + \Delta H)P^T,$$

where P is symplectic orthogonal, and an approximate basis of symplectic orthogonal eigenvectors $\widehat{P}$. We want to derive bounds for $\|\Delta H\|, \|\widehat{P}\widehat{P}^T - I\|$, and $\|\widehat{P}J\widehat{P}^T - J\|$.

5.1. Preliminaries. We use the standard model for floating point arithmetic [16]

$$fl(x \,\mathrm{op}\, y) = (x \,\mathrm{op}\, y)(1 + \delta), \quad |\delta| \le u, \quad \mathrm{op} = +, -, *, /, \tag{5.1}$$

where u is the unit roundoff. We assume that (5.1) holds also for the square roots operation. To keep track of the higher terms in u we make use of the following result [16, Lem. 3.1].

LEMMA 5.1. *If $|\delta_i| \le u$ and $\rho_i = \pm1$ for $i = 1\colon n$, and $nu < 1$, then*

$$\prod_{i=1}^{n}(1 + \delta_i)^{\rho_i} = 1 + \theta_n, \qquad \textit{where} \qquad |\theta_n| \le \frac{nu}{1 - nu} =: \gamma_n.$$

We define

$$\widetilde{\gamma}_k = \frac{pku}{1 - pku},$$

where p denotes a small integer constant whose value is unimportant. In the following, computed quantities will be denoted by hats.

First, we consider the construction of a 4×4 Givens rotation and left and right quaternion rotations.

LEMMA 5.2. *Let a 4×4 Givens rotation $G = G(p)$ be constructed according to (4.6) with $p \in \mathbb{R}^4$. Then the computed $\widehat{G}$ satisfies $|\widehat{G} - G| \leq \gamma_5 |G|$.*

Proof. This result is a straightforward extension of Lemma 18.6 in [16] concerning 2×2 Givens rotations. ∎

The rounding error properties of right and left quaternion rotations require more attention. When $p_s < 0$, the computation of $\|p\|_2 + p_s$ and therefore the computation of $Q_L(p, s)$ or $Q_R(p, s)$ is affected by cancellation. This problem can be overcome by using another formula as shown in the next lemma.

LEMMA 5.3. *Let 4×4 left and right quaternion rotations $Q_L = Q_L(p, s)$ and $Q_R = Q_R(p, s)$ be constructed according to*

$$Q_L(p,s) = \frac{1}{\sqrt{2\|p\|_2 \alpha}} \begin{bmatrix} \alpha & -x_2 & -x_3 & -x_4 \\ x_2 & \alpha & -x_4 & x_3 \\ x_3 & x_4 & \alpha & -x_2 \\ x_4 & -x_3 & x_2 & \alpha \end{bmatrix}, \tag{5.2}$$

$$Q_R(p,s) = \frac{1}{\sqrt{2\|p\|_2 \alpha}} \begin{bmatrix} \alpha & x_2 & x_3 & x_4 \\ -x_2 & \alpha & -x_4 & x_3 \\ -x_3 & x_4 & \alpha & -x_2 \\ -x_4 & -x_3 & x_2 & \alpha \end{bmatrix}, \tag{5.3}$$

where

$$[\, x_2 \quad x_3 \quad x_4 \,] = \begin{cases} [\, 0 \quad p_4 \quad -p_3 \,] & \text{if } s = 2, \\ [\, -p_4 \quad 0 \quad p_2 \,] & \text{if } s = 3, \\ [\, p_3 \quad -p_2 \quad 0 \,] & \text{if } s = 4 \end{cases}$$

and

$$\alpha = \begin{cases} \|p\|_2 + p_s \ \text{if } p_s \geq 0, \\ \displaystyle\sum_{i=2, i \neq s}^{4} \left(p_i^2\right) / (\|p\|_2 - p_s) \quad \text{otherwise,} \end{cases} \tag{5.4}$$

with $p \in \mathbb{R}^4$ given. Then the computed $\widehat{Q_L}$ and $\widehat{Q_R}$ satisfy

$$|\widehat{Q_L} - Q_L| \leq \widetilde{\gamma}_1 |Q_L|, \quad |\widehat{Q_R} - Q_R| \leq \widetilde{\gamma}_1 |Q_R|.$$

Proof. It is straightforward to verify that the expressions for $Q_L(p, s)$ and $Q_R(p, s)$ in (5.2) and (5.3) agree with the definitions in (4.4) and (4.5).

We have $fl(\|p\|_2) = \|p\|_2(1 + \theta_4)$ with $|\theta_4| \leq \gamma_4$. If $p_s \geq 0$,

$$fl(\alpha) = (\|p\|_2(1 + \theta_4) + p_s)(1 + \delta) = \|p\|_2(1 + \theta_5) + p_s(1 + \delta).$$

As $p_s \geq 0$, there exists θ_5' such that $fl(\alpha) = (\|p\|_2 + p_s)(1+\theta_5')$ with $|\theta_5'| \leq \gamma_5$. If $p_s < 0$, using the same argument we have

$$fl(\|p\|_2 - p_s) = (\|p\|_2 - p_s)(1+\theta_5), \quad |\theta_5| \leq \gamma_5.$$

We also have

$$fl\left(\sum_{i=2, i\neq s}^{4} p_i^2\right) = \left(\sum_{i=2, i\neq s}^{4} p_i^2\right)(1+\theta_3).$$

Using [16, Lem. 3.3] we have

$$fl(\alpha) = \frac{(\sum_{i=2,i\neq s}^{4} p_i^2)(1+\theta_3)}{(\|p\|_2 - p_s)(1+\theta_5)}(1+\delta) = \alpha(1+\theta_9), \quad |\theta_9| \leq \gamma_9,$$

and

$$fl(\sqrt{2\|p\|_2\alpha}) = \sqrt{2\|p\|_2\alpha}(1+\theta_{16}), \quad |\theta_{16}| \leq \gamma_{16}.$$

Hence, we certainly have

$$fl((Q_L)_{ij}) \leq (Q_L)_{ij}(1+\theta_{26}), \quad |\theta_{26}| \leq \gamma_{26} \leq \widetilde{\gamma}_1. \qquad \square$$

In the following we use the term *elementary symplectic orthogonal matrix* to describe any double Givens rotation, 4×4 Givens rotation, or left or right quaternion rotation that is embedded as a principal submatrix of the identity matrix $I \in \mathbb{R}^{2n\times 2n}$.

We have proved that any computed elementary symplectic orthogonal matrix $\widehat{P} = fl(P)$ used by the Jacobi algorithm satisfies a bound of the form

$$|\widehat{P} - P| \leq \widetilde{\gamma}_1 |P|. \tag{5.5}$$

LEMMA 5.4. *Let $x \in \mathbb{R}^{2n\times 2n}$ and consider the computation of $y = \widehat{P}x$, where $\widehat{P}$ is a computed elementary symplectic orthogonal matrix satisfying* (5.5). *The computed $\widehat{y}$ satisfies*

$$\widehat{y} = P(x + \Delta x), \quad \|\Delta x\|_2 \leq \widetilde{\gamma}_1 \|x\|_2,$$

where P is the exact elementary symplectic orthogonal matrix.

Proof. The vector $\widehat{y}$ differs from x only in elements $i, j, i+n$, and $j+n$. We have

$$\widehat{y}_i = e_i^T P x + \Delta y_i, \quad |\Delta y_i| \leq \widetilde{\gamma}_1 |e_i^T P||x|.$$

We obtain similar results for $\widehat{y}_j$, $\widehat{y}_{n+i}$, and $\widehat{y}_{n+j}$. Hence,

$$\widehat{y} = Px + \Delta y, \quad |\Delta y| \leq \widetilde{\gamma}_1 |P|\ |x|.$$

As $\| |P| \|_2 \leq 2$, we have $\|\Delta y\|_2 \leq 2\widetilde{\gamma}_1 \|x\|_2 = \widetilde{\gamma}_1' \|x\|_2$. Finally, we define $\Delta x = P^T \Delta y$ and note that $\|\Delta x\|_2 = \|\Delta y\|_2$. $\square$

Now, we consider the pre- and postmultiplication of a matrix H by an approximate elementary symplectic orthogonal matrix $\widehat{P}$.

LEMMA 5.5. *Let $H \in \mathbb{R}^{2n\times 2n}$ and $P \in \mathbb{R}^{2n\times 2n}$ be any elementary symplectic orthogonal matrix such that $fl(P)$ satisfies* (5.5). *Then,*

$$fl(PH) = P(H + \Delta H),\ \|\Delta H\|_F \leq \widetilde{\gamma}_1 \|H\|_F,$$
$$fl(PHP^T) = P(H + \Delta H)P^T,\ \|\Delta H\|_F \leq \widetilde{\gamma}_1 \|H\|_F.$$

Proof. Let h_i be the ith column of H. By Lemma 5.4 we have

$$fl(Ph_i) = P(h_i + \Delta h_i), \quad \|\Delta h_i\|_2 \le \widetilde{\gamma}_1 \|h_i\|_2.$$

The same result holds for h_j, h_{n+i}, and h_{n+j} and the other columns of H are unchanged. Hence, $fl(PH) = P(H + \Delta H)$, where $\|\Delta H\|_F \le \widetilde{\gamma}_1 \|H\|_F$. Similarly, $fl(\widehat{B}P^T) = (\widehat{B} + \Delta\widehat{B})P^T$ with $\|\Delta\widehat{B}\|_F \le \widetilde{\gamma}_1 \|\widehat{B}\|_F$. Then, with $\widehat{B} = fl(PH)$ we have

$$fl(PHP^T) = (PH + P\Delta H + \Delta\widehat{B})P^T = P(H + \Delta H + P^T\Delta\widehat{B})P^T,$$

with $\|\Delta\widehat{B}\|_F \le \widetilde{\gamma}_1(1 + \widetilde{\gamma}_1)\|H\|_F$ and therefore $\|\Delta H + P^T\Delta\widehat{B}\|_F \le \widetilde{\gamma}_1\|H\|_F$. ∎

As a consequence of Lemma 5.5, if H_{k+1} is the matrix obtained after one Jacobi update with S_k (which is the product up to six elementary symplectic orthogonal matrices), we have

$$\widehat{H}_{k+1} = S_k(\widehat{H}_k + \Delta\widehat{H}_k)S_k^T, \quad \|\Delta\widehat{H}_k\|_F \le \widetilde{\gamma}_1\|\widehat{H}_k\|_F, \tag{5.6}$$

where S_k is the exact transformation for $\widehat{H}_k$.

Up to now, we made no assumption on H. If H is a structured Hamiltonian matrix, the $(i, j, n+i, n+j)$-restriction of RHR^T is in canonical form. For instance, if H is a skew-symmetric Hamiltonian matrix, in a computer implementation the diagonal elements of H are not computed but are set to zero. Also, h_{ij}, $h_{i,j+n}$ and by skew-symmetry h_{ji}, $h_{j+n,i}$ are set to zero. But by forcing these elements to be zero, we are making the error smaller so the bounds still hold.

Because of the structure of the problem, both storage and the flop count can be reduced by a factor of four. Any structured Hamiltonian matrix needs less than n^2+n storage locations. If only the t parameters defining H are computed, the structure in the error is preserved and ΔH has the same structure as H. It is easy to see that the bounds in Lemma 5.6 are still valid with the property that ΔH has the same structure as H.

THEOREM 5.6. *Algorithm* 4.1 *for structured Hamiltonians H compute a canonical form $\widehat{T}$ such that*

$$\widehat{T} = P(H + \Delta H)P^T, \quad P^TP = I, \quad P^TJP = J,$$

where ΔH has the same structure as H and $\|\Delta H\|_F \le \widetilde{\gamma}_k\|H\|_F$, where k is the number of symplectic orthogonal transformations S_i applied for each Jacobi update.

The computed basis of symplectic orthogonal eigenvectors $\widehat{P} = fl(S_k \dots S_2S_1)$ satisfies

$$\|\widehat{P}^T\widehat{P} - I\|_F \le \widetilde{\gamma}_k \quad \textit{and} \quad \|\widehat{P}^TJ\widehat{P} - J\|_F \le \widetilde{\gamma}_k. \tag{5.7}$$

Proof. From (5.6), one Jacobi update of H satisfies

$$\widehat{H}_1 = fl(S_1HS_1^T) = S_1(H + \Delta H_1)S_1^T, \quad \|\Delta H_1\|_F \le \widetilde{\gamma}_1\|H\|_F.$$

For the second update we have

$$\begin{aligned}\widehat{H}_2 = fl(S_2\widehat{H}_1S_2^T) &= S_2(\widehat{H}_1 + \Delta\widehat{H}_1)S_2^T, \quad \|\Delta\widehat{H}_1\|_F \le \widetilde{\gamma}_1\|\widehat{H}_1\|_F \\ &= S_2S_1(H + \Delta H_1 + S_1^T\Delta\widehat{H}_1S_1)S_1^TS_2^T \\ &= S_2S_1(H + \Delta H_2)S_1^TS_2^T,\end{aligned}$$

where $\|\Delta H_2\|_F \le \|\Delta H_1\|_F + \|\Delta \widehat{H}_1\|_F \le \widetilde{\gamma}_1(1+(1+\widetilde{\gamma}_1))\|H\|_F \le \widetilde{\gamma}_2\|H\|_F$. Continuing in this fashion, we find that, after k updates,

$$\widehat{H}_k = S_k \dots S_1(H + \Delta H_k)S_1^T \dots S_k^T \quad \text{with} \quad \|\Delta H_k\|_F \le \widetilde{\gamma}_k\|H\|_F.$$

In a similar way, using the first part of Lemma 5.5 we have

$$\begin{aligned}
\widehat{P}_1 &= fl(S_1 I) = S_1 + \Delta P_1, \quad \|\Delta P_1\|_F \le \widetilde{\gamma}_1, \\
\widehat{P}_2 &= fl(S_2\widehat{P}_1) = S_2(S_1 + \Delta P_1) + \Delta \widehat{P}_2, \quad \|\Delta \widehat{P}_2\|_F \le \widetilde{\gamma}_1\|\widehat{P}_1\| \\
&= S_2 S_1 + \Delta P_2, \quad \|\Delta P_2\|_F \le \widetilde{\gamma}_1 + \widetilde{\gamma}_1(1+\widetilde{\gamma}_1) \le \widetilde{\gamma}_2.
\end{aligned}$$

After k updates, $\widehat{P}_k = fl(S_k\widehat{P}_{k-1}) = S_k \dots S_1 + \Delta P_k$, $\|\Delta P_k\|_F \le \widetilde{\gamma}_k$, and (5.7) follows readily. ∎

Theorem 5.6 shows that the computed eigenvalues are the exact eigenvalues of a nearby structured Hamiltonian matrix and that the computed basis of eigenvectors is orthogonal and symplectic up to machine precision. This proves the strong backward stability of the Jacobi algorithms.

6. Numerical experiments. To illustrate our results we present some numerical examples. All computations were carried out in MATLAB, which has unit roundoff $u = 2^{-53} \approx 2.2 \times 10^{-16}$.

For symmetric Hamiltonians, symmetric skew-Hamiltonians, and skew-symmetric Hamiltonians with approximate eigenvector $\widehat{x}$ of the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$, computing $\mu(\widehat{x}, \widehat{\lambda})$ in (3.1) involves a symplectic quasi-QR factorization of a $2n \times 2$ matrix, which can be done in order n^2 flops, a cost negligible compared with the $O(n^3)$ cost of the whole eigendecomposition.

For skew-symmetric Hamiltonians with approximate eigenvector $\widehat{x}$ not of the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$, and for skew-symmetric skew-Hamiltonians, the computation of $\mu(\widehat{x}, \widehat{\lambda})$ requires $O(n^3)$ flops as we have to find the minimal 2-norm solution of a large underdetermined system in (3.6). Thus, in this case, $\mu(\widehat{x}, \widehat{\lambda})$ is not a quantity we would compute routinely in the course of solving a problem.

Note that in our implementation of the Jacobi-like algorithm for skew-symmetric Hamiltonians we choose the approximate eigenvectors to be the columns of $P\left[\begin{smallmatrix} I & I \\ -iI & -iI \end{smallmatrix}\right]$, where P is the accumulation of the symplectic orthogonal transformations used by the algorithm to build the canonical form. In this case, the approximate eigenvectors $\widehat{x}$ are guaranteed to be of the form $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$.

To test the strong stability of numerical algorithms for solving structured Hamiltonian eigenproblems, we applied the direct search maximization routine `mdsmax` of the MATLAB Test Matrix Toolbox [15] to the function

$$f(E, F) = \max_{1 \le i \le 2n} \mu(\widehat{x}_i, \widehat{\lambda}_i),$$

where $(\widehat{\lambda}_i, \widehat{x}_i)$ are the computed eigenpairs. In this way we carried out a search for problems on which the algorithms performs unstably.

As expected from the theory, we could not generate examples for which the structured backward error for the Jacobi-like algorithms is large: $\mu(\widehat{x}, \widehat{\lambda}) < nu\|H\|_F$ in all our tests.

The symmetric QR algorithm does not use symplectic orthogonal transformations and is therefore not structure-preserving. To our surprise, we could not generate examples of symmetric Hamiltonian and symmetric skew-Hamiltonian matrices for which

TABLE 6.1
Backward error of the eigenpair for $\lambda = 0.741i$ of the 4×4 skew-symmetric Hamiltonian defined by (6.1).

	$\eta_{\max}(\widehat{x},\widehat{\lambda})$	$\omega_{\max}(\widehat{x},\widehat{\lambda})$	$\mu_{\max}(\widehat{x},\widehat{\lambda})$
QR algorithm	2×10^{-16}	4×10^{-16}	9×10^{-2}
Jacobi-like algorithm	5×10^{-17}	1×10^{-16}	9×10^{-17}

TABLE 6.2
Backward errors of the approximation of the eigenvalue 0 for a 30×30 random skew-symmetric skew-Hamiltonian matrix.

	$\lvert\widehat{\lambda}\rvert$	$\eta(\widehat{x},\widehat{\lambda})$	$\omega(\widehat{x},\widehat{\lambda})$	$\mu(\widehat{x},\widehat{\lambda})$
QR algorithm	3×10^{-11}	1×10^{-16}	6×10^{-16}	7×10^{-7}
Jacobi-like algorithm	0	6×10^{-17}	4×10^{-16}	1×10^{-15}

any of the eigenpairs computed by the symmetric QR algorithm has a large backward error. However, the QR algorithm does not compute a symplectic orthogonal basis of eigenvectors and also, it is easy to generate examples for which the $\pm\lambda$ structure for symmetric Hamiltonians and eigenvalue multiplicity 2 structure for symmetric skew-Hamiltonians is not preserved. If we generalize the definition of the structured backward error of a single eigenpair to a set of k eigenpairs, the symmetric QR algorithm is likely to produce sets of eigenpairs with an infinite structured backward error. The QR-like algorithm for symmetric skew-Hamiltonians is likely to provide eigenvectors that are complex instead of real, yielding an infinite structured backward error in (3.14).

The good backward stability of individual eigenpairs computed by the QR algorithm does not hold for the skew-symmetric Hamiltonian case. For instance, we considered the skew-symmetric Hamiltonian eigenproblem

$$H = \begin{bmatrix} E & F \\ -F & E \end{bmatrix}, \text{ with } E = \begin{bmatrix} 0 & 0.75 \\ -0.75 & 0 \end{bmatrix}, \; F = \begin{bmatrix} -0.1875 & 0.0938 \\ 0.0938 & -0.125 \end{bmatrix}, \tag{6.1}$$

whose eigenvalues are distinct: $\Lambda(H) = \{0.803i, -0.803i, 0.741i, -0.741i\}$. In Table 6.1, we give the normwise, componentwise, and structured normwise backward error of the eigenpair for $\lambda = 0.741i$ computed by the unsymmetric QR algorithm and the skew-symmetric Jacobi algorithm. The QR algorithm does not use symplectic orthogonal transformations and the computed eigenvectors do not have the structure $\left[\begin{smallmatrix} z \\ \pm iz \end{smallmatrix}\right]$. Therefore, for the computation of $\mu_{\max}(\widehat{x},\widehat{\lambda})$, we use the general formula (3.7).

In the skew-symmetric skew-Hamiltonian case, when n is odd, 0 is an eigenvalue of multiplicity two and is not always well approximated with the unsymmetric QR algorithm. We generated a random 15×15 E and F. We give in Table 6.2 the backward errors associated with the approximation of the eigenvalue 0 for both the QR algorithm and Jacobi algorithm.

7. Conclusion. The first contribution of this work is to extend existing definitions of backward errors in a way appropriate to structured Hamiltonian eigenproblems. We provided computable formulae that are inexpensive to evaluate except for skew-symmetric skew-Hamiltonians. Our numerical experiments showed that for symmetric structured Hamiltonian eigenproblems, the symmetric QR algorithm computes eigenpairs with a small structured backward error but the algebraic properties of the problem are not preserved.

Our second contribution is a detailed rounding error analysis of the new Jacobi algorithms of Faßbender, Mackey, and Mackey [10] for structured Hamiltonian eigenproblems. These algorithms are structure-preserving, inherently parallelizable, and hence attractive for solving large-scale eigenvalue problems. We proved their strong stability when the left and right quaternion rotations are implemented using our formulae (5.2), (5.3). Jacobi algorithms are easy to implement and offer a good alternative to QR algorithms, namely, the unsymmetric QR algorithm, which we showed to be not strongly backward stable for skew-symmetric Hamiltonian and skew-Hamiltonian eigenproblems, and the algorithm for symmetric skew-Hamiltonians based on applying the QR algorithm to (4.1), which does not respect the "realness" of the problem.

Acknowledgments. I thank Nil Mackey for pointing out the open question concerning the strong stability of the Jacobi algorithms for structured Hamiltonian eigenproblems and for her suggestion in fixing the cancellation problem when computing the quaternion rotations. I also thank Steve Mackey for his helpful comments on an earlier manuscript.

REFERENCES

[1] P. Benner, V. Mehrmann, and H. Xu, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, J. Comput. Appl. Math., 86 (1997), pp. 17–43.
[2] J. R. Bunch, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88/89 (1987), pp. 49–66.
[3] A. Bunse-Gerstner, *Matrix factorizations for symplectic QR-like methods*, Linear Algebra Appl., 83 (1986), pp. 49–77.
[4] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *A quaternion QR algorithm*, Numer. Math., 55 (1989), pp. 83–95.
[5] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *A chart of numerical methods for structured eigenvalue problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 419–453.
[6] R. Byers, *A Hamiltonian QR algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[7] R. Byers, *A Hamiltonian-Jacobi algorithm*, IEEE Trans. Automat. Control, 35 (1990), pp. 566–570.
[8] A. Deif, *A relative backward perturbation theorem for the eigenvalue problem*, Numer. Math., 56 (1989), pp. 625–626.
[9] J. Dongarra, J. R. Gabriel, D. D. Kölling, and J. H. Wilkinson, *The eigenvalue problem for Hermitian matrices with time reversal symmetry*, Linear Algebra Appl., 60 (1984), pp. 27–42.
[10] H. Faßbender, D. S. Mackey, and N. Mackey, *Hamilton and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems*, Linear Algebra Appl., to appear.
[11] K. V. Fernando, *Accurately counting singular values of bidiagonal matrices and eigenvalues of skew-symmetric tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 373–399.
[12] A. J. Geurts, *A contribution to the theory of condition*, Numer. Math., 39 (1982), pp. 85–96.
[13] 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.
[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 276, Manchester Centre for Computational Mathematics, Manchester, England, 1995.
[16] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[17] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, New York, 1995.
[18] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, London, 1985.
[19] W.-W. Lin, V. Mehrmann, and H. Xu, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 302/303 (1999), pp. 469–533.
[20] N. Mackey, *Hamilton and Jacobi meet again: Quaternions and the eigenvalue problem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 421–435.

[21] J. Olsen, H. Jørgen, A. Jensen, and P. Jørgensen, *Solution of the large matrix equations which occur in response theory*, J. Comput. Phys., 74 (1988), pp. 265–282.
[22] M. H. C. Paardekooper, *An eigenvalue algorithm for skew-symmetric matrices*, Numer. Math., 17 (1971), pp. 189–202.
[23] N. Rösch, *Time-reversal symmetry, Kramers' degeneracy and the algebraic eigenvalue problem*, Chemical Physics, 80 (1983), pp. 1–5.
[24] G. W. Stewart, *A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 853–864.
[25] P. M. Van Dooren, *Structured linear algebra problems in digital signal processing*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. H. Golub and P. M. Van Dooren, eds., vol. F70 of NATO ASI Series, Springer-Verlag, Berlin, 1991, pp. 361–384.
[26] C. F. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 126–137
© 2001 Society for Industrial and Applied Mathematics

THE SINGULARITY-INDUCED BIFURCATION AND ITS KRONECKER NORMAL FORM*

R. E. BEARDMORE†

Abstract. It is shown that the singularity-induced bifurcation theorem due to Venkatasubramanian, Schattler, and Zaborszky [*Proceedings of the IEEE*, 83 (1995), pp. 1530–1558] can be expressed as the perturbation of an infinite eigenvalue of a particular class of parameterized index-1 matrix pencil, denoted $(M, L(\lambda))$. It is shown that the matrix pencil at the singularity-induced bifurcation point, $(M, L(\lambda_0))$, has Kronecker index-2. It is also shown that a two-parameter unfolding of a singularity-induced bifurcation point results in a locus of index-0 pencils, denoted $(M(\epsilon), L(\lambda(\epsilon)))$, which has two purely imaginary eigenvalues near infinity.

Key words. unfoldings, bifurcation, matrix pencil

AMS subject classifications. 15A22, 47A55, 34A09

PII. S089547989936457X

1. Introduction.

Consider the parameterized semiexplicit, index-1 differential algebraic equation (DAE)

$$\dot{x} = f(x, y, \lambda), \tag{1.1a}$$

$$0 = g(x, y, \lambda), \tag{1.1b}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m$, and $\lambda \in \mathbb{R}$. As the parameter λ varies, an equilibrium of (1.1) may encounter the *singular set*

$$S = \{(x, y) \in \mathbb{R}^{n+m} : g(x, y, \lambda) = 0,\ \det(d_y g(x, y, \lambda)) = 0\}.$$

The singularity-induced bifurcation theorem (SIB), which was first proven in [16, 17], describes the behavior of an eigenvalue associated with the linearization of (1.1) at such an encounter.

For completeness, we state the SIB theorem from [17].

THEOREM 1 (SIB). *Define*

$$\Delta(x, y, \lambda) = \det(d_y g(x, y, \lambda)),$$

set $z = (x, y)$ *and let* $F(x, y, \lambda) = (f \times g)(x, y, \lambda)$. *Suppose that* z_0 *is an equilibrium of* (1.1) *and write* $w = (z_0, \lambda_0)$. *If*

1. $d_y g(w)$ *has a simple zero eigenvalue,*
2. *the matrix* $\begin{pmatrix} d_z F(w) & d_\lambda F(w) \\ d_z \Delta(w) & d_\lambda \Delta(w) \end{pmatrix}$ *is invertible,*
3. $tr(d_y f(w) \cdot \mathrm{adj}(d_y g(w)) \cdot d_x g(w)) \neq 0$, *and*
4. $d_z F(w)$ *is invertible,*

then there is a unique, parameterized equilibrium of (1.1), *defined in a neighborhood of* λ_0, *which is transversal to the singular set* S *at* λ_0. *One real eigenvalue of the*

*Received by the editors December 16, 1999; accepted for publication (in revised form) by V. Mehrmann October 11, 2000; published electronically May 3, 2001. This paper was written when the author was a member of the Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TR, UK.

http://www.siam.org/journals/simax/23-1/36457.html

†Department of Mathematics, Imperial College, South Kensington, University of London, London, SW7 2AZ, UK (r.beardmore@ic.ac.uk).

linearization of (1.1) *about this equilibrium locus moves from one open half of the complex plane to the other by diverging to infinity as* λ *passes* λ_0. *The remaining* $n-1$ *eigenvalues of the linearization lie in some bounded set which does not contain the origin.*

There are several ingredients in the method of proof of Theorem 1 used in [16] which are essentially nonlinear in nature. The approach used therein is to rescale time and subsequently rewrite (1.1) as a smooth, parameterized vector field. Venkatasubramanian, Schattler, and Zaborszky then apply the center manifold theorem to probe this vector field and the eigenvalues of its linearization in a neighborhood of the equilibrium which lies on S.

The described approach leads naturally to a description of the flow of the DAE (1.1) when the equilibrium lies on S, namely, when $\lambda = \lambda_0$. In [16] this description is called the local decomposition theorem (LDT). We shall show in a future paper [4] that the algebraic approach used in this paper, coupled to the implicit function theorem, can be used to derive an extension of the LDT. This is important due to the fact that the LDT is starting to find applications in wider mathematics [7, 14].

However, the SIB theorem concerns the perturbation of an infinite eigenvalue of a parameterized matrix pencil. This observation permitted a result equivalent to the SIB to be proven in [3]. This provided a much simpler method of proof and allowed hypothesis 4 of the SIB theorem to be removed, at the same time maintaining the asymptotic nature of the diverging eigenvalue.

We shall denote the diverging eigenvalue by $\alpha(\lambda)$ and in [16, 3] it was shown that there is some $\mu \neq 0$ such that

$$\alpha(\lambda) \sim \frac{\mu}{\lambda - \lambda_0} + O(1)$$

as $\lambda \to \lambda_0$. The point λ_0 is said to be a *singularity-induced bifurcation point.*

Let us define some notation. For any vector $v \in \mathbb{R}^p$ we shall write $\langle v \rangle = \{sv : s \in \mathbb{R}\}$. If $L : \mathbb{R}^p \to \mathbb{R}^p$ is a linear mapping, then $\sigma(L)$ denotes its spectrum, $\mathbf{N}(L)$ and $\mathbf{R}(L)$ denote its null-space and range, respectively. We shall use $\mathcal{N}(L)$ and $\mathcal{R}(L)$ to denote its generalized null-space and range, respectively. We shall use the symbol # to denote cardinality and $\mathcal{L}(\mathbb{R}^p)$ denotes the space of linear mappings over $\mathbb{R}^p$. We shall use Λ to denote some open interval containing λ_0.

Suppose that (R, S) is a matrix pencil over $\mathbb{R}^N$. It is said to be regular if there is some $s \in \mathbb{C}$ such that

$$\det(sR - S) \neq 0.$$

Moreover, we shall write

$$\sigma(R, S) = \{s \in \mathbb{C} : \det(sR - S) = 0\}.$$

The following is well known [9, 10, 6] and is the Kronecker normal form (KNF).

THEOREM 2 (KNF). *Suppose that* (R, S) *is a regular matrix pencil on* $\mathbb{R}^N$. *One can write* $\mathbb{R}^N = U \oplus V$ *and find nonsingular maps* $P \in \mathcal{L}(\mathbb{R}^N)$ *and* $Q \in \mathcal{L}(U \oplus V, \mathbb{R}^N)$ *such that*

$$PRQ = \begin{bmatrix} I_u & 0 \\ 0 & N \end{bmatrix} \text{ and } PSQ = \begin{bmatrix} C & 0 \\ 0 & I_v \end{bmatrix},$$

where there is an integer $k \geq 1$ *such that* $N^k = 0$ *and* $N^{k-1} \neq 0$.

Now $C, I_u \in \mathcal{L}(U)$, $N, I_v \in \mathcal{L}(V)$ *and* I_u *and* I_v *are identities on* U *and* V, *respectively. Moreover,* $\sigma(R,S) = \sigma(C)$ *and* $\#\sigma(R,S) = \dim U$. *Define the* index *of* (R,S), *written* $\mathrm{ind}(R,S)$, *to be the integer* k.

Suppose that (1.1) has a zero equilibrium for all $\lambda \in \Lambda$ and suppose that

$$L(\lambda) = \begin{bmatrix} A(\lambda) & B(\lambda) \\ C(\lambda) & D(\lambda) \end{bmatrix} \in \mathcal{L}(\mathbb{R}^{n+m})$$

is the linearization of $f \times g$ about that equilibrium. In addition, we suppose that $\lambda \mapsto L(\lambda)$ is a C^r mapping. If $D(\lambda) = d_y g(0,0,\lambda)$ is invertible, the implicit function theorem implies that the eigenvalues of

$$\mathcal{S}_\lambda \stackrel{\text{def}}{=} A(\lambda) - B(\lambda)D^{-1}(\lambda)C(\lambda) \tag{1.2}$$

determine the linear stability properties of the trivial equilibrium of (1.1).

Define the matrix

$$M(\epsilon) \stackrel{\text{def}}{=} \begin{bmatrix} I_n & 0 \\ 0 & \epsilon I_m \end{bmatrix} \in \mathcal{L}(\mathbb{R}^{n+m}),$$

where the n and m may be omitted for brevity. Write $M \stackrel{\text{def}}{=} M(0)$.

The paper is organized as follows. The purpose of section 2 is twofold. First we present a result which is equivalent to the SIB theorem, but with a method of proof that is different from both [16] and [3]. Moreover, the method of proof used in section 2 shows that the linearization

$$(M, L(\lambda))$$

of (1.1) has the following property. If $\lambda \neq \lambda_0$, then $(M, L(\lambda))$ has index-1. We compute the Kronecker normal form and show that $(M, L(\lambda_0))$ has index-2.

The following simple theorem is called the singular Hopf bifurcation (SHB).

THEOREM 3 (SHB). *Let* $a, b, c, d \in C^1(\mathbb{R})$ *and suppose that for some* $\lambda_0 \in \mathbb{R}$, $d(\lambda_0) = 0, d'(\lambda_0) \neq 0$, *and* $c(\lambda_0)b(\lambda_0) = -\omega_0^2 < 0$. *Then there is an interval* $I = (0, \epsilon_0) \subset \mathbb{R}$ *and* C^1 *functions* $\omega, \lambda : I \to \mathbb{R}$ *such that for all* $\epsilon \in I$

$$i\omega(\epsilon) \in \sigma \begin{pmatrix} a(\lambda(\epsilon)) & b(\lambda(\epsilon)) \\ \epsilon^{-1}c(\lambda(\epsilon)) & \epsilon^{-1}d(\lambda(\epsilon)) \end{pmatrix}.$$

Moreover, $\lambda(0_+) = \lambda_0$ *and* $\lim_{\epsilon \to 0_+} \epsilon^{1/2}\omega(\epsilon) = \omega_0$.

The SHB theorem plays a central role in [2, 15] and relates to the many discussions of duck solutions and ducks oscillations to be found, for instance, in [5, 18, 8, 1]. Motivated by Theorem 3, in section 3 we consider a singular unfolding of the linearization of (1.1), given by

$$M(\epsilon)\dot{z} = L(\lambda)z. \tag{1.3}$$

It is shown that the infinite eigenvalue from the SIB theorem perturbs to give a pair of purely imaginary eigenvalues of $(M(\epsilon), L(\lambda))$ near infinity.

2. KNF and the SIB. We now present several preliminary lemmas.

LEMMA 1. *Let* $\det : \mathcal{L}(\mathbb{R}^q) \to \mathbb{R}$ *be the determinant functional. It has derivative* $\det'$ *given by* $\det'(P)[H] = \mathrm{tr}(\mathrm{adj}P \cdot H)$, *where* $P, H \in \mathcal{L}(\mathbb{R}^q)$.

LEMMA 2. *If* $\det D(\lambda) \neq 0$, *then* $\dim \mathbf{N}(\mathcal{S}_\lambda) = \dim \mathbf{N}(L(\lambda))$ *and* $\det \mathcal{S}_\lambda \cdot \det D(\lambda) = \det L(\lambda)$.

LEMMA 3. *Let* $P \in \mathcal{L}(\mathbb{R}^q)$ *and suppose that there is a nonzero* $k \in \mathbb{R}^q$ *such that* $\mathbf{N}(P) = \langle k \rangle$, *then* $\mathbf{R}(\mathrm{adj}P) = \langle k \rangle$ *and* $\mathbf{N}(\mathrm{adj}P) = \mathbf{R}(P)$.

We shall often omit the reference to λ_0 for brevity and simply write

$$L(\lambda_0) = L = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathcal{L}(\mathbb{R}^{n+m}).$$

LEMMA 4. *Suppose* $L \in \mathcal{L}(\mathbb{R}^{n+m})$ *is invertible and has its inverse decomposed as*

$$L^{-1} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \in \mathcal{L}(\mathbb{R}^{n+m}).$$

Then the finite spectrum of the pencil (M, L) *satisfies* $\sigma(M, L) = [\sigma(A_1) \backslash \{0\}]^{-1}$.

Proof. Clearly

$$\begin{aligned} \det(\lambda M - L) = 0 &\Leftrightarrow \det(L^{-1}M - \lambda^{-1} I) = 0 \\ &\Leftrightarrow \det \begin{bmatrix} A_1 - \lambda^{-1} I_n & 0 \\ C_1 & -\lambda^{-1} I_m \end{bmatrix} = 0 \end{aligned}$$

and the result follows since $\det L \neq 0$. ∎

We shall also use $A_1(\lambda), B_1(\lambda)$, and so on, to denote the elements in the decomposition of the parameterized inverse $L(\lambda)^{-1}$.

LEMMA 5. *If* $\det D(\lambda) \neq 0$, *then* $\sigma(\mathcal{S}_\lambda) = \sigma(M, L(\lambda))$.

Lemma 4 shows that we can understand the infinite eigenvalue of (M, L) by studying the zero eigenvalues of the mapping A_1. Before studying the implications of Lemma 4 further, we present the generalized SIB theorem [3].

THEOREM 4. *Suppose that* $r \geq 3$ *and* $L \in C^r(\Lambda, L(\mathbb{R}^{n+m}))$. *Suppose further that* $\mathbf{N}(D(\lambda_0)) = \langle k \rangle$, *where* $k \neq 0$, $D'(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$, *and* $C(\lambda_0)B(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$. *Then there is a* $\delta > 0$ *such that for all* $\lambda \in N_\delta(\lambda_0) \backslash \{\lambda_0\}$, $\mathcal{S}_\lambda \in \mathcal{L}(\mathbb{R}^n)$ *exists. There is a* C^r *locus of algebraically simple, real eigenvalues of* $\mathcal{S}_\lambda$, *denoted* $\alpha(\lambda)$, *which has a simple pole at* λ_0

$$\alpha(\lambda) = \frac{\mu}{\lambda - \lambda_0} + \phi(\lambda - \lambda_0),$$

where $\mu \neq 0$ *and the function* ϕ *is* C^{r-2} *in* $N_\delta(\lambda_0)$.

This has one fewer conditions than Theorem 1. Namely, the invertibility of $L(\lambda_0) = d_z(f \times g)(0, 0, \lambda_0)$ is needed to prove Theorem 1, but not its generalization Theorem 4. However, Theorem 4 does not tell us how the remaining $n - 1$ elements of $\sigma(\mathcal{S}_\lambda)$ behave at λ_0, nor does it provide an expression for the residue μ. However, retaining this invertibility condition permits another proof of Theorem 1, based on Lemma 4, which yields a simple expression for μ.

THEOREM 5. *Suppose that* $r \geq 3$ *and* $L \in C^r(\Lambda, \mathcal{L}(\mathbb{R}^{n+m}))$. *Suppose further that* $\mathbf{N}(D(\lambda_0)) = \langle k \rangle$, *where* $k \neq 0$, $\mathbf{N}(D(\lambda_0)^T) = \langle u \rangle$, $D'(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$, *and* $C(\lambda_0)B(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$. *In addition, suppose that* $L(\lambda_0) \in GL(\mathbb{R}^{n+m})$.

Then there is a $\delta > 0$ *such that for all* $\lambda \in N_\delta(\lambda_0) \backslash \{\lambda_0\}$, $\mathcal{S}_\lambda \in \mathcal{L}(\mathbb{R}^n)$ *exists and is invertible. There is a locus of eigenvalues,* $\alpha(\lambda) \in \sigma(\mathcal{S}_\lambda)$, *and a* C^{r-2} *function* ϕ *such that* $\alpha(\lambda) = \mu/(\lambda - \lambda_0) + \phi(\lambda - \lambda_0)$, *where*

$$\mu = -\frac{u^T C(\lambda_0) B(\lambda_0) k}{u^T D'(\lambda_0) k}.$$

For all $|\lambda - \lambda_0| < \delta$ the remaining $n-1$ eigenvalue locii of $\mathcal{S}_\lambda$ lie in some compact annulus $\mathcal{A} \subset \mathbb{C}$ which does not contain the origin. Moreover, these eigenvalues can be uniquely extended to a continuous function on $N_\delta(\lambda_0)$.

In addition, in the notation of Lemma 4

$$A_1(\lambda_0) = \lim_{\lambda \to \lambda_0} \mathcal{S}_\lambda^{-1},$$

where $\mathcal{N}(A_1(\lambda_0)) = \langle B(\lambda_0)k \rangle$ and $\mathcal{R}(A_1(\lambda_0)) = \langle C^T(\lambda_0)u \rangle$.

Proof. Define the matrices $S(\lambda)$ and $T(\lambda) \in \mathcal{L}(\mathbb{R}^{n+m})$,

$$T(\lambda) \stackrel{\text{def}}{=} \begin{bmatrix} I & 0 \\ C(\lambda) & D(\lambda) \end{bmatrix} \text{ and } S(\lambda) \stackrel{\text{def}}{=} \begin{bmatrix} I & B(\lambda) \\ 0 & D(\lambda) \end{bmatrix},$$

which form the Schur decomposition

$$L(\lambda) = S(\lambda) \begin{bmatrix} \mathcal{S}_\lambda & 0 \\ 0 & D(\lambda)^{-1} \end{bmatrix} T(\lambda). \tag{2.1}$$

By Lemmas 1 and 3, $\frac{d}{d\lambda} \det D(\lambda)\big|_{\lambda=\lambda_0} \neq 0$ and it follows that there is some $\delta_1 > 0$ such that $\det D(\lambda)$ is nonzero on $N_{\delta_1}(\lambda_0) \backslash \{\lambda_0\}$. Therefore $\mathcal{S}_\lambda$ exists on this neighborhood and is a bijection using (2.1).

As $\det L(\lambda_0) \neq 0$ there is a $\delta_2 > 0$ such that $L(\lambda)^{-1}L(\lambda) \equiv L(\lambda)L(\lambda)^{-1} = I$ for all $\lambda \in N_{\delta_2}(\lambda_0)$. Let $\delta = \min(\delta_1, \delta_2)$.

Let us define the C^r map

$$\lambda \mapsto L(\lambda)^{-1} = \begin{bmatrix} A_1(\lambda) & B_1(\lambda) \\ C_1(\lambda) & D_1(\lambda) \end{bmatrix}.$$

One finds

$$A_1(\lambda)B(\lambda) + B_1(\lambda)D(\lambda) \equiv 0, \tag{2.2a}$$

$$C(\lambda)A_1(\lambda) + D(\lambda)C_1(\lambda) \equiv 0, \tag{2.2b}$$

$$C(\lambda)B_1(\lambda) + D(\lambda)D_1(\lambda) \equiv I_m, \tag{2.2c}$$

$$A(\lambda)A_1(\lambda) + B(\lambda)C_1(\lambda) \equiv I_n. \tag{2.2d}$$

By taking inverses, it follows from (2.1) that

$$\mathcal{S}_\lambda^{-1} \equiv A_1(\lambda)$$

when $\det D(\lambda) \neq 0$. We shall now prove that $A_1(\lambda_0)$ has an algebraically simple eigenvalue. This will provide a curve, $\beta(\lambda)$, of eigenvalues of $A_1(\lambda)$ such that $\alpha(\lambda) = 1/\beta(\lambda) \in \sigma(\mathcal{S}_\lambda)$ and the fact that $\beta(\lambda_0) = 0$ will complete the proof.

Equation (2.2a) implies

$$A_1(\lambda_0)B(\lambda_0)k = 0.$$

Suppose that there is some $v \neq 0$ such that $A_1(\lambda_0)v = 0$. Using (2.2d) evaluated at $\lambda = \lambda_0$, we find $B(\lambda_0)C_1(\lambda_0)v = v$. Using (2.2b) we see that $D(\lambda_0)C_1(\lambda_0)v = 0$, so $C_1(\lambda_0)v = \zeta k$ for some $\zeta \in \mathbb{R}$ by the simple null-space condition. Therefore $\zeta B(\lambda_0)k = v$ and it follows that

$$\mathbf{N}(A_1(\lambda_0)) = \langle B(\lambda_0)k \rangle .$$

Now suppose $B(\lambda_0)k \in \mathbf{R}(A_1(\lambda_0))$ so that $A_1(\lambda_0)w = B(\lambda_0)k$ for some nonzero w. This yields, from (2.2b), $C(\lambda_0)A_1(\lambda_0)w + D(\lambda_0)C_1(\lambda_0)w = 0$. It follows that $C(\lambda_0)B(\lambda_0)k = -D(\lambda_0)C_1(\lambda_0)w$ and therefore $u^TC(\lambda_0)B(\lambda_0)k = 0$, a contradiction. Hence 0 is an algebraically simple eigenvalue of $A_1(\lambda_0)$ with eigenvector $B(\lambda_0)k$, therefore 0 is an algebraically simple eigenvalue of $A_1(\lambda_0)^T$.

Since, from (2.2b), $0 = (u^TC(\lambda_0)A_1(\lambda_0))^T = A_1(\lambda_0)^TC(\lambda_0)^Tu$ we find

$$\mathcal{R}(A_1(\lambda_0)) = \mathbf{R}(A_1(\lambda_0)) = \mathbf{N}(A_1(\lambda_0)^T)^\perp = \left\langle C(\lambda_0)^Tu \right\rangle^\perp.$$

From spectral perturbation results in [11] one can find a C^r, algebraically simple, real eigenvalue locus $\beta(\lambda) \in \sigma(A_1(\lambda))$ such that $\beta(\lambda_0) = 0$. Differentiating the identities

$$\beta(\lambda)v(\lambda) \equiv A_1(\lambda)v(\lambda)$$

and also (2.2a) and then setting $\lambda = \lambda_0$ yields

$$\beta'(\lambda_0) = -\frac{u^TC(\lambda_0)B_1(\lambda_0)D'(\lambda_0)k}{u^TC(\lambda_0)B(\lambda_0)k}.$$

Thus $\beta'(\lambda_0) = 0$ if and only if

$$C(\lambda_0)B_1(\lambda_0)D'(\lambda_0)k \in \mathbf{R}(D(\lambda_0)).$$

Postmultiplying (2.2c) by $D'(\lambda_0)k$, one finds

$$C(\lambda_0)B_1(\lambda_0)D'(\lambda_0)k + D(\lambda_0)D_1(\lambda_0)D'(\lambda_0)k = D'(\lambda_0)k.$$

Therefore

$$u^TC(\lambda_0)B_1(\lambda_0)D'(\lambda_0)k = u^TD'(\lambda_0)k \neq 0$$

and consequently $\beta'(\lambda_0) \neq 0$. Define $\alpha(\lambda) = 1/\beta(\lambda)$.

Since $\lim_{\lambda\to\lambda_0}(\lambda - \lambda_0)\alpha(\lambda) = \mu$ and $\alpha(\lambda) = 1/\beta(\lambda) \in \sigma(A_1(\lambda)^{-1}) \equiv \sigma(S_\lambda)$, it is clear that $\mu = 1/\beta'(\lambda_0)$. Since $\mathcal{S}_\lambda = A_1(\lambda)^{-1}$ for $\lambda \neq \lambda_0$, and $A_1(\lambda_0)$ has an algebraically simple zero eigenvalue, it has $n-1$ nonzero eigenvalues whose reciprocals provide the elements of $\mathbb{C}$ required to remove the $n-1$ singularities in $\sigma(\mathcal{S}_\lambda)$. ∎

COROLLARY 1. *Under the conditions of Theorem 5, if $\lambda \in N_\delta(\lambda_0)\backslash\{\lambda_0\}$, then $\#\sigma(M, L(\lambda)) = n$, however, $\#\sigma(M, L(\lambda_0)) = n - 1$.*

Proof. Use Lemma 4 to count the number of nonzero eigenvalues of $A_1(\lambda)$ according to algebraic multiplicity. ∎

THEOREM 6. *If (M, L) is a regular pencil, then* $\mathrm{ind}(M, L) = 1$ *if and only if* $\det D \neq 0$.

Proof. If $\det D \neq 0$, then it is simple to show that[1] $\mathrm{ind}(M, L) = 1$.

Now suppose that (M, L) is regular and $\mathrm{ind}(M, L) = 1$. Hence there are matrices P_{ij} and Q_{ij}, for $i = 1, 2$ and $j = 1, 2$, which form nonsingular maps P and Q, such that for some n_1 and n_2 with $n_1 + n_2 = n + m$,

$$PMQ = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} I_{n_1} & 0 \\ 0 & 0_{n_2} \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{M}'. \tag{2.3}$$

Now $m = \dim \mathbf{N}(M) = \dim \mathbf{N}(PMQ) = \dim \mathbf{N}(\mathbf{M}') = n_2$ and it follows that $n = n_1$.

[1] See the proof of Theorem 7 for details.

Using (2.3) we find that $P_{11}Q_{11} = I_n$, $Q_{12} = 0$, and $P_{21} = 0$. From this it is clear that $\det P = \det P_{11} \det P_{22}$ and $\det Q = \det Q_{22}/\det P_{11}$. Also, for some $\hat{C} \in \mathcal{L}(\mathbb{R}^n)$

$$\begin{bmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} P_{11}^{-1} & 0 \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} \hat{C} & 0 \\ 0 & I_m \end{bmatrix}. \tag{2.4}$$

It follows that $P_{22}DQ_{22} = I_m$ and therefore $\det D \neq 0$. □

COROLLARY 2. *If (M, L) is regular and $\operatorname{ind}(M, L) = 1$ then, there is a $\delta > 0$ such that if $\|L - L'\| < \delta$, then (M, L') is regular and $\operatorname{ind}(M, L') = 1$.*

In what follows we shall use $P(M, L)Q$ to denote the pencil (PMQ, PLQ). Consider the following example which shows various KNFs of a class of matrix pencil which has the same form as (M, L).

Example 1. Consider the matrix pencil

$$(M, L) \stackrel{\text{def}}{=} \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right).$$

If $d \neq 0$, this pencil has index-1. If $d = 0$, but $bc \neq 0$, it has index-2. If $|d| + |bc| = 0$, it is not regular.

Clearly, if $d \neq 0$, the normal form is given by

$$\begin{bmatrix} 1 & -b/d \\ 0 & 1/d \end{bmatrix} (M, L) \begin{bmatrix} 1 & 0 \\ -c/d & 1 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} a - bc/d & 0 \\ 0 & 1 \end{bmatrix} \right).$$

If $d = 0$ but $bc \neq 0$, then multiplying by the inverse of the second matrix within this pencil gives the normal form

$$L^{-1}(M, L)I = \left(\begin{bmatrix} 0 & 0 \\ 1/b & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right).$$

Within the definition of KNF this corresponds to the vector subspace U being trivial, namely, $U = \emptyset$.

The following theorem generalizes Example 1 to matrix pencils over $\mathbb{R}^{n+m}$.

THEOREM 7. *Suppose that $n \geq 2, \det L(\lambda_0) \neq 0, \mathbf{N}(D(\lambda_0)) = \langle k \rangle, C(\lambda_0)B(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$, and $D'(\lambda_0)k \notin \mathbf{R}(D(\lambda_0))$. There is some $r_0 > 0$ such that*

$$\operatorname{ind}(M, L(\lambda)) = \begin{cases} 1 & \text{if} \quad \lambda \in N_{r_0}(\lambda_0) \backslash \{\lambda_0\}, \\ 2 & \text{if} \quad \lambda = \lambda_0. \end{cases}$$

Moreover, the KNF of $(M, L(\lambda))$ is given as follows. If $\lambda \in N_{r_0}(\lambda_0)\backslash\{\lambda_0\}$, then there are mappings P' and Q' such that

$$P'MQ' = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \text{ and } P'L(\lambda)Q' = \begin{bmatrix} \mathcal{S}_\lambda & 0 \\ 0 & 0 \end{bmatrix}.$$

There are mappings P, Q, and L_0 such that

$$PMQ = \begin{bmatrix} I_u & 0 & 0 \\ 0 & 0 & 0 \\ 0 & C_0 & 0 \end{bmatrix} \text{ and } PL(\lambda_0)Q = \begin{bmatrix} L_0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_m \end{bmatrix},$$

where $C_0 : \mathbb{R} \to \mathbb{R}^m$ is a linear map such that $C_0(1) = k$. If we denote $\mathbf{N}(D(\lambda_0)^T) = \langle u \rangle$ and write $U = \left\langle C(\lambda_0)^T u \right\rangle^\perp$, then $L_0 \in GL(U)$. The mappings PMQ and $PL(\lambda_0)Q$ are elements of $\mathcal{L}(U \oplus \mathbb{R} \oplus \mathbb{R}^m)$.

Proof. From Theorem 5 we know that $\mathcal{N}(A_1(\lambda_0)) = \langle B(\lambda_0)k\rangle$ and that there is an $r_0 > 0$ such that if $\lambda \in N_{r_0}(\lambda_0)$, $\det D(\lambda) = 0$ if and only if $\lambda = \lambda_0$. If $\lambda \neq \lambda_0$ is in this neighborhood, define

$$P' = \begin{bmatrix} I_n & -B(\lambda)D(\lambda)^{-1} \\ 0 & I_m \end{bmatrix} \text{ and } Q' = \begin{bmatrix} I_n & 0 \\ -D(\lambda)^{-1}C(\lambda) & D(\lambda)^{-1} \end{bmatrix}.$$

A multiplication shows that the KNF is given by

$$P'MQ' = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \text{ and } P'L(\lambda)Q' = \begin{bmatrix} (A - BD^{-1}C)(\lambda) & 0 \\ 0 & I_m \end{bmatrix}.$$

Hence $\operatorname{ind}(M, L(\lambda)) = 1$ for all $\lambda \in N_{r_0}(\lambda_0)\backslash\{\lambda_0\}$.

We now calculate $\operatorname{ind}(M, L(\lambda_0))$. We adopt the same notation as Theorem 5 and drop the reference to λ_0 for brevity.

Write $\mathcal{R} \stackrel{\text{def}}{=} \mathbf{R}(A_1) = \langle C^T u\rangle^{\perp}$, so that $\mathbb{R}^n = \mathcal{R} \oplus \langle Bk\rangle$ and $\dim \mathcal{R} = n-1$. Define the restriction

$$A_0 \stackrel{\text{def}}{=} A_1|_{\mathcal{R}} \in GL(\mathcal{R}).$$

Since 0 is an algebraically simple eigenvalue of A_1, there is an invertible map $J : \mathcal{R} \oplus \mathbb{R} \to \mathbb{R}^n$ such that

$$J^{-1}A_1J = \begin{bmatrix} A_0 & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{L}(\mathcal{R} \oplus \mathbb{R}).$$

Therefore

$$J \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \gamma Bk$$

for some $\gamma \in \mathbb{R}$. Without the loss of any generality, we may normalize k to ensure that $\gamma = 1$.

If we premultiply (M, L) by L^{-1} we obtain the pencil

$$L^{-1}(M, L) = \left(\begin{bmatrix} A_1 & 0 \\ C_1 & 0 \end{bmatrix}, I_{n+m} \right) \in \mathcal{L}(\mathbb{R}^{n+m}) \times \mathcal{L}(\mathbb{R}^{n+m}).$$

Conjugating this pencil with the map $\mathcal{J} : \mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m \to \mathbb{R}^{n+m}$ given by

$$\mathcal{J} \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} Jr \\ s \end{bmatrix} \quad (r \in \mathcal{R} \oplus \mathbb{R}, s \in \mathbb{R}^m)$$

yields

$$\mathcal{J}^{-1}(L^{-1}M, I_{n+m})\mathcal{J} = \left(\begin{bmatrix} A_0 & 0 & 0 \\ 0 & 0 & 0 \\ C_1^0 & C_1^1 & 0 \end{bmatrix}, I \right) \in \mathcal{L}(\mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m) \times \mathcal{L}(\mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m),$$

where $C_1^0 = C_1J|_{\mathcal{R}} : \mathcal{R} \to \mathbb{R}^m$ and $C_1^1 = C_1J|_{\mathbb{R}} : \mathbb{R} \to \mathbb{R}^m$. Hence, for $\eta \in \mathbb{R}$, $C_1^1(\eta) = \eta C_1^1(1)$ and

$$C_1^1(1) = C_1J \begin{pmatrix} 0 \\ 1 \end{pmatrix} = C_1Bk = k.$$

Here I is used to denote the identity on $\mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m$.

Conjugate $(\mathcal{J}^{-1}L^{-1}M\mathcal{J}, I)$ by

$$\begin{bmatrix} I_{\mathcal{R}} & 0 & 0 \\ 0 & 1 & 0 \\ -C_1^0 A_0^{-1} & 0 & I_m \end{bmatrix} \in GL(\mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m).$$

Now pre- or postmultiply the resulting pencil by

$$\begin{bmatrix} A_0^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_m \end{bmatrix} \in GL(\mathcal{R} \oplus \mathbb{R} \oplus \mathbb{R}^m).$$

This provides nonsingular transformations P and Q which, when applied simultaneously to M and L, yield

$$PMQ = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & C_1^1 & 0 \end{bmatrix} \text{ and } PL(\lambda_0)Q = \begin{bmatrix} A_0^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_m \end{bmatrix}.$$

Comparing (PMQ, PLQ) to the KNF, we see that the nilpotent mapping N is

$$\begin{bmatrix} 0 & 0 \\ C_1^1 & 0 \end{bmatrix} \in L(\mathbb{R} \oplus \mathbb{R}^m).$$

We see that this is nonzero because $C_1^1(1) = k$. However, $N^2 = 0$ so that $\operatorname{ind}(M, L(\lambda_0)) = 2$. □

The following theorem shows that the index of (M, L) is not always stable to perturbations in L. Of course, we can easily see that the index of (M, L) is not stable to perturbations in M because $\operatorname{ind}(M(\epsilon), L(\lambda)) = 0$ for $\lambda \neq \lambda_0$ and $\epsilon \neq 0$, yet $\operatorname{ind}(M, L(\lambda)) = 1$.

Let us note that one can easily formulate an analogy of Theorem 7 to cover the case when $n = 1$. We omit this for brevity.

Example 2. Let $\phi_i(t)$ be smooth functions and consider the DAE

$$\begin{aligned} \dot{x}(t) &= y_1(t) + \phi_1(t), \\ 0 &= x(t) + y_2(t) + \phi_2(t), \\ 0 &= \lambda y_1(t) + y_2(t) + \phi_3(t). \end{aligned}$$

This can be written in the form $M\dot{z}(t) = L(\lambda)z(t) + f(t)$ such that $(M, L(\lambda))$ satisfies the SIB theorem at $\lambda = 0$. If $\lambda \neq 0$, the solution is easily seen not to depend on any of the derivatives of ϕ_i. We find that the solution of this DAE when $\lambda = 0$ is given by

$$x = \phi_3 - \phi_2, \ y_1 = \dot{\phi}_3 - \dot{\phi}_2 - \phi_1, \ y_2 = -\phi_3.$$

The dependence of the solution on the first derivatives of ϕ_i is to be expected because the index of $(M, L(0))$ is 2.

3. Unfolding the SIB. We continue with the following definition.

Definition 1. *A* singular Hopf curve *is the graph of a continuous function* $\mathcal{H} = \{(\epsilon, \lambda_0(\epsilon)) \in \mathbb{R}^2 : \epsilon \in (0, \epsilon_0)\}$ *such that* $(M(\epsilon), L(\lambda))$ *has a purely imaginary eigenvalue* $i\omega(\epsilon)$ *for all* $(\epsilon, \lambda) \in \mathcal{H}$ *and* $\lim_{\epsilon \to 0_+} \omega(\epsilon) = \infty$. *A point on* $\mathcal{H}$ *is called a Hopf point.*

Let us remark that the existence of a Hopf curve is not sufficient to determine the existence of periodic solutions in an ODE whose linearization is $(M(\epsilon), L(\lambda))$ without further transversality assumptions such as those found in the Hopf bifurcation theorem [12, 1]. Nevertheless, we proceed with the following theorem which provides the *necessary* ingredient for Hopf bifurcation.

THEOREM 8. *Suppose that the matrix pencil $(M, L(\lambda))$ satisfies the hypotheses of Theorem 5 and, adopting the notation of that theorem, $u^T k \neq 0$ and*

$$\frac{u^T C(\lambda_0) B(\lambda_0) k}{u^T k} = -\omega_0^2 < 0. \tag{3.1}$$

Then there is an $\epsilon_0 > 0$ and unique C^r functions $\lambda_0 : (0, \epsilon_0) \to \mathbb{R}$ and $\omega_0 : (0, \epsilon_0) \to (0, \infty)$ such that $(M(\epsilon), L(\lambda_0(\epsilon)))$ has a pair of purely imaginary, algebraically simple eigenvalues $\pm i\omega_0(\epsilon)$. Moreover, $\lambda_0(\epsilon) = \lambda_0 + O(\epsilon)$ and $\epsilon^{1/2}\omega_0(\epsilon) = \omega_0 + O(\epsilon)$ as $\epsilon \to 0_+$.

Proof. If $z = (x, y) \in \mathbb{R}^{n+m}$, let $\Pi_1 z = x$ and $\Pi_2 z = y$ be the projections of z onto coordinate components. Let $N(\epsilon) = \epsilon M(\epsilon)^{-1}$. Write $\rho = \omega^2 \epsilon$ and seek a solution of the augmented system

$$[\rho M(\epsilon) + L(\lambda) N(\epsilon) L(\lambda)]\, z = 0, \tag{3.2a}$$

$$u^T C(\lambda)(\Pi_1 z) - 1 = 0, \tag{3.2b}$$

$$k^T(\Pi_2 z) - 1 = 0. \tag{3.2c}$$

It is clear that a solution of (3.2) provides a purely imaginary pair of eigenvalues of $(M(\epsilon), L(\lambda))$. Write (3.2) as $F(z, \lambda, \rho, \epsilon) = 0$, where $F : \mathbb{R}^{n+m} \times \mathbb{R}^3 \to \mathbb{R}^{n+m} \times \mathbb{R}^2$ is a C^r mapping. Set

$$\rho_0 = \omega_0^2, x_0 = \frac{B(\lambda_0)k}{u^T C(\lambda_0) B(\lambda_0) k}$$

and let y_0 be the unique solution of

$$\frac{-k}{u^T k} + \frac{C(\lambda_0)B(\lambda_0)k}{u^T C(\lambda_0)B(\lambda_0)k} + D y_0 = 0,\ k^T y_0 = 1.$$

It follows that $F((x_0, y_0), \lambda_0, \rho_0, 0) = 0$.

Now $d_{z,\lambda,\rho} F(z, \lambda, \rho, \epsilon) \in \mathcal{L}(\mathbb{R}^{n+m} \times \mathbb{R}^2)$ is given by the matrix

$$\begin{bmatrix} \rho M(\epsilon) + L(\lambda)N(\epsilon)L(\lambda) & L'(\lambda)N(\epsilon)L(\lambda) + L(\lambda)N(\epsilon)L(\lambda)' & M(\epsilon)z \\ u^T C(\lambda)\Pi_1 \cdot & 0 & 0 \\ k^T \Pi_2 \cdot & 0 & 0 \end{bmatrix}. \tag{3.3}$$

Suppose that $d_{z,\lambda,\rho} F(z_0, \lambda_0, \rho_0, 0)[h, \alpha, \beta] = (0, 0, 0)$ for some $(h, \alpha, \beta) \in \mathbb{R}^{n+m} \times \mathbb{R}^2$, and write $h_1 = \Pi_1 h$ and $h_2 = \Pi_2 h$. Then, using the fact that $L(\lambda_0)$ is invertible and removing the reference to λ_0 for brevity,

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} \rho_0 A_1 & 0 \\ \rho_0 C_1 + C & D \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \\ &\quad + \alpha \left\{ L^{-1} L' \begin{bmatrix} 0 \\ \frac{-\rho_0}{u^T C B k} k \end{bmatrix} + \begin{bmatrix} 0 \\ C' x_0 + D' y_0 \end{bmatrix} \right\} + \beta \begin{bmatrix} 0 \\ \frac{1}{u^T C B k} k \end{bmatrix}, \end{aligned} \tag{3.4}$$

where $u^T C h_1 = 0$ and $k^T h_2 = 0$.

Therefore,

$$\rho_0 A_1 h_1 + \alpha(A_1 B' + B_1 D')k = 0$$

and, premultiplying this by $u^T C$ and using the fact that $A_1^T C^T u = 0$, we find $0 = \alpha u^T C B_1 D' k$. Since $CB_1 = I - DD_1$, it follows that

$$0 = \alpha u^T C B_1 D' k = \alpha u^T (I - DD_1) D' k = \alpha u^T D' k$$

and $\alpha = 0$ follows. Hence there is a $\theta_1 \in \mathbb{R}$ such that $h_1 = \theta_1 Bk$, so that $0 = u^T C h_1 = \theta_1 u^T CBk$ and $h_1 = 0$ also follows.

Finally, therefore, $Dh_2 + \frac{\beta}{u^T CBk} k = 0$. As $u^T k \neq 0$, $\beta = 0$ must follow. Therefore, there is a $\theta_2 \in \mathbb{R}$ such that $h_2 = \theta_2 k$. However, $k^T h_2 = 0$ implies that $h_2 = 0$. Hence, the derivative $d_{z,\lambda,\rho} F((x_0, y_0), \lambda_0, \rho_0, 0)$ is a bijection and the result follows by the implicit function theorem. ∎

The sign condition (3.1) within Theorem 8 can be replaced by

$$\frac{u^T C(\lambda_0) B(\lambda_0) k}{u^T k} = \omega_0^2 > 0$$

and the conclusions of the above theorem remain true, but instead for the matrix pencil $(M(-\epsilon), L(\lambda))$.

The following system of ODEs from [13] probes Theorem 8 quite effectively. It is a model of cooperative phenomena which occur in oxygen-hemoglobin reactions taking place in the bloodstream (details of this can be found in [13, p. 119]). The model is

$$\frac{du}{dt} = -u + (u - a_3 u + a_1)v_1 + (a_4 + u)v_2, \tag{3.5a}$$

$$\epsilon \frac{dv_1}{dt} = u - (u + a_3 u + a_1 + a_2)v_1 + (a_4 + a_5 - u)v_2, \tag{3.5b}$$

$$\epsilon \frac{dv_2}{dt} = a_3 u v_1 - (a_4 + a_5)v_2, \tag{3.5c}$$

where $u(0) = 1$ and $v_1(0) = v_2(0) = 0$.

The author of [13] only considers the behavior of (3.5) for large time by finding the projection of the flow onto the one dimensional slow manifold. Instead, suppose we consider the stability of the trivial equilibrium which exists for all parameter values a_i. The linearization of (3.5) at this equilibrium point is the matrix

$$M(\epsilon)^{-1} L(\lambda, \mu) \stackrel{\text{def}}{=} \left[\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \epsilon^{-1} & 0 \\ 0 & 0 & \epsilon^{-1} \end{array}\right] \left[\begin{array}{c|cc} -1 & a_1 & a_4 \\ \hline 1 & -\lambda & \mu \\ 0 & 0 & -\mu \end{array}\right], \tag{3.6}$$

where $\lambda = a_1 + a_2$ and $\mu = a_4 + a_5$. We consider the stability properties of the origin when λ and μ vary, with the other parameters fixed and ϵ small.

For fixed $\lambda \neq 0$ and varying μ, a slow-manifold singularity is encountered by the trivial equilibrium at $\mu = 0$. However, $\omega_0 = 0$ in the notation of Theorem 8. One finds on inspection that a Hopf curve does not emanate from the point $(\mu, \epsilon) = (0, 0)$.

However, for a fixed $\mu \neq 0$ and varying λ, Theorem 8 is satisfied at $\lambda = 0$, provided $a_1 < 0$. A curve of Hopf points exists, and is given by the graph of λ_0. Theorem 8 indicates that

$$\omega_0(\epsilon) = \epsilon^{-1/2}\sqrt{-a_1} + O(\epsilon^{1/2}), \ \lambda_0(\epsilon) = -\epsilon + O(\epsilon^2).$$

A simple calculation shows that

$$\omega_0(\epsilon) = \epsilon^{-1/2}\sqrt{a_1 + \epsilon} \text{ and } \lambda_0(\epsilon) = -\epsilon.$$

The value of μ has no effect on the appearance of a Hopf curve in (λ, ϵ)-space, but μ must be nonzero for Theorem 8 to apply. We therefore see that Theorem 8 provides sufficient, but not necessary conditions for the existence of a Hopf curve. Namely, the invertibility of $L(\lambda_0)$, which is used to prove Theorems 5 and 8, is not needed for a Hopf curve to exist.

REFERENCES

[1] V. I. Arnold, V. S. Afrajmovich, Y. S. Il'yashenko, and L. P. Shil'nikov, *Bifurcation Theory and Catastrophe Theory*, Springer-Verlag, New York, 1999.

[2] S. M. Baer and T. Erneux, *Singular Hopf bifurcation to relaxation oscillations*, SIAM J. Appl. Math. II, 52 (1992), pp. 1651–1664.

[3] R. E. Beardmore, *Stability and bifurcation properties of index-1 DAEs*, Numer. Algorithms, 19 (1998), pp. 43–53.

[4] R. E. Beardmore, *Flows near singular equilibria in differential-algebraic equations*, SIAM J. Matrix Anal. Appl., submitted.

[5] E. Benoit, *Systèmes lents-rapides dans R^3 et leurs canards*, Astérisque, (1983), pp. 159–191.

[6] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, Amsterdam, 1989.

[7] S. Campbell, R. Hollenbeck, and W. Marszalek, *Mixed symbolic-numeric computations with general DAE I: System properties*, Numer. Algorithms, 19 (1998), pp. 73–84.

[8] F. Dumortier and R. Roussarie, *Canard cycles and center manifolds*, Memoirs of the AMS, 121 (1996), p. 1.

[9] F. R. Gantmacher, *Theory of Matrices*, Chelsea Publishing Co., New York, 1977.

[10] G. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1989.

[11] T. Kato, *Perturbation Theory for Linear Operators*. Reprint of the 1980 Edition, Springer-Verlag, New York, 1995.

[12] J. E. Marsden and M. McCracken, *The Hopf Bifurcation Theorem and its Applications*, Springer-Verlag, New York, 1976.

[13] J. D. Murray, *Mathematical Biology*, 2nd ed., Biomathematics 19, Springer-Verlag, Berlin, 1980.

[14] R. Riaza, S. L. Campbell, and W. Marszalek, *On singular equilibria of index-1 DAEs*, Circuits Systems Signal Process, (2000), pp. 131–157.

[15] M. Stiefenhofer, *Singular perturbation with limit points in the fast dynamics*, Z. Zngew. Math. Phys., 49 (1998), pp. 730–758.

[16] V. Venkatasubramanian, H. Schattler, and J. Zaborszky, *A Stability Theory of Large Differential Algebraic Systems, a Taxonomy, No. SSM* 9201 *Part* 1, Technical Report, Washington University, St. Louis, MO, 1992.

[17] V. Venkatasubramanian, H. Schattler, and J. Zaborszky, *Dynamics of large constrained nonlinear systems—a taxonomy theory*, Proceedings of the IEEE, 83 (1995), pp. 1530–1558.

[18] A. K. Zvonkin and M. A. Shubin, *Non-standard analysis and singular perturbations of ordinary differential equations*, Russian Math. Surveys, 39 (1984), pp. 69–131.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 138–166
© 2001 Society for Industrial and Applied Mathematics

THE DEGENERATE BOUNDED ERRORS-IN-VARIABLES MODEL*

S. CHANDRASEKARAN†, M. GU‡, A. H. SAYED§, AND K. E. SCHUBERT¶

Abstract. We consider the following problem: $\min_{x\in\mathcal{R}^n} \min_{\|E\|\le\eta} \|(A+E)x-b\|$, where A is an $m\times n$ real matrix and b is an n-dimensional real column vector when it has multiple global minima. This problem is an errors-in-variables problem, which has an important relation to total least squares with bounded uncertainty. A computable condition for checking if the problem is degenerate as well as an efficient algorithm to find the global solution with minimum Euclidean norm are presented.

Key words. least squares problems, total least squares, errors-in-variables, parameter estimation

AMS subject classifications. 65F05, 65F20, 65F25, 65G05, 93B40, 93E24

PII. S0895479800357766

1. Introduction.

In this paper we consider the following problem:

$$\min_{x\in\mathcal{R}^n} \min_{\|E\|\le\eta} \|(A+E)x-b\|, \tag{1.1}$$

where A is an $m\times n$ real matrix and b is a real n-vector. This problem is a special case of the errors-in-variables problem, which we have given the formal name of the degenerate bounded errors-in-variables problem. For ease of reference we usually call the problem the degenerate min-min problem, since degenerate bounded errors-in-variables problem is a bit long. This problem can be viewed as a total least squares (TLS) problem [3, 4] with bounds on the uncertainty in the coefficient matrix, which we will explain in more detail in section 3. In this paper we make frequent use of the terms degenerate and nondegenerate. Simply put, a degenerate problem is one where multiple solutions exist. The nondegenerate case of this problem occurs when η is small and b is in some sense far from the range of A. That η should be small is intuitive, since for $\eta = 0$ we are left with the least squares problem, which is nondegenerate (unique solution) when A has full column rank. Conversely, when η is larger than the smallest singular value of A, we would anticipate degeneracy (multiple solutions) as the perturbed matrix $A+E$ is not guaranteed to be full column rank. The intuition behind b needing to be far from the range of A for nondegeneracy comes from the fact that if b were close enough that multiple perturbations E existed such that b was in the range of $A+E$, then multiple solutions (degeneracy) would exist. In [2] we considered the nondegenerate case of this problem and showed how to

*Received by the editors April 7, 2000; accepted for publication (in revised form) by S. Van Huffel August 18, 2000; published electronically June 8, 2001. The first and fourth authors were partially supported by NSF grant CCR-9734290. The third author was supported in part by the NSF under award CCR-9732376.

http://www.siam.org/journals/simax/23-1/35776.html

†Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 (shiv@ece.ucsb.edu).

‡Department of Mathematics, University of California, Los Angeles, CA 90095 (mgu@math.ucla.edu).

§Department of Electrical Engineering, University of California, Los Angeles, CA 90095 (sayed@ee.ucla.edu).

¶Mathematics Department, University of Redlands, Redlands, CA 92373-0999 (schubert@jasper.uor.edu).

compute its unique solution in $O(mn^2)$ flops. In this paper we consider the problem when it is degenerate; that is, when it has multiple solutions. In particular, we present an $O(mn^2)$ algorithm to find the solution with the minimum Euclidean norm. The degenerate case is actually the generic case for this problem, and hence is more important than the nondegenerate case. This can be seen from the simple discussion above, since the nondegenerate case holds only for certain combinations of b and A when η is smaller than the smallest singular value of A. This is very restrictive, and hence the claim.

We begin this paper with a motivational problem, which shows the advantage of considering this criterion. We proceed by outlining the proof and presenting the algorithm to solve the problem. We then proceed with the full proof of the problem. We conclude with a tabulation of the results and an extension to the problem of a column partitioned matrix with uncertainty in only one partition.

2. Motivation. Many different methods exist for solving the basic estimation problem of finding some vector of unknowns x, from a vector of observations b, by using a matrix of relations A. Probably the two best known methods are least squares and TLS. We now want to get a feel for how these problems operate on a simple example and to see if there is any room for improvement. Consider, for example, a simple one dimensional "skyline" image that has been blurred. A "skyline" image is a one dimensional image that looks like a city skyline when graphed, and thus is the most basic image processing example. "Skyline" images involve sharp corners, and it is of key importance to accurately locate these corner transitions. Blurring occurs often in images; for example, atmospheric conditions, dust, or imperfections in the optics can cause a blurred image. Blurring is usually modeled as a Gaussian function or Gaussian blur, which incidentally is a great smoothing filter. The Gaussian blur causes greater distortion on the corners, which is exactly where we do not want it to happen. The Gaussian blur with standard deviation, σ, can be modeled as a matrix, A, with the component in position, (i,j), given by

$$A_{i,j} = e^{-(i-j)^2\sigma}.$$

If we go on the presumption that we do not know the exact blur that was applied (σ unknown) we cannot expect to get the exact system back. We realize that we will not be able to perfectly extract the original system, but we want to see if we can get a little more information than we have now. We "know" the blur is small compared to the information so we are confident that we should be able to get something. The least squares solution fails completely, yielding a result that is about three orders of magnitude off; see Figure 1. We notice that the TLS solution is better than the least squares solution, but still not acceptable. The degenerate min-min problem yields great results. From this simple example we can see that there is room for improvement.

3. Geometric understanding. Probably the easiest way to understand the problem at hand is to look at it geometrically. For ease of drawing we will consider A and b to be vectors of length 2. Note that while this is useful for getting a basic understanding some of the key features of the problem do not appear in this case. For instance, when A has multiple columns the problem can be degenerate for small values of η. In such a case the degenerate min-min problem has several advantages over other formulations, such as TLS. One such advantage is the perturbation on A is much smaller in the degenerate min-min problem than in the TLS problem.

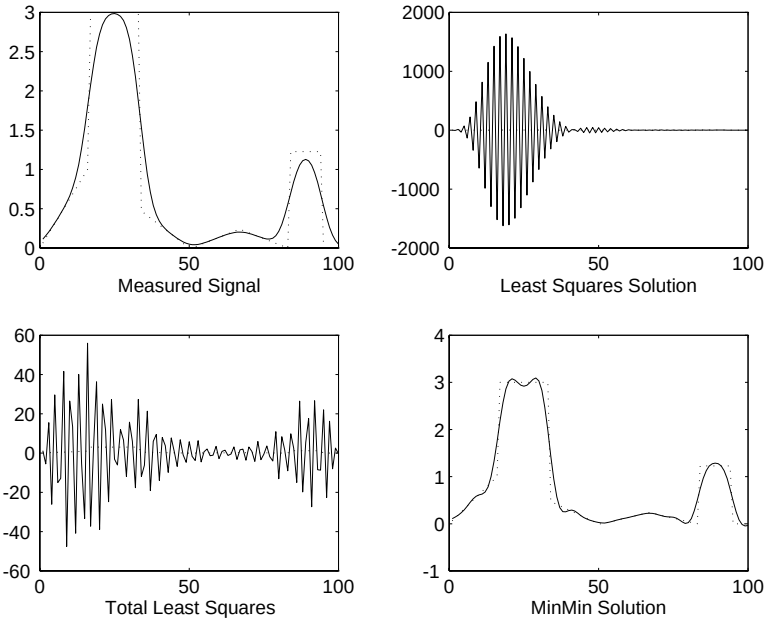


FIG. 1. *Skyline problem.*

For comparison we start with the classic problem of least squares (see Figure 2). The solution to the least squares problem is found by projecting b into A. This is a common geometric view of the problem, but forms a basis for understanding the other problems.

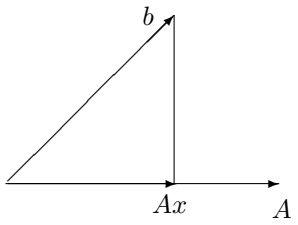


FIG. 2. *Least squares.*

In TLS, we allow A to be perturbed by a matrix E and b to be perturbed by a vector f (see Figure 3). The net effect is that both A and b are projected into a plane between the two such that the norm of $[\,E \quad f\,]$ is minimized. The TLS problem can thus be formulated as $\min \|[\,E \quad f\,]\|$ such that $(b+f) \in \mathcal{R}(A+E)$. Note that because of this A can be moved arbitrarily far.

In the general min-min problem (degenerate or not), we project A and b into a plane between the two as we did in the TLS problem, but we put a bound on how far A can be perturbed (see Figure 4). Note that the cone around A shows us the boundary of possible perturbations to A. We are in essence solving the problem $\min \|[\,E \quad f\,]\|$ such that $(b+f) \in \mathcal{R}(A+E)$ and $\|E\| \leq \eta$. The problem at hand can thus be thought of as a TLS problem with bounds on the errors in A.

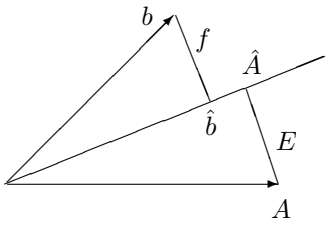


FIG. 3. *Total least squares (TLS).*

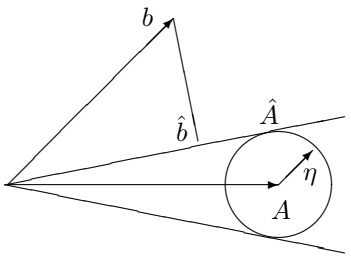


FIG. 4. *Min-min problem.*

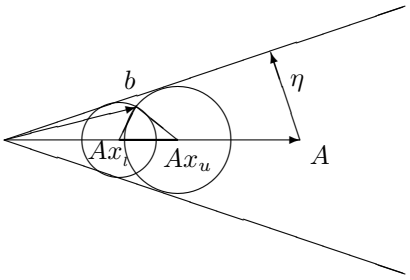


FIG. 5. *Degenerate min-min problem.*

To get a better understanding of the degenerate problem, we will consider one of the ways the problem can become degenerate. The easiest to visualize, and the only one that can be drawn in two dimensions, is the case when b lies in the cone of possible perturbations of A (see Figure 5). In this case we see that any $\hat{x}$ such that $x_l \leq \hat{x} \leq x_u$ is a solution to the problem. The perturbations $E(\hat{x})$ change, but each $\hat{x}$ in the range still solves the problem. We are now left with a problem, namely, which $\hat{x}$ do we choose. The most conservative choice is to pick the smallest one, which is what we do. This choice has a lot to recommend it, but a full discussion is outside the bounds of the paper at hand. In section 6, we take advantage of this basic insight (picking the smallest solution) to reformulate the problem into a unique problem.

4. Proof outline. The proof is long and technically involved, so we provide this overview. The cost function presented is useful for seeing how this problem handles the uncertainty in the matrix A, but it is not immediately useful in solving the problem. For instance, checking if a problem is degenerate in the original form of the problem

is tedious. We thus desire to rewrite the problem into a simpler form, and then find a computable condition for degeneracy. We start the proof with the cost function of the problem we want to solve,

$$\min_{x\in\mathcal{R}^n} \min_{\|E\|\leq\eta} \|(A+E)x-b\|,$$

and the degeneracy condition which was found in [2],

$$\eta\|x\| \geq \|Ax-b\|.$$

Since the nondegenerate case is already solved, we proceed by assuming the degeneracy condition holds. The first step is to minimize the cost function over $\|E\| \leq \eta$, and find that the optimal cost is zero. Since the problem is degenerate and the cost function is zero, we choose the solution with the smallest norm to obtain the problem

$$\min_{\|Ax-b\|\leq\eta\|x\|} \|x\|.$$

The condition $\eta\|x\| \geq \|Ax-b\|$ is not practical for checking for degeneracy in a problem, as mentioned above, since it requires the checking multiple values of x to hopefully find one that holds and thus showing the problem is degenerate. The second step is thus to find a computable condition for degeneracy. We proceed by squaring the condition for degeneracy and using the singular value decomposition (SVD) of A to find the two cases in which the problem is degenerate. The first case is when η is larger than the smallest singular value of A. The first case is always degenerate. The second case is when η is not larger than the smallest singular value of A. The second case is degenerate only when

$$b^T(I - A(A^TA - \eta^2 I)^{-1}A^T)b \leq 0.$$

While we now know when the problem is degenerate, we still need to show how to get the solution. We would like to be able to use Lagrange multiplier techniques to find the solution. We thus need to reduce the inequality $\eta\|x\| \geq \|Ax-b\|$ to an equality if possible. The third step of the proof is a proof that the solution, $\hat{x}$, is actually on the boundary of the inequality, and thus $\eta\|\hat{x}\| = \|A\hat{x}-b\|$.

We then proceed in the fourth step to use Lagrange multiplier techniques to parameterize the solution, $\hat{x} = x(\alpha)$, in terms of a single variable, α, thus reducing the problem to finding the zeros a secular equation. A secular equation is a rational expression of one variable, which we construct so that all the critical points of the original problem occur at zeros of the secular equation. The secular equation reduces our n-dimensional search for the solution, $\hat{x}$, to a one dimensional search. We denote the solution to the original problem as $x(\alpha^o)$, and note that it will occur at one of the $2n$ zeros of the secular equation. The zero of the secular equation which corresponds to $x(\alpha^o)$ is denoted α^o.

The remainder of the proof is concerned with showing which zero is α^o. Toward this end we start the second half of the proof with an assertion of the answer. The unique zero of the secular equation in the interval $[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$ is α^o, where σ_1 is the largest singular value of A and σ_n is the smallest. We prove this by a process of elimination.

To begin with we us Lagrange techniques (first and second order conditions on the Lagrangian) to narrow down the search area. By employing these techniques, we

find that α^o must lie in the interval $[\max(-\sigma_{n-1}^2, -\eta^2), \eta\sigma_1]$. This still admits several possibilities; see Figure 6. First of all there are two critical points ($\alpha = -\sigma_n^2$ and $\alpha = -\sigma_{n-1}^2$) which could be α^o. Second, α^o could be in either interval ($(-\sigma_n^2, \eta\sigma_1)$ or the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$).

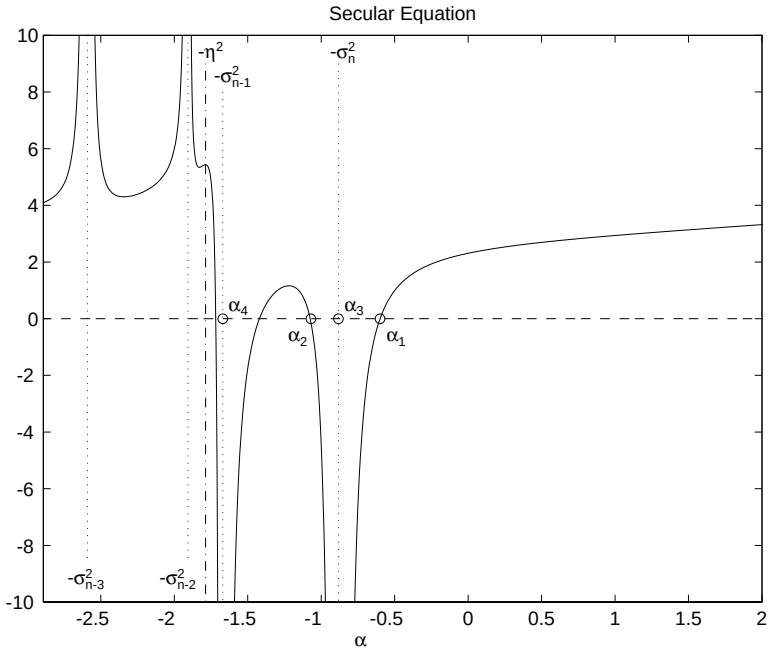


FIG. 6. *Secular equation.*

In particular, note that the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ can have multiple zeros in it, so we must also deal with this possibility. We put the arguments that only the rightmost root in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ is a candidate to be α^o in Appendix B. With this dealt with there are only four candidates zeros of $g(\alpha)$ to handle, which we denote by α_1 through α_4. We thus introduce the four candidates: $\alpha_1 \in (-\sigma_n^2, \eta\sigma_1]$, α_2 is the rightmost root in $(-\sigma_{n-1}^2, -\sigma_n^2)$, $\alpha_3 = -\sigma_n^2$, and $\alpha_4 = -\sigma_{n-1}^2$ (see Figure 6). To show that α^o is the unique root in $[-\sigma_n^2, \eta\sigma_1]$, we examine six cases. Most of the work is involved at this stage, and hence most of the mathematical difficulties occur here. The basic idea is to eliminate the possibility that any root except the one that occurs in the interval $[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$ can be α^o. Additionally we must show the existence and uniqueness of the zero. With this established we can then use bisection or Newton's method to find the root in our algorithm.

You might be wondering why we need to use six cases to prove the assertion that α^o lies in the interval $[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$. The reason lies in three basic factors which affect the shape of the secular equation. The first and most obvious is the size of η. Note, for instance, that if $\eta < \sigma_n$, then only one of the zeros α_1 is a candidate for α^o since we have from an earlier condition (first order condition on the Lagrangian) that $\alpha^o > -\eta^2$. Obviously to consider some of the candidates, such as α_4, we need to assume that η is large enough to admit the possibility. The cases just let us organize the assumptions into convenient groups to handle. See Figure 7. The dotted vertical lines mark where the singular values are, and the dash-dotted vertical line indicates

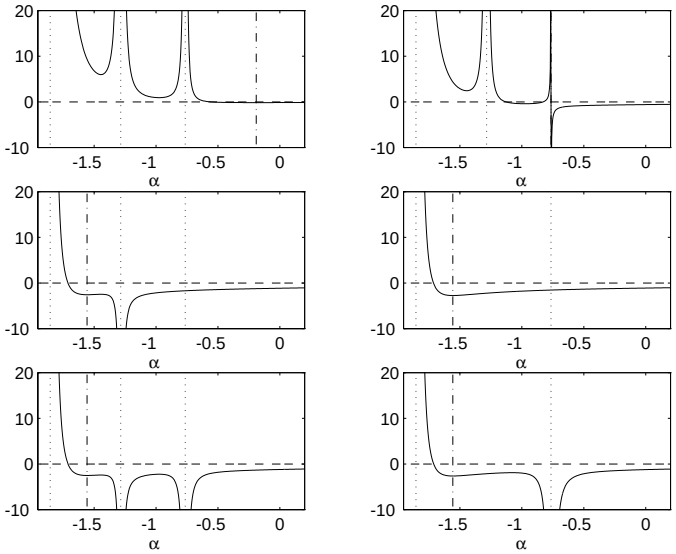


FIG. 7. *Six cases of proof. (UL) Case* 1*: $\eta < \sigma_n$; (UR) Case* 2*: $\eta = \sigma_n$; (ML) Case* 3*: $\eta > \sigma_n$, b is orthogonal to the left singular vector of σ_n, and $\sigma_n < \sigma_{n-1}$; (MR) Case* 4*: $\eta > \sigma_n$, b is orthogonal to the left singular vectors of σ_n, and σ_n has multiplicity k; (LL) Case* 5*: $\eta > \sigma_n$, b is not orthogonal to the left singular vector of σ_n, and $\sigma_n < \sigma_{n-1}$; (LR) Case* 6*: $\eta > \sigma_n$, b is not orthogonal to the left singular vectors of σ_n, and σ_n has multiplicity k.*

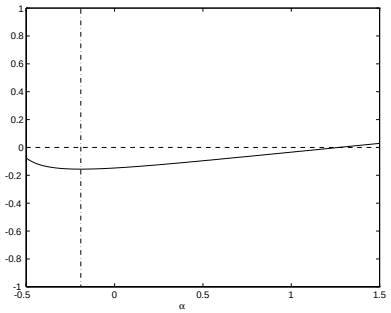


FIG. 8. *Expanded view of Case* 1 *zero.*

where $-\eta^2$ is. We note that in Case 1 of Figure 7, it looks like the secular equation becomes flat to the right of $\alpha = -0.5$ but it does not. The scale makes the graph hard to read, so we provide an expanded view of the region in Figure 8. In Case 1 we consider η small ($\eta < \sigma_n$), in Case 2 we consider the special case of $\eta = \sigma_n$, and finally in Cases 3–6 we consider η to be large ($\eta > \sigma_n$).

When η is large there are more possibilities. The first is that the smallest singular value might have multiplicity of two or more. This can be exploited to simplify the problem. In particular, α_2 does not exist in this case, and $\alpha_3 = \alpha_4$. The cases where $\sigma_n < \sigma_{n-1}$ are the more difficult ones. The second is that b might be orthogonal to the left singular vector(s) of A, which correspond to smallest singular value. This drastically changes the shape of the graph of the secular equation in the region around $\alpha = -\sigma_n^2$. See, for instance, the middle left graph in Figure 7. The pole which normally appears at $-\sigma_n^2$ is not present. In fact, the only time α_3 can be α^o is when

b is orthogonal to the left singular vector(s) of A, which corresponds to the smallest singular value (σ_n). Similarly, the only time α_4 can be α^o is when b is orthogonal to the left singular vector(s) of A, which corresponds to the second smallest singular value (σ_{n-1}). Note that if the smallest singular value has multiplicity of at least two, then $\sigma_n = \sigma_{n-1}$. This case is shown on the middle right graph of Figure 7. The last four cases cover all the combinations of singular value multiplicity and b vector orthogonality which occurs when η is large.

5. Algorithm. For the reader's convenience we present pseudocode for the algorithm in this section. The syntax has been designed to be Matlab-like. Three lines deserve particular attention, though. The first one to appear states "solve nondegenerate problem." In this case the problem is not degenerate so you will need to provide code for the nondegenerate case as outlined in [2]. The next line that could be confusing starts with "pick any Θ." In this case any unit vector, Θ, will solve the problem. An additional condition could be placed on the solution, $\hat{x}$, to select a specific Θ or to meet special requirements of the specific problem, so we leave it unspecified in our pseudocode. The final line that requires clarification starts with $\alpha \in \left[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1\right]$. In this case you are finding the root of $g(\alpha)$ in the specified range, so any root finder you prefer (for instance, bisection or Newton's method) can be used.

$[U, \Sigma, V] = SVD(A)$;
$b_1 = U^T b$;
$cond = 0$;
if ($\eta < \sigma_n$) or ($\eta = \sigma_n$ and $b_1(n) = 0$)
  if ($b^T(I - A(A^TA - \eta^2 I)^{-1}A^T)b > 0$)
    solve nondegenerate problem
  else
    $cond = 1$;
  end
else
  if ($\eta = \sigma_n$)
    $cond = 1$;
  else
    if ($\sigma_n < \sigma_{n-1}$) and ($b_1(n) = 0$) and ($g(-\sigma_n^2) \geq 0$)
      $\bar{\Sigma}_1 = \Sigma(1 : n-1, 1 : n-1)$;
      $\bar{b}_1 = b_1(1 : n-1)$;
      $\hat{x} = V \begin{bmatrix} (\bar{\Sigma}_1^2 - \sigma_n^2 I)^{-1}\bar{\Sigma}_1\bar{b}_1 \\ \pm\sqrt{\frac{g(-\sigma_n^2)}{\eta^2 - \sigma_n^2}} \end{bmatrix}$;
    elseif ($\sigma_n = \sigma_{n-k+1} < \sigma_{n-k}$) and ($\|b_1(n-k+1 : n)\| = 0$)
        and ($g(-\sigma_n^2) \geq 0$)
      $\bar{\Sigma}_1 = \Sigma(1 : n-k+1, 1 : n-k+1)$;
      $\bar{b}_1 = b_1(1 : n-k+1)$;
      $r = \sqrt{\frac{g(-\sigma_n^2)}{\eta^2 - \sigma_n^2}}$;
      Pick any $\Theta \in \mathbf{R}^k$ such that $\|\Theta\| = 1$;
      $\hat{x} = V \begin{bmatrix} (\bar{\Sigma}_1^2 - \sigma_n^2 I)^{-1}\bar{\Sigma}_1\bar{b}_1 \\ r\Theta \end{bmatrix}$;
    else
      $cond = 1$;
    end

```
    end
  end
  if cond == 1
    α ∈ [max(−σ_n^2, −η^2), ησ_1] such that g(α) = 0
    x̂ = (A^T A + αI)^† A^T b;
  end
```

Where $g(\alpha)$ is given by

$$g(\alpha) = b_2^T b_2 + b_1^T \left(\Sigma_1^2 + \alpha I\right)^{-2} \left(\alpha^2 I - \eta^2 \Sigma_1^2\right) b_1,$$

and

$$\begin{aligned} A &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T, \\ b_1 &= U_1^T b, \\ b_2 &= U_2^T b. \end{aligned}$$

6. Minimization over *E*. We now start the full proof. The proof will be much easier to read if the reader is familiar with the preceding papers [1, 2]. For the reader's convenience we will place major milestones in the proof in boxes at the end of the sections where the milestone occurs. We assume that the problem is degenerate and in particular that there exists an x such that $\eta\|x\| \geq \|Ax - b\|$. We will soon provide equivalent computable criteria for degeneracy; however, this formulation is more useful for the present. Our goal in this section is to reduce the problem to an equivalent formulation that does not involve E. The goal is accomplished by showing the degenerate problem is equivalent to requiring the solution to be in the set $\{x|\eta\|x\| \geq \|Ax - b\|\}$. We begin by showing that the problem requires that we be in the set, then show that any $\hat{x}$ in the set solves the problem. Note that the method used to get E is related to the formulation in [5], though we provide the full argument for the ease of the reader. Under the assumption that the problem is degenerate it follows that

$$\min_x \min_{\|E\| \leq \eta} \|Ax - b + Ex\| = 0,$$

since for any x such that $\eta\|x\| \geq \|Ax - b\|$ we can choose

$$E = -\gamma\eta \frac{(Ax - b)x^T}{\|Ax - b\|\|x\|}, \quad 0 \leq \gamma \leq 1,$$

and obtain

$$0 \leq \min_x \min_{\|E\| \leq \eta} \|Ax - b + Ex\| \leq \|Ax - b\| \left|1 - \gamma \frac{\eta\|x\|}{\|Ax - b\|}\right|,$$

and for the choice

$$\gamma = \frac{\|Ax - b\|}{\eta\|x\|} \leq 1,$$

the upper bound is zero. Since there exists an E which makes the minimum zero, the minimum value of the norm is zero. Therefore we only need consider the equation

$$Ax - b + Ex = 0 \tag{6.1}$$

with the constraint $\|E\| \leq \eta$. This constrained equation is equivalent to being on the set defined by

$$\|Ax - b\| \leq \eta \|x\|. \tag{6.2}$$

To prove this, we first show that if the constrained equation (6.1) is met, then we are in the set (6.2).

$$Ax - b + Ex = 0,$$

$$Ax - b = -Ex.$$

Taking the norm of both sides we obtain

$$\|Ax - b\| = \|Ex\|$$

and we note that this implies

$$\|Ax - b\| \leq \|E\| \|x\|.$$

Then using the constraint on the perturbation size, $\|E\| \leq \eta$, we obtain

$$\|Ax - b\| \leq \eta \|x\|$$

and we have the desired result. We now show that if we are in the set (6.2), then the constraint equation (6.1) is met. This is accomplished by showing that for any x in the set, there exists a perturbation, E_0, such that the constraint equation is satisfied. To do this consider

$$E_0 = -\frac{(Ax - b)x^T}{\|x\|^2}.$$

We first note that this perturbation satisfies the constraint on the size of the perturbations ($\|E\| \leq \eta$).

$$\|E_0\| \leq \frac{\|Ax - b\|}{\|x\|}.$$

Since on the set $\|Ax - b\| \leq \eta \|x\|$ we have

$$\|E_0\| \leq \eta,$$

we now consider the equation given by $Ax - b + E_0 x$. We note that this is

$$Ax - b + E_0 x = Ax - b - (Ax - b).$$

Thus we have trivially that $Ax - b + E_0 = 0$ and the assertion is proven.

We know there are multiple solutions which will solve the problem as stated. Since any will solve the original problem, we are free to add an additional constraint which will simplify the solution and ensure the solution meets other requirements. A reasonable choice is to pick the solution with the minimum norm. Other nice properties of this choice also recommend it. For instance, it is possible under certain conditions for the min-max solution (from [1]) to also solve the degenerate min-min

problem. When this occurs the min-max solution is the solution to the degenerate problem with minimum norm. We do not prove this for reasons of space, but it does provide a good understanding of the relationships between the problems and gives additional motivation for the choice. Using the choice of the minimum norm solution, the problem can be rewritten into the better form, as follows.

The degenerate problem can be reformulated as a unique problem by considering

$$\min_{\|Ax-b\| \leq \eta \|x\|} \|x\|.$$

7. Computable conditions for degeneracy. The constraint, $\|Ax - b\| \leq \eta\|x\|$, defines the set on which our solution lies and is thus referred to as the feasibility constraint. The feasibility constraint can be squared and expanded to obtain

$$x^T A^T A x - 2x^T A^T b + b^T b \leq \eta^2 x^T x. \tag{7.1}$$

Let $A = U\Sigma V^T$ be the SVD of A conformally partitioned as follows:

$$U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix},$$

and define both $b_i = U_i^T b$ for $i = 1, 2$, and $z = V^T x$. These definitions are made solely to simplify the expressions we are working with and provide a convenient shorthand for the rest of the problem. Then inequality (7.1) can be simplified to obtain

$$z^T \Sigma_1^2 z - 2z^T \Sigma_1 b_1 + b_1^T b_1 + b_2^T b_2 \leq \eta^2 z^T z. \tag{7.2}$$

Now assuming that the singular values are in decreasing order, partition Σ_1 as follows:

$$\Sigma_1 = \begin{pmatrix} \Sigma_+ & 0 \\ 0 & \Sigma_- \end{pmatrix},$$

where $\Sigma_+^2 - \eta^2 I \geq 0$ and $\Sigma_-^2 - \eta^2 I < 0$. Also conformally partition z and b_1

$$z = \begin{pmatrix} z_+ \\ z_- \end{pmatrix} \quad b_1 = \begin{pmatrix} b_{1+} \\ b_{1-} \end{pmatrix}.$$

Then inequality (7.2) can be expanded into

$$\begin{aligned} 0 \geq \; & z_+^T(\Sigma_+^2 - \eta^2 I)z_+ - 2z_+^T \Sigma_+ b_{1+} + b_{1+}^T b_{1+} \\ & + z_-^T(\Sigma_-^2 - \eta^2 I)z_- - 2z_-^T \Sigma_- b_{1-} + b_{1-}^T b_{1-} \\ & + b_2^T b_2. \end{aligned}$$

Now we observe that if Σ_- is nonempty, then the inequality always has at least one z which makes it true. In other words if $A^T A - \eta^2 I$ is indefinite, then the problem is always degenerate. On the other hand, if $A^T A - \eta^2 I$ is positive-semidefinite, then degeneracy depends on the vector b. To get a computable condition for degeneracy, we first note that when $x = 0$ we have that the constraint is nonnegative. We proceed by minimizing the expression

$$x^T(A^T A - \eta^2 I)x - 2x^T A^T b + b^T b$$

and when $\eta \neq \sigma_i$ we obtain

$$x_o = (A^T A - \eta^2 I)^{-1} A^T b.$$

Now, when $A^T A - \eta^2 I$ is positive, we must have that the constraint is nonpositive at this point. On plugging this back into the expression being minimized we obtain

$$b^T (I - A(A^T A - \eta^2 I)^{-1} A^T) b \leq 0 \tag{7.3}$$

as the required computable condition for the problem to be degenerate when $\eta < \sigma_n$.

The problem is degenerate if either

$$\eta > \sigma_n$$

or

$$b^T (I - A(A^T A - \eta^2 I)^{-1} A^T) b \leq 0.$$

8. Solution is on the boundary. We want to establish that the optimal solution is obtained at the boundary of the feasible set; that is, at the minimum norm solution the inequality is actually an equality. Mathematically this means the feasibility constraint, $\|Ax - b\| \leq \eta \|x\|$, is actually an equality, $\|Ax - b\| = \eta \|x\|$. To prove this we use the shorthand developed in the last section that given the SVD of A, then $b_i = U_i^T b$ for $i = 1, 2$, and $z = V^T x$. The problem of finding the solution with the smallest norm to the degenerate problem can now be recast as minimizing $z^T z$ subject to the inequality constraint (7.3).

Now if $b = 0$, then clearly the minimum norm solution is $z = 0$ which does lie on the boundary ($0 = 0$). So we restrict ourselves to the case when $b \neq 0$. Let us denote by $f(z)$ the expression on the left-hand side of inequality (7.3). Then it is clear that $f(0) > 0$, and therefore $z = 0$ is not a feasible point. Now suppose that contrary to our hypothesis that the optimal solution occurs at an interior point. Denote that optimal solution by z_0. Since it is an interior point we must have $0 > f(z_0)$. Let γ denote a scalar and consider the function $f(\gamma z_0)$ as γ varies. Since $f(\cdot)$ is a continuous function it follows that as γ is decreased from 1 towards 0, the value of $f(\gamma z_0)$ must at sometime become equal to 0. But now we have a contradiction as $\|\gamma z_0\| < \|z_0\|$ for $0 < \gamma < 1$. Hence we prove our hypothesis that the optimal solution must lie on the boundary of the feasible set.

Therefore we can restrict our attention to the problem

$$\min_{\|Ax-b\| = \eta \|x\|} \|x\|.$$

We note that the problem is unaffected by squaring, thus to simplify the algebra we will work with the squared problem.

The problem is equivalently stated as

$$\min_{\|Ax-b\|^2 = \eta^2 \|x\|^2} \|x\|^2.$$

9. Reduction to secular equation. Since we have reduced the problem to an equality constrained minimization problem, we can use the method of Lagrange multipliers. Letting λ denote the Lagrange multiplier we obtain the following set of equations that characterize the critical points

$$x + \lambda \left(A^T (Ax - b) - \eta^2 x \right) = 0.$$

Simplifying, we obtain

$$\left(A^TA + \frac{1-\lambda\eta^2}{\lambda}I\right)x = A^Tb.$$

Make the definition $(1-\lambda\eta^2)/\lambda = \alpha$. Then we have

$$x = (A^TA + \alpha I)^{-1}A^Tb.$$

Plugging this into $\|Ax-b\|^2 = \eta^2\|x\|^2$ and using the SVD of A we obtain

$$b_2^Tb_2 + b_1^T\Sigma_1^4(\Sigma_1^2+\alpha I)^{-2}b_1 - 2b_1^T\Sigma_1^2(\Sigma_1^2+\alpha I)^{-1}b_1 + b_1^Tb_1 = \eta^2 b_1^T\Sigma_1^2(\Sigma_1^2+\alpha I)^{-2}b_1.$$

Simplifying we get

$$b_2^Tb_2 + b_1^T(\Sigma_1^2+\alpha I)^{-2}(\alpha^2 I - \eta^2\Sigma_1^2)b_1 = 0.$$

Since we are interested in finding the values of α for which the right-hand side of the above equation is zero, we define the function $g(\alpha)$ as

$$g(\alpha) = b_2^Tb_2 + b_1^T(\Sigma_1^2+\alpha I)^{-2}(\alpha^2 I - \eta^2\Sigma_1^2)b_1$$

and then study the zeros of this function. The function $g(\alpha)$ is called the "secular equation," since it is rational function of one variable. If σ_i denotes the ith singular value of A, then the above secular equation has poles at $-\sigma_i^2$.

This secular equation can have up to $2n$ real zeros. One of them will give us the minimum norm solution to our problem, $x(\alpha^o)$. We note that if $\alpha > \eta\sigma_1$ in the secular equation, then we must have $b = 0$, which as we stated earlier requires $z = 0$, and thus $x = 0$. Since we are considering $b \neq 0$ we must have $\alpha \leq \eta\sigma_1$.

The secular equation, $g(\alpha)$ is given by

$$g(\alpha) = b_2^Tb_2 + b_1^T(\Sigma_1^2+\alpha I)^{-2}(\alpha^2 I - \eta^2\Sigma_1^2)b_1.$$

10. Main theorem. We claim that in all cases where a degenerate solution exists, the minimum norm solution is determined by the unique root of the secular equation in the interval $[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$.

The rest of the paper is devoted to establishing this claim. This is a difficult task due to the nonconvex nature of the problem and the presence of multiple local minima.

The solution to the problem, $\hat{x}$, is given by $\hat{x} = x(\alpha^o)$ with α^o the unique zero of

$$g(\alpha) = b_2^Tb_2 + b_1^T(\Sigma_1^2+\alpha I)^{-2}(\alpha^2 I - \eta^2\Sigma_1^2)b_1$$

in the interval $[\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$.

11. First and second order conditions. Since the Lagrange multiplier must be nonnegative at a local minimum and $\lambda = 1/(\alpha+\eta^2)$ we conclude that

$$\alpha \geq -\eta^2. \tag{11.1}$$

To narrow down the interesting zeros we look at the second order conditions for a local minimum. Our Lagrangian was

$$L(x,\lambda) = \|x\|^2 + \lambda(\|Ax-b\|^2 - \eta^2\|x\|^2).$$

The second order condition for a local minimum is that the Hessian of $L(x, \lambda)$ with respect to x be positive-semidefinite when restricted to the tangent subspace of the constraint. Differentiating once we have

$$\nabla_x L(x, \lambda) = 2x + \lambda \left(2A^T(Ax - b) - 2\eta^2 x\right).$$

Differentiating once more we get

$$\nabla_x^2 L(x, \lambda) = 2I + \lambda \left(2A^T A - 2\eta^2 I\right),$$

which on simplifying yields

$$\nabla_x^2 L(x, \lambda) = 2\lambda \left(\alpha I + A^T A\right).$$

The constraint is

$$c(x) = \|Ax - b\|^2 - \eta^2 \|x\|^2.$$

The gradient of the constraint is

$$\nabla_x c(x) = 2A^T(Ax - b) - 2\eta^2 x,$$

which can be simplified by noting that

$$A^T(Ax - b) = -\alpha x$$

thus

$$\nabla_x c(x) = -(\alpha + \eta^2)x.$$

The tangent subspace of the constraint has $n - 1$ dimensions (even when $\eta = \sigma_i$). We now construct a basis for this subspace. Using the SVD notation developed in section 7 we have

$$V^T \nabla_x c(x) = -(\alpha + \eta^2)z.$$

Similarly we can change the basis for the Hessian of the Lagrangian

$$V^T \nabla_x^2 L(x, \lambda) V = 2\lambda \left(\Sigma_1^2 + \alpha I\right).$$

We partition z as

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

where z_1 is a scalar. Let

$$H = \begin{pmatrix} z_2^T \\ -z_1 I \end{pmatrix}.$$

Then $H^T z = 0$. Therefore the restricted Hessian is

$$H^T V^T \nabla_x^2 L(x, \lambda) V H = 2\lambda \left(H^T \Sigma_1^2 H + \alpha H^T H\right).$$

We note that the second order condition requires that the restricted Hessian be positive-semidefinite, and so we can apply Cauchy's interlacing theorem. Cauchy's

interlacing theorem tells us that the smallest eigenvalue for this matrix must lie between the smallest and second smallest eigenvalues for the nonrestricted Hessian. Thus for a local minimum the second smallest eigenvalue of the nonrestricted Hessian must be greater than zero. For the condition on the second smallest eigenvalue to be met, α must satisfy the constraint $\alpha \geq -\sigma_{n-1}^2$, where σ_{n-1} is the second smallest singular value of A.

This raises the question of how many zeros of the secular equation are larger than $\max(-\eta^2, -\sigma_{n-1}^2)$ and which of them corresponds to the global minimum. We proceed by systematically eliminating zeros in this range. We have two critical points (where the secular equation becomes infinite) which correspond to $\alpha = -\sigma_{n-1}^2$ and $\alpha = -\sigma_n^2$. We also have two intervals to worry about, namely, $(-\sigma_n^2, \eta\sigma_1)$ and $(-\sigma_{n-1}^2, -\sigma_n^2)$. In the first interval we can show that there is only one zero, but this is not true for the second interval. In section 12 we use the second order condition to rule out half of the zeros in the second interval. We show in Appendix B that only the rightmost root in the second interval is actually a candidate. We are left with four candidates, two in the intervals and two critical points, and we then use six cases to prove which one corresponds to the global minimum.

$$\alpha^o > \max(-\eta^2, -\sigma_{n-1}^2).$$

12. Squeezing the second order conditions. We can use the second order conditions to discard some zeros in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$. Recall that the restricted Hessian is

$$H^T V^T \nabla_x^2 L(x,\lambda) V H = 2\lambda \left(H^T \Sigma_1^2 H + \alpha H^T H \right).$$

This can be expanded to obtain

$$H^T V^T \nabla_x^2 L(x,\lambda) V H = 2\lambda \left(\sigma_1^2 z_2 z_2^T + z_1^2 \Sigma_2^2 + \alpha z_2 z_2^T + \alpha z_1^2 I \right),$$

where

$$\Sigma_1 = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}.$$

We then make the conformal partition

$$b_1 = \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix}$$

and use the representation $z = (\Sigma_1^2 + \alpha I)^{-1} \Sigma_1 b_1$ to simplify the expansion. Additionally, we make the definition $M = H^T V^T \nabla_x^2 L(x,\lambda) V H$ for ease of reading and we can thus get the simplified expansion

$$M = 2\lambda \frac{b_{11}^2 \sigma_1^2}{(\sigma_1^2 + \alpha)^2} (\Sigma_2^2 + \alpha I) \left(I + \frac{(\sigma_1^2 + \alpha)^3}{b_{11}^2 \sigma_1^2} (\Sigma_2^2 + \alpha I)^{-2} \Sigma_2 b_{12} b_{12}^T \Sigma_2 (\Sigma_2^2 + \alpha I)^{-1} \right).$$

We now compute the determinant,

$$\det(M) = \left(\frac{2\lambda b_{11}^2 \sigma_1^2}{(\sigma_1^2 + \alpha)^2} \right)^n \det(\Sigma_2^2 + \alpha I) \left(1 + \frac{(\sigma_1^2 + \alpha)^3}{b_{11}^2 \sigma_1^2} b_{12}^T \Sigma_2^2 (\Sigma_2^2 + \alpha I)^{-3} b_{12} \right),$$

which can be further simplified to obtain

$$\det(M) = \left(\frac{2\lambda b_{11}^2\sigma_1^2}{(\sigma_1^2+\alpha)^2}\right)^n \frac{(\sigma_1^2+\alpha)^3}{b_{11}^2\sigma_1^2}\det(\Sigma_2^2+\alpha I)\left(b_1^T\Sigma_1^2(\Sigma_1^2+\alpha I)^{-3}b_1\right). \tag{12.1}$$

We recall the definition of the secular equation, $g(\alpha)$, given in section 9:

$$g(\alpha) = b_2^Tb_2 + b_1^T(\Sigma_1^2+\alpha I)^{-2}(\alpha^2 I - \eta^2\Sigma_1^2)b_1.$$

Then differentiating once we obtain

$$g'(\alpha) = 2(\alpha+\eta^2)b_1^T\Sigma_1^2(\Sigma_1^2+\alpha I)^{-3}b_1. \tag{12.2}$$

Using this we can rewrite (12.1) as

$$\det(M) = \left(\frac{2\lambda b_{11}^2\sigma_1^2}{(\sigma_1^2+\alpha)^2}\right)^n \frac{(\sigma_1^2+\alpha)^3}{2(\alpha+\eta^2)b_{11}^2\sigma_1^2}\det(\Sigma_2^2+\alpha I)g'(\alpha).$$

Therefore we see that when a root of the secular equation lies in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, then it can correspond to a local minimum only if $g'(\alpha)$ is nonpositive. **This essentially means that only half of the zeros in the interval correspond to local minima.**

A zero, α_k, of $g(\alpha)$ in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ can correspond to a local minimum of the Lagrangian (and thus have a chance of being the global minimum α^o) only if

$$g'(\alpha_k) \le 0.$$

13. Four candidate zeros. At this point we can see several potential candidates for α. First, we have the possibility of a root in the interval $[-\sigma_n^2, \eta\sigma_1]$ designated α_1. The uniqueness and conditions for existence of α_1 will be shown later. Second, we potentially have many roots in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, but only the rightmost one matters as will be shown later and it is thus designated α_2. Finally, we could have up to two critical points, $\alpha_3 = -\sigma_n^2$ and $\alpha_4 = -\sigma_{n-1}^2$. We summarize the candidates in Table 1.

TABLE 1
Candidate zeros.

$\alpha_1 \in [-\sigma_n^2, \eta\sigma_1]$
$\alpha_2 \in (-\sigma_{n-1}^2, -\sigma_n^2)$
$\alpha_3 = -\sigma_n^2$
$\alpha_4 = -\sigma_{n-1}^2$

The proof involves six cases, which cover special conditions for the problem. See Table 2. The first two cases involve small values of η. The second two cases cover

when b is orthogonal to the left singular vector(s) of the smallest singular value. The last two cases cover when b is not orthogonal to the left singular vector(s) of the smallest singular value. We now proceed to prove this and show which candidate root will yield the solution to the problem, $\hat{x}$.

TABLE 2
Six cases of the proof.

Case 1: $\eta < \sigma_n$
Case 2: $\eta = \sigma_n$
Case 3: $\eta > \sigma_n$, $b_{1,n} = 0$, $\sigma_n < \sigma_{n-1}$
Case 4: $\eta > \sigma_n$, $\|b_{1,(n-k+1,n)}\| = 0$, $\sigma_n = \sigma_{n-k+1}$
Case 5: $\eta > \sigma_n$, $b_{1,n} \neq 0$, $\sigma_n < \sigma_{n-1}$
Case 6: $\eta > \sigma_n$, $\|b_{1,n-k+1}\| \neq 0$, $\sigma_n = \sigma_{n-k+1}$

14. Case 1: $\eta < \sigma_n$. There is only one root in the interval $[-\eta^2, \eta\sigma_1]$ and this must correspond to the global minimum, as there are no other local minima to worry about. The only candidate zero is α_1 because of the first order condition, (11.1). We need to only prove the existence and uniqueness of α_1.

Since $\alpha + \eta^2 \geq 0$ from (11.1), it follows by using (12.2) that $g'(\alpha)$ is positive in the interval $(-\eta^2, \infty)$ when $\eta \leq \sigma_n$. Therefore, there can be at most one root in the interval $[-\eta^2, \eta\sigma_1]$.

We now show that there is at least one root in the interval $[-\eta^2, \eta\sigma_1]$. Simplifying the degeneracy condition in (7.3) by using the SVD of A we obtain

$$b_2^T b_2 - \eta^2 b_1^T (\Sigma_1^2 - \eta^2 I)^{-1} b_1 \leq 0,$$

which is identical to $g(-\eta^2) \leq 0$. Furthermore,

$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) > 0.$$

Therefore, there must be a zero of $g(\alpha)$ in the interval $[-\eta^2, \eta\sigma_1]$.

15. Case 2: $\eta = \sigma_n$. We claim that there is a unique root of $g(\alpha)$ in $[-\sigma_n^2, \eta\sigma_1]$, and this is the global minimum. Uniqueness is established by the same method as in section 14, and thus if a root exists in the interval $[-\sigma_n^2, \eta\sigma_1]$, it is unique. Only two candidates, the zero α_1 and the critical point α_3, are possible because of the first order condition, (11.1). We will proceed to prove the claim in two steps. Before we start

the first step we note that if σ_n is multiple with multiplicity k, then $\tilde{b}_1 = b_{1,(n-k+1:n)}$ is the partitioning of b_1 corresponding to the multiple singular values of σ_n.

The first case is when $b_{1,n} \neq 0$ or $\|\tilde{b}_1\| \neq 0$. We first note that in this case the candidate zero α_3 is not possible. To see this we first partition Σ_1 as

$$\Sigma_1 = \begin{pmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \sigma_n \end{pmatrix}.$$

We similarly partition z into $\bar{z}$ and z_n, and b_1 into $\bar{b}_1$ and $b_{1,n}$. We can use these to rewrite the Lagrange condition, $(A^TA + \alpha I)x = A^Tb$, as

$$\begin{pmatrix} \bar{\Sigma}_1^2 + \alpha_3 I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{z} \\ z_n \end{pmatrix} = \begin{pmatrix} \bar{\Sigma}_1 \bar{b}_1 \\ b_{1,n} \end{pmatrix}.$$

Since $b_{1,n} \neq 0$, we see that α_3 cannot be α^o. The existence of a root in the interval $(-\sigma_n^2, \eta\sigma_1)$ follows from the observation that

$$\lim_{\alpha \to -\sigma_n^2 +} g(\alpha) = -\infty,$$
$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

Thus when $b_{1,n} \neq 0$ or $\|\tilde{b}_1\| \neq 0$, $\alpha^o = \alpha_1$.

The second case is $b_{1,n} = 0$ when $\sigma_n < \sigma_{n-1}$ or $\|\tilde{b}_1\| = 0$ when σ_n is multiple. In this case we note that there is no longer a pole in $g(\alpha)$ at $\alpha = -\sigma_n^2$. By observing the degeneracy condition given by (7.3) that the degeneracy in this case is determined by b so for degeneracy, (7.3) must hold for a smaller problem. Simplifying the (7.3) using the SVD of A we obtain

$$b_2^T b_2 - \eta^2 b_1^T (\Sigma_1^2 - \eta^2 I)^{-1} b_1 \leq 0,$$

which is identical to $g(-\eta^2) \leq 0$. Furthermore,

$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

Therefore, there must be a root in the interval $[-\eta^2, \eta\sigma_1]$, so α_1 exists. We will show that when α_3 is α^o, then $\alpha_3 = \alpha_1$. To satisfy the equation

$$\begin{pmatrix} \bar{\Sigma}_1^2 + \alpha_3 I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{z} \\ z_n \end{pmatrix} = \begin{pmatrix} \bar{\Sigma}_1 \bar{b}_1 \\ 0 \end{pmatrix},$$

we must have that

$$\bar{z} = \left(\bar{\Sigma}_1^2 + \alpha_3 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1.$$

The constraint equation can be written in z and simplified to

$$\alpha_3 \bar{b}_1^T \left(\bar{\Sigma}_1^2 + \alpha_3 I\right)^{-1} \bar{b}_1 + b_2^T b_2 = 0.$$

We note that this is exactly $g(\alpha_3) = 0$. Thus for α_3 to be a candidate it must also be the unique root in the interval $[-\eta^2, \eta\sigma_1]$. The condition for α_3 to be α^o is that $\alpha_1 = \alpha_3$, and thus we can easily see that in all cases the unique zero which corresponds to the problem solution, $x(\alpha^o)$, is given by α_1.

16. Case 3: $\eta > \sigma_n$, $b_{1,n} = 0$, $\sigma_n < \sigma_{n-1}$. We claim that there is a unique root in $[-\sigma_n^2, \eta\sigma_1]$ and this is the global minimum. We now establish this claim. Two cases arise when $b_{1,n} = 0$ by observing the equation

$$\begin{pmatrix} \bar{\Sigma}_1^2 + \alpha I & 0 \\ 0 & \sigma_n^2 + \alpha \end{pmatrix} \begin{pmatrix} \bar{z} \\ z_n \end{pmatrix} = \begin{pmatrix} \bar{\Sigma}_1 \bar{b}_1 \\ 0 \end{pmatrix}. \tag{16.1}$$

First, we could have $\alpha = \alpha_3 = -\sigma_n^2$, which we note can only happen when $b_{1,n} = 0$. The second case is $z_n = 0$. First, we note that we still have

$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

We also know that $g'(\alpha) > 0$ on the interval $(-\sigma_{n-1}^2, \infty)$, thus if a root exists, it is unique. We will start by finding the form of the solution $\hat{x}$ when $\alpha = \alpha_3$ and then we will show the conditions for determining which candidate zero yields the global minimum.

When $\alpha = -\sigma_n^2$ the solution is found in two steps. First we solve for $\bar{z}$ from (16.1). We obtain

$$\bar{z} = \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1.$$

We note that the constraint can be written in z as

$$\left\| \begin{matrix} \Sigma_1 z - b_1 \\ b_2 \end{matrix} \right\|^2 - \eta^2 \|z\|^2 = 0.$$

We now separate z_n in the constraint and obtain

$$\bar{b}_1^T \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-2} \left(\sigma_n^4 I - \eta^2 \bar{\Sigma}_1^2\right) \bar{b}_1 + b_2^T b_2 + \left(\sigma_n^2 - \eta^2\right) z_n^2 = 0.$$

We note that this can be rewritten in terms of $g(-\sigma_n^2)$ as

$$g\left(-\sigma_n^2\right) + \left(\sigma_n^2 - \eta^2\right) z_n^2 = 0.$$

We thus see there are two answers (positive and negative squares) for z_n. The answers for z_n are given by

$$z_n^2 = \frac{g\left(-\sigma_n^2\right)}{\eta^2 - \sigma_n^2}. \tag{16.2}$$

Note that for a solution for z_n to exist we must have $g(-\sigma_n^2) \geq 0$. The solution is then given by

$$\hat{x} = V \begin{bmatrix} \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1 \\ \pm\sqrt{\frac{g(-\sigma_n^2)}{\eta^2 - \sigma_n^2}} \end{bmatrix}.$$

We still need to identify which of the potential roots is the actual one we want. We break this into two steps. The first is when $g(-\sigma_n^2) \leq 0$, and the second is $g(-\sigma_n^2) > 0$. If $g(-\sigma_n^2) \leq 0$, then we trivially have a unique root in $[-\sigma_n^2, \eta\sigma_1]$. Moreover, no root exists in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ so α_2 is not a candidate. We note that for $\alpha_4 = -\sigma_{n-1}^2$ to be a candidate, it must be true that $b_{1,n-1} = 0$. When $b_{1,n-1} = 0$, we have $g'(\alpha) > 0$ on the interval $(-\sigma_{n-2}^2, \infty)$, which means $g(-\sigma_{n-1}^2) < 0$. If we assume

$\alpha = \alpha_4$ and proceed similarly to section 16 we see that we must have $g(-\sigma_{n-1}^2) \geq 0$ and thus α_4 cannot be α^o. Note that when $g(-\sigma_n^2) < 0$, it is impossible for $\alpha = -\sigma_n^2$. When $g(-\sigma_n^2) = 0$, the unique root is $\alpha = -\sigma_n^2$ and thus the two remaining candidate zeros can easily be seen to coincide. Thus when $g(-\sigma_n^2) \leq 0$, the unique zero is given by α_1.

When $g(-\sigma_n^2) > 0$ no root exists in $(-\sigma_n^2, \eta\sigma_1]$ so α_1 is not α^o but as we saw in section 16 this is the condition for $\alpha = \alpha_3 = -\sigma_n^2$. We note that when $g(-\sigma_n^2) > 0$, there can be a root in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, but we know that the slope is positive in this interval and by the results of section 12 it cannot be a minimum. The only remaining question in this case is if $\alpha_4 = -\sigma_{n-1}^2$ is a candidate when $g(-\sigma_n^2) > 0$. We again recall that for $-\sigma_{n-1}^2$ to be a candidate, it must be true that $b_{1,n-1} = 0$ and $g(-\sigma_{n-1}^2) \geq 0$. We must thus satisfy the equation

$$\begin{pmatrix} \tilde{\Sigma}_1^2 + \alpha I & 0 & 0 \\ 0 & \sigma_{n-1}^2 + \alpha & 0 \\ 0 & 0 & \sigma_n^2 + \alpha \end{pmatrix} \begin{pmatrix} \tilde{z} \\ z_{n-1} \\ z_n \end{pmatrix} = \begin{pmatrix} \tilde{\Sigma}_1 \tilde{b}_1 \\ 0 \\ 0 \end{pmatrix}. \tag{16.3}$$

We proceed to show that $-\sigma_{n-1}^2$ is not a candidate when $g(-\sigma_{n-1}^2) \geq 0$. We note that since $b_{1,n-1} = 0 = b_{1,n}$ we must have $g'(\alpha) > 0$ on the interval $(-\sigma_{n-2}^2, \infty)$. Now introduce the parameter $\gamma = \|b_{1,n-1}\|^2$ and we will consider a continuity argument on γ similar to the continuity argument we will consider in section 18. Since the argument is very similar to the one we will be constructing, we will only sketch the details here. Note that for $\gamma \neq 0$ we have a root in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ which is not the global minimum. As γ goes to zero we make this root move to the left, and it reaches $-\sigma_{n-1}^2$ when $\gamma = 0$, since $g(-\sigma_{n-1}^2) \geq 0$. The derivative of the cost with respect to γ can be seen to be negative in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ by the following method. First, take the derivative and note that there appears the term $d\alpha(\gamma)/d\gamma$, which we solve for by taking the derivative of $g(\alpha(\gamma)) = 0$ with respect to γ. Substituting back in and simplifying we see that as γ increases, the cost decreases in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ and thus the x corresponding to the root which appears in the interval when $\gamma \neq 0$ has a lower cost than the x which corresponds to $-\sigma_{n-1}^2$. The root is not a global minimum, however, and so neither can be α^o at $-\sigma_{n-1}^2$. The only possibility when $g(-\sigma_n^2) \geq 0$ is thus $\alpha^o = -\sigma_n^2$.

17. Case 4: $\eta > \sigma_n$, $\|b_{1,(n-k+1,n)}\| = 0$, $\sigma_n = \sigma_{n-k+1}$. We claim that there is a unique root in $[-\sigma_n^2, \eta\sigma_1]$, and this is the global minimum. We now establish this claim. For simplicity partition Σ_1 as

$$\Sigma_1 = \begin{pmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \sigma_n I \end{pmatrix},$$

where $\bar{\Sigma}_1$ corresponds to the singular values that are strictly greater than σ_n. We similarly partition z into $\bar{z}$ and $\tilde{z}$, and b_1 into $\bar{b}_1$ and $\tilde{b}_1$. Two cases arise when $\tilde{b}_1 = 0$ by observing the equation

$$\begin{pmatrix} \bar{\Sigma}_1^2 + \alpha I & 0 \\ 0 & (\sigma_n^2 + \alpha) I \end{pmatrix} \begin{pmatrix} \bar{z} \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} \bar{\Sigma}_1 \bar{b}_1 \\ 0 \end{pmatrix}. \tag{17.1}$$

First we could have $\alpha = -\sigma_n^2$, which we note can only happen when $b_{1,n} = 0$. The second case is $\tilde{z} = 0$. First we note that we still have

$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

We also know that $g'(\alpha) > 0$ on the interval $(-\sigma_{n-1}^2, \infty)$, thus if a root exists, it is unique.

When $\alpha = -\sigma_n^2$, the solution is found in two steps. First we solve for $\bar{z}$ from (17.1). We obtain

$$\bar{z} = \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1.$$

We note that the constraint can be written in z as

$$\left\| \begin{matrix} \Sigma_1 z - b_1 \\ b_2 \end{matrix} \right\|^2 - \eta^2 \|z\|^2 = 0.$$

We now separate $\tilde{z}$ in the constraint and obtain

$$\bar{b}_1^T \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-2} \left(\sigma_n^4 I - \eta^2 \bar{\Sigma}_1^2\right) \bar{b}_1 + b_2^T b_2 + \left(\sigma_n^2 - \eta^2\right) \tilde{z}^T \tilde{z} = 0.$$

Similar to what we saw in the last section, we note that the above equation can be written in terms of $g(-\sigma_n^2)$. Doing so, we obtain

$$g\left(-\sigma_n^2\right) + \left(\sigma_n^2 - \eta^2\right) \tilde{z}^T \tilde{z} = 0.$$

We note that this defines a hypersphere with radius

$$r = \sqrt{\frac{g\left(-\sigma_n^2\right)}{\eta^2 - \sigma_n^2}}.$$

To be able to solve for the radius we must have $g(-\sigma_n^2) \geq 0$, and thus this is a condition on the solution when $\alpha = -\sigma_n^2$. Let Θ be any vector with unit Euclidean norm. The solutions for $\tilde{z}$ are given by

$$\tilde{z} = r\Theta.$$

The solution is then given by

$$\hat{x} = V \begin{bmatrix} \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1 \\ r\Theta \end{bmatrix}.$$

We note that the second order condition requires that $\alpha \leq -\sigma_n^2$ and thus the only candidates are α_1 and α_3. If $g(-\sigma_n^2) \leq 0$, then we trivially have a unique root in $[-\sigma_n^2, \eta\sigma_1]$, and it is impossible for $\alpha = -\sigma_n^2$. If $g(-\sigma_n^2) > 0$, no root exists in $(-\sigma_n^2, \eta\sigma_1]$ but as we saw above this is the condition for $\alpha = -\sigma_n^2$. When $g(-\sigma_n^2) = 0$ the two zeros can easily be seen to coincide.

18. Case 5: $\eta > \sigma_n$, $b_{1,n} \neq 0$, $\sigma_n < \sigma_{n-1}$. We now claim that there is a unique root in $(-\sigma_n^2, \eta\sigma_1]$ and it is the global minimum. We note first that since $b_{1,n} \neq 0$, we cannot have $\alpha = -\sigma_n^2$.

The existence of a root in the interval $[-\sigma_n^2, \eta\sigma_1]$ follows from the observation that

$$\lim_{\alpha \to -\sigma_n^2 +} g(\alpha) = -\infty,$$
$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

Uniqueness is established by the same method as in section 14.

We now proceed to show that of the three candidate roots only the one in the interval $(-\sigma_n^2, \eta\sigma_1]$ can be the global minimum. The argument proceeds by continuation on $\beta = b_{1,n}^2$. We begin by defining

$$\bar{g}(\alpha) = b_2^T b_2 + \bar{b}_1^T \left(\bar{\Sigma}_1^2 + \alpha I\right)^{-2} \left(\alpha^2 I - \eta^2 \bar{\Sigma}_1^2\right) \bar{b}_1.$$

We can thus rewrite the secular equation $g(\alpha)$ in terms of α and β as

$$g(\alpha, \beta) = \bar{g}(\alpha) + \beta \frac{\alpha^2 - \eta^2\sigma_n^2}{(\sigma_n^2 + \alpha)^2}.$$

We note that when $\beta = 0$, we have $g(\alpha, 0) = \bar{g}(\alpha)$. Also note that $\bar{g}'(\alpha, 0) > 0$ when α lies in the interval $(\max(-\sigma_{n-1}^2, -\eta^2), \infty)$. Let $\alpha_1(\beta)$ denote the unique root in the interval $(-\sigma_n^2, \eta\sigma_1]$ and $\alpha_2(\beta)$ denote the rightmost root in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ of $g(\alpha, \beta)$. Also let $y_1(\beta)$ denote the stationary point $V^T\hat{x}$ corresponding to $\alpha_1(\beta)$ and similarly for $y_2(\beta)$ corresponding to $\alpha_2(\beta)$.

When $\bar{g}(-\sigma_n^2) < 0$, we note that neither $\alpha_1(\beta)$ nor $\alpha_2(\beta)$ converge to $-\sigma_n^2$ as β goes to zero. As already observed, at $\beta = 0$ we have that $g'(\alpha, 0) > 0$ when α lies in the interval $(\max(-\sigma_{n-1}^2, -\eta^2), \infty)$, and since $\bar{g}(-\sigma_n^2) < 0$ this implies that $g'(\alpha, 0) > 0$ when α lies in the interval $(\max(-\sigma_{n-1}^2, -\eta^2), -\sigma_n^2)$. Thus we know that $y_2(\beta)$ does not exist at $\beta = 0$ and thus it must not exist for some open neighborhood around $\beta = 0$. For $y_2(\beta)$ to be a candidate there must exist some value of β, say, β_2, for which $y_2(\beta)$ first exists. At the point β_2, $\alpha_2(\beta_2)$ must be at least a double root, and thus the slope of $g(\alpha_2(\beta))$ must be zero at β_2. From section 12, we note that $\alpha_2(\beta_2)$ cannot be the α^o, so we note that we must have $\|y_2(\beta_2)\|^2 \geq \|y_1(\beta_2)\|^2$.

We now proceed with the case when $\bar{g}(-\sigma_n^2) \geq 0$, and we will then show that in both cases $\|y_2(\beta)\|^2$ gets larger as β increases, while $\|y_1(\beta)\|^2$ decreases. It is easy to note from the form of $g(\alpha)$ that

$$\lim_{\beta \to 0+} \alpha_1(\beta) = -\sigma_n^2 = \lim_{\beta \to 0+} \alpha_2(\beta)$$

when $\bar{g}(-\sigma_n^2) \geq 0$. We now proceed to show that

$$\lim_{\beta \to 0+} |y_{1,i}(\beta)| = |y_{1,i}(0)| = |y_{2,i}(0)| = \lim_{\beta \to 0+} |y_{2,i}(\beta)|, \quad 1 \leq i \leq n.$$

First observe that this is trivially true for $i \neq n$. Next we note that $\bar{g}(\alpha)$ is continuous at $\alpha = -\sigma_n^2$; thus

$$\begin{aligned}
\lim_{\beta \to 0+} \left(y_{2,n}(\beta)^2 - y_{1,n}(\beta)^2\right) = \lim_{\beta \to 0+} \Bigg(& \frac{\sigma_n^2}{\alpha_2(\beta)^2 - \eta^2\sigma_n^2} \frac{\alpha_2(\beta)^2 - \eta^2\sigma_n^2}{\left(\alpha(\beta) + \sigma_n^2\right)^2} \beta \\
& - \frac{\sigma_n^2}{\alpha_1(\beta)^2 - \eta^2\sigma_n^2} \frac{\alpha_1(\beta)^2 - \eta^2\sigma_n^2}{\left(\alpha(\beta) + \sigma_n^2\right)^2} \beta \Bigg) \\
= \frac{1}{\sigma_n^2 - \eta^2} & \\
& \lim_{\beta \to 0+} \Bigg(\frac{\alpha_2(\beta)^2 - \eta^2\sigma_n^2}{\left(\alpha_2(\beta) + \sigma_n^2\right)^2} \beta + \bar{g}(\alpha_2(\beta), \beta) \\
& \qquad - \frac{\alpha_1(\beta)^2 - \eta^2\sigma_n^2}{\left(\alpha_1(\beta) + \sigma_n^2\right)^2} \beta - \bar{g}(\alpha_1(\beta), \beta) \Bigg)
\end{aligned}$$

$$= \frac{1}{\sigma_n^2 - \eta^2} \lim_{\beta \to 0+} \left(g(\alpha_2(\beta), \beta) - g(\alpha_1(\beta), \beta) \right)$$
$$= 0.$$

We note that we have shown what we desired to and therefore, $\|y_1(\beta)\|$ and $\|y_2(\beta)\|$ are continuous for $\beta \geq 0$, with $\|y_1(0)\| = \|y_2(0)\|$.

We now examine the derivative of the cost function, $\|x\|^2$, with respect to β. We will use this to show that in both cases $\|y_1(\beta)\|$ is less than $\|y_2(\beta)\|$ for all $\beta \geq 0$. The derivative is

$$\frac{d \left\| x(\alpha(\beta)) \right\|^2}{d\beta} = \frac{\sigma_n^2}{\left(\sigma_n^2 + \alpha(\beta) \right)^2} - 2 \frac{d\alpha(\beta)}{d\beta} b_1^T \left(\Sigma_1^2 + \alpha(\beta) I \right)^{-3} \Sigma_1^2 b_1.$$

We need to calculate the derivative of $\alpha(\beta)$ with respect to β, so we take the derivative of $g(\alpha(\beta)) = 0$:

$$0 = \frac{dg(\alpha(\beta))}{d\beta}$$
$$= \frac{\alpha(\beta)^2 - \eta^2 \sigma_n^2}{\left(\alpha(\beta) + \sigma_n^2 \right)^2} + 2 \left(\alpha(\beta) + \eta^2 \right) \frac{d\alpha(\beta)}{d\beta} \left(b_1^T \left(\Sigma_1^2 + \alpha(\beta) I \right)^{-3} \Sigma_1^2 b_1 \right).$$

Solving for the derivative of $\alpha(\beta)$ with respect to β yields

$$\frac{d\alpha(\beta)}{d\beta} = - \frac{\alpha(\beta)^2 - \eta^2 \sigma_n^2}{2 \left(\alpha(\beta) + \eta^2 \right) \left(\sigma_n^2 + \alpha(\beta) \right)^2 \left(b_1^T \left(\Sigma_1^2 + \alpha(\beta) I \right)^{-3} \Sigma_1^2 b_1 \right)}.$$

Substituting this into the derivative of $\|x\|^2$ with respect to β we obtain

$$\frac{d \left\| x(\alpha(\beta)) \right\|^2}{d\beta} = \frac{\sigma_n^2}{\left(\sigma_n^2 + \alpha(\beta) \right)^2} + \frac{\alpha(\beta)^2 - \eta^2 \sigma_n^2}{\left(\alpha(\beta) + \eta^2 \right) \left(\sigma_n^2 + \alpha(\beta) \right)^2}.$$

Simplifying this we get

$$\frac{d \left\| x(\alpha(\beta)) \right\|^2}{d\beta} = \frac{\alpha(\beta)}{\left(\alpha(\beta) + \eta^2 \right) \left(\alpha(\beta) + \sigma_n^2 \right)}.$$

Clearly, for increasing β we have that $\alpha_1(\beta)$ decreases the cost function when $\alpha_1(\beta) < 0$, while $\alpha_2(\beta)$ increases the cost function for all β. When $0 \leq \alpha_1(\beta) \leq \eta\sigma_n$ we have that $d\alpha(\beta)/d\beta \geq 0$ and we note that the cost is increasing for both $y_1(\beta)$ and $y_2(\beta)$. Since the cost is increasing for $y_1(\beta)$ when $0 \leq \alpha_1(\beta) \leq \eta\sigma_n$, we know that $\|y_1(\beta)\|^2 \leq \|y_1(\eta\sigma_n)\|^2$ on this interval. Additionally, note that for $\alpha_1(\beta)$ in the interval $[\eta\sigma_n, \eta\sigma_1]$ we have $d\alpha(\beta)/d\beta \leq 0$ and the cost increases with increasing β. Note that while these observations are true for $[\eta\sigma_n, \infty]$, we specify the interval $[\eta\sigma_n, \eta\sigma_1]$ because the root cannot lie in $[\eta\sigma_1, \infty]$. Observe that we now have $\|y_1(\beta)\|^2 \leq \|y_1(\eta\sigma_n)\|^2$ when $\alpha_1(\beta)$ is in the interval $[\eta\sigma_n, \eta\sigma_1]$. Thus the maximum value of the cost, when $\alpha_1(\beta)$ is in the interval $[\eta\sigma_n, \eta\sigma_1]$, occurs at $\beta = \eta\sigma_n$. We can easily find the maximum rate of change for the cost, when $\alpha_1(\beta)$ is in the interval $[0, \eta\sigma_1]$, to be

$$\max \frac{d \left\| x(\alpha(\beta)) \right\|^2}{d\beta} = \frac{\eta\sigma_n}{\left(\eta\sigma_n + \eta^2 \right) \left(\eta\sigma_n + \sigma_n^2 \right)}.$$

Simplifying we obtain

$$\max \frac{d\left\|x(\alpha(\beta))\right\|^2}{d\beta} = \frac{1}{(\eta+\sigma_n)^2}.$$

We can do similar calculation for the interval $(\max(-\eta^2, -\sigma_{n-1}^2), -\sigma_{n-1}^2)$ and we find that the minimum increase in the cost occurs at $\beta = -\eta\sigma_n$ and is given by

$$\min \frac{d\left\|x(\alpha(\beta))\right\|^2}{d\beta} = \frac{1}{(\eta-\sigma_n)^2}.$$

We now note that the maximum rate of increase for $y_1(\beta)$ is less than the minimum rate of increase for $y_2(\beta)$, and for β sufficiently small, we have $\|y_1(\beta)\| \leq \|y_2(\beta)\|$. We can now easily see that $\|y_1(\beta)\| \leq \|y_2(\beta)\|$ for all β; thus α_2 cannot be the global minimum.

We now consider the third candidate zero, namely, $-\sigma_{n-1}^2$. We note that for it to be a candidate we must have that $b_{1,n-1} = 0$ and $g(-\sigma_{n-1}^2) \geq 0$. We observe that similar to what we saw in Appendix B, the minimum on the interval $(-\sigma_{n-2}^2, -\sigma_n^2)$ must occur between the second to the rightmost and the rightmost roots of the secular equation on the interval. Recall in that section the only options were the roots themselves, but in this case there is also the possibility of $-\sigma_{n-1}^2$. Note that if $-\sigma_{n-1}^2$ is not one of the two rightmost roots on the interval $(-\sigma_{n-2}^2, -\sigma_n^2)$, then it cannot be the global minimum. We already know the rightmost root, designated α_2, is not the global minimum, and additionally the second most right root cannot be the global minimum since the slope of $g(\alpha)$, is not negative at this point.

We now reintroduce the parameter $\gamma = \|b_{1,n-1}\|^2$ and we will consider a continuity argument on γ similar to the continuity argument presented in this section. Since the argument is very similar to the one we constructed, we will again only sketch the details here. Note that for $\gamma \neq 0$ we have multiple roots in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, none of which are the global minimum. As γ goes to zero we make all of the roots move to the left, and all but the rightmost either reaches $-\sigma_{n-1}^2$ or pops off the real line as $\gamma \to 0$, since $g(-\sigma_{n-1}^2) \geq 0$. The derivative of the cost with respect to γ can be seen to be negative in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ by the following method. First take the derivative and note there appears the term $d\alpha(\gamma)/d\gamma$, which we solve for by taking the derivative of $g(\alpha(\gamma)) = 0$ with respect to γ. Substituting back in and simplifying we see that as γ increases the cost decreases in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ and thus the x which corresponds to the root which appears in the interval when $\gamma \neq 0$ has a lower cost than the x which corresponds to $-\sigma_{n-1}^2$. That root is not a global minimum, however, and so neither can the root be at $-\sigma_{n-1}^2$. We can thus exclude the possibility that $-\sigma_{n-1}^2$ is α^o, and we are done.

19. Case 6: $\eta > \sigma_n$, $\|b_{1,n-k+1}\| \neq 0$, $\sigma_n = \sigma_{n-k+1}$. We again claim that there is a unique root, α_1, in the interval $(-\sigma_n^2, \eta\sigma_1]$ and it is the global minimum.

The existence of a root , α_1, in the interval $(-\sigma_n^2, \eta\sigma_1]$ follows from the observation that

$$\lim_{\alpha \to -\sigma_n^2 +} g(\alpha) = -\infty,$$
$$\lim_{\alpha \to \eta\sigma_1} g(\alpha) \geq 0.$$

Uniqueness is established by the same method as in section 14.

TABLE 3
Degeneracy conditions.

$\eta < \sigma_n$ and $b^T(I - A(A^TA - \eta^2 I)^{-1}A^T)b \leq 0$
$\eta = \sigma_n$, $b_{1,n} = 0$, and $\bar{b}_1^T(I - \bar{\Sigma}_1^2(\bar{\Sigma}_1 - \eta^2 I)^{-1})\bar{b}_1 \leq 0$
$\eta = \sigma_n$, $b_{1,n} \neq 0$
$\eta > \sigma_n$

Since $\|b_{1,n-k+1}\| \neq 0$, we cannot have $\alpha = \alpha_3 = -\sigma_n^2$. Note that the second order condition gives us the additional requirement that $\alpha \geq -\sigma_n^2$. Since $\alpha \geq -\sigma_n^2$ then trivially we do not have additional roots to worry about. The only candidate is thus the unique root, α_1, in the interval $(-\sigma_n^2, \eta\sigma_1]$.

20. Summary of results. The problem we have been considering is

$$\min_{x \in \mathcal{R}^n} \min_{\|E\| \leq \eta} \|(A + E)x - b\|,$$

where A is an $m \times n$ real matrix and b is an n-dimensional real column vector. We assume that the problem is degenerate and in particular that there exists an x such that $\eta\|x\| \geq \|Ax - b\|$. Degeneracy can be easily checked as outlined in Table 3. To obtain a solution to the degenerate problem we consider the optimization problem

$$\min_{\|Ax-b\| \leq \eta\|x\|} \|x\|.$$

The SVD of A is given by

$$A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T,$$

and we define $b_1 = U_1^T b$ and $b_2 = U_2^T b$. When $b_{1,n} = 0$ if σ_n is unique or $\|b_{1,n-k+1,n}\| = 0$ if σ_n is of multiplicity k, we can partition Σ_1 as

$$\Sigma_1 = \begin{pmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \sigma_n I \end{pmatrix}.$$

We similarly partition b_1 into $\bar{b}_1$ and $b_{1,n} = 0$. The secular equation is given by

$$g(\alpha) = b_2^T b_2 + b_1^T(\Sigma_1^2 + \alpha I)^{-2}(\alpha^2 I - \eta^2 \Sigma_1^2)b_1.$$

Given these definitions, the solution to the problem is given in Table 4. Note that to find the unique root of the secular equation, $g(\alpha)$, in the interval specified can be easily and quickly done by a method such as bisection or Newton's method.

TABLE 4
Solution to the problem.

Condition	Solution
$\eta > \sigma_n$, $\sigma_n < \sigma_{n-1}$, $b_{1,n} = 0$, $g(-\sigma_n^2) \geq 0$	$x = V \begin{bmatrix} \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1 \\ \pm\sqrt{\frac{g(-\sigma_n^2)}{\eta^2 - \sigma_n^2}} \end{bmatrix}$
$\eta > \sigma_n$, $\sigma_n = \sigma_{n-k+1}$, $\|b_{1,(n-k+1,n)}\| = 0$, $g(-\sigma_n^2) \geq 0$	$\hat{x} = V \begin{bmatrix} \left(\bar{\Sigma}_1^2 - \sigma_n^2 I\right)^{-1} \bar{\Sigma}_1 \bar{b}_1 \\ r\Theta \end{bmatrix}$ $r = \sqrt{\frac{g(-\sigma_n^2)}{\eta^2 - \sigma_n^2}}$ $\|\Theta\| = 1$
else	$x = (A^T A + \alpha I)^\dagger A^T b$ $\alpha_1 \in [\max(-\sigma_n^2, -\eta^2), \eta\sigma_1]$ such that $g(\alpha_1) = 0$

21. Restricted perturbations. We have so far considered the case in which all the columns of the A matrix are subject to perturbations. It may happen in practice, however, that only selected columns are uncertain, while the remaining columns are known precisely. This situation can be handled by the approach of this paper as we now clarify.

Given $A \in \Re^{m \times n}$, we partition it into block columns,

$$A = [\, A_1 \quad A_2 \,],$$

and assume, without loss of generality, that only the columns of A_2 are subject to perturbations while the columns of A_1 are known exactly. We then pose the following problem:

Given $A \in \Re^{m \times n}$, with $m \geq n$ and A full rank, $b \in \Re^m$, and nonnegative real number η_2, determine $\hat{x}$ such that

$$\min_{\hat{x}} \min_{\|\delta A_2\| \leq \eta_2} \{\|[\, A_1 \quad A_2 + \delta A_2 \,]\, \hat{x} - b\|\}. \tag{21.1}$$

If we partition $\hat{x}$ accordingly with A_1 and A_2, say,

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix},$$

then we can write

$$\|[\, A_1 \quad A_2 + \delta A_2 \,]\, \hat{x} - b\| = \|A\hat{x} - b + \delta A_2 \hat{x}_2\|.$$

Assuming the fundamental condition for this case, which is

$$\eta_2 \|\hat{x}_2\| \geq \|Ax - b\|,$$

and following the development of section 6 we conclude the problem is equivalent to

$$\min_{\|Ax-b\|^2 = \eta_2^2 \|x_2\|^2} \|x\|^2.$$

We note that we can rewrite the constraint as

$$\|Ax - b\|^2 + \eta_2^2\|x_1\|^2 = \eta_2^2\|x_2\|^2 + \eta_2^2\|x_1\|^2,$$

which becomes

$$\left\| \begin{bmatrix} A_1 & A_2 \\ \eta_2 I & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2 = \eta_2^2\|x\|^2.$$

We now define the following:

$$\tilde{A} = \begin{bmatrix} A_1 & A_2 \\ \eta_2 I & 0 \end{bmatrix}$$

and

$$\tilde{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

The problem thus becomes

$$\min_{\|\tilde{A}x - \tilde{b}\|^2 = \eta_2^2\|x\|^2} \|x\|^2,$$

which is easily seen to be of the same form as our original problem, though of slightly larger dimension. This can thus be solved by the method discussed earlier in this paper.

Appendix A. Piecewise convexity of $\|x(\alpha)\|$.

We now show that $\|x(\alpha)\|^2$ is strictly convex in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, which will allow us to show that only the zero closest to $-\sigma_n^2$ can correspond to a potential candidate for the global minimum.

We have that

$$\|x(\alpha)\|^2 = b_1^T \Sigma_1^2 (\Sigma_1^2 + \alpha I)^{-2} b_1.$$

Differentiating once with respect to α we get

$$\frac{d}{d\alpha}\|x(\alpha)\|^2 = -2b_1^T \Sigma_1^2 (\Sigma_1^2 + \alpha I)^{-3} b_1.$$

Differentiating once more we get

$$\frac{d^2}{d\alpha^2}\|x(\alpha)\|^2 = 6b_1^T \Sigma_1^2 (\Sigma_1^2 + \alpha I)^{-4} b_1,$$

from which we can conclude that $\|x(\alpha)\|^2$ is strictly convex on the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ and hence that it has a unique minimum on that interval.

Appendix B. Rightmost root.

We now show that of all the roots in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$ only the rightmost one can possibly correspond to the global minimum.

Let $\alpha_0, \ldots, \alpha_l$ denote the zeros of the secular equation $g(\alpha)$ in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$, in increasing order; that is,

$$-\sigma_{n-1}^2 < \alpha_0 < \alpha_1 < \cdots < \alpha_l < -\sigma_n^2.$$

From the result in section 12 we know that only the roots corresponding to negative slopes of the secular equation can correspond to local minima. Since

$$\lim_{\alpha \to -\sigma_n^2 -} g(\alpha) = -\infty,$$

it follows that

$$g'(\alpha_l) < 0 \quad \text{and} \quad g'(\alpha_{l-1}) > 0.$$

(We ignore the degenerate multiple root cases for now as the argument can be extended to them by continuity.)

Now there are two possibilities. Either $\|x(\alpha_l)\| \leq \|x(\alpha_{l-1})\|$ or not. The first case implies that $\|x(\alpha_{i+1})\| < \|x(\alpha_i)\|$ due to the convexity of $\|x(\alpha)\|$ on $(-\sigma_{n-1}^2, -\sigma_n^2)$.

For the second case we have that $\|x(\alpha_{l-1})\| < \|x(\alpha_l)\|$. We need to show this implies $\|x(\alpha_{l-1})\| < \|x(\alpha)\|$ for $-\sigma_{n-1}^2 < \alpha < -\alpha_{l-1}$, and that this is not the global minimum. Toward this end we take the derivative of $x(\alpha)$ with respect to α and get

$$\frac{dx(\alpha)}{d\alpha} = -\left(A^T A + \alpha I\right)^{-1} x(\alpha).$$

We have already shown that $\|x(\alpha)\|$ is convex on this interval, and thus it suffices to find if the derivative of $\|x(\alpha)\|^2$ with respect to α is negative at α_{l-1}, which shows that $x(\alpha)$ is then decreasing. We note that the derivative of $\|x(\alpha)\|^2$ is obtained by premultiplying the derivative of $x(\alpha)$ by $x(\alpha)^T$. To do the analysis we use the SVD of A and thus have

$$\frac{d\|x(\alpha)\|^2}{d\alpha} = -b_1^T \Sigma_1^2 \left(\Sigma_1^2 + \alpha I\right)^{-3} b_1.$$

We note that the matrix in parenthesis is indefinite and thus we must determine if the expression is negative or not at $\alpha = \alpha_{l-1}$. To do this we consider another function whose derivative we have already examined. Consider the constraint function, $\|Ax - b\|^2 - \eta^2 \|x\|^2$, and since at $\alpha = \alpha_{l-1}$ we are entering the infeasible region for increasing α, the derivative of the constraint must be positive. This condition can be expressed as

$$2(\alpha_{l-1} + \eta^2) b_1^T \Sigma_1^2 \left(\Sigma_1^2 + \alpha I\right)^{-3} b_1 > 0.$$

We note that $2(\alpha_{l-1} + \eta^2) > 0$, thus the condition is

$$b_1^T \Sigma_1^2 \left(\Sigma_1^2 + \alpha I\right)^{-3} b_1 > 0.$$

This trivially gives us

$$\frac{d\|x(\alpha)\|^2}{d\alpha} < 0,$$

and thus $x(\alpha)$ must be decreasing at $\alpha = \alpha_{l-1}$ for increasing α. Applying convexity to this result gives us $\|x(\alpha_{l-1})\| < \|x(\alpha)\|$ for $-\sigma_{n-1}^2 < \alpha < -\alpha_{l-1}$, and thus the minimum feasible value for $x(\alpha)$ on $-\sigma_{n-1}^2 < \alpha < -\sigma_n^2$ is $x(\alpha_{l-1})$.

Now since $x(\alpha_{l-1})$ does not correspond to a local minimum it follows that there is a neighborhood of $x(\alpha_{l-1})$ of the constraint surface such that in this neighborhood we have $\|x\| < \|x(\alpha_{l-1})\|$. Thus since $x(\alpha_{l-1})$ does not correspond to a local minimum, we can discard it from further consideration, since it is not the global minimum. Either way we are down to only the rightmost in the interval $(-\sigma_{n-1}^2, -\sigma_n^2)$.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.

[5] B. Walden, R. Karlson, and J. Sun, *Optimal backward perturbation bounds for the linear least squares problem*, Numer. Linear Algebra Appl., 2 (1995), pp. 271–286.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 167–186
© 2001 Society for Industrial and Applied Mathematics

SYMMETRIC PARAUNITARY MATRIX EXTENSION AND PARAMETRIZATION OF SYMMETRIC ORTHOGONAL MULTIFILTER BANKS*

QINGTANG JIANG†

Abstract. This paper is devoted to a study of symmetric paraunitary matrix extensions. The problem for a given compactly supported orthonormal scaling vector with some symmetric property, to construct a corresponding multiwavelet which also has the symmetric property, is equivalent to the symmetric paraunitary extension of a given matrix. In this paper we study symmetric paraunitary extensions of two types of matrices which correspond to two different cases for the symmetry of the scaling vector: the components of the scaling vector have or don't have the same symmetric center. In this paper we also discuss parametrizations of symmetric orthogonal multifilter banks.

Key words. symmetric extension, paraunitary, parametrization, factorization, orthogonality, symmetry, multifilter bank, scaling vector, multiwavelet

AMS subject classifications. 42C40, 15A23, 13B25, 94A11

PII. S0895479800372924

1. Introduction.

Unlike one-dimensional scalar filters, the matrix filter for a multiwavelet of $L_2(\mathbb{R})$ cannot in general be given in terms of the matrix filter for the scaling vector (except in some special cases). So for a given $r \times r$ ($r \geq 2$) finite impulse response (FIR) matrix filter $H(z) = \sum_{k\in\mathbb{Z}} h_k z^{-k}$ (called a low-pass filter) for a compactly supported orthonormal (o.n.) scaling vector $\phi = (\phi_1, \ldots, \phi_r)^T$, one needs an algorithm to construct another $r \times r$ FIR matrix filter $G(z) = \sum_{k\in\mathbb{Z}} g_k z^{-k}$ (called a high-pass filter) such that

$$H(z)G(z)^* + H(-z)G(-z)^* = 0_r, \quad G(z)G(z)^* + G(-z)G(-z)^* = I_r \tag{1.1}$$

for all $z = e^{i\omega}$. With such a filter G, the vector $\psi = (\psi_1, \ldots, \psi_r)^T$ defined by

$$\widehat{\psi}(\omega) := G(e^{\frac{i\omega}{2}})\widehat{\phi}\left(\frac{\omega}{2}\right) \tag{1.2}$$

is a compactly supported multiwavelet, i.e., the collection $\{2^{\frac{j}{2}}\psi_\ell(2^j x - k), 1 \leq \ell \leq r, j, k \in \mathbb{Z}\}$ forms an o.n. basis of $L_2(\mathbb{R})$ (see [3]). We call a vector of functions $\phi = (\phi_1, \ldots, \phi_r)^T$ an o.n. scaling vector if ϕ is refinable (that is, ϕ satisfies $\widehat{\phi}(\omega) = H(e^{\frac{i\omega}{2}})\widehat{\phi}(\frac{\omega}{2})$ for some FIR H), $\phi_j \in L_2(\mathbb{R})$, and

$$\int \phi_j(x-k)\overline{\phi_i(x)}dx = \delta(j-i)\delta(k), \quad 1 \leq j, i \leq r, k \in \mathbb{Z}.$$

For a matrix filter $P(z) = \sum_{k\in\mathbb{Z}} p_k z^{-k}$, it is said to be a *finite impulse response* (FIR) filter if each entry of $P(z)$ is a Laurent polynomial of z^{-1}, i.e., there exist integers k_1, k_2 such that $p_k = 0, k < k_1, k > k_2$. If $p_{k_1} \neq 0, p_{k_2} \neq 0$, we use $\text{len}(P) :=$

*Received by the editors June 5, 2000; accepted for publication (in revised form) by A. H. Sayed October 12, 2000; published electronically June 8, 2001. The research of this author was supported in part by NSERC Canada under grant OGP 121336.
http://www.siam.org/journals/simax/23-1/37292.html

†Department of Mathematical Sciences, University of Alberta, Edmonton, Canada T6G 2G1. Current address: Department of Mathematics, West Virginia University, Morgantown, WV 26506-6310 (jiang@math.wvu.edu, http://www.math.wvu.edu/∼jiang).

$k_2 - k_1 + 1$ to denote its filter length. An FIR $P(z)$ is said to be *causal* if each entry of $P(z)$ is a polynomial of z^{-1}, i.e., $p_k = 0$ for $k < 0$. Throughout this paper, P^T (resp., P^*) denotes the transpose (resp., the complex conjugate and transpose) of P, and I_r and 0_r denote the $r \times r$ identity matrix and zero matrix, respectively. We also let $0_{j\times l}$ denote the $j \times l$ zero matrix, and we would drop the subscript $j \times l$ when it does not cause any confusion. A necessary condition for H generating an o.n. scaling vector ϕ is that H is a matrix *conjugate quadrature filter* (CQF) (see, e.g., [12], [5], [6] for the necessary and sufficient conditions), i.e.,

$$H(z)H(z)^* + H(-z)H(-z)^* = I_r, \quad z \in \mathbb{C}. \tag{1.3}$$

A pair $\{H, G\}$ of matrix filters is called a multifilter bank, and it is said to be orthogonal if H, G satisfy (1.1) and (1.3).

For an FIR matrix filter H, write

$$\begin{aligned} H(z) &= \sum h_{2k} z^{-2k} + \left(\sum h_{2k+1} z^{-2k}\right) z^{-1} \\ &=: \frac{\sqrt{2}}{2} H_e(z^2) + \frac{\sqrt{2}}{2} H_o(z^2) z^{-1}. \end{aligned} \tag{1.4}$$

Then H satisfies (1.3) if and only if $[H_e(z), H_o(z)]$ is paraunitary. A $j \times l$ $(j \le l)$ matrix filter $P(z)$ with real coefficients p_k is called paraunitary if

$$P(z)P(z^{-1})^T = I_j, \quad z \neq 0,$$

that is, $P(e^{i\omega})$ is a matrix of o.n. rows for all $\omega \in \mathbb{R}$. Throughout this paper we assume that the coefficients of the matrix filters are real.

Let G be another FIR matrix filter, and let G_e, G_o be the corresponding filters defined in the way of (1.4). Then G satisfies (1.1) if and only if

$$[G_e(z), G_o(z)]E(z^{-1})^T = [0, I_r],$$

where $E(z)$ is the polyphase matrix of the multifilter bank $\{H, G\}$ defined by

$$E(z) := \begin{bmatrix} H_e(z) & H_o(z) \\ G_e(z) & G_o(z) \end{bmatrix}. \tag{1.5}$$

Thus given an H satisfying (1.3), to find G to satisfy (1.1) is equivalent to the paraunitary extension problem of a paraunitary matrix: Given an $r \times 2r$ paraunitary matrix $[H_e(z), H_o(z)]$, to find $[G_e(z), G_o(z)]$ such that $E(z)$ defined by (1.5) is paraunitary. It was shown in [9] and [10] that this paraunitary extension problem is always solvable, i.e., given a paraunitary matrix $[H_e(z), H_o(z)]$, one can always find its paraunitary extension $[G_e(z), G_o(z)]$.

The problem considered in this paper is as follows: Given an FIR matrix filter H generating an o.n. scaling vector ϕ with some symmetric property, is there a corresponding compactly supported multiwavelet ψ with some symmetric property, and if it exists, how can we construct the high-pass filter G? Equivalently, the problem we consider is to decide for a given paraunitary matrix $[H_e(z), H_o(z)]$ with some symmetry, if there exists its paraunitary extension $[G_e(z), G_o(z)]$ which also has some symmetry and, if it exists, how to construct it. Since symmetry is one of the most important properties of multiwavelets, the problem for a given symmetric o.n. scaling vector, to construct a corresponding symmetric multiwavelet, deserves our study.

There are two types of symmetric causal filters H. The first one is that H satisfies

$$z^{-\gamma} S_0 H(z^{-1}) S_0 = H(z), \quad S_0 = \mathrm{diag}(I_s, -I_{r-s}) \tag{1.6}$$

for a nonnegative integer $s \le r$. In this case, if H generates an o.n. scaling vector $\phi = (\phi_1, \ldots, \phi_r)^T$, then $\phi_1, \ldots, \phi_s$ are symmetric about $\gamma/2$ while $\phi_{s+1}, \ldots, \phi_r$ are antisymmetric about $\gamma/2$ (see, e.g., [1], [7], [15] about the relationship between the symmetry of ϕ, ψ and the property of H, G). Filters with this type of symmetry are called filters with the same symmetric center.

The second type of the symmetric filter H is that H will generate a symmetric o.n. scaling vector with its components not having the same symmetric center. We call the filter of this type a filter with different symmetric centers. In this paper we consider $H(z) = \sum_{k=0}^{2\gamma+1} h_k z^{-k}$ satisfying

$$z^{-(2\gamma+1)} \mathrm{diag}(S_0 z^2, 1) H(z^{-1}) \mathrm{diag}(S_0, z) = H(z), \quad S_0 = \mathrm{diag}(I_s, -I_{r-s-1}) \tag{1.7}$$

for a nonnegative integer $s \le r-1$. In this case, if H generates an o.n. scaling vector $\phi = (\phi_1, \ldots, \phi_r)^T$, then $\phi_1, \ldots, \phi_s$ are symmetric about $\gamma - \frac{1}{2}$ while $\phi_{s+1}, \ldots, \phi_{r-s-1}$ are antisymmetric about $\gamma - \frac{1}{2}$, and ϕ_r is symmetric about γ. $\phi_j, 1 \le j \le r-1$ are supported on $[0, 2\gamma - 1]$, while ϕ_r is supported on $[0, 2\gamma]$ (see [13] for the discussion on the supports of scaling vectors).

The symmetric extensions $[G_e, G_o]$ of the paraunitary matrices $[H_e, H_o]$ related to these two types of filters H are carried out in sections 2 and 3, respectively. We will construct their paraunitary extensions $[G_e, G_o]$ such that ψ defined by (1.2) with

$$G(z) := \frac{\sqrt{2}}{2} G_e(z^2) + \frac{\sqrt{2}}{2} G_o(z^2) z^{-1} \tag{1.8}$$

have symmetry and $\mathrm{len}(G) \le \mathrm{len}(H)$. More precisely, for H satisfying (1.6), the constructed G satisfies

$$z^{-\gamma} S_0 G(z^{-1}) S_0 = -G(z). \tag{1.9}$$

Thus components of the corresponding multiwavelet ψ are symmetric/antisymmetric about $\gamma/2$. For H satisfying (1.7), the constructed G satisfies

$$z^{-(2\gamma+1)} \mathrm{diag}(\mathcal{S}_1 z^2, \mathcal{S}_2) G(z^{-1}) \mathrm{diag}(S_0, z) = G(z), \tag{1.10}$$

where

$$\begin{aligned} &\mathcal{S}_1 := -I_{2s-r}, \mathcal{S}_2 := \mathrm{diag}(I_{r-s}, -I_{r-s}), \text{if } 2s \ge r; \\ &\mathcal{S}_1 := I_{r-2s}, \mathcal{S}_2 := \mathrm{diag}(I_s, -I_s), \text{if } 2s < r. \end{aligned} \tag{1.11}$$

The corresponding multiwavelet ψ has the following symmetric properties: (1) if $2s \ge r$, then $\psi_1, \ldots, \psi_{2s-r}$ are antisymmetric about $\gamma - \frac{1}{2}$, and $\psi_{2s-r+1}, \ldots, \psi_s$ and $\psi_{s+1}, \ldots, \psi_r$ are symmetric and antisymmetric about γ, respectively; (2) if $2s < r$, then $\psi_1, \ldots, \psi_{r-2s}$ are symmetric about $\gamma - \frac{1}{2}$, and $\psi_{r-2s+1}, \ldots, \psi_{r-s}$ and $\psi_{r-s+1}, \ldots, \psi_r$ are symmetric and antisymmetric about γ, respectively. Our construction also answers the problem on the existence of symmetric multiwavelets.

In section 4, we discuss the parametrization of symmetric orthogonal multifilter banks. Parametrizations of FIR orthogonal systems are of fundamental importance to the design of filter banks (see, e.g., [14], [16], [17]). Parametrizations of orthogonal filter banks are equivalent to the factorizations of paraunitary matrices. The

parametrization of symmetric orthogonal multifilter banks $\{H, G\}$ with the low-pass filter H satisfying (1.6) for $\gamma = 2N + 1$ was obtained in [7] (see [11] for the special case). For the case $r = 2, S_0 = (1)$, the parametrization of orthogonal multifilter banks $\{H, G\}$ with H satisfying (1.7) was provided in [8]. In section 4 we present the parametrization of orthogonal multifilter banks $\{H, G\}$ with H satisfying (1.6) for $\gamma = 2N$ and the parametrization of $\{H, G\}$ with H satisfying (1.7).

In this paper we use $\mathbb{N}, \mathbb{N}_0, \mathbb{Z}$ to denote sets of all natural numbers, nonnegative integers and integers, respectively. For $n \in \mathbb{N}$, denote

$$D_n := \mathrm{diag}(I_n, -I_{n-1}). \tag{1.12}$$

We use $O(n)$ to denote the set of all $n \times n$ real orthogonal matrices, and we use $\mathrm{Tr}(M)$ to denote the trace of a matrix M.

2. Symmetric extension of matrices related to the same symmetric center filters. In this section we discuss the symmetric matrix extension related to low-pass filters H satisfying (1.6). We consider the cases $\gamma = 2N + 1$ and $\gamma = 2N$, $N \in \mathbb{N}$ in the following two subsections, respectively.

2.1. The case $\gamma = 2N + 1$. Let $H = \sum_{k=0}^{2N+1} h_k z^{-k}$ be a matrix CQF satisfying (1.6) with $\gamma = 2N + 1$, and $h_0 \neq 0, h_{2N+1} \neq 0$. Let H_e, H_o be the filters defined by (1.4). Then (1.6) for $\gamma = 2N + 1$ is equivalent to

$$z^{-N} S_0 [H_e(z^{-1}), H_o(z^{-1})] \begin{bmatrix} & S_0 \\ S_0 & \end{bmatrix} = [H_e(z), H_o(z)]. \tag{2.1}$$

Denote

$$P(z) := [H_e(z), H_o(z)] U_0^T,$$

where

$$U_0 := \frac{\sqrt{2}}{2} \begin{bmatrix} I_r & S_0 \\ -I_r & S_0 \end{bmatrix}. \tag{2.2}$$

One can check that P satisfies

$$z^{-N} S_0 P(z^{-1}) \mathrm{diag}(I_r, -I_r) = P(z). \tag{2.3}$$

Note that $U_0 \in O(2r)$. Thus P satisfies $P(z)P(z^{-1})^T = I_r$, i.e., P is also paraunitary. In the following we give a symmetric paraunitary extension of P.

We need a lemma which will be used here and in the following sections.

LEMMA 2.1. (i) *Suppose an $\ell \times 2k$ ($\ell \geq k$) real matrix A satisfies*

$$A \mathrm{diag}(I_k, -I_k) A^T = 0. \tag{2.4}$$

Then there exists $u \in O(k)$ such that

$$A \begin{bmatrix} I_k \\ u^T \end{bmatrix} = 0. \tag{2.5}$$

(ii) *Suppose an $\ell \times (2k-1)$ ($\ell \geq k-1$) real matrix A satisfies*

$$A \mathrm{diag}(I_k, -I_{k-1}) A^T = 0. \tag{2.6}$$

Then there exists $u \in O(k)$ such that

$$A \begin{bmatrix} u^T \\ (I_{k-1}, 0) \end{bmatrix} = 0. \tag{2.7}$$

Proof. (i) By (2.4), the rank of A, denoted by n, is not greater than k. Let $\{x_1, x_2, \ldots, x_n\}$ be an o.n. basis for the columns of the matrix A^T (found by the Gram–Schmidt process). Write

$$[x_1, \ldots, x_n] =: \begin{bmatrix} Y_1 \\ Z_1 \end{bmatrix}, \quad Y_1, Z_1 \text{ are } k \times n \text{ matrices.}$$

Then $Y_1^T Y_1 + Z_1^T Z_1 = I_n$. By (2.4), we have $Y_1^T Y_1 = Z_1^T Z_1$. Thus

$$Y_1^T Y_1 = Z_1^T Z_1 = \frac{1}{2} I_n.$$

Therefore $\sqrt{2}Y_1, \sqrt{2}Z_1$ are $k \times n$ matrices of o.n. columns. Let Y_2, Z_2 be the $k \times (k-n)$ matrices such that $\sqrt{2}[Y_1, Y_2], \sqrt{2}[Z_1,\ Z_2] \in O(k)$. Then one has

$$\begin{bmatrix} Y_1^T & -Z_1^T \\ Y_2^T & -Z_2^T \end{bmatrix} x_j = 0, \quad 1 \le j \le n.$$

Since each column of A^T is a linear combination of $x_j, 1 \le j \le n$, we have

$$\begin{bmatrix} Y_1^T & -Z_1^T \\ Y_2^T & -Z_2^T \end{bmatrix} A^T = 0.$$

Thus (2.5) holds true with $u = -2[Y_1,\ Y_2][Z_1, Z_2]^T \in O(k)$.

(ii) The proof is similar. In this case write

$$[x_1, \ldots, x_n] =: \begin{bmatrix} Y_1 \\ Z_1 \end{bmatrix}, \quad Y_1, Z_1 \text{ are } k \times n \text{ and } (k-1) \times n \text{ matrices,}$$

where $\{x_1, x_2, \ldots, x_n\}$ is an o.n. basis for the columns of A^T. Then $\sqrt{2}Y_1, \sqrt{2}Z_1$ are $k \times n$ and $(k-1) \times n$ matrices of o.n. columns, respectively. Let Y_2, Z_2 be the $k \times (k-n)$ and $(k-1) \times (k-1-n)$ matrices such that $\sqrt{2}[Y_1, Y_2] \in O(k), \sqrt{2}[Z_1,\ Z_2] \in O(k-1)$. Then one has

$$\begin{bmatrix} Y_1^T & -Z_1^T \\ Y_2^T & -[Z_2, 0]^T \end{bmatrix} A^T = 0, \quad 1 \le j \le n.$$

Thus

$$A \begin{bmatrix} Y_1 & Y_2 \\ -Z_1 & -[Z_2, 0] \end{bmatrix} = 0,$$

and (2.7) holds with $u = -2\mathrm{diag}([Z_1, Z_2], 1)[Y_1,\ Y_2]^T \in O(k)$. ∎

From the proof of Lemma 2.1, we know that orthogonal matrices u in (2.5) and (2.7) are constructed by the Gram–Schmidt process of the rows of A.

For $v \in O(r)$, define

$$V(z) := \frac{1}{2} \begin{bmatrix} I_r & -v \\ -v^T & I_r \end{bmatrix} + \frac{1}{2} \begin{bmatrix} I_r & v \\ v^T & I_r \end{bmatrix} z^{-1}, \quad v \in O(r). \tag{2.8}$$

Then one has the following lemma.

LEMMA 2.2. *Let $V(z)$ be the matrix defined by* (2.8) *with some $v \in O(r)$. Then*

(i) $V(z)^T = V(z^{-1})$, $V(z)V(z^{-1}) = I_{2r}$.

(ii) $z^{-1}\mathrm{diag}(I_r, -I_r)V(z^{-1})\mathrm{diag}(I_r, -I_r) = V(z)$.

Proof. Statements (i) and (ii) follow from the direct calculations. ∎

For a causal paraunitary matrix P satisfying (2.3), write

$$P = p_0 + \cdots + p_N z^{-N}.$$

By (2.3), $p_N = S_0 p_0 \mathrm{diag}(I_r, -I_r)$. On the other hand, the paraunitariness of P implies that $p_0 p_N^T = 0$. Thus

$$p_0 \mathrm{diag}(I_r, -I_r) p_0^T = 0.$$

By Lemma 2.1, we can find $v_N \in O(r)$ such that $p_0[I_r, v_N]^T = 0$.

Let V_N be the matrix defined by (2.8) with $v = v_N$. Then $\tilde{P}$ defined by

$$\tilde{P}(z) := P(z)V_N(z^{-1})$$

is causal. Since $V_N(z)$ is paraunitary and satisfies condition (ii) of Lemma 2.2, $\tilde{P}$ is also paraunitary and satisfies (2.3) with $N-1$. Continuing this process, we construct $v_{N-1}, \ldots, v_1 \in O(r)$ similarly such that P can be written as

$$P(z) = \tilde{P}(z)V_N(z) = \cdots = P_0 V_1(z) \cdots V_N(z),$$

where V_j are defined by (2.8) with $v = v_j$, $1 \le j \le N$, and P_0 is an $r \times 2r$ matrix of constant entries satisfying

$$P_0 P_0^T = I_r, \quad S_0 P_0 = P_0 \mathrm{diag}(I_r, -I_r).$$

One has that for P_0 satisfying the above conditions, it can be written as

$$P_0 = \begin{bmatrix} a_0 & 0 \\ 0 & b_0 \end{bmatrix},$$

where a_0 and b_0 are $s \times r$ and $(r-s) \times r$ matrices, respectively, with $a_0 a_0^T = I_s, b_0 b_0^T = I_{r-s}$. Let a_1 and b_1 be such matrices that $\left[\begin{smallmatrix} a_0 \\ a_1 \end{smallmatrix}\right], \left[\begin{smallmatrix} b_0 \\ b_1 \end{smallmatrix}\right] \in O(r)$. Denote

$$Q_0 = \begin{bmatrix} 0 & b_1 \\ a_1 & 0 \end{bmatrix}.$$

Then $\left[\begin{smallmatrix} P_0 \\ Q_0 \end{smallmatrix}\right] \in O(2r)$ and $-S_0 Q_0 = Q_0 \mathrm{diag}(I_r, -I_r)$. Thus Q defined by

$$Q(z) := Q_0 V_1(z) \cdots V_N(z)$$

satisfies that

$$z^{-N} S_0 Q(z^{-1}) \mathrm{diag}(I_r, -I_r) = -Q(z),$$

and $\left[\begin{smallmatrix} P \\ Q \end{smallmatrix}\right]$ is causal and paraunitary. Therefore, Q is a symmetric paraunitary extension of P. We note that the degree of each entry of Q as a polynomial of z^{-1} is not greater than N. Let $[G_e, G_o] = Q(z)U_0$. Then we have the following theorem.

THEOREM 2.3. *Suppose $[H_e, H_o]$ is an $r \times 2r$ paraunitary matrix satisfying* (1.6) *for $\gamma = 2N+1$. Then $[G_e, G_o]$ obtained by the above algorithm is a symmetric paraunitary extension of $[H_e, H_o]$ with*

$$z^{-N} S_0 [G_e(z^{-1}), G_o(z^{-1})] \begin{bmatrix} & S_0 \\ S_0 & \end{bmatrix} = -[G_e(z), G_o(z)].$$

Let G be the filter defined by (1.8). Then G is causal, $\mathrm{len}(G) \le 2N+1$, G satisfies (1.9), and $\{H, G\}$ is an orthogonal multifilter bank. Thus we have the following corollary.

COROLLARY 2.4. *Suppose the causal FIR H generates an o.n. scaling vector $\phi = (\phi_1, \dots, \phi_r)^T$ supported on $[0, 2N+1]$ with the first s components symmetric and the other components antisymmetric about $N + \frac{1}{2}$. Let G be the causal matrix filter constructed by the above algorithm. Then ψ defined by* (1.2) *is a multiwavelet supported on $[0, 2N+1]$ with the first s components antisymmetric and the other components symmetric about $N + \frac{1}{2}$.*

Example 1. Let $H(z) = \sum_{k=0}^{5} h_k z^{-k}$ be a matrix CQF with

$$h_0 = \frac{1}{101} \begin{bmatrix} 100/101 & 10/101 \\ 10e & e \end{bmatrix}, \quad h_1 = \frac{1}{101} \begin{bmatrix} 100/101 & 1000/101 \\ 10e & 100e \end{bmatrix},$$
$$h_2 = \frac{1}{101} \begin{bmatrix} 9801/202 & 990/101 \\ 101f & 0 \end{bmatrix}, \quad h_j = S_0 h_{5-j} S_0, \quad 3 \le j \le 5,$$

where $S_0 = \mathrm{diag}(1, -1)$, and

$$e := \frac{261}{4} \frac{7\sqrt{1147} - 202}{707\sqrt{1147} - 41282}, \quad f := \frac{101}{4} \frac{14\sqrt{1147} - 143}{707\sqrt{1147} - 41282}.$$

H satisfies (1.6) with $\gamma = 5$, and it generates a symmetric/antisymmetric o.n. scaling vector ϕ with $\phi \in W^{1.87659}(\mathbb{R})$. Here $W^s(\mathbb{R})$ denotes the Sobolev space consisting of all functions with $\hat{f}(\omega)(1 + |\omega|^2)^{\frac{s}{2}} \in L_2(\mathbb{R})$, and we use the smoothness estimate of ϕ provided in [4]. We will construct the corresponding symmetric high-pass filter by the above algorithm.

Let H_e, H_o be the filters defined by (1.4). Then $P(z) := [H_e, H_o] U_0^T$ is $p_0 + p_1 z^{-1} + p_2 z^{-2}$ with

$$p_0 = \frac{1}{101} \begin{bmatrix} 200/101 & -990/101 & 0 & -10 \\ 20e & -99e & 0 & -101e \end{bmatrix},$$
$$p_1 = \frac{1}{101} \begin{bmatrix} 99^2/101 & 1980/101 & 0 & 0 \\ 0 & 0 & -202f & 0 \end{bmatrix},$$
$$p_2 = S_0 p_0 \mathrm{diag}(I_2, -I_2).$$

By the above algorithm, we can construct $v_2 \in O(2)$, then $v_1 \in O(2)$ with

$$v_2 = -\frac{1}{101} \begin{bmatrix} 99 & -20 \\ 20 & 99 \end{bmatrix}, \quad v_1 = \frac{1}{101} \begin{bmatrix} 99 & 20 \\ 20 & -99 \end{bmatrix}$$

such that $P(z) V_2(z^{-1}) V_1(z^{-1})$ is

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -2f & -2e \end{bmatrix},$$

where V_j are the matrices defined by (2.8) with $v = v_j, j = 1, 2$. Let Q_0 defined by

$$Q_0 = \begin{bmatrix} 0 & 0 & 2e & -2f \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

be the orthogonal extension of P_0. Then $Q(z) = Q_0 V_1(z) V_2(z)$ is a symmetric extension of $P(z)$. Finally, we get $G(z) = \frac{\sqrt{2}}{2} Q(z^2) U_0 \begin{bmatrix} I_2 \\ z^{-1} I_2 \end{bmatrix} =: \sum_{k=0}^{5} g_k z^{-k}$ with

$$g_0 = \frac{1}{101} \begin{bmatrix} 10f & f \\ -495/101 & -99/202 \end{bmatrix}, \quad g_1 = \frac{1}{101} \begin{bmatrix} 10f & 100f \\ -495/101 & -4950/101 \end{bmatrix},$$
$$g_2 = \frac{1}{101} \begin{bmatrix} -101e & 0 \\ 990/101 & 200/101 \end{bmatrix}, \quad g_j = -S_0 g_{5-j} S_0, \quad 3 \le j \le 5.$$

The corresponding multiwavelet ψ is symmetric/antisymmetric about 5/2.

2.2. The case $\gamma = 2N$. Suppose H satisfies (1.6) with $\gamma = 2N, N \in \mathbb{N}$. We hope to find such a causal G that each component of the corresponding ψ has the same symmetry center N, i.e., to find G to satisfy

$$z^{-2N} S_1 G(z^{-1}) S_0 = G(z), \quad S_1 = \operatorname{diag}(\pm 1, \dots, \pm 1). \tag{2.9}$$

First we have the following proposition.

PROPOSITION 2.5. *Suppose $\{H, G\}$ is orthogonal, and H, G satisfy* (1.6) *for $\gamma = 2N$ and* (2.9)*, respectively. Then r is even and by permutations S_0 and S_1 are* $\operatorname{diag}(I_{\frac{r}{2}}, -I_{\frac{r}{2}})$.

Proof. By (1.6) for $\gamma = 2N$ and (2.9),

$$\begin{bmatrix} S_0 & \\ & S_1 \end{bmatrix} \begin{bmatrix} H(1) & H(-1) \\ G(1) & G(-1) \end{bmatrix} \begin{bmatrix} S_0 & \\ & S_0 \end{bmatrix} = \begin{bmatrix} H(1) & H(-1) \\ G(1) & G(-1) \end{bmatrix} \tag{2.10}$$

and

$$(-1)^N \begin{bmatrix} S_0 & \\ & S_1 \end{bmatrix} \begin{bmatrix} H(-i) & H(i) \\ G(-i) & G(i) \end{bmatrix} \begin{bmatrix} S_0 & \\ & S_0 \end{bmatrix} = \begin{bmatrix} H(-i) & H(i) \\ G(-i) & G(i) \end{bmatrix} \begin{bmatrix} & I_r \\ I_r & \end{bmatrix}. \tag{2.11}$$

Equation (2.10) implies that $\operatorname{diag}(S_0, S_1)$ is similar to $\operatorname{diag}(S_0, S_0)$. Thus $\operatorname{Tr}(S_1) = \operatorname{Tr}(S_0)$, while (2.11) implies that $\operatorname{Tr}(S_1) + \operatorname{Tr}(S_0) = 0$. Therefore, $\operatorname{Tr}(S_1) = \operatorname{Tr}(S_0) = 0$. Hence, r is even, and half diagonal entries of both S_0 and S_1 are 1 and the other half diagonal entries are -1. ∎

Due to Proposition 2.5, in the rest of this subsection we always assume that $r = 2m$ for some $m \in \mathbb{N}$ and

$$S_0 = \operatorname{diag}(I_m, -I_m).$$

We will discuss the following symmetric extension problem: Given a causal H satisfying (1.6) for $\gamma = 2N$, find G such that G satisfies (2.9) with $S_1 = \operatorname{diag}(-I_m, I_m)$ and $\{H, G\}$ is orthogonal. For this we introduce a paraunitary matrix $U(z)$ defined by

$$U(z) := \frac{1}{2} \begin{bmatrix} S_0 U_1 S_0 & U_1 \\ U_1 z^{-1} & S_0 U_1 S_0 \end{bmatrix}, \quad U_1 = \begin{bmatrix} I_r & u \\ u^T & I_r \end{bmatrix}, \quad u \in O(m). \tag{2.12}$$

LEMMA 2.6. *Let $U(z)$ be the matrix defined by* (2.12) *for some $u \in O(m)$. Then*
(i) $U(z)U(z^{-1})^T = I_{2m}$.
(ii) $U(z^{-1})\mathrm{diag}(S_0 z^{-1}, S_0)U(z^{-1})^T = \left[\begin{smallmatrix} & S_0 \\ S_0 & \end{smallmatrix}\right]$.

Proof. One can obtain (i) by a direct calculation. For (ii), we have

$$U(z^{-1})\mathrm{diag}(S_0 z^{-1}, S_0) = \frac{1}{2}\begin{bmatrix} S_0 U_1 z^{-1} & U_1 S_0 \\ U_1 S_0 & S_0 U_1 \end{bmatrix} = \begin{bmatrix} & S_0 \\ S_0 & \end{bmatrix} U(z). \qquad \square$$

For a causal matrix CQF H satisfying (1.6) for $\gamma = 2N$, let H_e, H_o be the causal filters defined by (1.4). Then (1.6) for $\gamma = 2N$ is equivalent to

$$z^{-(N-1)} S_0 [H_e(z^{-1}), H_o(z^{-1})]\mathrm{diag}(z^{-1}S_0, S_0) = [H_e(z), H_o(z)]. \tag{2.13}$$

By (1.3) and symmetry of H, $h_0 h_{2N}^T = 0$ and $h_{2N} = S_0 h_0 S_0$. Thus

$$h_0 S_0 h_0^T = 0.$$

By Lemma 2.1, we can find $u_0 \in O(m)$ such that $h_0 [I_m, u_0]^T = 0$. Thus

$$h_0 \begin{bmatrix} I_m & u_0 \\ u_0^T & I_m \end{bmatrix} = 0. \tag{2.14}$$

Let $U_0(z)$ be the paraunitary matrix defined by (2.12) with $u = u_0$. Equation (2.14) implies that the $r \times 2r$ matrix $[\widetilde{H}_e, \widetilde{H}_o]$ defined by

$$[\widetilde{H}_e(z), \widetilde{H}_o(z)] = [H_e(z), H_o(z)]U_0(z^{-1})^T$$

is causal. The paraunitariness of $[H_e, H_o]$ and $U_0(z)$ imply that $[\widetilde{H}_e, \widetilde{H}_o]$ is also paraunitary. On the other hand, by (ii) in Lemma 2.6 and (2.13), one has

$$z^{-(N-1)} S_0 [\widetilde{H}_e(z^{-1}), \widetilde{H}_o(z^{-1})] \begin{bmatrix} & S_0 \\ S_0 & \end{bmatrix} = [\widetilde{H}_e(z), \widetilde{H}_o(z)].$$

Thus by Theorem 2.3, there exist causal FIR filters $\widetilde{G}_e(z), \widetilde{G}_o(z)$ such that $[\widetilde{G}_e(z), \widetilde{G}_o(z)]$ is a symmetric paraunitary extension of $[\widetilde{H}_e, \widetilde{H}_o]$ with

$$z^{-(N-1)} S_0 [\widetilde{G}_e(z^{-1}), \widetilde{G}_o(z^{-1})] \begin{bmatrix} & S_0 \\ S_0 & \end{bmatrix} = -[\widetilde{G}_e(z), \widetilde{G}_o(z)].$$

Define

$$[G_e(z), G_o(z)] := [\widetilde{G}_e(z), \widetilde{G}_o(z)]U_0(z).$$

Then $[G_e(z), G_o(z)]$ is a symmetric paraunitary extension of $[H_e, H_o]$, and it satisfies

$$z^{-(N-1)} S_0 [G_e(z^{-1}), G_o(z^{-1})]\mathrm{diag}(z^{-1}S_0, S_0) = -[G_e(z), G_o(z)]. \tag{2.15}$$

THEOREM 2.7. *Suppose $[H_e, H_o]$ is an $r \times 2r$ causal paraunitary matrix satisfying* (2.13). *Then $[G_e, G_o]$ obtained by the above algorithm is a symmetric paraunitary extension of $[H_e, H_o]$ with $[G_e(z), G_o(z)]$ satisfying* (2.15). *Furthermore,* $\mathrm{len}(G_e) \le N$, $\mathrm{len}(G_o) \le N - 1$.

Let G be the filter defined by (1.8). Then G is causal, $\mathrm{len}(G) \le 2N$, G satisfies (2.9) with $S_1 = \mathrm{diag}(-I_m, I_m)$, and $\{H, G\}$ is orthogonal.

COROLLARY 2.8. *Suppose the causal filter H generates an o.n. scaling vector $\phi = (\phi_1, \ldots, \phi_{2m})^T$ supported on $[0, 2N]$ with the first m components symmetric and the other m components antisymmetric about N. Let G be the matrix filter obtained by the above algorithm. Then ψ defined by* (1.2) *is a multiwavelet supported on $[0, 2N]$ with the first m components antisymmetric and the other m components symmetric about N.*

3. Symmetric extension of matrices related to different symmetric center filters. Suppose $H = \sum_{k=0}^{2\gamma+1} h_k z^{-k}$ is a matrix CQF satisfying (1.7). Let H_e, H_o be the causal filters defined by (1.4). Then $[H_e, H_o]$ satisfies

$$z^{-\gamma}\mathrm{diag}(S_0 z, 1)[H_e(z^{-1}), H_o(z^{-1})]\mathrm{diag}(J_0, z) = [H_e(z), H_o(z)], \tag{3.1}$$

where

$$J_0 := \begin{bmatrix} & & S_0 \\ & 1 & \\ S_0 & & \end{bmatrix}. \tag{3.2}$$

In this section, we discuss the symmetric extension of $[H_e, H_o]$. We will construct $[G_e, G_o]$ such that it is a paraunitary matrix of $[H_e, H_o]$ and

$$z^{-\gamma}\mathrm{diag}(\mathcal{S}_1 z, \mathcal{S}_2)[G_e(z^{-1}), G_o(z^{-1})]\mathrm{diag}(J_0, z) = [G_e(z), G_o(z)], \tag{3.3}$$

where $\mathcal{S}_1$ and $\mathcal{S}_2$ are defined by (1.11). Then G defined by (1.8) satisfies (1.10).

Define $R_1 \in O(2r-1)$ by

$$R_1 := \frac{\sqrt{2}}{2}\begin{bmatrix} I_{r-1} & 0 & I_{r-1} \\ 0 & \sqrt{2} & 0 \\ -I_{r-1} & 0 & I_{r-1} \end{bmatrix}. \tag{3.4}$$

Then

$$R_1 J_0 R_1^T = \mathrm{diag}(S_0, 1, -S_0).$$

Let M_0 be such a $2r \times 2r$ permutation matrix that

$$M_0\mathrm{diag}(S_0, 1, -S_0, z)M_0 = \mathrm{diag}(z, I_r, -I_{r-1}) = \mathrm{diag}(z, D_r). \tag{3.5}$$

Recall a matrix is called a permutation matrix if its columns are a permutation of the columns of the identity matrix. D_r is the matrix defined by (1.12). Denote

$$P(z) := [H_e(z), H_o(z)]\mathrm{diag}(R_1, 1)M_0.$$

Then P is causal and paraunitary, and $[H_e, H_o]$ satisfies (3.1) if and only if P satisfies

$$z^{-\gamma}\mathrm{diag}(S_0 z, 1)P(z^{-1})\mathrm{diag}(z, D_r) = P(z). \tag{3.6}$$

We now consider the symmetric extension of P. We want to construct a causal filter Q such that Q is a paraunitary extension of P and satisfies

$$z^{-\gamma}\mathrm{diag}(\mathcal{S}_1 z, \mathcal{S}_2)Q(z^{-1})\mathrm{diag}(z, D_r) = Q(z). \tag{3.7}$$

If Q satisfies (3.7), then $[G_e, G_o]$ defined by $[G_e, G_o] = QM_0\mathrm{diag}(R_1^T, 1)$ satisfies (3.3).

First let us consider the case $\gamma = 1$. In this case, (3.6) implies that P can be written in the form of

$$\begin{bmatrix} 0 & a_0 & 0 \\ 0 & 0 & b_0 \\ c_0 & y_1 & y_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & y_1 & -y_2 \end{bmatrix} z^{-1},$$

where a_0, b_0 are $s \times r$ and $(r-s-1) \times (r-1)$ matrices, $c_0 \in \mathbb{R}$ and y_1, y_2 are $1 \times r$ and $1 \times (r-1)$ row vectors. The paraunitariness of P implies that

$$a_0 a_0^T = I_s, \quad b_0 b_0^T = I_{r-s-1}, \quad a_0 y_1^T = 0, \quad b_0 y_2^T = 0, \quad y_1 y_1^T = y_2 y_2^T, \quad c_0^2 + 4 y_1 y_1^T = 1.$$

Thus we know a_0, b_0 are $s \times r$ and $(r-s-1) \times (r-1)$ matrices of o.n. rows.

Let θ be such a real number that

$$\cos\theta = c_0, \quad \sin\theta = \sqrt{1 - c_0^2}.$$

Then y_1, y_2 can be written as

$$y_1 = \frac{1}{2}\sin\theta u_0, \quad y_2 = \frac{1}{2}\sin\theta v_0, \tag{3.8}$$

where u_0 and v_0 are $1 \times r$ and $1 \times (r-1)$ row vectors such that $\left[\begin{smallmatrix} a_0 \\ u_0 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} b_0 \\ v_0 \end{smallmatrix}\right]$ are $(s+1) \times r$ and $(r-s) \times (r-1)$ matrices of o.n. rows. Indeed, if $\sin\theta = 0$, then $y_1 = 0, y_2 = 0$ and any unit vectors u_0, v_0 o.n. to rows of a_0, b_0, respectively, will do. If $\sin\theta \neq 0$, $u_0 = 2y_1/\sin\theta$, $v_0 = 2y_2/\sin\theta$.

Consider the case $2s \geq r$. Choose $(r-s-1) \times r$, $(2s-r) \times (r-1)$, and $(r-s-1) \times (r-1)$ matrices $\tilde{u}$, $\tilde{v}_1$, $\tilde{v}$ such that

$$[a_0^T, u_0^T, \tilde{u}^T] \in O(r), \quad [b_0^T, v_0^T, \tilde{v}^T, \tilde{v}_1^T] \in O(r-1),$$

where u_0, v_0 are the vectors satisfying (3.8). Then Q defined by

$$Q(z) = \frac{1}{2}\begin{bmatrix} 0 & 0 & 2\tilde{v}_1 \\ -2\sin\theta & \cos\theta u_0 & \cos\theta v_0 \\ 0 & \tilde{u} & \tilde{v} \\ 0 & u_0 & v_0 \\ 0 & \tilde{u} & \tilde{v} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & \cos\theta u_0 & -\cos\theta v_0 \\ 0 & \tilde{u} & -\tilde{v} \\ 0 & -u_0 & v_0 \\ 0 & -\tilde{u} & \tilde{v} \end{bmatrix} z^{-1}$$

is a symmetric paraunitary extension of P with Q satisfying (3.7) for $\gamma = 1$.

For the case $2s < r$, choose $(r-2s) \times r$, $(s-1) \times (r-1)$, and $(s-1) \times (r-1)$ matrices $\tilde{u}_1$, $\tilde{u}$, and $\tilde{v}$ such that

$$[a_0^T, u_0^T, \tilde{u}^T, \tilde{u}_1^T] \in O(r), \quad [b_0^T, v_0^T, \tilde{v}^T] \in O(r-1),$$

where u_0, v_0 are the vectors satisfying (3.8). Then Q defined by

$$Q(z) = \frac{1}{2}\begin{bmatrix} 0 & 2\tilde{u}_1 & 0 \\ -2\sin\theta & \cos\theta u_0 & \cos\theta v_0 \\ 0 & \tilde{u} & \tilde{v} \\ 0 & u_0 & v_0 \\ 0 & \tilde{u} & \tilde{v} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & \cos\theta u_0 & -\cos\theta v_0 \\ 0 & \tilde{u} & -\tilde{v} \\ 0 & -u_0 & v_0 \\ 0 & -\tilde{u} & \tilde{v} \end{bmatrix} z^{-1}$$

is a symmetric paraunitary extension of P with Q satisfying (3.7) for $\gamma = 1$.

PROPOSITION 3.1. *Suppose P is a causal paraunitary matrix that satisfies* (3.6) *for $\gamma = 1$. Then Q constructed above is a symmetric paraunitary extension of P satisfying* (3.7) *for $\gamma = 1$.*

Now let us discuss the case $\gamma \geq 2$. First we introduce a paraunitary matrix $W(z)$. For $w =: \left[\begin{smallmatrix} \tilde{w} \\ w_r \end{smallmatrix}\right] \in O(r)$ with w_r the last row of w, define

$$W(z) := \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & \tilde{w} & -I_{r-1} \\ 0 & -\tilde{w} & I_{r-1} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 2w_r & 0 \\ 0 & 0 & 0 \\ 0 & \tilde{w} & I_{r-1} \\ 0 & \tilde{w} & I_{r-1} \end{bmatrix} z^{-1}. \tag{3.9}$$

Then by a direct calculation, one has the following lemma.

LEMMA 3.2. *Let $W(z)$ be the matrix defined by* (3.9) *with some $w \in O(r)$. Then*
(i) $W(z)W(z^{-1})^T = I_{2r}$.
(ii) $z^{-1}\mathrm{diag}(z^{-1}, D_r)W(z^{-1})\mathrm{diag}(z, D_r) = W(z)$.

Suppose P is a paraunitary matrix satisfying (3.6) for $\gamma \geq 2$. Then P has the form of

$$P = \begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} + \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} z^{-1} + \cdots + \begin{bmatrix} S_0 a_0 & S_0 b_1 D_r \\ c_1 & d_2 D_r \end{bmatrix} z^{-(\gamma-2)} + \begin{bmatrix} 0 & S_0 b_0 D_r \\ c_0 & d_1 D_r \end{bmatrix} z^{-(\gamma-1)} + \begin{bmatrix} 0 & 0 \\ 0 & d_0 D_r \end{bmatrix} z^{-\gamma}$$

for some $c_j \in \mathbb{R}$, $(r-1)\times 1$ and $1 \times (2r-1)$ vectors a_j and d_j, and $(r-1)\times(2r-1)$ matrices b_j. The paraunitariness of P implies that

$$\begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & d_0 D_r \end{bmatrix}^T = 0,$$

$$\begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \begin{bmatrix} 0 & S_0 b_0 D_r \\ c_0 & d_1 D_r \end{bmatrix}^T + \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & d_0 D_r \end{bmatrix}^T = 0,$$

which leads to

$$\begin{bmatrix} b_0 \\ d_0 \end{bmatrix} D_r [b_0^T, d_0^T] = 0.$$

By Lemma 2.1, we can construct $w_\gamma \in O(r)$ satisfying

$$\begin{bmatrix} b_0 \\ d_0 \end{bmatrix} \begin{bmatrix} w_\gamma^T \\ (I_{r-1}, 0) \end{bmatrix} = 0. \tag{3.10}$$

Write

$$w_\gamma =: \begin{bmatrix} \tilde{w}_\gamma \\ w_{\gamma,r} \end{bmatrix},$$

where $w_{\gamma,r}$ is the last row of w_γ. From (3.10), we have

$$d_0 D_r \begin{bmatrix} w_{\gamma,r}^T & \tilde{w}_\gamma^T \\ 0 & -I_{r-1} \end{bmatrix} = 0. \tag{3.11}$$

Let $W_\gamma(z)$ be the matrix defined by (3.9) with $w = w_\gamma$. Define $\tilde{P}$ by

$$\tilde{P}(z) := P(z) W_\gamma(z^{-1})^T.$$

Then (3.10) and (3.11) imply that $\tilde{P}$ is causal, and it can be written as

$$\tilde{p}_0 + \cdots + \tilde{p}_{N-1} z^{-(\gamma-1)}$$

for some $r \times 2r$ matrices $\tilde{p}_j$. Since $W_\gamma(z)$ is paraunitary and satisfies (ii) of Lemma 3.2, $\tilde{P}$ is also paraunitary and satisfies (3.6) with $\gamma - 1$. In this way, we construct $w_{\gamma-1}, \ldots, w_2 \in O(r)$ similarly such that P can be written as

$$P(z) = \tilde{P}(z) W_\gamma(z) = \cdots = P_1(z) W_2(z) \cdots W_\gamma(z),$$

where W_j is defined by (3.9) with $w = w_j$, and P_1 is an $r \times 2r$ matrix satisfying (3.6) with $\gamma = 1$. By Proposition 3.1, we can construct a causal filter Q_1 such that Q_1 is a symmetric paraunitary extension of P_1. Let

$$Q(z) = Q_1(z)W_2(z)\cdots W_\gamma(z).$$

Then Q is a symmetric extension of P satisfying (3.7). Define

$$[G_e(z), G_o(z)] = Q(z)M_0\mathrm{diag}(R_1^T, 1).$$

Then $[G_e, G_o]$ is a symmetric paraunitary extension of $[H_e, H_o]$ with $[G_e, G_o]$ satisfying (3.3).

THEOREM 3.3. *Suppose $[H_e, H_o]$ is an $r \times 2r$ causal paraunitary matrix satisfying* (3.1). *Then $[G_e, G_o]$ obtained by the above algorithm is a symmetric paraunitary extension of $[H_e, H_o]$ satisfying* (3.3). *Furthermore, the filter length of $[G_e, G_o]$ is not greater than γ.*

Let G be the matrix defined by (1.8). Then G is causal, and it satisfies (1.10) and $\mathrm{len}(G) \le 2\gamma + 1$.

COROLLARY 3.4. *Assume that the causal FIR H generates an o.n. scaling vector $\phi = (\phi_1, \ldots, \phi_r)^T$ with $\phi_1, \ldots, \phi_s$ and $\phi_{s+1}, \ldots, \phi_{r-s-1}$ symmetric and antisymmetric about $\gamma - \frac{1}{2}$, and ϕ_r is symmetric about γ. Let G be the causal matrix filter obtained by the above algorithm. Then $\psi = (\psi_1, \ldots, \psi_r)^T$ defined by* (1.2) *is such a multiwavelet that $\psi_1, \ldots, \psi_{2s-r}$ are antisymmetric about $\gamma - \frac{1}{2}$, and $\psi_{2s-r+1}, \ldots, \psi_s$ and $\psi_{s+1}, \ldots, \psi_r$ are symmetric and antisymmetric about γ, respectively, for the case $2s \ge r$; and $\psi_1, \ldots, \psi_{r-2s}$ are symmetric about $\gamma - \frac{1}{2}$, and $\psi_{r-2s+1}, \ldots, \psi_{r-s}$ and $\psi_{r-s+1}, \ldots, \psi_r$ are symmetric and antisymmetric about γ, respectively, for the case $2s < r$.*

Example 2. Let $\phi = (\phi_1, \phi_2)^T$ be the o.n. scaling vector constructed in [2]. The low-pass filter H for ϕ is given by

$$H(z) = \frac{1}{20}\begin{bmatrix} 6 + 6z^{-1} & 8\sqrt{2} \\ (-1 + 9z^{-1} + 9z^{-2} - z^{-3})/\sqrt{2} & -3 + 10z^{-1} - 3z^{-2} \end{bmatrix}.$$

In this case $S_0 = (1)$ and

$$R_1 = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 0 & 1 \\ 0 & \sqrt{2} & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad M_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & I_2 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Let H_e, H_o be the filters defined by (1.4). Then $P := [H_e, H_o]\mathrm{diag}(R_1, 1)M_0$ is

$$\begin{bmatrix} 0 & \frac{4}{5} & \frac{3}{5} & 0 \\ \frac{\sqrt{2}}{2} & -\frac{3\sqrt{2}}{20} & \frac{\sqrt{2}}{5} & -\frac{\sqrt{2}}{4} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{3\sqrt{2}}{20} & \frac{\sqrt{2}}{5} & \frac{\sqrt{2}}{4} \end{bmatrix} z^{-1}.$$

By the above algorithm, one can find P's symmetric extension Q:

$$Q(z) = \frac{1}{2}\begin{bmatrix} -\sqrt{2} & -\frac{3\sqrt{2}}{10} & \frac{2\sqrt{2}}{5} & \frac{\sqrt{2}}{2} \\ 0 & -\frac{3}{5} & \frac{4}{5} & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & -\frac{3\sqrt{2}}{10} & \frac{2\sqrt{2}}{5} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{3}{5} & -\frac{4}{5} & 1 \end{bmatrix} z^{-1}.$$

Then we get $[G_e, G_o] = Q(z)M_0\mathrm{diag}(R_1^T, 1)$, and finally we have $G(z) = G_e(z^2) + G_o(z^2)z^{-1}$:

$$G(z) = \frac{1}{20}\begin{bmatrix} (9 - z^{-1} - z^{-2} + 9z^{-3})/\sqrt{2} & -3 - 10z^{-1} - 3z^{-2} \\ 9 - z^{-1} + z^{-2} - 9z^{-3} & 3\sqrt{2}(z^{-2} - 1) \end{bmatrix}.$$

The first and the second components of the corresponding $\psi = (\psi_1, \psi_2)^T$ are symmetric and antisymmetric about 1, respectively.

4. Parametrization of symmetric multifilter banks. In this section we discuss parametrizations of symmetric orthogonal filter banks. We consider two types of symmetry filters, those having or not having the same symmetric centers, in the following two subsections, respectively.

4.1. Filter banks with the same symmetric center. Assume that $\{H, G\}$ is a causal orthogonal filter bank satisfying

$$z^{-\gamma} S_0 H(z^{-1}) S_0 = H(z), \quad z^{-\gamma} S_1 G(z^{-1}) S_0 = G(z), \tag{4.1}$$

where

$$S_0 = \operatorname{diag}(I_s, -I_{r-s}), \quad S_1 = \operatorname{diag}(\pm 1, \dots, \pm 1), \quad s \in \mathbb{N}_0.$$

One can show as in subsection 2.2 that $\operatorname{Tr}(S_1) = \operatorname{Tr}(S_0)$. In this subsection we assume that $S_1 = -S_0$.

THEOREM 4.1. *A causal FIR multifilter bank $\{H, G\}$ is orthogonal and satisfies* (4.1) *with $\gamma = 2N + 1$ for some $N \in \mathbb{N}$, $S_1 = -S_0$ if and only if it can be factorized in the form of*

$$\begin{aligned} H(z) &= \frac{\sqrt{2}}{2} \begin{bmatrix} a_0 & 0 \\ 0 & b_0 \end{bmatrix} V_1(z^2) \cdots V_N(z^2) U_0 \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix}, \\ G(z) &= \frac{\sqrt{2}}{2} \begin{bmatrix} 0 & b_1 \\ a_1 & 0 \end{bmatrix} V_1(z^2) \cdots V_N(z^2) U_0 \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix}, \end{aligned} \tag{4.2}$$

where V_j are the matrices defined by (2.8) *with $v_j \in O(r)$, a_0, b_1 and a_1, b_0 are $s \times r$ and $(r - s) \times r$ matrices, respectively, with $[a_0^T, a_1^T], [b_0^T, b_1^T] \in O(r)$, and U_0 is the matrix defined by* (2.2).

THEOREM 4.2. *A causal FIR multifilter bank $\{H, G\}$ is orthogonal and satisfies* (4.1) *with $\gamma = 2N$ for some $N \in \mathbb{N}$, $S_0 = \operatorname{diag}(I_m, -I_m)$, $S_1 = -S_0$ if and only if it can be factorized in the form of*

$$\begin{aligned} H(z) &= \frac{\sqrt{2}}{2} \begin{bmatrix} a_0 & 0 \\ 0 & b_0 \end{bmatrix} V_2(z^2) \cdots V_N(z^2) U_0 U(z^2) \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix}, \\ G(z) &= \frac{\sqrt{2}}{2} \begin{bmatrix} 0 & b_1 \\ a_1 & 0 \end{bmatrix} V_2(z^2) \cdots V_N(z^2) U_0 U(z^2) \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix}, \end{aligned} \tag{4.3}$$

where V_j are the matrices defined by (2.8) *with $v_j \in O(r)$, a_0, b_1 and a_1, b_0 are $s \times r$ and $(r - s) \times r$ matrices, respectively, with $[a_0^T, a_1^T], [b_0^T, b_1^T] \in O(r)$, and U_0 and $U(z)$ are the matrices defined by* (2.2) *and* (2.12) *with $u \in O(m)$, respectively.*

Let M_1 be the permutation matrix defined by

$$M_1 := \operatorname{diag}\left(I_s, \begin{bmatrix} 0 & I_r \\ I_{r-s} & 0 \end{bmatrix} \right).$$

One can easily show that for H, G given by (4.2) and (4.3), respectively, they can also be written in the forms of

$$\begin{bmatrix} H(z) \\ G(z) \end{bmatrix} = \frac{1}{2} M_1 V_1(z^2) \cdots V_N(z^2) \begin{bmatrix} A & AS_0 \\ B & -BS_0 \end{bmatrix} \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix} \tag{4.4}$$

and

$$\begin{bmatrix} H(z) \\ G(z) \end{bmatrix} = \frac{1}{2} M_1 V_2(z^2) \cdots V_N(z^2) \begin{bmatrix} A & AS_0 \\ B & -BS_0 \end{bmatrix} U(z^2) \begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix}, \tag{4.5}$$

where $A, B \in O(r)$.

Parametric expressions of causal orthogonal multifilter banks (4.4) and (4.5) were provided in [6]. It was shown in [7] that the factorization (4.4) is complete. Theorem 4.2 shows that the factorization (4.5) is also complete. By the completeness of the factorization (4.4) and the equivalence of forms (4.2) and (4.4), Theorem 4.1 is in fact not new. For completeness of this paper, the sketch of its proof is provided here.

Proof of Theorem 4.1. Clearly if $\{H, G\}$ is given by (4.2), then it is a causal symmetric orthogonal filter bank. Conversely, let E be the polyphase matrix of H, G. Then E satisfies

$$z^{-N}\mathrm{diag}(S_0, -S_0)E(z^{-1})\begin{bmatrix} 0 & S_0 \\ S_0 & 0 \end{bmatrix} = E(z). \tag{4.6}$$

Define $E_1(z) := E(z)U_0^T$, where U_0 is the matrix defined by (2.2). Then E_1 satisfies

$$z^{-N}\mathrm{diag}(S_0, -S_0)E_1(z^{-1})\mathrm{diag}(I_r, -I_r) = E_1(z).$$

Write

$$E_1(z) = e_0 + \cdots + e_N z^{-N}.$$

By the symmetry of E, $e_N = \mathrm{diag}(S_0, -S_0)e_0\mathrm{diag}(I_r, -I_r)$. By the paraunitariness of E, $e_0 e_N^T = 0$. Thus, $e_0\mathrm{diag}(I_r, -I_r)e_0^T = 0$. By Lemma 2.1, we can find $v_N \in O(r)$ such that

$$e_0\begin{bmatrix} I_r \\ v_N^T \end{bmatrix} = 0.$$

Let $V_N(z)$ be the matrix defined by (2.8) with $v = v_N$. Then $\tilde{E}_1(z) = E_1(z)V_N(z^{-1})$ is causal, paraunitary, and satisfies

$$z^{-(N-1)}\mathrm{diag}(S_0, -S_0)\widetilde{E}_1(z^{-1})\mathrm{diag}(I_r, -I_r) = \widetilde{E}_1(z).$$

Continuing this process, we can find $v_{N-1}, \ldots, v_1 \in O(r)$ such that $E_1(z)V_N(z^{-1})\cdots V_1(z^{-1})$ is

$$\begin{bmatrix} a_0^T & 0 & 0 & a_1^T \\ 0 & b_0^T & b_1^T & 0 \end{bmatrix}^T,$$

where a_0, b_1 and a_1, b_0 are $s \times r$ and $(r-s) \times r$ matrices, respectively, satisfying $[a_0^T, a_1^T], [b_0^T, b_1^T] \in O(r)$. Thus E can be factorized into

$$E(z) = \begin{bmatrix} a_0^T & 0 & 0 & a_1^T \\ 0 & b_0^T & b_1^T & 0 \end{bmatrix}^T V_1(z)\cdots V_N(z)U_0. \tag{4.7}$$

Hence H, G can be written in the form of (4.2). □

Proof of Theorem 4.2. Clearly if $\{H, G\}$ is given by (4.3), then it is a causal symmetric orthogonal filter bank. Conversely, let E be the polyphase matrix of H, G. Then E satisfies

$$z^{-(N-1)}\mathrm{diag}(S_0, -S_0)E(z^{-1})\mathrm{diag}(z^{-1}S_0, S_0) = E(z).$$

Write

$$E(z) = [e_{0,1}, e_{0,2}] + [e_{1,1}, e_{1,2}]z^{-1} + \cdots + [e_{N,1}, e_{N,2}]z^{-N},$$

where $e_{j,1}, e_{j,2}$ are $2r \times r$ matices. Then

$$e_{N,2} = 0, \quad e_{N,1} = \mathrm{diag}(S_0, -S_0)e_{0,1}S_0.$$

By the paraunitariness of E, $e_{0,1}e_{N,1}^T = 0$. Thus, $e_{0,1}S_0e_{0,1}^T = 0$. By Lemma 2.1, we can find $u_0 \in O(m)$ such that

$$e_{0,1}\begin{bmatrix} I_m \\ u_0^T \end{bmatrix} = 0.$$

Let $U(z)$ be the matrix defined by (2.12) with $u = u_0$. Then $\tilde{E}(z) = E(z)U(z^{-1})^T$ is causal, paraunitary, and satisfies (4.6) with $N-1$. By the proof of Theorem 4.1, $\tilde{E}$ can be factorized into the product (4.7) with $N-1$. Thus H, G can be factorized into the form of (4.3). ∎

4.2. Filter banks with different symmetric centers. Suppose $H(z) = \sum_{k=0}^{2\gamma+1} h_k z^{-k}, G(z) = \sum_{k=0}^{2\gamma+1} g_k z^{-k}$ satisfy (1.3), (1.1), and

$$\begin{aligned} &z^{-(2\gamma+1)}\mathrm{diag}(S_0z^2, s_0)H(z^{-1})\mathrm{diag}(S_0, s_0z) = H(z), \\ &z^{-(2\gamma+1)}\mathrm{diag}(S_1z^2, S_2)G(z^{-1})\mathrm{diag}(S_0, s_0z) = G(z), \end{aligned} \tag{4.8}$$

where $s_0 = \pm 1$, S_0, S_1, S_2 are diagonal matrices with diagonal entries 1 or -1.

PROPOSITION 4.3. *Suppose a causal multifilter bank $\{H, G\}$ is orthogonal and satisfies* (4.8). *Then*

$$\mathrm{Tr}(S_0) + \mathrm{Tr}(S_1) = s_0, \quad \mathrm{Tr}(S_2) = 0.$$

Proof. By (4.8),

$$\begin{aligned} &\mathrm{diag}(S_0, s_0, S_1, S_2)\begin{bmatrix} H(1) & H(-1) \\ G(1) & G(-1) \end{bmatrix}\mathrm{diag}(S_0, s_0, S_0, -s_0) \\ &= \begin{bmatrix} H(1) & H(-1) \\ G(1) & G(-1) \end{bmatrix} \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} &(-1)^\gamma i\mathrm{diag}(S_0, -s_0, S_1, -S_2)\begin{bmatrix} H(-i) & H(i) \\ G(-i) & G(i) \end{bmatrix}\mathrm{diag}(S_0, s_0i, S_1, -s_0i) \\ &= \begin{bmatrix} H(-i) & H(i) \\ G(-i) & G(i) \end{bmatrix}\begin{bmatrix} & I_r \\ I_r & \end{bmatrix}. \end{aligned} \tag{4.10}$$

By (4.9), $\mathrm{Tr}(S_0) + \mathrm{Tr}(S_1) + \mathrm{Tr}(S_2) = s_0$, and by (4.10), $\mathrm{Tr}(S_0) + \mathrm{Tr}(S_1) - \mathrm{Tr}(S_2) = s_0$. Thus, $\mathrm{Tr}(S_0) + \mathrm{Tr}(S_1) = s_0$ and $\mathrm{Tr}(S_2) = 0$. ∎

In the following we assume that $s_0 = 1$ and suppose

$$S_0 = \mathrm{diag}(I_s, -I_{r-s-1}), \quad S_1 = \mathrm{diag}(I_{r-p-s}, -I_{s-p}), \quad S_2 = \mathrm{diag}(I_p, -I_p)$$

for some $s, p \in \mathbb{N}_0$ with $s \le p, 2p \le r$. Let E be the polyphase matrix of H, G. Then E satisfies

$$z^{-\gamma}\mathrm{diag}(S_0z, 1, S_1z, S_2)E(z^{-1})\mathrm{diag}(J_0, z) = E(z),$$

where J_0 is the matrix defined by (3.2).

Let M_2 be such a permutation matrix that

$$M_2\mathrm{diag}(S_0z, 1, S_1z, S_2)M_2 = \mathrm{diag}(D_{r-p}z, D_{p+1}).$$

Let R_1 be the matrix defined by (3.4) and M_0 be such a permutation matrix that (3.5) holds. Denote

$$\mathcal{E}(z) := M_2E(z)\mathrm{diag}(R_1, 1)M_0.$$

Then $\mathcal{E}$ is causal, paraunitary, and satisfies

$$z^{-\gamma}\mathrm{diag}(D_{r-p}z, D_{p+1})\mathcal{E}(z^{-1})\mathrm{diag}(z, D_r) = \mathcal{E}(z). \tag{4.11}$$

In the following we discuss the factorization of $\mathcal{E}$. First we consider the case $\gamma = 1$. For this we introduce a paraunitary matrix $\mathcal{W}(z)$ defined as follows. For $u =: \left[\begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix}\right] \in O(r), v =: \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right] \in O(r-1)$, where u_1, u_2, v_1, and v_2 are $(r-p)\times r$, $p\times r$, $(r-p-1)\times(r-1)$, and $p\times(r-1)$ matrices, respectively, define

$$\mathcal{W}(z) := \frac{1}{2}\begin{bmatrix} 0 & 2u_1 & 0 \\ 0 & 0 & 2v_1 \\ 2 & 0 & 0 \\ 0 & u_2 & v_2 \\ 0 & u_2 & v_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & u_2 & -v_2 \\ 0 & -u_2 & v_2 \end{bmatrix} z^{-1}. \tag{4.12}$$

One can show that $\mathcal{W}$ is paraunitary and satisfies (4.11) for $\gamma = 1$.

Proposition 4.4. *A causal paraunitary $\mathcal{E}$ satisfies* (4.11) *for $\gamma = 1$ if and only if it can be written as*

$$\mathcal{E}(z) = \mathrm{diag}(I_{2r-2p-1}, c, I_p)\mathcal{W}(z), \quad c \in O(p+1). \tag{4.13}$$

Proof. It is clear that if $\mathcal{E}$ is given by (4.13), then it is paraunitary and satisfies (4.11) for $\gamma = 1$. Conversely, condition (4.11) implies that $\mathcal{E}$ has the form of

$$\begin{bmatrix} 0 & L_1 & 0 \\ 0 & 0 & L_2 \\ c_0 & & d_0 \end{bmatrix} + \begin{bmatrix} 0 & & 0 & 0 \\ 0 & & 0 & 0 \\ 0 & & D_{p+1}d_0D_r & \end{bmatrix} z^{-1},$$

where L_1, L_2 are $(r-p)\times r$ and $(r-p-1)\times(r-1)$ matrices, respectively, c_0 is $(2p+1)\times 1$ column vector satisfying $D_{p+1}c_0 = c_0$, and d_0 is a $(2p+1)\times(2r-1)$ matrix. The paraunitariness of $\mathcal{E}$ implies that

$$L_1L_1^T = I_{r-p}, \quad L_2L_2^T = I_{r-p-1}, \quad \mathrm{diag}(L_1, L_2)d_0^T = 0, \quad d_0D_rd_0^T = 0. \tag{4.14}$$

Again let $\{x_1, x_2, \ldots, x_n\}$ be an o.n. basis for the columns of d_0^T, where $n \le r-1$ is the rank of d_0. Write

$$[x_1, \ldots, x_n] = \begin{bmatrix} Y_1 \\ Z_1 \end{bmatrix}, \quad Y_1, Z_1 \text{ are } r\times n \text{ and } (r-1)\times n \text{ matrices.}$$

Then $\sqrt{2}Y_1, \sqrt{2}Z_1$ are $r\times n$ and $(r-1)\times n$ matrices of o.n. columns, respectively. By (4.14), we know L_1, L_2 are $(r-p)\times r$ and $(r-p-1)\times(r-1)$ matrices of o.n. columns, and $L_1Y_1 = 0, L_2Z_1 = 0$. Thus $[\sqrt{2}Y_1, L_1^T], [\sqrt{2}Z_1, L_2^T]$ are $r\times(n+r-p)$

and $(r-1)\times(n+r-p-1)$ matrices of orthonormal columns, respectively. Thus, $n \le p$. Let Y_2, Z_2 be $r\times(p-n)$ and $(r-1)\times(p-n)$ matrices such that

$$[\sqrt{2}Y_1, L_1^T, \sqrt{2}Y_2] \in O(r), \quad [\sqrt{2}Z_1, L_2^T, \sqrt{2}Z_2] \in O(r-1).$$

Thus,

$$\begin{bmatrix} Y_1^T & -Z_1^T \\ Y_2^T & -Z_2^T \end{bmatrix} x_j = 0.$$

Therefore,

$$d_0 \begin{bmatrix} Y_1 & Y_2 \\ -Z_1 & -Z_2 \end{bmatrix} = 0, \quad \operatorname{diag}(L_1, L_2) \begin{bmatrix} Y_1 & Y_2 \\ -Z_1 & -Z_2 \end{bmatrix} = 0.$$

Let $\mathcal{W}(z)$ be the matrix defined by (4.12) with $u_1 = L_1, v_1 = L_2$, and $u_2 = \sqrt{2}[Y_1, Y_2]^T, v_2 = \sqrt{2}[Z_1, Z_2]^T$. Then $\mathcal{E}_0(z) := \mathcal{E}(z)\mathcal{W}(z^{-1})^T$ is a causal paraunitary matrix satisfying

$$\operatorname{diag}(D_{r-p}z, D_{p+1})\mathcal{E}_0(z^{-1})\operatorname{diag}(D_{r-p}z^{-1}, D_{p+1}) = \mathcal{E}_0(z),$$

which implies that $\mathcal{E}_0(z)$ is $\operatorname{diag}(a,b,c,d)$ for $a \in O(r-p), b \in O(r-p-1), c \in O(p+1), d \in O(p)$. One can check that some parameters in a, b, c, d are redundant, and we can choose $a = I_{r-p}, b = I_{r-p-1}, d = I_p$. Hence $\mathcal{E}$ can be written in the form of (4.13). ∎

Now let us consider the case $\gamma \ge 2$.

LEMMA 4.5. *If a causal, paraunitary $\mathcal{E}(z) = e_0 + e_1 z^{-1} + \cdots e_\gamma z^{-\gamma}$ satisfies* (4.11) *for $\gamma \ge 2$, then there exists $w_\gamma \in O(r)$ such that*

$$e_0[0, \tilde{w}_\gamma, -I_{r-1}]^T = 0,$$

where $\tilde{w}_\gamma$ is the matrix consisting of the first $r-1$ rows of w_γ.

Proof. By (4.11), $\mathcal{E}$ can be written as

$$\begin{aligned}\mathcal{E} = &\begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} + \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \\ &+ \cdots + \begin{bmatrix} 0 & D_{r-p}b_0D_r \\ D_{p+1}c_0 & D_{p+1}d_1D_r \end{bmatrix} z^{-(\gamma-1)} + \begin{bmatrix} 0 & 0 \\ 0 & D_{p+1}d_0D_r \end{bmatrix} z^{-\gamma},\end{aligned}$$

where a_j, b_j, c_j, and d_j are $(2r-2p-1)\times 1$, $(2r-2p-1)\times(2r-1)$, $(2p+1)\times 1$, and $(2p+1)\times(2r-1)$ matrices. The paraunitariness of $\mathcal{E}$ implies that

$$\begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & D_{p+1}d_0D_r \end{bmatrix}^T = 0$$

and

$$\begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \begin{bmatrix} 0 & D_{r-p}b_0D_r \\ D_{p+1}c_0 & D_{p+1}d_1D_r \end{bmatrix}^T + \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & D_{p+1}d_0D_r \end{bmatrix}^T = 0.$$

Thus,

$$\begin{bmatrix} b_0 \\ d_0 \end{bmatrix} D_r[b_0^T, d_0^T] = 0.$$

Therefore, by Lemma 2.1, there exists $w_\gamma \in O(r)$ such that

$$\begin{bmatrix} b_0 \\ d_0 \end{bmatrix} \begin{bmatrix} w_\gamma^T \\ (I_{r-1}, 0) \end{bmatrix} = 0.$$

The proof of Lemma 4.5 is complete. □

By Lemma 4.5, for a causal, paraunitary $\mathcal{E}$ satisfying (4.11) for $\gamma \geq 2$, there exists $w_\gamma \in O(r)$ such that $\mathcal{E}(z)W_\gamma(z^{-1})^T$ is causal, paraunitary, and satisfies (4.11) for $\gamma - 1$, where $W_\gamma(z)$ is the matrix defined by (3.9) with $w = w_\gamma$. In this way, we can find $w_{\gamma-1}, \ldots, w_2 \in O(r)$ such that $\mathcal{E}(z)W_\gamma(z^{-1})^T \cdots W_2(z^{-1})^T$ is causal, paraunitary, and satisfies (4.11) for $\gamma = 1$. This together with Proposition 4.4 leads to the following theorems.

THEOREM 4.6. *A causal paraunitary FIR $\mathcal{E}$ satisfies* (4.11) *if and only if it can be factorized in the form of*

$$\mathcal{E}(z) = \mathrm{diag}(I_{2r-2p-1}, c, I_p)\mathcal{W}(z)W_2(z)\cdots W_\gamma(z),$$

where $c \in O(p+1)$, $\mathcal{W}$ is the matrix defined (4.12) *for $u \in O(r), v \in O(r-1)$, and $W_j(z)$ are the matrices defined by* (3.9) *with $w_j \in O(r)$.*

THEOREM 4.7. *A causal FIR multifilter bank $\{H, G\}$ is orthogonal and satisfies* (4.8) *if and only if H, G can be factorized in the form*

$$\begin{bmatrix} H(z) \\ G(z) \end{bmatrix} = \frac{\sqrt{2}}{2} M_2 \mathrm{diag}(I_{2r-2p-1}, c, I_p)\mathcal{W}(z^2)W_2(z^2)\cdots W_\gamma(z^2)M_0\mathrm{diag}(R_1^T, 1)\begin{bmatrix} I_r \\ I_r z^{-1} \end{bmatrix},$$

where M_2, R_1, M_0 are the matrices defined above, $c \in O(p+1)$, $\mathcal{W}$ is the matrix defined (4.12) *for $u \in O(r), v \in O(r-1)$, and $W_j(z)$ are the matrices defined by* (3.9) *with $w_j \in O(r)$.*

For the special case $r = 2, s = p = 1$, another form of the complete factorization of orthogonal $\{H, G\}$ satisfying (4.8) was obtained in [8]. By the parametric expression of symmetric multifilter banks, one can construct multiwavelets with various properties. We will carry out such work elsewhere.

REFERENCES

[1] C. K. CHUI AND J. LIAN, *A study on orthonormal multi-wavelets*, J. Appl. Numer. Math., 20 (1996), pp. 273–298.

[2] J. S. GERONIMO, D. P. HARDIN, AND P. R. MASSOPUST, *Fractal functions and wavelet expansions based on several scaling functions*, J. Approx. Theory, 78 (1994), pp. 373–401.

[3] R. Q. JIA AND Z. W. SHEN, *Multiresolution and wavelets,* Proc. Edinburgh Math. Soc. (2), 7 (1994), pp. 271–300.

[4] Q. T. JIANG, *On the regularity of matrix refinable functions*, SIAM J. Math. Anal., 29 (1998), pp. 1157–1176.

[5] Q. T. JIANG, *On the design of multifilter banks and orthogonal multiwavelet bases*, IEEE Trans. Signal Process., 46 (1998), pp. 3292–3303.

[6] Q. T. JIANG, *Multivariate matrix refinable functions with arbitrary matrix dilation*, Trans. Amer. Math. Soc., 351 (1999), pp. 2407-2438.

[7] Q. T. JIANG, *Parameterization of M–channel orthogonal multifilter banks*, Adv. Comput. Math., 12 (2000), pp. 189–211.

[8] Q. T. JIANG, *Parameterizations of symmetric orthogonal multifilter banks with different filter lengths*, Linear Algebra Appl., 311 (2000), pp. 79–96.

[9] W. Lawton, S. L. Lee, and Z. W. Shen, *An algorithm for matrix extension and wavelet construction*, Math. Comp., 65 (1996), pp. 723–737.
[10] H. Park, *A Computational Theory of Laurent Polynomial Rings and Multidimensional FIR Systems*, Ph.D. dissertation, University of California, Berkeley, 1995.
[11] P. Rieder, *Parametrization of symmetric multiwavelets*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Munich, IEEE, Los Alamitos, 1997, pp. 2461–2464.
[12] Z. W. Shen, *Refinable function vectors*, SIAM J. Math. Anal., 29 (1998), pp. 235–250.
[13] W. So and J. Z. Wang, *Estimating the support of a scaling vector*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 66–73.
[14] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996.
[15] R. Turcajova, *Construction of symmetric biorthogonal multiwavelets by lifting*, in Proceedings of Wavelet Applications in Signal and Image Processing VII, Denver, 1999, SPIE Proc. 3813, 1999, pp. 443–454.
[16] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, 1993.
[17] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice-Hall Signal Processing Series 48, Prentice-Hall, Englewood Cliffs, 1995.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 187–208
© 2001 Society for Industrial and Applied Mathematics

STRUCTURED PSEUDOSPECTRA FOR POLYNOMIAL EIGENVALUE PROBLEMS, WITH APPLICATIONS*

FRANÇOISE TISSEUR† AND NICHOLAS J. HIGHAM†

Abstract. Pseudospectra associated with the standard and generalized eigenvalue problems have been widely investigated in recent years. We extend the usual definitions in two respects, by treating the polynomial eigenvalue problem and by allowing structured perturbations of a type arising in control theory. We explore connections between structured pseudospectra, structured backward errors, and structured stability radii. Two main approaches for computing pseudospectra are described. One is based on a transfer function and employs a generalized Schur decomposition of the companion form pencil. The other, specific to quadratic polynomials, finds a solvent of the associated quadratic matrix equation and thereby factorizes the quadratic λ-matrix. Possible approaches for large, sparse problems are also outlined. A collection of examples from vibrating systems, control theory, acoustics, and fluid mechanics is given to illustrate the techniques.

Key words. polynomial eigenvalue problem, λ-matrix, matrix polynomial, pseudospectrum, stability radius, backward error, transfer function, quadratic matrix equation, solvent, structured perturbations, Orr–Sommerfeld equation

AMS subject classifications. 65F15, 15A22

PII. S0895479800371451

1. Introduction.

Pseudospectra are an established tool for gaining insight into the sensitivity of the eigenvalues of a matrix to perturbations. Their use is widespread with applications in areas such as fluid mechanics, Markov chains, and control theory. Most of the existing work is for the standard eigenproblem, although attention has also been given to matrix pencils [4], [23], [33], [40], [46]. The literature on pseudospectra is large and growing. We refer to Trefethen [41], [42], [43] for thorough surveys of pseudospectra and their computation for a single matrix; see also the Web site [3].

In this work we investigate pseudospectra for polynomial matrices (or λ-matrices)

$$P(\lambda) = \lambda^m A_m + \lambda^{m-1} A_{m-1} + \cdots + A_0, \tag{1.1}$$

where $A_k \in \mathbb{C}^{n\times n}$, $k = 0\colon m$. We first define the ϵ-pseudospectrum and obtain a computationally useful characterization. We examine the relation between the backward error of an approximate eigenpair of the polynomial eigenvalue problem associated with (1.1), the ϵ-pseudospectrum, and the stability radius. We consider both unstructured perturbations and structured perturbations of a type commonly used in control theory.

Existing methods for the computation of pseudospectra in the case $m = 1$ (the standard and generalized eigenvalue problems) do not generalize straightforwardly to matrix polynomials. We develop two techniques that allow efficient computation for $m > 1$. A transfer function approach employs the generalized Schur decomposition of the $mn \times mn$ companion form pencil. For the quadratic case ($m = 2$) an alternative

*Received by the editors May 1, 2000; accepted for publication (in revised form) by M. Chu February 7, 2001; published electronically June 8, 2001. This work was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

http://www.siam.org/journals/simax/23-1/37145.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/). The work of the second author was supported by a Royal Society Leverhulme Trust Senior Research Fellowship.

solvent approach computes a solvent of the associated quadratic matrix equation $A_2X^2 + A_1X + A_0 = 0$ and thereby factorizes the quadratic λ-matrix; it works all the time with $n \times n$ matrices once the solvent has been obtained. We give a detailed comparison of these approaches and also outline techniques that can be efficiently used when n is so large as to preclude factorizations.

In the last section, we illustrate our theory and techniques on applications from vibrating systems, control theory, acoustics, and fluid mechanics.

2. Pseudospectra.

2.1. Definition. The polynomial eigenvalue problem is to find the solutions (x, λ) of

$$P(\lambda)x = 0, \tag{2.1}$$

where $P(\lambda)$ is of the form (1.1). If $x \neq 0$ then λ is called an eigenvalue and x the corresponding right eigenvector; $y \neq 0$ is a left eigenvector if $y^*P(\lambda) = 0$. The set of eigenvalues of P is denoted by $\Lambda(P)$. When A_m is nonsingular P has mn finite eigenvalues, while if A_m is singular P has infinite eigenvalues. Good references for the theory of λ-matrices are [8], [20], [21], [37].

Throughout this paper we assume that P has only finite eigenvalues (and pseudoeigenvalues); how to deal with infinite eigenvalues is described in [16].

For notational convenience, we introduce

$$\Delta P(\lambda) = \lambda^m \Delta A_m + \lambda^{m-1} \Delta A_{m-1} + \cdots + \Delta A_0. \tag{2.2}$$

We define the ϵ-pseudospectrum of P by

$$\begin{aligned} \Lambda_\epsilon(P) = \big\{ \lambda \in \mathbb{C} : (P(\lambda) + \Delta P(\lambda))x = 0 \ \text{ for some } x \neq 0 \text{ and } \Delta P(\lambda) \\ \text{with } \|\Delta A_k\| \leq \epsilon \alpha_k,\ k = 0\colon m \big\}. \end{aligned} \tag{2.3}$$

Here the α_k are nonnegative parameters that allow freedom in how perturbations are measured—for example, in an absolute sense ($\alpha_k \equiv 1$) or a relative sense ($\alpha_k = \|A_k\|$). By setting $\alpha_k = 0$ we can force $\Delta A_k = 0$ and thus keep A_k unperturbed. The norm, here and throughout, is any subordinate matrix norm. Occasionally, we will specialize to the norm $\|\cdot\|_p$ subordinate to the Hölder vector p-norm.

When $P(\lambda) = A - \lambda I$, $\Delta P(\lambda) = \Delta A$ and $\alpha_1 = 1$, definition (2.3) reduces to the standard definition of ϵ-pseudospectrum of a single matrix:

$$\Lambda_\epsilon(A) = \big\{ \lambda \in \mathbb{C} : \lambda \in \Lambda(A + \Delta A) \ \text{ for some } \Delta A \text{ with } \|\Delta A\| \leq \epsilon \big\}. \tag{2.4}$$

It is well known [43] that (2.4) is equivalent to

$$\Lambda_\epsilon(A) = \big\{ \lambda \in \mathbb{C} : \|(\lambda I - A)^{-1}\| \geq \epsilon^{-1} \big\}.$$

In the following lemma, we provide a generalization of this equivalence for the ϵ-pseudospectrum of P.

LEMMA 2.1.

$$\Lambda_\epsilon(P) = \big\{ \lambda \in \mathbb{C} : \|P(\lambda)^{-1}\| \geq (\epsilon\, p(|\lambda|))^{-1} \big\},$$

where $p(x) = \sum_{k=0}^m \alpha_k x^k$.

Proof. Let $\mathcal{S}$ denote the set on the right-hand side of the claimed equality. We first show that $\lambda \in \Lambda_\epsilon(P)$ implies $\lambda \in \mathcal{S}$. If λ is an eigenvalue of P this is immediate, so we can assume that λ is not an eigenvalue of P and hence that $P(\lambda)$ is nonsingular. Since

$$P(\lambda) + \Delta P(\lambda) = P(\lambda)\big(I + P(\lambda)^{-1}\Delta P(\lambda)\big)$$

is singular, we have

$$1 \le \|P(\lambda)^{-1}\Delta P(\lambda)\| \le \|P(\lambda)^{-1}\| \left(\sum_{k=0}^{m} |\lambda|^k \alpha_k \epsilon\right) = \|P(\lambda)^{-1}\|\, \epsilon\, p(|\lambda|),$$

so that $\lambda \in \mathcal{S}$.

Now let $\lambda \in \mathcal{S}$. Again we can assume that $P = P(\lambda)$ is nonsingular. Choose y with $\|y\| = 1$ so that $\|P^{-1}y\| = \|P^{-1}\|$ and let $x = P^{-1}y/\|P^{-1}\|$, so that $\|x\| = 1$. Then there exists a matrix H with $\|H\| = 1$ such that $Hx = y$ (see, for example, [11, Lem. 6.3]). Let $E = -H/\|P^{-1}\|$. Then

$$(P + E)x = \frac{y}{\|P^{-1}\|} - \frac{y}{\|P^{-1}\|} = 0$$

and

$$\|E\| = 1/\|P^{-1}\| \le \epsilon\, p(|\lambda|).$$

We now apportion E between the A_k by defining

$$\Delta A_k = \operatorname{sign}(\lambda^k)\alpha_k p(|\lambda|)^{-1} E,$$

where for complex z we define

$$\operatorname{sign}(z) = \begin{cases} \overline{z}/|z|, & z \ne 0, \\ 0, & z = 0. \end{cases}$$

Then

$$\Delta P(\lambda) = \sum_{k=0}^{m} \lambda^k \Delta A_k = \left(\sum_{k=0}^{m} |\lambda|^k \alpha_k\right) p(|\lambda|)^{-1} E = E$$

and $\|\Delta A_k\| \le \alpha_k \epsilon$, $k = 0\colon m$. Hence $\lambda \in \Lambda_\epsilon(P)$. ∎

The characterization of the ϵ-pseudospectrum in Lemma 2.1 will be the basis of our algorithms for computing pseudospectra.

We note that for $n = 1$, $\Lambda_\epsilon(P)$ is the root neighborhood of the polynomial P introduced by Mosier [28], that is, the set of all polynomials obtained by elementwise perturbations of P of size at most ϵ. This set is also investigated by Toh and Trefethen [38], who call it the ϵ-pseudozero set.

2.2. Connection with backward error. A natural definition of the normwise backward error of an approximate eigenpair (x, λ) of (2.1) is

$$\eta(x, \lambda) := \min\{\, \epsilon : (P(\lambda) + \Delta P(\lambda))x = 0,\ \|\Delta A_k\| \le \epsilon\alpha_k,\ k = 0\colon m \,\}, \tag{2.5}$$

and the backward error for an approximate eigenvalue λ is given by

$$\eta(\lambda) := \min_{x \ne 0} \eta(x, \lambda). \tag{2.6}$$

By comparing the definitions (2.3) and (2.6) it is clear that the ϵ-pseudospectrum can be expressed in terms of the backward error of λ as

$$\Lambda_\epsilon(P) = \{\, \lambda \in \mathbb{C} : \eta(\lambda) \le \epsilon \,\}. \tag{2.7}$$

The following lemma gives an explicit expression for $\eta(x,\lambda)$ and $\eta(\lambda)$. This lemma generalizes results given in [36] for the 2-norm and earlier in [5], [10] for the generalized eigenvalue problem.

LEMMA 2.2. *The normwise backward error $\eta(x,\lambda)$ is given for $x \neq 0$ by*

$$\eta(x,\lambda) = \frac{\|r\|}{p(|\lambda|)\|x\|}, \tag{2.8}$$

where $r = P(\lambda)x$ and $p(x) = \sum_{k=0}^m \alpha_k x^k$. If λ is not an eigenvalue of P then

$$\eta(\lambda) = \frac{1}{p(|\lambda|)\|P(\lambda)^{-1}\|}. \tag{2.9}$$

Proof. It is straightforward to show that the right-hand side of (2.8) is a lower bound for $\eta(x,\lambda)$. That the lower bound is attained is proved using a construction for ΔA_k similar to that in the proof of Lemma 2.1. The expression (2.9) follows on using the equality, for nonsingular $C \in \mathbb{C}^{n\times n}$, $\min_{x\neq 0} \|Cx\|/\|x\| = \|C^{-1}\|^{-1}$. ∎

We observe that the expressions (2.7) and (2.9) lead to another proof of Lemma 2.1.

2.3. Structured perturbations. We now suppose that $P(\lambda)$ is subject to structured perturbations that can be expressed as

$$[\,\Delta A_0, \quad \ldots, \quad \Delta A_m\,] = D\Theta\,[\,E_0, \quad \ldots, \quad E_m\,], \tag{2.10}$$

with $D \in \mathbb{C}^{n\times s}$, $\Theta \in \mathbb{C}^{s\times t}$, and $E = [\,E_0, \quad \ldots, \quad E_m\,] \in \mathbb{C}^{t\times n(m+1)}$. The matrices D and E are fixed and assumed to be of full rank, and they define the structure of the perturbations; Θ is an arbitrary matrix whose elements are the free parameters. Note that $\Delta A_0, \ldots, \Delta A_m$ in (2.10) are linear functions of the parameters in Θ, but that not all linear functions can be represented in this form. We choose this particular structure for the perturbations because it is one commonly used in control theory [17], [18], [30] and it leads to more tractable formulae than a fully general approach. Note, for instance, that the system

$$\dot{x}(t) = (A + D\Theta E)x(t), \quad t > 0$$

(which leads to a polynomial eigenvalue problem with $m = 1$), may be interpreted as a closed loop system with unknown static linear output feedback Θ; see Figure 2.1.

Note that unstructured perturbations are represented by the special case of (2.10) with

$$s = n, \quad t = n(m+1), \quad D = I_n, \quad \Theta = [\,\Delta A_0, \ldots, \Delta A_m\,], \quad E = I_{n(m+1)}. \tag{2.11}$$

For notational convenience, we introduce

$$E(\lambda) = E\,[\,I_n, \quad \lambda I_n, \quad \ldots, \quad \lambda^m I_n\,]^T = \lambda^m E_m + \lambda^{m-1}E_{m-1} + \cdots + E_0.$$

Corresponding to (2.10) we have the following definition of structured backward error for an approximate eigenpair (x,λ):

$$\eta(x,\lambda; D, E) := \min_{\Theta\in\mathbb{C}^{s\times t}} \{\, \|\Theta\| : (P(\lambda) + \Delta P(\lambda))x = 0,\ \Delta P(\lambda) = D\Theta E(\lambda) \,\},$$

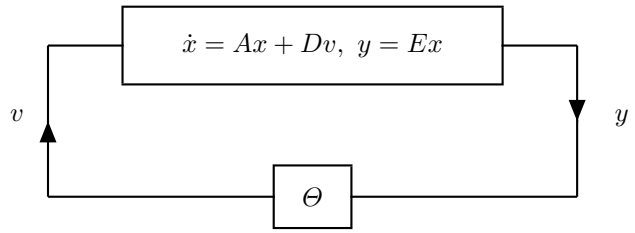


FIG. 2.1. *Closed loop system with unknown static linear output feedback* Θ.

and the backward error for an approximate eigenvalue is

$$\eta(\lambda; D, E) := \min_{x \neq 0} \eta(x, \lambda; D, E).$$

In the next result we use a superscript "+" to denote the pseudo-inverse [9].

LEMMA 2.3. *The structured backward error* $\eta(x, \lambda; D, E)$ *in the Frobenius norm is given by*

$$\eta_F(x, \lambda; D, E) = \|D^+ P(\lambda)\, x\, [E(\lambda)x]^+\|_F \tag{2.12}$$

if the system

$$D\Theta E(\lambda)x = -P(\lambda)x \tag{2.13}$$

is consistent; otherwise $\eta_F(x, \lambda; D, E)$ *is infinite.*

Proof. It is immediate that $\eta_F(x, \lambda; D, E)$ is the Frobenius norm of the minimum Frobenius norm solution to (2.13). The result follows from the fact that $X = A^+CB^+$ is the solution of minimum Frobenius norm to the consistent system $AXB = C$ [31, sect. 3.4.8]. ∎

To gain some insight into the expression (2.12) we consider the case of unstructured but weighted perturbations, as in (2.11) but with

$$E = \operatorname{diag}(\alpha_0 I_n, \ldots, \alpha_m I_n) =: I^{\alpha}_{n(m+1)}, \qquad E(\lambda) = [\, \alpha_0 I_n, \quad \ldots, \quad \alpha_m \lambda^m I_n \,]^T.$$

The system (2.13) is now trivially consistent and (2.12) gives

$$\begin{aligned} \eta_F(x, \lambda; I_n, I^{\alpha}_{n(m+1)}) &= \left\| P(\lambda) x \begin{bmatrix} \alpha_0 x \\ \vdots \\ \alpha_m \lambda^m x \end{bmatrix}^+ \right\|_F \\ &= \frac{\|P(\lambda)x\|_2}{\left(\sum_{i=0}^m \alpha_i^2 |\lambda|^{2i}\right)^{1/2} \|x\|_2}, \end{aligned} \tag{2.14}$$

using the fact that $\|ab^*\|_F = \|a\|_2 \|b\|_2$ for $a, b \in \mathbb{C}^n$. The expression (2.14) differs from that for $\eta(x, \lambda)$ in (2.8) for the 2-norm only by having the 2-norm of the vector $[\alpha_0 \ \ldots \ \alpha_m \lambda^m]$ rather than the 1-norm in the denominator.

LEMMA 2.4. *If* λ *is not an eigenvalue of* $P(\lambda)$ *then the structured backward error* $\eta(\lambda; D, E)$ *is given by*

$$\eta(\lambda; D, E) = \|E(\lambda)P(\lambda)^{-1}D\|^{-1}. \tag{2.15}$$

Proof. We have

$$\begin{aligned}\eta(\lambda; D, E) &= \min_{x\neq 0} \eta(x, \lambda; D, E)\\ &= \min_{x\neq 0} \min_{\Theta\in\mathbb{C}^{s\times t}} \{\, \|\Theta\| : (P(\lambda) + \Delta P(\lambda))x = 0,\ \Delta P(\lambda) = D\Theta E(\lambda)\,\}\\ &= \min_{\Theta\in\mathbb{C}^{s\times t}} \{\, \|\Theta\| : \det(P(\lambda) + \Delta P(\lambda)) = 0,\ \Delta P(\lambda) = D\Theta E(\lambda)\,\}.\end{aligned}$$

The companion form of $P(\lambda) + \Delta P(\lambda)$ is given by

$$F - \lambda G + \Delta F - \lambda \Delta G,$$

where

$$F = \begin{bmatrix} 0 & I & 0 & \cdots & 0\\ 0 & 0 & I & \ddots & \vdots\\ \vdots & & & \ddots & 0\\ & & & & I\\ -A_0 & -A_1 & -A_2 & \cdots & -A_{m-1}\end{bmatrix}, \qquad G = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & A_m\end{bmatrix}, \tag{2.16}$$

and

$$\Delta F = \begin{bmatrix} & & 0 & \\ -\Delta A_0 & \cdots & -\Delta A_{m-1}\end{bmatrix}, \qquad \Delta G = \begin{bmatrix} 0 & \\ & \Delta A_m\end{bmatrix}.$$

As $\Delta A_i = D\Theta E_i$, we have

$$\Delta F - \lambda \Delta G = \widetilde{D}\Theta\,[\,E_0, \quad \ldots, \quad E_{m-1} + \lambda E_m\,] \quad \text{with} \quad \widetilde{D} = -\begin{bmatrix} 0\\ \vdots\\ 0\\ D\end{bmatrix}.$$

Then, using the identity $\det(I + AB) = \det(I + BA)$, valid whenever both AB and BA are defined [47, p. 54],

$$\begin{aligned}\det(P(\lambda) + \Delta P(\lambda)) = 0 &\Leftrightarrow \det(F - \lambda G + \Delta F - \lambda \Delta G) = 0\\ &\Leftrightarrow \det(I + (F - \lambda G)^{-1}(\Delta F - \lambda \Delta G)) = 0\\ &\Leftrightarrow \det(I + (F - \lambda G)^{-1}\widetilde{D}\Theta\,[\,E_0, \quad \ldots, \quad E_{m-1} + \lambda E_m\,]) = 0\\ &\Leftrightarrow \det(I + \Theta\,[\,E_0, \quad \ldots, \quad E_{m-1} + \lambda E_m\,]\,(F - \lambda G)^{-1}\widetilde{D}) = 0.\end{aligned}$$

Let $M = [\,E_0, \quad \ldots, \quad E_{m-1} + \lambda E_m\,]\,(F - \lambda G)^{-1}\widetilde{D} \in \mathbb{C}^{t\times s}$. Then, using [45, Lem. 1], we have

$$\eta(\lambda; D, E) = \min_{\Theta\in\mathbb{C}^{s\times t}} \{\, \|\Theta\| : \det(I + \Theta M) = 0\,\} = \|M\|^{-1}.$$

But it is easily verified that

$$(F - \lambda G)\begin{bmatrix} P(\lambda)^{-1}D\\ \vdots\\ \lambda^{m-1}P(\lambda)^{-1}D\end{bmatrix} = \widetilde{D},$$

so that $M = E(\lambda)P(\lambda)^{-1}D$. ∎

We define the structured ϵ-pseudospectrum by

$$\Lambda_\epsilon(P; D, E) = \{\, \lambda \in \mathbb{C} : (P(\lambda) + D\Theta E(\lambda))x = 0 \quad \text{for some } x \neq 0,\ \|\Theta\| \leq \epsilon \,\}.$$

Analogously to the unstructured case, $\Lambda_\epsilon(P; D, E) = \{\, \lambda \in \mathbb{C} : \eta(\lambda; D, E) \leq \epsilon \,\}$, and so from Lemma 2.4 we have

$$\Lambda_\epsilon(P; D, E) = \{\, \lambda \in \mathbb{C} : \|E(\lambda)P(\lambda)^{-1}D\| \geq \epsilon^{-1} \,\}, \tag{2.17}$$

which is a generalization of a result of Hinrichsen and Kelb [17, Lem. 2.2] for the ϵ-pseudospectrum of a single matrix.

2.4. Connection between backward error and stability radius. In many mathematical models (e.g., those of a dynamical system) it is required for stability that a matrix has all its eigenvalues in a given open subset $\mathbb{C}_g \neq \emptyset$ of the complex plane. Various stability radii have been defined that measure the ability of a matrix to preserve its stability under perturbations.

We partition the complex plane $\mathbb{C}$ into two disjoint subsets $\mathbb{C}_g$ and $\mathbb{C}_b$, with

$$\mathbb{C} = \mathbb{C}_g \cup \mathbb{C}_b, \quad \mathbb{C}_g \neq \emptyset \text{ an open set.} \tag{2.18}$$

Consider perturbations of the form in (2.10). Following Pappas and Hinrichsen [30] and Genin and Van Dooren [7], we define the complex structured stability radius of the λ-matrix P with respect to the perturbation structure (D, E) and the partition (2.18) by

$$r_{\mathbb{C}}(P; D, E) = \inf_{\Theta \in \mathbb{C}^{s\times t}} \{\, \|\Theta\| : \Lambda(P(\lambda) + \Delta P(\lambda)) \cap \mathbb{C}_b \neq \emptyset,\ \Delta P(\lambda) = D\Theta E(\lambda) \,\}.$$

Let $\partial\mathbb{C}_b$ be the boundary of $\mathbb{C}_b$. By continuity, we have

$$\begin{aligned} r_{\mathbb{C}}(P; D, E) &= \inf_{\Theta \in \mathbb{C}^{s\times t}} \{\, \|\Theta\| : \Lambda(P(\lambda) + D\Theta E(\lambda)) \cap \partial\mathbb{C}_b \neq \emptyset \,\} \\ &= \inf_{\lambda \in \partial\mathbb{C}_b} \inf_{\Theta \in \mathbb{C}^{s\times t}} \{\, \|\Theta\| : \det(P(\lambda) + D\Theta E(\lambda)) = 0 \,\} \\ &= \inf_{\lambda \in \partial\mathbb{C}_b} \inf_{x \neq 0} \inf_{\Theta \in \mathbb{C}^{s\times t}} \{\, \|\Theta\| : (P(\lambda) + D\Theta E(\lambda))x = 0 \,\} \\ &= \inf_{\lambda \in \partial\mathbb{C}_b} \eta(\lambda; D, E). \end{aligned}$$

Thus we have expressed the stability radius as an infimum of the eigenvalue backward error. Using Lemma 2.4 we obtain the following result.

LEMMA 2.5. *If λ is not an eigenvalue of P then*

$$r_{\mathbb{C}}(P; D, E) = \inf_{\lambda \in \partial\mathbb{C}_b} \|E(\lambda)P(\lambda)^{-1}D\|^{-1}$$

and for unstructured perturbations and the p-norm we have

$$r_{\mathbb{C}}(P; I_n, I_{(m+1)n}) = \inf_{\lambda \in \partial\mathbb{C}_b} \Big(\|[1 \ \lambda \ \dots \lambda^m]\|_p \|P(\lambda)^{-1}\|_p \Big)^{-1}.$$

The result for the unstructured case in the second part of this lemma is also obtained by Pappas and Hinrichsen [30, Cor. 2.4] and Genin and Van Dooren [7, Thm. 2].

3. Computation of pseudospectra. In this section, we consider the computation of $\Lambda_\epsilon(P)$, concentrating mainly on the 2-norm. We develop methods for unstructured perturbations and show how they can be extended to structured perturbations of the form in (2.10).

Lemma 2.1 shows that the boundary of $\Lambda_\epsilon(P)$ comprises points z for which the scaled resolvent norm $p(|z|)\|P(z)^{-1}\|$ equals ϵ^{-1}. Hence, as for pseudospectra of a single matrix, we can obtain a graphical representation of the pseudospectra of a polynomial eigenvalue problem by evaluating the scaled resolvent norm on a grid of points z in the complex plane and sending the results to a contour plotter. We refer to Trefethen [42] for a survey of the state of the art in computation of pseudospectra of a single matrix.

The region of interest in the complex plane will usually be determined by the underlying application or by prior knowledge of the spectrum of P. In the absence of such information we can select a region guaranteed to enclose the spectrum. If A_m is nonsingular (so that all eigenvalues are finite) then by applying the result "$\max_j |\lambda_j(A)| \le \|A\|$" to the companion form (2.16) we deduce that

$$\max_j |\lambda_j(P)| \le 1 + \sum_{j=0}^{m-1} \|A_m^{-1} A_j\|_p$$

for any p-norm. Alternatively, we could bound $\max_j |\lambda_j(P)|$ by the largest absolute value of a point in the numerical range of P [24], but computation of this number is itself a nontrivial problem. For much more on bounding the eigenvalues of matrix polynomials see [15].

For the 2-norm, $\|P(z)^{-1}\|_2 = (\sigma_{\min}(P(z)))^{-1}$, where $\sigma_{\min}$ denotes the smallest singular value. If the grid is $\nu \times \nu$ and $\sigma_{\min}$ is computed using the Golub–Reinsch SVD algorithm then the whole computation requires roughly

$$\nu^2(8n^3/3 + n^2 m) \text{ flops}, \tag{3.1}$$

which is prohibitively expensive for matrices of large dimension and a fine grid. Using the fact that $\sigma_{\min}(P(z))$ is the square root of $\lambda_{\min}(P(z)^* P(z))$, we can approximate $\|P(z)^{-1}\|_2$ with the power iteration or Lanczos iteration applied to $P(z)^{-1}P(z)^{-*}$. In the case of a single matrix, Lui [25] introduced the idea of using the Schur form of A in order to speed up the computation of $\lambda_{\min}((A - zI)^*(A - zI))$. Unfortunately, for matrix polynomials of degree $m \ge 2$ no analogue of the Schur form exists (that is, at most two general matrices can be simultaneously reduced to triangular form). We therefore look for other ways to efficiently evaluate or approximate $\|P(z)^{-1}\|$ for many different z.

3.1. Transfer function approach. The idea of writing pseudospectra in terms of transfer functions is not new. Simoncini and Gallopoulos [34] used a transfer function framework to rewrite most of the techniques used to approximate ϵ-pseudospectra of large matrices, yielding interesting comparisons as well as better understanding of the techniques. Hinrichsen and Kelb [17] investigated structured pseudospectra of a single matrix with perturbations of the form in (2.10), and they expressed the structured ϵ-pseudospectrum in terms of a transfer function.

Consider the equation

$$P(z)v = (z^m A_m + z^{m-1} A_{m-1} + \cdots + A_0)v = u.$$

It can be rewritten as

$$(F - zG)\begin{bmatrix} v \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -u \end{bmatrix},$$

where F and G are defined in (2.16). Hence

$$P(z)^{-1}u = v = [\, I \quad 0 \quad \dots \quad 0\,](F - zG)^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -u \end{bmatrix} = [\, I \quad 0 \quad \dots \quad 0\,](F - zG)^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I \end{bmatrix} u.$$

Since this equation holds for all u, it follows that

$$P(z)^{-1} = [\, I \quad 0 \quad \cdots \quad 0\,](F - zG)^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I \end{bmatrix}. \tag{3.2}$$

This equality can also be deduced from the theory of λ-matrices [21, Thm. 14.2.1]. We have thus expressed the resolvent in terms of a transfer function.

In control theory, $P(z)^{-1}$ corresponds to the transfer function of the linear time-invariant multivariate system described by

$$G\dot{x}(t) = Fx(t) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix} u(t),$$
$$y(t) = [\, I \quad 0 \quad \cdots \quad 0\,]\, x(t).$$

Several algorithms have been proposed in the literature [22], [27] to compute transfer functions at a large number of frequencies, most of them assuming that $G = I$. Our objective is to efficiently compute the norm of the transfer function, rather than to compute the transfer function itself.

For structured perturbations we see from (2.17) that the transfer function $P(z)^{-1}$ is replaced by

$$E(z)P(z)^{-1}D = [\, E(z) \quad 0 \quad \cdots \quad 0\,](F - zG)^{-1}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -D \end{bmatrix}.$$

All the methods described below for the dense case are directly applicable with obvious changes.

We would like a factorization of $F - zG$ that enables efficient evaluation or application of $(F - zG)^{-1}$ for many different z. There are various possibilities, including, when G is nonsingular,

$$F - zG = G(G^{-1}F - zI) = G(W^*TW - zI) = GW^*(T - zI)W,$$

where $G^{-1}F = W^*TW$ is a Schur decomposition, with W unitary and T upper triangular. However this approach is numerically unstable when G is ill conditioned. A numerically stable reduction is obtained by computing the generalized Schur decomposition

$$W^*FZ = T, \quad W^*GZ = S, \tag{3.3}$$

where W and Z are unitary and T and S are upper triangular. Then

$$P(z)^{-1} = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} Z\,(T - zS)^{-1}\, W^* \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I \end{bmatrix}. \tag{3.4}$$

Hence once the generalized Schur decomposition has been computed, we can compute $P(z)^{-1}x$ and $P(z)^{-*}x$ at a cost of $O((mn)^2)$ flops, since $T - zS$ is triangular of dimension mn. For the 2-norm we can therefore efficiently approximate $\|P(z)^{-1}\|$ using inverse iteration or the inverse Lanczos iteration, that is, the power method or the Lanczos method applied to $P(z)^{-1}P(z)^{-*}$.

The cost of the computation breaks into two parts: the cost of the initial transformations and the cost of the computations at each of the ν^2 grid points. Assuming that (3.3) is computed using the QZ algorithm [9, Sec. 7.7] and the average number of power method or Lanczos iterations per grid point is k, the total cost is about

$$66(mn)^3 + k\nu^2(8mn^2 + 3(mn)^2) \text{ flops}.$$

For the important special case $m = 2$ (the quadratic eigenvalue problem), this cost is

$$528n^3 + 28k\nu^2n^2 \text{ flops}. \tag{3.5}$$

Comparing with (3.1) we see that this method is a significant improvement over the SVD-based approach for a sufficiently fine grid and a small degree m.

For the 2-norm note that, because of the two outer factors in (3.4), we cannot discard the unitary matrices Z and W, unlike in the analogous expression for the resolvent of a single matrix in the standard eigenproblem. For the 1- and ∞-norms we can efficiently estimate $\|P(z)^{-1}\|$ using the algorithm of Higham and Tisseur [14], which requires only the ability to multiply matrices by $P(z)^{-1}$ and $P(z)^{-*}$.

An alternative to the generalized Schur decomposition is the generalized Hessenberg-triangular form, which differs from (3.3) in that one of T and S is upper Hessenberg. The Hessenberg form is cheaper to compute but more expensive to work with. It leads to a smaller overall flop count when $k\nu^2 \gtrsim 25mn$.

3.2. Factorizing the quadratic polynomial. The transfer function-based method of the previous section has the drawback that it factorizes matrices of dimension m times those of the original polynomial matrix. We now describe another method, particular to the quadratic case, that does not increase the size of the problem.

Suppose we can find a matrix S such that $A_2S^2 + A_1S + A_0 = 0$, that is, a solvent of the quadratic matrix equation $A_2X^2 + A_1X + A_0 = 0$. Then

$$Q(z) := z^2A_2 + zA_1 + A_0 = -(A_1 + A_2S + zA_2)(S - zI). \tag{3.6}$$

If we compute the Schur decomposition

$$S = QTQ^*$$

and the generalized Schur decomposition

$$W^*(A_1 + A_2S)Z = R_1, \quad W^*A_2Z = R_2$$

then

$$Q(z)^{-1} = -Q(T - zI)^{-1}Q^*Z\,(R_1 + zR_2)^{-1}\,W^*, \tag{3.7}$$

so a vector can be premultiplied by $Q(z)^{-1}$ or its conjugate transpose in $O(n^2)$ flops for any z. Moreover, for the 2-norm we can drop the outer Q and W^* factors in (3.7), by unitary invariance, and hence we do not need to form W. For the 2-norm, the total cost of this method is

$$c_S + 77n^3 + 10k\nu^2n^2 \text{ flops}, \tag{3.8}$$

where c_S is the cost of computing a solvent and we have assumed that we precompute Q^*Z. Comparing this flop count with (3.5) we see that the cost per grid point of the solvent approach is much lower.

The success of this method depends on two things: the existence of solvents and being able to compute one at a reasonable cost. Some sufficient conditions for the existence of a solvent are summarized in [13]. In particular, for an overdamped problem, one for which A_2 and A_1 are Hermitian positive definite, A_0 is Hermitian positive semidefinite, and $(x^*A_1x)^2 > 4(x^*A_2x)(x^*A_0x)$ for all $x \neq 0$, a solvent is guaranteed to exist.

Various methods are available for computing solvents [12], [13]. One of the most generally useful is Newton's method, optionally with exact line searches, which requires a generalized Sylvester equation in $n \times n$ matrices to be solved on each iteration, at a total cost of about $56n^3$ flops per iteration. If Newton's method converges within 8 iterations or so, so that $c_S \leq 448n^3$ flops, this approach is certainly competitive in cost with the transfer function approach.

When there is a gap between the n largest and n smallest eigenvalues ordered by modulus, as is the case for overdamped problems [20, Sec. 7.6], Bernoulli iteration is an efficient way of computing the dominant or minimal solvent S [13]. If t iterations are needed for convergence to the dominant or minimal solvent then the cost of Bernoulli iteration is about $c_S = 4tn^3$ flops. Bernoulli iteration converges only linearly, but convergence is fast if the eigenvalue gap is large.

A third approach to computing a solvent is to use a Schur method from [13], based on the following theorem. Let F and G be defined as in (2.16), so that

$$F = \begin{bmatrix} 0 & I \\ -A_0 & -A_1 \end{bmatrix}, \qquad G = \begin{bmatrix} I & 0 \\ 0 & A_2 \end{bmatrix}.$$

THEOREM 3.1 (Higham and Kim [13]). *All solvents of $Q(X)$ are of the form $X = Z_{21}Z_{11}^{-1} = Q_{11}T_{11}S_{11}^{-1}Q_{11}^{-1}$, where*

$$Q^*FZ = T, \quad Q^*GZ = S \tag{3.9}$$

is a generalized Schur decomposition with Q and Z unitary and T and S upper triangular, and where all matrices are partitioned as block 2×2 matrices with $n \times n$ blocks.

The method consists of computing the generalized Schur decomposition (3.9) by the QZ algorithm and then forming $S = Z_{21}Z_{11}^{-1}$. The generalized Schur decomposition may need to be reordered in order to obtain a nonsingular Z_{11}. Note that the

unitary factor Q does not need to be formed. For this method, $c_S = 50(2n)^3 + rn^2$, where the constant r depends on the amount of reordering required. From (3.8), the total cost is now

$$479n^3 + rn^2 + 10k\nu^2n^2 \text{ flops},$$

which is much more favorable than the cost (3.5) of the transfer function method.

For higher degree polynomials we can generalize this approach by attempting to factorize P into linear factors by recursively computing solvents. However, for degrees m greater than 2 classes of problem for which a factorization into linear factors exists are less easily identified and the cost of Newton's method (for example) is much higher than for $m = 2$ [19].

3.3. Large-scale computation. All the methods described above are intended for small- to medium-scale problems for which Schur and other reductions are possible. For large, possibly sparse, problems, different techniques are necessary. These techniques can be classified into two categories: those that project to reduce the size of the problem and then compute the pseudospectra of the reduced problem, and those that approximate the norm of the resolvent directly.

3.3.1. Projection approach. For a single matrix, A, Toh and Trefethen [39] and Wright and Trefethen [48] approximate the resolvent norm by the Arnoldi method; that is, they approximate $\|(A - zI)^{-1}\|_2$ by $\|(H_m - zI)^{-1}\|_2$ or by $\sigma_{\min}(\widetilde{H}_m - z\widetilde{I})$, where H_m is the square Hessenberg matrix of dimension $m \ll n$ obtained from the Arnoldi process and $\widetilde{H}_m$ is the matrix H_m augmented by an extra row. Simoncini and Gallopoulos [34] show that a better but more costly approximation is obtained by approximating $\|(A-zI)^{-1}\|_2$ with $\|V_m^*(A-zI)^{-1}V_{m+1}\|_2$, where V_m is the orthonormal basis generated during the Arnoldi process. These techniques are not applicable to the polynomial eigenvalue problem of degree larger than one because of the lack of a Schur form for the Arnoldi method to approximate.

A way of approximating $\|P(z)^{-1}\|$ for all z is through a projection of $P(z)^{-1}$ onto a lower dimensional subspace. Let V_k be an $n \times k$ matrix with orthonormal columns. We can apply one of the techniques described in the previous sections to compute pseudospectra of the projected polynomial eigenvalue problem $\widetilde{P}(\lambda) = V_k^* P(\lambda) V_k$. A possible choice for V_k is an orthonormal basis of k selected linearly independent eigenvectors of $P(\lambda)$. In this case, $\widetilde{P}(\lambda)$ is the matrix representation of the projection of $P(\lambda)$ onto the subspace spanned by the selected eigenvectors. The eigenvectors can be chosen to correspond to parts of the spectrum of interest and can be computed using the Arnoldi process on the companion form pencil (F, G) or directly on $P(\lambda)$ with the Jacobi–Davidson method or its variants [26], [35]. In the latter case, the matrix V_k is built during the Davidson process.

3.3.2. Direct approach. This approach consists of approximating $\|P(z)^{-1}\|$ at each grid point z. Techniques analogous to those used for single matrices can be applied, such as the Lanczos method applied to $P(z)^*P(z)$ or its inverse. We refer the reader to [42] for more details and further references.

4. Applications and numerical experiments. We give a selection of applications of pseudospectra for polynomial eigenvalue problems, using them to illustrate the performance of our methods for computing pseudospectra. All our examples are for 2-norm pseudospectra.

4.1. The wing problem. The first example is based on a quadratic polynomial $Q(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ from [6, Sec. 10.11], with numerical values modified as in [20, Sec. 5.3]. The eigenproblem for $Q(\lambda)$ arose from the analysis of the oscillations of a wing in an airstream. The matrices are

$$A_2 = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.6 & 0.756 & 0.658 \end{bmatrix},$$

$$A_0 = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix}.$$

The left plot in Figure 4.1 shows the boundaries of ϵ-pseudospectra with perturbations measured in the absolute sense ($\alpha_i \equiv 1$), with ϵ between 10^{-3} and $10^{-0.8}$. The eigenvalues are plotted as dots. Another way of approximating a pseudospectrum is by random perturbations of the original matrices [41]. We generated 200 triples of complex random normal perturbation matrices $(\Delta A_1, \Delta A_2, \Delta A_3)$ with $\|\Delta A_j\|_2 = 10^{-0.8}$, $j = 1\colon 3$. In the right plot of Figure 4.1 are superimposed as small dots the eigenvalues of the perturbed polynomials $\lambda^2(A_2 + \Delta A_2) + \lambda(A_1 + \Delta A_1) + \Delta A_0 + \Delta A_0$. The solid curve marks the boundary of the ϵ-pseudospectrum for $\epsilon = 10^{-0.8}$. Both pictures show that the pair of complex eigenvalues $\lambda = -0.88 \pm 8.4i$ are much more sensitive to perturbations than the other two complex pairs.

The eigenvalues of $Q(\lambda)$ are the same as those of the linearized problem $\mathcal{A} - \lambda I$, where

$$\mathcal{A} = \begin{bmatrix} 0 & I \\ -A_2^{-1}A_0 & -A_2^{-1}A_1 \end{bmatrix}. \tag{4.1}$$

Figure 4.2 shows boundaries of ϵ-pseudospectra for this matrix, for the same ϵ as in Figure 4.1. Clearly, the ϵ-pseudospectra of the linearized problem (4.1) do not give useful information about the behavior of the eigensystem of $Q(\lambda)$ under perturbations. This emphasizes the importance of defining and computing pseudospectra for the quadratic eigenvalue problem in its original form.

4.2. Mass-spring system. We now consider the connected damped mass-spring system illustrated in Figure 4.3. The ith mass of weight m_i is connected to the $(i+1)$st mass by a spring and a damper with constants k_i and d_i, respectively. The ith mass is also connected to the ground by a spring and a damper with constants κ_i and τ_i, respectively. The vibration of this system is governed by a second-order differential equation

$$M\frac{d^2}{dt^2}x + C\frac{d}{dt}x + Kx = 0,$$

where the mass matrix $M = \operatorname{diag}(m_1, \ldots, m_n)$ is diagonal, and the damping matrix C and stiffness matrix K are symmetric tridiagonal. The differential equation leads to the quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)x = 0. \tag{4.2}$$

In our experiments, we took all the springs (respectively, dampers) to have the same constant $\kappa = 5$ (respectively, $\tau = 10$), except the first and last, for which the constant is 2κ (respectively, 2τ), and we took $m_i \equiv 1$. Then

$$C = \tau\,\operatorname{tridiag}(-1, 3, -1), \quad K = \kappa\,\operatorname{tridiag}(-1, 3, -1),$$

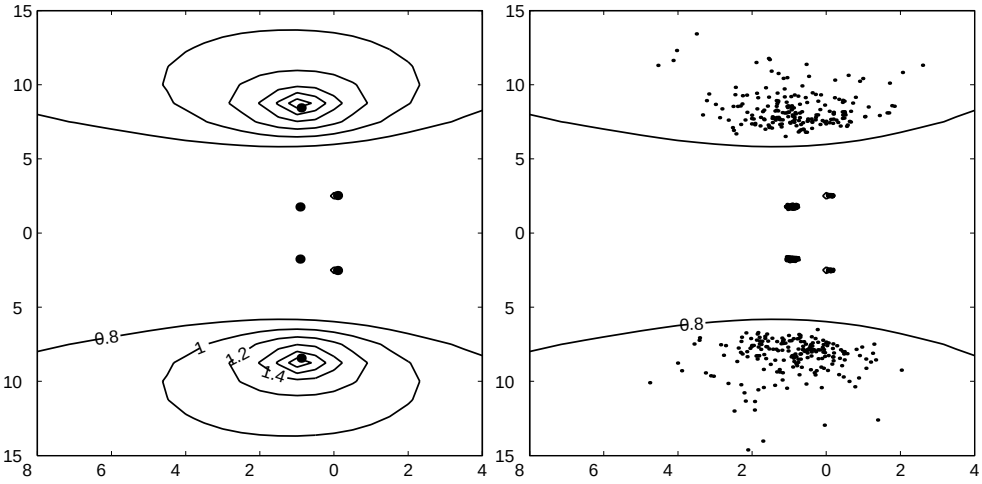


FIG. 4.1. *Wing problem. Left:* $\Lambda_\epsilon(Q)$*, for* $\epsilon \in [10^{-3}, 10^{-0.8}]$*. Right: approximation to* ϵ*-pseudospectrum with* $\epsilon = 10^{-0.8}$*.*

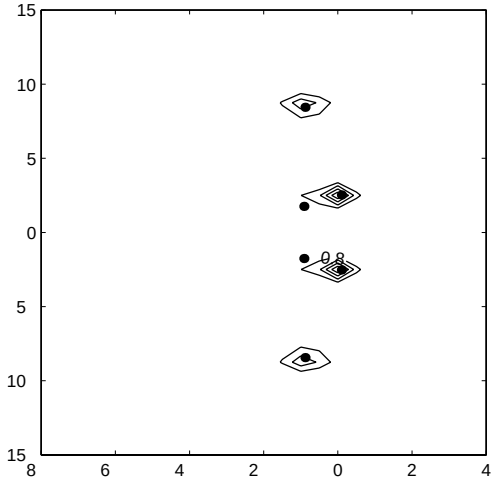


FIG. 4.2. *Wing problem.* $\Lambda_\epsilon(\mathcal{A})$*, for* $\mathcal{A}$ *in* (4.1) *with* $\epsilon \in [10^{-3}, 10^{-0.8}]$*.*

and the quadratic eigenvalue problem is overdamped. We take an $n = 250$ degree of freedom mass-spring system over a 100×100 grid. A plot of the pseudospectra is given in Figure 4.4.

For this problem we compare all the methods described. In the solvent approach exact line searches were used in Newton's method and no reordering was used in the generalized Schur method. The solvents from the Bernoulli and Schur methods were refined by one step of Newton's method. The Bernoulli iteration converged in 12 iterations while only 6 iterations were necessary for Newton's method. The Lanczos inverse iteration converged after 3 iterations on average. In Table 4.1 we give the estimated flop counts, using the formulae from section 3, together with execution times. The computations were performed in MATLAB 6, which is an excellent environment for investigating pseudospectra. While the precise times are not important, the con-

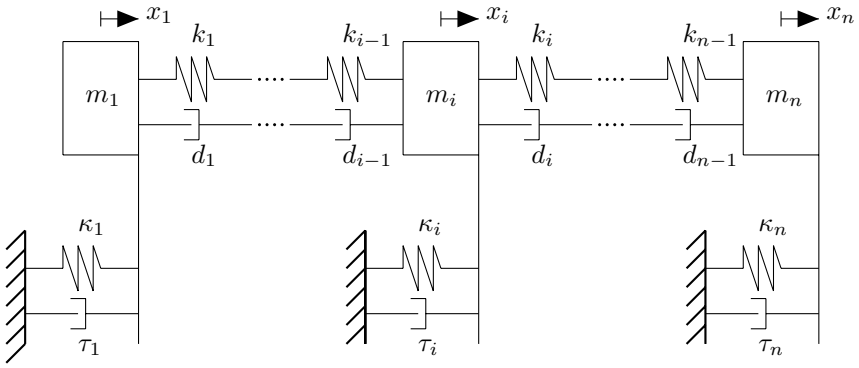


FIG. 4.3. *An n degree of freedom damped mass-spring system.*

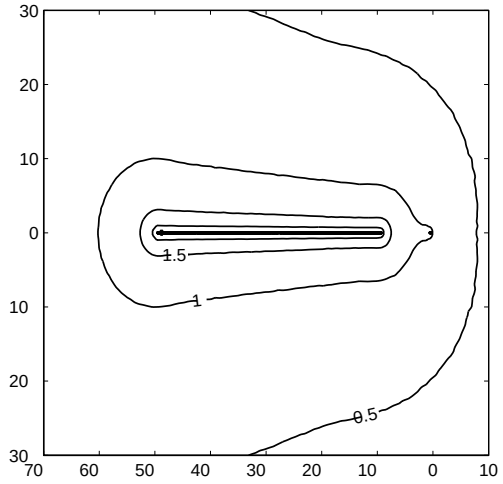


FIG. 4.4. *Pseudospectra of a* 250 *degree of freedom damped mass-spring system on a* 100×100 *grid.*

clusion is clear: in this example, the three solvent-based methods are much faster than the SVD and transfer function methods. (The high speed of the SVD method relative to its flop count is attributable to MATLAB's very efficient `svd` function.)

4.3. Acoustic problem. Acoustic problems with damping can give rise to large quadratic eigenvalue problems (4.2), where, again, M is the mass matrix, C is the damping matrix, and K the stiffness matrix. We give in Figure 4.5 the sparsity pattern of the three matrices M, C, and K of order 107 arising from a model of a speaker box [1]. These matrices are symmetric and the sparsity patterns of M and K are identical. There is a large variation in the norms: $\|M\|_2 = 1$, $\|C\|_2 = 0.06$, $\|K\|_2 = 9.9 \times 10^6$.

We plot in Figure 4.6 pseudospectra with perturbations measured in both an absolute sense ($\alpha_1 = \alpha_2 = \alpha_3 = 1$) and a relative sense ($\alpha_1 = \|M\|_2$, $\alpha_2 = \|C\|_2$, $\alpha_3 = \|K\|_2$), together with pseudospectra of the corresponding standard eigenvalue problem of the form (4.1). The eigenvalues are all pure imaginary and are marked by dots on the plot. The two first plots are similar, both showing that the most sensitive

TABLE 4.1
Comparison in terms of flops and execution time of different techniques.

Method	Estimated cost in flops	Execution time
Golub–Reinsch SVD	$26747n^3$	102 min
Transfer function	$3408n^3$	106 min
Solvent: Newton	$1589n^3$	39 min
Solvent: Bernoulli	$1325n^3$	36 min
Solvent: Schur	$1677n^3$	37 min

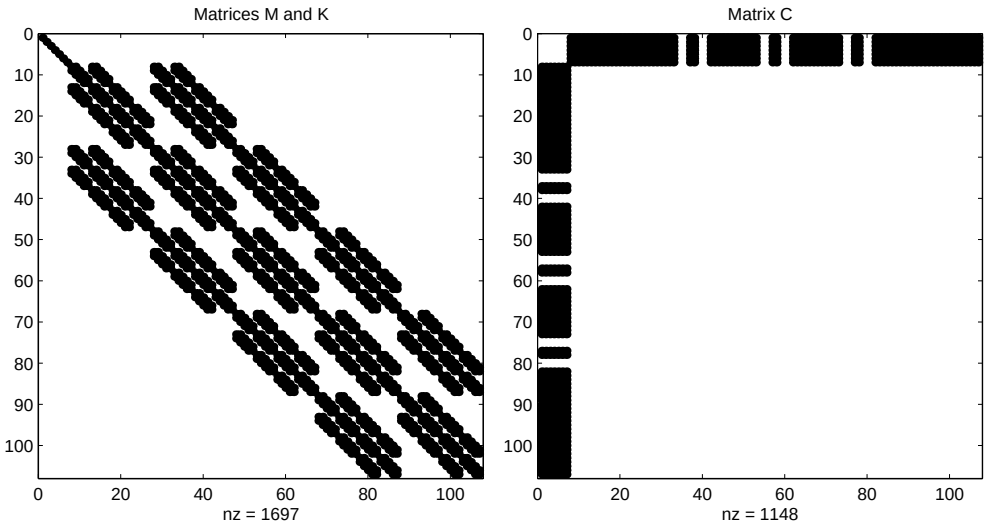


FIG. 4.5. *Sparsity patterns of the three* 107×107 *matrices* M, C, *and* K *of an acoustic problem.*

eigenvalues are located at the extremities of the spectrum; the contour lines differ mainly around the zero eigenvalue. The last plot is very different; clearly it is the eigenvalues close to zero that are the most sensitive to perturbations of the standard eigenproblem form.

We mention that for this problem we have been unable to compute a solvent.

4.4. Closed loop system. In multi-input and multioutput systems in control theory the location of the eigenvalues of matrix polynomials determine the stability of the system. Figure 4.7 shows a closed-loop system with feedback with gains 1 and $1+\alpha$, $\alpha > 0$. The associated matrix polynomial is given by

$$P(z) = z^2 I + z \begin{bmatrix} 0 & 1+\alpha \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1/2 & 0 \\ 0 & 1/4 \end{bmatrix}.$$

We are interested in the values of α for which $P(z)$ has all its eigenvalues inside the unit circle. By direct calculation with $\det(P(z))$, using the Routh array, for example, it can be shown that $P(z)$ has all its eigenvalues inside the unit circle if and only if $\alpha < 0.875$.

The matrix $P(z)$ can be viewed as a perturbed matrix polynomial with structured perturbations:

$$P(z) = P(z) \mid_{\alpha=0} + D\Theta E(z),$$

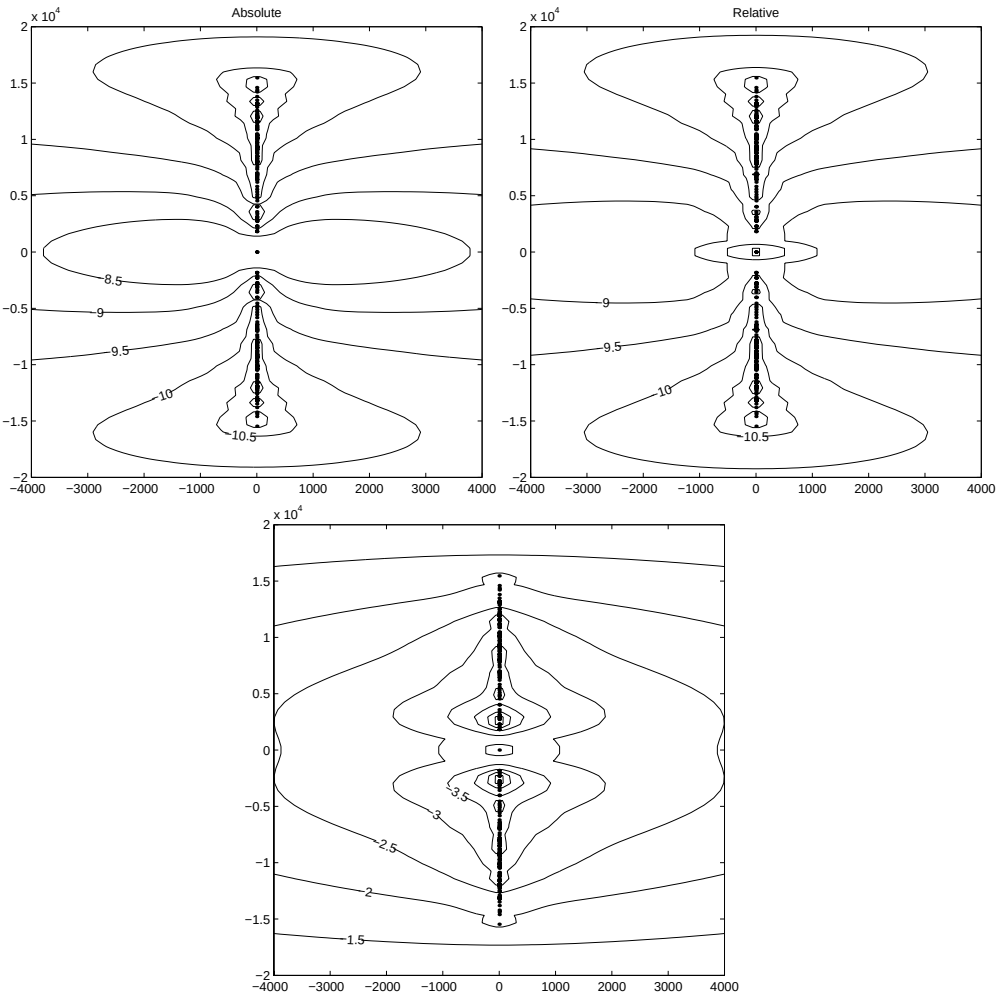


FIG. 4.6. *Acoustic problem, $n = 107$, 70×70 grid. Perturbations measured in an absolute sense (top left) and relative sense (top right). Pseudospectra of the equivalent standard eigenvalue problem are shown at the bottom.*

where

$$D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \Theta = \alpha, \qquad E(z) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} I \\ zI \\ z^2 I \end{bmatrix}.$$

We show in Figure 4.8 the structured pseudospectra as defined by (2.17). The dashed lines mark the unit circle. Since the outermost contour has value $\alpha = 0.875$ and just touches the unit circle, this picture confirms the value for the maximal α that we obtained analytically.

4.5. The Orr–Sommerfeld equation. The Orr–Sommerfeld equation is a linearization of the incompressible Navier–Stokes equations in which the perturbations in velocity and pressure are assumed to take the form

$$\Phi(x, y, t) = \phi(y) e^{i(\lambda x - \omega t)},$$

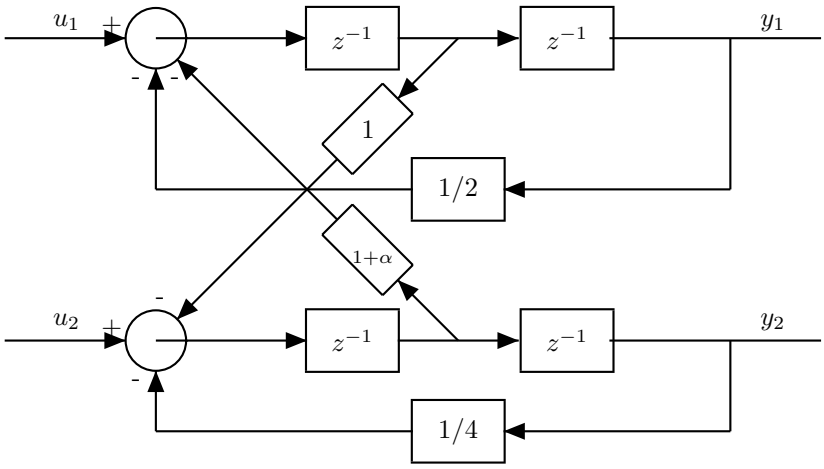


FIG. 4.7. *Closed-loop system with feedback gains* 1 *and* $1+\alpha$.

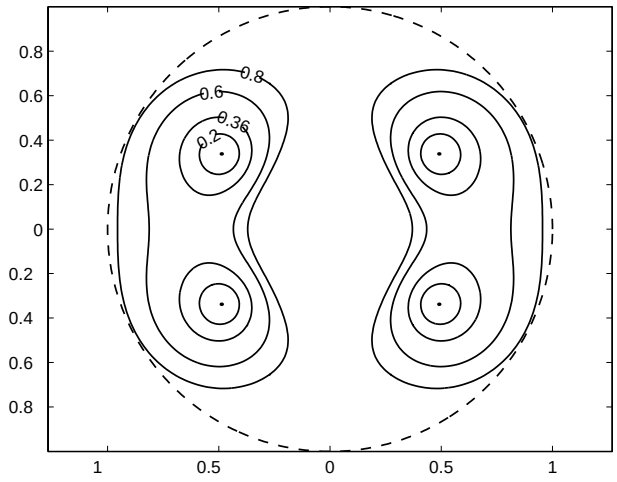


FIG. 4.8. *Structured pseudospectra of a closed-loop system with one-parameter feedback.*

where λ is a wavenumber and ω is a radian frequency. For a given Reynolds number R, the Orr–Sommerfeld equation may be written

$$\left[\left(\frac{d^2}{dy^2}-\lambda^2\right)^2 - iR\left\{(\lambda U-\omega)\left(\frac{d^2}{dy^2}-\lambda^2\right)-\lambda U''\right\}\right]\phi = 0. \tag{4.3}$$

We consider plane Poiseuille flow between walls at $y=\pm 1$ and with velocity $U(y) = 1-y^2$ in the streamwise x direction, for which the boundary conditions are

$$\phi(\pm 1) = 0, \quad \phi'(\pm 1) = 0.$$

For a given real value of R, the boundary conditions will be satisfied only for certain combinations of values of λ and ω. Two cases are of interest.

Case 1. *Temporal stability.* If λ is fixed and real, then (4.3) is linear in the parameter ω and corresponds to a generalized eigenvalue problem. The perturbations

are periodic in x and grow or decay in time depending on the sign of the imaginary part of ω. This case has been studied with the help of pseudospectra by Reddy, Schmid, and Henningson [32].

Case 2. *Spatial stability.* For most real flows, the perturbations are periodic in time, which means that ω is real. Then the sign of the imaginary part of λ determines whether the perturbations will grow or decay in space. In this case, the parameter is λ, which appears to the fourth power in (4.3), so we obtain a quartic polynomial eigenvalue problem. Bridges and Morris [2] calculated the spectrum of (4.3) using a finite Chebyshev series expansion of ϕ combined with the Lanczos tau method and they computed the spectrum of the quartic polynomial by two methods: the QR algorithm applied to the corresponding standard eigenvalue problem in companion form, and Bernoulli iteration applied to determine a minimal solvent and hence to obtain the n eigenvalues of minimal modulus.

For our estimation of the pseudospectra of the Orr–Sommerfeld equation we use a Chebyshev spectral discretization that combines an expansion in Chebyshev polynomials and collocation at the Chebyshev points with explicit enforcement of the boundary conditions. We are interested in the eigenvalues λ that are the closest to the real axis, and we need $\mathrm{Im}(\lambda) > 0$ for stability. The linear eigenvalue problem (Case 1) has been solved by Orszag [29]. The critical neutral point corresponding to λ and ω both real for minimum R was found at $R = 5772$ and $\lambda = 1.02056$ with the frequency $\omega = 0.26943$ [2], [29]. For our calculations we set R and ω to these values and we computed the modes λ, taking $N = 64$, which gives matrices of order $N - 1$. The first few modes are plotted in Figure 4.9. For the first mode we obtained $\lambda = 1.02056 + 9.7 \times 10^{-7} i$, which compares favorably with the result of Orszag. Figure 4.10 shows the pseudospectra in a region around the first few modes on a 100×100 grid, with $\alpha_i = \|A_i\|_2$ except that $\alpha_4 = 0$, since A_4 is the identity matrix and is not subject to uncertainty. The plot shows that the first mode is very sensitive. Interestingly, the second and subsequent modes are almost as sensitive, with perturbations of order 10^{-9} in the matrix coefficients being sufficient to move all these modes across the real axis, making the flow unstable. The pseudospectra thus give a guide to the accuracy with which computations must be carried out for the numerical approximations to the modes to correctly determine the location of the modes. For more on the interpretation of pseudospectra for this problem, see [32] and [44].

Again, for comparison we computed the pseudospectra of the corresponding standard eigenvalue problem. The picture was qualitatively similar, but the contour levels were several orders of magnitude smaller, thus not revealing the true sensitivity of the problem.

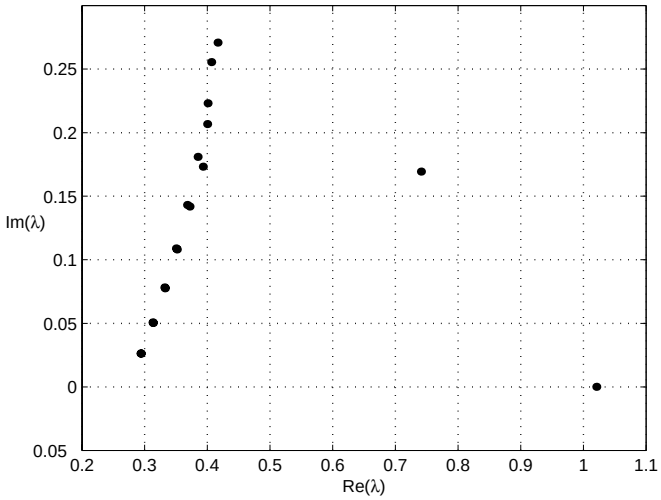


FIG. 4.9. *The first few modes of the spectrum of the Orr–Sommerfeld equation for* $R = 5572$ *and* $\omega = 0.26943$.

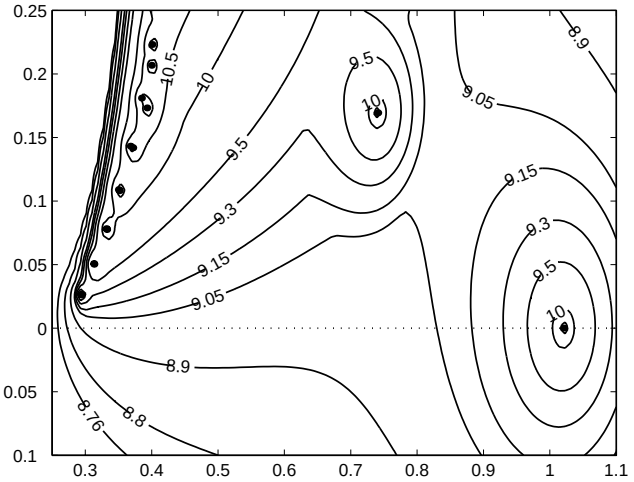


FIG. 4.10. *Pseudospectra of the Orr–Sommerfeld equation for* $R = 5572$ *and* $\omega = 0.26943$.

REFERENCES

[1] Z. Bai, *private communication*, 1999.

[2] T. J. Bridges and P. J. Morris, *Differential eigenvalue problems in which the parameter appears nonlinearly*, J. Comput. Phys., 55 (1984), pp. 437–460.

[3] M. Embree and L. N. Trefethen, *Pseudospectra gateway*, http://www.comlab.ox.ac.uk/pseudospectra/.

[4] V. Frayssé, M. Gueury, F. Nicoud, and V. Toumazou, *Spectral Portraits for Matrix Pencils*, Technical Report TR/PA/96/19, CERFACS, Toulouse, France, 1996.

[5] V. Frayssé and V. Toumazou, *A note on the normwise perturbation theory for the regular generalized eigenproblem*, Numer. Linear Algebra Appl., 5 (1998), pp. 1–10.

[6] R. A. Frazer, W. J. Duncan, and A. R. Collar, *Elementary Matrices and Some Applications to Dynamics and Differential Equations*, 10th ed., Cambridge University Press, 1938, 1963 printing.

[7] Y. Genin and P. M. Van Dooren, *Stability Radii of Polynomial Matrices*, manuscript, 1999.

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

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 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] N. J. Higham and H.-M. Kim, *Solving a Quadratic Matrix Equation by Newton's Method with Exact Line Searches*, Numerical Analysis Report 339, Manchester Centre for Computational Mathematics, Manchester, 1999; SIAM J. Matrix Anal. Appl., to appear.

[13] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.

[14] N. J. Higham and F. Tisseur, *A block algorithm for matrix* 1*-norm estimation, with an application to* 1*-norm pseudospectra*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1185–1201.

[15] N. J. Higham and F. Tisseur, *Bounds for Eigenvalues of Matrix Polynomials*, Numerical Analysis Report 371, Manchester Centre for Computational Mathematics, Manchester, 2001; Linear Algebra Appl., to appear.

[16] N. J. Higham and F. Tisseur, *More on Pseudospectra for Polynomial Eigenvalue Problems and Applications in Control Theory*, Numerical Analysis Report 372, Manchester Centre for Computational Mathematics, Manchester, 2001.

[17] D. Hinrichsen and B. Kelb, *Spectral value sets: A graphical tool for robustness analysis*, Systems Control Lett., 21 (1993), pp. 127–136.

[18] D. Hinrichsen and A. J. Pritchard, *Real and complex stability radii: A survey*, in Control of Uncertain Systems, D. Hinrichsen and B. Mårtensson, eds., Progr. Systems Control Theory 6, Birkhäuser, Boston, 1990, pp. 119–162.

[19] W. Kratz and E. Stickel, *Numerical solution of matrix polynomial equations by Newton's method*, IMA J. Numer. Anal., 7 (1987), pp. 355–369.

[20] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966.

[21] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, London, 1985.

[22] A. J. Laub, *Efficient multivariable frequency response computations*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 407–408.

[23] P.-F. Lavallée, *Nouvelles Approches de Calcul du ϵ-Spectre de Matrices et de Faisceaux de Matrices*, Ph.D. thesis, L'Université de Rennes 1, Rennes, France, 1997.

[24] C.-K. Li and L. Rodman, *Numerical range of matrix polynomials*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1256–1265.

[25] S. H. Lui, *Computation of pseudospectra by continuation*, SIAM J. Sci. Comput., 18 (1997), pp. 565–573.

[26] K. Meerbergen, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput., 22 (2001), pp. 1814–1839.

[27] P. Misra and R. V. Patel, *A determinant identity and its application in evaluating frequency response matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 248–255.

[28] R. G. Mosier, *Root neighborhoods of a polynomial*, Math. Comp., 47 (1986), pp. 265–273.

[29] S. A. Orszag, *Accurate solution of the Orr–Sommerfeld stability equation*, J. Fluid Mech., 50 (1971), pp. 689–703.

[30] G. Pappas and D. Hinrichsen, *Robust stability of linear systems described by higher order*

dynamic equations, IEEE Trans. Automat. Control, 38 (1993), pp. 1430–1435.
[31] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*, Wiley, New York, 1971.
[32] S. C. Reddy, P. J. Schmid, and D. S. Henningson, *Pseudospectra of the Orr–Sommerfeld operator*, SIAM J. Appl. Math., 53 (1993), pp. 15–47.
[33] K. S. Riedel, *Generalized epsilon-pseudospectra*, SIAM J. Numer. Anal., 31 (1994), pp. 1219–1225.
[34] V. Simoncini and E. Gallopoulos, *Transfer functions and resolvent norm approximation of large matrices*, Electron. Trans. Numer. Anal., 7 (1998), pp. 190–201.
[35] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.
[36] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[37] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, Numerical Analysis Report 370, Manchester Centre for Computational Mathematics, Manchester, 2000; SIAM Rev., 43 (2001), pp. 235–286.
[38] K.-C. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math., 68 (1994), pp. 403–425.
[39] K.-C. Toh and L. N. Trefethen, *Calculation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput., 17 (1996), pp. 1–15.
[40] V. Toumazou, *Portraits Spectraux de Matrices: Un Outil d'Analyse de la Stabilité*, Ph.D. thesis, Université Henri Poincaré, Nancy-I, France, 1996.
[41] L. N. Trefethen, *Pseudospectra of matrices*, in Numerical Analysis 1991, Proceedings of the 14th Dundee Conference, D. F. Griffiths and G. A. Watson, eds., Pitman Res. Notes Math. Ser. 260, Longman Scientific and Technical, Harlow, UK, 1992, pp. 234–266.
[42] L. N. Trefethen, *Computation of pseudospectra*, Acta Numer., 8 (1999), pp. 247–295.
[43] L. N. Trefethen, *Spectra and pseudospectra*, in The Graduate Student's Guide to Numerical Analysis '98, M. Ainsworth, J. Levesley, and M. Marletta, eds., Springer-Verlag, Berlin, 1999, pp. 217–250.
[44] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll, *Hydrodynamic stability without eigenvalues*, Science, 261 (1993), pp. 578–584.
[45] P. M. Van Dooren and V. Vermaut, *On stability radii of generalized eigenvalue problems*, in European Control Conference, paper FR-M-H6, 1997.
[46] J. L. M. Van Dorsselaer, *Pseudospectra for matrix pencils and stability of equilibria*, BIT, 37 (1997), pp. 833–845.
[47] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.
[48] T. G. Wright and L. N. Trefethen, *Large-scale computation of pseudospectra using ARPACK and* `eigs`, Report 00/11, Numerical Analysis Group, Oxford University Computing Laboratory, Oxford, 2000; SIAM J. Sci. Comput., to appear.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 209–224
© 2001 Society for Industrial and Applied Mathematics

RECURSIVE ALGORITHM FOR THE FUNDAMENTAL/GROUP INVERSE MATRIX OF A MARKOV CHAIN FROM AN EXPLICIT FORMULA*

ISAAC SONIN† AND JOHN THORNTON†

Abstract. We present a new accurate algorithm (REFUND) for computing the fundamental matrix (or closely related group inverse matrix) of a finite regular Markov chain. This algorithm is developed within the framework of the state reduction approach exemplified by the GTH (Grassmann, Taksar, Heyman)/S (Sheskin) algorithm for recursively finding invariant measure. The first (reduction) stage of the GTH/S algorithm is shared by REFUND, as well as by an earlier algorithm FUND developed for the fundamental matrix by Heyman in 1995, and by a modified version of Heyman and O'Leary in 1998. Unlike FUND, REFUND is recursive, being based on an explicit formula relating the group inverse matrix of an initial Markov chain and the group inverse matrix of a Markov chain with one state removed. Operation counts are approximately the same: $\Theta(\frac{7}{3}n^3)$ for REFUND versus $\Theta(\frac{8}{3}n^3)$ for FUND. Numerical tests indicate that REFUND is accurate. The structure of REFUND makes it easily combined with the other algorithms based on the state reduction approach. We also discuss the general properties of this approach, as well as connections to the optimal stopping problem and to tree decompositions of graphs related to Markov chains.

Key words. Markov chain, fundamental/group inverse matrix, recursive algorithm, GTH/S algorithm

AMS subject classifications. Primary, 60J10; Secondary, 65U05

PII. S0895479899351234

1. Introduction.

Let $P = [p(i,j)]$, $i,j = 1,2,\dots,n$, be a stochastic (transition) matrix. The calculation of various characteristics of a Markov chain specified by P is an important part of applied probability theory and computational algebra. These characteristics include the distribution of a Markov chain at the moment of the first visit to a subset of its state space, the mean time spent at given states until such visit, the invariant distribution, the fundamental matrices for both transient and regular Markov chains, the covariance matrix, and many others. Mainly we will discuss two of the most important ones, the invariant distribution and the fundamental matrix for a regular Markov chain.

The *invariant* (steady state) *distribution* $\boldsymbol{\pi}$ is the solution of the system of linear equations

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T P, \tag{1.1}$$

where ^{T} denotes transposition and all vectors are assumed to be column vectors. In the regular (ergodic) case, i.e., when there is a k for which all elements of P^k are strictly positive, π is the limiting distribution for any initial point. The matrix

$$A = \lim_n P^n = \boldsymbol{e}\boldsymbol{\pi}^T \tag{1.2}$$

has all rows equal to the vector $\boldsymbol{\pi}^T$; and $\mathbf{e}$ is a vector all of whose entries are ones.

*Received by the editors February 5, 1999; accepted for publication (in revised form) by C. Meyer November 30, 2000; published electronically July 2, 2001.
http://www.siam.org/journals/simax/23-1/35123.html
†Department of Mathematics, University of North Carolina at Charlotte, Charlotte, NC 28223 (imsonin@email.uncc.edu, thornton@uncc.edu).

The *fundamental matrix* Z, for the regular case, is given (see [10]) by

$$Z = (I - (P - A))^{-1} = I + \sum_{n=1}^{\infty}(P - A)^n = I + \sum_{n=1}^{\infty}(P^n - A). \tag{1.3}$$

Instead of calculating Z, we will calculate the *group (generalized) inverse matrix* V,

$$V = Z - A = \sum_{n=0}^{\infty}(P^n - A). \tag{1.4}$$

This matrix has a simple relationship to Z, but it has its own important role. Its significance was explained in a pioneering paper [15], which also discusses the relationship between the group inverse and other generalized matrix inverses. For applications of the group inverse to Markov decision processes, see [14] (and references therein) and the comprehensive monograph [18] (especially Appendix A.5). The elements $v(x, y)$ of V have the following probabilistic interpretation (see [10]):

$$v(x, y) = \lim_n E_x[\eta^n(y) - n\pi(y)], \tag{1.5}$$

where $\eta^n(y)$ is the number of visits to y during the first n moments; and E_x denotes mathematical expectation, given that the Markov chain starts from the initial point x. Thus the $v(x, y)$ measure the expected deviation in the number of visits to state y due to starting in state x instead of starting randomly according to the invariant distribution $\boldsymbol{\pi}$.

The classical formulas (closed form solutions) for V and $\boldsymbol{\pi}$, as well as many other related probabilistic quantities, are well known (see, e.g., [10]) and involve matrix inversion or the solution of a system of linear equations.

There is a vast literature on methods for computing various characteristics of Markov chains. We refer the reader to [16], [31], and the proceedings in which [29] appears, which give a thorough description of the current situation in this field and describe both traditional and some more recent methods to calculate characteristics of Markov chains.

The development of a new class of algorithms was initiated in 1985 by two pioneering papers in which Sheskin [19], and Grassmann, Taksar, and Heyman [2] independently proposed practically the same algorithm to calculate invariant distribution. Later it became known as the GTH algorithm. Taking into account the short but very precise paper of Sheskin [19], we refer to it as the GTH/S algorithm.

The algorithm constructs a sequence of stochastic matrices, each having dimension one less than the previous, and has a simple and transparent probabilistic interpretation (see section 2). Numerous papers (see more references in section 2 and in volumes [16] and [31]) have studied the computational properties of this algorithm, different generalizations, and particular cases. It has been shown, among other things, that the GTH/S algorithm has significant advantages over traditional methods to calculate $\boldsymbol{\pi}$.

In 1995 on the basis of this algorithm, Heyman proposed an algorithm FUND [5] for the sequential computation of the fundamental matrix of a regular Markov chain. In 1998 it was improved and modified by Heyman and O'Leary in [8]. This algorithm uses the idea, outlined by Grassmann in [3], of a triangular factorization of the matrix $(I - P)$ that is produced by the first stage of the GTH/S algorithm.

The main goal of this paper is to present a new algorithm, REFUND, to calculate the fundamental/group inverse matrix. This algorithm, like Heyman's FUND, begins

with the sequence of stochastic matrices constructed by the GTH/S algorithm. The primary distinction is that we forego the triangular factorization, basing this algorithm instead on an explicit formula that relates the group inverse matrices of two stochastic matrices that are adjacent in the sequence of matrices produced by the GTH/S algorithm. A very similar formula can be written for the fundamental matrix. The repeated application of this formula recursively produces the sequence of associated group inverse matrices.

This provides us with the opportunity to begin calculation with any submodel for which the fundamental matrix or group inverse matrix is known, and to bring probabilistic (in addition to numerical) techniques to bear in analyzing where accuracy is lost; and it aids us in taking corrective steps. Like FUND, REFUND requires $O(n^3)$ arithmetic operations to complete, where P is $n \times n$, although REFUND, with a leading constant of $\frac{7}{3}$ versus $\frac{8}{3}$ for FUND, is slightly faster. Like FUND, REFUND can also be applied to calculate the fundamental matrix for a continuous time Markov chain.

The GTH/S algorithm, Heyman's FUND algorithm, REFUND, Sheskin's algorithm [22] to compute the fundamental matrix of a transient Markov chain, the algorithm of optimal stopping of Markov chains proposed in [25] and some others can be viewed as examples of the application of a more general approach, which can be called the state reduction (SR) approach. The elements of the SR approach can be found in the works of many authors, so we do not claim authorship. But we have found no attempts, other than [26], to analyze these algorithms together in a general framework. Since the approach itself has become important enough, the brief presentation of an overview is another goal of our paper. We begin this in section 2. Although the reading of that section is not strictly necessary to a purely formal understanding of the REFUND algorithm, it does furnish a general framework in which all SR algorithms can be compared. (Part of this description and related results were presented in [26].) Section 2 also contains brief descriptions of the GTH/S and FUND algorithms. In section 3 we present Theorem 3.1, which provides an auxiliary characterization of the group inverse matrix, and our main result, Theorem 3.2, which provides the exact formula(s) on which our algorithm is based. Section 4 specifies the REFUND algorithm, gives operation counts, summarizes the results of numerical testing, interprets these results, and compares REFUND to FUND. A detailed study of the computational properties of REFUND and a comparison to FUND was presented in [29].

2. The GTH/S and FUND algorithms, the SR approach, and related problems. In our subsequent presentation an important role is played by the transformations of state spaces and transition matrices. So in what follows, instead of the term "Markov chain," we prefer to use the term "Markov model." A *Markov model* M is a pair (X, P), where X is a finite or countable state space, and P is a stochastic matrix, indexed by elements of X.

In his original paper, Sheskin [19] states that the GTH/S algorithm is motivated by a result from Kemeny and Snell [10], while Grassmann [3] and Heyman [5] describe GTH/S as a variant of Gaussian elimination. Though it is difficult to object to either statement, at the same time (in our opinion) it can be said that the SR approach is based on the following simple probabilistic idea that appeared in the pioneering works of Kolmogorov and Döeblin more than sixty years ago. This idea, described in Proposition 2.1 below, has been used since that time in probability theory in several contexts on numerous occasions.

Let us assume that a finite Markov model $M_1 = (X_1, P_1)$ is given and let (Z_n), $n = 1, 2, \dots$, be a Markov chain specified by the model M_1. Let $X_2 \subset X_1$ and let $\tau_1, \tau_2, \dots, \tau_n, \dots$, be the sequence of Markov times of first, second, and so on visits of (Z_n) to the set X_2, so that $\tau_1 = \min\{k > 0 : Z_k \in X_2\}$, $\tau_{n+1} = \min\{k : \tau_n < k, Z_k \in X_2\}$, $0 < \tau_1 < \tau_2 < \dots$. Let $u_1^{X_2}(x, \cdot)$ be the distribution of Markov chain (Z_n) for the initial model M_1 at the moment τ_1 of first visit to set X_2 (first exit from $X_1 \setminus X_2$) starting at x, $x \in X_1 \setminus X_2$. Let us consider the random sequence $Y_n = Z_{\tau_n}, n = 1, 2, \dots$.

PROPOSITION 2.1. (a) *The random sequence* (Y_n) *is a Markov chain in a model* $M_2 = (X_2, P_2)$, *where* (b) *the transition matrix* $P_2 = \{p_2(i, j)\}$ *is given by the formula*

$$p_2(i,j) = p_1(i,j) + \sum_{x \in X_1 \setminus X_2} p_1(i,x) u_1^{X_2}(x,j), \quad i, j \in X_2. \tag{2.1}$$

Part (a) is immediately implied by the strong Markov property for (Z_n), while the proof of part (b) is straightforward.

Formula (2.1) can be represented in matrix form. This representation is proved, for example, in [10, pp. 114–116]. For the sake of brevity, we will call M_2 the (X_2-) *reduced* model of M_1. (Proposition 2.1 is also true for countable X, with minor modifications.)

An important case is when the set $X_1 \setminus X_2$ consists of one point z. In this case formula (2.1) obviously takes the form

$$p_2(i,j) = p_1(i,j) + \frac{p_1(i,z)p_1(z,j)}{(1 - p_1(z,z))}, \quad (i, j \in X_2). \tag{2.2}$$

According to this formula, each row-vector of the new stochastic matrix P_2 is a linear combination of two rows of P_1 (with the z-column deleted). For a given row of P_2, these two rows are the corresponding row of P_1 and the zth row of P_1. This transformation corresponds formally to one step of the Gaussian elimination method.

It is easy to understand that although the initial and reduced Markov models are different, some of their characteristics will either coincide or be related in a simple way.

The theoretical basis for the GTH/S algorithm is provided by Proposition 2.2, which we formulate here for the case where the set $X_1 \setminus X_2$ consists of one point z. It shows the relation between the invariant distribution in the initial and the reduced models.

PROPOSITION 2.2. *Let* $M_1 = (X_1, P_1)$ *be a Markov model,* $X_1 = X_2 \cup \{z\}$, *and let* $M_2 = (X_2, P_2)$ *be the corresponding* X_2*-reduced Markov model with* $p_2(i, j)$ *defined according to* (2.2). *Let set* X_2 *and state* z *communicate in the model* M_1; *i.e., there are states* i, $j \in X_2$, *such that* $p_1(i, z) > 0$, $p_1(z, j) > 0$. *Then*

(a) *if the invariant distribution* $\pi_2(\cdot)$ *exists in model* M_2, *the invariant distribution* $\pi_1(\cdot)$ *also exists (in* M_1*) and can be calculated by the formulas* $\left(\sum_{y \in X_2} = \sum_y\right)$

$$\pi_1(j) = \alpha_1 \pi_2(j), \quad j \in X_2, \tag{2.3}$$

$$\pi_1(z) = \alpha_1 \sum_i \pi_2(i) p_1(i,z)/s_1 \equiv 1 - \alpha_1, \tag{2.4}$$

where

$$\alpha_1 = 1/\left(1 + \sum_i \pi_2(i) p_1(i,z)/s_1\right), \qquad s_1 = 1 - p_1(z,z). \tag{2.5}$$

(b) *If the invariant distribution* $\pi_1(\cdot)$ *exists in model* M_1*, then the invariant distribution* $\pi_2(\cdot)$ *also exists (in* M_2*) and is given by formula* (2.3)*, with* $\alpha_1 = 1 - \pi_1(z)$.

Relations (2.3) and (2.4) have a transparent probabilistic meaning. The invariant distribution is the long-term proportion of time spent at a state. Therefore, the invariant distributions must be proportional on X_2, i.e., equality (2.3) holds. Formula (2.4) can be easily received from (2.3) and a balance equation for distribution $\boldsymbol{\pi}_1$ at the point z. The formulations of Propositions 2.1 and 2.2, as well as a formal proof of the latter, omitted here, were given in [26].

2.1. The GTH/S algorithm. We now describe briefly the GTH/S algorithm as given in [2] and [19]. In contrast to those papers, we index the states to be eliminated in the order that is customary for Gaussian elimination without pivoting, beginning with state number one.

2.1.1. GTH/S reduction stage (generic for SR algorithms). Let an initial Markov model $(X, P) = M \equiv M_1 = (X_1, P_1)$ be given. A sequence of *stochastic* matrices (P_k), $k = 2, \ldots, n$, is calculated recursively on the basis of formula (2.2), in which the subscripts "1" and "2" are replaced by "k" and "k+1," respectively. Each matrix P_k corresponds to a model $M_k = (X_k, P_k)$, $X_k = \{k, k+1, \ldots, n\}$, and has dimension $(n-k+1) \times (n-k+1)$, and P_n is an identity matrix of dimension 1. A Markov chain in a model M_k is specified by a corresponding Markov chain in the initial model at the moments of its visits to the reduced state space X_k. For the subsequent recovery of $\boldsymbol{\pi}$, only the first (scaled) columns of each of the matrices P_k are used.

An important role in maintaining accuracy is played by the sequence $s_1, s_2, \ldots, s_{n-1}$, where each s_k (see (2.2) and (2.5), where the subscript "1" is again replaced by "k") is calculated as the sum

$$s_k = \sum_{j \neq z} p_k(z, j) = 1 - p_k(z, z)$$

rather than the mathematically equivalent difference. This choice avoids subtractive cancellation without adding significantly to computational effort. The kth step (for $k = 1$) of the reduction phase of the GTH/S algorithm can be represented as

$$P_1 = \begin{bmatrix} a & \mathbf{p}^T \\ s\mathbf{q} & Q \end{bmatrix}, \quad P_2 = Q + \mathbf{q}\mathbf{p}^T, \quad \overline{P}_2 = \begin{bmatrix} a & \mathbf{p}^T \\ \mathbf{q} & P_2 \end{bmatrix}, \tag{2.6}$$

where

$$s \equiv s_1 = \mathbf{p}^T \mathbf{e} = 1 - a,$$

and $\overline{P}_2$ is the matrix stored after the first step of computation. Thus $\mathbf{p}^T$ is the first (zth) row, and $\mathbf{q}$ is the first column (scaled by $s = s_1$) of the matrix P_1, both without the first element $p_1(z, z)$.

2.1.2. GTH/S second (recovery) stage. Three normalizations and tree decomposition. Proposition 2.2 provides the possibility to compute the invariant distribution $\boldsymbol{\pi}_k$ for each of the models M_k on the basis of $\boldsymbol{\pi}_{k+1}$ in the model M_{k+1}, beginning from the trivial invariant distribution $\boldsymbol{\pi}_n = \{1\}$ of the model M_n. This can be done in three different ways. The first way of normalizing is to use formulas

(2.3) and (2.4), i.e., to receive each time vector $\boldsymbol{\pi}_k$. From (2.3) and (2.4), this can be represented as follows:

$$\boldsymbol{\pi}_1 = \alpha_1 \begin{bmatrix} \boldsymbol{\pi}_2^T \mathbf{q} \\ \boldsymbol{\pi}_2 \end{bmatrix} \equiv \begin{bmatrix} 1-\alpha_1 \\ \alpha_1 \boldsymbol{\pi}_2 \end{bmatrix}. \tag{2.7}$$

Notice that this way provides an extra opportunity to increase the accuracy of calculations because the sum of the elements of the obtained vector must equal one. This is the formula used later by REFUND.

Because the goal of the GTH/S algorithm is to produce only the invariant distribution $\boldsymbol{\pi}_1$ in the initial model M_1, that algorithm uses a second method of normalizing. The first equality of (2.7) is used with α_k being replaced by 1, i.e., each new vector is calculated by appending a single element to its predecessor. Only the last vector in the sequence is normalized to produce $\boldsymbol{\pi}_1$.

A third way to normalize is to use the first equality of (2.7) again with α_k now being replaced by s_k. This gives valuable information about a tree decomposition of a Markov chain as follows. In [26] the relationship between GTH/S and the interesting formula discussed below was considered. In their 1979 book on large deviations [1], Freidlin and Wentzell used the following interesting approach to calculate $\boldsymbol{\pi}$ on the basis of a tree decomposition. Their book uses the formula $\pi(x) = q(x)/\sum_{y\in X} q(y)$, where $q(y)$ is defined as follows. Let X be a finite set and P be a stochastic matrix. Let T be a *spanning tree directed* to y. This means that T is a connected graph without cycles (tree), that it contains all the vertices of X (spanning), and that a vertex y is designated as a *root*. In any rooted tree with a root y there is a unique path, directed to y, between any vertex v and y; this direction makes the tree a tree directed to y. Let $G(y) = \{\text{spanning trees on } X \text{ directed to } y\}$. Then $q(y) = \sum_{T\in G(y)} r(T)$, where $r(T) = \prod_{(u,v)\in T} p(u,v)$.

Theorem 1 of [26] establishes that $q(x)$ can be computed by normalizing the recovery steps of the GTH/S algorithm by the replacements $\alpha_k = s_k$, instead of $\alpha_k = 1$ as in GTH/S. This opens the way to use results from Markov chain theory to obtain some results in graph theory. Note also that in [26] the relationship between the SR approach and graph-based computational methods in electrical engineering was noted. In particular the formula mentioned above, which relates $\boldsymbol{\pi}$ to $\mathbf{q}$, is well known in electrical engineering as the star mesh transformation, though the interpretation is quite different.

2.2. The two FUND algorithms. Based on the triangular factorization ($LU = I-P$) provided by the first stage of the GTH/S algorithm, Heyman proposed in (1995) [5] the algorithm FUND to calculate the fundamental matrix Z ($= V+A$) of a regular Markov chain. This led to a fast ($O(n^3)$) algorithm that performed accurately on test problems. A paper of Heyman and O'Leary (1998) [8] gives examples for which the factor U is badly ill-conditioned even though $(I-P)$ is not, and presents a new version of FUND which avoids this instability by modifying U and by introducing pivoting. Both versions of the algorithm are based on the equation

$$Z = A + (I - A)X, \tag{2.8}$$

where X is any solution to

$$(I - P)X = (I - A). \tag{2.9}$$

The solution to the latter equation is not unique, as $(I - P)$ is rank-deficient by 1. In both versions, the triangular factorization produced by the GTH/S algorithm

is used (but in somewhat different ways) to find an X which satisfies (2.9), and then Z is found by substituting X into (2.8).

Notice also that the specific form taken by the triangular factorization depends on the order in which states are eliminated: when states are eliminated beginning with the largest indices (as in the usual presentation of the GTH/S algorithm), the triangular factorization has the form $(I - P) = UL$ (upper followed by lower triangular factors); however, when states are eliminated in the order $1, 2, \ldots$, then the factorization takes the familiar LU form.

2.3. The SR approach. We have cited examples of algorithms (the GTH/S algorithm [2], [19], the two FUND algorithms [5], [8], and the elimination algorithm for optimal stopping [25], [26], [27]) that share a common feature: they are based on a sequence of models in which each model (except the first) is constructed from its predecessor by removing states and recalculating transition probabilities according to Proposition 2.1. We will refer to such algorithms as SR algorithms, and to the general approach to their development as the state reduction (SR) approach. Additional SR algorithms include the algorithms to compute mean first passage times and absorption probabilities in Markov and semi-Markov chains that are discussed by Kohlas in [12] and by Sheskin in [21] and [23], the algorithm of Sheskin [22] for calculating the fundamental matrix for a reducible Markov chain, and the algorithms that Lal and Bhat discuss in [13]. (Sheskin also gives algorithms for matrix inversion [20] and for solving linear systems [24], whose structures are similar to those of the above SR algorithms, but no stochastic interpretation is given for them.) Although Proposition 2.1 does provide for the elimination of several, or even infinitely many, states in a single reduction step, the majority of given examples eliminate one state at each reduction step, and we will confine our discussion to those. Such algorithms must begin, up to minor variation, with the reduction stage of the GTH/S algorithm, whose appearance has stimulated an outpouring of works in recent years.

The algorithms under consideration differ only in their portions that follow the standard reduction stage. In this regard, all of the given examples except one, FUND, are recursive in the sense that the necessary reduction stage is followed by a stage of backward iteration during which some characteristic of, or quantity related to, the smallest model is deduced or calculated, and then the analogous characteristic or quantity in each larger model is inferred or calculated from its counterpart in the adjacent smaller model. (Whenever we use the term "recursive" in what follows, we shall mean it in this sense.) The GTH/S algorithm is a good and well-known example in which backward iteration (up to normalization) retraces all reduction steps.

Another example, also based on Proposition 2.1, is an algorithm for optimal stopping proposed by Sonin in 1995 [25] (see also [26] and [27]). Briefly, it can be described as the construction of a sequence of models where each time a set (often, but not always of size one) of states, which have been shown not to belong to the stopping set, is eliminated, and new transition probabilities are computed on the basis of (2.1) or (2.2). The stopping sets in both models coincide, and this offers the possibility of recursive calculation of the stopping set. In contrast to other state reduction algorithms, in this algorithm the number of steps required is not known in advance. In some SR algorithms (e.g., membership in an optimal stopping set, mean time to reach a designated subset of states) the characteristic to be calculated is preserved by reduction (i.e., coincides on the shared portions of the domains of the initial and reduced models). In these cases part (e.g., mean times) or all (e.g., membership in an optimal stopping set) of the backward stage is trivial, but nearly

all of the SR algorithms are recursive. In the sole exception, FUND, the quantity to be calculated is obtained directly by solving a linear system using a triangular factorization received as a byproduct of the GTH/S reduction stage. The algorithm REFUND introduced in this paper is another example of a two-stage algorithm, with the backward stage being nontrivial and based on an explicit formula.

3. Sequential calculation of the fundamental matrix. Let P be a regular (i.e., irreducible, aperiodic with no transients states) finite stochastic matrix. Equivalently, there is some $k > 0$ for which the matrix P^k has all positive elements. We have already defined $\boldsymbol{\pi}$, A, Z, and V in (1.1), (1.2), (1.3), and (1.4), respectively. We also have

$$PA = AP = A, \quad A^n = A \quad \text{for any } n = 1, 2, \dots. \tag{3.1}$$

The following theorem provides a useful characterization of the matrix V. We will substantially use this theorem in the construction of our main result. All elements of this theorem are well known, but we fail to find such a formulation. (Formula (3.2) below is equivalent to Theorem 2.3 in [15]. Compare this also with Theorem 1 in [5] which is very similar but without the uniqueness, or compare it with results in Appendix A.5 in [18].)

THEOREM 3.1. *Let $M = (X, P)$ be a regular Markov model, π be its invariant measure, $A = \lim_{n\to\infty} P^n$, Z be the fundamental matrix, and $V = Z - A$. Then V is the unique solution of the system of equations*

$$V = (I - A) + PV = (I - A) + VP, \tag{3.2}$$

$$\boldsymbol{\pi}^T V = 0, (AV = 0) \tag{3.3}$$

and also satisfies

$$V\mathbf{e} = 0, (VE = VA = 0). \tag{3.4}$$

Note that the equations in (3.2) are of Bellman type in forward and backward time, and that explains why the group inverse (fundamental) matrix plays a role in the theory of Markov decision processes with average criterium. The formulas (3.3) and (3.4) just say that a scalar product of the invariant vector and any column of V is equal to zero, and the sum of every row of V is also equal to zero. Both relationships have a simple probabilistic meaning according to (1.5).

Proof. Let V be the group inverse. Then from (1.4) and (3.1) we have $PV = P(I - A + \sum_{n=1}(P^n - A)) = (P - A) + (P^2 - A) + \dots = V + A - I$, i.e., the first of the formulas (3.2). The second formula in (3.2) is derived similarly. Formulas (3.3) and (3.4) follow immediately from (1.5) since $E_\pi \eta^n(y) = n\pi(y)$ for all n, and $\sum_y E_x \eta^n(y) = n = \sum_y n\pi(y)$. (Formally, from (1.4) and (3.2) we have $AV = A(I - A) + A\sum_{n=1}(P^n - A) = 0$, i.e., (3.3).) Similarly $VA = 0$, i.e., (3.4). To prove the uniqueness of V, let V and V' be two solutions of (3.2). Then $V - V' = P(V - V')$. According to the well-known statement that for a regular matrix P any solution of equation $\mathbf{x} = P\mathbf{x}$ has the form $c\mathbf{e}$, where c is a constant (see [10, Th. 4.1.7]), thus each column of $V - V'$ is $c\mathbf{e}$ for some constant c. By (3.3) $\pi^T(V - V') = 0$, i.e., all these constants are equal to zero. □

Let $M_i = (X_i, P_i), i = 1, 2$ be two models with $X_1 = X_2 \cup \{z\}$; M_2 is an X_2-reduced model of M_1, i.e., P_2 is calculated by formula (2.2). Then according to Proposition 2.2 the relation between invariant measures in these models is given by

the formulas (2.3) and (2.4) with the constants defined by (2.5). Without loss of generality $X_1 = \{1, 2, \ldots, n\}$, $X_2 = \{2, 3, \ldots, n\}$, i.e., $z = 1$, but we will continue to use the letter z. Denote $\sum_j \equiv \sum_{j \in X_2} \equiv \sum_{j \geq 2}$. Consider $x, y \in X_2 \subset X_1$. If in model M_1 state y can be reached from x in k steps, then obviously in M_2 state y can be reached from x in k or fewer steps. Therefore, if P_1 is a regular matrix, then P_2 will also be a regular matrix.

Our aim is to express the group inverse V_1 through the group inverse V_2. We will denote the row-vectors of matrix V_2 as $\mathbf{v}_i$ and the columns as $\mathbf{v}^j$, i.e., $\mathbf{v}_i = (v(i, \cdot))$, $\mathbf{v}^j = (v(\cdot, j))$. Let us denote vector $\boldsymbol{\pi}$=$\{\pi_2(2), \pi_2(3), \ldots, \pi_2(n)\}$.

For the model M_1, let the constants $s = s_1$ and $\alpha = \alpha_1$ be given by (2.5); also let vectors $\mathbf{p}$ and $\mathbf{q}$ be given by (2.6). The scalar product of vectors $\mathbf{x}$ and $\mathbf{y}$ will be denoted by $\mathbf{x}^T\mathbf{y}$.

We define the (column) vectors $\mathbf{r}, \mathbf{t}, \mathbf{c}$ and the constant c by

$$\mathbf{r} = \alpha V_2 \mathbf{q}, \tag{3.5}$$

$$\mathbf{t}^T = \frac{(1-\alpha)}{s} \mathbf{p}^T V_2, \tag{3.6}$$

$$c = \frac{(1-\alpha)}{s}(\alpha + \mathbf{p}^T\mathbf{r}), \qquad \text{and} \qquad \mathbf{c} = c\boldsymbol{\pi}_2 - \mathbf{t}. \tag{3.7}$$

It is clear from (3.4) and (3.6) that $\mathbf{t}^T\mathbf{e} = 0$; hence (3.7) implies that $\mathbf{c}^T\mathbf{e} = c$.

Our main result is the following.

THEOREM 3.2. *Let $M_2 = (X_1 \backslash \{z\}, P_2)$ be the reduced model of $M_1 = (X_1, P_1)$,as defined in section 2, with P_2 related to P_1 by (2.2). Denote their associated group inverse matrices by V_1, V_2, and their invariant vectors by $\boldsymbol{\pi}_1$,$\boldsymbol{\pi}_2$. If states are indexed so that $z = 1$, vectors $\mathbf{r}$ and $\mathbf{c}$ and constant c are as defined in (3.5) through (3.7), $\alpha = \alpha_1$ is given by (2.5), and $\mathbf{q}$ is given by (2.6), then the group inverse matrix V_1 can be described in terms of four matrix blocks as follows:*

$$V_1 = \begin{bmatrix} v_{11} & \mathbf{v}_{12}^T \\ \mathbf{v}_{21} & W_1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha}{1-\alpha} c & \frac{-\alpha}{1-\alpha}\mathbf{c}^T \\ \mathbf{r} - c\mathbf{e} & V_2 + U \end{bmatrix}, \quad \textit{where } U = -\mathbf{r}\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}^T. \tag{3.8}$$

Proof. It is possible to prove the result by checking that the matrix V_1 given in (3.8) satisfies Theorem 3.1. Instead, our proof will show explicitly how we arrive at each block of (3.8). To simplify our notation we will omit the index "2" in all references to the matrices P_2, $A_2 = \mathbf{e}\boldsymbol{\pi}_2^T, V_2$, identity matrix $I_2 \equiv I$, and to the invariant measure $\boldsymbol{\pi}_2$=$\boldsymbol{\pi}$.

The first step is to express $v_1(i, j)$ for $i, j \neq z$ in terms of the elements of matrix $V_2 \equiv V$. Formula (3.2) (the first of two equalities), applied to V_1, implies (using the Kronecker symbol $\delta(i, j)$)

$$v_1(i, j) = \delta(i, j) - \pi_1(j) + p_1(i, z)v_1(z, j) + \sum_{k \neq z} p_1(i, k)v_1(k, j). \tag{3.9}$$

Recall that $z \equiv 1$. When $i = z$, $j \neq z$, formula (3.9), using $1 - p_1(z, z) = s$ becomes

$$sv_1(z, j) = -\pi_1(j) + \sum_{k \neq z} p_1(z, k)v_1(k, j). \tag{3.10}$$

Substituting the expression for $v_1(z, j)$ from (3.10) into (3.9) we obtain for $i \geq 2$, $j \geq 2$,

$$v_1(i, j) = \delta(i, j) - [1 + p_1(i, z)/s]\pi_1(j) \tag{3.11}$$

$$+\sum_{k\neq z}[p_1(i,k)+p_1(i,z)p_1(z,k)/s]v_1(k,j).$$

Now, replacing $\pi_1(j)$ by $\alpha\pi(j)$ (formula (2.3)) and the expression in the brackets in the sum by $p(i,k)$ (formula (2.2) with $p_2\equiv p$), we can rewrite formula (3.11) for $i\geq 2$, $j\geq 2$ in matrix form (the restriction of V_1 for i, $j\neq z$ is denoted by W_1)

$$W_1 = I - TA + PW_1, \tag{3.12}$$

where T is a diagonal matrix with elements equal to $\alpha(1+p_1(i,z)/s)$, $i\geq 2$.

Recall (see (3.8)) that $U=W_1-V_2\equiv W_1-V$. Then subtracting $V=(I-A)+PV$ (formula (3.2) for $V=V_2$) from (3.12) we obtain the equation for U

$$U = GA + PU, \tag{3.13}$$

where $G=I-T$ is a diagonal matrix with elements (using $\alpha=1-\pi_1(z)$)

$$1-\alpha(1+p_1(i,z)/s) = \pi_1(z)-\alpha p_1(i,z)/s, \quad i\geq 2. \tag{3.14}$$

LEMMA 3.3. *Any solution of* (3.13) *has the form*

$$U = VGA + C, \tag{3.15}$$

where the jth column of C is $c(j)\mathbf{e}$, $c(j)$ a constant.

Proof of Lemma 1. As we mentioned earlier, it is well known that any matrix solution of $X=PX$ for a regular P is a matrix C with constant columns. Hence any solution of (3.13) is a particular solution of this equation plus such a matrix C. Therefore we need to show only that the matrix VGA is a solution of (3.13). By formula (3.2) for model M_2, ($P\equiv P_2$, $A\equiv A_2$), we have $VGA=GA-AGA+PVGA$. Let us show that $AGA=0$. It is easy to see this is equivalent to $\boldsymbol{\pi}^T\mathbf{g}=0$, where $\mathbf{g}$ is a vector of diagonal elements of G, i.e., given by (3.14). Using the equality $\sum_i\pi(i)=1$ and formula (2.4), we obtain

$$\boldsymbol{\pi}^{\boldsymbol{T}}\mathbf{g}=\sum_i\pi(i)(\pi_1(z)-\alpha p_1(i,z)/s)=\pi_1(z)-\alpha\sum_i\pi(i)p_1(i,z)/s=0,$$

which establishes Lemma 1.

Thus we have calculated $W_1=V+U$ (the restriction of V_1 for i, $j\neq z$) up to unknown constants $c(j)$. To finish the calculation of V_1 we need to show that unknown constants $c(j)$ (matrix C in (3.15)) coincide with the components of vector $\mathbf{c}$ defined in (3.7), and to provide formulas for the first row and the first column of V_1.

First, we can simplify VGA further, noticing that by (3.14) $G=\pi_1(z)I+D$, where $d(i,z)=-\alpha p_1(i,z)/s, i\geq 2$, and $VIA=0$ by formula (3.4). Therefore $U=VDA+C$, and we have $W_1=V(I+DA)+C$. Thus, using $d(i,z)$ and the definition of vector $\mathbf{r}$ in (3.5), we obtain for k, $j\geq 2$

$$v_1(k,j) = v(k,j)-\alpha\pi(j)\sum_i v(k,i)p_1(i,z)/s + c(j) \tag{3.16}$$

$$= v(k,j)-\pi(j)r(k)+c(j), \tag{3.17}$$

which verifies that $W_1=V+U$, as claimed in (3.8).

Using (3.16), we can rewrite the jth column of (3.3) for $V = V_1$ as

$$\pi_1(z)v_1(z,j) + \sum_k \pi_1(k)[(k,j) - \pi(j)r(k) + c(j)] = 0, \quad j \geq 2.$$

Using the equalities $\pi_1(k) = \alpha\pi(k)$, $\sum_k \pi(k)v(k,j) = 0$ (i.e., $\boldsymbol{\pi}^T V = 0$; see (3.3) for $V = V_2$), $\sum_k \pi_1(k) = 1 - \pi_1(z) = \alpha$, and the equality $\boldsymbol{\pi}^T\mathbf{r} = \alpha$ $\boldsymbol{\pi}^T(V\mathbf{q}) = \alpha$ $(\boldsymbol{\pi}^T V)\mathbf{q} = 0$, we obtain

$$(1-\alpha)v_1(z,j) + \alpha c(j) = 0, \quad j \geq 2, \tag{3.18}$$

which is equivalent to $\mathbf{v}_{12}^T = -\mathbf{c}^T \frac{\alpha}{1-\alpha}$ given in (3.8).

It remains to be verified that $c(j)$ satisfy (3.7). Substituting $v_1(k,j)$ from (3.16) into (3.10) and using the equalities $\sum_k p_1(z,k) = 1 - p_1(z,z) = s$, $\pi_1(j) = \alpha\pi(j)$, the definition of vector $\mathbf{t}$ (3.6), and the definition of constant c in (3.7), we can rewrite (3.10) as

$$v_1(z,j) - c(j) = (-c\pi(j) + t(j))/(1-\alpha), \quad j \geq 2. \tag{3.19}$$

Using (3.18) to replace $v_1(z,j)$ in (3.19), we obtain (3.7).

Now the entries of $\mathbf{v}_{21}$ in (3.8) can be found using equality (3.4) as follows. For $i \geq 2$, using $W_1 = V - \mathbf{r}\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}^T$ from (3.8) we obtain

$$v_1(i,z) = -\sum_j v_1(i,j) = -\sum_j v(i,j) + r(i)\sum_j \pi(j) - \sum_j c(j).$$

The first sum of the rightmost expression is equal to zero by (3.4) for $V = V_2$. Using $\sum \pi_j = 1$, $\sum c(j) = c$, we obtain $\mathbf{v}_{21} = \mathbf{r} - c\mathbf{e}$. The 1×1 block v_{11} is obtained similarly, using (the now-established representation of $\mathbf{v}_{21}$ in) (3.8) and (3.4) for $V = V_1$. ∎

Remark. Note that only the first two expressions in (3.2) have been used in this section and that an additional opportunity to check (increase) the accuracy of computations of $V = V_1$ is provided by considering the rightmost expression in that equation.

4. The REFUND algorithm and numerical tests.

4.1. The REFUND algorithm. The results of the previous section lead to a recursive algorithm, REFUND, to calculate the group inverse matrix $V = V_1$ of $n \times n$ stochastic matrix $P = P_1$. The reduction stage implicitly produces a finite sequence $M = M_1, M_2, \ldots, M_n$ of models, where in each model $M_k = (X_k, P_k)$, X_k is a state space, and P_k is a stochastic matrix. A single step of reduction, in which P_2 (and $\bar{P}_2$) are calculated from P_1, was depicted in (2.6), which also describes (with obvious changes of index) any such step yielding P_{k+1} from P_k. During computation, the array in which P was originally stored is altered by repeated application of (2.6): at each step P_k is replaced by $\bar{P}_{k+1}$. Thus the reduction stage duplicates (up to minor variation in the order in which states are eliminated) that of the GTH/S algorithm and is shared by all of the SR algorithms described in section 2. The output of the reduction stage is the matrix/array $\bar{P}_n = \bar{P}$, containing (for $k = 1, \ldots, n-1$) the vectors $\mathbf{p}_k^T$ and $\mathbf{q}_k$ given (without subscripts) by (2.6).

REFUND's recovery stage is initialized with information from the smallest (one state) model M_n: the stochastic matrix $P_n = [1]$, invariant distribution $\boldsymbol{\pi}_n = [1]$, and group inverse matrix $V_n = [0]$. Each recovery step ($k = n-1, \ldots, 1$) begins with

the pair ($\boldsymbol{\pi}_{k+1}$, V_{k+1}) and calculates ($\boldsymbol{\pi}_k$,V_k). The vector $\boldsymbol{\pi}_k$ is obtained (with the appropriate change of index) from (2.7); then the matrix V_k is calculated (with similar reindexing) from (3.8), using information from (3.5) through (3.7).

Remark. As was mentioned above, to avoid unnecessary subtractive cancellation, the GTH/S algorithm calculates s_k $(= s)$ (see the formula immediately above (2.6)) as the sum $\sum_{j\neq z} p_k(z,j)$. Similarly, REFUND calculates the scalars $\frac{\alpha}{1-\alpha}$ (in (3.8)) and $\frac{1-\alpha}{s}$ (in (3.5) and (3.7)) as $\frac{1}{\boldsymbol{\pi}_2^T\mathbf{q}}$ and $\frac{\boldsymbol{\pi}_2^T\mathbf{q}}{s\left(1+\boldsymbol{\pi}_2^T\mathbf{q}\right)}$, respectively. Thus in all division operations, divisors are calculated without subtraction from the output of the GTH/S algorithm, which contains no subtraction at all. Since the group inverse matrix and fundamental matrix generally contain both positive and negative elements, any algorithm calculating either must contain subtraction. Whether or not some of these subtractions involve numbers that are "nearly equal" (thus reducing the number of significant digits in some element) depends on the structure of the particular matrix given as input. Note that the explicit formula on which REFUND is based provides an opportunity to analyze this question directly. We are going to address the application of REFUND to the NCD case in a separate paper.

Operation counts. The number of arithmetic operations encountered by the REFUND algorithm is an $O(n^3)$ function f, where the stochastic matrix $P = P_1$ is $n \times n$. Taken together, the calculations of the reduction stage, and those entailed in the recovery of $\boldsymbol{\pi}$, duplicate the GTH/S algorithm, which is $\Theta(\frac{2}{3}n^3)$. The additional operations required to recover V contribute $\Theta(\frac{7}{3}n^3)$ to the dominant term of f, which is affected only by the products $V_2\mathbf{q}$ (in (3.5)), $\mathbf{p}^TV$ (in (3.6)), $\mathbf{r}\boldsymbol{\pi}_2^T$ (and not $\mathbf{e}\mathbf{c}^T$) in the lower right block of (3.8), and by the matrix addition and subtraction in that block.

4.1.1. Example. We include the calculations for the well-known "Land of Oz" example from Kemeny and Snell [10].

Reduction. Initialize: $\bar{P}_0 = P\ (= P_1)$.

$$\bar{P}_0 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{bmatrix} \rightarrow \bar{P}_1 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 3/4 \\ 1/2 & 3/8 & 5/8 \end{bmatrix} \rightarrow \bar{P}_2 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1 & 1/4 & 3/4 \\ 1/2 & 1/2 & 1 \end{bmatrix}.$$

Recovery. Initialize: $\boldsymbol{\pi}_3 = \begin{bmatrix} 1 \end{bmatrix}, V_3 = \begin{bmatrix} 0 \end{bmatrix}$.

Step 1:

$\boldsymbol{\pi}_3^T\mathbf{q}_2 = \begin{bmatrix} 1 \end{bmatrix}\begin{bmatrix} 1/2 \end{bmatrix} = 1/2$, $\alpha_2 = \frac{1}{1+\boldsymbol{\pi}_3^T\mathbf{q}_2} = 2/3$, $\boldsymbol{\pi}_2 = \alpha_2 \begin{bmatrix} \boldsymbol{\pi}_3^T\mathbf{q}_2 \\ \boldsymbol{\pi}_3 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 2 \end{bmatrix}$;

$\mathbf{r}_2 = \alpha_2 V_3\mathbf{q}_2 = \begin{bmatrix} 0 \end{bmatrix}$; $\mathbf{t}_2 = \left(\frac{\alpha_2\boldsymbol{\pi}_3^T\mathbf{q}_2}{s_2}\right)\mathbf{p}_2^TV_3 = \begin{bmatrix} 0 \end{bmatrix}$,

$c_2 = \left(\frac{\alpha_2\boldsymbol{\pi}_3^T\mathbf{q}_2}{s_2}\right)(\alpha_2 + \mathbf{p}_2^T\mathbf{r}_2) = 8/27$, $\mathbf{c}_2 = c_2\boldsymbol{\pi}_3 - \mathbf{t}_2^T = \frac{1}{27}\begin{bmatrix} 8 \end{bmatrix}$,

$$V_2 = \begin{bmatrix} \frac{\alpha_2}{1-\alpha_2}c_2 & \frac{-\alpha_2}{1-\alpha_2}\mathbf{c}_2^T \\ \mathbf{r}_2 - c_2\mathbf{e} & V_3 - \mathbf{r}_2\boldsymbol{\pi}_3^T + \mathbf{e}\mathbf{c}_2^T \end{bmatrix} = \frac{1}{27}\begin{bmatrix} 16 & -16 \\ -8 & 8 \end{bmatrix}.$$

Step 2:

$$\boldsymbol{\pi}_2^T\mathbf{q}_1 = 2/3,\ \alpha_1 = \tfrac{1}{1+\boldsymbol{\pi}_2^T\mathbf{q}_1} = 3/5,\ \boldsymbol{\pi}_1 = \alpha_1 \begin{bmatrix} \boldsymbol{\pi}_2^T\mathbf{q}_1 \\ \boldsymbol{\pi}_2 \end{bmatrix} = \tfrac{1}{5}\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix};$$

$$\mathbf{r}_1 = \alpha_1 V_2\mathbf{q}_1 = \tfrac{1}{45}\begin{bmatrix} 8 \\ -4 \end{bmatrix},\ \mathbf{t}_1 = \left(\tfrac{\alpha_1\boldsymbol{\pi}_2^T\mathbf{q}_1}{s_1}\right)\mathbf{p}_1^T V_2 = \tfrac{1}{135}\begin{bmatrix} 8 & -8 \end{bmatrix},$$

$$c_1 = \left(\tfrac{\alpha_1\boldsymbol{\pi}_2^T\mathbf{q}_1}{s_1}\right)\left(\alpha_1 + \mathbf{p}_1^T\mathbf{r}_1\right) = 112/225,\ \mathbf{c}_1 = c_1\boldsymbol{\pi}_2 - \mathbf{t}_1^T = \tfrac{1}{225}\begin{bmatrix} 24 \\ 88 \end{bmatrix},$$

$$V_1 = \begin{bmatrix} \frac{\alpha_1}{1-\alpha_1}c_1 & \frac{-\alpha_1}{1-\alpha_1}\mathbf{c}_1^T \\ \mathbf{r}_1 - c_1\mathbf{e} & V_2 - \mathbf{r}_1\boldsymbol{\pi}_2^T + \mathbf{e}\mathbf{c}_1^T \end{bmatrix} = \tfrac{1}{75}\begin{bmatrix} 56 & -12 & -44 \\ -24 & 48 & -24 \\ -44 & -12 & 56 \end{bmatrix}.$$

4.2. Numerical tests.

4.2.1. Implementation. The REFUND algorithm was coded and run in MATLAB, using IEEE arithmetic with 16 decimal digit working precision. Pivoting is easily incorporated, but tabulated results are for tests runs with pivoting disabled.

4.2.2. Measures of accuracy. We define measures of residual error for all conditions required in Theorem 3.1. So that these measures will continue to be appropriate during our later comparison of REFUND to FUND, we first let $D = \mathrm{diag}(P\mathbf{e})$ be the diagonal matrix whose nonzero entries are the rowsums of P; thus (as now) when P is stochastic, $D = I$. Define (block) matrices $\Phi = \begin{bmatrix} (D-P)^T & \boldsymbol{\pi} \end{bmatrix}^T$ and $S = S(\beta) = \begin{bmatrix} (I-A)^T & \beta\boldsymbol{\pi} \end{bmatrix}^T$. When $\beta = 0$, V is the unique solution to

$$\Phi X = S(\beta), \tag{4.1}$$

which combines the first equality of (3.2) with (3.3). Now let $\tilde{V}$ denote the calculated value for matrix V; let $H = \Phi\tilde{V} - S(0)$; and (for each $j = 1, \ldots, n$) $\mathbf{h}_j$ will denote the jth column of H. Let $W = \tilde{V}P - P\tilde{V}$, with jth column $\mathbf{w}_j$; and let $\mathbf{e}$ denote a vector of ones. We define measures $\delta_1, \delta_2, \delta_3$ by

$$\delta_1 = \max_j\{||\mathbf{h}_j||_2\}, \qquad \delta_2 = ||\tilde{V}\mathbf{e}||_\infty, \qquad \delta_3 = \max_j\{||\mathbf{w}_j||_2\}. \tag{4.2}$$

Condition number. The $n\times n$ matrix $(I-P)$ in (3.2) has rank $n-1$, so its matrix condition number is undefined. To compare relative error to problem condition, we use

$$\kappa = \sigma_{\max}/\sigma_{\min}, \tag{4.3}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are, respectively, the largest and smallest nonzero singular values of $(I-P)$. Sonin and Thornton [29] remark that κ, rather than the analogous (and larger) ratio of singular values of Φ in (4.1), is a true relative condition number for the calculation of V. The ratio κ was used earlier by Heyman [5] and by Heyman and O'Leary [8] for the problem of calculating the fundamental matrix.

4.2.3. Test problems. Two sets of test problems were used. The first problem set consists of seven problems and comes from Harrod and Plemmons [4]. Although the stochastic matrices in this set are not large, the set contains problems that are numerically difficult and has provided test problems used in Heyman and Reeves [7], Heyman and O'Leary [6], two examples in Heyman [5], and Sonin and Thornton [29]. Three of these problems have condition numbers of more than 10^5, and one exceeds

10^7. The last four problems involve NCD chains. Such chains have subsets of states between which transitions occur only rarely and are known to be ill-conditioned. They are discussed in several works of Stewart, for example, in [30] and [32]. The test matrices are available in any of the foregoing sources, and we omit them here.

The problems in the second set were used by Heyman and O'Leary [8] to test the stabilized version of FUND and concern continuous time Markov chains. The matrix entries represent transition rates, not probabilities, so these matrices, as given in [8], are not stochastic. State spaces for these chains have the form $\{0, 1, 2, \ldots, n\}$, so the resulting matrices have dimension $(n+1)\times(n+1)$. All transition rates are zero except $p_{i,i+1} = \lambda$ (for $0 \le i < n$) and $p_{i,i-1} = i$ (for $0 < i \le n$). As Heyman and O'Leary do in [8], we solve these problems for $n = 5, 10, 15, \ldots, 50$, and choose $\lambda = n$ in each problem.

4.2.4. Test results and interpretation. For each test problem of the first set Sonin and Thornton [29] tabulate δ_1, δ_2, δ_3 (given by (4.2)), and scaled measures $\delta_1/(\kappa\varepsilon)$, $\delta_2/(\kappa\varepsilon)$, $\delta_3/(\kappa\varepsilon)$, where κ is the condition number given by (4.3), and $\varepsilon = 2.22 \times 10^{-16}$ (reported by MATLAB) is the smallest positive floating point number γ such that the floating point result $1 + \gamma \ne 1$. For all problems of the first set, $\delta_1/(\kappa\varepsilon) \le 0.48$, $\delta_2/(\kappa\varepsilon) \le 0.65$, $\delta_3/(\kappa\varepsilon) \le 0.74$, indicating that all requirements of Theorem 3.1 were satisfied as well as can be expected, given problem conditioning. Now δ_1 corresponds to conditions which were used in calculation. That both $\delta_2/(\kappa\varepsilon)$ and $\delta_3/(\kappa\varepsilon)$ were consistently small is significant, since REFUND does not make explicit use of either the rowsum conditions (3.4) or of the commutativity conditions (second equality of (3.2)).

Similar results are also tabulated in [29] for a less well-known set of ten problems. For all of these $\delta_1/(\kappa\varepsilon) \le 0.95$, $\delta_2/(\kappa\varepsilon) \le 2.88$, $\delta_3/(\kappa\varepsilon) \le 1.52$, still indicating close agreement with all requirements.

4.3. Comparison with the FUND algorithms. The FUND algorithms described in section 2.2 calculate the fundamental matrix Z defined in (1.3). Since both algorithms require that π be calculated by the GTH/S algorithm, either V or Z can be obtained accurately from the other by (1.4) at a cost that does not affect the dominant term of the workload.

Speed. In addition to the $\Theta(\frac{2}{3}n^3)$ operations required by reduction, REFUND's $\Theta(\frac{7}{3}n^3)$ operation count is slightly smaller than the comparable figure for FUND, $\Theta(\frac{8}{3}n^3)$.

Much more substantial savings in time are possible in some cases: because REFUND is recursive while FUND is not, REFUND can reduce model (X, P) to any submodel (X_k, P_k) for which $\boldsymbol{\pi}_k$ and V_k are available and begin recovery immediately.

Accuracy. To make a direct comparison between REFUND's accuracy and that of the stabilized (1998) FUND, Sonin and Thornton [29] tested REFUND using the second problem set, which was used by Heyman and O'Leary in [8]. The measures of accuracy that were defined in (4.2) remain appropriate, since V remains the unique solution to (4.1) for $\beta = 0$. But for $\beta = 1$, (4.1) subsumes (2.8) and (2.9), and Z becomes the unique solution. We let $\tilde{Z}$ denote the matrix computed by the stabilized FUND. For each test problem, [29] tabulates δ_1, measured for REFUND, beside a comparable measure for FUND obtained from [8]. Their measure $r_{imp} = r_{improved}$ is the norm (2-norm assumed) of the residual error in the last column $\tilde{\mathbf{z}}_n$ of Z relative to the first equality of (3.2) only: $r_{imp} = \|\mathbf{e}_n - \pi_n\mathbf{e} - (D - P)\tilde{\mathbf{z}}_n\|$.

Since δ_1 measures residual errors in all columns and includes both conditions (3.2) and (3.3), it is a (slightly) more sensitive measure of error than r_{imp} is: If for the

same matrix P, $\tilde{V}$ is calculated by REFUND and δ_1 is calculated from $\tilde{V}$ as discussed, and similarly $\tilde{Z}$ is calculated by FUND and r_{imp} calculated from $\tilde{Z}$, and the residual errors are identical (i.e., $\Phi\tilde{V} - S(0) = \Phi\tilde{Z} - S(1)$), then $\delta_1 \geq r_{imp}$. But for every test problem, $\delta_1 < r_{imp}$; thus (at least for this problem set) REFUND appears to be the more accurate algorithm. Also, the results tabulated for REFUND were obtained without pivoting, while those tabulated for FUND come from the stabilized version, which pivots in order to achieve stability.

Structure. Because REFUND is a recursive algorithm like most other SR algorithms, which also share the reduction stage of the GTH/S algorithm, it can be readily implemented along with other SR algorithms in computer code that produces a variety of information in one run. When calculating several characteristics of a system together in such a simultaneous recursion, it becomes possible to exploit any known relationships among them, either to save time, to improve accuracy, or to derive further information about the system under study. Of course REFUND produces results for submodels and allows a user to reduce to and restart from any solved submodel. Also, the explicit formula on which REFUND is based provides a new means by which to analyze and compare various cases, e.g., sparse matrices, NCD chains, or decompositions required by parallel implementations.

REFERENCES

[1] M. I. Freidlin and A. D. Wentzell, *Perturbations of Stochastic Dynamic Systems*, Springer-Verlag, New York, 1984.

[2] W. K. Grassmann, M. I. Taksar, and D. P. Heyman, *Regenerative analysis and steady-state distributions for Markov chains*, Oper. Res., 33 (1985), pp. 1107–1116.

[3] W. K. Grassmann, *Means and variances in Markov reward systems*, 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. 193–204.

[4] W. J. Harrod and R. J. Plemmons, *Comparison of some direct methods for computing stationary distributions of Markov chains*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 453–469.

[5] D. P. Heyman, *Accurate computation of the fundamental matrix of a Markov chain*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 954–963.

[6] D. P. Heyman and D. P. O'Leary, *What is fundamental for Markov chains: First passage times, fundamental matrices, and group generalized inverses*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer, Boston, 1995, pp. 151–159.

[7] D. P. Heyman and A. Reeves, *Numerical solution of linear equations arising in Markov chain models*, ORSA J. Comput., 1 (1989), pp. 52–60.

[8] 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.

[9] D. L. Isaacson and R. W. Madsen, *Markov Chains: Theory and Applications*, Wiley, New York, 1976.

[10] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Van Nostrand Reinhold, Princeton, 1960.

[11] J. G. Kemeny, J. L. Snell, and A. V. Knapp, *Denumerable Markov Chains*, Springer-Verlag, New York, 1976.

[12] J. Kohlas, *Numerical computation of mean passage times and absorption probabilities in Markov and semi-Markov models*, Z. Oper. Res., 30 (1986), pp. A197–A207.

[13] R. Lal and U. N. Bhat, *Reduced systems algorithms for Markov chains*, Management Sci., 34 (1988), pp. 1202–1220.

[14] B. F. Lamond and M. L. Puterman, *Generalized inverses in discrete time Markov decision processes,* SIAM J. Matrix Anal. Appl., 10 (1989), pp. 118–134.

[15] 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.

[16] C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, IMA Vol. Math. Appl. 48, Springer, New York, 1993.

[17] C. O. O'Cinneide, *Entrywise perturbation theory and error analysis for Markov chains*, Numer. Math., 65 (1993), pp. 109–120.
[18] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York, 1994.
[19] T. J. Sheskin, *A Markov partitioning algorithm for computing steady-state probabilities*, Oper. Res., 33 (1985), pp. 228–235.
[20] T. J. Sheskin, *Matrix inversion by augmentation and reduction*, Internat. J. Math. Ed. Sci. Tech., 22 (1991), pp. 103–110.
[21] T. J. Sheskin, *Computing absorption probabilities for a Markov chain*, Internat. J. Math. Ed. Sci. Tech., 22 (1991), pp. 799–805.
[22] T. J. Sheskin, *Computing the fundamental matrix for a reducible Markov chain*, Math. Mag., 68 (1995), pp. 393–398.
[23] T. J. Sheskin, *Computing mean first passage times for a Markov chain*, Internat. J. Math. Ed. Sci. Tech., 26 (1995), pp. 729–735.
[24] T. J. Sheskin, *A partitioning algorithm for solving systems of linear equations*, Internat. J. Math. Ed. Sci. Tech., 27 (1996), pp. 641–648.
[25] I. M. Sonin, *Two simple theorems in the problems of optimal stopping*, in Proceedings of the Eighth INFORMS Applied Probability Conference, Atlanta, 1995, INFORMS, Linthicum, MD, 1995.
[26] I. M. Sonin, *The state reduction and related algorithms and their applications to the study of Markov chains, graph theory and the optimal stopping problem*, Adv. Math., 145 (1999), pp. 159–188.
[27] I. M. Sonin, *The Elimination Algorithm for the Problem of Optimal Stopping,* Math. Methods Oper. Res, 49 (1999), pp. 111–123.
[28] I. M. Sonin and J. R. Thornton, *The elimination algorithm and its applications*, in Proceedings of the Ninth INFORMS Applied Probability Conference, Boston, 1997, INFORMS, Linthicum, MD, 1997.
[29] I. M. Sonin and J. R. Thornton, *Computational properties of algorithm REFUND for the fundamental/group inverse matrix of a Markov chain,* in Numerical Solution of Markov Chains, B. Plateau, W. J. Stewart, and M. Silva, eds., Proceedings of the Third International Workshop, Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 131–148.
[30] J. R. Koury, D. F. McAllister, and W. J. Stewart, *Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 164–186.
[31] W. J. Stewart, ed., *Computations with Markov Chains*, Kluwer, Boston, 1995.
[32] W. J. Stewart, *Numerical methods for computing stationary distributions of finite irreducible Markov Chains*, in Advances in Computational Probability, W. Grassmann, ed., Kluwer, Boston, to appear.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 225–242
© 2001 Society for Industrial and Applied Mathematics

NONSYMMETRIC ALGEBRAIC RICCATI EQUATIONS AND WIENER–HOPF FACTORIZATION FOR M-MATRICES*

CHUN-HUA GUO†

Abstract. We consider the nonsymmetric algebraic Riccati equation for which the four coefficient matrices form an M-matrix. Nonsymmetric algebraic Riccati equations of this type appear in applied probability and transport theory. The minimal nonnegative solution of these equations can be found by Newton's method and basic fixed-point iterations. The study of these equations is also closely related to the so-called Wiener–Hopf factorization for M-matrices. We explain how the minimal nonnegative solution can be found by the Schur method and compare the Schur method with Newton's method and some basic fixed-point iterations. The development in this paper parallels that for symmetric algebraic Riccati equations arising in linear quadratic control.

Key words. nonsymmetric algebraic Riccati equations, M-matrices, Wiener–Hopf factorization, minimal nonnegative solution, Schur method, Newton's method, fixed-point iterations

AMS subject classifications. 15A24, 15A48, 65F30, 65H10

PII. S0895479800375680

1. Introduction.

Symmetric algebraic Riccati equations have been the topic of extensive research. The theory, applications, and numerical solution of these equations are the subject of the monographs [20] and [24]. The algebraic Riccati equation that has received the most attention comes from linear quadratic control. It has the form

$$XDX - XA - A^TX - C = 0, \tag{1.1}$$

where $A, C, D \in \mathbb{R}^{n\times n}$; C, D are symmetric positive semidefinite; the pair (A, D) is stabilizable, i.e., there is a $K \in \mathbb{R}^{n\times n}$ such that $A - BK$ is stable (a square matrix is stable if all its eigenvalues are in the open left half-plane); and the pair (C, A) is detectable, i.e., (A^T, C^T) is stabilizable. It is well known that (1.1) has a unique symmetric positive semidefinite solution X and the matrix $A - DX$ is stable (see [20], for example). This solution is the one required in applications and can be found numerically by iterative methods [3, 7, 10, 12, 13, 19] and subspace methods [4, 6, 22, 27, 32, 33].

In this paper we consider the nonsymmetric algebraic Riccati equation

$$\mathcal{R}(X) = XCX - XD - AX + B = 0, \tag{1.2}$$

where A, B, C, D are real matrices of sizes $m \times m, m \times n, n \times m, n \times n$, respectively. Equation (1.2) in its general form has been studied in [8, 26, 30], for example. All the solutions of (1.2) can be found, in theory, by finding all the Jordan chains of the matrix

$$H = \begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \tag{1.3}$$

*Received by the editors July 24, 2000; accepted for publication (in revised form) by V. Mehrmann February 21, 2001; published electronically July 2, 2001. This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/simax/23-1/37568.html

†Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca).

(see [20, Theorem 7.1.2]). However, as pointed out in [22], it would be more appropriate to use Schur vectors instead of Jordan vectors.

To get a nice theory for (1.2), we need to add some conditions on the matrices A, B, C, and D, much the same as done for the symmetric equation (1.1).

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. A real square matrix A is called a Z-matrix if all its off-diagonal elements are nonpositive. It is clear that any Z-matrix A can be written as $sI - B$ with $B \geq 0$. A Z-matrix A is called an M-matrix if $s \geq \rho(B)$, where $\rho(\cdot)$ is the spectral radius. It is called a singular M-matrix if $s = \rho(B)$; it is called a nonsingular M-matrix if $s > \rho(B)$. Note that only nonsingular M-matrices defined here are called M-matrices in [14]. The slight change of definitions is made here for future convenience. The spectrum of a square matrix A will be denoted by $\sigma(A)$. The open left half-plane, the open right half-plane, the closed left half-plane, and the closed right half-plane will be denoted by $\mathbb{C}_<$, $\mathbb{C}_>$, $\mathbb{C}_\leq$, and $\mathbb{C}_\geq$, respectively.

In [14], iterative methods are studied for the numerical solution of (1.2) with the condition

$$B > 0, \quad C > 0, \quad I \otimes A + D^T \otimes I \text{ is a nonsingular } M\text{-matrix}, \tag{1.4}$$

where $\otimes$ is the Kronecker product (for basic properties of the Kronecker product, see [21], for example). It is shown there 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.

The condition (1.4) is motivated by a nonsymmetric algebraic Riccati equation arising in transport theory. That equation has the form (1.2) with $m = n$ and the matrices $A, B, C, D \in \mathbb{R}^{n\times n}$ have the following structures:

$$A = \frac{1}{\beta(1+\alpha)} W^{-1} - eq^T,\ B = ee^T,\ C = qq^T,\ D = \frac{1}{\beta(1-\alpha)} W^{-1} - qe^T. \tag{1.5}$$

In the above, $0 \leq \alpha < 1,\ 0 < \beta \leq 1$, and

$$e = (1, 1, \ldots, 1)^T, \quad q = \frac{1}{2} W^{-1} c,$$

where $W = \operatorname{diag}(w_1, w_2, \ldots, w_n),\ c = (c_1, c_2, \ldots, c_n)^T > 0$ with

$$0 < w_n < \cdots < w_2 < w_1 < 1, \quad c^T e = 1.$$

It is shown in [14] that $I \otimes A + D^T \otimes I$ is a nonsingular M-matrix for this equation. For descriptions on how the equation arises in transport theory, see [17] and references cited therein. The existence of positive solutions of this equation has been shown in [16] and [17]. However, only the minimal positive solution is physically meaningful. Numerical methods for finding the minimal solution have also been discussed in [16] and [17].

A more interesting equation of the form (1.2) has recently come to our attention. The equation arises from the Wiener–Hopf factorization of Markov chains [1, 23, 28, 29, 35]. Let Q be the Q-matrix associated with an irreducible continuous-time finite Markov chain $(X_t)_{t\geq 0}$. (A Q-matrix has nonnegative off-diagonal elements and nonpositive row sums; $\exp(tQ)$ is the transition matrix function of the Markov chain.) We need to find a quadruple (Π_1, Q_1, Π_2, Q_2) such that

$$\begin{pmatrix} A & B \\ -C & -D \end{pmatrix} \begin{pmatrix} I & \Pi_2 \\ \Pi_1 & I \end{pmatrix} = \begin{pmatrix} I & \Pi_2 \\ \Pi_1 & I \end{pmatrix} \begin{pmatrix} Q_1 & 0 \\ 0 & -Q_2 \end{pmatrix}, \tag{1.6}$$

where

$$Q = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

is a partitioning of Q with A, D being square matrices, and Q_1, Q_2 are Q-matrices. It turns out that the matrices Π_1 and Π_2 of practical interest are the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations $ZBZ+ZA+DZ+C = 0$ and $ZCZ+ZD+AZ+B = 0$, respectively (see [35]). The relation in (1.6) has been called a Wiener–Hopf factorization of the leftmost matrix in (1.6). The factorization (1.6) makes perfect sense for *any* Q-matrix, with or without probabilistic significance. Barlow, Rogers, and Williams [1] noted that they did not know how to establish the factorization without appealing to probability theory (and ultimately to martingale theory). Rogers [28] studied the factorization in more detail, again using probabilistic results and interpretations.

Note that $-Q$ is an M-matrix for any Q-matrix Q. Thus, the Riccati equations arising from the study of Markov chains are essentially special cases of the Riccati equation (1.2) with condition (1.4). However, the strict positiveness of B and C could be restrictive. We will thus relax the condition (1.4) to conditions

$$B, C \geq 0, \quad I \otimes A + D^T \otimes I \text{ is a nonsingular } M\text{-matrix}, \tag{1.7}$$

and

$$B, C \neq 0, \quad (I \otimes A + D^T \otimes I)^{-1} \mathrm{vec} B > 0, \tag{1.8}$$

where the vec operator stacks the columns of a matrix into one long vector. For some of our discussions, condition (1.7) alone will be sufficient.

The theory of M-matrices will play an important role in our discussions. The following result is well known (see [5] and [9], for example).

THEOREM 1.1. *For a Z-matrix A, the following are equivalent:*

(1) *A is a nonsingular M-matrix.*
(2) *$A^{-1} \geq 0$.*
(3) *$Av > 0$ for some vector $v > 0$.*
(4) *$\sigma(A) \subset \mathbb{C}_>$.*

The next result follows from the equivalence of statements (1) and (3) in Theorem 1.1 and can be found in [25], for example.

THEOREM 1.2. *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular 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 a nonsingular M-matrix.

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 [21], for example), it follows from the equivalence of statements (1) and (4) in Theorem 1.1 that $I \otimes A + D^T \otimes I$ is a nonsingular M-matrix when A, D are both nonsingular M-matrices.

2. Iterative methods. Newton's method and a class of basic fixed-point iterations are studied in [14] for the numerical solution of (1.2) under condition (1.4). In this section, we represent the main results in [14] under weaker conditions. These

results will be needed in later discussions. For Newton's method, we need (1.7) and (1.8). For basic fixed-point iterations, condition (1.8) is not necessary.

We first consider the application of Newton's method to (1.2). 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

$$\mathcal{R}'_X(Z) = -\big((A - XC)Z + Z(D - CX)\big). \tag{2.1}$$

The Newton method for the solution of (1.2) is

$$X_{i+1} = X_i - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(X_i), \quad i = 0, 1, \ldots, \tag{2.2}$$

given that the maps $\mathcal{R}'_{X_i}$ are all invertible. In view of (2.1), the iteration (2.2) is equivalent to

$$(A - X_iC)X_{i+1} + X_{i+1}(D - CX_i) = B - X_iCX_i, \quad i = 0, 1, \ldots. \tag{2.3}$$

THEOREM 2.1. *Consider* (1.2) *with conditions* (1.7) *and* (1.8). *If there is a positive matrix* X *such that* $\mathcal{R}(X) \le 0$, *then* (1.2) *has a positive solution* S *such that* $S \le X$ *for every positive matrix* X *for which* $\mathcal{R}(X) \le 0$. *In particular,* S *is the minimal positive solution of* (1.2). *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*

$$I \otimes (A - SC) + (D - CS)^T \otimes I \tag{2.4}$$

is an M*-matrix.*

The proof of the above theorem is exactly the same as that of [14, Theorem 2.1]. We do not have an analogous result for nonnegative solutions if the condition (1.8) is dropped. Note that, under the conditions of Theorem 2.1, any nonnegative matrix satisfying $\mathcal{R}(X) \le 0$ must be positive (see the remark following Theorem 2.3).

Concerning the convergence rate of Newton's method, we have the following result. The proof is again the same as in [14].

THEOREM 2.2. *Let the sequence* $\{X_i\}$ *be as in Theorem* 2.1. *If the matrix* (2.4) *is a nonsingular* M*-matrix, then* $\{X_i\}$ *converges to* S *quadratically. If* (2.4) *is an irreducible singular* M*-matrix, then* $\{X_i\}$ *converges to* S *either quadratically or linearly with rate* $1/2$.

We believe that quadratic convergence is impossible in the singular case, but we have no proof for this.

We now consider a class of fixed-point iterations for (1.2) under condition (1.7) only. If we write $A = A_1 - A_2,\ D = D_1 - D_2$, then (1.2) 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 a nonsingular M-matrix by Theorem 1.2. We then have a class of fixed-point iterations

$$X_{k+1} = \mathcal{L}^{-1}(X_kCX_k + X_kD_2 + A_2X_k + B), \tag{2.5}$$

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 a nonsingular M-matrix, the operator $\mathcal{L}$ is invertible and $\mathcal{L}^{-1}(X) \ge 0$ for $X \ge 0$.

THEOREM 2.3. *Consider* (1.2) *with condition* (1.7). *For the fixed-point iterations* (2.5) *and* $X_0 = 0$, *we have* $X_k \le X_{k+1}$ *for any* $k \ge 0$. *If* $\mathcal{R}(X) \le 0$ *for some nonnegative matrix* X, *then we also have* $X_k \le X$ *for any* $k \ge 0$. *Moreover,* $\{X_k\}$ *converges to the minimal nonnegative solution of* (1.2).

Proof. It is easy to prove by induction that $X_k \le X_{k+1}$ for any $k \ge 0$. When $\mathcal{R}(X) \le 0$ for some nonnegative matrix X, we can prove by induction that $X_k \le X$ for any $k \ge 0$. The limit X^* of $\{X_k\}$ is then a solution of $\mathcal{R}(X) = 0$ and must be the minimal nonnegative solution, since $X^* \le X$ for any nonnegative matrix such that $\mathcal{R}(X) \le 0$. □

Remark 2.1. If condition (1.8) is also satisfied, then the matrix X_1 produced by (2.5) with $A_1 = A$ and $D_1 = D$ is positive. This is because $\mathrm{vec}X_1 = (I \otimes A + D^T \otimes I)^{-1}\mathrm{vec}B$. Thus, for any nonnegative matrix X such that $\mathcal{R}(X) \le 0$, we have $X \ge X_1 > 0$.

The next comparison result follows easily from Theorem 2.3.

THEOREM 2.4. *Consider* (1.2) *with condition* (1.7) *and let* S *be the minimal nonnegative solution of* (1.2). *If any element of* B *or* C *decreases but remains nonnegative, 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 nonnegative solution* $\tilde{S}$. *Moreover,* $\tilde{S} \le S$.

Proof. Let the new equation be

$$\tilde{R}(X) = X\tilde{C}X - X\tilde{D} - \tilde{A}X + \tilde{B} = 0.$$

It is clear that $\tilde{R}(S) \le 0$. Since $I \otimes \tilde{A} + \tilde{D}^T \otimes I$ is still a nonsingular M-matrix by Theorem 1.2, the conclusions follow from Theorem 2.3. □

The following result is concerned with the convergence rates of the fixed-point iterations. It is a slight modification of Theorem 3.2 in [14]. The proof given there is valid without change.

THEOREM 2.5. *Consider* (1.2) *with condition* (1.7) *and let* S *be the minimal nonnegative solution of* (1.2). *For the fixed-point iterations* (2.5) *with* $X_0 = 0$, *we have*

$$\limsup_{k\to\infty} \sqrt[k]{\|X_k - S\|} \le \rho\big((I \otimes A_1 + D_1^T \otimes I)^{-1}(I \otimes (A_2 + SC) + (D_2 + CS)^T \otimes I)\big).$$

Equality holds if S *is positive.*

COROLLARY 2.6. *For* (1.2) *with condition* (1.7), *if the minimal nonnegative solution* S *of* (1.2) *is positive, then the matrix* (2.4) *is an* M*-matrix.*

Proof. Let A_1 and D_1 be the diagonal part of A and D, respectively. By Theorem 2.5, we have $\rho\big((I\otimes A_1+D_1^T\otimes I)^{-1}(I\otimes(A_2+SC)+(D_2+CS)^T\otimes I)\big) \le 1$. Therefore, for any $\epsilon > 0$, $\rho\big((I\otimes(A_1+\epsilon A_1)+(D_1+\epsilon D_1)^T\otimes I)^{-1}(I\otimes(A_2+SC)+(D_2+CS)^T\otimes I)\big) < 1$. Thus, $\epsilon(I \otimes A_1 + D_1^T \otimes I) + I \otimes (A - SC) + (D - CS)^T \otimes I$ is a nonsingular M-matrix (see [5], for example). It follows that (2.4) is an M-matrix. □

As in [14], we have the following result about the spectral radius in Theorem 2.5.

THEOREM 2.7. *Consider* (1.2) *with condition* (1.7) *and let* S *be the minimal nonnegative solution of* (1.2). *If* (2.4) *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 (2.4) *is a nonsingular* 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*

$$\rho\big((I\otimes\tilde{A}_1+\tilde{D}_1^T\otimes I)^{-1}(I\otimes(\tilde{A}_2+SC)+(\tilde{D}_2+CS)^T\otimes I)\big)$$
$$\le\rho\big((I\otimes A_1+D_1^T\otimes I)^{-1}(I\otimes(A_2+SC)+(D_2+CS)^T\otimes I)\big)<1.$$

Therefore, the convergence of these iterations is linear if (2.4) is a nonsingular M-matrix. When (2.4) is a singular M-matrix, the convergence is sublinear. Within this class of iterative methods, three iterations deserve special attention. 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 $D_1=D$. It will be called FP3.

3. A sufficient condition for the existence of nonnegative solutions. In the last section, the existence of a nonnegative solution of (1.2) is guaranteed under the assumption that there is a nonnegative matrix X such that $\mathcal{R}(X)\le 0$. The usefulness of this kind of assumption was evident in the ease we had in proving Theorem 2.4. However, if (1.2) does not have a nonnegative solution, the search for a nonnegative matrix X such that $\mathcal{R}(X)\le 0$ will necessarily be fruitless. In this section, we will give a sufficient condition of a different kind for the existence of nonnegative solutions of (1.2). This condition is suggested by the Wiener–Hopf factorization of Markov chains.

THEOREM 3.1. *If the matrix*

$$K=\begin{pmatrix} D & -C\\ -B & A\end{pmatrix} \tag{3.1}$$

is a nonsingular M-matrix, then (1.2) *has a nonnegative solution S such that $D-CS$ is a nonsingular M-matrix. If* (3.1) *is an irreducible singular M-matrix, then* (1.2) *has a nonnegative solution S such that $D-CS$ is an M-matrix.*

Proof. If (3.1) is a nonsingular M-matrix, then $T=\operatorname{diag}(D,A)$ is also a nonsingular M-matrix by Theorem 1.2. If (3.1) is an irreducible singular M-matrix, then T is a nonsingular M-matrix by the Perron–Frobenius theory (see [5] or [34]). Thus, in either case, A and D are nonsingular M-matrices. Therefore, condition (1.7) is satisfied. We take $X_0=0$ and use FP1:

$$A_1X_{i+1}+X_{i+1}D_1=X_iCX_i+X_iD_2+A_2X_i+B,\quad i=0,1,\dots. \tag{3.2}$$

By Theorem 2.3, $X_i\le X_{i+1}$ for any $i\ge 0$.

If (3.1) is a nonsingular M-matrix, we can find $v_1,v_2>0$ such that

$$D_1v_1-D_2v_1-Cv_2=u_1>0,\quad A_1v_2-A_2v_2-Bv_1=u_2>0. \tag{3.3}$$

We will show that $X_kv_1\le v_2-A_1^{-1}u_2$ for all $k\ge 0$. The inequality is true for $k=0$ since $v_2-A_1^{-1}u_2=A_1^{-1}(A_2v_2+Bv_1)\ge 0$ by the second equation in (3.3). Assume that $X_iv_1\le v_2-A_1^{-1}u_2\,(i\ge 0)$. Then, by (3.2) and (3.3),

$$\begin{aligned}A_1X_{i+1}v_1+X_{i+1}D_1v_1&=X_iCX_iv_1+X_iD_2v_1+A_2X_iv_1+Bv_1\\&\le X_iCv_2+X_iD_2v_1+A_2v_2+Bv_1\\&\le X_iD_1v_1+A_1v_2-u_2.\end{aligned}$$

Since $X_{i+1}D_1v_1\ge X_iD_1v_1$, we have $A_1X_{i+1}v_1\le A_1v_2-u_2$. Therefore, $X_{i+1}v_1\le v_2-A_1^{-1}u_2$. Thus, we have proved by induction that $X_kv_1\le v_2-A_1^{-1}u_2$ for all $k\ge 0$.

Now, the sequence $\{X_i\}$ is monotonically increasing and bounded above, and hence has a limit. Let $S = \lim_{i\to\infty} X_i$. It is clear that S is a nonnegative solution of (1.2) and $Sv_1 \le v_2 - A_1^{-1}u_2 < v_2$. Thus, $(D - CS)v_1 \ge Dv_1 - Cv_2 = u_1 > 0$. Therefore, $D - CS$ is a nonsingular M-matrix by Theorem 1.1. If (3.1) is an irreducible singular M-matrix, there are $v_1, v_2 > 0$ (by the Perron–Frobenius theory) such that

$$D_1v_1 - D_2v_1 - Cv_2 = 0, \quad A_1v_2 - A_2v_2 - Bv_1 = 0.$$

We can prove as before that the sequence $\{X_i\}$ produced by FP1 is such that $X_iv_1 \le v_2$ for all $i \ge 0$. The limit S of the sequence is a nonnegative solution of (1.2) with $Sv_1 \le v_2$. Therefore, $(D - CS)v_1 \ge Dv_1 - Cv_2 = 0$. Thus, $D - CS + \epsilon I$ is a nonsingular M-matrix for any $\epsilon > 0$. So, $D - CS$ is an M-matrix. □

Remark 3.1. We know from Theorem 2.3 that the matrix S in the proof is the minimal nonnegative solution of (1.2). Note also that we have obtained in the proof some additional information about the minimal solution. It will be seen later (from Theorem 4.2) that the minimal solution S is the only solution X that makes $D - CX$ an M-matrix when the matrix (3.1) is a nonsingular M-matrix. If the matrix (3.1) is an irreducible singular M-matrix, then (1.2) may have more than one nonnegative solution X such that $D - CX$ is an M-matrix. For example, for $A = B = 1$ and $C = D = 2$, the scalar equation (1.2) has two positive solutions $X = 1$ and $X = 1/2$. The first makes $D - CX$ a singular M-matrix. The second makes $D - CX$ a nonsingular M-matrix.

We have seen in the proof of Theorem 3.1 that condition (1.7) is satisfied when the matrix (3.1) is a nonsingular M-matrix or an irreducible singular M-matrix. It is clear that (1.8) is not necessarily true when (3.1) is a nonsingular M-matrix. If (3.1) is an irreducible singular M-matrix, we have $B, C \ne 0$. However,

$$(I \otimes A + D^T \otimes I)^{-1}\mathrm{vec}B > 0 \tag{3.4}$$

is not necessarily true.

Assume that (3.1) is an irreducible M-matrix. If (3.4) is true, then the minimal nonnegative solution S of (1.2) must be positive. However, a more practical method to verify the positivity of S is to apply FP1 with $X_0 = 0$ to (1.2).

Example 3.1. For (1.2) with

$$A = C = D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

the matrix (3.1) is an irreducible singular M-matrix, but (3.4) is not true. However, the minimal nonnegative solution S is still positive. In fact, if we apply FP1 with $X_0 = 0$ to (1.2), we get $X_2 > 0$. Thus, $S \ge X_2 > 0$.

We also have the following sufficient condition for the positivity of S.

PROPOSITION 3.2. *If* (3.1) *is an* M*-matrix such that* $B, C \ne 0$*, and* A, D *are irreducible, then* (3.1) *is irreducible,* (3.4) *is true, and* $S > 0$*.*

Proof. The matrix (3.1) is irreducible by a graph argument (see Theorem 2.2.7 of [5]). As shown in the proof of Theorem 3.1, the matrices A and D are now irreducible nonsingular M-matrices. Thus, $I \otimes A + D^T \otimes I$ is an irreducible nonsingular M-matrix (the irreducibility is shown by a graph argument). Therefore, by Theorem 6.2.7 of [5], $(I \otimes A + D^T \otimes I)^{-1} > 0$. Thus, (3.4) is true and $S > 0$. □

More can be said about the matrix $D - CS$ when the minimal nonnegative solution S of (1.2) is positive.

THEOREM 3.3. *If* (3.1) *is an irreducible* M*-matrix and the minimal nonnegative solution* S *of* (1.2) *is positive, then* $D - CS$ *is an irreducible* M*-matrix* (*we use the convention that a* 1×1 *zero matrix is irreducible*).

Proof. We need only to prove that $D - CS$ is irreducible for $n \geq 2$. Write $D = (d_{ij})$. Let $V_1 = \{i \mid 1 \leq i \leq n,$ the ith row of C is zero$\}$ and $V_2 = \{i \mid 1 \leq i \leq n,$ the ith row of C is not zero$\}$. Since (3.1) is irreducible, its graph is strongly connected (see Theorem 2.2.7 of [5]). Therefore, for any $i \in V_1$ we can find $i_1, \ldots, i_{k-1} \in V_1$ (void if $k = 1$) and $i_k \in V_2$ such that $d_{ii_1}, d_{i_1 i_2}, \ldots, d_{i_{k-1} i_k}$ are nonzero. Now, take any $i : 1 \leq i \leq n$. If $i \in V_2$, then the off-diagonal elements of $D - CS$ in the ith row are negative since S is positive; the diagonal element of $D - CS$ in the ith row must be positive since it is shown in the proof of Theorem 3.1 that $(D - CS)v_1 \geq 0$ for some $v_1 > 0$. If $i \in V_1$, then the ith row of $D - CS$ is the same as the ith row of D. It follows readily that the graph of $D - CS$ is strongly connected. So, $D - CS$ is irreducible. □

Equation (1.2) with no prescribed sign structure for the matrices $A, B, C,$ and D has been considered in [8] and [30]. It is shown that the solution of (1.2) with minimal Frobenius norm can be found by FP3 and Newton's method starting with $X_0 = 0$, if $\kappa < 1/4$ for FP3 and $\kappa < 1/12$ for Newton's method, where $\kappa = \|B\|_F \|C\|_F / s^2$ and s is the smallest singular value of $I \otimes A + D^T \otimes I$. If the matrix (3.1) is a singular M-matrix with no zero elements, for example, then the minimal positive solution can be found by FP3 and Newton's method with $X_0 = 0$. It is interesting to see how often the condition $\kappa < 1/4$ is satisfied when (3.1) is a singular M-matrix with no zero elements. We use MATLAB to obtain a 4×4 positive matrix R using rand(4,4), so $W = \mathrm{diag}(Re) - R$ is a singular M-matrix with no zero elements. We let the matrix W be in the form (3.1), so the 2×2 matrices A, B, C, D are determined. We find that $\kappa < 1/4$ is satisfied 198 times for 10000 random matrices R ($\kappa < 1/5$ is satisfied 35 times). When we use rand(6,6) to get 3×3 matrices A, B, C, D in the same way, we find that $\kappa < 1/4$ is satisfied 2 times for 10000 random matrices.

It is interesting to note that (1.2) from transport theory also satisfies the conditions in Theorem 3.1.

PROPOSITION 3.4. *Let the matrices* A, B, C, D *be defined by* (1.5). *Then the matrix* K *given by* (3.1) *is irreducible. The matrix* K *is a nonsingular* M*-matrix for* $0 < \beta < 1$ *and is a singular* M*-matrix for* $\beta = 1$.

Proof. By definition,

$$K = \begin{pmatrix} \frac{1}{\beta(1-\alpha)} W^{-1} - qe^T & -qq^T \\ -ee^T & \frac{1}{\beta(1+\alpha)} W^{-1} - eq^T \end{pmatrix}.$$

It is clear that K is irreducible. Since K is a singular (nonsingular) M-matrix if and only if

$$\begin{pmatrix} (1-\alpha)W & 0 \\ 0 & (1+\alpha)W \end{pmatrix} K = \begin{pmatrix} \frac{1}{\beta} I - (1-\alpha) W q e^T & -(1-\alpha) W q q^T \\ -(1+\alpha) W e e^T & \frac{1}{\beta} I - (1+\alpha) W e q^T \end{pmatrix}$$

is a singular (nonsingular) M-matrix, we need only to find a positive vector v such that $Qv = v$ for the positive matrix

$$Q = \begin{pmatrix} (1-\alpha) W q e^T & (1-\alpha) W q q^T \\ (1+\alpha) W e e^T & (1+\alpha) W e q^T \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(1-\alpha) c e^T & \frac{1}{4}(1-\alpha) c c^T W^{-1} \\ (1+\alpha) W e e^T & \frac{1}{2}(1+\alpha) W e c^T W^{-1} \end{pmatrix},$$

where we have used $q = \frac{1}{2}W^{-1}c$. Now, since $e^Tc = c^Te = 1$, direct computation shows that $Qv = v$ for

$$v = \begin{pmatrix} (1-\alpha)c \\ 2(1+\alpha)We \end{pmatrix} > 0.$$

This completes the proof. □

Therefore, with Remark 2.1 in mind, the existence of positive solutions of (1.2) with A, B, C, D given by (1.5) is established as a special case of Theorem 3.1. The existence in this special case was proved in [16] using the degree theory and was proved in [17] using the secular equation and other tools.

4. Wiener–Hopf factorization for *M*-matrices. The Wiener–Hopf factorization for Q-matrices associated with finite Markov chains has been studied in [1, 23, 28, 29, 35]. The factorization was obtained by using probabilistic results and interpretations. In this section, we will establish Wiener–Hopf factorization for M-matrices. Our results include Wiener–Hopf factorization for Q-matrices as a special case. The proof will be purely algebraic.

THEOREM 4.1. *If the matrix* (3.1) *is a nonsingular M-matrix or an irreducible singular M-matrix, then there exist nonnegative matrices S_1 and S_2 such that*

$$\begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} I & S_2 \\ S_1 & I \end{pmatrix} = \begin{pmatrix} I & S_2 \\ S_1 & I \end{pmatrix} \begin{pmatrix} G_1 & 0 \\ 0 & -G_2 \end{pmatrix}, \tag{4.1}$$

where G_1 and G_2 are M-matrices.

Proof. By Theorem 3.1, (1.2) has a nonnegative solution S_1 such that $D - CS_1$ is an M-matrix. By taking $G_1 = D - CS_1$, we get

$$\begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} I \\ S_1 \end{pmatrix} = \begin{pmatrix} I \\ S_1 \end{pmatrix} G_1. \tag{4.2}$$

Since

$$\begin{pmatrix} A & -B \\ -C & D \end{pmatrix}$$

is also a nonsingular M-matrix or an irreducible singular M-matrix, Theorem 3.1 implies that the equation

$$XBX - XA - DX + C = 0 \tag{4.3}$$

has a nonnegative solution S_2 such that $A - BS_2$ is an M-matrix. Letting $G_2 = A - BS_2$, we have

$$\begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} S_2 \\ I \end{pmatrix} = \begin{pmatrix} S_2 \\ I \end{pmatrix} (-G_2). \tag{4.4}$$

The factorization (4.1) is obtained by combining (4.2) and (4.4). □

We can make stronger statements when the matrix (3.1) is a nonsingular M-matrix.

THEOREM 4.2. *If the matrix* (3.1) *is a nonsingular M-matrix, then the only matrices S_1 and S_2 satisfying* (4.1) *with G_1 and G_2 being M-matrices are the minimal*

nonnegative solution of (1.2) *and the minimal nonnegative solution of* (4.3)*, respectively. In this case,* G_1 *and* G_2 *are nonsingular* M*-matrices and the matrix*

$$\begin{pmatrix} I & S_2 \\ S_1 & I \end{pmatrix} \tag{4.5}$$

is nonsingular.

Proof. Let S_1 and S_2 be the minimal nonnegative solutions of (1.2) and (4.3), respectively. Let $G_1 = D - CS_1$ and $G_2 = A - BS_2$. Then (4.1) holds and G_1, G_2 are nonsingular M-matrices (see Theorem 3.1). Let $v_1, v_2 > 0$ be as in the proof of Theorem 3.1. Then, $S_1 v_1 < v_2$ and $S_2 v_2 < v_1$. Since

$$\begin{pmatrix} I & -S_2 \\ -S_1 & I \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} > 0,$$

the matrix (4.5) is a generalized strictly diagonally dominant matrix and hence nonsingular. Thus, (4.1) gives a similarity transformation and, as a result, the matrix (1.3) has n eigenvalues in $\mathbb{C}_>$ and m eigenvalues in $\mathbb{C}_<$. Now, if $\tilde{S}_1 \in \mathbb{R}^{m\times n}$ and $\tilde{S}_2 \in \mathbb{R}^{n\times m}$ satisfy (4.1) with $\tilde{G}_1$ and $\tilde{G}_2$ being M-matrices, then

$$\begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} I \\ \tilde{S}_1 \end{pmatrix} = \begin{pmatrix} I \\ \tilde{S}_1 \end{pmatrix} \tilde{G}_1.$$

Therefore, the eigenvalues of $\tilde{G}_1$ are precisely the n eigenvalues of (1.3) in $\mathbb{C}_>$. Since the column spaces of $(I \ \tilde{S}_1^T)^T$ and $(I \ S_1^T)^T$ are the same invariant subspace associated with these eigenvalues, we conclude that $\tilde{S}_1 = S_1$. Similarly, $\tilde{S}_2 = S_2$. ∎

From the above theorem and its proof, it is already clear that we can find the minimal solution using an appropriate invariant subspace (details will be provided in the next section).

The rest of this section is devoted to the case where (3.1) is an irreducible singular M-matrix. The minimal nonnegative solutions of (1.2) and (4.3) will be denoted by S_1 and S_2, respectively.

Let v_1, v_2, u_1, u_2 be positive vectors such that

$$\begin{pmatrix} D & -C \\ -B & A \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0, \quad (u_1^T \ u_2^T) \begin{pmatrix} D & -C \\ -B & A \end{pmatrix} = 0. \tag{4.6}$$

Multiplying (4.1) by $\begin{pmatrix} u_1^T & -u_2^T \end{pmatrix}$ from the left gives

$$(u_1^T - u_2^T S_1)G_1 = 0, \quad (u_1^T S_2 - u_2^T)G_2 = 0.$$

If G_1 is nonsingular, then $u_1^T - u_2^T S_1 = 0$. Moreover, we see from the proof of Theorem 3.1 that $S_1 v_1 \le v_2$ and $S_1 v_1 \ne v_2$. (If $S_1 v_1 = v_2$, we would have $G_1 v_1 = (D - CS_1)v_1 = Dv_1 - Cv_2 = 0$, which is contradictory to the nonsingularity of G_1.) So $u_1^T v_1 < u_2^T v_2$. Similarly, $u_1^T v_1 > u_2^T v_2$ when G_2 is nonsingular. Therefore, the following result is true.

LEMMA 4.3. *When* (3.1) *is an irreducible singular* M*-matrix,* G_1 *is singular if* $u_1^T v_1 > u_2^T v_2$*;* G_2 *is singular if* $u_1^T v_1 < u_2^T v_2$*; both* G_1 *and* G_2 *are singular if* $u_1^T v_1 = u_2^T v_2$.

Further discussions will be dependent on the positivity of S_1 and S_2.

LEMMA 4.4. *Assume that* (3.1) *is an irreducible singular* M*-matrix and* $S_1, S_2 > 0$*. Then the matrix* (1.3) *has* $n-1$ *eigenvalues in* $\mathbb{C}_>$*,* $m-1$ *eigenvalues in* $\mathbb{C}_<$*,*

one zero eigenvalue, and one more eigenvalue which either is zero or has nonzero real part.

Proof. By Theorem 3.3, G_1 and G_2 are irreducible M-matrices. Therefore, $G_1(G_2)$ has $n(m)$ eigenvalues in $\mathbb{C}_>$ when it is nonsingular; $G_1(G_2)$ has a zero eigenvalue and $n-1(m-1)$ eigenvalues in $\mathbb{C}_>$ when it is singular. By (4.1), the eigenvalues of G_1(resp., $-G_2$) are precisely the eigenvalues of the matrix (1.3) restricted to the column space of $(I \;\; S_1^T)^T$ (resp., $(S_2^T \;\; I)^T$). The result follows immediately. ∎

LEMMA 4.5. *Under the assumptions of Lemma* 4.4*, zero is a double eigenvalue of* (1.3) *if and only if* $u_1^T v_1 = u_2^T v_2$.

Proof. If zero is a double eigenvalue of (1.3), then the Jordan canonical form for (1.3) is

$$P^{-1} \begin{pmatrix} D & -C \\ B & -A \end{pmatrix} P = \begin{pmatrix} J_1 & 0 \\ 0 & J_2 \end{pmatrix},$$

where $J_1 = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ and J_2 consists of Jordan blocks associated with nonzero eigenvalues. (Note that the null space of (1.3) is one-dimensional since (3.1) is an irreducible M-matrix.) By (4.6), we get

$$(u_1^T \;\; -u_2^T)P = k_1 e_2^T, \quad P^{-1} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = k_2 e_1, \tag{4.7}$$

where e_1, e_2 are the first two standard unit vectors and k_1, k_2 are nonzero constants. Multiplying the two equations in (4.7) gives $u_1^T v_1 = u_2^T v_2$. If zero is a simple eigenvalue of (1.3), then we have $J_1 = (0)$ instead and we have (4.7) with e_2 replaced by e_1. Thus, $u_1^T v_1 \neq u_2^T v_2$. ∎

We will also need the following general result, which can be found in [26], for example.

LEMMA 4.6. *If* X *is any solution of* (1.2)*, then*

$$\begin{pmatrix} I & 0 \\ X & I \end{pmatrix}^{-1} \begin{pmatrix} D & -C \\ B & -A \end{pmatrix} \begin{pmatrix} I & 0 \\ X & I \end{pmatrix} = \begin{pmatrix} D - CX & -C \\ 0 & -(A - XC) \end{pmatrix}.$$

Thus, the eigenvalues of $D - CX$ *are eigenvalues of* (1.3) *and the eigenvalues of* $A - XC$ *are the negative of the remaining eigenvalues of* (1.3).

The next result determines the signs of the real parts for all eigenvalues of the matrix (1.3), and it also paves the way for finding S_1 and S_2 using subspace methods.

THEOREM 4.7. *Assume that the matrix* (3.1) *is an irreducible singular* M*-matrix and* $S_1, S_2 > 0$. *Let the vectors* u_1, u_2, v_1, v_2 *be as in* (4.6)*. Then we have the following.*

1. *If* $u_1^T v_1 = u_2^T v_2$*, then* (1.3) *has* $n-1$ *eigenvalues in* $\mathbb{C}_>$*,* $m-1$ *eigenvalues in* $\mathbb{C}_<$*, and two zero eigenvalues. Moreover,* G_1 *and* G_2 *are singular* M*-matrices.*
2. *If* $u_1^T v_1 > u_2^T v_2$*, then* (1.3) *has* $n-1$ *eigenvalues in* $\mathbb{C}_>$*,* m *eigenvalues in* $\mathbb{C}_<$*, and one zero eigenvalue. Moreover,* G_1 *is a singular* M*-matrix and* G_2 *is a nonsingular* M*-matrix.*
3. *If* $u_1^T v_1 < u_2^T v_2$*, then* (1.3) *has* n *eigenvalues in* $\mathbb{C}_>$*,* $m-1$ *eigenvalues in* $\mathbb{C}_<$*, and one zero eigenvalue. Moreover,* G_1 *is a nonsingular* M*-matrix and* G_2 *is a singular* M*-matrix.*

Proof. Assertion (1) follows from Lemmas 4.5, 4.4, and 4.3. We will prove assertion (2) only since the proof of assertion (3) is very similar. When $u_1^T v_1 > u_2^T v_2$, $G_1 = D - CS_1$ is singular by Lemma 4.3. The last-mentioned eigenvalue in Lemma 4.4 cannot be zero by Lemma 4.5, and we need to show that it is in $\mathbb{C}_<$. If this

eigenvalue were in $\mathbb{C}_>$, the matrix $A - S_1C$ would have $m-1$ eigenvalues in $\mathbb{C}_>$ and one eigenvalue in $\mathbb{C}_<$, in view of Lemma 4.6. Since the eigenvalues of $G = I \otimes (A - S_1C) + (D - CS_1)^T \otimes I$ are the sums of eigenvalues of $A - S_1C$ and eigenvalues of $D - CS_1$, the matrix G would then have an eigenvalue in $\mathbb{C}_<$. This is a contradiction since G is an M-matrix by Corollary 2.6. A similar argument then shows that the eigenvalues of $G_2 = A - BS_2$ must be the negative of the m eigenvalues of (1.3) in $\mathbb{C}_<$. Therefore, G_2 is a nonsingular M-matrix by Theorem 1.1. □

Remark 4.1. Case (1) of Theorem 4.7 poses a great challenge to basic fixed-point iterations. Since the matrix (2.4) is a singular M-matrix in this case, the convergence of the fixed-point iterations for (1.2) is sublinear (see Theorems 2.5 and 2.7). The convergence of Newton's method for (1.2) is typically linear with rate 1/2 in this case if condition (3.4) is also satisfied, but the performance of Newton's method can be improved by using a double Newton step (see discussions in [14]).

When the matrix (3.1) is an irreducible singular M-matrix, we know from the proof of Theorem 3.1 that $S_1v_1 \le v_2$ and $S_2v_2 \le v_1$. With the additional assumption that $S_1, S_2 > 0$, we can say something more about S_1 and S_2.

THEOREM 4.8. *Under the conditions of Theorem* 4.7*, we have the following:*

(1) *If $u_1^Tv_1 = u_2^Tv_2$, then $CS_1v_1 = Cv_2$ and $BS_2v_2 = Bv_1$. Consequently, $S_1v_1 = v_2$ and $S_2v_2 = v_1$ if C and B have no zero columns.*
(2) *If $u_1^Tv_1 > u_2^Tv_2$, then $S_2v_2 \ne v_1$ and $CS_1v_1 = Cv_2$. Consequently, $S_1v_1 = v_2$ if C has no zero columns.*
(3) *If $u_1^Tv_1 < u_2^Tv_2$, then $S_1v_1 \ne v_2$ and $BS_2v_2 = Bv_1$. Consequently, $S_2v_2 = v_1$ if B has no zero columns.*

Moreover, the matrix (4.5) *is singular if and only if $S_1v_1 = v_2$ and $S_2v_2 = v_1$.*

Proof. We will prove assertion (3). The proof of assertions (1) and (2) is similar. If $u_1^Tv_1 < u_2^Tv_2$, then G_1 is a nonsingular M-matrix and G_2 is a singular M-matrix by Theorem 4.7. That $S_1v_1 \ne v_2$ has been proved in the discussions leading to Lemma 4.3. Note that G_2 is irreducible and $G_2v_2 = (A - BS_2)v_2 \ge Av_2 - Bv_1 = 0$. If $BS_2v_2 \ne Bv_1$, then G_2v_2 would be nonnegative and nonzero.

Therefore, G_2 would be a nonsingular M-matrix by Theorem 6.2.7 of [5]. The contradiction shows that $BS_2v_2 = Bv_1$. Thus, $B(v_1 - S_2v_2) = 0$. It follows that $v_1 - S_2v_2 = 0$ if B has no zero columns. The proof of assertion (3) is completed.

If $S_1v_1 = v_2$ and $S_2v_2 = v_1$, then

$$\begin{pmatrix} I & S_2 \\ S_1 & I \end{pmatrix} \begin{pmatrix} v_1 \\ -v_2 \end{pmatrix} = 0.$$

Thus, the matrix (4.5) is singular. If $S_1v_1 = v_2$ and $S_2v_2 = v_1$ are not both true, then

$$\begin{pmatrix} I & -S_2 \\ -S_1 & I \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{4.8}$$

is nonnegative and nonzero. Since S_1 and S_2 are positive, the matrix on the left side of (4.8) is irreducible. Therefore, it is a nonsingular M-matrix by Theorem 6.2.7 of [5] and hence the matrix (4.5) is nonsingular. □

For (1.2) from transport theory, the matrix (3.1) is an irreducible singular M-matrix if $\beta = 1$ (see Proposition 3.4). The next result shows that, for this special equation, only cases (1) and (3) are possible in Theorems 4.7 and 4.8.

PROPOSITION 4.9. *For* (1.2) *with A, B, C, D given by* (1.5) *with $\beta = 1$, we have $u_1^Tv_1 = u_2^Tv_2$ for $\alpha = 0$ and $u_1^Tv_1 < u_2^Tv_2$ for $0 < \alpha < 1$.*

Proof. By the proof of Proposition 3.4, we can take $v_1 = (1-\alpha)c$ and $v_2 = 2(1+\alpha)We$. Similarly, we can take $u_1 = 2(1-\alpha)We$ and $u_2 = (1+\alpha)c$. The conclusions follow immediately. ∎

5. The Schur method. In this section, we will explain how to use the Schur method to find the minimal nonnegative solution of (1.2).

THEOREM 5.1. *Assume that* (3.1) *is a nonsingular M-matrix or an irreducible singular M-matrix such that the minimal nonnegative solutions of* (1.2) *and* (4.3) *are positive. Let H be the matrix given by* (1.3). *Let U be an orthogonal matrix such that*

$$U^T H U = F$$

is a real Schur form of H, where the 1×1 *or* 2×2 *diagonal blocks of F are arranged in the order for which the real parts of the corresponding eigenvalues are nonincreasing. If U is partitioned as*

$$\begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix},$$

where $U_{11} \in \mathbb{R}^{n\times n}$, *then* U_{11} *is nonsingular and* $U_{21}U_{11}^{-1}$ *is the minimal nonnegative solution of* (1.2).

Proof. From the discussions in sections 3 and 4, we know that the minimal nonnegative solution S exists and the n-dimensional column space of $(I \;\; S^T)^T$ is either the n-dimensional invariant subspace of H corresponding to the eigenvalues in $\mathbb{C}_>$ or, when the largest invariant subspace $\mathcal{V}$ of H corresponding to the eigenvalues in $\mathbb{C}_>$ is only $(n-1)$-dimensional, the direct sum of $\mathcal{V}$ and the one-dimensional eigenspace of H corresponding to the zero eigenvalue. From $HU = UF$ and the specified ordering of the diagonal blocks of F, we can see that the column space of $(U_{11}^T \;\; U_{21}^T)^T$ is the same n-dimensional invariant subspace. (No difficulties will arise when H has a double zero eigenvalue, since there is only one eigenvector (up to a factor) associated with the zero eigenvalue.) So,

$$\begin{pmatrix} I \\ S \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix} W$$

for some nonsingular $W \in \mathbb{R}^{n\times n}$. Thus, U_{11} is nonsingular and $S = U_{21}U_{11}^{-1}$. ∎

Remark 5.1. A Wiener–Hopf factorization for (3.1) can also be obtained by solving the dual equation (4.3).

We now consider (1.2) with conditions (1.7) and (1.8). In this case, any nonnegative solution of (1.2) must be positive (see Remark 2.1).

THEOREM 5.2. *Consider* (1.2) *with conditions* (1.7) *and* (1.8). *Let H be the matrix given by* (1.3). *Let U be an orthogonal matrix such that*

$$U^T H U = F$$

is a real Schur form of H, where the 1×1 *or* 2×2 *diagonal blocks of F are arranged in the order for which the real parts of the corresponding* $n+m$ *eigenvalues are nonincreasing.*

(1) *Assume that* λ_n *and* λ_{n+1} *are a conjugate pair corresponding to a* 2×2 *diagonal block,* $\mathrm{Re}(\lambda_{n-1}) > \mathrm{Re}(\lambda_n)$ (*if* $n > 1$), *and* $\mathrm{Re}(\lambda_{n+1}) > \mathrm{Re}(\lambda_{n+2})$ (*if* $m > 1$). *Then* (1.2) *has no positive solutions.*

(2) *Assume that* $\mathrm{Re}(\lambda_n) > \mathrm{Re}(\lambda_{n+1})$ *and* U *is partitioned as*

$$\begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix},$$

where $U_{11} \in \mathbb{R}^{n\times n}$. *If* U_{11} *is nonsingular and* $S = U_{21}U_{11}^{-1}$ *is positive, then* S *is the minimal positive solution of* (1.2). *Otherwise,* (1.2) *has no positive solutions.*

(3) *Assume that* $\lambda_n = \lambda_{n+1}$ *are real,* $\mathrm{Re}(\lambda_{n-1}) > \lambda_n$ (*if* $n > 1$), *and* $\lambda_{n+1} > \mathrm{Re}(\lambda_{n+2})$ (*if* $m > 1$). *Assume further that there is only one eigenvector* (*up to a factor*) *associated with* $\lambda_n = \lambda_{n+1}$ *and let* U *be partitioned as in part* (2). *If* U_{11} *is nonsingular and* $S = U_{21}U_{11}^{-1}$ *is positive, then* S *is the minimal positive solution of* (1.2). *Otherwise,* (1.2) *has no positive solutions.*

Proof. If (1.2) has a positive solution, then it has a minimal positive solution S by Theorem 2.1. Since $I \otimes (A - SC) + (D - CS)^T \otimes I$ is an M-matrix by Theorem 2.1, the real part of each eigenvalue of $D - CS$ must be greater than or equal to the negative of the real part of each eigenvalue of $A - SC$. In other words, in view of Lemma 4.6, the real part of each eigenvalue of $D - CS$ must be greater than or equal to the real part of each of the remaining m eigenvalues of H. Under the assumptions of part (1), the eigenvalues of the real matrix $D - CS$ must be $\lambda_1, \ldots, \lambda_{n-1}$ and one of the eigenvalues λ_n and λ_{n+1}. This is impossible since λ_n and λ_{n+1} are a conjugate pair with nonzero imaginary parts. Part (1) is thus proved. Under the assumptions of part (2), if (1.2) has a positive solution, then the column space of $(I \;\; S^T)^T$ for the minimal positive solution S must be the n-dimensional invariant subspace of H corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$. The proof can thus be completed as in the proof of Theorem 5.1. Under the assumptions of part (3), if (1.2) has a positive solution, then the column space of $(I \;\; S^T)^T$ for the minimal positive solution S must be the direct sum of the $(n-1)$-dimensional invariant subspace of H corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ and the one-dimensional eigenspace of H corresponding to $\lambda_n = \lambda_{n+1}$. The proof can again be completed as in the proof of Theorem 5.1. □

Remark 5.2. If the minimal positive solution found by the Schur method in Theorem 5.2, part (2) is not accurate enough, we can use Newton's method as a correction method. Local quadratic convergence of Newton's method is guaranteed since the Fréchet derivative at the solution is nonsingular in this case.

Remark 5.3. In Theorem 5.2, part (3), the additional assumption that there is only one eigenvector associated with $\lambda_n = \lambda_{n+1}$ is essential. Without this assumption, no definitive information can be obtained about positive solutions of (1.2) from the real Schur form. Newton's method can find the minimal positive solution of (1.2) if it has a positive solution, with or without the additional assumption. However, we cannot expect Newton's method to have quadratic convergence since the Fréchet derivative at the minimal solution is singular in this case.

As we can see from Theorems 5.1 and 5.2, the real Schur form with the prescribed ordering of the diagonal blocks is essential for finding the minimal nonnegative solution using the Schur method. This real Schur form can be obtained by using orthogonal transformations to reduce H to upper Hessenberg form and then using a slight modification of Stewart's algorithm HQR3 [31]. In Stewart's HQR3, the 1×1 or 2×2 diagonal blocks of the real Schur form are arranged in the order for which the moduli (not the real parts) of the corresponding eigenvalues are nonincreasing.

In Theorems 5.1 and 5.2, the minimal nonnegative solution S is found by solving $SU_{11} = U_{21}$. The accuracy of S is thus dependent on $\kappa(U_{11})$, the condition number

of the matrix U_{11}.

PROPOSITION 5.3. *Let $S = U_{21}U_{11}^{-1}$ be the minimal nonnegative solution of* (1.2)*, where U_{11} and U_{21} are as in Theorems* 5.1 *and* 5.2*. Then*

$$\kappa_2(U_{11}) \le 1 + \|S\|_2^2.$$

Proof. The proof is omitted here since it is very similar to that of corresponding results in [15] and [18]. □

6. Comparison of solution methods. For (1.2) with conditions (1.7) and (1.8), the minimal positive solution can be found by FP1, FP2, FP3, Newton's method, or the Schur method whenever (1.2) has a positive solution. In this section, we will compare these methods on a few test examples.

For the Newton iteration (2.2), 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 [2] and [11]. If we use the Bartels–Stewart algorithm [2] to solve the Sylvester equation, the computational work for each Newton iteration is about $62n^3$ flops when $m = n$. 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. The Schur method needs roughly $200n^3$ flops to get an approximate solution.

Example 6.1. We generate (and save) a random 100×100 matrix R with no zero elements using rand(100,100) in MATLAB. Let $W = \mathrm{diag}(Re) - R$. So W is a singular M-matrix with no zero elements. We introduce a real parameter α and let

$$\alpha I + W = \begin{pmatrix} D & -C \\ -B & A \end{pmatrix},$$

where the matrices A, B, C, D are all 50×50. The existence of a positive solution of (1.2) is guaranteed for $\alpha \ge 0$. In Tables 6.1–6.3, 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 $X_0 = 0$. The initial residual error is $\|\mathcal{R}(X_0)\|_\infty = \|B\|_\infty = 0.2978 \times 10^2$. As predicted by Theorem 2.7, FP2 has faster convergence than FP1, while FP3 has faster convergence than FP2. With the required computational work per iteration in mind, we find that, for this example, FP2 is the best among the three basic fixed-point iterations. When $\alpha = 10$, the fixed-point iterations are quite good. However, Newton's method is much better for $\alpha = 0$. As shown in [14], we can also use Newton's method after any number of fixed-point iterations and still have the monotone convergence. We now apply the Schur method to find the minimal solution. The method turns out to be very successful. The residual norm for the approximate solution obtained from the Schur method (SM) is listed in Table 6.4, along with the residual norm for the approximate solution obtained by Newton's method after $12, 6, 5$ iterations for $\alpha = 0, 1, 10$, respectively. The accuracy achieved by the Schur method is very impressive, although not as high as that achieved by Newton's method. The good performance of the Schur method is partly due to the small condition number of the matrix U_{11}. For $\alpha = 0$, for example, we find that $\kappa_2(U_{11}) = 1.4114$ and $\|S\|_2 = 0.9960$. A rough estimate can actually be obtained beforehand for any $\alpha \ge 0$. Since $Se \le e$ by the proof of Theorem 3.1, we have $\|S\|_\infty \le 1$. So, by Proposition 5.3, $\kappa_2(U_{11}) \le 1 + (\sqrt{50}\|S\|_\infty)^2 \le 51$. When $\alpha = 0$, the Schur method is much better than the basic fixed-point iterations. It is also considerably cheaper than Newton's method, although

TABLE 6.1
Iteration counts for Example 6.1, $\alpha = 0$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
NM	6	9	10	11	12
FP1	196	1515	4003	6564	9125
FP2	154	1173	3050	4977	6904
FP3	96	758	2017	3313	4609

TABLE 6.2
Iteration counts for Example 6.1, $\alpha = 1$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
NM	4	5	5	6	6
FP1	40	71	101	131	161
FP2	30	52	75	97	119
FP3	19	33	47	62	76

TABLE 6.3
Iteration counts for Example 6.1, $\alpha = 10$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
NM	3	4	4	4	5
FP1	14	23	32	41	49
FP2	11	17	23	29	35
FP3	6	10	14	18	22

TABLE 6.4
Residual errors for Example 6.1.

α	0	1	10
NM	0.1999×10^{-13}	0.1570×10^{-13}	0.1149×10^{-13}
SM	0.6419×10^{-12}	0.5715×10^{-12}	0.6984×10^{-12}

Newton's method produces a more accurate approximation. When $\alpha = -10^{-4}$, the equation also has a positive solution by Theorem 5.2, part (2). The residual norm for the approximate solution obtained from the Schur method is 0.7463×10^{-12}, while the residual norm for the approximate solution obtained by Newton's method after 13 iterations is 0.1955×10^{-13}. By Theorem 2.4, (1.2) has a positive solution for all $\alpha \geq -10^{-4}$. When $\alpha = -10^{-3}$, the equation does not have a positive solution. In this case, Newton's method exhibits no convergence and the Schur method produces a 2×2 block in the middle of the real Schur form (see Theorem 5.2, part (1)). When $\alpha = 0$, the matrix (1.3) has 50 eigenvalues in $\mathbb{C}_>$, 49 eigenvalues in $\mathbb{C}_<$, and one zero eigenvalue. The eigenvalue with the smallest positive real part is the real eigenvalue $\lambda_{50} = 0.1790$. Thus, for all $\alpha \geq 0$, the convergence of Newton's method is quadratic and the convergence of basic fixed-point iterations is linear.

Example 6.1 is not particularly tough for the basic fixed-point iterations since $\lambda_{50} = 0.1790$ is not too close to zero. The next example *is*.

Example 6.2. Let

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}$$

be a doubly stochastic matrix (i.e., $R \geq 0, Re = e, R^T e = e$), where $R_{11}, R_{22} \in \mathbb{R}^{m\times m}$ are irreducible and $R_{21}, R_{12} \neq 0$. Let $W = a(I - R)$, where a is a given positive number. So W is a singular M-matrix satisfying the assumptions in Proposition 3.2 and the situation in Theorem 4.7, part (1) happens. Let

$$W = \begin{pmatrix} D & -C \\ -B & A \end{pmatrix},$$

where $A, B, C, D \in \mathbb{R}^{m\times m}$. We will find the minimal positive solution of (1.2). As noted in Remark 4.1, the convergence of basic fixed-point iterations will be sublinear and the convergence of Newton's method will typically be linear with rate 1/2. However, the minimal positive solution can be found easily by the Schur method described in Theorem 5.1. Since $We = 0$, we have $Se \leq e$ by the proof of Theorem 3.1. Since $W^T e = 0$, we can also get $S^T e \leq e$ by taking transpose for (1.2) and applying the proof of Theorem 3.1 to the new equation. Therefore, $S^T Se \leq S^T e \leq e$. Thus, $\rho(S^T S) \leq 1$. Now, by Proposition 5.3, $\kappa_2(U_{11}) \leq 1 + \rho(S^T S) \leq 2$. We apply the Schur method to a special example with $m = 100$, $B = C = I$, and

$$A = D = \begin{pmatrix} 2 & -1 & & \\ & 2 & \ddots & \\ & & \ddots & -1 \\ -1 & & & 2 \end{pmatrix}.$$

For this example, the minimal solution S must be doubly stochastic. In fact, $Se = e$ follows directly from Theorem 4.8, part (1) and $S^T e = e$ is obtained by taking transpose for (1.2) and applying Theorem 4.8, part (1) to the new equation. The approximate minimal solution is found by the Schur method with residual error 0.9896×10^{-13}. We also apply Newton's method to this equation. The residual error is 0.5683×10^{-13} after 22 iterations. The performance of Newton's method can be improved significantly by using the double Newton strategy as described in [14]. After 6 Newton iterations and 1 double Newton step, the residual error is 0.4649×10^{-14}. The basic fixed-point iterations are indeed extremely slow. We apply FP1 to the special example with $m = 5$ instead. It needs 399,985 iterations to make the residual error less than 10^{-10}. For this example, if an approximate solution has more digits than needed, chopping is recommended. By using chopping instead of rounding, we will have a much better chance to secure $\sigma(D - CS) \subset \mathbb{C}_{\geq}$ by the theory of nonnegative matrices.

Acknowledgment. The author thanks the referees for their very helpful comments.

REFERENCES

[1] M. T. Barlow, L. C. G. Rogers, and D. Williams, *Wiener–Hopf factorization for matrices*, in Séminaire de Probabilités XIV, Lecture Notes in Math. 784, Springer, Berlin, 1980, pp. 324–331.

[2] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.

[3] P. Benner and R. Byers, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Trans. Automat. Control, 43 (1998), pp. 101–107.

[4] P. Benner, V. Mehrmann, and H. Xu, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, J. Comput. Appl. Math., 86 (1997), pp. 17–43.

[5] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
[6] R. Byers, *A Hamiltonian QR algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[7] W. A. Coppel, *Matrix quadratic equations*, Bull. Austral. Math. Soc., 10 (1974), pp. 377–401.
[8] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.
[9] 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.
[10] I. Gohberg, P. Lancaster, and L. Rodman, *On Hermitian solutions of the symmetric algebraic Riccati equation*, SIAM J. Control Optim., 24 (1986), pp. 1323–1334.
[11] 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.
[12] C.-H. Guo and P. Lancaster, *Analysis and modification of Newton's method for algebraic Riccati equations*, Math. Comp., 67 (1998), pp. 1089–1105.
[13] C.-H. Guo and A. J. Laub, *On a Newton-like method for solving algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 694–698.
[14] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.
[15] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.
[16] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.
[17] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 228–243.
[18] C. Kenney, A. J. Laub, and M. Wette, *A stability-enhancing scaling procedure for Schur–Riccati solvers*, Systems Control Lett., 12 (1989), pp. 241–250.
[19] D. L. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.
[20] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, 1995.
[21] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, Orlando, FL, 1985.
[22] A. J. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.
[23] R. R. London, H. P. McKean, L. C. G. Rogers, and D. Williams, *A martingale approach to some Wiener–Hopf problems* II, in Séminaire de Probabilités XVI, Lecture Notes in Math. 920, Springer, Berlin, 1982, pp. 68–90.
[24] V. L. Mehrmann, *The Autonomous Linear Quadratic Control Problem*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Berlin, 1991.
[25] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric* M*-matrix*, Math. Comp., 31 (1977), pp. 148–162.
[26] H.-B. Meyer, *The matrix equation* $AZ+B-ZCZ-ZD=0$, SIAM J. Appl. Math., 30 (1976), pp. 136–142.
[27] C. Paige and C. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 41 (1981), pp. 11–32.
[28] L. C. G. Rogers, *Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains*, Ann. Appl. Probab., 4 (1994), pp. 390–413.
[29] L. C. G. Rogers and Z. Shi, *Computing the invariant law of a fluid model*, J. Appl. Probab., 31 (1994), pp. 885–896.
[30] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[31] G. W. Stewart, *HQR3 and EXCHNG: Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.
[32] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 121–135.
[33] C. Van Loan, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 61 (1984), pp. 233–251.
[34] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[35] D. Williams, *A "potential-theoretic" note on the quadratic Wiener–Hopf equation for Q-matrices*, in Séminaire de Probabilités XVI, Lecture Notes in Math. 920, Springer, Berlin, 1982, pp. 91–94.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 243–255
© 2001 Society for Industrial and Applied Mathematics

ORTHOGONAL TENSOR DECOMPOSITIONS*

TAMARA G. KOLDA†

Abstract. We explore the orthogonal decomposition of tensors (also known as multidimensional arrays or n-way arrays) using two different definitions of orthogonality. We present numerous examples to illustrate the difficulties in understanding such decompositions. We conclude with a counterexample to a tensor extension of the Eckart–Young SVD approximation theorem by Leibovici and Sabatier [*Linear Algebra Appl.*, 269 (1998), pp. 307–329].

Key words. tensor decomposition, singular value decomposition, principal components analysis, multidimensional arrays

AMS subject classifications. 15A69, 49M27, 62H25

PII. S0895479800368354

1. Introduction.

The problem of decomposing tensors (also called n-way arrays or multidimensional arrays) is approached in a variety of ways by extending the SVD, principal components analysis (PCA), and other methods to higher orders; see, e.g., [1, 3, 9, 10, 11, 12, 13, 14, 15]. Tensor decompositions are most often used for multimode statistical analysis and clustering, but they may also be used for compression of multidimensional arrays in ways similar to using a low-rank SVD for matrix compression. For example, color images are often stored as a sequence of RGB triplets, i.e., as separate red, green and blue overlays. An $m \times n$ pixel RGB image is represented by an $m \times n \times 3$ array, and a collection of p such images is an $m \times n \times 3 \times p$ array and can be compressed by a low-rank approximation.

The notation and basic properties of tensors are set forth in section 2. Several definitions of orthogonality and several rank orthogonal decompositions for tensors are given in section 3. Computational issues for orthogonal decompositions are discussed in section 4. Finally in section 5, we present a counterexample to Leibovici and Sabatier's extension to tensors of the well-known Eckart–Young SVD approximation theorem [13].

2. Tensors.

Let A be an $m_1 \times m_2 \times \cdots \times m_n$ *tensor* over $\mathbb{R}$. The *order* of A is n. The jth *dimension* of A is m_j. An element of A is specified as

$$A_{i_1 i_2 \cdots i_n},$$

where $i_j \in \{1, 2, \ldots, m_j\}$ for $j = 1, \ldots, n$. The set of all tensors of size $m_1 \times m_2 \times \cdots \times m_n$ is denoted by $\mathcal{T}(m_1, m_2, \ldots, m_n)$. The shorthand $\mathcal{T}_n$ may be used when only the order needs to be specified, or just $\mathcal{T}$ may be used when the order and dimensions are unambiguous.

*Received by the editors March 1, 2000; accepted for publication (in revised form) by N. Higham March 2, 2001; published electronically July 2, 2001. This work was supported by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy, under contracts DE–AC05–96OR22464 with Lockheed Martin Energy Research Corporation and DE-AC04-94AL85000 with Sandia Corporation. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/simax/23-1/36835.html

†Computational Science and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94551–9217 (tgkolda@sandia.gov).

Let $A, B \in \mathcal{T}(m_1, m_2, \ldots, m_n)$. The *inner product*[1] of A and B is defined as

$$A \cdot B \equiv \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_n=1}^{m_n} A_{i_1 i_2 \cdots i_n} B_{i_1 i_2 \cdots i_n}.$$

Correspondingly, the *norm* of A, $\|A\|$, is defined as

$$\|A\|^2 \equiv A \cdot A = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \cdots \sum_{i_n=1}^{m_n} A^2_{i_1 i_2 \cdots i_n}.$$

We say A is a *unit* tensor if $\|A\| = 1$.

Example 2.1. Let $x, y \in \mathcal{T}(m)$; that is, x, y are vectors in $\mathbb{R}^m$. Then $x \cdot y = x^T y$, where the superscript T denotes transpose. ∎

A *decomposed tensor* is a tensor $U \in \mathcal{T}(m_1, m_2, \ldots, m_n)$ that can be written as

$$U = u^{(1)} \otimes u^{(2)} \otimes \cdots \otimes u^{(n)}, \tag{2.1}$$

where $\otimes$ denotes the outer product and each $u^{(j)} \in \mathbb{R}^{m_j}$ for $j = 1, \ldots, n$. The vectors $u^{(j)}$ are called the *components* of U. In this case,

$$U_{i_1 i_2 \cdots i_n} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(n)}_{i_n}.$$

A decomposed tensor is a tensor of rank one for all the definitions of rank that we present in the next section. Decomposed tensors form the building blocks for tensor decompositions. The set of all decomposed tensors of size $m_1 \times m_2 \times \cdots \times m_n$ is denoted by $\mathcal{D}(m_1, m_2, \ldots, m_n)$ with shorthands analogous to $\mathcal{T}$.

LEMMA 2.2. *Let $U, V \in \mathcal{D}$, where U is defined as in* (2.1) *and V is defined by*

$$V = v^{(1)} \otimes v^{(2)} \otimes \cdots \otimes v^{(n)}. \tag{2.2}$$

Then

$$\text{(a)} \quad U \cdot V = \prod_{j=1}^{n} u^{(j)} \cdot v^{(j)}, \quad \text{(b)} \quad \|U\| = \prod_{j=1}^{n} \|u^{(j)}\|_2,$$

and (c) *$U + V \in \mathcal{D}$ if and only if all but at most one of the components of U and V are equal (within a scalar multiple).*

Proof. Items (a) and (b) follow directly from the definitions. For item (c), consider $U, V \in \mathcal{D}$ such that $n-1$ components are equal, i.e., $u^{(i)} = v^{(i)}$ for $i = 2, \ldots, n$. Then $W \equiv U + V$ can be written as

$$W = w^{(1)} \otimes u^{(2)} \otimes \cdots \otimes u^{(n)},$$

where $w^{(1)} = u^{(1)} + v^{(1)}$, so the "if" statement of (c) is true. Next we show the "only if" statement of (c). First consider the special case where $n = 2$, $m_1 = m_2 = 2$,

$$U \equiv \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix}, \quad V \equiv \begin{bmatrix} e \\ f \end{bmatrix} \otimes \begin{bmatrix} g \\ h \end{bmatrix},$$

[1] In [13], the term is "contracted product" and the notation is $\langle A, B \rangle$.

and $W \equiv U + V \in \mathcal{D}$. Since $W \in \mathcal{D}$, we can write it as

$$W \equiv \begin{bmatrix} p \\ q \end{bmatrix} \otimes \begin{bmatrix} r \\ s \end{bmatrix}.$$

Then, we have

$$pr = ac + eg, \tag{2.3}$$

$$ps = ad + eh, \tag{2.4}$$

$$qr = bc + fg, \tag{2.5}$$

$$qs = bd + fh. \tag{2.6}$$

Dividing (2.3) by (2.5) and (2.4) by (2.6) yields two ratios for p/q, and setting those equal gives

$$\frac{ac + eg}{bd + fh} = \frac{bc + fg}{ad + eh}. \tag{2.7}$$

Cross-multiplying and simplifying (2.7) finally yields

$$(af - be)(ch - dg) = 0.$$

In other words, either $u^{(1)} = v^{(1)}$ or $u^{(2)} = v^{(2)}$ (within a scalar multiple). So, all but at most one of the components of U and V must match if $W \in \mathcal{D}$. This argument can be extended to arbitrary n and m_j. ∎

We have shown that for two decomposed tensors to be combined to one decomposed tensor, they must match in all but at most one component. The same is not necessarily true, however, when combining three or more decomposed tensors, as shown in the next example.

Example 2.3. Consider the following example. Let $a, b \in \mathbb{R}^m$ with $a \perp b$ and $\|a\| = \|b\| = 1$. Define $c = \frac{1}{\sqrt{2}}(a + b)$, and

$$U_1 = a \otimes a \otimes a, \quad U_2 = a \otimes b \otimes c, \quad U_3 = a \otimes c \otimes b.$$

Then the sum of these three decomposed tensors can be rewritten as the sum of two despite the fact that they only match in one component:

$$U_1 + U_2 + U_3 = \sqrt{\frac{3}{2}}\,(V_1 + V_2)\,,$$

where

$$V_1 = a \otimes d \otimes a, \quad V_2 = a \otimes e \otimes b,$$

with

$$d = \sqrt{\frac{2}{3}}\,a + \sqrt{\frac{1}{3}}\,b, \quad e = \sqrt{\frac{2}{3}}\,c + \sqrt{\frac{1}{3}}\,b.$$

This is the result of splitting U_2 into two pieces based on the third component. ∎

We may also operate on tensors of different sizes. Specifically, tensors of different orders may be multiplied as follows. Suppose $C \in \mathcal{T}(m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_n)$ is

a tensor of order $n-1$ (note that m_j is missing). Then the *contracted product*[2] of A and C is a vector of length m_j, and its i_jth ($1 \le i_j \le m_j$) element is defined as

$$\langle A \cdot C \rangle_{i_j}^{(j)} \equiv \sum_{i_1=1}^{m_1} \cdots \sum_{i_{j-1}=1}^{m_{j-1}} \sum_{i_{j+1}=1}^{m_{j+1}} \cdots \sum_{i_n=1}^{m_n} A_{i_1 \cdots i_{j-1} i_j i_{j+1} \cdots i_n} C_{i_1 \cdots i_{j-1} i_{j+1} \cdots i_n}.$$

Note that the superscript on the bracketed product indicates which dimension is missing in the lower-order tensor C.

Example 2.4. Suppose $A \in \mathcal{T}(m_1, m_2)$ is a tensor of order two, i.e., A is a matrix. If $b \in \mathcal{T}(m_1)$, then $\langle A \cdot b \rangle^{(2)} = A^T b$ in matrix notation. Similarly, if $c \in \mathcal{T}(m_2)$, then $\langle A \cdot c \rangle^{(1)} = Ac$. □

LEMMA 2.5. *Let $U \in \mathcal{D}$ as defined in* (2.1) *and $A \in \mathcal{T}$. Then*

$$A \cdot U = \left\langle A \cdot u^{(1)} \otimes \cdots \otimes u^{(j-1)} \otimes u^{(j+1)} \otimes \cdots \otimes u^{(n)} \right\rangle^{(j)} \cdot u^{(j)}.$$

Proof. The proof follows from the definitions. □

3. Orthogonal rank decompositions.

3.1. Notions of orthogonality. Let $U, V \in \mathcal{D}$, be defined as in (2.1) and (2.2), respectively. Without loss of generality, we assume $\|U\| = \|V\| = 1$ and that the components are unit vectors. We say that U and V are *orthogonal* ($U \perp V$) if

$$U \cdot V = \prod_{j=1}^{n} u^{(j)} \cdot v^{(j)} = 0.$$

We say that U and V are *completely orthogonal* ($U \perp_{\mathrm{c}} V$) if for every $j = 1 \ldots, n$,

$$u^{(j)} \perp v^{(j)}.$$

We say that U and V are *strongly orthogonal* ($U \perp_{\mathrm{s}} V$) if $U \perp V$ and for every $j = 1, \ldots, n$,

$$u^{(j)} = \pm\, v^{(j)} \text{ or } u^{(j)} \perp v^{(j)}.$$

From the definition of strong orthogonality, it follows that at least one pair must satisfy $u^{(j)} \perp v^{(j)}$ since we require $U \perp V$. Note that we could write $u^{(j)} = \pm\, v^{(j)}$ more generally as $u^{(j)} = \lambda_j\, v^{(j)}$ for some $\lambda_j \neq 0$, which is useful when $\|U\| \neq \|V\|$.

The relationship between the different orthogonality definitions is given in the following lemma.

LEMMA 3.1. *Let the decomposed tensors U and V of order n be defined as in* (2.1) *and* (2.2)*, respectively. Then*

$$U \perp_{\mathrm{c}} V \Rightarrow U \perp_{\mathrm{s}} V \Rightarrow U \perp V.$$

3.2. Rank decompositions. Our goal is to express a tensor $A \in \mathcal{T}$ as a weighted sum of decomposed tensors,

$$A = \sum_{i=1}^{r} \sigma_i U_i, \tag{3.1}$$

where $\sigma_i > 0$ for $i = 1, \ldots, r$ and each $U_i \in \mathcal{D}$ and $\|U_i\| = 1$ for $i = 1, \ldots, r$.

[2] In [13], the notation $A \,..\, C$ is used for contracted products.

- The *rank* of A, denoted $\text{rank}(A)$, is defined to be the minimal r such that A can be expressed as in (3.1). The decomposition is called the *rank decomposition*.
- The *orthogonal rank* of A, denoted $\text{rank}_{\perp}(A)$, is defined to be the minimal r such that A can be expressed as in (3.1) and $U_i \perp U_j$ for all $i \neq j$. The decomposition is called the *orthogonal rank decomposition*.
- The *strong orthogonal rank* of A, denoted $\text{rank}_{\perp_s}(A)$, is defined to be the minimal r such that A can be expressed as in (3.1) and $U_i \perp_s U_j$ for all $i \neq j$. The decomposition is called the *strong orthogonal rank decomposition*.[3]

As reported in [13], the definition of rank is due to Kruskal, although it was proposed even earlier by Strassen and others (see [11] and references therein), and the definitions of orthogonal and strong orthogonal rank is due to Franc [7]. The general *decomposition*, *orthogonal decomposition*, and *strong orthogonal decomposition* satisfy the orthogonality constraints (if any) but are not necessarily minimal in terms of r. For matrices, all three rank decompositions are equivalent to the SVD.

LEMMA 3.2. *The rank, orthogonal rank, and strong orthogonal rank decomposition are each equivalent to the SVD for tensors of order two.*

Proof. This follows from the properties of the SVD (cf. [8]). ∎

In our discussion of rank decomposition, we did not present a *completely orthogonal decomposition*. In fact, we are not in general guaranteed that such a decomposition can be found, as we discuss later in this section.

A slightly different notion of rank that depends on special orthogonal decomposition is the *combinatorial orthogonal rank*, denoted $\text{rank}_{\perp_t}(A)$. It is defined as the minimal r such that A can be written as

$$A = \sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \cdots \sum_{i_n=1}^{r} \sigma_{i_1 i_2 \cdots i_n} \, u_{i_1}^{(1)} \otimes u_{i_2}^{(2)} \otimes \cdots u_{i_n}^{(n)}, \tag{3.2}$$

where $\sigma_{i_1 i_2 \cdots i_n} > 0$; $u_i^{(j)} \in \mathbb{R}^{m_j}$ with $\|u_i^{(j)}\| = 1$ for $1 \leq i \leq r$ and $1 \leq j \leq n$; and further, $u_{i_1}^{(j)} \perp u_{i_2}^{(j)}$ for all $i_1 \neq i_2$, $1 \leq i_2, i_2 \leq r$, $1 \leq j \leq n$. Equivalently, let

$$U_i = u_i^{(1)} \otimes u_i^{(2)} \otimes \cdots \otimes u_i^{(n)}$$

and require $U_{i_1} \perp_c U_{i_2}$ for all $i_1 \neq i_2$, $1 \leq i_1, i_2 \leq r$, and $\|U_i\| = 1$, $1 \leq i \leq r$. In other words, the decomposition (3.2) is the result of combining the components of the U_i's in every possible way and is called the *combinatorial orthogonal rank decomposition*. In this case, there are r^n scalar multiples (i.e., σ-values) that are involved rather than just r as in the other decompositions. This is the Tucker decomposition with orthogonality constraints [14], hence the subscript in the notation. Note that the SVD of a matrix is a combinatorial orthogonal rank decomposition, but the reverse is not necessarily true.

Now we consider several examples that illustrate that the rank decompositions are not necessarily unique.

Example 3.3. Let $a, b \in \mathbb{R}^m$ with $a \perp b$ and $\|a\| = \|b\| = 1$, and let $\sigma_1 > \sigma_2 > \sigma_3 > 0$. Define $A \in \mathcal{T}(m, m, m)$ as

$$A = \sigma_1 \underbrace{a \otimes b \otimes b}_{U_1} + \sigma_2 \underbrace{b \otimes b \otimes b}_{U_2} + \sigma_3 \underbrace{a \otimes a \otimes a}_{U_3}. \tag{3.3}$$

[3] In [13], the terms "free orthogonal rank" and "free rank decomposition" are used rather than "strong orthogonal rank" and "strong orthogonal rank decomposition."

Note that $U_i \perp_s U_j$ for all $i \neq j$, so (3.3) is a strong orthogonal decomposition of A. Furthermore, A cannot be expressed as the sum of fewer weighted strong orthogonal decomposed tensors, so the strong orthogonal rank of A is three. Observe that A can also be expressed as

$$A = \hat{\sigma}_1 \underbrace{\hat{a} \otimes b \otimes b}_{\hat{U}_1} + \hat{\sigma}_2 \underbrace{\hat{a} \otimes a \otimes a}_{\hat{U}_2} + \hat{\sigma}_3 \underbrace{\hat{b} \otimes a \otimes a}_{\hat{U}_3}, \tag{3.4}$$

where

$$\hat{\sigma}_1 = \sqrt{\sigma_1^2 + \sigma_2^2}\ , \quad \hat{\sigma}_2 = \frac{\sigma_1\ \sigma_3}{\hat{\sigma}_1}\ , \quad \hat{\sigma}_3 = \frac{\sigma_2\ \sigma_3}{\hat{\sigma}_1}\ ,$$

$$\hat{a} = \frac{\sigma_1\ a + \sigma_2\ b}{\hat{\sigma}_1}\ , \quad \text{and} \quad \hat{b} = \frac{\sigma_2\ a - \sigma_1\ b}{\hat{\sigma}_1}\ .$$

Since $\hat{a} \perp \hat{b}$, we have $\hat{U}_i \perp_s \hat{U}_j$ for all $i \neq j$. Therefore (3.4) is also a strong orthogonal rank decomposition of A, and so the strong orthogonal rank decomposition is not unique. ∎

Example 3.4. Consider the tensor A as defined by (3.3); A can also be written as

$$A = \bar{\sigma}\bar{U} + \sigma_3 U_3, \tag{3.5}$$

where

$$\bar{\sigma} = \sqrt{\sigma_1^2 + \sigma_2^2} \quad \text{and} \quad \bar{U} = \frac{\sigma_1\ a + \sigma_2\ b}{\bar{\sigma}} \otimes b \otimes b.$$

Observe that $\bar{U} \perp U_3$; in fact, (3.5) is an orthogonal rank decomposition of A, and therefore the orthogonal rank of A is two. An alternative orthogonal rank decomposition of A is given by

$$A = \tilde{\sigma}\ \tilde{U} + \sigma_2 U_2, \tag{3.6}$$

where

$$\tilde{\sigma} = \sqrt{\sigma_1^2 + \sigma_3^2} \quad \text{and} \quad \tilde{U} = a \otimes \frac{\sigma_1\ b + \sigma_3\ a}{\tilde{\sigma}} \otimes b.$$

Note that $\tilde{U} \perp \hat{U}_2$, so (3.6) is also an orthogonal rank decomposition of A and the orthogonal rank decomposition is not unique. ∎

LEMMA 3.5. *Neither the orthogonal rank, strong orthogonal rank, nor combinatorial orthogonal rank decomposition is unique.*

Proof. See Examples 3.3 and 3.4. ∎

Although the SVD for matrices is known to be unique up to rotation [8], the rank tensor decompositions are not. This is an important difference which we return to later in this section.

Example 3.6. We show how to "orthogonalize" a tensor in a relatively simple situation. Suppose that we have an order three tensor $A \in \mathcal{T}(m_1, m_2, m_3)$ defined as follows:

$$A = \sigma_1 U + \sigma_2 V,$$

where $\sigma_1 \geq \sigma_2$ and

$$U = u^{(1)} \otimes u^{(2)} \otimes u^{(3)},$$
$$V = v^{(1)} \otimes v^{(2)} \otimes v^{(3)}$$

with $u^{(i)}, v^{(i)}$ unequal, nonorthogonal unit vectors in $\mathbb{R}^{m_i}$ for $i = 1, 2, 3$.

For $i = 1, 2, 3$, we can decompose $v^{(i)}$ as

$$v^{(i)} = \alpha^{(i)} u^{(i)} + \hat{\alpha}^{(i)} \hat{u}^{(i)},$$

where

$$\alpha^{(i)} = v^{(i)} \cdot u^{(i)},$$
$$\hat{\alpha}^{(i)} = \|v^{(i)} - \alpha^{(i)} u^{(i)}\|, \text{ and}$$
$$\hat{u}^{(i)} = (v^{(i)} - \alpha^{(i)} u^{(i)})/\hat{\alpha}^{(i)}.$$

Then, we can rewrite A as

$$\begin{array}{rcll} A & = & (\sigma_1 + \sigma_2\, \alpha^{(1)}\alpha^{(2)}\alpha^{(3)}) & u^{(1)} \otimes u^{(2)} \otimes u^{(3)} \\ & + & \sigma_2\, \alpha^{(1)}\alpha^{(2)}\hat{\alpha}^{(3)} & u^{(1)} \otimes u^{(2)} \otimes \hat{u}^{(3)} \\ & + & \sigma_2\, \alpha^{(1)}\hat{\alpha}^{(2)}\alpha^{(3)} & u^{(1)} \otimes \hat{u}^{(2)} \otimes u^{(3)} \\ & + & \sigma_2\, \alpha^{(1)}\hat{\alpha}^{(2)}\hat{\alpha}^{(3)} & u^{(1)} \otimes \hat{u}^{(2)} \otimes \hat{u}^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\alpha^{(2)}\alpha^{(3)} & \hat{u}^{(1)} \otimes u^{(2)} \otimes u^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\alpha^{(2)}\hat{\alpha}^{(3)} & \hat{u}^{(1)} \otimes u^{(2)} \otimes \hat{u}^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\hat{\alpha}^{(2)}\alpha^{(3)} & \hat{u}^{(1)} \otimes \hat{u}^{(2)} \otimes u^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\hat{\alpha}^{(2)}\hat{\alpha}^{(3)} & \hat{u}^{(1)} \otimes \hat{u}^{(2)} \otimes \hat{u}^{(3)}. \end{array} \tag{3.7}$$

Equation (3.7) shows that $\text{rank}_{\perp_s}(A) \leq 8$. Because of the way U and V were chosen (components neither equal nor orthogonal), (3.7) is a strong orthogonal rank decomposition of A, and $\text{rank}_{\perp_s}(A) = 8$. (From (3.7), we can also deduce that $\text{rank}_{\perp_t}(A) = 2$.) This is not, however, an orthogonal rank decomposition. Combining each pair of lines in (3.7), we get

$$\begin{array}{rcll} A & = & \sqrt{\gamma^2 + \hat{\gamma}^2} & u^{(1)} \otimes u^{(2)} \otimes (\gamma u^{(3)} + \hat{\gamma}\hat{u}^{(3)})/\sqrt{\gamma^2 + \hat{\gamma}^2} \\ & + & \sigma_2\, \alpha^{(1)}\hat{\alpha}^{(2)} & u^{(1)} \otimes \hat{u}^{(2)} \otimes v^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\alpha^{(2)} & \hat{u}^{(1)} \otimes u^{(2)} \otimes v^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)}\hat{\alpha}^{(2)} & \hat{u}^{(1)} \otimes \hat{u}^{(2)} \otimes v^{(3)}, \end{array} \tag{3.8}$$

where

$$\gamma = \sigma_1 + \sigma_2\, \alpha^{(1)}\alpha^{(2)}\alpha^{(3)} \quad \text{and} \quad \hat{\gamma} = \sigma_2\, \alpha^{(1)}\alpha^{(2)}\hat{\alpha}^{(3)}.$$

Finally, combining the last two lines of (3.8), we arrive at an orthogonal rank decomposition

$$\begin{array}{rcll} A & = & \sqrt{\gamma^2 + \hat{\gamma}^2} & u^{(1)} \otimes u^{(2)} \otimes (\gamma u^{(3)} + \hat{\gamma}\hat{u}^{(3)})/\sqrt{\gamma^2 + \hat{\gamma}^2} \\ & + & \sigma_2\, \alpha^{(1)}\hat{\alpha}^{(2)} & u^{(1)} \otimes \hat{u}^{(2)} \otimes v^{(3)} \\ & + & \sigma_2\, \hat{\alpha}^{(1)} & \hat{u}^{(1)} \otimes v^{(2)} \otimes v^{(3)}, \end{array}$$

so $\text{rank}_{\perp}(A) = 3$. Note that combining vectors from (3.7) in different order would have resulted in a different orthogonal rank decomposition. ∎

We now see some relationship between the different ranks, stated formally in the next theorem.

THEOREM 3.7 (see [13]). *For a given tensor A,*

$$\operatorname{rank}(A) \leq \operatorname{rank}_{\perp}(A) \leq \operatorname{rank}_{\perp_{\mathrm{s}}}(A). \tag{3.9}$$

Further, for any order $n > 2$, there exists $A \in \mathcal{T}_n$ such that strict inequality holds.

Proof. The first part follows from Lemma 3.1. An example of strict inequality for a tensor of order three ($n = 3$) is given in Example 3.6, and that example can be generalized to any order. ∎

For a matrix, all four definitions of tensor rank reduce to the standard definition of matrix rank.

COROLLARY 3.8 (see [13]). *For any $A \in \mathcal{T}_2$,*

$$\operatorname{rank}(A) = \operatorname{rank}_{\perp}(A) = \operatorname{rank}_{\perp_{\mathrm{s}}}(A) = \operatorname{rank}_{\perp_{\mathrm{t}}}(A).$$

Proof. This follows from Lemma 3.2. ∎

Earlier we mentioned the notion of a completely orthogonal decomposition; this corresponds to a combinatorial orthogonal decomposition in which only the diagonal elements ($\sigma_{ii\cdots i}$) are nonzero; and so, in general, tensors cannot be *diagonalized*. A similar observation was made by Denis and Dhorne [4]. When a tensor can be diagonalized, all the ranks are equal.

COROLLARY 3.9 (see [13]). *For any order $n > 2$, there exists $A \in \mathcal{T}_n$ such that A cannot be decomposed as the weighted sum of completely orthogonal tensors. If a tensor can be decomposed as the weighted sum of completely orthogonal decomposed tensors, then equality holds in* (3.9).

Proof. See the construction of the decompositions of A in Example 3.6 to prove the first statement. The second statement follows intuitively from the fact that each subspace has dimension r, and the rank of the tensor cannot be less than the smallest-dimensional subspace. ∎

Franc [6] made observations similar to Theorem 3.7 and Corollary 3.9. Matrices (i.e., tensors of order two) are special cases that always have a completely orthogonal decomposition, as follows from Corollaries 3.8 and 3.9.

We now return to the concept of uniqueness in the rank decomposition. We have several examples illustrating that the strong orthogonal rank and orthogonal rank decompositions are not unique. A partial "fix" for lack of uniqueness is the following. Without loss of generality, assume that the σ_i's in (3.1) are always ordered so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Then define the *unique* (strong) orthogonal rank decomposition to be the (strong) orthogonal rank decomposition that has the largest possible σ_1, and given that choice for σ_1, has the largest possible σ_2, and so forth. This decomposition is unique in the sense that the weights are unique. The unit decomposed tensors are unique if and only if no two σ_i's are equal, similar to the fact that the SVD is unique up to rotation. A *unique* combinatorial orthogonal rank decomposition can be defined in a more complicated way by sequentially choosing each U_k so that

$$\sum_{i_1=1}^{k} \sum_{i_2=1}^{k} \cdots \sum_{i_n=1}^{k} \sigma_{i_1 i_2 \cdots i_n}^2$$

is maximized.

Example 3.10. In Example 3.3, the unique strong orthogonal rank decomposition is given by (3.4). Similarly, in Example 3.4, the unique orthogonal rank decomposition is given by (3.5). ∎

4. Greedy tensor decompositions. We now consider the computation of an orthogonal decomposition and present a method for generating a *greedy orthogonal decomposition*. Our goal is to compute a sequence (for $p = 1, 2, \ldots$) of weighted decomposed tensors such that

$$A = \sum_{i=1}^{p} \sigma_i U_i,$$

where $U_i \perp U_j$ for all $i \neq j$ and $\|U_i\| = 1$ for all i. We call this the greedy orthogonal decomposition because the $\{\sigma, U\}$ pairs are computed iteratively. We do not yet make any claims as to whether or not this greedy orthogonal decomposition yields an orthogonal rank decomposition.

In the greedy orthogonal decomposition, define the kth residual tensor as

$$R_k \equiv A - \sum_{i=1}^{k} \sigma_i U_i$$

with $R_0 = A$, and let the set of tensors $\mathcal{U}_k$ be defined as

$$\mathcal{U}_k = \{U_1, U_2, \ldots, U_k\}$$

with $\mathcal{U}_0 = \emptyset$. Our goal is to find the best rank one approximation to the current residual subject to orthogonality constraints; that is, we wish to solve

$$\min \ f_k(\sigma, U) \equiv \|R_k - \sigma U\|^2 \ \text{ subject to (s.t.) } \ U \in \mathcal{D}, \ \ \|U\| = 1, \ \ U \perp \mathcal{U}_k.$$

We can rewrite f_k as

$$f_k(\sigma, U) = \|R_k\|^2 - 2\sigma R_k \cdot U + \sigma^2 \|U\|^2.$$

At the solution, we have

$$\frac{\partial f_k}{\partial \sigma} \equiv -2R_k \cdot U + 2\sigma \|U\|^2 = 0,$$

so we can solve for σ and conclude that minimizing f_k is the same as solving

$$\max \ R_k \cdot U \ \text{ s.t. } \ U \in \mathcal{D}, \ \ \|U\| = 1, \ \ U \perp \mathcal{U}_k. \tag{4.1}$$

We define U_{k+1} to be the solution of (4.1) and let $\sigma_{k+1} = R_k \cdot U_{k+1}$. We repeat the process until $R_{k+1} = 0$.

A *greedy strong orthogonal decomposition* can be similarly described, and it reduces to solving

$$\max \ R_k \cdot U \ \text{ s.t. } \ U \in \mathcal{D}, \ \ \|U\| = 1, \ \ U \perp_{\mathrm{s}} \mathcal{U}_k, \tag{4.2}$$

at each iteration. Likewise, we may also construct a sort of greedy approach for the combinatorial orthogonal decomposition.

LEMMA 4.1. *The greedy orthogonal, strong orthogonal, and combinatorial decompositions are finite.*

Proof. This is a consequence of the fact that there are at most $M = \prod_{j=1}^{n} m_j$ orthogonal or strong orthogonal decomposed tensors. ∎

Solving (4.1) or (4.2) is a very challenging task. For example, in order to solve (4.1), we might use an *alternating least squares* (ALS) approach as follows. For $\ell = 1, \ldots, n$, fix all components of U but the ℓth and solve

$$\max \; s \cdot u^{(\ell)} \;\text{ s.t. }\; \|U\| = 1, \;\; U \perp \mathcal{U}_k,$$

where

$$s = \left\langle R_k \cdot u^{(1)} \otimes \cdots \otimes u^{(\ell-1)} \otimes u^{(\ell+1)} \otimes \cdots \otimes u^{(n)} \right\rangle^{(\ell)}.$$

The difficulty with this approach is in enforcing the constraints.

Zhang and Golub [15] explore various computational techniques when the tensor has a completely orthogonal decomposition, in which case the problem is much simpler. In [13], the RPVSCC method uses ALS to find the *modes*, i.e., the completely orthogonal decomposed tensors, and then fills in the values associated with the combinations of the components of the modes. De Lathauwer [3] presents several ALS methods for computing the higher-order SVD. Kroonenberg and de Leeuw [10] propose an ALS solution to (3.2) so that at each step an entire set $\{u_i^{(j)}\}_{i=1}^{m_j}$ is solved for some j while everything else is fixed. In other words, the method concentrates on one subspace at a time.

5. Approximation of a tensor. The well-known Eckart–Young approximation theorem [5, 8] says that if the SVD of a matrix is given by

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, then the best rank-$k$ approximation is given by

$$A_k \equiv \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

A consequence of this result is that the SVD can be computed via a greedy method which calculates each triplet $\{\sigma_i, u_i, v_i\}$ in sequence. Now we can ask whether or not the Eckart–Young theorem can be extended to tensor rank decompositions; i.e., is the best rank-k approximation of a tensor given by the sum of the first k terms in its rank decomposition? This relates directly to whether or not the greedy orthogonal, strong orthogonal, or combinatorial decompositions produce a corresponding rank decomposition.

In the case of the strong orthogonal rank decomposition, the answer is definitely no, contrary to the result stated in [13], as the following counterexample shows.

Example 5.1. Consider the strong orthogonal rank decomposition of a matrix $A \in \mathcal{T}(m, m, m)$ defined by

$$A = \sum_{i=1}^{6} \sigma_i U_i,$$

where the $\{\sigma_i, U_i\}$ pairs are defined as follows. Let the vectors $a, b, c, d \in \mathbb{R}^m$ be

two-by-two orthogonal; then let

$$\begin{array}{rclrcl} \sigma_1 &=& 1.00, & U_1 &=& a \otimes a \otimes a, \\ \sigma_2 &=& 0.75, & U_2 &=& b \otimes b \otimes b, \\ \sigma_3 &=& 0.70, & U_3 &=& a \otimes c \otimes d, \\ \sigma_4 &=& 0.70, & U_4 &=& a \otimes d \otimes c, \\ \sigma_5 &=& 0.65, & U_5 &=& b \otimes c \otimes d, \\ \sigma_6 &=& 0.65, & U_6 &=& b \otimes d \otimes c. \end{array}$$

Note that $\sigma_3 U_3$ and $\sigma_5 U_5$ can be combined to form the decomposed tensor

$$\gamma_1 V_1 \equiv \sqrt{\sigma_3^2 + \sigma_5^2}\, \frac{\sigma_3 a + \sigma_5 b}{\sqrt{\sigma_3^2 + \sigma_5^2}} \otimes c \otimes d. \tag{5.1}$$

Similarly, $\sigma_4 U_4$ and $\sigma_6 U_6$ can be combined to form

$$\gamma_2 V_2 \equiv \sqrt{\sigma_4^2 + \sigma_6^2}\, \frac{\sigma_4 a + \sigma_6 b}{\sqrt{\sigma_4^2 + \sigma_6^2}} \otimes d \otimes c. \tag{5.2}$$

However,

$$\gamma_1 = \gamma_2 \approx 0.9552 < \sigma_1 = 1,$$

so neither (5.1) nor (5.2) is the best rank one approximation to A; $A_1 \equiv \sigma_1 U_1$ is. However, the best strong orthogonal rank two approximation is given by

$$A_2 \equiv \gamma_1 V_1 + \gamma_2 V_2$$

because $V_1 \perp_{\mathrm{s}} V_2$ and

$$\gamma_1^2 + \gamma_2^2 = 1.825 > \sigma_1^2 + \sigma_2^2 = 1.5625.$$

Thus, we have a counterexample to any Eckart–Young-type theorem for strong orthogonal rank decompositions. ∎

Example 5.1 can be reworked as follows to show that the combinatorial orthogonal rank decomposition does not yield a best rank-k approximation either.

Example 5.2. Consider the tensor defined in Example 5.1. Let e and f be any vectors that are orthogonal to each other and also to a and b. We can express a combinatorial orthogonal rank decomposition of A as follows:

$$A = \sum_{i_1=1}^{4} \sum_{i_2=1}^{4} \sum_{i_3=1}^{4} \bar{\sigma}_{i_1 i_2 i_3} \bar{u}_{i_1}^{(1)} \otimes \bar{u}_{i_2}^{(2)} \otimes \bar{u}_{i_3}^{(3)},$$

where

$$\begin{array}{rclrcl} \bar{U}_1 &=& a \otimes a \otimes a, & \bar{U}_3 &=& e \otimes c \otimes d, \\ \bar{U}_2 &=& b \otimes b \otimes b, & \bar{U}_4 &=& f \otimes d \otimes c, \end{array}$$

and the only nonzero $\bar{\sigma}$'s are

$$\bar{\sigma}_{111} = \sigma_1, \;\; \bar{\sigma}_{222} = \sigma_2, \;\; \bar{\sigma}_{133} = \sigma_3, \;\; \bar{\sigma}_{233} = \sigma_4, \;\; \bar{\sigma}_{144} = \sigma_5, \;\; \bar{\sigma}_{244} = \sigma_6.$$

So, $\text{rank}_{\perp_t}(A) = 4$. The best combinatorial orthogonal rank one approximation to A is $\bar{A}_1 = \bar{\sigma}_{111}\bar{U}_1 = \sigma_1 U_1$ (the same as the best strong orthogonal rank one approximation). However, the best combinatorial orthogonal rank two approximation is yielded by

$$\bar{A}_2 = \sum_{i_1=1}^{2}\sum_{i_2=1}^{2}\sum_{i_3=1}^{2} \bar{\gamma}_{i_1 i_2 i_3}\bar{v}_{i_1}^{(1)} \otimes \bar{v}_{i_2}^{(2)} \otimes \bar{v}_{i_3}^{(3)}.$$

Here

$$\bar{V}_1 \equiv V_1 \quad \text{and} \quad \bar{V}_2 \equiv g \otimes d \otimes c,$$

where g is some vector orthogonal to $v_1^{(1)}$, and the only nonzero $\bar{\gamma}$'s are $\bar{\gamma}_{111} = \gamma_1$ and $\bar{\gamma}_{122} = \gamma_2$. □

The problem of whether or not the Eckart–Young result can be extended to the orthogonal decomposition is still an open question. Example 2.3 shows that it is possible to add an orthogonal decomposed tensor to a sum without increasing its rank ($U_1 + U_2$ has rank 2 as does $U_1 + U_2 + U_3$). This is contrary to a fundamental assumption used in the proof of Theorem 2 in [13]. We also have the problem of uniqueness since, by Example 3.4, we know that the orthogonal decomposition is not unique. One possible solution to this problem is the definition proposed at the end of section 3.2. We now seek either a proof or counterexample of the following.

Open Problem 5.3 (Eckart–Young extended). *Let the unique orthogonal rank decomposition of a tensor A be given as in* (3.1) *and assume that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Then the best orthogonal rank p $(p < r)$ approximation to A satisfies*

$$\min_{\text{rank}_\perp A_p = p} \|A - A_p\|^2 = \sum_{i=p+1}^{r} \sigma_i^2$$

and is given by

$$A_p \equiv \sum_{i=1}^{p} \sigma_i u_i.$$

6. Conclusions. There are multiple ways to orthogonally decompose tensors, depending both on the definition of orthogonality as well as on the definitions of decomposition and rank. An Eckart–Young type of best rank-k approximation theorem for tensors continues to elude our investigations but can perhaps eventually be attained by using a different norm or yet other definitions of orthogonality and rank.

Computing an orthogonal tensor decomposition is a challenge as well. Most methods are variations on ALS, a method which can be very slow to converge, although recently several authors (cf. [3, 15]) have presented new ideas.

Acknowledgments. The author is grateful to the referees for their careful reading of this paper and for their many useful comments, including pointing out references [4, 6]. The author also wishes to thank Dianne O'Leary and Paul Boggs for their helpful suggestions in the preparation of this manuscript.

REFERENCES

[1] J. D. Carroll and J.-J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[2] R. Coppi and S. Bolasco, eds., *Multiway Data Analysis*, Elsevier Science Publishers, North-Holland, Amsterdam, 1989.

[3] L. de Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, Katholieke Universiteit Leuven, Belgium, 1997.

[4] J. B. Denis and T. Dhorne, *Orthogonal tensor decomposition of* 3*-way tables*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Elsevier Science Publishers, North-Holland, Amsterdam, 1989, pp. 31–37.

[5] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[6] A. Franc, *Multiway arrays: Some algebraic remarks*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Elsevier Science Publishers, North-Holland, Amsterdam, 1989, pp. 19–29.

[7] A. Franc, *Etude Algébrique des Multitableaux: Apports de l'Algébre Tensorielle*, Thèse de Doctorat, Spécialité Statistiques, Univ. de Montpellier II, Montpellier, 1992.

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

[9] A. Kapteyn, H. Neudecker, and T. Wansbeek, *An approach to n-mode components analysis*, Psychometrika, 51 (1986), pp. 269–275.

[10] P. M. Kroonenberg and J. de Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[11] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[12] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and n-way arrays*, in Multiway Data Analysis, Elsevier Science Publishers, North-Holland, Amsterdam, 1989, pp. 7–18.

[13] D. Leibovici and R. Sabatier, *A singular value decomposition of a k-way array for principal component analysis of multiway data, PTA-k*, Linear Algebra Appl., 269 (1998), pp. 307–329.

[14] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[15] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., submitted.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 256–270
© 2001 Society for Industrial and Applied Mathematics

THE EVEN-ODD SPLIT LEVINSON ALGORITHM FOR TOEPLITZ SYSTEMS*

A. MELMAN†

Abstract. We derive an algorithm for real symmetric Toeplitz systems with an arbitrary right-hand side, which differs from both the Levinson and the so-called "split Levinson" algorithms. While exploiting ideas from the split Levinson approach, it also takes advantage of the even-odd properties of Toeplitz matrices. For a system of order n, our algorithm achieves a complexity of $\frac{5}{2}n^2 + \mathcal{O}(n)$ flops on a sequential machine, compared to $3n^2 + \mathcal{O}(n)$ flops for the split Levinson algorithm and $4n^2 + \mathcal{O}(n)$ flops for the classical Levinson algorithm.

Key words. Toeplitz system, even-odd structure, Durbin, Levinson, split Levinson

AMS subject classification. 65F05

PII. S0895479800367129

1. Introduction.

Toeplitz matrices occur in a host of applications in engineering (signal and image processing), numerical analysis, and elsewhere. The most common problems involving these matrices are the computation of some or all of their eigenvalues and the solution of systems of linear equations that have such matrices as their coefficient matrix. A good overview of the many areas in which Toeplitz problems are encountered can be found in [5].

Eigenvalue problems appear in certain signal processing problems such as harmonic retrieval (see, e.g., [21]) and were considered, among many others, e.g., in [8] and [22]. There is a wide range of applications of linear systems with a Toeplitz coefficient matrix: the computation of Padé approximations is but one example. In signal processing, such systems are ubiquitous, most notably in linear prediction (see, e.g., [19]). There exists a vast body of literature on Toeplitz matrices of which it would be impossible to give a complete list. Let us just mention that many early general results can be found in [14].

This work concentrates on real symmetric Toeplitz systems, and we restrict ourselves to the important case where all principal submatrices are nonsingular. Such cases are common in many physical applications in which the matrices involved are positive-definite.

Two classes of methods, specially tailored for solving such systems, are the so-called *fast methods*, with a complexity of $\mathcal{O}(n^2)$ flops (for an overview, we refer the reader to [13]) and the so-called *superfast methods* (see, e.g., [1], [2], [4]), which are based on the fast Fourier transform and have a complexity of $\mathcal{O}(n \log^2 n)$ flops. (Following [13], we define a flop, or floating-point operation, as an addition, subtraction, multiplication, or division.) However, for matrix dimensions of up to several hundred, the fast methods need fewer operations, and it is on those methods that we will concentrate.

A classical algorithm for solving symmetric Toeplitz systems with an arbitrary right-hand side is the one by Levinson [18], which is related to the theory of orthog-

*Received by the editors February 7, 2000; accepted for publication (in revised form) by L. Reichel January 16, 2001; published electronically July 2, 2001.
http://www.siam.org/journals/simax/23-1/36712.html

†Department of Mathematics, University of San Francisco, San Francisco, CA 94117-1080 (melman@euclid.math.usfca.edu). The author is currently on leave from Ben-Gurion University, Beer-Sheva, Israel.

onal polynomials on the unit circle (see, e.g., [15], [16]). This algorithm requires the solution of a special system, the so-called *Yule–Walker* system, for which the right-hand side contains part of the first row of the matrix. If Durbin's method [12] is used for this specialized system, the complexity of Levinson's algorithm on a sequential machine is $4n^2 + \mathcal{O}(n)$ flops for a system of order n. The complexity of Durbin's algorithm for the Yule–Walker system is $2n^2 + \mathcal{O}(n)$ flops. This algorithm was improved by Delsarte and Genin in [10] by proposing the "split Levinson" method for the Yule–Walker system, achieving a complexity of $\frac{3}{2}n^2 + \mathcal{O}(n)$ flops. An algorithm for a system with arbitrary right-hand side, proposed by the same authors in [11], improved upon Levinson's algorithm by achieving a complexity of $3n^2 + \mathcal{O}(n)$ flops.

However, Toeplitz matrices exhibit an interesting even-odd structure, which manifests itself most clearly in their eigenvalues, of which there are two types: even and odd, belonging to even and odd eigenvectors, respectively (see, e.g., [6]). Furthermore, this even-odd structure can be exploited for linear systems as well. This was done in [20], where it led to yet another fast method for the general right-hand side problem. Since this method is able to exploit even-odd properties of the right-hand side, which the aforementioned methods cannot, we will call it the "even-odd Levinson algorithm." It has a complexity of $\frac{7}{2}n^2 + \mathcal{O}(n)$ flops.

In this work we combine the even-odd Levinson algorithm with the split Levinson algorithm from [11] to construct a method for the general right-hand side problem, which we call the "even-odd split Levinson algorithm," and which achieves a complexity of $\frac{5}{2}n^2 + \mathcal{O}(n)$ flops. Moreover, it computes either the even or the odd part of the solution with only $2n^2 + \mathcal{O}(n)$ flops, so that, if two independent processors were available, the solution itself could also be obtained with only $2n^2 + \mathcal{O}(n)$ flops. This contrasts with the Levinson and split Levinson methods, the complexity of which in such cases remains the same at $4n^2 + \mathcal{O}(n)$ and $3n^2 + \mathcal{O}(n)$ flops, respectively.

We reported only the coefficient of n^2 in all complexity results, which we consider justified because of the relatively small coefficient of n (never more than roughly 20) and the negative constant term for all methods concerned. This means that for $n \geq 15$ the coefficient of n^2 is an accurate indicator of which algorithm requires fewer operations.

All aforementioned algorithms can be extended to the nonsymmetric case, although this would be beyond the scope of the present paper, as would be possible extensions to the complex case or to block Toeplitz systems. We also did not consider numerical stability, though we expect the numerical behavior of all methods described here to be similar to that of the Durbin algorithm (see [7]) and the split Levinson algorithm (see [17]).

A substantial part of this paper is devoted to a review of previous algorithms in a unified notation without which the new algorithm we propose would be difficult to explain. The paper is organized as follows. In section 2 we present some basic definitions and results, and then, in section 3, we introduce the classical Durbin and Levinson algorithms. In section 4 the split Levinson algorithms are described, in section 5 the same is done for the even-odd Levinson algorithm, and in section 6 the new method, the even-odd split Levinson algorithm, is presented. Finally, in section 7, we present a summary of the complexities for the different methods.

2. Preliminaries. A symmetric matrix $T_n \in \mathbb{R}^{(n,n)}$ is said to be *Toeplitz* if its elements $(T_n)_{ij}$ satisfy $(T_n)_{ij} = \rho_{|j-i|}$, where $\{\rho_j\}_{j=0}^{n-1}$ are the components of a vector $(\rho_0, t_{n-1})^T \in \mathbb{R}^n$ with $t_{n-1} = (\rho_1, \ldots, \rho_{n-1})^T \in \mathbb{R}^{n-1}$ so that

$$T_n = \begin{pmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_{n-1} \\ \rho_1 & \rho_0 & \rho_1 & \cdots & \rho_{n-2} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \rho_{n-2} & \rho_{n-3} & \rho_{n-4} & \cdots & \rho_1 \\ \rho_{n-1} & \rho_{n-2} & \rho_{n-3} & \cdots & \rho_0 \end{pmatrix}. \tag{2.1}$$

Many early results about such matrices can be found in, e.g., [3], [6], [9], [12], and [18].

The identity matrix is denoted by I throughout this paper, and we will not specifically indicate its dimension, which is assumed to be clear from the context. We denote by J the matrix with ones on its southwest-northeast diagonal and zeros everywhere else (the exchange matrix). As with I, we will not specifically indicate its dimension.

Toeplitz matrices are *persymmetric*, i.e., they are symmetric about their southwest-northeast diagonal. For such a matrix T_n, this is the same as requiring that $JT_n^T J = T_n$. It is easy to see that the inverse of a persymmetric matrix is also persymmetric. A matrix that is both symmetric and persymmetric is called *doubly symmetric*.

An *even* (sometimes also referred to as *symmetric*) vector v is defined as a vector satisfying $Jv = v$, and an *odd* (sometimes called *antisymmetric* or *skew-symmetric*) vector w is defined as one that satisfies $Jw = -w$.

Throughout this paper, it will be assumed that the matrix T_n is strongly nonsingular, i.e., that its principal submatrices T_k, $k = 1, 2, \ldots, n$, are all nonsingular.

3. The Durbin and Levinson algorithms. For the sake of convenience, we now briefly present both Durbin's and Levinson's algorithms, as the rest of this paper builds heavily on these two methods. At the same time, it allows us to introduce the notation we will need later on.

3.1. Durbin's algorithm. The *Yule–Walker* equations we referred to in the introduction are given by $T_n y^{(n)} = -t_n$, where T_n is as in (2.1) and $t_k = (\rho_1, \ldots, \rho_k)^T$. Durbin's algorithm solves this system by recursively computing the solutions to lower-dimensional systems. Let us now describe a basic step of Durbin's algorithm, while referring to [13, pp. 194–196] for full details. Assuming that the solution to $T_{k-1}y^{(k-1)} = -t_{k-1}$ is available, the algorithm computes the solution to $T_k y^{(k)} = -t_k$ as follows. Compute $\bar{y}^{(k-1)}$, by which we denote the first $k-1$ components of $y^{(k)}$, and α_{k-1}, the last component of $y^{(k)}$, from

$$\begin{pmatrix} T_{k-1} & Jt_{k-1} \\ (Jt_{k-1})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \bar{y}^{(k-1)} \\ \alpha_{k-1} \end{pmatrix} = - \begin{pmatrix} t_{k-1} \\ \rho_k \end{pmatrix},$$

which leads to

$$\bar{y}^{(k-1)} = T_{k-1}^{-1}(-t_{k-1} - \alpha_{k-1} J t_{k-1}) = y^{(k-1)} + \alpha_{k-1} J y^{(k-1)} \tag{3.1}$$

and

$$\alpha_{k-1} = -\frac{\rho_k + t_{k-1}^T J y^{(k-1)}}{\rho_0 + t_{k-1}^T y^{(k-1)}}\,. \tag{3.2}$$

In addition, we define, as in [13], $\beta_k = \rho_0 + t_k^T y^{(k)}$. The following recursion then holds (see [13, p. 195]):

$$\beta_k = (1 - \alpha_{k-1}^2)\beta_{k-1}. \tag{3.3}$$

The first step of the method consists of solving a trivial 1×1 system, whereas in the final step $y^{(n)}$ is computed from $y^{(n-1)}$, β_{n-1}, and α_{n-1}. The quantities α_k are called reflection coefficients, or Schur–Szegö parameters.

Complexity. The complexity of this algorithm can be calculated by observing that to obtain $y^{(k)}$ from $y^{(k-1)}$, a scalar product needs to be computed, after which $y^{(k-1)}$ is updated. All together, this requires k additions and k multiplications (not counting a constant number of operations) for each step. This means that the algorithm requires a total of $n^2 + \mathcal{O}(n)$ additions and $n^2 + \mathcal{O}(n)$ multiplications, or $2n^2 + \mathcal{O}(n)$ flops.

3.2. Levinson's algorithm. We now turn to the general right-hand side problem $T_n x^{(n)} = b^{(n)}$, with $b^{(k)} = (b_1, b_2, \ldots, b_k)^T$, and a method for solving it—Levinson's algorithm. It is very similar in structure to Durbin's algorithm, and its basic step is given as follows. Assuming that the solutions to $T_{k-1}x^{(k-1)} = b^{(k-1)}$ and $T_{k-1}y^{(k-1)} = -t_{k-1}$ are available, the algorithm computes the solution to $T_k x^{(k)} = b^{(k)}$ as follows. Compute $\bar{x}^{(k-1)}$, the first $k-1$ components of $x^{(k)}$, and μ_{k-1}, by which we denote the last component of $x^{(k)}$, from

$$\begin{pmatrix} T_{k-1} & Jt_{k-1} \\ (Jt_{k-1})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \bar{x}^{(k-1)} \\ \mu_{k-1} \end{pmatrix} = \begin{pmatrix} b^{(k-1)} \\ b_k \end{pmatrix},$$

which leads to

$$\bar{x}^{(k-1)} = T_{k-1}^{-1}(b^{(k-1)} - \mu_{k-1} Jt_{k-1}) = x^{(k-1)} + \mu_{k-1} Jy^{(k-1)} \tag{3.4}$$

and

$$\mu_{k-1} = \frac{b_k - t_{k-1}^T Jx^{(k-1)}}{\rho_0 + t_{k-1}^T y^{(k-1)}} \,. \tag{3.5}$$

Durbin's algorithm is used "in parallel" for computing the solutions $y^{(k)}$ of the Yule–Walker subsystems.

Complexity. Since for this algorithm it is assumed that the solutions to the Yule–Walker subsystems are already available from Durbin's algorithm, its complexity is obtained by noting that, to calculate $x^{(k)}$ from $x^{(k-1)}$, a scalar product needs to be computed, after which $x^{(k-1)}$ is updated. All together, this requires k additions and k multiplications (not counting a constant number of operations) per step. Taking into account the complexity of Durbin's algorithm, this means that the algorithm requires a total of $2n^2 + \mathcal{O}(n)$ additions and $2n^2 + \mathcal{O}(n)$ multiplications, or $4n^2 + \mathcal{O}(n)$ flops.

4. The split Durbin and split Levinson algorithms. The "split Levinson" algorithms for the Yule–Walker equations and for the general case were introduced in [10] and [11], respectively. We present those algorithms in quite some detail because we will need those details later on and because the different notation used in the aforementioned references makes it difficult to interpret the results in our own notation. We stress that we will merely rewrite known results from [10] and [11] in a notation that is more useful for us. However, in the spirit of [13], we will call the split Levinson algorithm for Yule–Walker equations from [10] the "split Durbin algorithm" to distinguish it from the split Levinson algorithm for the general right-hand side case in [11].

4.1. The split Durbin algorithm for the Yule–Walker equations. We start with the split Levinson algorithm from [10] for the Yule–Walker equations $T_n y^{(n)} = -t_n$, from now on called the "split Durbin" algorithm. Defining an *even* solution $u^{(k)}$ of these equations as the solution of $T_k u^{(k)} = -(t_k + Jt_k)$, or $u^{(k)} = y^{(k)} + Jy^{(k)}$, and an *odd* solution as the solution of $T_k v^{(k)} = -(t_k - Jt_k)$, or $v^{(k)} = y^{(k)} - Jy^{(k)}$, this algorithm is based on the remarkable observation that the solution $y^{(k)}$ can be written either as a combination of the two successive even solutions $u^{(k)}$ and $u^{(k-1)}$ or as a combination of the two successive odd solutions $v^{(k)}$ and $v^{(k-1)}$. It is therefore sufficient to compute either the even or the odd solutions. As we shall see, this can be achieved with fewer operations than Durbin's algorithm.

In what follows, we concentrate on the even solutions and refer to [10] for the corresponding (and almost entirely analogous) results for the odd solutions.

We begin by showing the relation between $y^{(k)}$ and the even solutions $u^{(k)}$ and $u^{(k-1)}$. We recall that the first $k-1$ components of $y^{(k)}$ were defined as $\bar{y}^{(k-1)}$, i.e., $y^{(k)} = ((\bar{y}^{(k-1)})^T, \alpha_{k-1})^T$, where α_{k-1} is defined by (3.2). We then have with (3.1)

$$\begin{aligned} \bar{y}^{(k-1)} + J\bar{y}^{(k-1)} &= (y^{(k-1)} + \alpha_{k-1} Jy^{(k-1)}) + (Jy^{(k-1)} + \alpha_{k-1} y^{(k-1)}) \\ &= (1+\alpha_{k-1})(y^{(k-1)} + Jy^{(k-1)}) \\ &= (1+\alpha_{k-1}) u^{(k-1)}. \end{aligned} \tag{4.1}$$

We can now use (4.1) to determine $y^{(k)}$ from $u^{(k)}$ and $u^{(k-1)}$:

$$\begin{aligned} u^{(k)} = y^{(k)} + Jy^{(k)} &= \begin{pmatrix} \bar{y}^{(k-1)} \\ \alpha_{k-1} \end{pmatrix} + \begin{pmatrix} \alpha_{k-1} \\ J\bar{y}^{(k-1)} \end{pmatrix} \\ &= \begin{pmatrix} \bar{y}^{(k-1)} \\ \alpha_{k-1} \end{pmatrix} + \begin{pmatrix} \alpha_{k-1} \\ -\bar{y}^{(k-1)} + (1+\alpha_{k-1}) u^{(k-1)} \end{pmatrix} \end{aligned}$$

or

$$\begin{aligned} u_1^{(k)} = u_k^{(k)} &= \bar{y}_1^{(k-1)} + \alpha_{k-1}, \\ u_j^{(k)} &= \bar{y}_j^{(k-1)} - \bar{y}_{j-1}^{(k-1)} + (1+\alpha_{k-1}) u_{j-1}^{(k-1)} \quad (2 \le j \le k-1). \end{aligned}$$

Taking into account the definitions of $\bar{y}^{(k-1)}$ and α_{k-1}, we obtain $y^{(k)}$ in terms of $u^{(k)}$ and $u^{(k-1)}$:

$$y_1^{(k)} = u_1^{(k)} - \alpha_{k-1}, \tag{4.2}$$

$$y_j^{(k)} = y_{j-1}^{(k)} + u_j^{(k)} - (1+\alpha_{k-1}) u_{j-1}^{(k-1)} \quad (2 \le j \le k). \tag{4.3}$$

It is therefore possible to obtain $y^{(n)}$ by computing only the even solutions. Equations (4.2) and (4.3) correspond to (14) in [10]. The odd versions of these results are represented by (26) in [10].

Of course, a recursive algorithm which computes even solutions needs to be able to express even solutions in terms of previous even solutions. Let us now show that this is indeed possible. We start by observing that for $2 \le j \le k-1$

$$y_j^{(k)} = y_j^{(k-1)} + \alpha_{k-1} (Jy^{(k-1)})_j = y_j^{(k-1)} + \alpha_{k-1} y_{k-j}^{(k-1)}, \tag{4.4}$$

and therefore

$$y_{j-1}^{(k)} = y_{j-1}^{(k-1)} + \alpha_{k-1} y_{k-j+1}^{(k-1)}. \tag{4.5}$$

Combining (4.4) and (4.5) with (4.3) yields for $2 \le j \le k-1$

$$\begin{aligned} y_j^{(k)} - y_{j-1}^{(k)} &= \left(y_j^{(k-1)} - y_{j-1}^{(k-1)}\right) - \alpha_{k-1}\left(y_{k-j+1}^{(k-1)} - y_{k-j}^{(k-1)}\right) \\ &= \left(u_j^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)}\right) - \alpha_{k-1}\left(u_{k-j+1}^{(k-1)} - (1+\alpha_{k-2})u_{k-j}^{(k-2)}\right) \\ &= u_j^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)} - \alpha_{k-1}\left(u_{j-1}^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)}\right). \end{aligned}$$

With (4.3), this gives for $2 \le j \le k-1$

$$\begin{aligned} u_j^{(k)} = (1+\alpha_{k-1})u_{j-1}^{(k-1)} + u_j^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)} \\ - \alpha_{k-1}\left(u_{j-1}^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)}\right), \end{aligned}$$

and therefore

$$u_j^{(k)} = u_j^{(k-1)} + u_{j-1}^{(k-1)} + (\alpha_{k-1}-1)(1+\alpha_{k-2})u_{j-1}^{(k-2)}. \tag{4.6}$$

For the first and last components of $u^{(k)}$, we start from (4.3) for $j = k$. This yields

$$\begin{aligned} u_k^{(k)} &= y_k^{(k)} - y_{k-1}^{(k)} + (1+\alpha_{k-1})u_{k-1}^{(k-1)} \\ &= y_k^{(k)} - \left(y_{k-1}^{(k-1)} + \alpha_{k-1}(Jy^{(k-1)})_{k-1}\right) + (1+\alpha_{k-1})u_{k-1}^{(k-1)} \\ &= y_k^{(k)} - \left(y_{k-1}^{(k-1)} + \alpha_{k-1}u_{k-1}^{(k-1)} - \alpha_{k-1}y_{k-1}^{(k-1)}\right) + (1+\alpha_{k-1})u_{k-1}^{(k-1)} \\ &= y_k^{(k)} + (\alpha_{k-1}-1)y_{k-1}^{(k-1)} + u_{k-1}^{(k-1)}. \end{aligned}$$

Because of the definition of α_k and because $u^{(k)}$ is an even vector, we can write

$$u_1^{(k)} = u_k^{(k)} = u_1^{(k-1)} + (\alpha_{k-1}-1)(1+\alpha_{k-2}) + 1. \tag{4.7}$$

We note that (4.6) and (4.7) correspond to (17) in [10], and the odd counterpart is given by (25) in [10]. Expressions (4.6) and (4.7) form a three-term recurrence relation, expressing $u^{(k)}$ in terms of $u^{(k-1)}$ and $u^{(k-2)}$. Together with (4.2) and (4.3), they are the essence of the split Durbin algorithm for the Yule–Walker equations.

To make the algorithm practical, it should be possible to express the quantity $(\alpha_{k-1}-1)(1+\alpha_{k-2})$ in terms of known "even" quantities. To that effect, we consider the following:

$$\begin{aligned} t_k^T u^{(k)} &= t_k^T\left(y^{(k)} + Jy^{(k)}\right) \\ &= \left(\rho_0 + t_k^T y^{(k)}\right) + \left(\rho_{k+1} + t_k^T Jy^{(k)}\right) - (\rho_0 + \rho_{k+1}) \\ &= \beta_k + (-\alpha_k\beta_k) - (\rho_0 + \rho_{k+1}) \end{aligned}$$

or

$$\rho_0 + t_k^T u^{(k)} + \rho_{k+1} = (1-\alpha_k)\beta_k. \tag{4.8}$$

Using (4.8) and recalling that $\beta_k = \left(1-\alpha_{k-1}^2\right)\beta_{k-1}$, we therefore have

$$(1-\alpha_{k-1})(1+\alpha_{k-2}) = \frac{(1-\alpha_{k-1})\beta_{k-1}}{\beta_{k-1}} \cdot \frac{(1-\alpha_{k-2}^2)\beta_{k-2}}{(1-\alpha_{k-2})\beta_{k-2}}$$

$$= \frac{(1-\alpha_{k-1})\beta_{k-1}}{\beta_{k-1}} \cdot \frac{\beta_{k-1}}{(1-\alpha_{k-2})\beta_{k-2}}$$
$$= \frac{\rho_0 + t_{k-1}^T u^{(k-1)} + \rho_k}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}} \cdot$$

As one can see, the right-hand side contains only "even" quantities, which are easily computed. This is the first expression in (19) in [10]; the second one represents its odd counterpart. That the denominator in the last expression can never be zero because of the strong nonsingularity assumptions on the coefficient matrix was also explicitly shown in [20, bottom of p. 148].

For the even solutions, the reflection coefficients α_k can also be obtained from (4.2) and (4.3), as follows:

$$y_1^{(k)} + \sum_{j=2}^{k} \left(y_j^{(k)} - y_{j-1}^{(k)} \right) = \sum_{j=1}^{k} u_j^{(k)} - (1+\alpha_{k-1}) \sum_{j=1}^{k-1} u_j^{(k-1)} - \alpha_{k-1},$$

where we have used the fact that $y_k^{(k)} = \alpha_{k-1}$. Since the left-hand side telescopes into $y_k^{(k)}$, solving for α_{k-1} then yields

$$\alpha_{k-1} = \frac{\sum_{j=1}^{k} u_j^{(k)} - \sum_{j=1}^{k-1} u_j^{(k-1)}}{\sum_{j=1}^{k-1} u_j^{(k-1)} + 2} \cdot \tag{4.9}$$

This expression, which corresponds to (23) in [10], can be shown to be well defined under the strong nonsingularity of T_n. It has no odd analogue: in the odd case the reflection coefficients α_k must be computed recursively (see (18) in [10]).

We summarize this algorithm for the even solutions (the odd case is analogous) as follows.

Given $u^{(k-2)}$ and $u^{(k-1)}$, *compute* $u^{(k)}$ for $k = 3, \ldots, n$ from

$$u_1^{(k)} = u_k^{(k)} = u_1^{(k-1)} - \frac{\rho_0 + t_{k-1}^T u^{(k-1)} + \rho_k}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}} + 1,$$
$$u_j^{(k)} = u_j^{(k-1)} + u_{j-1}^{(k-1)} - \left(\frac{\rho_0 + t_{k-1}^T u^{(k-1)} + \rho_k}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}} \right) u_{j-1}^{(k-2)} \quad (2 \le j \le k-1).$$

After this, compute α_{n-1} from (4.9) and $y^{(n)}$ from

$$y_1^{(n)} = u_1^{(n)} - \alpha_{n-1},$$
$$y_j^{(n)} = y_{j-1}^{(n)} + u_j^{(n)} - (1+\alpha_{n-1}) u_{j-1}^{(n-1)} \quad (2 \le j \le n).$$

All necessary quantities for $k = 1, 2$ are easily computed before starting the algorithm. As was mentioned before, the reflection coefficients α_k can also be computed recursively (see (18) in [10]).

Of course, one needs only to compute roughly half of the components of the even or odd vectors, and this was taken into consideration in the following complexity calculations.

Complexity. Taking into account the fact that an even vector is determined by roughly half of its components, the number of operations required to compute $u^{(k)}$

from $u^{(k-1)}$ and $u^{(k-2)}$ is given by $2k$ additions and k multiplications for each step (once again, not counting a constant number of operations), which stem from the computation of one scalar product and the update. The operation count for the odd case is identical. The total number of operations is therefore $n^2 + \mathcal{O}(n)$ additions and $\frac{1}{2}n^2 + \mathcal{O}(n)$ multiplications, or $\frac{3}{2}n^2 + \mathcal{O}(n)$ flops.

4.2. The split Levinson algorithm for the general case. We now turn to the general right-hand side problem $T_n x^{(n)} = b^{(n)}$, where $b^{(k)} = (b_1, \ldots, b_k)^T$. We define the even vectors $s^{(k)} = b^{(k)} + Jb^{(k)}$ for $k = 1, \ldots, n$. We also define an *even* solution $w^{(k)}$ as the solution of $T_k w^{(k)} = s^{(k)}$ or $w^{(k)} = x^{(k)} + Jx^{(k)}$. As in the case of the Yule–Walker equations, the solution $x^{(k)}$ can be written in terms of the two successive even solutions $w^{(k)}$ and $w^{(k-1)}$. It is therefore sufficient, once again, to compute only the even solutions, which requires fewer operations than Levinson's algorithm. Analogous results exist for the odd solutions, but we will discuss only the even case as this is the case we will need in section 6. We begin by showing the relation between the general solution and successive even solutions.

Defining the first $k-1$ components of $x^{(k)}$ as $\bar{x}^{(k-1)}$, i.e., $x^{(k)} = ((\bar{x}^{(k-1)})^T, \mu_{k-1})^T$, we obtain from (3.4)

$$\begin{aligned} \bar{x}^{(k-1)} + J\bar{x}^{(k-1)} &= \left(x^{(k-1)} + \mu_{k-1} J y^{(k-1)}\right) + \left(Jx^{(k-1)} + \mu_{k-1} y^{(k-1)}\right) \\ &= w^{(k-1)} + \mu_{k-1} u^{(k-1)}. \end{aligned} \tag{4.10}$$

We now use (4.10) to determine $x^{(k)}$ from $w^{(k)}$ and $w^{(k-1)}$:

$$\begin{aligned} w^{(k)} = x^{(k)} + Jx^{(k)} &= \begin{pmatrix} \bar{x}^{(k-1)} \\ \mu_{k-1} \end{pmatrix} + \begin{pmatrix} \mu_{k-1} \\ J\bar{x}^{(k-1)} \end{pmatrix} \\ &= \begin{pmatrix} \bar{x}^{(k-1)} \\ \mu_{k-1} \end{pmatrix} + \begin{pmatrix} \mu_{k-1} \\ -\bar{x}^{(k-1)} + w^{(k-1)} + \mu_{k-1} u^{(k-1)} \end{pmatrix} \end{aligned}$$

or

$$\begin{aligned} w_1^{(k)} = w_k^{(k)} &= \bar{x}_1^{(k-1)} + \mu_{k-1}, \\ w_j^{(k)} &= \bar{x}_j^{(k-1)} - \bar{x}_{j-1}^{(k-1)} + w_{j-1}^{(k-1)} + \mu_{k-1} u_{j-1}^{(k-1)} \quad (2 \le j \le k-1). \end{aligned}$$

Taking into account the definitions of $\bar{x}^{(k)}$ and μ_{k-1}, these last two expressions yield $x^{(k)}$ in terms of $w^{(k)}$, $w^{(k-1)}$, and $u^{(k-1)}$:

$$x_1^{(k)} = w_1^{(k)} - \mu_{k-1}, \tag{4.11}$$

$$x_j^{(k)} = x_{j-1}^{(k)} + w_j^{(k)} - w_{j-1}^{(k-1)} - \mu_{k-1} u_{j-1}^{(k-1)} \quad (2 \le j \le k). \tag{4.12}$$

Once again, to be able to construct a useful recursive algorithm for a system with an even right-hand side, we must be able to express even solutions in terms of previous even solutions. We begin with the observation that, for $2 \le j \le k-1$,

$$x_j^{(k)} = x_j^{(k-1)} + \mu_{k-1} (J y^{(k-1)})_j = x_j^{(k-1)} + \mu_{k-1} y_{k-j}^{(k-1)}, \tag{4.13}$$

and therefore

$$x_{j-1}^{(k)} = x_{j-1}^{(k-1)} + \mu_{k-1} y_{k-j+1}^{(k-1)}. \tag{4.14}$$

Combining (4.13) and (4.14) yields for $2 \le j \le k-1$

$$x_j^{(k)} - x_{j-1}^{(k)} = \left(x_j^{(k-1)} - x_{j-1}^{(k-1)}\right) - \mu_{k-1}\left(y_{k-j+1}^{(k-1)} - y_{k-j}^{(k-1)}\right).$$

With (4.3) and (4.12) this becomes

$$x_j^{(k)} - x_{j-1}^{(k)} = \left(w_j^{(k-1)} - w_{j-1}^{(k-2)} - \mu_{k-2}u_{j-1}^{(k-2)}\right) - \mu_{k-1}\left(u_{j-1}^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)}\right).$$

With the help of (4.12), we then obtain for $2 \le j \le k-1$

$$\begin{aligned} w_j^{(k)} = w_{j-1}^{(k-1)} &+ \mu_{k-1}u_{j-1}^{(k-1)} + \left(w_j^{(k-1)} - w_{j-1}^{(k-2)} - \mu_{k-2}u_{j-1}^{(k-2)}\right) \\ &- \mu_{k-1}\left(u_{j-1}^{(k-1)} - (1+\alpha_{k-2})u_{j-1}^{(k-2)}\right), \end{aligned}$$

and therefore

$$w_j^{(k)} = w_j^{(k-1)} + w_{j-1}^{(k-1)} - w_{j-1}^{(k-2)} + \left(\mu_{k-1}(1+\alpha_{k-2}) - \mu_{k-2}\right)u_{j-1}^{(k-2)}. \tag{4.15}$$

For the first and last components of $w^{(k)}$, we start from (4.12) for $j = k$. This gives

$$\begin{aligned} w_k^{(k)} &= x_k^{(k)} - x_{k-1}^{(k)} + w_{k-1}^{(k-1)} + \mu_{k-1}u_{k-1}^{(k-1)} \\ &= x_k^{(k)} - \left(x_{k-1}^{(k-1)} + \mu_{k-1}(Jy^{(k-1)})_{k-1}\right) + w_{k-1}^{(k-1)} + \mu_{k-1}\left(y_{k-1}^{(k-1)} + (Jy^{(k-1)})_{k-1}\right) \\ &= x_k^{(k)} - x_{k-1}^{(k-1)} + w_{k-1}^{(k-1)} + \mu_{k-1}y_{k-1}^{(k-1)}. \end{aligned}$$

Because of the definition of μ_k and because $w^{(k)}$ is an even vector, we can thus write

$$w_1^{(k)} = w_k^{(k)} = w_1^{(k-1)} + \left(\mu_{k-1}(1+\alpha_{k-2}) - \mu_{k-2}\right). \tag{4.16}$$

Equations (4.15) and (4.16) correspond to (28) in [11]. These expressions form a recurrence relation, expressing $w^{(k)}$ in terms of $w^{(k-1)}$, $w^{(k-2)}$, and $u^{(k-2)}$. Together with (4.11) and (4.12), they are the essence of the split Levinson algorithm for arbitrary right-hand side.

To make this a working algorithm, we will now express $(\mu_{k-1}(1+\alpha_{k-2}) - \mu_{k-2})$ in terms of known "even" quantities. For that purpose, we consider the following for $1 \le k \le n-1$:

$$\begin{aligned} t_k^T w^{(k)} &= t_k^T\left(x^{(k)} + Jx^{(k)}\right) \\ &= t_k^T x^{(k)} - \left(b_{k+1} - t_k^T Jx^{(k)}\right) + b_{k+1}, \end{aligned}$$

and therefore

$$t_k^T x^{(k)} = t_k^T w^{(k)} + \mu_k\beta_k - b_{k+1}. \tag{4.17}$$

On the other hand,

$$\begin{aligned} t_k^T x^{(k)} &= t_{k-1}^T \bar{x}^{(k-1)} + \rho_k x_k^{(k)} \\ &= t_{k-1}^T x^{(k-1)} + \mu_{k-1}t_{k-1}^T Jy^{(k-1)} + \rho_k\mu_{k-1} \\ &= t_{k-1}^T x^{(k-1)} + \mu_{k-1}\left(\rho_k + t_{k-1}^T Jy^{(k-1)}\right) \\ &= t_{k-1}^T x^{(k-1)} - \mu_{k-1}\alpha_{k-1}\beta_{k-1}, \end{aligned}$$

where α_{k-1} and β_{k-1} are as defined in Durbin's algorithm. Applying (4.17) with $k-1$ instead of k, this becomes

$$\begin{aligned} t_k^T x^{(k)} &= t_{k-1}^T w^{(k-1)} + \mu_{k-1}\beta_{k-1} - b_k - \mu_{k-1}\alpha_{k-1}\beta_{k-1} \\ &= t_{k-1}^T w^{(k-1)} + (1-\alpha_{k-1})\beta_{k-1}\mu_{k-1} - b_k. \end{aligned} \tag{4.18}$$

Combining (4.17) with (4.18), we obtain

$$t_k^T w^{(k)} + \mu_k\beta_k - b_{k+1} = t_{k-1}^T w^{(k-1)} + (1-\alpha_{k-1})\beta_{k-1}\mu_{k-1} - b_k,$$

which then yields

$$\mu_{k-1}\beta_{k-1}(1-\alpha_{k-1}) - \mu_k\beta_k = t_k^T w^{(k)} - t_{k-1}^T w^{(k-1)} - b_{k+1} + b_k. \tag{4.19}$$

With $\beta_k = (1-\alpha_{k-1}^2)\beta_{k-1}$, the left-hand side in (4.19) becomes

$$\begin{aligned} \mu_{k-1}\beta_{k-1}(1-\alpha_{k-1}) - \mu_k\beta_k &= (1-\alpha_{k-1})\beta_{k-1}\left(\mu_{k-1} - \mu_k\frac{\beta_k}{(1-\alpha_{k-1})\beta_{k-1}}\right) \\ &= (1-\alpha_{k-1})\beta_{k-1}\left(\mu_{k-1} - \mu_k\frac{(1-\alpha_{k-1}^2)\beta_{k-1}}{(1-\alpha_{k-1})\beta_{k-1}}\right) \\ &= (1-\alpha_{k-1})\beta_{k-1}\left(\mu_{k-1} - \mu_k(1+\alpha_{k-1})\right). \end{aligned}$$

We recall from (4.8) that $(1-\alpha_{k-1})\beta_{k-1} = \rho_0 + t_{k-1}^T u^{(k-1)} + \rho_k$, which, when substituted back into (4.19) for $k-1$ instead of k, finally leads to

$$\mu_{k-2} - \mu_{k-1}(1+\alpha_{k-2}) = \frac{t_{k-1}^T w^{(k-1)} - t_{k-2}^T w^{(k-2)} - b_k + b_{k-1}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}}\ . \tag{4.20}$$

The right-hand side now contains only computable "even" quantities. The expression in (4.20) corresponds to (23) in [11].

The reflection coefficients μ_k can be obtained from (4.11) and (4.12) as follows:

$$x_1^{(k)} + \sum_{j=2}^{k}\left(x_j^{(k)} - x_{j-1}^{(k)}\right) = \sum_{j=1}^{k} w_j^{(k)} - \sum_{j=1}^{k-1} w_j^{(k-1)} - \mu_{k-1}\sum_{j=1}^{k-1} u_j^{(k-1)} - \mu_{k-1},$$

where we have used the fact that $x_k^{(k)} = \mu_{k-1}$. Since the left-hand side telescopes into $x_k^{(k)}$, solving for μ_{k-1} then yields

$$\mu_{k-1} = \frac{\sum_{j=1}^{k} w_j^{(k)} - \sum_{j=1}^{k-1} w_j^{(k-1)}}{2 + \sum_{j=1}^{k-1} u_j^{(k-1)}}\ . \tag{4.21}$$

This expression is, once again, well defined, and corresponds to (16) in [11].

The algorithm can be summarized in the following way. Given $w^{(k-2)}$, $w^{(k-1)}$, $u^{(k-2)}$, and $u^{(k-1)}$, *compute* for $k = 3, \ldots, n$, $u^{(k)}$ by using the split Durbin algorithm, and compute $w^{(k)}$ from

$$w_1^{(k)} = w_k^{(k)} = w_1^{(k-1)} - \frac{t_{k-1}^T w^{(k-1)} - t_{k-2}^T w^{(k-2)} - b_k + b_{k-1}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}},$$

$$w_j^{(k)} = w_j^{(k-1)} + w_{j-1}^{(k-1)} - w_{j-1}^{(k-2)} - \left(\frac{t_{k-1}^T w^{(k-1)} - t_{k-2}^T w^{(k-2)} - b_k + b_{k-1}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}}\right) u_{j-1}^{(k-2)}$$
$$(2 \le j \le k-1).$$

After this, compute μ_{n-1} from (4.21), and compute $x^{(n)}$ from

$$x_1^{(n)} = w_1^{(n)} - \mu_{n-1},$$
$$x_j^{(n)} = x_{j-1}^{(n)} + w_j^{(n)} - w_{j-1}^{(n-1)} - \mu_{n-1} u_{j-1}^{(n-1)} \quad (2 \le j \le n).$$

All necessary quantities for $k = 1, 2$ are easily computed before starting the algorithm. The reflection coefficients μ_k can also be computed recursively (see (26) in [11]).

Complexity. Since for this algorithm it is assumed that the solutions to the even Yule–Walker subsystems are already available from the split Durbin algorithm in [10], its complexity is determined by an additional scalar product per step and the update from $w^{(k-1)}$ and $w^{(k-2)}$ to $w^{(k)}$. All this requires $2k$ additions and k multiplications (not including a constant number of operations) for each step. Taking into account the complexity of the split Durbin's algorithm for the Yule–Walker equations, this means that the algorithm requires a total of $2n^2 + \mathcal{O}(n)$ additions and $n^2 + \mathcal{O}(n)$ multiplications, or $3n^2 + \mathcal{O}(n)$ flops.

5. The even-odd Levinson algorithm. All previous algorithms are based on recursions, derived from the following partition of T_k:

$$\begin{pmatrix} T_{k-1} & Jt_{k-1} \\ (Jt_{k-1})^T & \rho_0 \end{pmatrix}.$$

However, such a partition is rather ill-suited for a matrix with all the symmetry properties of a Toeplitz matrix. In [20] an algorithm was proposed for the general right-hand side problem $T_n x^{(n)} = b^{(n)}$, based on a recursion derived from a different and more appropriate partition, as we will see below. This algorithm, which we will refer to as the "even-odd Levinson algorithm," needs the solutions to the Yule–Walker equations, just like Levinson's algorithm and the split Levinson algorithm. In this method, these are provided by Durbin's algorithm. We summarize the basic properties of this algorithm, and we refer the reader to [20] for the precise details.

We recall that $s^{(k)} = b^{(k)} + Jb^{(k)}$, $w^{(k)} = T_k^{-1} s^{(k)}$ and define $a^{(k)} = b^{(k)} - Jb^{(k)}$ and $z^{(k)} = T_k^{-1} a^{(k)}$. With $\lfloor \xi \rfloor$ denoting the integer part of ξ when ξ is a nonnegative number, we also define the even vectors $p^{(k)} = (s^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1}, \ldots, s^{(n)}_{\lfloor \frac{n+k}{2} \rfloor})^T$ so that $p^{(n)} = s^{(n)}$, and $h^{(k)} = T_k^{-1} p^{(k)}$ so that $h^{(n)} = w^{(n)}$. Analogously, we define the odd vectors $r^{(k)} = (a^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1}, \ldots, a^{(n)}_{\lfloor \frac{n+k}{2} \rfloor})^T$, and $g^{(k)} = T_k^{-1} r^{(k)}$. We have $r^{(n)} = a^{(n)}$ and $g^{(n)} = z^{(n)}$. By their construction, the vectors $s^{(n)}$ and $a^{(n)}$ satisfy

$$s^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} = s^{(n)}_{\lfloor \frac{n+k}{2} \rfloor} \quad \text{and} \quad a^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} = -a^{(n)}_{\lfloor \frac{n+k}{2} \rfloor}.$$

We note that when $k \neq n$, then, in general, $p^{(k)} \neq s^{(k)}$, $r^{(k)} \neq a^{(k)}$, $h^{(k)} \neq w^{(k)}$, and $g^{(k)} \neq z^{(k)}$.

The even-odd Levinson algorithm computes the solutions of $T_k h^{(k)} = p^{(k)}$ and $T_k g^{(k)} = r^{(k)}$ for $k = 2, 4, \ldots, n$ when n is even and for $k = 1, 3, \ldots, n$ when n is odd. Let us now describe its basic step, referring to [20] for the details.

Assuming that the solutions of $T_{k-2} h^{(k-2)} = p^{(k-2)}$ and $T_{k-2} g^{(k-2)} = r^{(k-2)}$ are available, along with the solution of $T_{k-2} y^{(k-2)} = -t_{k-2}$, this method computes $h^{(k)}$ and $g^{(k)}$ from

$$\begin{pmatrix} \rho_0 & t_{k-2}^T & \rho_{k-1} \\ t_{k-2} & T_{k-2} & Jt_{k-2} \\ \rho_{k-1} & (Jt_{k-2})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \lambda_{k-2} \\ \bar{h}^{(k-2)} \\ \lambda_{k-2} \end{pmatrix} = \begin{pmatrix} s^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} \\ p^{(k-2)} \\ s^{(n)}_{\lfloor \frac{n+k}{2} \rfloor} \end{pmatrix} \tag{5.1}$$

and

$$\begin{pmatrix} \rho_0 & t_{k-2}^T & \rho_{k-1} \\ t_{k-2} & T_{k-2} & Jt_{k-2} \\ \rho_{k-1} & (Jt_{k-2})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \theta_{k-2} \\ \bar{g}^{(k-2)} \\ -\theta_{k-2} \end{pmatrix} = \begin{pmatrix} a^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} \\ r^{(k-2)} \\ a^{(n)}_{\lfloor \frac{n+k}{2} \rfloor} \end{pmatrix}, \tag{5.2}$$

where $\bar{h}^{(k-2)}$ and $\bar{g}^{(k-2)}$ are vectors containing the $k-2$ middle components of $h^{(k)}$ and $g^{(k)}$, respectively, and λ_{k-2} and θ_{k-2} are the first components of $h^{(k)}$ and $g^{(k)}$, respectively (their last components are λ_{k-2} and $-\theta_{k-2}$, respectively). This leads to

$$\bar{h}^{(k-2)} = T_{k-2}^{-1} \left(p^{(k-2)} - \lambda_{k-2}(t_{k-2} + Jt_{k-2}) \right) = h^{(k-2)} + \lambda_{k-2} u^{(k-2)},$$
$$\bar{g}^{(k-2)} = T_{k-2}^{-1} \left(r^{(k-2)} - \theta_{k-2}(t_{k-2} - Jt_{k-2}) \right) = g^{(k-2)} + \theta_{k-2} v^{(k-2)},$$

and

$$\lambda_{k-2} = \frac{s^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} - t_{k-2}^T h^{(k-2)}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}}, \qquad \theta_{k-2} = \frac{a^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1} - t_{k-2}^T g^{(k-2)}}{\rho_0 + t_{k-2}^T v^{(k-2)} - \rho_{k-1}}.$$

We recall that $u^{(k)}$ and $v^{(k)}$ were defined as $u^{(k)} = y^{(k)} + Jy^{(k)}$ and $v^{(k)} = y^{(k)} - Jy^{(k)}$. The algorithm starts with trivial 1×1 or 2×2 systems, depending on whether the dimension n is odd or even, respectively. The even and odd solutions to the Yule–Walker equations are obtained from Durbin's algorithm. The algorithm ends with the even and odd solutions $h^{(n)} = w^{(n)}$ and $g^{(n)} = z^{(n)}$, respectively, after which the solution $x^{(n)}$ is computed as $\frac{1}{2}\left(h^{(n)} + g^{(n)}\right)$.

Complexity. Since it is assumed that the even and odd solutions are formed from the solutions of the Yule–Walker subsystems, provided by Durbin's algorithm, the complexity is determined by an additional two scalar products and two updates per step. This requires $4k$ additions and $2k$ multiplications, not including a constant number of operations, per step. Since the recursion for this part of the algorithm progresses in steps of two, this means that a total of $n^2 + \mathcal{O}(n)$ additions and $\frac{1}{2}n^2 + \mathcal{O}(n)$ multiplications should be added to Durbin's algorithm's complexity, bringing the total number of operations to $2n^2 + \mathcal{O}(n)$ additions and $\frac{3}{2}n^2 + \mathcal{O}(n)$ multiplications or $\frac{7}{2}n^2 + \mathcal{O}(n)$ flops.

6. The even-odd split Levinson algorithm. In this section we combine the split Durbin algorithm with the even-odd Levinson algorithm into a new method for the general right-hand side problem $T_n x^{(n)} = b^{(n)}$, which we will call the "even-odd split Levinson algorithm." We saw in section 4.2 that to compute $x^{(n)}$ it is sufficient to have available $w^{(n)}$, $w^{(n-1)}$, and $u^{(n-1)}$.

We now propose to use the even-odd Levinson algorithm to compute the even solutions $w^{(n)}$ and $w^{(n-2)}$, while the even solutions to the Yule–Walker equations are provided by the split Durbin algorithm from [10], which was summarized in section 4.1. The even solutions $w^{(n)}$ and $w^{(n-2)}$ are then used to find $w^{(n-1)}$ from (4.15) and (4.16), after which the solution $x^{(n)}$ is computed from (4.11) and (4.12). We consider this in detail after describing the recursive structure of the algorithm.

Before proceeding with the latter, we define

$$p^{(k)} = \left(s^{(n)}_{\lfloor \frac{n-k}{2} \rfloor + 1}, \ldots, s^{(n)}_{\lfloor \frac{n+k}{2} \rfloor} \right)^T,$$

$$q^{(k)} = \left(s^{(n-2)}_{\lfloor\frac{n-k}{2}\rfloor}, \ldots, s^{(n-2)}_{\lfloor\frac{n+k}{2}\rfloor-1}\right)^T,$$
$$h_p^{(k)} = T_k^{-1} p^{(k)},$$
$$h_q^{(k)} = T_k^{-1} q^{(k)}.$$

This means that $p^{(n)} = s^{(n)}$, $q^{(n-2)} = s^{(n-2)}$, $h_p^{(n)} = w^{(n)}$, and $h_q^{(n-2)} = w^{(n-2)}$. We also define $\bar{h}_p^{(k-2)}$ and $\bar{h}_q^{(k-2)}$ as the vectors containing the middle $k-2$ components of $h_p^{(k)}$ and $h_q^{(k)}$, respectively, whereas λ_{k-2}^p and λ_{k-2}^q are the first (and therefore also last) components of $h_p^{(k)}$ and $h_q^{(k)}$, respectively.

Assuming that $h_p^{(k-2)}$, $h_q^{(k-2)}$, and $u^{(k-2)}$ are available, the basic step is given as follows. For $k = 4, 6, \ldots, n-2$ (n even), or $k = 3, 5, 7, \ldots, n-2$ (n odd), compute the solution of

$$\begin{pmatrix} \rho_0 & t_{k-2}^T & \rho_{k-1} \\ t_{k-2} & T_{k-2} & Jt_{k-2} \\ \rho_{k-1} & (Jt_{k-2})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \lambda_{k-2}^p \\ \bar{h}_p^{(k-2)} \\ \lambda_{k-2}^p \end{pmatrix} = \begin{pmatrix} s^{(n)}_{\lfloor\frac{n-k}{2}\rfloor+1} \\ p^{(k-2)} \\ s^{(n)}_{\lfloor\frac{n+k}{2}\rfloor} \end{pmatrix}, \tag{6.1}$$

which, as in the even-odd Levinson algorithm, leads to

$$\bar{h}_p^{(k-2)} = T_{k-2}^{-1}\left(p^{(k-2)} - \lambda_{k-2}^p(t_{k-2} + Jt_{k-2})\right) = h_p^{(k-2)} + \lambda_{k-2}^p u^{(k-2)} \tag{6.2}$$

and

$$\lambda_{k-2}^p = \frac{s^{(n)}_{\lfloor\frac{n-k}{2}\rfloor+1} - t_{k-2}^T h_p^{(k-2)}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}}\,. \tag{6.3}$$

In addition, compute

$$\begin{pmatrix} \rho_0 & t_{k-2}^T & \rho_{k-1} \\ t_{k-2} & T_{k-2} & Jt_{k-2} \\ \rho_{k-1} & (Jt_{k-2})^T & \rho_0 \end{pmatrix} \begin{pmatrix} \lambda_{k-2}^q \\ \bar{h}_q^{(k-2)} \\ \lambda_{k-2}^q \end{pmatrix} = \begin{pmatrix} s^{(n-2)}_{\lfloor\frac{n-k}{2}\rfloor} \\ q^{(k-2)} \\ s^{(n-2)}_{\lfloor\frac{n+k}{2}\rfloor-1} \end{pmatrix}, \tag{6.4}$$

which leads to

$$\bar{h}_q^{(k-2)} = T_{k-2}^{-1}\left(q^{(k-2)} - \lambda_{k-2}^q(t_{k-2} + Jt_{k-2})\right) = h_q^{(k-2)} + \lambda_{k-2}^q u^{(k-2)} \tag{6.5}$$

and

$$\lambda_{k-2}^q = \frac{s^{(n-2)}_{\lfloor\frac{n-k}{2}\rfloor} - t_{k-2}^T h_q^{(k-2)}}{\rho_0 + t_{k-2}^T u^{(k-2)} + \rho_{k-1}}\,. \tag{6.6}$$

This produces $h_q^{(n-2)} = w^{(n-2)}$. Perform (6.1), (6.2), and (6.3) once more, for $k = n$, to obtain $h_p^{(n)} = w^{(n)}$. The algorithm is initialized by solving a trivial 1×1 or 2×2 system, depending on whether n is odd or even, respectively.

Let us now show how we can compute the remaining quantities, necessary to calculate $x^{(n)}$, in $\mathcal{O}(n)$ operations. We start by computing $w^{(n-1)}$ from $w^{(n)}$ and $w^{(n-2)}$. From (4.15) and (4.16), we have

$$w_1^{(n-1)} = w_1^{(n)} + \eta, \tag{6.7}$$
$$w_j^{(n-1)} = -w_{j-1}^{(n-1)} + w_j^{(n)} + w_{j-1}^{(n-2)} + \eta u_{j-1}^{(n-2)} \quad (2 \le j \le n-1), \tag{6.8}$$

where

$$\eta = \frac{t_{n-1}^T w^{(n-1)} - t_{n-2}^T w^{(n-2)} - b_n + b_{n-1}}{\rho_0 + t_{n-2}^T u^{(n-2)} + \rho_{n-1}} \,. \tag{6.9}$$

To compute $w^{(n-1)}$, we first compute η, for which we use the fact that

$$\begin{aligned} t_{n-1}^T w^{(n-1)} = t_{n-1}^T T_{n-1}^{-1}(b^{(n-1)} + Jb^{(b-1)}) &= (t_{n-1} + Jt_{n-1})^T T_{n-1}^{-1} b^{(n-1)} \\ &= -(u^{(n-1)})^T b^{(n-1)}. \end{aligned}$$

The remaining components of $w^{(n-1)}$ then follow from (6.8).

Finally, we compute μ_{n-1} from (4.21) and obtain for the solution $x^{(n)}$

$$\begin{aligned} x_1^{(n)} &= w_1^{(n)} - \mu_{n-1}, \\ x_j^{(n)} &= -x_{j-1}^{(n)} + w_j^{(n)} - w_{j-1}^{(n-1)} - \mu_{n-1} u_{j-1}^{(n-1)} \quad (2 \le j \le n). \end{aligned}$$

Therefore, the computation of $w^{(n-1)}$ and, subsequently, of $x^{(n)}$, requires an additional $\mathcal{O}(n)$ flops.

We conclude the description of this new algorithm, the even-odd split Levinson algorithm, by noting that we used the even solutions to obtain the general solution. It would not be possible to do the same with the odd solutions because the reflection coefficients μ_k are not computed by the even-odd Levinson algorithm. In the even case this is not necessary because we can obtain these also from (4.21), as we did at the end of the algorithm when we calculated μ_{n-1} to obtain the general solution. However, (4.21) has no analogue in the odd case. Of course, if one wants to compute only the odd solution, then the reflection coefficients are not necessary.

Complexity. Since the even solutions of the Yule–Walker equations are assumed to be available from the split Durbin algorithm, several quantities in the even-odd Levinson algorithm need not be recomputed. The only remaining operations to be carried out are two scalar products and two updates, representing k additions and k multiplications, per step. Since k increases by two at every step, this means a total of $\frac{1}{2}n^2 + \mathcal{O}(n)$ additions and $\frac{1}{2}n^2 + \mathcal{O}(n)$ multiplications. Taking into account the complexity of the split Durbin algorithm, the complexity of this combined algorithm is $\frac{3}{2}n^2 + \mathcal{O}(n)$ additions and $n^2 + \mathcal{O}(n)$ multiplications or $\frac{5}{2}n^2 + \mathcal{O}(n)$ flops. To compute only the even or only the odd solutions of a system with arbitrary right-hand side, there is no need to compute $w^{(n-2)}$, so that the even-odd split Levinson method requires $\frac{5}{4}n^2 + \mathcal{O}(n)$ additions and $\frac{3}{4}n^2 + \mathcal{O}(n)$ multiplications or $2n^2 + \mathcal{O}(n)$ flops. This also means that if one has two independent processors available, rather than just one, then the solution $x^{(n)}$ can be computed in only $2n^2 + \mathcal{O}(n)$ flops. We note that the split Levinson algorithm in [11] cannot reduce its complexity in such cases.

7. Summary. Table 7.1 contains the number of floating point operations needed by the different methods described in this work to solve an arbitrary right-hand side problem on a sequential machine.

TABLE 7.1
Comparison of the complexity of methods for an arbitrary right-hand side problem.

Method	Additions	Multiplications	Total number of flops
Levinson	$2n^2 + \mathcal{O}(n)$	$2n^2 + \mathcal{O}(n)$	$4n^2 + \mathcal{O}(n)$
Split Levinson	$2n^2 + \mathcal{O}(n)$	$n^2 + \mathcal{O}(n)$	$3n^2 + \mathcal{O}(n)$
Even-odd Levinson	$2n^2 + \mathcal{O}(n)$	$\frac{3}{2}n^2 + \mathcal{O}(n)$	$\frac{7}{2}n^2 + \mathcal{O}(n)$
Even-odd split Levinson	$\frac{3}{2}n^2 + \mathcal{O}(n)$	$n^2 + \mathcal{O}(n)$	$\frac{5}{2}n^2 + \mathcal{O}(n)$

REFERENCES

[1] G.S. Ammar and W.B. Gragg, *The generalized Schur algorithm for the superfast solution of Toeplitz systems*, in Rational Approximations and its Applications in Mathematics and Physics, J. Gilewicz, M. Pindor, and W. Siemaszko, eds., Lecture Notes in Math. 1237, Springer, New York, 1987, pp. 315–330.

[2] G.S. Ammar and W.B. Gragg, *Numerical experience with a superfast Toeplitz solver*, Linear Algebra Appl., 121 (1989), pp. 185–206.

[3] A.L. Andrew, *Eigenvectors of certain matrices*, Linear Algebra Appl., 7 (1973), pp. 151–162.

[4] R.R. Bitmead and B.D.O. Anderson, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra Appl., 34 (1980), pp. 103–116.

[5] J.R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 349–364.

[6] A. Cantoni and F. Butler, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275-288.

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

[8] G. Cybenko and C. Van Loan, *Computing the minimum eigenvalue of a symmetric positive definite Toeplitz matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 123–131.

[9] P. Delsarte and Y. Genin, *Spectral properties of finite Toeplitz matrices*, in Mathematical Theory of Networks and Systems, Proceedings of the MTNS-83 International Symposium, Beer-Sheva, Israel, 1983, pp. 194–213.

[10] P. Delsarte and Y. Genin, *The split Levinson algorithm*, IEEE Trans. Acoust. Speech Signal Process., 34 (1986), pp. 470–478.

[11] P. Delsarte and Y. Genin, *An extension of the split Levinson algorithm and its relatives to the joint process estimation problem*, IEEE Trans. Inform. Theory, 34 (1989), pp. 482–485.

[12] J. Durbin, *The fitting of time series model*, Rev. Inst. Int. Stat., 28 (1960), pp. 233–243.

[13] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.

[14] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, 1958.

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

[16] T. Kailath, A. Vieira, and M. Morf, *Inverses of Toeplitz operators, innovations, and orthogonal polynomials*, SIAM Rev., 20 (1978), pp. 106–119.

[17] H. Krishna and Y. Wang, *The split Levinson algorithm is weakly stable*, SIAM J. Numer. Anal., 30 (1993), pp. 1498–1508.

[18] N. Levinson, *The Wiener RMS (root mean square) error criterion in filter design and prediction*, J. Math. Phys., 25 (1947), pp. 261–278.

[19] J. Makhoul, *Linear prediction: A tutorial review*, Proc. IEEE, 63 (1975), pp. 561–580.

[20] A. Melman, *A symmetric algorithm for Toeplitz systems*, Linear Algebra Appl., 301 (1999), pp. 145–152.

[21] V.F. Pisarenko, *The retrieval of harmonics from a covariance function*, Geophys. J. Royal Astron. Soc., 33 (1973), pp. 347–366.

[22] W.F. Trench, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 135–146.

SIAM J. Matrix Anal. Appl.
Vol. 23, No. 1, pp. 271–294
© 2001 Society for Industrial and Applied Mathematics

MINIMAL ORDERINGS REVISITED*

BARRY W. PEYTON†

Abstract. When minimum orderings proved too difficult to deal with, Rose, Tarjan, and Lueker instead studied minimal orderings and how to compute them [*SIAM J. Comput.*, 5 (1976), pp. 266–283]. This paper introduces an algorithm that is capable of computing much better minimal orderings much more efficiently than the algorithm of Rose, Tarjan, and Lueker. The new insight is a way to use certain structures and concepts from modern sparse Cholesky solvers to reexpress one of the basic results of Rose, Tarjan, and Lueker. The new algorithm begins with any initial ordering and then refines it until a minimal ordering is obtained. It is simple to obtain high-quality low-cost minimal orderings by using fill-reducing heuristic orderings as initial orderings for the algorithm. We examine several such initial orderings in some detail. Our results here and previous work by others indicate that the improvements obtained over the initial heuristic orderings are relatively small because the initial orderings are minimal or nearly minimal. Nested dissection orderings provide some significant exceptions to this rule.

Key words. minimal orderings, minimal fill, LEX M algorithm, supernodes, minimum degree, nested dissection, multisection, sparse matrix computations

AMS subject classifications. 65F05, 65F50, 65-04, 68R10

PII. S089547989936443X

1. Introduction.

Let A be an $n \times n$ symmetric positive definite matrix, let P be an $n \times n$ permutation matrix, and let L be the Cholesky factor of PAP^T. A *minimum ordering* is any ordering P that minimizes the number of nonzero entries in L, subject to the usual assumption that no lucky cancellation occurs. Rose, Tarjan, and Lueker [19] conjectured that the problem of computing a minimum ordering is NP-complete, and later Yannakakis [22] verified this conjecture. Rose, Tarjan, and Lueker [19] turned their attention instead to the easier problem of computing a minimal ordering. This paper revisits this problem: we give a new method for refining any initial ordering to obtain a minimal ordering whose fill is a subset of the initial ordering's fill. Others [3, 4, 5, 6, 7] have revisited this problem previously and taken a similar overall approach. The work of Blair, Heggernes, and Telle [5, 6] is particularly relevant. How the work herein relates to previous work, especially [5] and [6], will be discussed at various points throughout the paper, and also will be summarized in the concluding remarks in section 6.

Following Rose, Tarjan, and Lueker, we use graphs to define minimal orderings. Let $G = (V, E)$ be the graph of PAP^T; that is, $V = \{1, 2, \ldots, n\}$ and an undirected edge $\{i, j\}$, $i \neq j$, belongs to E if and only if the (i, j)-entry of PAP^T is not zero. (Only the labeling of the vertices varies as P varies; the structure of the graph and of course the number of edges, $e = |E|$, remain the same.) Define G^+ to be the *fill graph* associated with PAP^T; that is, G^+ is the graph of $L + L^T$ under the usual assumption that no lucky cancellation occurs. Note that $G^+ = (V, E \cup F)$, where F is composed of the fill edges created by the elimination process; hence, G^+ is a *supergraph* of G.

*Received by the editors December 3, 1999; accepted for publication (in revised form) by S. Vavasis December 15, 2000; published electronically July 2, 2001. This work was supported by the Applied Mathematical Sciences subprogram of the Office of Science, U.S. Department of Energy contract DE-AC05-00OR22725 with UT-Battelle, LLC.

http://www.siam.org/journals/simax/23-1/36443.html

†Computational Sciences Section, Computer Science and Mathematics Division, Oak Ridge National Laboratory, P. O. Box 2008, Building 6012, Oak Ridge, TN 37831-6367 (peytonbw@ornl.gov).

A graph is *chordal* if every cycle of length greater than three has a chord, that is, an edge joining two nonadjacent vertices in the cycle. It is well known [17, 18] that G^+ is a chordal supergraph of G. A minimum ordering P minimizes the number of edges in G^+ over all orderings; in this case, G^+ is a *minimum chordal supergraph* of G. For a *minimal chordal supergraph* $G^* = (V, E \cup F^*)$ of G, every supergraph $G' = (V, E \cup F')$ of G such that $F' \subset F^*$ is not chordal. A *minimal ordering* produces a fill graph G^+ that is a minimal chordal supergraph of G.

We are motivated primarily, but not solely, by the application to sparse symmetric positive definite factorization. At the most general level, we are interested simply in computing minimal orderings with genuinely low fill as efficiently as possible. We are also motivated by the following questions and issues.

1. We would like to investigate how close to minimal various heuristic orderings are.
2. We would like to observe how much savings in factor storage and work can be obtained by obtaining a minimal ordering from a good heuristic initial ordering.
3. Sometimes a different perfect elimination ordering of the fill graph is desired. Such an ordering is known as an *equivalent ordering*. If the fill graph is not minimal, then an equivalent ordering may drop fill from the graph and hence not be truly equivalent; the so-called equivalent ordering then perturbs the original storage scheme. Minimal fill ensures that any equivalent ordering does not drop fill and hence is truly equivalent.
4. More broadly speaking, a primary goal of this paper is to carry a few of the key insights in Rose, Tarjan, and Lueker [19] back into the sparse factorization setting in a fruitful way.

We use a key result in Rose, Tarjan, and Lueker [19] to lay the groundwork for a new minimal ordering algorithm. Beginning with any initial ordering, the new algorithm generates a sequence of reorderings, each removing additional fill from the current fill graph, until a minimal chordal supergraph, and hence a minimal ordering, is obtained. Several familiar concepts and algorithms from sparse Cholesky factorizations are used to formulate and implement the algorithm; these include elimination trees, supernodes, supernodal elimination trees, topological orderings, the minimum degree algorithm, and column counts. Although we assume that readers will have some familiarity with these concepts and algorithms, we also include references and a minimum of background material where needed.

Both the LEX M algorithm of Rose, Tarjan, and Lueker [19] and the algorithm of Ohtsuki [16] compute a minimal ordering in $O(ne)$ time; both the algorithm of Berry [3] and the algorithm of Dahlhaus [7] also compute a minimal ordering in $O(ne)$ time. Let $f = |F|$. The algorithm of Blair, Heggernes, and Telle [5, 6] computes a minimal ordering in $O(f(e + f))$ time. Partly because of the use of quotient graphs and minimum degree in the new algorithm, the new algorithm's time complexity remains unknown; consequently, we rely exclusively on empirical testing to evaluate the algorithm's time efficiency. The first tests we conduct in section 5 confirm that LEX M does not measure up to the new minimal ordering algorithm in either ordering quality or ordering time. The same should hold true for the algorithm of Ohtsuki. The algorithms of Berry and Dahlhaus also produce high-quality minimal orderings from high-quality initial orderings, but we did not determine empirically how fast they do it because

1. the algorithms of Berry and Dahlhaus also run in $O(ne)$ time, and

2. implementing these algorithms is significantly more difficult than implementing LEX M.

However, an implementation of the algorithm of Blair, Heggernes, and Telle was made available to us by the authors, so we were able to run some empirical tests for this algorithm. The first tests for our algorithm use minimum degree with true degree and no multiple elimination (MDtru) to produce the initial orderings. These initial orderings are often minimal or very close to minimal; often the minimal ordering algorithm serves merely as a relatively cheap means to verify that the initial ordering is minimal.

Many of the tests in section 5 with other initial orderings produce similar results, though there are some differences worth observing. The orderings tested include random orderings, METIS nested dissection (ND) orderings [11], multisection (MS) orderings [2, 10] based on METIS ND, and multiple minimum degree orderings with external degree (MMDext) as introduced by Liu [13]. Roughly speaking, most of the MDtru orderings are minimal, the MMDext orderings are very close to minimal, and the MS orderings are close to minimal. The ND orderings are not as close to minimal as the other ordering heuristics, and some of the ND orderings are far from minimal. Finally, the random orderings are extremely far from minimal, but the minimal orderings obtained from random initial orderings are poor fill-reducing orderings and are expensive to compute. Blair, Heggernes, and Telle [6] first observed that minimum degree orderings are far closer to minimal than ND orderings when they applied their algorithm to MMDext and METIS ND initial orderings.

The following gives an outline of this paper. Section 2 presents background material from Rose, Tarjan, and Lueker [19] and from the area of sparse Cholesky factorization. It also uses a simple example to introduce the key idea behind the algorithm. Section 3 presents the main result, which uses some concepts and tools from sparse Cholesky factorization to recast one of the insights in Rose, Tarjan, and Lueker. In section 4 the main result forms the basis for a new minimal ordering algorithm. Section 5 compares the new algorithm with the LEX M algorithm, experiments with various initial orderings, and finally compares the new algorithm with the algorithm of Blair, Heggernes, and Telle. Section 6 summarizes and adds a few concluding remarks.

2. Background. In section 2.1, we state a scheme for obtaining a minimal chordal supergraph from a nonminimal chordal supergraph; the scheme is implicit in a result of Rose, Tarjan, and Lueker [19]. With further development and refinement, this scheme will become an algorithm for computing minimal orderings. Section 2.1 also states another key result from [19]. In section 2.2, we give some concepts and tools from sparse Cholesky factorization that will be used to develop the new minimal ordering algorithm. Section 2.3 highlights the key idea behind the algorithm as it plays out on a very small example.

2.1. Computing minimal chordal supergraphs. Let G^* be a chordal supergraph of the graph G. A *candidate edge* $\{u, v\}$ is any fill edge such that G^* remains chordal after $\{u, v\}$ has been removed from the graph. Rose, Tarjan, and Lueker [19] showed that every nonminimal chordal supergraph has a candidate edge. As an immediate consequence of this result and the definition of candidate edges, we have the following proposition.

PROPOSITION 2.1 (Rose, Tarjan, and Lueker [19]). *A chordal supergraph is minimal if and only if it has no candidate edges.*

```
Input: a chordal supergraph G* of the graph G.
while there is a candidate edge in G* do
    remove a candidate edge from the graph G*;
endwhile;
```

FIG. 2.1. *Scheme for generating a minimal chordal supergraph.*

As an immediate consequence of Proposition 2.1, the scheme shown in Figure 2.1 will produce a minimal chordal supergraph. Both the algorithm introduced in this paper and the algorithm in Blair, Heggernes, and Telle [5, 6] explicitly remove candidate edges until minimality is achieved. The set of candidate edges changes as edges are removed from the graph: some noncandidate fill edges may become candidate edges; some candidate edges may cease to be candidate edges.

The following proposition from Rose, Tarjan, and Lueker [19] characterizes the candidate edges.

PROPOSITION 2.2 (Rose, Tarjan, and Lueker [19]). *Let G^*, which is $(V, E \cup F)$, be a chordal supergraph of G, which is (V, E). A fill edge $\{u, v\} \in F$ of G^* is not a candidate edge if and only if there exist two vertices a and z such that a and z are both adjacent to u and v in G^* but are not adjacent to one another in G^*.*

In section 3, we use concepts and tools from sparse Cholesky factorization to recast this characterization in the case where G^* is the fill graph G^+ associated with the graph G of PAP^T.

2.2. Concepts and tools from sparse factorization. The fill graph G^+ is obtained from the graph G of PAP^T by an elimination process that models the factorization elimination process. Let G_k be the graph obtained from G by adding every edge needed to make the vertices adjacent to k ($\text{adj}_G[k]$) a clique and then by eliminating k and the edges incident upon k. The elimination process replaces G with G_1, G with G_2, G with G_3, and so on, until it finally replaces G with G_{n-1}. The fill graph has the edges belonging to the original graph G along with the fill edges generated by the elimination process. We also define an elimination graph G_X for an arbitrary subset of vertices X. This graph is obtained by using the elimination process to eliminate in any order the vertices of X (and only the vertices of X). The resulting graph is independent of the order in which the vertices of X are removed.

For a vertex k of a graph G', let $\text{madj}_{G'}[k]$ be the neighbors of k in G' that are numbered higher than k. The parent function of the *elimination tree* (or forest) associated with a fill graph G^+ is defined as follows: if $\text{madj}_{G^+}[k]$ is empty, then the parent of k is null and k is a root in the forest; otherwise, the parent of k is the lowest numbered member of $\text{madj}_{G^+}[k]$. The following fact [15] proves useful later on. Let $c_1, c_2, \ldots, c_t$ be the children of a vertex p in the elimination tree. Then

$$\{p\} \cup \text{madj}_{G^+}[p] = \left(\bigcup_{i=1}^{t} \text{madj}_{G^+}[c_i] \right) \cup \{p\} \cup \text{madj}_G[p]. \tag{2.1}$$

Note that (2.1) holds only for fill graphs G^+ and not for arbitrary chordal supergraphs.

A vertex a is an *ancestor* of vertex d (and d is a *descendant* of a) if a lies on the path from d to the root of d's tree in the elimination forest. The vertex a is a *proper ancestor* of vertex d (and d is a *proper descendant* of a) if a is distinct from d and an ancestor of d.

Supernodes have become a familiar tool in various computations associated with sparse factorizations. The fundamental supernode partition is commonly used and has received some attention. Liu, Ng, and Peyton [12] give an algorithm that computes the fundamental supernode partition in $O(n + e)$ time. The supernode partition defined here and used by our algorithm is similar to supernode partitions used in practice, but it does not consist of fundamental supernodes nor does it define the maximal cliques of the chordal graph. This departure from the usual supernode partitions is motivated entirely by the problem at hand; the reason for it will become apparent in the proof of our main result, presented in section 3.

DEFINITION 2.3. *Let G^+ be the fill graph associated with the graph G of PAP^T. We define a supernode partition as follows: a child-parent pair c and p in the elimination tree belong to the same supernode if and only if c is the only child of p for which* $\mathrm{madj}_{G^+}[c] = \{p\} \cup \mathrm{madj}_{G^+}[p]$.

For a given elimination tree, this supernode partition is unique. Note that each supernode is a path in the elimination tree from a lowest vertex to an ancestor of the lowest vertex. Unless stated otherwise, all references to supernodes in this paper are to those defined by Definition 2.3.

We will also need the *supernodal elimination tree* associated with this supernode partition. Each supernode S is a vertex in the supernodal elimination tree. Supernode P is the parent of supernode C if the parent (in the elimination tree) of the "top" vertex in C is a vertex in P. Supernode R is a root if the top vertex in R is a root vertex in the elimination tree (or forest).

Let $S = \{f = u_1, u_2, \ldots, u_r\}$ be a supernode with the vertices listed in elimination order. A vertex v is a *proper descendant* of S if $v \notin S$ and v is a descendant of some vertex in S in the elimination tree. Let $T[S]$ be the subtree of the elimination tree rooted at S; that is, $T[S]$ includes the vertices of S and all vertices that are proper descendants of S in the elimination tree. Let $D[S] := T[S] \setminus S$ so that it contains precisely the proper descendants of S in the elimination tree. Note that a supernode S' is a proper descendant of a supernode S in the supernodal elimination tree if and only if the vertices of S' are proper descendants of S in the elimination tree.

From Definition 2.3 it follows that

$$\begin{aligned}\mathrm{madj}_{G^+}[u_1] &= \{u_2\} \cup \mathrm{madj}_{G^+}[u_2] \\ &= \{u_2, u_3\} \cup \mathrm{madj}_{G^+}[u_3] \\ &\vdots \\ &= \{u_2, u_3, \ldots, u_r\} \cup \mathrm{madj}_{G^+}[u_r].\end{aligned}$$

Hence, $\{f\} \cup \mathrm{madj}_{G^+}[f]$ is a clique in G^+, S is a clique in G^+, and $S \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f]$. In sections 4.1 and 4.2, we need to know something about the impact of supernodes on certain elimination graphs. For example, it follows from a basic result in Liu [15] that

$$\{f\} \cup \mathrm{adj}_{G_{D[S]}}[f] = \{f\} \cup \mathrm{madj}_{G^+}[f],$$

and for every i, $2 \le i \le r$, it also follows that

$$\{u_i\} \cup \mathrm{adj}_{G_{D[S]}}[u_i] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f].$$

These and similar facts are used, and in some cases more closely argued, in sections 4.1 and 4.2.

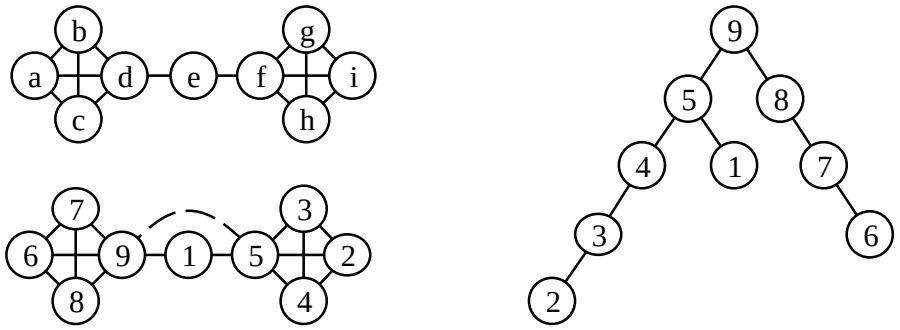


FIG. 2.2. *A chordal graph, a minimum degree ordering of the graph, and the associated elimination tree. The single fill edge is a dashed line.*

2.3. An example. The graph shown in the upper left-hand corner of Figure 2.2 is chordal, and hence it has a perfect elimination ordering, that is, a no-fill ordering. It follows that any ordering of this graph is minimal if and only if it is a perfect elimination ordering. Let us consider a minimum degree ordering of the graph. Any minimum degree ordering will order vertex e first since $\deg_G(e)$, which is 2, is smallest among the degrees; this creates fill edge $\{d, f\}$, and consequently, any minimum degree ordering is not minimal.

The reader should verify that the numbering given in Figure 2.2 is a minimum degree ordering of the original graph. Note the single fill edge $\{5, 9\}$, which is critical and which our minimal ordering algorithm must eliminate. Implementations of the minimum degree ordering algorithm find supernodes when they use mass elimination and indistinguishable vertices to eliminate vertices together. For the example, the supernode partition found by our minimum degree algorithm is $\{1\}$, $\{2, 3, 4\}$, $\{5\}$, $\{6, 7, 8\}$, and $\{9\}$. Some implementations of minimum degree find so-called fundamental supernodes; the only vertex in a supernode that can have more than one child in the elimination tree is the first vertex in the supernode. These are *not* the supernodes of Definition 2.3. The key observation is that by Definition 2.3, $\{1, 5\}$ is a supernode; the reader should verify that this is the case.

It is well known that any topological ordering of the elimination tree produces the same factor fill and work as the original topological ordering [15]. It is convenient to change to a different topological ordering in order to number the vertices of each supernode of Definition 2.3 together. For the example, a postordering of the elimination tree serves this purpose. Figure 2.3 displays this equivalent ordering of the fill graph and elimination tree. Under the new ordering, the supernodes of Definition 2.3 are $\{1, 2, 3\}$, $\{4, 5\}$, $\{6, 7, 8\}$, and $\{9\}$, also shown in Figure 2.3.

The algorithm we detail in section 4 uses a topological ordering of the supernodal elimination tree to eliminate the supernodes one after another. It uses the minimum degree algorithm (with true degree) to order the vertices within supernodes. Note that no new fill can be introduced by this process but old fill may disappear, and this is what makes the algorithm work. The algorithm first uses minimum degree to eliminate the vertices of $\{1, 2, 3\}$ from the original graph. The vertices can be removed in any order; let us say that the current order is retained. Note that no fill disappears. The resulting elimination graph $G_{\{1,2,3\}}$ is shown in Figure 2.4.

Next the algorithm uses minimum degree to eliminate the vertices of $\{4, 5\}$ from $G_{\{1,2,3\}}$. Note that vertex 5 has degree 1 and vertex 4 has degree 2; consequently,

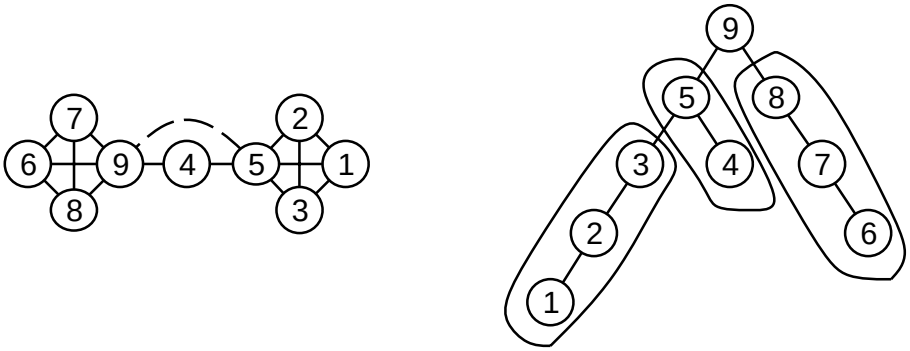


FIG. 2.3. *A renumbering of the minimum degree fill graph and elimination tree using a postordering of the tree. Displayed is the supernode partition of Definition* 2.3.

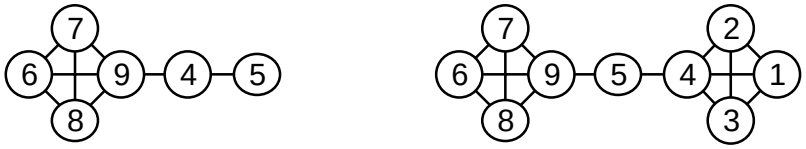


FIG. 2.4. *On the left, the elimination graph* $G_{\{1,2,3\}}$*; on the right, a perfect elimination ordering generated by the algorithm.*

they change places in the ordering and the fill edge disappears. Continuing, the algorithm uses minimum degree to eliminate the vertices of $\{6, 7, 8\}$ and then $\{9\}$; no fill disappears in either case. Suppose that the order of the vertices in $\{6, 7, 8\}$ is retained. Then the new ordering is shown in Figure 2.4. It is a perfect elimination ordering, and hence minimal.

This small example shows how the algorithm applies a restricted (or blocked) version of minimum degree to the supernodal elimination tree of Definition 2.3 to make candidate edges disappear. The step described above is applied to a succession of fill graphs that become smaller and smaller until there are no candidate edges left to remove. Our main result in section 3 shows why we can count on the absence of candidate edge $\{5, 9\}$ from the elimination graph $G_{\{1,2,3\}}$ shown in Figure 2.4. This is what ensures that the algorithm works. The algorithmic details are given in section 4.

3. A characterization of candidate edges. We now state and prove an alternative characterization of the candidate edges, which is the basis for the algorithm we sketched in section 2.3 and will detail in section 4.

PROPOSITION 3.1. *Let* G^+ *be the fill graph associated with the graph* G *of* PAP^T*. Let* S *be a supernode in the supernode partition given by Definition* 2.3*. Assume that* (1) $u \in S$, (2) $u < v$ *in the elimination order, and* (3) $\{u, v\}$ *is a fill edge. We then have the following:* $\{u, v\}$ *is a candidate edge if and only if* $\{u, v\}$ *is not an edge in the elimination graph* $G_{D[S]}$*.*

Proof. Throughout the proof, let f be the first vertex (i.e., the lowest numbered vertex) in supernode S.

Assume that $\{u, v\}$ is not a candidate edge. By Proposition 2.2 there exist then two vertices a and z that are adjacent to u and v in G^+ but not adjacent to each other in G^+. Now, by assumption, $u \in S$ and f is the first vertex in supernode S. It follows

that $\mathrm{madj}_{G^+}[u] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f]$. The lower-numbered neighbors of u in G^+ must be descendants of u in the elimination tree [15, 21]. Therefore, these lower-numbered neighbors belong to $T[S]$, which is $S \cup D[S]$. Since $S \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f]$, we have $\mathrm{adj}_{G^+}[u] \subseteq D[S] \cup \{f\} \cup \mathrm{madj}_{G^+}[f]$. It then follows that both a and z belong to $D[S] \cup \{f\} \cup \mathrm{madj}_{G^+}[f]$. Since $\{f\} \cup \mathrm{madj}_{G^+}[f]$ is a clique in G^+, at least one of the two nonadjacent vertices a and z belongs to $D[S]$. It follows that $\{u, v\}$ is an edge in the elimination graph $G_{D[S]}$.

To prove the other direction, assume that $\{u, v\}$ is a candidate edge and that $\{u, v\}$ is a fill edge in the elimination graph $G_{D[S]}$. It suffices to derive a contradiction from these assumptions.

Since $\{u, v\}$ is a fill edge in the elimination graph $G_{D[S]}$, there exists a vertex $a \in D[S]$ that is adjacent to both u and v in G^+. For the following reasons we may assume without loss of generality that a is a child of a vertex in S. Any descendant vertex d of S has as one of its ancestors a vertex c that is a child of some vertex in S; moreover, for any child c of a vertex in S and any descendant d of c, we have

$$\mathrm{madj}_{G^+}[d] \cap \big(\{f\} \cup \mathrm{madj}_{G^+}[f]\big) \subseteq \mathrm{madj}_{G^+}[c] \cap \big(\{f\} \cup \mathrm{madj}_{G^+}[f]\big).$$

Now, since $\{u, v\}$ is a candidate edge by assumption, it follows that any pair of vertices adjacent to both u and v are adjacent to one another. Since u and v are both adjacent in G^+ to every vertex in $\{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u, v\}$, it follows that a is adjacent in G^+ to every vertex in $\{f\} \cup \mathrm{madj}_{G^+}[f]$, so we can write $\{f\} \cup \mathrm{madj}_{G^+}[f] \subseteq \mathrm{madj}_{G^+}[a]$. Moreover, it follows that a is a child of f because if it were a child of some other vertex of S, then it could not be adjacent in G^+ to f because f would then be neither an ancestor nor a descendant of a. Furthermore, $\mathrm{madj}_{G^+}[a] = \{f\} \cup \mathrm{madj}_{G^+}[f]$ because by (2.1), $\mathrm{madj}_{G^+}[a] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f]$.

Now, supernode S begins at vertex f; that is, no descendants of f belong to S. Consequently, existence of the child a of f for which $\mathrm{madj}_{G^+}[a] = \{f\} \cup \mathrm{madj}_{G^+}[f]$ implies the existence of another child z of f for which $\mathrm{madj}_{G^+}[z] = \{f\} \cup \mathrm{madj}_{G^+}[f]$; were there no such vertex z, vertex a would have been incorporated into the supernode S and S would not begin at f. Vertices a and z clearly are adjacent to u and v in G^+, but they are not adjacent to each other because they are siblings in the elimination tree. From Proposition 2.2 it follows that $\{u, v\}$ *is not* a candidate edge, contrary to our assumption that it *is* a candidate edge. The result follows from this contradiction. ∎

4. A minimal ordering algorithm. Let G^+ be the fill graph associated with the graph G of PAP^T, and consider the supernode partition given by Definition 2.3. In the following definition, we explicitly partition the candidate edges among the supernodes, as suggested in the statement of Proposition 3.1. The vertices of S are listed in elimination order.

DEFINITION 4.1. *Let $S = \{f = u_1, u_2, \ldots, u_r\}$ be a supernode in the supernode partition given by Definition* 2.3. *The candidate edges of S in G^+ include every candidate edge $\{u, v\}$ for which $u \in S$ and $v \geq f$.*

Proposition 3.1 says that the candidate edges of S are missing from the elimination graph $G_{D[S]}$. If S has any candidate edges, then some of them can be removed simply by reordering the vertices of S; what follows in sections 4.1 and 4.2 expands on and justifies the preceding statement. The algorithm is presented in section 4.3.

4.1. Candidate edges and degrees in an elimination graph. First, we need two more definitions. Define the *true degree* of a vertex v in a graph G' to be the

number of neighbors of v in G', that is, $\deg_{G'}(v) := \left|\mathrm{adj}_{G'}[v]\right|$. Two vertices v and w in a graph G' are said to be *indistinguishable* if

$$\{v\} \cup \mathrm{adj}_{G'}[v] = \{w\} \cup \mathrm{adj}_{G'}[w].$$

Let $S = \{f = u_1, u_2, \ldots, u_r\}$ be a supernode in the supernode partition given by Definition 2.3. Again, the vertices of S are listed in elimination order. Consider the elimination graph $G_{D[S]}$. Recall that for each vertex in S the set $D[S]$ contains every descendant in the elimination tree that does not belong to S. Since vertex f has no proper descendant that belongs to S, every proper descendant of f belongs to $D[S]$. Note also that none of the ancestors of f are included in $D[S]$. It follows [15] that

$$\{f\} \cup \mathrm{adj}_{G_{D[S]}}[f] = \{f\} \cup \mathrm{madj}_{G^+}[f]. \tag{4.1}$$

Now, consider the elimination graph obtained by eliminating f from $G_{D[S]}$, that is, the elimination graph G_X where $X = \{f\} \cup D[S]$. From (4.1) it follows that $\mathrm{madj}_{G^+}[f]$ is a clique in G_X. Since $\{u_2, \ldots, u_r\} \subseteq \mathrm{madj}_{G^+}[f]$, it follows that for i, $2 \le i \le r$,

$$\{u_i\} \cup \mathrm{adj}_{G_X}[u_i] \supseteq \mathrm{madj}_{G^+}[f]. \tag{4.2}$$

From (4.2) and the fact that S is a supernode in G^+, it follows that for i, $2 \le i \le r$,

$$\{u_i\} \cup \mathrm{adj}_{G_X}[u_i] = \mathrm{madj}_{G^+}[f]. \tag{4.3}$$

In other words, the vertices $u_2, \ldots, u_r$ become indistinguishable from one another after f is removed from $G_{D[S]}$ to obtain G_X, as described previously. But the vertices $f, u_2, \ldots, u_r$ are not necessarily indistinguishable from one another in $G_{D[S]}$, and this is the key to the algorithm.

Now we shift the focus back to $G_{D[S]}$. It follows directly from (4.3) that for every i, $2 \le i \le r$, we have

$$\{u_i\} \cup \mathrm{adj}_{G_{D[S]}}[u_i] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f]. \tag{4.4}$$

Note that since $\{f\} \cup \mathrm{madj}_{G^+}[f]$ is a clique in G^+, any pair of vertices from this clique that is not joined by an edge in the original graph is joined by a fill edge in G^+. From (4.1) and Proposition 3.1, it follows that S has no candidate edges incident upon f. It follows from (4.4), Proposition 3.1, and the preceding comment on fill edges that the candidate edges of S incident upon u_i ($2 \le i \le r$) are precisely those joining u_i to any vertex in

$$\big(\{f\} \cup \mathrm{madj}_{G^+}[f]\big) \setminus \big(\{u_i\} \cup \mathrm{adj}_{G_{D[S]}}[u_i]\big). \tag{4.5}$$

It follows from (4.1) and the preceding statement that the number of candidate edges of S incident upon u_i is

$$\deg_{G_{D[S]}}(f) - \deg_{G_{D[S]}}(u_i). \tag{4.6}$$

4.2. Using minimum degree to remove candidate edges. From the last statement of the preceding subsection, it follows that a vertex $u_i \in S$ with minimum true degree $\deg_{G_{D[S]}}(u_i)$ has the most candidate edges of S incident upon it in G^+. Select such a vertex u_i to be eliminated from $G_{D[S]}$. The set $\mathrm{adj}_{G_{D[S]}}[u_i]$ will be the

monotone adjacency set of u_i no matter how the elimination process is completed. The candidate edges of S incident upon u_i in G^+ join u_i with the vertices of

$$\bigl(\{f\} \cup \mathrm{madj}_{G^+}[f]\bigr) \setminus \bigl(\{u_i\} \cup \mathrm{adj}_{G_{D[S]}}[u_i]\bigr);$$

such candidate edges exist if and only if $\deg_{G_{D[S]}}(u_i) < \deg_{G_{D[S]}}(f)$. Since the set $\mathrm{adj}_{G_{D[S]}}[u_i]$ will be the monotone adjacency set of u_i, it follows that all the candidate edges of S incident upon u_i do not appear in the new elimination graph; moreover, these candidate edges are the only edges to disappear from the monotone adjacency set of u_i. Note that no new edges that are not in G^+ are introduced since

$$\mathrm{adj}_{G_{D[S]}}[u_i] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u_i\}.$$

We then repeat the process. Let $X = \{u_i\} \cup D[S]$, and consider the elimination graph G_X. Choose from the uneliminated vertices of S a vertex u_j with minimum true degree $\deg_{G_X}(u_j)$. Vertex u_j has the most candidate edges of S incident upon it in G^+ that were not filled in by the previous elimination of u_i. The vertex u_j will be eliminated from G_X; hence, the set $\mathrm{adj}_{G_X}[u_j]$ will be the monotone adjacency set of u_j no matter how the elimination process is completed. Any nonfilled candidate edges of S incident upon u_j in G^+ join u_j with the vertices of

$$\bigl(\{f\} \cup \mathrm{madj}_{G_{D[S]}}[f]\bigr) \setminus \bigl(\{u_i, u_j\} \cup \mathrm{adj}_{G_X}[u_j]\bigr);$$

such candidate edges exist if and only if $\deg_{G_X}(u_j) < \deg_{G_X}(f)$. Since the set $\mathrm{adj}_{G_X}[u_j]$ will be the monotone adjacency set of u_j in the new elimination graph, it follows that all the nonfilled candidate edges of S incident upon u_i do not appear in the new elimination graph; moreover, these candidate edges are the only edges to disappear from the monotone adjacency set. Note that no new edges that are not in G^+ are introduced since

$$\mathrm{adj}_{G_X}[u_j] \subseteq \{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u_i, u_j\}.$$

We continue this process until all the vertices of S are removed from the original elimination graph $G_{D[S]}$. If S has any candidate edges at all, then some are removed by applying the minimum degree ordering heuristic to the vertices of S in the elimination graph $G_{D[S]}$, as we just did. Moreover, only candidate edges of S are removed by this process. If S has no candidate edges, then the vertices of S are indistinguishable from one another in $G_{D[S]}$, and applying the minimum degree ordering heuristic to the vertices of S in the elimination graph $G_{D[S]}$ produces an arbitrary ordering of S and does not change the fill. In this case, the sequence of true degrees is $\deg_{G_{D[S]}}(f)$, $\deg_{G_{D[S]}}(f)-1, \ldots,$ and $\deg_{G_{D[S]}}(f)-r+1$; moreover, if $u_{i_1}, u_{i_2}, \ldots, u_{i_r}$ is any ordering of S, then the monotone adjacency sets are $\{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u_{i_1}\}$, $\{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u_{i_1}, u_{i_2}\}, \ldots,$ and $\{f\} \cup \mathrm{madj}_{G^+}[f] \setminus \{u_{i_1}, u_{i_2}, \ldots, u_{i_r}\}$.

4.3. The algorithm. The algorithm for computing a minimal ordering, including some implementation details, appears in Figure 4.1. The algorithm requires an initial ordering; it will work with any initial ordering, including a random one. The algorithm repeats the major step until there is no reduction in the factor nonzero count (i.e., the fill edges). A major step breaks into the following two parts:

1. symbolic processing using the current ordering, and
2. a block elimination process with minimum degree refinement on each block (supernode) to obtain a new ordering.

```
[Initial ordering: can be any ordering]

[Repeat ordering refinement step until no progress is encountered]
until the total factor nonzero count is not reduced do

    [Symbolic preprocessing using current ordering]
    Compute elimination tree and postorder it [15];
    Compute column factor nonzero counts [9];
    Compute the supernodes (Definition 2.3);
    Compute supernodal elimination tree and a topological ordering
    of this tree;

    [block elimination process in block topological order and with
    minimum degree refinement to obtain new refinement of old ordering]
    Begin elimination process with original graph;
    for each supernode S (in topological order) do
        until all vertices in S have been eliminated do
            Select a vertex v ∈ S whose true degree in the
            current elimination graph is minimum among the
            uneliminated vertices in S;
            Eliminate v and form the quotient graph
            representation of the new elimination graph;
        end until;
    end for;

    Replace old ordering with new ordering;
    [Here we check new total factor nonzero count against old]
end until;
```

FIG. 4.1. *Algorithm for computing a minimal ordering, including some implementation details.*

During the symbolic part of a major step, the algorithm first computes the elimination tree, and then postorders it. We use the fast algorithm in [15] to compute the elimination tree; the postordering is needed to compute the column factor nonzero counts. (Column factor nonzero counts refer to the number of nonzero entries in each column of the Cholesky factor under the current postordering of the elimination tree.) Computing the column factor nonzero counts is achieved using the fast algorithm in [9]. With the elimination tree and column counts in hand, it is trivial to compute the supernodes of Definition 2.3 and the associated supernodal elimination tree. Finally, the algorithm computes a topological ordering of the supernodal elimination tree; this also is trivial.

During the elimination part of a major step, the algorithm processes the supernodes in the given topological order of the current supernodal elimination tree. The elimination process is a block elimination process; for any supernode S, the algorithm uses the minimum degree algorithm to eliminate the vertices of S together, and only after it has removed in the same fashion, supernode by supernode, the descendant vertices of S, $D[S]$. The topological ordering also ensures that no vertex from an ancestor supernode is removed before the vertices of the supernode and its descendants. In short, the analysis presented in sections 4.1 and 4.2 applies directly to the elimination graphs generated by the algorithm. That is, any supernode S that has candidate edges will have some of the candidate edges removed with no edges added

beyond the fill generated by the current ordering. Moreover, any edges removed are candidate edges.

The elimination process generates a new ordering. During the elimination process, the algorithm accumulates the amount of fill incurred by this ordering, and when the elimination process is finished, the algorithm compares it with the amount of fill incurred by the old ordering. If there is no reduction, then the algorithm stops. From the argument in the previous paragraph, a major step of the algorithm removes only candidate edges and removes them if and only if there exist such edges in the current fill graph. Since there are a finite number of edges, the algorithm must terminate at some point, and clearly it will be at the point where there are no candidate edges in the old fill graph and no candidate edges in the new fill graph. It follows from Proposition 2.1 that the algorithm terminates with a minimal ordering and minimal chordal supergraph. The algorithm is, indeed, merely an elaboration of the scheme shown in Figure 2.1.

As mentioned earlier, the algorithm of Blair, Heggernes, and Telle [5, 6] is also an elaboration of the scheme shown in Figure 2.1. They take a more direct approach: processing the fill edges in reverse the order they were introduced, they compute which examined fill edges are candidate edges in the current chordal supergraph; they use LEX M on subgraphs to compute which candidate edges need to be kept as fill. Processing the fill edges in reverse enables them to consider a fill edge as a candidate edge once, and this is the key to their $O(f(e+f))$ time efficiency.

Modern implementations of the minimum degree algorithm improve ordering quality by using *external degree* rather than true degree [8, 13]. External degree counts only the neighbors outside the current indistinguishable set to be "mass eliminated." Note that it is important for our minimal ordering algorithm to use true degree rather than external degree during the block elimination process. True degree gives priority to vertices incident upon the most candidate edges, as desired. External degree may give priority to a vertex incident upon no candidate edges even though there are vertices available that are incident upon candidate edges. Consequently, external degree could fail to detect candidate edges for a supernode that has some, while true degree is sure to detect and remove some candidate edges from any supernode that has some.

Note also that we have adapted a minimum degree code to perform the supernode-by-supernode elimination process. The adapted code inherits several of the standard enhancements that have been incorporated into such codes [8, 13]; these include mass elimination, indistinguishable nodes, incomplete degree update, and the generalized element storage scheme. Because any two vertices from distinct supernodes must be treated as separate vertices, some opportunities for mass elimination or detecting indistinguishability must be passed over. On the other hand, because the elimination process needs to know degrees of vertices in one supernode at a time, we can greatly reduce the number of degree calculations needed. Our timings indicate that the gains from the latter typically far outweigh the costs from the former.

5. Test results. We wrote Fortran 77 implementations of the new minimal ordering algorithm described in the previous section and the LEX M algorithm described in Rose, Tarjan, and Lueker [19, pp. 273, 280–281]. We used Fortran compiler `f77` with compiler optimization level `-O`, and we ran the tests on a SUN Sparc 20 workstation (model 41).

The primary purpose of section 5.1 is to provide a simple empirical proof of concept for the new algorithm. To achieve this goal it suffices to show that, in practice, the new algorithm can produce high-quality minimal orderings very efficiently com-

TABLE 5.1

Average factor floating point operations associated with MDtru/new-minimal, random/new-minimal, and LEX M. Averages are taken over ten runs with randomly permuted adjacency structures.

	MDtru/new-minimal		Random/new-minimal		
Matrix	% decrease	Final flops	% decrease	Final flops	LEX M flops
DIS060	none	1.16×10^7	99.2	3.77×10^7	2.05×10^7
DIS090	none	4.93×10^7	99.6	2.21×10^8	1.15×10^8
DIS120	none	1.33×10^8	99.8	7.56×10^8	3.46×10^8
NASA1824	none	5.59×10^6	96.0	4.23×10^7	4.98×10^7
NASA2910	none	3.22×10^7	97.8	1.43×10^8	9.65×10^7
NASA4704	none	4.42×10^7	97.9	3.84×10^8	3.42×10^8
SPA060	none	1.88×10^7	98.7	6.29×10^7	3.57×10^7
SPA090	none	8.11×10^7	99.4	3.39×10^8	1.87×10^8
SPA120	none	2.14×10^8	99.7	1.11×10^9	6.18×10^8
BCSSTK13	none	6.87×10^7	85.2	2.77×10^8	1.45×10^8
BCSSTK14	none	1.04×10^7	97.4	3.20×10^7	2.84×10^7
BCSSTK15	none	1.94×10^8	92.3	1.05×10^9	4.82×10^8
BCSSTK16	none	1.69×10^8	98.1	5.53×10^8	1.45×10^8
BCSSTK17	0.3	2.19×10^8	99.5	1.36×10^9	6.37×10^8
BCSSTK18	none	1.56×10^8	97.1	4.01×10^9	2.01×10^9
BCSSTK19	3.4	1.17×10^5	98.5	1.98×10^5	1.27×10^5
BCSSTK23	none	1.60×10^8	75.5	1.17×10^9	3.04×10^8
BCSSTK24	none	4.02×10^7	99.1	9.51×10^7	1.07×10^8
BCSSTK25	none	4.00×10^8	96.3	2.04×10^{10}	5.26×10^8
BCSSTK26	0.0	1.65×10^6	98.9	9.27×10^6	1.47×10^7

pared with the LEX M algorithm. Since we are interested in obtaining high-quality minimal orderings as efficiently as possible, we chose minimum degree orderings to be the initial orderings in section 5.1. To be consistent with the later use of true degree in the refinement step, we use true degree rather than the superior external degree [13] to compute the initial ordering, too. For the same reason we do not use multiple elimination; we use MDtru. While establishing proof of concept, we will also observe that MDtru often computes minimal orderings.

We examine random initial orderings in section 5.2, MMDext in section 5.3, METIS ND initial orderings in section 5.4, and MS initial orderings based on METIS ND in section 5.5. Some comparisons with the algorithm of Blair, Heggernes, and Telle are presented in section 5.6.

5.1. Proof of concept. The large run times of LEX M limited us to relatively small matrices for our test runs. Despite their limited size, the test matrices suffice for our purposes. For more accurate comparisons we computed ten random permutations for each graph, we computed ten permuted versions of the adjacency structure for each graph, and we ran both algorithms on each of the ten permuted adjacency structures for each of the matrices. We then averaged the reported statistics over the ten runs for each matrix. We report factor floating point operations rather than fill because comparing factor flops usually emphasizes the differences between orderings more than comparing factor nonzero counts. Table 5.1 reports average factor flops associated with the LEX M and the MDtru/new-minimal orderings. Table 5.2 reports average run times for both algorithms and the average number of major iterations taken by

TABLE 5.2

Average ordering times (in CPU seconds) and average number of major iterations for MDtru/new-minimal and for LEX M. Averages are taken over ten runs with randomly permuted adjacency structures.

	MDtru/new-minimal				
Matrix	MDtru time	Major iter.	Minimal time	Total time	LEX M time
DIS060	0.23	1.0	0.21	0.44	22.30
DIS090	0.80	1.0	0.54	1.34	134.27
DIS120	1.44	1.0	1.07	2.51	440.42
NASA1824	0.08	1.0	0.08	0.16	6.09
NASA2910	0.22	1.0	0.28	0.50	31.02
NASA4704	0.28	1.0	0.24	0.52	47.88
SPA060	0.18	1.0	0.15	0.33	18.89
SPA090	0.62	1.0	0.48	1.10	126.72
SPA120	1.17	1.0	0.92	2.09	452.17
BCSSTK13	0.24	1.0	0.15	0.39	11.88
BCSSTK14	0.10	1.0	0.09	0.19	7.87
BCSSTK15	0.54	1.0	0.28	0.82	37.15
BCSSTK16	0.40	1.0	0.43	0.83	90.07
BCSSTK17	1.02	2.1	1.80	2.82	331.14
BCSSTK18	1.15	1.0	0.70	1.85	138.01
BCSSTK19	0.02	8.1	0.17	0.19	0.37
BCSSTK23	0.53	1.0	0.18	0.71	12.84
BCSSTK24	0.13	1.0	0.19	0.32	37.98
BCSSTK25	3.35	1.0	1.58	4.93	565.62
BCSSTK26	0.11	1.3	0.12	0.23	4.43

the new algorithm.

A "1.0" in column three of Table 5.2, or equivalently a "none" in column two of Table 5.1, means that for all 10 permutations of the adjacency structure the initial MDtru ordering is minimal. Consequently, for all 10 permutations of the adjacency structure the new minimal ordering algorithm takes one major step. For 17 of the 20 problems, the initial MDtru ordering is minimal for all 10 permutations of the adjacency structure. For these problems, the initial factor flops and the final factor flops are identical. For BCSSTK17 and BCSSTK26, there is, on average, a small change: less than a 0.5% reduction in factor flops. The largest change is for the very small problem BCSSTK19, but it is still only a 3.4% reduction in factor flops. Clearly, the new minimal ordering algorithm serves most often merely to verify that the initial MDtru ordering is minimal; in the other three cases, it trims away fill (and factor flops) from nearly minimal MDtru orderings until minimality is achieved. In short, the MDtru ordering heuristic comes very close to being a minimal ordering algorithm in our tests.

Also note in Table 5.1 that the MDtru/new-minimal orderings consistently cost far fewer factor floating point operations than the LEX M orderings. For only one problem, BCSSTK16, does LEX M outperform MDtru/new-minimal in this regard. Typically, LEX M requires anywhere from a factor of two to a factor of four more factor flops than MDtru/new-minimal; sometimes LEX M requires a factor of eight or nine more factor flops than MDtru/new-minimal. This superiority of MDtru/new-minimal is not surprising; LEX M, which is a breadth-first search ordering, has much in common with bandwidth and profile reducing orderings, and hence a good general

sparse ordering like MDtru is naturally expected to reduce factor flops better than LEX M. The tests serve merely to confirm this expectation.

In Table 5.2, the total run times of the MDtru/new-minimal algorithm are, with one exception (BCSSTK19), a very small fraction of the corresponding run times of the LEX M algorithm. This is not surprising; the time complexity for LEX M is $O(ne)$, and the $O(ne)$ time complexity is fully realized by the implementation [19]. Although the time complexity of the minimum degree heuristic is unknown, the empirical efficiency of modern implementations of this heuristic is well established [8, 13]; it would be somewhat surprising to see minimum degree ordering times exceed any significant fraction of the corresponding LEX M ordering times.

There are known problems with the time efficiency of the minimum degree algorithm. For example, it is well known that a dense or near-dense row in the matrix seriously degrades the time efficiency of conventional implementations. Such rows are rare in practice and rare in most test collections; we apparently included no test problems on which the minimum degree algorithm runs very inefficiently.

In the cases where the new minimal ordering algorithm merely confirms the minimality of the initial MDtru ordering, the time to confirm minimality is usually smaller than, but comparable to, the MDtru ordering time. Exceptions include BCSSTK23 and BCSSTK25, for which the time to confirm minimality is unusually small, and NASA2910 and BCSSTK24, for which the time to confirm minimality is significantly greater than the MDtru ordering time. For BCSSTK17, BCSSTK19, and BCSSTK26, the average time to compute the minimal ordering divided by the average number of major iterations gives the average time per major iteration, which for each of these problems is less than but roughly comparable to the MDtru ordering time. Only BCSSTK19, with its 8.1 major iterations, has an MDtru/new-minimal time (0.19) that approaches in magnitude the time for LEX M (0.37).

5.2. Random. In the previous subsection, on the whole, the MDtru initial ordering did the work of computing minimal orderings, while the new minimal ordering algorithm merely detected minimality. In this subsection we look at the opposite extreme: the initial orderings are random, and the effort to achieve minimality is exerted solely by the new minimal ordering algorithm. Because of large run times, we run tests on the same test set of relatively small problems used in the previous subsection. The results of the random/new-minimal runs are presented in Tables 5.1 and 5.3.

The initial random orderings are very poor, as expected; the factor flops for the random orderings are one or two orders of magnitude larger than the factor flops for the corresponding final minimal orderings. The large reductions in factor flops require many major iterations; generally, hundreds of iterations are required, with over a thousand iterations required for BCSSTK18 and BCSSTK25. Though the final orderings greatly improve upon the initial random orderings, the final orderings are poor compared with the MDtru orderings; moreover, the random/new-minimal orderings are poor compared with the LEX M orderings, with random/new-minimal orderings producing more factor flops than LEX M for 17 of the 20 problems. For the following matrices, the random/new-minimal ordering produces over twice as many factor flops as LEX M: DIS120, BCSSTK15, BCSSTK16, BCSSTK17, BCSSTK23, and BCSSTK25. The three matrices where random/new-minimal outperforms LEX M are small (NASA1824, BCSSTK24, and BCSSTK26), and the improvement of random/new-minimal over LEX M orderings for these matrices is relatively small.

TABLE 5.3
Average ordering times (in CPU seconds) and average number of major iterations for random/new-minimal and for LEX M. Averages are taken over ten runs with randomly permuted adjacency structures.

	Random/new-minimal				
Matrix	Random time	Major iter.	Minimal time	Total time	LEX M time
DIS060	0.02	178.8	50.30	50.32	22.30
DIS090	0.05	371.4	383.54	383.59	134.27
DIS120	0.09	570.5	1665.96	1666.05	440.42
NASA1824	0.01	112.9	15.36	15.37	6.09
NASA2910	0.02	83.6	25.37	25.39	31.02
NASA4704	0.03	232.6	133.07	133.10	47.88
SPA060	0.02	188.1	52.46	52.48	18.89
SPA090	0.05	380.5	403.84	403.89	126.72
SPA120	0.09	591.8	1766.61	1766.70	452.17
BCSSTK13	0.01	188.4	52.60	52.61	11.88
BCSSTK14	0.01	69.0	8.45	8.46	7.87
BCSSTK15	0.02	291.5	190.56	190.58	37.15
BCSSTK16	0.03	133.3	84.36	84.39	90.07
BCSSTK17	0.07	410.1	1315.54	1315.61	331.14
BCSSTK18	0.07	1171.7	9608.18	9608.25	138.01
BCSSTK19	0.01	45.1	1.20	1.21	0.37
BCSSTK23	0.02	473.5	350.30	350.32	12.84
BCSSTK24	0.02	103.3	31.20	31.22	37.98
BCSSTK25	0.09	1441.3	17374.76	17374.85	565.62
BCSSTK26	0.01	120.1	16.11	16.12	4.43

It is not surprising that the run times to compute the initial random orderings are extremely small compared with the large run times to compute the associated minimal orderings. After all, a random ordering is obtained by a single $O(n \log n)$ sort. Several factors contribute to the exceptionally large run times for random/new-minimal orderings. First, and most obvious, is the large number of major iterations required for each problem. Second, the cost of each iteration is increased by the large amounts of fill that must be represented by the sequence of quotient graphs. Third, random orderings lead to relatively small supernodes, so the minimum degree refinement algorithm enjoys limited compression from supernodes. Fourth, the minimum degree refinement code uses the trick described by Amestoy, Davis, and Duff [1] of recompressing the quotient graphs when space is exhausted; the large amounts of fill and relatively small supernodes lead to many recompressions.

The run times for random/new-minimal are often poor compared with the LEX M orderings, with 17 out of 20 problems requiring more run time than LEX M. For many of the matrices, it requires a factor of two up to a factor of four more time. The results in this subsection, along with the rest of the results in this section, indicate that the new minimal ordering algorithm depends on a high-quality initial ordering to obtain a high-quality minimal ordering using small run time.

5.3. Multiple minimum degree with external degree. Next, we obtained initial orderings from Liu's MMDext [13]. We include it because it has become a standard by which other orderings are evaluated and because we wish to compare it with MDtru. We include all the test matrices used in the previous two subsections and

TABLE 5.4

Average factor floating point operations, average ordering times (in CPU seconds), and average number of major iterations for MMDext/new-minimal. Averages are taken over ten runs with randomly permuted adjacency structures.

	MMDext/new-minimal					
Matrix	% decrease	Final flops	MMDext time	Major iter.	Minimal time	Total time
CRY01	0.0	3.18×10^8	0.51	1.3	0.60	1.11
CRY02	0.0	4.04×10^9	2.11	1.2	2.15	4.26
CRY03	0.0	1.34×10^{10}	5.46	1.3	4.91	10.37
DIS060	none	9.85×10^6	0.13	1.0	0.13	0.26
DIS090	none	4.05×10^7	0.34	1.0	0.35	0.69
DIS120	none	1.13×10^8	0.73	1.0	0.83	1.56
NASA1824	0.2	4.91×10^6	0.08	1.2	0.08	0.16
NASA2910	0.1	2.30×10^7	0.23	2.0	0.51	0.74
NASA4704	none	3.62×10^7	0.24	1.0	0.24	0.48
SPA060	none	1.68×10^7	0.13	1.0	0.13	0.26
SPA090	none	6.63×10^7	0.34	1.0	0.36	0.70
SPA120	none	1.81×10^8	0.72	1.0	0.80	1.52
BCSSTK13	0.2	5.94×10^7	0.30	1.2	0.19	0.49
BCSSTK14	none	9.28×10^6	0.13	1.0	0.09	0.22
BCSSTK15	0.0	1.70×10^8	0.77	1.1	0.35	1.12
BCSSTK16	0.1	1.40×10^8	0.43	1.6	0.71	1.14
BCSSTK17	0.1	1.98×10^8	1.13	1.5	1.43	2.56
BCSSTK18	0.4	1.34×10^8	1.13	2.2	1.60	2.73
BCSSTK19	2.1	9.89×10^4	0.01	5.6	0.12	0.13
BCSSTK23	0.1	1.41×10^8	0.65	1.3	0.25	0.90
BCSSTK24	none	3.62×10^7	0.16	1.0	0.19	0.35
BCSSTK25	1.0	3.23×10^8	2.48	3.0	4.12	6.60
BCSSTK26	0.0	1.73×10^6	0.11	2.0	0.14	0.25
BCSSTK28	0.0	3.88×10^7	0.21	2.0	0.53	0.74
BCSSTK29	0.4	4.27×10^8	2.00	2.0	2.78	4.78
BCSSTK30	none	9.34×10^8	4.20	1.0	3.71	7.91
BCSSTK31	0.0	2.51×10^9	6.03	2.0	7.60	13.63
BCSSTK32	0.2	1.06×10^9	6.76	2.3	11.71	18.47
BCSSTK33	0.0	1.32×10^9	1.38	1.5	1.53	2.91
BCSSTK35	0.3	3.99×10^8	3.54	3.1	8.94	12.48
BCSSTK36	0.0	6.20×10^8	2.08	1.4	2.80	4.88
BCSSTK37	0.0	5.56×10^8	2.43	2.0	4.42	6.85

add some larger matrices to the test set, increasing the total from 20 to 33 matrices. Table 5.4 presents the results of our tests.

The results are quite similar to those obtained with initial orderings from MDtru. However, among the 20 smaller problems used in the previous two subsections, only 9 have all 10 initial orderings minimal, in contrast to 17 when the initial orderings are produced by MDtru. Nonetheless, for every matrix but 2, the reduction in factor flops from initial to final ordering is less than 0.5%. For BCSSTK25, an average of 3 major iterations leads to an average of 1% reduction in factor flops; for the extremely small problem BCSSTK19, an average of 5.6 major iterations leads to an average of 2.1% reduction in factor flops. The initial MMDext orderings are very nearly minimal and stand to gain very little reduction in factor storage and work by trying to squeeze

out critical edges.

5.4. Nested dissection. An ND ordering finds a node bisector of the graph of A and numbers these vertices last in the ordering. It applies this numbering process recursively to the remaining pieces (i.e., connected components) of the graph. Current implementations apply this numbering process to the graph until each of the remaining pieces has fewer than some given number of vertices. (The ordering package we use subdivides no piece with 200 or fewer vertices.) Following Ashcraft and Liu [2], we call these small pieces that remain to be labeled *domains*.

We obtained initial ND orderings from the METIS ND algorithm [11]. We executed routine `METIS_NODEND` in version 3.0.3 of METIS with the default user-supplied options (`option(0)=0`). We made one change to the algorithm in METIS; we postprocessed the ordering obtained from METIS so that each domain is ordered using the constrained minimum degree algorithm (with external degrees). Constrained minimum degree was introduced by Liu [14] and has been used in the ND algorithms of Hendrickson and Rothberg [10] and Ashcraft and Liu [2]. Constrained minimum degree applies minimum degree to the vertices of a domain, using degrees in the complete elimination graph.

The results of our tests are shown in Table 5.5.

None of the initial ND orderings are minimal; some are quite close to minimal, while others are quite far from minimal (as measured by the percent decrease in factor flops from the initial to the final orderings). For 20 of the 32 matrices there is a decrease in factor flops of 2% or greater; for 12 of the 32 matrices there is a decrease in factor flops of 4% or greater; for 5 of the 32 matrices there is a decrease in factor flops of 11% or greater. Note that the number of major iterations is only loosely connected with the percent reduction in factor flops: for example, BCSSTK16 has a 14.1% reduction in 4.7 major iterations, while BCSSTK13 has a 0.7% reduction in 6.3 major iterations.

It is known that ND does not necessarily order the vertices of the separators in the most efficient manner [2, 10, 20]. The vertices within a separator are numbered arbitrarily by ND even though one ordering of the separator may reduce fill better than another. On a more global level, an ND ordering may create significant amounts of extra fill when it is used to order matrices arising from long, narrow structures. A canonical example of this is ND applied to a path: minimum degree successively numbers leaves on the path creating no fill, while ND imposes an order on the singleton separators that creates fill. The matrix BCSSTK25 in our test set is an example of this phenomenon. It arises from a finite element model of a long narrow structure, namely, a 76-story skyscraper, and it incurs by far the greatest reduction in factor flops: 31.4%.

The times for the ND orderings are greater than those for the MMDext orderings, but they are still quite reasonable. Because the ND orderings are not so nearly minimal, the number of major iterations for the initial ND orderings are much greater than the number of major iterations for the initial MMDext orderings. Rising with the number of major iterations is the time to compute the minimal orderings, which can be quite substantial. Neither the number of major iterations nor the run times, however, approaches in magnitude the number of major iterations or the run times, respectively, for random/new-minimal.

5.5. Multisection. The MS ordering algorithm was a response to difficulties encountered using the ND ordering algorithm. An MS ordering is obtained from an ND algorithm as follows. The set of separators and domains is computed as before,

TABLE 5.5

Average factor floating point operations, average ordering times (in CPU seconds), and average number of major iterations for ND/new-minimal. Averages are taken over ten runs with randomly permuted adjacency structures.

	ND/new-minimal					
Matrix	% decrease	Final flops	ND time	Major iter.	Minimal time	Total time
CRY01	2.6	2.63×10^8	0.69	2.0	0.92	1.61
CRY02	0.3	1.87×10^9	3.09	2.0	3.33	6.42
CRY03	4.5	5.32×10^9	6.20	2.0	7.14	13.34
DIS060	2.6	1.11×10^7	0.65	2.3	0.30	0.95
DIS090	2.4	4.10×10^7	1.85	3.6	1.27	3.12
DIS120	3.5	1.05×10^8	3.73	2.9	2.31	6.04
NASA1	1.5	5.70×10^6	0.16	3.0	0.24	0.40
NASA2	1.9	2.27×10^7	0.73	5.0	1.19	1.92
NASA4	2.3	3.48×10^7	0.65	4.1	0.84	1.49
SPA060	0.0	1.52×10^7	0.62	1.6	0.21	0.83
SPA090	0.5	5.56×10^7	1.88	2.1	0.72	2.60
SPA120	0.1	1.38×10^8	3.54	2.7	2.14	5.68
BCSSTK13	0.7	5.22×10^7	0.58	6.3	0.94	1.52
BCSSTK14	0.4	7.96×10^6	0.38	1.2	0.11	0.49
BCSSTK15	1.5	8.48×10^7	1.75	6.0	1.69	3.44
BCSSTK16	14.1	1.30×10^8	0.66	4.7	2.01	2.67
BCSSTK17	14.9	1.61×10^8	2.40	6.9	5.99	8.39
BCSSTK18	2.4	8.48×10^7	3.48	11.7	8.18	11.66
BCSSTK19	14.6	9.98×10^4	0.04	19.9	0.43	0.47
BCSSTK23	1.7	9.41×10^7	0.84	10.2	2.08	2.92
BCSSTK24	2.1	3.63×10^7	0.20	2.3	0.44	0.64
BCSSTK25	31.4	2.56×10^8	5.56	12.4	15.76	21.32
BCSSTK26	6.1	1.93×10^6	0.19	7.6	0.53	0.72
BCSSTK28	4.1	4.52×10^7	0.19	5.5	1.57	1.76
BCSSTK29	4.5	3.15×10^8	5.54	8.8	12.13	17.67
BCSSTK30	11.2	1.03×10^9	7.47	8.5	30.53	38.00
BCSSTK31	1.4	1.17×10^9	11.42	8.3	32.92	44.34
BCSSTK32	5.5	1.26×10^9	9.96	9.0	46.97	56.93
BCSSTK33	1.1	8.39×10^8	3.59	2.5	2.56	6.15
BCSSTK35	5.4	4.61×10^8	3.35	11.9	34.93	38.28
BCSSTK36	2.1	6.39×10^8	1.97	6.0	11.91	13.88
BCSSTK37	7.2	5.95×10^8	3.44	7.6	17.56	21.00

and the vertices of the domains are to be ordered before the vertices of the separators as before. The domains are again eliminated using constrained minimum degree. Let X be the set of vertices obtained by forming the union of all the domains. An MS ordering is then obtained by applying minimum degree to the quotient graph representation of the elimination graph G_X. This strategy for ordering the separators has appeared in [2, 10, 20]. Our results using an MS initial ordering appear in Table 5.6.

Our results with MS ordering corroborate those reported in Ashcraft and Liu [2]. Overall, the MS ordering reduces factor flops better than either the MMDext or ND orderings. For four of the matrices, the MS initial ordering is a minimal ordering. Overall, the number of major iterations lies between the number of major iterations for the MMDext ordering and the number of major iterations for the ND ordering. Applying minimum degree to the elimination graph G_X causes the initial MS order-

TABLE 5.6

Average factor floating point operations, average ordering times (in CPU seconds), and average number of major iterations for MS/new-minimal. Averages are taken over ten runs with randomly permuted adjacency structures.

	MS/new-minimal					
Matrix	% decrease	Final flops	MS time	Major iter.	Minimal time	Total time
CRY01	0.0	2.48×10^8	1.21	1.2	0.58	1.79
CRY02	none	1.88×10^9	4.96	1.0	1.74	6.70
CRY03	0.0	5.59×10^9	10.15	1.1	3.95	14.10
DIS060	0.0	9.42×10^6	0.80	1.1	0.14	0.94
DIS090	0.0	3.26×10^7	2.34	1.6	0.56	2.90
DIS120	0.0	8.21×10^7	4.48	1.5	1.17	5.65
NASA1	0.1	5.43×10^6	0.23	1.8	0.12	0.35
NASA2	1.1	2.27×10^7	1.05	3.7	0.86	1.91
NASA4	0.1	3.52×10^7	0.86	2.4	0.49	1.35
SPA060	none	1.42×10^7	0.79	1.0	0.13	0.92
SPA090	none	5.05×10^7	2.31	1.0	0.33	2.64
SPA120	0.0	1.25×10^8	4.55	1.8	1.47	6.02
BCSSTK13	0.1	5.11×10^7	0.76	1.6	0.23	0.99
BCSSTK14	0.3	7.95×10^6	0.49	1.2	0.11	0.60
BCSSTK15	0.1	8.51×10^7	2.23	2.7	0.85	3.08
BCSSTK16	0.6	1.25×10^8	1.13	2.6	1.11	2.24
BCSSTK17	0.5	1.46×10^8	3.38	3.3	3.03	6.41
BCSSTK18	0.7	8.37×10^7	4.42	7.8	5.74	10.16
BCSSTK19	14.6	9.98×10^4	0.07	19.9	0.47	0.54
BCSSTK23	0.2	9.67×10^7	1.08	5.9	1.16	2.24
BCSSTK24	none	3.47×10^7	0.40	1.0	0.19	0.59
BCSSTK25	1.3	2.39×10^8	7.48	8.7	11.42	18.90
BCSSTK26	3.4	1.91×10^6	0.28	6.3	0.43	0.71
BCSSTK28	0.1	3.62×10^7	0.45	2.6	0.70	1.15
BCSSTK29	0.5	3.41×10^8	7.19	7.0	9.89	17.08
BCSSTK30	0.0	8.67×10^8	11.21	4.6	15.79	27.00
BCSSTK31	0.1	1.46×10^9	15.71	5.3	20.45	36.16
BCSSTK32	0.2	9.22×10^8	14.94	5.0	24.57	39.51
BCSSTK33	0.0	7.55×10^8	4.61	1.2	1.22	5.83
BCSSTK35	0.4	3.77×10^8	6.24	4.4	13.15	19.39
BCSSTK36	0.0	4.99×10^8	4.12	2.9	5.77	9.89
BCSSTK37	0.1	4.66×10^8	5.87	4.4	9.90	15.77

ings to be much closer to minimal than the initial ND orderings were. For all but four matrices, the reduction in factor flops is under 1.0%. For three of these four matrices the reduction is small: 1.1% for NASA2910, 1.3% for BCSSTK25, and 3.4% for BCSSTK26. The only problem for which there is a large reduction is the tiny problem BCSSTK19, and here we are merely obtaining exactly the same results that we obtained for the initial ND ordering.

The MS run times are simply ND run times with the time for ordering G_X by minimum degree added in. The code was written for ease of programming and not for optimal time efficiency; still the MS run times are reasonably small. Because the major iterations are reduced in number, the total run times are generally smaller than those for ND/new-minimal.

TABLE 5.7
Ordering times (in CPU seconds) for MMDext/new-minimal and MMDext/Blair-et-al. Times are for one run without permuting adjacency structure.

	MMDext/new-minimal		MMDext/Blair-et-al	
Matrix	MMDext time	Minimal time	MMDext time	Minimal time
DIS060	0.01	0.01	0.01	0.37
DIS090	0.02	0.03	0.03	1.40
DIS120	0.03	0.05	0.05	3.44
NASA1824	0.01	0.01	0.01	0.18
NASA2910	0.01	0.04	0.02	0.67
NASA4704	0.02	0.02	0.03	1.17
SPA060	0.01	0.01	0.01	0.52
SPA090	0.02	0.03	0.03	1.99
SPA120	0.03	0.05	0.05	5.09
BCSSTK13	0.02	0.01	0.03	1.42
BCSSTK14	0.01	0.01	0.01	0.33
BCSSTK15	0.05	0.05	0.07	4.96
BCSSTK16	0.02	0.06	0.03	4.52
BCSSTK17	0.06	0.11	0.08	4.94
BCSSTK18	0.08	0.11	0.11	4.57
BCSSTK19	0.00	0.01	0.00	0.01
BCSSTK23	0.05	0.02	0.08	2.86
BCSSTK24	0.01	0.02	0.01	1.11
BCSSTK25	0.14	0.24	0.20	10.44
BCSSTK26	0.01	0.01	0.01	0.08

5.6. Comparison with algorithm of Blair, Heggernes, and Telle. Because the algorithm of Blair, Heggernes, and Telle is implemented as a Fortran 90 code, we moved the codes to a machine that has a modern Fortran 90 compiler. We ran the tests on a Compaq Alphaserver SC; we used Fortran compiler `f77` to compile the new code developed for this paper, and we used Fortran compiler `f90` to compile the code of Blair, Heggernes, and Telle. We used compiler optimization level `-O` for all compilations.

Timings for both methods applied to MMDext and ND initial orderings appear in Tables 5.7 and 5.8, respectively. We use the same set of 20 smaller test problems used earlier in this section. To make corresponding initial orderings identical for the two codes, it was necessary to remove random reordering of the adjacency structure from the new code. Moreover, to make corresponding initial ND orderings identical for the two codes, it was necessary to remove the reordering of domains by minimum degree from the new code. We focus solely on timings; the fill for corresponding initial orderings is always identical, and the fill for corresponding final orderings is always quite close to one another. The ND initial orderings are computed by the same METIS routine. By contrast, the MMDext initial orderings are computed using separate routines (one a Fortran 77 routine and the other a Fortran 90 routine). Note, however, that they compute identical corresponding initial orderings. Observe in Table 5.7 that Fortran 77 MMDext orderings are obtained somewhat more efficiently than Fortran 90 MMDext orderings.

The primary observation in the tables is that, for these problems, the new minimal ordering code is much faster than the minimal ordering code of Blair, Heggernes, and

TABLE 5.8
Ordering times (in CPU seconds) for ND/new-minimal and ND/Blair-et-al. Times are for one run without permuting adjacency structure.

	ND/new-minimal		ND/Blair-et-al	
Matrix	ND time	Minimal time	ND time	Minimal time
DIS060	0.07	0.06	0.06	0.59
DIS090	0.20	0.18	0.20	2.23
DIS120	0.33	0.28	0.33	5.49
NASA1824	0.02	0.01	0.02	0.21
NASA2910	0.08	0.04	0.08	0.72
NASA4704	0.06	0.05	0.05	1.40
SPA060	0.07	0.04	0.06	0.71
SPA090	0.21	0.14	0.20	2.78
SPA120	0.35	0.22	0.34	5.80
BCSSTK13	0.07	0.04	0.07	1.84
BCSSTK14	0.04	0.04	0.04	0.39
BCSSTK15	0.16	0.10	0.17	4.26
BCSSTK16	0.07	0.10	0.07	4.69
BCSSTK17	0.26	0.33	0.26	7.32
BCSSTK18	0.34	0.59	0.33	4.94
BCSSTK19	0.00	0.04	0.00	0.02
BCSSTK23	0.08	0.18	0.08	4.08
BCSSTK24	0.02	0.06	0.02	1.16
BCSSTK25	0.52	1.10	0.51	20.19
BCSSTK26	0.02	0.06	0.02	0.13

Telle. For these problems, the new minimal ordering code runs in time comparable to the initial ordering time, while the code of Blair, Heggernes, and Telle runs in time consistently one order of magnitude to one-and-a-half orders of magnitude larger than the initial ordering time. The comparison, however, is very preliminary. The minimum degree algorithm, on which the new algorithm is based, has been scrutinized for years for ways to improve its efficiency; by contrast, the code of Blair, Heggernes, and Telle is a straightforward first cut at implementing their algorithm. It is likely that its performance can be significantly improved. One obvious notion to explore is the use of supernodes.

6. Concluding remarks. We devised a new characterization of candidate edges that leads to a simple "block-restricted minimum degree" elimination process to remove candidate edges. We then devised an algorithm that removes candidate edges until a minimal chordal supergraph and a minimal ordering are obtained. Empirically, the method MDtru/new-minimal improves on both the ordering quality and the ordering time of the old method LEX M.

In the past, minimal orderings were not good heuristic orderings; they did not approximate minimum orderings well [19, p. 282]. A new approach seen in earlier work [3, 5, 6, 7] and also used in this paper deals with this shortcoming of past minimal ordering algorithms. Because it starts with any initial fill graph and refines that fill graph until minimality is achieved, the new approach can turn good heuristic orderings into good minimal orderings.

However, we saw that this capability makes little or no difference when the initial ordering is produced by MDtru. It also makes little difference when the initial ordering

is produced by MS or MMDext. Each of these ordering heuristics produces orderings that are nearly minimal, and trying to improve them by making them minimal is justified solely by ensuring that equivalent orderings are truly equivalent. We did see, however, that initial ND orderings can be quite far from minimal. MS can be viewed as a way to fix this problem with ND.

The algorithms in Berry [3] and Dahlhaus [7] are quite distinct from those in Blair, Heggernes, and Telle [5, 6] and this paper. The former arise from links between minimal vertex separators and minimal fill graphs, while the latter arise explicitly from the existence of critical edges in nonminimal fill graphs. One of the variants mentioned in Berry [3] promises much better run times than LEX M [4].

Among the contributions of this work are the following. The algorithm runs quite fast in our tests. This has enabled computation of minimal orderings using several different initial orderings of many large, standard test problems. Hence we were able to confirm and extend the range of the results reported by Blair, Heggernes, and Telle [6] for MMDext and METIS ND initial orderings. The algorithm is most efficient when the initial ordering is very close to minimal. This is due in large part to the fact that the number of major steps cannot exceed the total number of candidate edges removed. When the initial ordering is minimal, only one major step is required to detect minimality. This is one of the more distinct and appealing features of the algorithm.

The time complexity of the new minimal ordering algorithm is unknown, in part because of the use of minimum degree and quotient graphs, and in part because a bound on the number of major steps in the new algorithm is unknown. We conjecture that the number of major steps is $O(n)$. We also conjecture that it may be worthwhile to explore more efficient implementations of the algorithm, especially in the case where the initial ordering is MDint or MMDext. It may be possible to exploit features of the minimum degree ordering process to know in advance that a supernode has no candidate edges, and hence there is no need to order vertices within such a supernode.

Acknowledgments. The author thanks Bengt Aspvall and Jean Blair and her family for their gracious hospitality during a visit to Bergen, Norway a few years ago. In the stimulating environment provided during that visit, the ideas behind this paper began to come together. The author is grateful to Joseph Liu for access to his multiple minimum degree code. The author thanks one referee for pointing out previous work in references [3, 5, 6, 7], the other referee for suggesting a section along the line of section 2.3, and both referees for other suggestions that improved the presentation of the paper. The author especially thanks Jean Blair, Pinar Heggernes, and Jan Arne Telle for generously providing their code for comparison. The author is grateful for the support given by Oak Ridge National Laboratory's Computer Science and Mathematics Division and the U.S. Department of Energy's Applied Mathematical Sciences subprogram, which allowed the project to be seen through to its conclusion.

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] C. Ashcraft and J. W. H. Liu, *Robust ordering of sparse matrices using multisection*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 816–832.

[3] A. Berry, *A wide-range efficient algorithm for minimal triangulation*, in Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1999, pp. S860–S861.

[4] A. Berry and P. Heggernes, *A Wide-Range Algorithm for Minimal Triangulation from an Arbitrary Ordering*, in preparation.
[5] J. Blair, P. Heggernes, and J. Telle, *Making an arbitrary filled graph minimal by removing fill edges*, in Algorithm Theory—SWAT'96, R. Karlsson and A. Lingas, eds., Lecture Notes in Comput. Sci. 1097, Springer, New York, 1996, pp. 173–184.
[6] J. Blair, P. Heggernes, and J. Telle, *A practical algorithm for making filled graphs minimal*, Theoret. Comput. Sci., 250 (2000), pp. 125–141.
[7] E. Dahlhaus, *Minimal elimination ordering inside a given chordal graph*, in Graph Theoretical Concepts in Computer Science, Lecture Notes in Comput. Sci. 1335, Springer, New York, 1997, pp. 132–143.
[8] A. George and J. W.-H. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Rev., 31 (1989), pp. 1–19.
[9] J. R. Gilbert, E. G. Ng, and B. W. Peyton, *An efficient algorithm to compute row and column counts for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1075–1091.
[10] B. Hendrickson and E. Rothberg, *Improving the run time and quality of nested dissection ordering*, SIAM J. Sci. Comput., 20 (1999), pp. 468–489.
[11] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1999), pp. 359–392.
[12] J. W. H. Liu, E. G. Ng, and B. W. Peyton, *On finding supernodes for sparse matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 242–252.
[13] J. W.-H. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.
[14] J. W.-H. Liu, *The minimum degree ordering with constraints. Sparse matrix algorithms on supercomputers*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1136–1145.
[15] J. W.-H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
[16] T. Ohtsuki, *A fast algorithm for finding an optimal ordering in the vertex elimination on a graph*, SIAM J. Comput., 5 (1976), pp. 133–145.
[17] D. Rose, *Triangulated graphs and the elimination process*, J. Math. Anal. Appl., 32 (1970), pp. 597–609.
[18] D. Rose, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, New York, 1972, pp. 183–217.
[19] 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.
[20] E. Rothberg, *Robust Ordering of Sparse Matrices: A Minimum Degree Nested Dissection Hybrid*, Silicon Graphics manuscript, 1996.
[21] R. Schreiber, *A new implementation of sparse Gaussian elimination*, ACM Trans. Math. Software, 8 (1982), pp. 256–276.
[22] M. Yannakakis, *Computing the minimum fill-in is NP-complete*, SIAM J. Alg. Discrete Methods, 2 (1981), pp. 77–79.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 1, pp. 295–302
© 2001 Society for Industrial and Applied Mathematics

CONVERGENCE RATE OF AN ITERATIVE METHOD FOR A NONLINEAR MATRIX EQUATION*

CHUN-HUA GUO†

Abstract. We prove a convergence result for an iterative method, proposed recently by Meini, for finding the maximal Hermitian positive definite solution of the matrix equation $X+A^*X^{-1}A=Q$, where Q is Hermitian positive definite.

Key words. matrix equation, maximal Hermitian solution, cyclic reduction, iterative methods, convergence rate

AMS subject classifications. 15A24, 65F10, 65H10

PII. S0895479800374017

1. Introduction. Nonlinear matrix equations occur in many applications. Examples of these equations are algebraic Riccati equations of continuous or discrete type, which have been studied extensively and have been the subject of the monographs [12] and [13]. Another example is the quadratic matrix equation

$$AX^2 + BX + C = 0, \tag{1.1}$$

where A, B, C are given coefficient matrices. This equation has also been the topic of many papers, including two recent papers by Higham and Kim (see [9] and [10]). In this paper, our interest is in the matrix equation

$$X + A^*X^{-1}A = Q, \tag{1.2}$$

where $A, Q \in \mathbb{C}^{m\times m}$ with Q Hermitian positive definite and a Hermitian positive definite solution is required. This equation has been studied recently by several authors (see [1], [4], [5], [8], [14], [16], [17]). For the application areas in which the equation arises, see the references given in [1]. Note also that a solution X of (1.2) is such that the Schur complement of X in the matrix

$$\begin{pmatrix} X & A \\ A^* & Q \end{pmatrix}$$

is X itself (see [1]).

There is some connection between (1.2) and (1.1). For example, if X is a solution of (1.2), then $X^{-1}A$ is a solution of $A^*Y^2 - QY + A = 0$. Equation (1.2) is also a special case of the discrete algebraic Riccati equation

$$-X + C^*XC + Q - (A + B^*XC)^*(R + B^*XB)^{-1}(A + B^*XC) = 0 \tag{1.3}$$

with $C = R = 0$ and $B = I$.

*Received by the editors June 20, 2000; accepted for publication by V. Mehrmann February 8, 2001; published electronically August 3, 2001. This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.
http://www.siam.org/journals/simax/23-1/37401.html
†Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca).

For Hermitian matrices X and Y, we write $X \geq Y$ ($X > Y$) if $X - Y$ is positive semidefinite (definite). A Hermitian solution X_+ of a matrix equation is called maximal if $X_+ \geq X$ for any Hermitian solution X of the matrix equation.

For algebraic Riccati equations, it is well known that the desirable solution is the maximal solution. For (1.2), the maximal solution is also the right choice in view of the presence of X^{-1} in the equation.

The main purpose of the paper is to prove a convergence result for an iterative method proposed by Meini [14] for finding the maximal solution of (1.2). Roughly speaking, our new result together with the results obtained in [14] shows that the convergence of Meini's method is no slower than Newton's method. Meini's method is thus preferable when we try to find the maximal solution of (1.2), since the computational work per iteration for Newton's method is $5 \sim 10$ times that for Meini's method. To put our result in a proper setting, we review in section 2 some theoretical results for the solution of (1.2) and present in section 3 three iterative methods, with emphasis on Meini's method. Our convergence result for Meini's method is then presented in section 4. The paper ends with some discussions in section 5.

2. Theoretical background. Necessary and sufficient conditions for the existence of a positive definite solution of (1.2) have been given in [5].

THEOREM 2.1. *Equation* (1.2) *has a positive definite solution if and only if the rational matrix function* $\psi(\lambda) = \lambda A + Q + \lambda^{-1}A^*$ *is regular (i.e., the determinant of* $\psi(\lambda)$ *is not identically zero) and* $\psi(\lambda) \geq 0$ *for all* λ *on the unit circle.*

The existence of the maximal solution of (1.2) has also been established in [5], along with a characterization of the maximal solution.

THEOREM 2.2. *If* (1.2) *has a positive definite solution, then it has a maximal solution* X_+*. Moreover,* X_+ *is the unique positive definite solution such that* $X + \lambda A$ *is nonsingular for all* λ *with* $|\lambda| < 1$*.*

This result has the following immediate corollary, where $\rho(\cdot)$ is the spectral radius.

COROLLARY 2.3. *For the maximal solution* X_+ *of* (1.2)*,* $\rho(X_+^{-1}A) \leq 1$*; for any other positive definite solution* X*,* $\rho(X^{-1}A) > 1$*.*

We also have the following characterization for the eigenvalues of the matrix $X_+^{-1}A$ (see [8]).

THEOREM 2.4. *For* (1.2)*, the eigenvalues of* $X_+^{-1}A$ *are precisely the eigenvalues of the matrix pencil*

$$\lambda \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -I & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & -I \\ Q & -I & A^* \\ -A & 0 & 0 \end{pmatrix}$$

inside or on the unit circle, with half of the partial multiplicities for each eigenvalue on the unit circle.

3. Iterative methods. The maximal solution X_+ of (1.2) can be found by the following basic fixed point iteration.

ALGORITHM 3.1.

$$\begin{aligned} X_0 &= Q, \\ X_{n+1} &= Q - A^* X_n^{-1} A, \quad n = 0, 1, \ldots. \end{aligned}$$

For Algorithm 3.1, we have $X_0 \geq X_1 \geq \cdots$, and $\lim_{n\to\infty} X_n = X_+$ (see, e.g., [5]). Moreover, the following result is proved in [8].

THEOREM 3.2. *Let* $\{X_n\}$ *be given by Algorithm* 3.1. *Then*

$$\limsup_{n\to\infty} \sqrt[n]{\|X_n - X_+\|} \le (\rho(X_+^{-1}A))^2,$$

where $\|\cdot\|$ *is any matrix norm.*

Note that $\rho(X_+^{-1}A) \le 1$ is always true by Corollary 2.3. From the above result, we know that the convergence of the fixed point iteration is R-linear whenever $\rho(X_+^{-1}A) < 1$. For detailed definitions of the rates of convergence, see [15]. If $\rho(X_+^{-1}A) = 1$, the convergence of the fixed point iteration is typically sublinear. Therefore, the convergence of Algorithm 3.1 would be excruciatingly slow when $X_+^{-1}A$ has eigenvalues on, or near, the unit circle. Naturally, one would turn to Newton's method for help with this situation.

Newton's method is studied in [7] for the discrete algebraic Riccati equation of the form (1.3). For (1.2), a special case of (1.3), Newton's method is as follows (see [8]).

ALGORITHM 3.3 (Newton's method for (1.2)). *Take* $X_0 = Q$. *For* $n = 1, 2, \ldots$, *compute* $L_n = X_{n-1}^{-1}A$ *and solve*

$$X_n - L_n^* X_n L_n = Q - 2L_n^* A. \tag{3.1}$$

Note that the Stein equation (3.1) is uniquely solvable when $\rho(L_n) < 1$. The convergence behavior of Algorithm 3.3 is described in [8].

THEOREM 3.4. *If* (1.2) *has a positive definite solution, then Algorithm* 3.3 *determines a sequence of Hermitian matrices* $\{X_n\}_{n=0}^{\infty}$ *for which* $\rho(L_n) < 1$ *for* $n = 0, 1, \ldots, X_0 \ge X_1 \ge \cdots$, *and* $\lim_{n\to\infty} X_n = X_+$. *The convergence is quadratic if* $\rho(X_+^{-1}A) < 1$. *If* $\rho(X_+^{-1}A) = 1$ *and all eigenvalues of* $X_+^{-1}A$ *on the unit circle are semisimple (i.e., all elementary divisors associated with these eigenvalues are linear), then the convergence is either quadratic or linear with rate* $1/2$.

Recently, Meini proposed a new algorithm by following the strategy successfully devised in [2], [3] for solving nonlinear matrix equations arising in Markov chains. Her algorithm is described below.

For the maximal solution X_+ of (1.2),

$$-I + QX_+^{-1} - A^* X_+^{-1} A X_+^{-1} = 0, \tag{3.2}$$

and the matrix $G_+ = X_+^{-1}A$ satisfies

$$-A + QG_+ - A^* G_+^2 = 0. \tag{3.3}$$

Equations (3.2) and (3.3) can be rewritten as

$$\begin{pmatrix} Q & -A^* & & 0 \\ -A & Q & -A^* & \\ & -A & Q & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} I \\ G_+ \\ G_+^2 \\ \vdots \end{pmatrix} X_+^{-1} = \begin{pmatrix} I \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \tag{3.4}$$

The cyclic reduction algorithm is then applied to (3.4). This consists of performing an even-odd permutation of the block rows and columns, followed by one step of block Gaussian elimination on the resulting 2×2 block system. This results in a reduced

system with a structure similar to (3.4). Repeated application of the cyclic reduction algorithm generates the following sequence of systems:

$$\begin{pmatrix} X_n & -A_n^* & & 0 \\ -A_n & Q_n & -A_n^* & \\ & -A_n & Q_n & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} I \\ G_+^{2^n} \\ G_+^{2\cdot 2^n} \\ \vdots \end{pmatrix} X_+^{-1} = \begin{pmatrix} I \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \quad n = 0, 1, \ldots, \tag{3.5}$$

where the matrices A_n, Q_n, and X_n are recursively defined as follows.

ALGORITHM 3.5.

$$\begin{aligned} &A_0 = A, \quad Q_0 = X_0 = Q, \\ &A_{n+1} = A_n Q_n^{-1} A_n, \\ &Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^* - A_n^* Q_n^{-1} A_n, \\ &X_{n+1} = X_n - A_n^* Q_n^{-1} A_n, \quad n = 0, 1, \ldots. \end{aligned}$$

Meini proposed using the above algorithm to find the maximal solution X_+ and she proved the following result (see [14]).

THEOREM 3.6. *For the matrices Q_n and X_n in Algorithm* 3.5*, it holds that $Q_n \geq Q_{n+1} > 0$, $X_n \geq X_{n+1} > 0$ $(n = 0, 1, \ldots)$. Moreover, if $\rho(X_+^{-1}A) < 1$, then the sequence $\{X_n\}$ converges to X_+ quadratically.*

Meini's method and Newton's method are most useful when $\rho(X_+^{-1}A)$ is close to 1, since otherwise the basic fixed point iteration is adequate. It is therefore important to investigate the convergence behavior of Meini's method when $\rho(X_+^{-1}A) = 1$.

4. Convergence rate. In this section, we prove a convergence result for Algorithm 3.5 when $\rho(X_+^{-1}A) = 1$. In our proof, we will need the following two equations from (3.5):

$$X_n - X_+ = A_n^* G_+^{2^n}, \tag{4.1}$$

$$-A_n + Q_n G_+^{2^n} - A_n^* G_+^{2\cdot 2^n} = 0. \tag{4.2}$$

These two equations were also used in Meini's proof of Theorem 3.6.

THEOREM 4.1. *If $\rho(X_+^{-1}A) = 1$ and all eigenvalues of $X_+^{-1}A$ on the unit circle are semisimple, then the sequence $\{X_n\}$ produced by Algorithm* 3.5 *converges to X_+ and the convergence is at least linear with rate* $1/2$.

Proof. By Theorem 3.6 the sequence $\{X_n\}$ is monotonically decreasing and bounded below and hence has a limit. Therefore, $\lim_{n\to\infty} A_n^* Q_n^{-1} A_n = 0$ by the last equation in Algorithm 3.5. Since $\|Q\|_2 I \geq Q \geq Q_n > 0$, $Q_n^{-1} \geq I/\|Q\|_2$. Thus,

$$0 \leq A_n^* A_n / \|Q\|_2 \leq A_n^* Q_n^{-1} A_n.$$

Therefore, $\lim_{n\to\infty} A_n = 0$. Since all eigenvalues of $G_+ = X_+^{-1}A$ on the unit circle are semisimple by assumption, the sequence $\{G_+^{2^n}\}$ is bounded. It follows from (4.1) that $\lim_{n\to\infty} X_n = X_+$. As a result, $X_n \geq X_+$ for each $n \geq 0$.

To prove the assertion about the convergence rate, we need to make some simplifications. Let $P^{-1}G_+P = J$ be the Jordan canonical form of G_+. Accordingly, let

$$B_n = P^* A_n P, \quad R_n = P^* Q_n P, \quad Y_n = P^* X_n P, \quad Y_+ = P^* X_+ P.$$

It is easily verified that for $n = 0, 1, \ldots,$

$$B_{n+1} = B_n R_n^{-1} B_n, \tag{4.3}$$

$$R_{n+1} = R_n - B_n R_n^{-1} B_n^* - B_n^* R_n^{-1} B_n, \tag{4.4}$$

$$Y_{n+1} = Y_n - B_n^* R_n^{-1} B_n, \tag{4.5}$$

and

$$Y_n - Y_+ = B_n^* J^{2^n}, \tag{4.6}$$

$$-B_n + R_n J^{2^n} - B_n^* J^{2\cdot 2^n} = 0. \tag{4.7}$$

We may assume that

$$J = \operatorname{diag}(e^{i\theta_1}, \ldots, e^{i\theta_k}, J_<),$$

where $e^{i\theta_1}, \ldots, e^{i\theta_k}$ are the eigenvalues of G_+ on the unit circle (not necessarily distinct) and $J_<$ consists of Jordan blocks associated with the eigenvalues of G_+ inside the unit disk. Let the corresponding block diagonals of $Y_n - Y_+$ and B_n be

$$\operatorname{diag}(\alpha_n^{(1)}, \ldots, \alpha_n^{(k)}, Z_n) \tag{4.8}$$

and

$$\operatorname{diag}(\beta_n^{(1)}, \ldots, \beta_n^{(k)}, C_n),$$

respectively. Since $Y_n - Y_+ = P^*(X_n - X_+)P$ is positive semidefinite, $\alpha_n^{(i)} \geq 0$ for $i = 1, \ldots, k$.

We first examine how fast $\alpha_n^{(1)}$ converges to zero. By (4.6) we have

$$\alpha_n^{(1)} = \overline{\beta_n^{(1)}} e^{i2^n\theta_1}. \tag{4.9}$$

To find the relation between $\beta_{n+1}^{(1)}$ and $\beta_n^{(1)}$ from (4.3), we let

$$B_n = \begin{pmatrix} \beta_n^{(1)} & t_n^* \\ s_n & U_n \end{pmatrix}, \quad R_n = \begin{pmatrix} \gamma_n & v_n^* \\ v_n & W_n \end{pmatrix},$$

where $s_n, t_n, v_n \in \mathbb{C}^{m-1}$ and $U_n, W_n \in \mathbb{C}^{(m-1)\times(m-1)}$. Since R_n is positive definite, it is well known (see [11], for example) that $H_n = W_n - \frac{1}{\gamma_n} v_n v_n^*$, the Schur complement of γ_n in R_n, is also positive definite, and

$$R_n^{-1} = \begin{pmatrix} \frac{1}{\gamma_n} + \frac{1}{\gamma_n^2} v_n^* H_n^{-1} v_n & -\frac{1}{\gamma_n} v_n^* H_n^{-1} \\ -\frac{1}{\gamma_n} H_n^{-1} v_n & H_n^{-1} \end{pmatrix}.$$

Now, a straightforward computation shows that

$$\beta_{n+1}^{(1)} = \frac{(\beta_n^{(1)})^2}{\gamma_n} + \frac{(\beta_n^{(1)})^2}{\gamma_n^2} v_n^* H_n^{-1} v_n - \frac{\beta_n^{(1)}}{\gamma_n} v_n^* H_n^{-1} s_n - \frac{\beta_n^{(1)}}{\gamma_n} t_n^* H_n^{-1} v_n + t_n^* H_n^{-1} s_n.$$

By (4.7), we have

$$-\beta_n^{(1)} + \gamma_n e^{i2^n\theta_1} - \overline{\beta_n^{(1)}} e^{i2\cdot 2^n\theta_1} = 0 \tag{4.10}$$

and

$$-s_n + v_n e^{i2^n\theta_1} - t_n e^{i2\cdot 2^n\theta_1} = 0. \tag{4.11}$$

Since $\alpha_n^{(1)}$ is real, we have by (4.9) $\overline{\beta_n^{(1)}} e^{i2^n\theta_1} = \beta_n^{(1)} e^{-i2^n\theta_1}$. Thus, $\overline{\beta_n^{(1)}} e^{i2\cdot 2^n\theta_1} = \beta_n^{(1)}$. It follows from (4.10) that

$$\frac{\beta_n^{(1)}}{\gamma_n} = \frac{1}{2} e^{i2^n\theta_1}.$$

The relation between $\beta_{n+1}^{(1)}$ and $\beta_n^{(1)}$ is thus simplified to

$$\begin{aligned}\beta_{n+1}^{(1)} = {} & \frac{1}{2} e^{i2^n\theta_1}\beta_n^{(1)} + \frac{1}{4} e^{i2\cdot 2^n\theta_1} v_n^* H_n^{-1} v_n - \frac{1}{2} e^{i2^n\theta_1} v_n^* H_n^{-1} s_n \\ & - \frac{1}{2} e^{i2^n\theta_1} t_n^* H_n^{-1} v_n + t_n^* H_n^{-1} s_n.\end{aligned}$$

Multiplying both sides by $e^{-i2^{n+1}\theta_1}$, we get

$$\begin{aligned}\alpha_{n+1}^{(1)} = {} & \frac{1}{2}\alpha_n^{(1)} + \frac{1}{4} v_n^* H_n^{-1} v_n - \frac{1}{2} e^{-i2^n\theta_1} v_n^* H_n^{-1} s_n \\ & - \frac{1}{2} e^{-i2^n\theta_1} t_n^* H_n^{-1} v_n + e^{-i2\cdot 2^n\theta_1} t_n^* H_n^{-1} s_n.\end{aligned}$$

Substituting $s_n = v_n e^{i2^n\theta_1} - t_n e^{i2\cdot 2^n\theta_1}$ (from (4.11)) into the above identity, we get after some manipulations that

$$\alpha_{n+1}^{(1)} = \frac{1}{2}\alpha_n^{(1)} - \left(t_n - \frac{1}{2} e^{-i2^n\theta_1} v_n\right)^* H_n^{-1} \left(t_n - \frac{1}{2} e^{-i2^n\theta_1} v_n\right).$$

Therefore, $\alpha_{n+1}^{(1)} \le \frac{1}{2}\alpha_n^{(1)}$ for each $n \ge 0$. Thus,

$$\alpha_n^{(1)} \le \frac{1}{2^n}\alpha_0^{(1)}, \quad n = 0, 1, \ldots.$$

Using appropriate permutations, we can show that, for $i = 2, \ldots, k$,

$$\alpha_n^{(i)} \le \frac{1}{2^n}\alpha_0^{(i)}, \quad n = 0, 1, \ldots.$$

The matrices Z_n in (4.8) are positive semidefinite and it is shown below that $\{Z_n\}$ converges to the zero matrix quadratically. Note that $Z_n = C_n^* J_<^{2^n}$ by (4.6). Fix an $\epsilon > 0$ such that $\rho(J_<) + \epsilon < 1$ and choose a norm $\|\cdot\|_\epsilon$ such that $\|J_<\|_\epsilon \le \rho(J_<) + \epsilon$. Since $\lim_{n\to\infty} C_n = 0$ and all matrix norms are equivalent, $\|Z_n\|_2 \le c_1(\rho(J_<) + \epsilon)^{2^n}$ for some constant c_1.

Now, noting that $\operatorname{trace}(Z_n) \le (m-k)\|Z_n\|_2$,

$$\begin{aligned}\|X_n - X_+\|_2 &= \|P^{-*}(Y_n - Y_+)P^{-1}\|_2 \\ &\le \|P^{-*}\|_2\|Y_n - Y_+\|_2\|P^{-1}\|_2 \\ &\le \|P^{-*}\|_2\|P^{-1}\|_2 \operatorname{trace}(Y_n - Y_+) \\ &= \|P^{-*}\|_2\|P^{-1}\|_2\big(\alpha_n^{(1)} + \cdots + \alpha_n^{(k)} + \operatorname{trace}(Z_n)\big) \\ &\le c_2\left(\frac{1}{2^n} + (\rho(J_<) + \epsilon)^{2^n}\right)\end{aligned}$$

for some constant c_2. Thus,

$$\limsup_{n\to\infty} \sqrt[n]{\|X_n - X_+\|} \le \frac{1}{2}$$

for any matrix norm $\|\cdot\|$. □

5. Discussions. In section 3, we reviewed three iterative methods for finding the maximal solution of (1.2). For Newton's method, (3.1) can be solved by a complex version of the algorithm described in [6]. Meini's method and the basic fixed point iteration can be implemented easily. The computational work per iteration for Meini's method is roughly twice that for the fixed point iteration, while the computational work per iteration for Newton's method is about 15 times that for the fixed point iteration. With the establishment of Theorem 4.1, we have had a much better picture for the convergence behavior of all three methods. When $\rho(X_+^{-1}A) < 1$, the convergence of the fixed point iteration is linear and the convergence of Newton's method and Meini's method is quadratic. When $\rho(X_+^{-1}A) = 1$, the convergence of the fixed point iteration is typically sublinear, while the convergence of Newton's method and Meini's method is at least linear with rate 1/2 provided that all unimodular eigenvalues of $X_+^{-1}A$ are semisimple. (We conjecture that the convergence is *exactly* linear with rate 1/2 for both methods.) When $X_+^{-1}A$ has nonsemisimple unimodular eigenvalues, Newton's method is still convergent, but the rate of convergence is only conjectured to be $1/\sqrt[p]{2}$, where p is the size of the largest Jordan blocks associated with unimodular eigenvalues of $X_+^{-1}A$. The conjecture was made in [7] for (1.3), which includes (1.2) as a special case. Also, when $X_+^{-1}A$ has nonsemisimple unimodular eigenvalues, it is still not known whether the sequence $\{X_n\}$ produced by Meini's method will converge to X_+.

When $\rho(X_+^{-1}A) = 1$, we cannot expect Newton's method to approximate X_+ with full accuracy since the linear system (3.1) is eventually nearly singular. Meini's method has the same problem in this case: $\tilde{Q} = \lim_{n\to\infty} Q_n$ is necessarily singular in this case. In fact, it follows easily from (4.7) that $\tilde{R} = \lim_{n\to\infty} R_n$ is singular. Therefore, $\tilde{Q} = P^{-*}\tilde{R}P^{-1}$ is singular as well.

Our final comments are about test examples for the iterative methods we discussed. In [14], Meini gives an example of (1.2) with $Q = I$ and A Hermitian, in which case an analytical expression is available for the maximal solution. More informative examples can be generated as follows. First note that we may assume that $X_+ = I$ (otherwise we can premultiply and postmultiply the equation by $X_+^{-1/2}$). The test examples will thus be of the form $X + A^*X^{-1}A = I + A^*A$. By Corollary 2.3, I is the maximal solution of this equation if and only if $\rho(A) \le 1$. We can then produce a lot of test examples by taking $A = S/r$ with $r \ge \rho(S)$ and S a random matrix. For test examples generated in this way, the convergence behavior of the three methods is very much similar to that reported by Meini [14] for her example. In fact, it is all but certain that all those examples are covered by the theory we have available by now. Note, however, that all three methods will run into difficulties when A is a large Jordan block with eigenvalue 1, for example.

REFERENCES

[1] W. N. Anderson, Jr., T. D. Morley, and G. E. Trapp, *Positive solutions to $X = A - BX^{-1}B^*$*, Linear Algebra Appl., 134 (1990), pp. 53–62.

[2] D. Bini and B. Meini, *On the solution of a nonlinear matrix equation arising in queueing problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 906–926.

[3] D. A. Bini and B. Meini, *Effective methods for solving banded Toeplitz systems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 700–719.
[4] J. C. Engwerda, *On the existence of a positive definite solution of the matrix equation* $X + A^T X^{-1} A = I$, Linear Algebra Appl., 194 (1993), pp. 91–108.
[5] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer, *Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation* $X + A^* X^{-1} A = Q$, Linear Algebra Appl., 186 (1993), pp. 255–275.
[6] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler, *Solution of the Sylvester matrix equation* $AXB^T + CXD^T = E$, ACM Trans. Math. Software, 18 (1992), pp. 223–231.
[7] C.-H. Guo, *Newton's method for discrete algebraic Riccati equations when the closed-loop matrix has eigenvalues on the unit circle*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 279–294.
[8] C.-H. Guo and P. Lancaster, *Iterative solution of two matrix equations*, Math. Comp., 68 (1999), pp. 1589–1603.
[9] N. J. Higham and H.-M. Kim, *Solving a quadratic matrix equation by Newton's method with exact line searches*, SIAM J. Matrix Anal. Appl., to appear.
[10] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.
[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[12] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, 1995.
[13] V. L. Mehrmann, *The Autonomous Linear Quadratic Control Problem*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Berlin, 1991.
[14] B. Meini, *Efficient computation of the extreme solutions of* $X + A^* X^{-1} A = Q$ *and* $X - A^* X^{-1} A = Q$, Math. Comp., to appear.
[15] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[16] X. Zhan, *Computing the extremal positive definite solutions of a matrix equation*, SIAM J. Sci. Comput., 17 (1996), pp. 1167–1174.
[17] X. Zhan and J. Xie, *On the matrix equation* $X + A^T X^{-1} A = I$, Linear Algebra Appl., 247 (1996), pp. 337–345.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 303–316
© 2001 Society for Industrial and Applied Mathematics

SOLVING A QUADRATIC MATRIX EQUATION BY NEWTON'S METHOD WITH EXACT LINE SEARCHES*

NICHOLAS J. HIGHAM† AND HYUN-MIN KIM†

Abstract. We show how to incorporate exact line searches into Newton's method for solving the quadratic matrix equation $AX^2 + BX + C = 0$, where A, B and C are square matrices. The line searches are relatively inexpensive and improve the global convergence properties of Newton's method in theory and in practice. We also derive a condition number for the problem and show how to compute the backward error of an approximate solution.

Key words. quadratic matrix equation, solvent, Newton's method, generalized Sylvester equation, exact line searches, quadratic eigenvalue problem, condition number, backward error

AMS subject classifications. 65F30, 65H10

PII. S0895479899350976

1. Introduction.

Nonlinear matrix equations occur in a variety of applications. An important class of examples, arising in control theory, is algebraic Riccati equations, such as $XBX + XA + A^*X + C = 0$, where A, B, and C are given coefficient matrices. Theory of Riccati equations and numerical methods for their solution are well developed [1], [4], [31]. Our interest here is in the quadratic matrix equation

$$Q(X) = AX^2 + BX + C = 0, \qquad A, B, C \in \mathbb{C}^{n\times n}. \tag{1.1}$$

Although some Riccati equations are quadratic matrix equations, and vice versa, the two classes of equations require different techniques for analysis and solution in general.

Motivation for studying the quadratic matrix equation comes from the quadratic eigenvalue problem

$$Q(\lambda)x = \lambda^2 Ax + \lambda Bx + Cx = 0, \qquad A, B, C \in \mathbb{C}^{n\times n}, \tag{1.2}$$

which arises in the analysis of structural systems and vibration problems [30], [36], [37]. The standard approach is to reduce (1.2) to a generalized eigenproblem (GEP) $Gx = \lambda Hx$ of twice the dimension, $2n$. However, as is well known [7], [10], [30], if we can find a solution X of the associated quadratic matrix equation (1.1) then we can write

$$\lambda^2 A + \lambda B + C = -(B + AX + \lambda A)(X - \lambda I) \tag{1.3}$$

and so the eigenvalues of (1.2) are those of X together with those of the GEP $(B + AX)x = -\lambda Ax$, both of which are $n \times n$ problems. Bridges and Morris [5] employ this approach in the solution of differential eigenproblems.

A solution X of (1.1) is called a solvent [10]. More precisely, X is called a right solvent to distinguish it from a left solvent, which is a solution of $X^2A + XB + C = 0$.

*Received by the editors February 2, 1999; accepted for publication (in revised form) by A. Bunse-Gerstner June 9, 2000; published electronically August 8, 2001.
http://www.siam.org/journals/simax/23-2/35097.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/, kim@ma.man.ac.uk, http://www.ma.man.ac.uk/~kim/). The work of the first author was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

Transposing the latter equation yields one of the form (1.1), so we concentrate on (1.1) here.

A dominant (minimal) solvent X is one for which every eigenvalue is greater (less than) in modulus than all the eigenvalues of the quotient $B + AX + \lambda A$ in (1.3). In earlier work, Dennis, Traub, and Weber gave two linearly convergent algorithms for computing a dominant solvent of an arbitrary degree matrix polynomial [11]. One of these is a generalization of Bernoulli's method for scalar polynomials and is also described by Gohberg, Lancaster, and Rodman [20, sec. 4.2]. These algorithms have the drawbacks that it is difficult to check in advance whether a dominant solvent exists and the convergence can be extremely slow (see [27] for more details). Davis [7], [8] applied Newton's method to the quadratic matrix equation, giving supporting theory and implementation details. Kratz and Stickel [29] investigated Newton's method for the general matrix polynomial.

This work has two main contributions. First, following an idea of Benner and Byers [2] (and, much earlier, of Man [33]) in the context of the algebraic Riccati equation, we incorporate exact line searches into Newton's method for the quadratic matrix equation in order to improve the global convergence properties. We show experimentally that exact line searches improve the reliability of Newton's method, leading to more frequent convergence and, often, faster convergence. Our second contribution is to derive the true condition number for the quadratic matrix equation, thus obtaining a sharper perturbation bound than Davis [7], and to obtain the backward error of an approximate solution.

Solving even the scalar quadratic equation reliably in floating point arithmetic is a difficult problem, as pointed out by Forsythe [15], principally due to the difficulty of handling underflow and overflow. We do not consider here the effects of underflow and overflow, but rather concentrate on the difficulties present with exact computation.

2. Theory. Before considering numerical solution of the quadratic matrix equation we examine the existence and enumeration of solvents. The fundamental theorem of algebra does not hold for matrix polynomials, as is shown by the special case of the matrix square root problem $X^2 = A$, which does not always have a solution when A is singular [28, sec. 6.4].

The quadratic matrix equation can be solved explicitly when $A = I$, B commutes with C, and $B^2 - 4C$ has a square root. We can complete the square in the usual way to obtain the solution

$$X = -\tfrac{1}{2}B + \tfrac{1}{2}(B^2 - 4C)^{1/2},$$

where $A^{1/2}$ denotes any square root that is a polynomial in A. This case pertains, for example, when A and B are scalar multiples of the identity and $B^2 - 4AC$ is nonsingular, after scaling though by A^{-1}. However, no generalization of the formula for the solution of a scalar quadratic is available for general A, B, and C.

Various sufficient conditions for the existence of a solvent are given by Eisenfeld [12] and Lancaster and Rokne [32]. In the former paper the results are obtained using the contraction mapping principle and in the latter paper using the Newton–Kantorovich theorem. Roughly speaking, all these results require that B or B^{-1} be small in norm compared with A and C, so they are of limited practical applicability.

The existence of dominant and minimal solvents is guaranteed for problems coming from overdamped quadratic eigenvalue problems (1.2): those for which A, B, and C are all symmetric positive definite and $(x^TBx)^2 > 4(x^TAx)(x^TCx)$ for all nonzero x; see Lancaster [30, sec. 7.6].

General information about existence of solvents comes from the connection between the quadratic matrix equation and the quadratic eigenvalue problem (1.2). Note, first, that if A is nonsingular then $\det(Q(\lambda)) = \det(A)\det(\lambda^2 I + A^{-1}B\lambda + A^{-1}C)$, so $\det(Q(\lambda))$ has degree exactly $2n$ and hence $Q(\lambda)$ has $2n$ eigenvalues, all of which are finite. If A is singular then $\det(Q(\lambda))$ has degree less than $2n$ and hence $Q(\lambda)$ has either less than $2n$ finite eigenvalues or infinitely many if $\det(Q(\lambda)) \equiv 0$.

The next result gives information on the number of solvents of $Q(X)$; it generalizes [10, Cor. 4.1].

THEOREM 2.1. *Suppose $Q(\lambda)$ has p distinct eigenvalues $\{\lambda_i\}_{i=1}^p$, with $n \le p \le 2n$, and that the corresponding set of p eigenvectors $\{v_i\}_{i=1}^p$ satisfies the Haar condition (that is, every subset of n of them is linearly independent). Then there are at least $\binom{p}{n}$ different solvents of $Q(X)$, and exactly this many if $p = 2n$, which are given by*

$$X = W\,\mathrm{diag}(\mu_i)W^{-1}, \qquad W = [\,w_1, \quad \ldots \quad , w_n\,], \tag{2.1}$$

where the eigenpairs $(\mu_i, w_i)_{i=1}^n$ are chosen from among the eigenpairs $(\lambda_i, v_i)_{i=1}^p$ of Q.

Proof. There are clearly $\binom{p}{n}$ choices of X in (2.1). Since $\mu_i^2 A w_i + \mu_i B w_i + C w_i = 0$, we have $AW\,\mathrm{diag}(\mu_i)^2 + BW\,\mathrm{diag}(\mu_i) + CW = 0$ and thence, on postmultiplying by W^{-1}, $Q(X) = 0$. That the $\binom{p}{n}$ solvents are different follows from the fact that no two have the same eigenvalues. Now suppose that $p = 2n$. From (1.3), every eigenpair of X is also an eigenpair of Q, and it follows that X is diagonalizable and of the form (2.1). ∎

When $p = n$ in Theorem 2.1 the distinctness of the eigenvalues is not needed in the proof, and we obtain a sufficient condition for the existence of a solvent.

COROLLARY 2.2. *If $Q(\lambda)$ has n linearly independent eigenvectors $v_1, \ldots, v_n$ then $Q(X)$ has a solvent.*

An example helps to clarify the theory. Consider the quadratic [10]

$$Q(X) = X^2 + \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} X + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}.$$

$Q(\lambda)$ has four distinct eigenvalues, with eigenpairs (λ_i, v_i) given by

i	1	2	3	4
λ_i	1	2	3	4
v_i	$\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$

To apply Theorem 2.1 we can take p no bigger than 3, in view of the Haar condition. If we take eigenvalues 1, 2, 3, then the theorem gives three solvents, having eigenvalues 1 and 2, 1 and 3, and 2 and 3. But the eigenvectors corresponding to eigenvalues 1, 2, 4 also satisfy the Haar condition and this gives us another two solvents, having eigenvalues 1 and 4, and 2 and 4. Note that there is no dominant solvent, which would have to have eigenvalues 3 and 4. The complete set of solvents is

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \quad \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix}, \quad \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}.$$

We were able to find all these solvents using the `solve` command of MATLAB's Symbolic Math Toolbox [34], but symbolic solution is clearly impractical for large n.

For a characterization of solvents via the generalized Schur decomposition of an associated matrix pencil, see [27].

3. Newton's method. Newton's method for solving the quadratic matrix equation (1.1) is readily obtained from the expansion

$$\begin{aligned} Q(X+E) &= Q(X) + \big(AEX + (AX+B)E\big) + AE^2 \\ &= Q(X) + D_X(E) + AE^2, \end{aligned} \tag{3.1}$$

where $D_X(E) : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is the Fréchet derivative of Q at X in the direction E. Newton's method drops the second order term, defines E as the solution of $Q(X) + D_X(E) = 0$, and replaces X by $X+E$. Each step of Newton's method involves finding the solution E of

$$AEX + (AX+B)E = -Q(X), \tag{3.2}$$

which is a special case of the generalized Sylvester equation "$AXB + CXD = E$."

We would like to know when the Fréchet derivative D_X is nonsingular, both at a solvent and at an iterate X, so that (3.2) has a solution. From a result of Chu [6] on the generalized Sylvester equation it follows that D_X is nonsingular if and only if the pair $(-A, AX+B)$ is regular (that is, $\det(-A - \lambda(AX+B))$ is not identically zero in λ) and the eigenvalues of the pair are distinct from the eigenvalues of X. If A is nonsingular, the regularity condition holds. When X is a solvent, we see from (1.3) that the second condition is equivalent to the eigenvalues of X being distinct from the remaining n eigenvalues of $Q(\lambda)$. We can therefore identify some sufficient conditions for nonsingularity of D_X at a solvent.

LEMMA 3.1. *If A is nonsingular then D_X is nonsingular at*

1. *a dominant or minimal solvent X,*
2. *all solvents X if the eigenvalues of $Q(\lambda)$ are distinct.*

For efficiency, $Q(X)$ should be calculated by nested multiplication as $(AX+B)X+C$, which requires two matrix multiplications instead of the three if X^2 is explicitly formed and provides the coefficient matrix $AX+B$ in (3.2) as a byproduct.

To solve (3.2) we can adapt methods for solving the generalized Sylvester equation described by Golub, Nash, and Van Loan [21] and Epton [13] (see also Chu [6] and Gardiner et al. [17], [18]). First we consider a Schur algorithm.

Compute the generalized Schur decomposition of A and $AX+B$ [22, Thm. 7.7.1],

$$W^*AZ = T, \qquad W^*(AX+B)Z = S, \tag{3.3}$$

where W and Z are unitary and T and S are upper triangular, and the Schur decomposition of X, $U^*XU = R$, where U is unitary and R is upper triangular. Then, pre- and postmultiplying (3.2) by W^* and U, respectively, transforms the system to

$$TYR + SY = F, \qquad F = -W^*Q(X)U, \quad Y = Z^*EU. \tag{3.4}$$

Equating kth columns and rearranging leads to

$$(S + r_{kk}T)y_k = f_k - \sum_{i=1}^{k-1} r_{ik}Ty_i, \qquad Y = [y_1, y_2, \ldots, y_n]. \tag{3.5}$$

By solving these upper triangular systems in the order $k = 1\colon n$, Y can be computed a column at a time. The cost of this algorithm is as follows, where a flop denotes a floating point operation. The generalized Schur decomposition requires $66n^3$ flops [22, sec. 7.7.6] and the Schur decomposition $25n^3$ flops [22, sec. 7.5.6]. Forming F

and transforming from Y to E in (3.4) costs $8n^3$ flops, and solving (3.5) requires $3n^3$ flops. The total is therefore $102n^3$ flops.

The Schur algorithm is used by Davis [7]. However, as noted by Golub, Nash, and Van Loan [21], Epton [13], and Gardiner et al. [17], one of the Schur decompositions can be replaced by a Hessenberg-triangular decomposition with a potentially substantial computational saving. Suppose we replace (3.3) by the Hessenberg-triangular decomposition [22, sec. 7.7.4]

$$W^*AZ = T, \qquad W^*(AX + B)Z = H,$$

where the only difference from (3.3) is that H is upper Hessenberg (this decomposition is a preliminary step to computing (3.3) by the QZ algorithm). The analogue of (3.5) is

$$(H + r_{kk}T)y_k = f_k - \sum_{i=1}^{k-1} r_{ik}Ty_i, \tag{3.6}$$

which is an upper Hessenberg system. The Hessenberg-triangular decomposition requires $15n^3$ flops [22, sec. 7.7.6] and the systems (3.6) can be solved in $4n^3$ flops. Hence the total cost of the Hessenberg–Schur algorithm is $52n^3$ flops, which is a 51 percent saving compared with the Schur algorithm.

Versions of the Schur and Hessenberg–Schur algorithms that employ real Schur decompositions and so use only real arithmetic can be developed; see [17] for details.

Standard convergence results for Newton's method apply [9, Thm. 5.2.1], as detailed in [29, Thm. 1]. In particular, if Newton's method is started sufficiently close to a solvent for which the Fréchet derivative is nonsingular, the iteration converges and at a quadratic rate. The Kantorovich theorem can also be applied to provide sufficient conditions for existence of a solvent and convergence of Newton's method to that solvent [9, Thm. 5.3.1].

4. Incorporating line searches. In the solution of unconstrained optimization problems by Newton or quasi-Newton methods it is common to use the Newton direction as a search direction and to define the next iterate by (approximately or exactly) minimizing the objective function along this direction [35, Chap. 3]; the minimization is called a line search. Line searches can also be used on nonlinear equation problems, given a suitable function for the line search to minimize. Benner and Byers [2] (see also [3]) investigate the use of exact line searches in Newton's method for solving the algebraic Riccati equation. (Man [33] had earlier used exact line searches in a quasi-Newton method for the same problem, but did not give any details.) Here, we apply exact line searches with Newton's method for the quadratic matrix equation.

The motivation for line searches is that, far from a solution, the linear model of $Q(X)$ on which Newton's method is based may be inaccurate, and so the Newton step E may not be a good one. Line searches are expected to give better global convergence (that is, convergence from arbitrary starting points). An example adapted from [2] illustrates the point. Consider the quadratic matrix equation

$$X^2 - \begin{bmatrix} 1 & 0 \\ 0 & \delta^{1/2} \end{bmatrix} = 0, \qquad 0 < \delta \ll 1,$$

which has solutions $X = \mathrm{diag}(\pm 1, \pm\delta^{1/4})$. With $X_0 = \mathrm{diag}(1, \delta)$, Newton's method gives $E = \mathrm{diag}(0, (\delta^{-1/2} - \delta)/2)$, so that $X_1 = X_0 + E$ is a much worse approximate

solvent than X_0. However, it is clear that $X_0 + tE$ is a solvent for suitable choice of the scalar t.

In our Newton method with line searches we take a multiple of the Newton step that minimizes the merit function

$$p(t) = \|Q(X + tE)\|_F^2, \tag{4.1}$$

where the Frobenius norm $\|A\|_F = (\mathrm{trace}(A^*A))^{1/2}$. Other choices of merit function could be tried (for example, based on other norms of Q), but this one is convenient to work with and has some theoretical backing, as explained below. Recalling that Newton's method defines E by $Q(X) + D_X(E) = 0$, from (3.1) we have, for this E,

$$\begin{aligned} Q(X + tE) &= Q(X) + tD_X(E) + t^2AE^2 \\ &= (1 - t)Q(X) + t^2AE^2. \end{aligned} \tag{4.2}$$

Thus

$$\begin{aligned} p(t) &= (1 - t)^2\|Q(X)\|_F^2 + t^4\|AE^2\|_F^2 \\ &\quad + (1 - t)t^2\,\mathrm{trace}\big(Q(X)^*AE^2(AE^2)^*Q(X)\big) \\ &\equiv \alpha(1 - t)^2 + \gamma t^4 + \beta(1 - t)t^2 \\ &= \gamma t^4 - \beta t^3 + (\alpha + \beta)t^2 - 2\alpha t + \alpha. \end{aligned} \tag{4.3}$$

If $\gamma = \|AE^2\|_F = 0$ then $p(t) = \alpha(1 - t)^2$, which attains its global minimum at $t = 1$, yielding the standard Newton step. If $\alpha = 0$ then X is a solvent. We can therefore assume that $\gamma > 0$ and $\alpha > 0$.

We have a quartic polynomial p of which we wish to find the global minimum. A quartic has at most two minima, of which one is the global minimum. We have

$$p'(t) = 2\alpha(t - 1) + \beta(2t - 3t^2) + 4\gamma t^3.$$

Hence

$$p'(0) = -2\alpha < 0, \tag{4.4}$$

and

$$\begin{aligned} p'(2) &= 2(\alpha - 4\beta + 16\gamma) \\ &= 2\,\mathrm{trace}\big(Q(X)^*Q(X) - 4(Q(X)^*AE^2 + (AE^2)^*Q(X)) + 16(AE^2)^*AE^2\big) \\ &= 2\,\mathrm{trace}\big((Q(X) - 4AE^2)^*(Q(X) - 4AE^2)\big) \\ &\geq 0. \end{aligned}$$

Since $p'(0) < 0$ and $p'(2) \geq 0$, p' has a real zero in the interval $(0, 2]$, and this zero corresponds to a minimum or a point of inflection of p. Since $t = 1$ corresponds to a pure Newton step, it is therefore reasonable to restrict our attention to the interval $[0, 2]$, although there is no guarantee that there is a minimum of p in this interval when we are far from a solution. Thus we define t by

$$p(t) = \min_{x \in [0,2]} p(x). \tag{4.5}$$

There are two cases to consider.

(1) If p' has one real zero and a (nonreal) complex conjugate pair of roots then the real zero, which must lie in $(0, 2]$, is the desired global minimum.

(2) If p' has three real zeros then at most two are minima of p. If the global minimum lies outside $(0, 2]$ then $t = 2$ needs to be checked, as it may yield a smaller value of p than the zero of p' in $(0, 2]$.

Knowing these cases, it is easy to implement the choice of t in (4.5), since the zeros of the cubic p' and the values of p at these zeros are easily computed.

The question arises of whether the exact line searches interfere with the quadratic convergence of Newton's method, necessitating the explicit setting of $t = 1$ once convergence is approached. The answer is no, under a mild assumption, as we now show. Assume that X_j is within a region where quadratic convergence to X occurs, and let $X_{j+1} = X_j + E_j$ and $\widetilde{X}_{j+1} = X_j + tE_j$ be the standard Newton update and the update with exact line search, respectively. Defining $\Delta_j = X - X_j$, we have

$$\|\Delta_{j+1}\| = O\big(\|\Delta_j\|^2\big).$$

The definition of t ensures that, using (4.2),

$$\begin{aligned} \|(1-t)Q(X_j) + t^2 A E_j^2\| &= \|Q(X_j + tE_j)\| \le \|Q(X_j + E_j)\| \\ &= \|Q(X_{j+1})\| = \|Q(X - \Delta_{j+1})\| \\ &= \|Q(X)\| + O(\|\Delta_{j+1}\|) \\ &= O\big(\|\Delta_j\|^2\big). \end{aligned} \tag{4.6}$$

Now $E_j = -\Delta_{j+1} + \Delta_j$, so $\|E_j\| = O(\|\Delta_j\|)$ and, by (3.1),

$$Q(X_j) = Q(X - \Delta_j) = -D_X(\Delta_j) + O(\|\Delta_j\|^2).$$

Hence, as long as the Fréchet derivative is nonsingular at X, (4.6) implies that $|1-t| = O(\|\Delta_j\|)$. Thus

$$X - \widetilde{X}_{j+1} = X - X_{j+1} + X_{j+1} - \widetilde{X}_{j+1} = O\big(\|\Delta_j\|^2\big) + (1-t)E_j = O\big(\|\Delta_j\|^2\big),$$

as required.

The global convergence properties of Newton's method with exact line searches can be obtained from standard theory. We are effectively solving a nonlinear system $f(x) = 0$ by Newton's method, where $f : \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$, doing line searches on the function $F(x) = f(x)^T f(x)$, as advocated by Dennis and Schnabel [9, sec. 6.5] and Fletcher [14, sec. 6.2]. The global convergence results of [9, sec. 6.3], [14, sec. 2.5] apply provided that certain restrictions known as the Armijo–Goldstein conditions are imposed on the line search. In our notation these conditions may be written as

$$p(t) \le p(0) + c_1 t p'(0), \tag{4.7a}$$

$$p'(t) \ge c_2 p'(0), \tag{4.7b}$$

where c_1 and c_2 are parameters with $0 < c_1 < c_2 < 1$. The first condition ensures that the reduction in p is at least as big as that predicted by a first order model, while the second ensures that the step is not too small, by requiring that the derivative at t be at least some fraction of the derivative at 0. It is easy to see using (4.4) that (4.7a) is equivalent to

$$\|Q(X + tE)\|_F^2 \le (1 - 2c_1 t)\|Q(X)\|_F^2,$$

which requires a sufficient decrease in the merit function. The use of exact line searches does not necessarily imply that the conditions (4.7) are satisfied. However, (4.7b) certainly holds in the usual case when the optimal t is a zero of $p'(t)$, since $p'(0) < 0$. Both conditions have been checked and found to be satisfied in all our numerical tests (with $c_1 = 1/4$, $c_2 = 1/2$), so we have not considered any modifications to the exact line search.

The line search requires three matrix multiplications to compute the coefficients of p in (4.3) ($Q(X)$ is already available), the remaining computations being scalar ones. The total cost of the line search is $5n^3$ flops, which is negligible compared with the cost of computing the Newton direction E (at least $56n^3$ flops).

5. Conditioning. We now derive a condition number for a solvent of the quadratic matrix equation (1.1). The analyses in this section and the next have close connections with analyses for Sylvester and algebraic Riccati equations in [19], [23], [25], [26].

Consider the perturbed equation

$$(A + \Delta A)(X + \Delta X)^2 + (B + \Delta B)(X + \Delta X) + C + \Delta C = 0. \tag{5.1}$$

We will measure the perturbations normwise by

$$\epsilon = \| \, [\, \alpha^{-1}\Delta A, \quad \beta^{-1}\Delta B, \quad \gamma^{-1}\Delta C \,] \, \|_F,$$

where α, β, and γ are nonnegative parameters. A zero value of α, say, simply forces the corresponding perturbation ΔA to be zero. Expanding (5.1) we obtain

$$AX\Delta X + A\Delta X X + B\Delta X = -\Delta A X^2 - \Delta B X - \Delta C + O(\epsilon^2). \tag{5.2}$$

We now use the vec operator, which stacks the columns of a matrix into one long vector, and the Kronecker product $A \otimes B = (a_{ij}B)$, and we use the property $\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X)$ [28, Chap. 4]. Applying the vec operator to (5.2) we obtain

$$\begin{aligned} P \operatorname{vec}(\Delta X) &= -((X^2)^T \otimes I_n)\operatorname{vec}(\Delta A) - (X^T \otimes I_n)\operatorname{vec}(\Delta B) - \operatorname{vec}(\Delta C) + O(\epsilon^2) \\ &= -\, [\, \alpha (X^2)^T \otimes I_n, \quad \beta X^T \otimes I_n, \quad \gamma I_{n^2} \,] \begin{bmatrix} \operatorname{vec}(\Delta A)/\alpha \\ \operatorname{vec}(\Delta B)/\beta \\ \operatorname{vec}(\Delta C)/\gamma \end{bmatrix} + O(\epsilon^2), \end{aligned}$$

where

$$P = I_n \otimes AX + X^T \otimes A + I_n \otimes B.$$

Multiplying by P^{-1}, taking 2-norms, and using $\| \operatorname{vec}(X) \|_2 = \|X\|_F$, we obtain the bound

$$\frac{\|\Delta X\|_F}{\|X\|_F} \le \Psi(X)\epsilon + O(\epsilon^2), \tag{5.3}$$

where

$$\Psi(X) = \|P^{-1} \, [\, \alpha (X^2)^T \otimes I_n, \quad \beta X^T \otimes I_n, \quad \gamma I_{n^2} \,] \, \|_2 / \|X\|_F.$$

This is a sharp bound, to first order in ϵ, so $\Psi(X)$ is the condition number of X. Note that P is nonsingular, and hence $\Psi(X)$ finite, precisely when the Fréchet derivative D_X in (3.1) is nonsingular.

An upper bound for $\Psi(X)$ involving $\|P^{-1}\|_2$ can of course be obtained by bounding the norm of the product by the product of the norms, but this bound can be arbitrarily weaker than (5.3). A perturbation bound for (1.1) that contains a factor $\|D_X^{-1}\|_F$ is derived by Davis [7], and it is easy to show that $\|P^{-1}\|_2 = \|D_X^{-1}\|_F$.

For the special case of the matrix square root we have $A = I$, $B = 0$, $\alpha = \beta = 0$, and the condition number Ψ simplifies to

$$\Psi(X) = \frac{\|P^{-1}\|_2 \gamma}{\|X\|_F}, \qquad P = I_n \otimes X + X^T \otimes I_n,$$

which is the matrix square root condition number identified in [25].

We give an illustrative example from [20, Ex. 4.4], with

$$A = I_2, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix}.$$

The eigenvalues of $Q(\lambda)$ are -1, 0, 0, 1 and there are three solvents:

$$X_1 = \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad X_3 = \begin{bmatrix} -1 & 0 \\ -2 & 0 \end{bmatrix}.$$

The solvent X_1 is dominant and so Theorem 3.1 implies it has a finite condition number; in fact $\Psi(X_1) = 3.64$. The other two solvents are both easily seen to have singular P and hence infinite condition numbers.

6. Backward error. We define the backward error of an approximate solution Y to (1.1) by

$$\begin{aligned} \eta(Y) = \min\big\{ \epsilon : (A + \Delta A)Y^2 + (B + \Delta B)Y + C + \Delta C = 0, \\ \|\,[\,\alpha^{-1}\Delta A, \quad \beta^{-1}\Delta B, \quad \gamma^{-1}\Delta C\,]\,\|_F \le \epsilon \big\}. \end{aligned} \tag{6.1}$$

Defining

$$R = AY^2 + BY + C,$$

the constraint equation in (6.1) can be written as

$$\begin{aligned} -R &= \Delta A Y^2 + \Delta B Y + \Delta C \\ &= [\,\alpha^{-1}\Delta A, \quad \beta^{-1}\Delta B, \quad \gamma^{-1}\Delta C\,] \begin{bmatrix} \alpha Y^2 \\ \beta Y \\ \gamma I_n \end{bmatrix}. \end{aligned} \tag{6.2}$$

Taking Frobenius norms leads to the lower bound for the backward error

$$\eta(Y) \ge \frac{\|R\|_F}{(\alpha^2\|Y^2\|_F^2 + \beta^2\|Y\|_F^2 + n\gamma^2)^{1/2}}.$$

Applying the vec operator to (6.2) gives

$$[\,\alpha(Y^2)^T \otimes I_n, \quad \beta Y^T \otimes I_n, \quad \gamma I_{n^2}\,] \begin{bmatrix} \operatorname{vec}(\Delta A)/\alpha, \\ \operatorname{vec}(\Delta B)/\beta, \\ \operatorname{vec}(\Delta C)/\gamma \end{bmatrix} = -\operatorname{vec}(R), \tag{6.3}$$

which we write as

$$Hz = r, \qquad H \in \mathbb{R}^{n^2 \times 3n^2}.$$

We assume that H is of full rank, which guarantees that (6.3) has a solution, that is, that the backward error is finite. The backward error is the minimum 2-norm solution to this underdetermined system:

$$\eta(Y) = \|H^+ r\|_2,$$

where a superscript "+" denotes the pseudoinverse. To obtain an upper bound for $\eta(Y)$ we use

$$\eta(Y) \le \|H^+\|_2 \|r\|_2 = \frac{\|r\|_2}{\sigma_{\min}(H)},$$

where $\sigma_{\min}$ denotes the smallest singular value, which is nonzero by assumption. Now

$$\begin{aligned}
\sigma_{\min}(H)^2 &= \lambda_{\min}(HH^*) \\
&= \lambda_{\min}(\alpha^2 (Y^2)^T \overline{Y^2} \otimes I_n + \beta^2 Y^T \overline{Y} \otimes I_n + \gamma^2 I_{n^2}) \\
&= \lambda_{\min}(\alpha^2 (Y^2)^* Y^2 \otimes I_n + \beta^2 Y^* Y \otimes I_n + \gamma^2 I_{n^2}) \\
&\ge \alpha^2 \sigma_{\min}(Y^2)^2 + \beta^2 \sigma_{\min}(Y)^2 + \gamma^2.
\end{aligned}$$

Thus

$$\eta(Y) \le \frac{\|R\|_F}{(\alpha^2 \sigma_{\min}(Y^2)^2 + \beta^2 \sigma_{\min}(Y)^2 + \gamma^2)^{1/2}}.$$

We conclude from this analysis that a small relative residual does not necessarily imply a small backward error for the quadratic matrix equation. The same is true for the Sylvester equation [26] and, more generally, the algebraic Riccati equation [19].

7. Numerical experiments. Davis [7], [8] demonstrated the usefulness of Newton's method for solving the quadratic matrix equation. Our purpose in this section is to show experimentally the benefits of exact line searches in Newton's method. Our experiments were done in MATLAB, which has unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$.

First, we give a few details about our Newton implementation. The default starting matrix is, as in [7],

$$X_0 = \left(\frac{\|B\|_F + \sqrt{\|B\|_F^2 + 4\|A\|_F \|C\|_F}}{2\|A\|_F} \right) I,$$

which is designed to have norm roughly of the same order of magnitude as a solvent. We terminate the iteration when the residual $Q(X_k)$ is of the same order of magnitude as the rounding error in computing it, namely, when the relative residual $\rho(X_k)$ satisfies

$$\rho(X_k) = \frac{\|fl(Q(X_k))\|_F}{\|A\|_F \|X_k\|_F^2 + \|B\|_F \|X_k\|_F + \|C\|_F} \le nu. \tag{7.1}$$

Our MATLAB code has an option to choose whether to use line searches. When line searches are being used, they are turned off (t is set to 1) once $\rho(X_k) \le 10^{-7}$; this is not necessary in theory (see section 4), but is done to save work and as a precaution to

avoid rounding errors destroying the quadratic convergence. In evaluating backward errors and condition numbers we took $\alpha = \|A\|_F$, $\beta = \|B\|_F$, $\gamma = \|C\|_F$.

The potential benefits of exact line searches are easily demonstrated. Consider the quadratic matrix equation with

$$A = I_2, \qquad B = \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{7.2}$$

As noted in [8] there are real solvents I_2 and $\left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$ and an infinite number of complex solvents. Applying Newton's method with and without line searches for the default X_0 and $X_0 = 10^j I$, $j = 1, 5, 10$, gave the results in Table 7.1, which show the substantial reduction in iterations that exact line searches can bring. In each case the computed solvent $\widehat{X}$ was within roundoff of I_2, with condition number $\Psi(\widehat{X}) = 1.4$ and backward error $\eta(\widehat{X}) \approx u$.

Our next example is the quadratic matrix equation with

$$A = B = I_2, \qquad C = \begin{bmatrix} -8 & -12 \\ -18 & -26 \end{bmatrix}, \tag{7.3}$$

again from [8], which has four solvents, all real and well conditioned. With the default starting matrix, convergence was obtained in 6 iterations with line searches and 10 without line searches, to the same matrix. We chose starting matrices

$$X_0 = \begin{bmatrix} 1 & x \\ y & 1 \end{bmatrix}, \qquad -1000 \le x,\ y \le 1000,$$

with an equally spaced grid of 100 points (x, y). Table 7.2 shows how many times a solvent was produced within 30, 50, and 100 iterations, respectively. Convergence was obtained to all four solvents, depending on the starting matrix, and a different solvent was sometimes obtained with exact line searches than without. Exact line searches result in more frequent convergence, though in 10 of the cases convergence was obtained without line searches but not with them, and in another 22 cases where both gave convergence faster convergence was obtained without line searches. Thus exact line searches do not lead to uniformly better convergence than when no line searches are used. An interesting phenomenon is that in 48 cases when line searches were not used the test (7.1) was satisfied within 100 iterations, but with $\|X\|_F \gg u^{-1}$, so that X was far from a solvent (these cases were counted as failure to converge for the statistics). This behavior did not happen with line searches: the line searches force $\|Q(X_k)\|_F$ to be a decreasing sequence, which tends to keep X_k from becoming large if there is no large solvent.

Our final example is based on a quadratic eigenvalue problem (1.2) from [16, sec. 10.11], with numerical values modified as in [30, sec. 5.3], modelling oscillations in an airplane wing:

$$A = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}, \qquad B = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.6 & 0.756 & 0.658 \end{bmatrix},$$
$$C = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix}. \tag{7.4}$$

TABLE 7.1
Number of iterations for convergence for problem (7.2).

X_0	Without line searches	With exact line searches
Default	6	5
$10I$	9	6
$10^5 I$	22	6
$10^{10} I$	39	7

TABLE 7.2
Number of times convergence obtained for problem (7.3) *with* 100 *different starting matrices.*

No. iterations allowed	Without line searches	With exact line searches
30	46	54
50	52	73
100	53	88

The 6 eigenvalues are distinct and come in 3 complex conjugate pairs; since any solvent must have 3 eigenvalues chosen from the 6, it follows that there are no real solvents. Starting Newton's method with $X_0 = iI$ we obtained the results displayed in Figure 7.1. Convergence was obtained to the same solvent with and without line searches, with condition number $\Psi(\widehat{X}) = 50$ and backward error $\eta(\widehat{X}) \approx u$. The eigenvalues of the computed solvent are

$$\begin{aligned} &\texttt{-8.8483e-001} + \texttt{8.4415e+000i}, \\ &\texttt{9.4722e-002} + \texttt{2.5229e+000i}, \\ &\texttt{-9.1800e-001} + \texttt{1.7606e+000i}, \end{aligned}$$

and these and their conjugates are the eigenvalues of the quadratic eigenvalue problem.

Finally, we note that in all our tests the global minimum of the merit function p in (4.1) was in $(0, 2]$ and never to the right of 2.

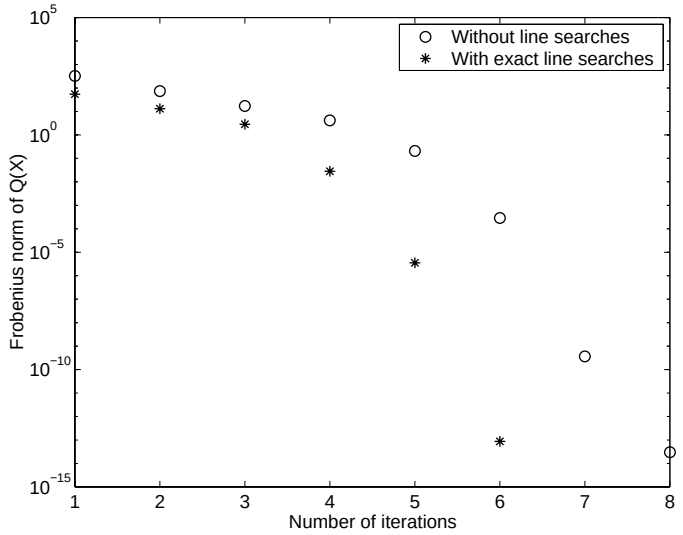


FIG. 7.1. *Convergence for problem* (7.4).

8. Concluding remarks. Newton's method is a useful tool in our stock of methods for solving quadratic matrix equations. In its favor is its applicability to the whole class of problems and its quadratic convergence, the latter making it a useful way to refine approximate solvents obtained with other methods. On the other hand each iteration is relatively expensive. The exact line searches introduced here frequently reduce the number of iterations and make standard global convergence results from optimization applicable.

A number of open problems remain, including guaranteeing convergence for particular starting matrices, determining to which solvent Newton's method will converge, and improving the convergence to solvents at which the Fréchet derivative is singular. These questions have been answered for certain types of Riccati equations, by exploiting their structure [2], [24], but the lack of structure in the quadratic matrix equation has so far precluded any useful results.

Acknowledgments. We thank Françoise Tisseur and the referees for their helpful suggestions.

REFERENCES

[1] P. Benner, *Computational methods for linear-quadratic optimization*, Rend. Circ. Mat. Palermo (2) Suppl., 58 (1999), pp. 21–56. Extended version available as Berichte aus der Technomathematik, Report 98–04, Universität Bremen, August 1998, from http://www.math.uni-bremen.de/zetem/berichte.html.

[2] P. Benner and R. Byers, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Trans. Automat. Control, 43 (1998), pp. 101–107.

[3] P. Benner, R. Byers, E. S. Quintana-Ortí, and G. Quintana-Ortí, *Solving algebraic Riccati equations on parallel computers using Newton's method with exact line search*, Parallel Comput., 26 (2000), pp. 1345–1368.

[4] S. Bittanti, A. J. Laub, and J. C. Willems, eds., *The Riccati Equation*, Springer-Verlag, Berlin, 1991.

[5] T. J. Bridges and P. J. Morris, *Differential eigenvalue problems in which the parameter appears nonlinearly*, J. Comput. Phys., 55 (1984), pp. 437–460.

[6] K.-w. E. Chu, *The solution of the matrix equations* $AXB - CXD = E$ *and* $(YA - DZ, YC - BZ) = (E, F)$, Linear Algebra Appl., 93 (1987), pp. 93–105.

[7] G. J. Davis, *Numerical solution of a quadratic matrix equation*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 164–175.

[8] G. J. Davis, *Algorithm* 598: *An algorithm to compute solvents of the matrix equation* $AX^2 + BX + C = 0$, ACM Trans. Math. Software, 9 (1983), pp. 246–254.

[9] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[10] J. E. Dennis, Jr., J. F. Traub, and R. P. Weber, *The algebraic theory of matrix polynomials*, SIAM J. Numer. Anal., 13 (1976), pp. 831–845.

[11] J. E. Dennis, Jr., J. F. Traub, and R. P. Weber, *Algorithms for solvents of matrix polynomials*, SIAM J. Numer. Anal., 15 (1978), pp. 523–533.

[12] J. Eisenfeld, *Operator equations and nonlinear eigenparameter problems*, J. Funct. Anal., 12 (1973), pp. 475–490.

[13] M. A. Epton, *Methods for the solution of* $AXD - BXC = E$ *and its application in the numerical solution of implicit ordinary differential equations*, BIT, 20 (1980), pp. 341–345.

[14] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, Chichester, UK, 1987.

[15] G. E. Forsythe, *What is a satisfactory quadratic equation solver?*, in Constructive Aspects of the Fundamental Theorem of Algebra, B. Dejon and P. Henrici, eds., Wiley-Interscience, London, 1969, pp. 53–61.

[16] R. A. Frazer, W. J. Duncan, and A. R. Collar, *Elementary Matrices and Some Applications to Dynamics and Differential Equations*, 10th ed., Cambridge University Press, New York, 1963. Reprint of 1938 edition.

[17] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler, *Solution of the Sylvester matrix equation* $AXB^T + CXD^T = E$, ACM Trans. Math. Software, 18 (1992), pp. 223–231.

[18] J. D. Gardiner, M. R. Wette, A. J. Laub, J. J. Amato, and C. B. Moler, *Algorithm* 705: *A FORTRAN-77 software package for solving the Sylvester matrix equation* $AXB^T + CXD^T = E$, ACM Trans. Math. Software, 18 (1992), pp. 232–238.
[19] 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.
[20] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.
[21] G. H. Golub, S. Nash, and C. F. Van Loan, *A Hessenberg–Schur method for the problem* $AX + XB = C$, IEEE Trans. Automat. Control, AC-24 (1979), pp. 909–913.
[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[23] T. Gudmundsson, C. S. Kenney, and A. J. Laub, *Scaling of the discrete-time algebraic Riccati equation to enhance stability of the Schur solution method*, IEEE Trans. Automat. Control, 37 (1992), pp. 513–518.
[24] C.-H. Guo and P. Lancaster, *Analysis and modification of Newton's method for algebraic Riccati equations*, Math. Comp., 67 (1998), pp. 1089–1105.
[25] N. J. Higham, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.
[26] N. J. Higham, *Perturbation theory and backward error for* $AX - XB = C$, BIT, 33 (1993), pp. 124–136.
[27] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.
[28] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[29] W. Kratz and E. Stickel, *Numerical solution of matrix polynomial equations by Newton's method*, IMA J. Numer. Anal., 7 (1987), pp. 355–369.
[30] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966.
[31] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, The Clarendon Press, Oxford University Press, New York, 1995.
[32] P. Lancaster and J. G. Rokne, *Solutions of nonlinear operator equations*, SIAM J. Math. Anal., 8 (1977), pp. 448–457.
[33] F. T. Man, *The Davidon method of solution of the algebraic matrix Riccati equation*, Internat. J. Control, 10 (1969), pp. 713–719.
[34] C. Moler and P. J. Costa, *Symbolic Math Toolbox Version* 2.0: *User's Guide*, The MathWorks, Inc., Natick, MA, 1997.
[35] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[36] H. A. Smith, R. K. Singh, and D. C. Sorensen, *Formulation and solution of the non-linear, damped eigenvalue problem for skeletal systems*, Internat. J. Numer. Methods Engrg., 38 (1995), pp. 3071–3085.
[37] Z. C. Zheng, G. X. Ren, and W. J. Wang, *A reduction method for large scale unsymmetric eigenvalue problems in structural dynamics*, J. Sound Vibration, 199 (1997), pp. 253–268.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 317–338
© 2001 Society for Industrial and Applied Mathematics

AN ITERATIVE METHOD WITH VARIABLE RELAXATION PARAMETERS FOR SADDLE-POINT PROBLEMS*

QIYA HU† AND JUN ZOU‡

Abstract. In this paper, we propose an inexact Uzawa method with variable relaxation parameters for iteratively solving linear saddle-point problems. The method involves two variable relaxation parameters, which can be updated easily in each iteration, similar to the evaluation of the two iteration parameters in the conjugate gradient method. This new algorithm has an advantage over most existing Uzawa-type algorithms: it is always convergent without any a priori estimates on the spectrum of the preconditioned Schur complement matrix, which may not be easy to achieve in applications. The rate of the convergence of the inexact Uzawa method is analyzed. Numerical results of the algorithm applied for the Stokes problem and a purely linear system of algebraic equations are presented.

Key words. saddle-point, inexact Uzawa method, indefinite systems, preconditioning

AMS subject classifications. 65F10, 65N20

PII. S0895479899364064

1. Introduction.

The major interest of this paper is to solve the indefinite system of equations

$$\begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.1}$$

where A is a symmetric and positive definite $n \times n$ matrix, and B is an $n \times m$ matrix with $m \leq n$. We assume that the global coefficient matrix

$$M = \begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix}$$

is nonsingular, which is equivalent to the positive definiteness of the Schur complement matrix

$$C = B^t A^{-1} B. \tag{1.2}$$

Linear systems such as (1.1) are called saddle-point problems, which may arise from finite element discretizations of Stokes equations and Maxwell equations [6], [8], [12]; mixed finite element formulations for second order elliptic problems [2], [6]; or from Lagrange multiplier formulations for optimization problems [1], [13], for parameter identification, and domain decomposition problems [9], [14], [15].

*Received by the editors November 18, 1999; accepted for publication (in revised form) by L. Eldén March 5, 2001; published electronically August 8, 2001.
http://www.siam.org/journals/simax/23-2/36406.html

†Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing 100080, China (hqy@lsec.cc.ac.cn). The work of this author was partially supported by National Natural Science Foundation grant 19801030 and a grant from the Institute of Mathematical Sciences of the Chinese University of Hong Kong.

‡Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (zou@math.cuhk.edu.hk). The work of this author was partially supported by Hong Kong RGC grants CUHK4004/98P and CUHK4292/00P and the Visiting Scholar Foundation of Key Laboratory in University (China).

In recent years, there has been a rapidly growing interest in preconditioned iterative methods for solving the indefinite system of equations like (1.1); see [3], [4], [5], [7], [11], [14], [16], [17], and [18]. In particular, the inexact Uzawa-type algorithms have attracted wide attention; see [3], [4], [7], [11], [17], and the references therein. The main merit of these Uzawa-type algorithms is that they preserve the minimal memory requirement and do not need actions of the inverse matrix A^{-1}.

Let $\hat{A}$ and $\hat{C}$ be two positive definite matrices, which are assumed to be the preconditioners of the matrices A and C, respectively. Also let R^l be the usual l-dimensional Euclidean space. For any $l \times l$ positive definite matrix G, we use $\|x\|_G$ to denote the G-induced norm, i.e., $\|x\|_G = (Gx,\ x)^{1/2}$ for all $x \in R^l$. However, we write $\|x\|$ (the Euclidean norm) when G is the identity. Then the standard inexact Uzawa algorithm can be described as follows (cf. [4] and [11]).

ALGORITHM 1.1 (inexact Uzawa). Given $x_0 \in R^n$ and $y_0 \in R^m$, the sequence $\{x_i, y_i\} \subset R^n \times R^m$ is defined for $i = 1, 2, \ldots$ by

$$x_{i+1} = x_i + \hat{A}^{-1}[f - (Ax_i + By_i)] \tag{1.3}$$

and

$$y_{i+1} = y_i + \hat{C}^{-1}(B^t x_{i+1} - g). \tag{1.4}$$

There are several earlier versions of the above algorithm; see, e.g., [3] and [17]. The existing convergence results indicate that these algorithms are convergent by assuming some good knowledge of the spectrum of the preconditioned matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}C$ or under some proper scalings of the preconditioners $\hat{A}$ and $\hat{C}$. This "preprocessing" may not be easy to achieve in some applications.

To avoid the proper estimate of the generalized eigenvalues of $\hat{C}$ with respect to $B^t\hat{A}^{-1}B$, the Uzawa-type algorithm proposed in [3] introduced a preconditioned conjugate gradient (PCG) algorithm as an inner iteration of (1.4) and proved that when the number of the PCG iteration is suitably large this Uzawa-type algorithm converges. However, it requires subtle skill in implementations to determine when to terminate this inner iteration.

The preconditioned minimal residual method is always convergent, but its convergence depends on the ratio of the smallest eigenvalue of $\hat{A}^{-1}A$ over the smallest eigenvalue of $\hat{C}^{-1}(B^t\hat{A}^{-1}B)$ (cf. [18]). Hence one should have some good knowledge of the smallest eigenvalues of these preconditioned matrices in order to achieve a practical convergence rate. Without a good scaling based on some a priori estimate of these smallest eigenvalues, the condition number of the (global) preconditioned system still may be very large even if the condition numbers of the matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}(B^t\hat{A}^{-1}B)$ are small (cf. [18]). In this case, the convergence of this iterative method may be slow (see section 4).

In this paper we propose a new variant of the inexact Uzawa algorithm to relax some aforementioned drawbacks by introducing two variable relaxation parameters in the algorithm (1.3)–(1.4). That is, we define the sequence $\{x_i, y_i\}$ for $i = 1, 2, \ldots$ by

$$x_{i+1} = x_i + \omega_i\hat{A}^{-1}[f - (Ax_i + By_i)] \tag{1.5}$$

and

$$y_{i+1} = y_i + \tau_i\hat{C}^{-1}(B^t x_{i+1} - g). \tag{1.6}$$

The parameters ω_i and τ_i above can be computed effectively, similar to the evaluation of the two iteration parameters in the conjugate gradient method. It will be shown

that our algorithm always converges provided the preconditioner $\hat{A}$ for A is properly scaled so that the eigenvalues of $A^{-1}\hat{A}$ are bounded by one. It is very interesting to know whether this is a technical or necessary assumption, a question to which we still do not have a definite answer. But the numerical experiments of section 4 seem to imply that the proposed algorithm converges even when this assumption is violated. Furthermore, it is important to remark that the convergence of the new algorithm is independent of the constant scalings of the preconditioners $\hat{A}$ and $\hat{C}$ while the convergences of the preconditioned minimum residual (MINRES) method and Algorithm 1.1 are strongly affected by such constant scalings; see section 4 for some numerical verifications. Also the new algorithm is always convergent for general preconditioners $\hat{C}$, while the convergences of most existing Uzawa-type algorithms are guaranteed only under certain conditions on the extreme eigenvalues of the preconditioned matrix $\hat{C}^{-1}C$ or $\hat{C}^{-1}H$ (cf. [3] and [4]).

The rest of the paper is arranged as follows. In section 2, we describe the algorithm and its convergence results, which indicate that the algorithm converges with an optimal rate (independent of mesh sizes) if the preconditioned matrices $\hat{A}^{-1}A$ and $\hat{C}^{-1}C$ or $\hat{C}^{-1}(B^t\hat{A}^{-1}B)$ are well-conditioned. The analysis of convergence rates will be given in section 3. In section 4, we apply the proposed algorithm for solving the Stokes problem and a linear system of purely algebraic equations.

2. Algorithm and main results. We start with some illustrations about how to choose the relaxation parameters ω_i and τ_i in (1.5)–(1.6). We first claim that it is impractical to determine these two parameters by the standard steepest descent method. To see this, let $\{x, y\}$ be the true solution of the saddle-point problem (1.1) and set

$$e_i^x = x - x_i, \qquad e_i^y = y - y_i,$$

$$f_i = f - (Ax_i + By_i), \quad g_i = B^t x_{i+1} - g.$$

Consider two arbitrary symmetric and positive definite $n \times n$ and $m \times m$ matrices A_0 and C_0. Suppose we choose the parameters ω_i and τ_i such that the errors

$$\|e_{i+1}^x\|_{A_0}^2 = \|e_i^x\|_{A_0}^2 - 2\,\omega_i(e_i^x, \hat{A}^{-1}f_i)_{A_0} + \omega_i^2\|\hat{A}^{-1}f_i\|_{A_0}^2$$

and

$$\|e_{i+1}^y\|_{C_0}^2 = \|e_i^y\|_{C_0}^2 - 2\,\tau_i(e_i^y, \hat{C}^{-1}g_i)_{C_0} + \tau_i^2\|\hat{C}^{-1}g_i\|_{C_0}^2$$

are minimized; then we have

$$\omega_i = \frac{(A_0e_i^x, \hat{A}^{-1}f_i)}{\|\hat{A}^{-1}f_i\|_{A_0}^2}, \quad f_i \neq 0; \quad \tau_i = \frac{(C_0e_i^y, \hat{C}^{-1}g_i)}{\|\hat{C}^{-1}g_i\|_{C_0}^2}, \quad g_i \neq 0.$$

This requires the evaluations of $A_0e_i^x = A_0x - A_0x_i$ and $C_0e_i^y = C_0y - C_0y_i$. Clearly such evaluations are usually very expensive no matter how we choose A_0 and C_0, since the action of A^{-1} is always involved. This verifies our claim.

Now, we are going to find a more efficient way to compute the parameters ω_i and τ_i. Note that the exact version of the inner iteration (1.3) is

$$x_{i+1} = x_i + A^{-1}f_i.$$

Comparing this with the inexact iteration (1.5), we see that ω_i may be chosen such that the norm

$$\|A^{-1}f_i - \omega_i \hat{A}^{-1}f_i\|_A^2$$

is minimized. A direct computation yields that

$$\omega_i = \begin{cases} \frac{(f_i, \hat{A}^{-1}f_i)}{\|\hat{A}^{-1}f_i\|_A^2}, & f_i \neq 0, \\ 1, & f_i = 0. \end{cases} \tag{2.1}$$

With this parameter ω_i, the outer iteration (1.4) is changed to

$$y_{i+1} = y_i + \hat{C}^{-1}(b_i - \omega_i B^t \hat{A}^{-1} B y_i)$$

with

$$b_i = B^t x_i + \omega_i B^t \hat{A}^{-1}(f - Ax_i) - g,$$

which is independent of y_i. When replacing $\hat{C}$ by $\omega_i B^t \hat{A}^{-1} B$, we get the exact version of this outer iteration:

$$y_{i+1} = y_i + (\omega_i B^t \hat{A}^{-1} B)^{-1} g_i.$$

Comparing this with the inexact form (1.6), we see that the parameter τ_i may be chosen such that the norm

$$\|(\omega_i B^t \hat{A}^{-1} B)^{-1} g_i - \tau_i \hat{C}^{-1} g_i\|^2_{(\omega_i B^t \hat{A}^{-1} B)}$$

is minimized. A direct calculation gives

$$\tau_i = \begin{cases} \omega_i^{-1} \frac{(\hat{C}^{-1}g_i,\ g_i)}{\|\hat{C}^{-1}g_i\|^2_{B^t\hat{A}^{-1}B}}, & g_i \neq 0; \\ 1, & g_i = 0. \end{cases} \quad \text{or} \quad \tau_i = \begin{cases} \omega_i^{-1} \frac{(\hat{C}^{-1}g_i,\ g_i)}{\|B\hat{C}^{-1}g_i\|^2_{\hat{A}^{-1}}}, & g_i \neq 0; \\ 1, & g_i = 0. \end{cases} \tag{2.2}$$

Unfortunately, such a relaxation parameter τ_i chosen as in (2.2) may cause the corresponding algorithm (1.5)–(1.6) to diverge, especially when ω_i is very small. This has been confirmed by our numerical experiments. Also we will see from the subsequent analysis that the factor ω_i^{-1} in (2.2) needs to be corrected appropriately to guarantee the convergence.

With the above preparations, we are now ready to formulate a new inexact Uzawa algorithm.

ALGORITHM 2.1 (Uzawa algorithm with variable relaxation parameters). Given the initial guesses $x_0 \in R^n$ and $y_0 \in R^m$, compute the sequences $\{x_i, y_i\}$ for $i = 1, 2, \ldots$ as follows.

Step 1. Compute $f_i = f - (Ax_i + By_i)$, $r_i = \hat{A}^{-1}f_i$, and

$$\omega_i = \begin{cases} \frac{(f_i,\ r_i)}{(Ar_i, r_i)}, & f_i \neq 0, \\ 1, & f_i = 0. \end{cases}$$

Set

$$x_{i+1} = x_i + \omega_i\, r_i\,. \tag{2.3}$$

Step 2. Compute $g_i = B^t x_{i+1} - g$, $d_i = \hat{C}^{-1} g_i$, and

$$\tau_i = \begin{cases} \frac{(g_i,\ d_i)}{(\hat{A}^{-1}Bd_i,\ Bd_i)}, & g_i \neq 0, \\ 1, & g_i = 0\,. \end{cases}$$

Set

$$y_{i+1} = y_i + \theta_i \tau_i\, d_i \tag{2.4}$$

with

$$\theta_i = \frac{1 - \sqrt{1 - \omega_i}}{2}. \tag{2.5}$$

Remark 2.1. Intuitively, it is not easy to see why one needs to introduce the additional parameter θ_i in (2.4), but its presence is essential to guarantee the convergence of Algorithm 2.1. This will become transparent from our subsequent convergence proof. Also, the choices of θ_i in (2.4) are not unique. In fact, θ_i can be chosen to be any real numbers such that

$$0 < \theta_i \leq \frac{1 - \sqrt{1 - \omega_i}}{2}.$$

We refer to the remarks at the end of section 3 for more details.

Remark 2.2. It is clear that when both f_i and g_i vanish, the vectors x_i and y_i are the exact solution of the system (1.1). In this case Algorithm 2.1 terminates.

Now we are ready to state our main results. Let $H = B^t \hat{A}^{-1} B$ and

$$\kappa_1 = \mathrm{cond}(\hat{A}^{-1}A), \quad \alpha = \frac{\kappa_1 - 1}{\kappa_1 + 1},$$

$$\kappa_2 = \mathrm{cond}(\hat{C}^{-1}H), \quad \beta = \frac{\kappa_2 - 1}{\kappa_2 + 1}\,.$$

We shall frequently use a new norm $|||\cdot|||$ given by

$$|||v||| = \left(\|v_1\|^2 + \|v_2\|_C^2\right)^{\frac{1}{2}}, \quad v = \{v_1, v_2\} \in R^n \times R^m.$$

Without loss of generality, from now on we will always assume that $\alpha > 0$, and the preconditioner $\hat{A}$ for A is properly scaled so that

$$(\hat{A}v, v) \leq (Av, v) \quad \text{for all } v \in R^n. \tag{2.6}$$

The numerical experiments of section 4 indicate that Algorithm 2.1 still converges when the condition (2.6) is violated. But our convergence proof will make use of this assumption, and it is still an open question whether the convergence of Algorithm 2.1 is guaranteed without this assumption.

The following two theorems summarize the main results of the paper, and their proofs will be given in section 3.

THEOREM 2.1. *With the assumption* (2.6)*, there is a positive number* $\rho < 1$ *such that*

$$|||E_{i+1}||| \leq \rho\, |||E_i|||$$

with $E_i = \{\sqrt{\alpha}A^{-\frac{1}{2}}f_i,\ e_i^y\}$. *Also the positive number* ρ *can be estimated by*

$$\rho \le \rho_0 = \frac{|c(\gamma,\alpha)| + \sqrt{c(\gamma,\alpha)^2 + 4\alpha}}{2} \tag{2.7}$$

with

$$\gamma \equiv \frac{(1-\beta)(\sqrt{\lambda_0} - \sqrt{\lambda_0 - 1})}{2\lambda_0\sqrt{\lambda_0}} < 1-\alpha, \quad c(\gamma,\alpha) = 1 - \gamma - \alpha(1+\gamma).$$

Here λ_0 *is any positive number such that*

$$(Av, v) \le \lambda_0(\hat{A}v, v) \quad \textit{for all } v \in R^n. \tag{2.8}$$

Moreover, we have

$$\rho_0 < \begin{cases} 1 - \frac{1}{2}\gamma(1+\alpha), & 0 < \gamma \le \frac{1-\alpha}{1+\alpha}, \\ 1 - \frac{1}{2}(1-\alpha)^2, & \frac{1-\alpha}{1+\alpha} < \gamma < 1-\alpha. \end{cases} \tag{2.9}$$

THEOREM 2.2. *With the assumption* (2.6)*, Algorithm* 2.1 *converges, and we have*

$$\|e_i^x\|_A \le (\sqrt{1+4\alpha} + \rho)\rho^{i-1}|||E_0|||, \quad i = 1, 2, \ldots,$$

and

$$\|e_i^y\|_C \le \rho^i|||E_0|||, \quad i = 1, 2, \ldots.$$

Remark 2.3. There always exists a λ_0 such that (2.8) holds. It follows from (2.6) that $\lambda_0 \ge 1$.

Remark 2.4. Theorem 2.2 indicates that Algorithm 2.1 is always convergent for general preconditioners $\hat{C}$. This seems to be a big advantage over most existing inexact Uzawa-type algorithms for saddle-point problems, whose convergences are guaranteed only under certain conditions on the extreme eigenvalues of the preconditioned matrix $\hat{C}^{-1}C$ or $\hat{C}^{-1}H$; see, for example, [3] and [4].

3. Analysis of the convergence rate. This section will focus on the proofs of our main results stated in Theorems 2.1 and 2.2. Unless otherwise specified, the notation below will be the same as that defined in section 2. In our subsequent proofs we will often use the following well-known inequality:

$$\frac{(v,v)\,(v,v)}{(Gv,v)\,(G^{-1}v,v)} \ge \frac{4\lambda_1\lambda_2}{(\lambda_1+\lambda_2)^2} \quad \text{for all } v \in R^l, \tag{3.1}$$

where λ_1 and λ_2 are the smallest and largest eigenvalues of the $l \times l$ symmetric positive definite matrix G. First we will show some auxiliary lemmas.

For $f_i \ne 0$, let α_i denote the following ratio:

$$\alpha_i = \frac{\|(I - \omega_i A^{\frac{1}{2}}\hat{A}^{-1}A^{\frac{1}{2}})A^{-\frac{1}{2}}f_i\|}{\|A^{-\frac{1}{2}}f_i\|}.$$

LEMMA 3.1. *With the assumption* (2.6)*, the above ratio* α_i *and the parameter* ω_i *given in Algorithm* 2.1 *can be bounded above and below as follows:*

$$\lambda_0^{-1} \le \omega_i \le 1 - \alpha_i^2, \quad 0 \le \alpha_i \le \alpha.$$

Proof. By the definition of the parameter ω_i, we have

$$\begin{aligned}\|(I-\omega_i A^{\frac{1}{2}}\hat{A}^{-1}A^{\frac{1}{2}})A^{-\frac{1}{2}}f_i\|^2 &= \|A^{-1}f_i-\omega_i\hat{A}^{-1}f_i\|_A^2\\ &= \|A^{-1}f_i\|_A^2-\omega_i(f_i,\hat{A}^{-1}f_i)\\ &= \left(1-\omega_i\frac{(f_i,\hat{A}^{-1}f_i)}{(f_i,A^{-1}f_i)}\right)\|A^{-1}f_i\|_A^2. \end{aligned} \tag{3.2}$$

Using the Cauchy–Schwarz inequality and assumption (2.6), we obtain

$$\begin{aligned}(A^{-1}f_i,f_i)=(\hat{A}(A^{-1}f_i),\hat{A}^{-1}f_i) &\le \|A^{-1}f_i\|_{\hat{A}}\,\|\hat{A}^{-1}f_i\|_{\hat{A}}\\ &\le \|A^{-1}f_i\|_A\,\|\hat{A}^{-\frac{1}{2}}f_i\| = (A^{-1}f_i,f_i)^{\frac{1}{2}}\,(\hat{A}^{-1}f_i,f_i)^{\frac{1}{2}}.\end{aligned}$$

Thus

$$(A^{-1}f_i,f_i)\le(\hat{A}^{-1}f_i,f_i),$$

and this with (3.2) leads to $\alpha_i^2\le 1-\omega_i$ or $\omega_i\le 1-\alpha_i^2$. The desired lower bound of ω_i is a direct consequence of (2.8) and the definition of ω_i.

We next show that $0\le\alpha_i\le\alpha$. It follows from (3.1) that

$$\begin{aligned}\omega_i\frac{(f_i,\hat{A}^{-1}f_i)}{(f_i,A^{-1}f_i)} &= \frac{(f_i,\hat{A}^{-1}f_i)^2}{(A\hat{A}^{-1}f_i,\hat{A}^{-1}f_i)\,(f_i,A^{-1}f_i)}\\ &= \frac{(\hat{A}^{-\frac{1}{2}}f_i,\hat{A}^{-\frac{1}{2}}f_i)^2}{(\hat{A}^{-\frac{1}{2}}A\hat{A}^{-\frac{1}{2}}(\hat{A}^{-\frac{1}{2}}f_i),\hat{A}^{-\frac{1}{2}}f_i)\,(\hat{A}^{\frac{1}{2}}A^{-1}\hat{A}^{\frac{1}{2}}(\hat{A}^{-\frac{1}{2}}f_i),\hat{A}^{-\frac{1}{2}}f_i)}\\ &\ge \frac{4\lambda_1\lambda_2}{(\lambda_1+\lambda_2)^2},\end{aligned}$$

where λ_1 and λ_2 are the minimal and maximal eigenvalues of the matrix $\hat{A}^{-\frac{1}{2}}A\hat{A}^{-\frac{1}{2}}$, respectively. This with (3.2) implies that

$$\alpha_i^2\le 1-\frac{4\lambda_1\lambda_2}{(\lambda_1+\lambda_2)^2}=\alpha^2. \qquad \square$$

The following lemma introduces an auxiliary matrix Q_{Bi} which plays an important role in the subsequent spectral estimates of the propagation matrix associated with Algorithm 2.1.

LEMMA 3.2. *With the assumption* (2.6)*, for any natural number i, there is a symmetric and positive definite $m\times m$ matrix Q_{Bi} such that*

(i) $Q_{Bi}^{-1}g_i=\theta_i\tau_i\hat{C}^{-1}g_i$ *with* $g_i=B^tx_{i+1}-g$ *as defined in Algorithm* 2.1;

(ii) *all eigenvalues of the matrix $Q_{Bi}^{-1}C$ lie in the interval* $[\frac{\theta_i(1-\beta)}{\lambda_0},\theta_i(1+\beta)]$.

Proof. If $g_i=0$, $Q_{Bi}=[\theta_i(1+\beta)]^{-1}C$ is the desired matrix. We next consider the case with $g_i\neq 0$. Using $H=B^t\hat{A}^{-1}B$, we can write

$$\|B\hat{C}^{-1}g_i\|_{\hat{A}^{-1}}^2=\|\hat{C}^{-1}g_i\|_H^2;$$

then by the definition of the parameter τ_i we have

$$\|\tau_i\hat{C}^{-1}g_i-H^{-1}g_i\|_H^2=\|H^{-1}g_i\|_H^2-\tau_i(g_i,\hat{C}^{-1}g_i)=\left(1-\tau_i\frac{(g_i,\hat{C}^{-1}g_i)}{(g_i,H^{-1}g_i)}\right)\|H^{-1}g_i\|_H^2.$$

It follows from (3.1) that

$$\tau_i \frac{(g_i, \hat{C}^{-1} g_i)}{(g_i, H^{-1} g_i)} = \frac{(\hat{C}^{-\frac{1}{2}} g_i, \hat{C}^{-\frac{1}{2}} g_i)^2}{(\hat{C}^{-\frac{1}{2}} H \hat{C}^{-\frac{1}{2}} (\hat{C}^{-\frac{1}{2}} g_i), \hat{C}^{-\frac{1}{2}} g_i)\,(\hat{C}^{\frac{1}{2}} H^{-1} \hat{C}^{\frac{1}{2}} (\hat{C}^{-\frac{1}{2}} g_i), \hat{C}^{-\frac{1}{2}} g_i)} \geq \frac{4\lambda_1' \lambda_2'}{(\lambda_1' + \lambda_2')^2},$$

where λ_1' and λ_2' are the minimal and maximal eigenvalues of the matrix $\hat{C}^{-\frac{1}{2}} H \hat{C}^{-\frac{1}{2}}$, respectively. Hence we obtain

$$\|\tau_i \hat{C}^{-1} g_i - H^{-1} g_i\|_H \leq \left\{ 1 - \frac{4\lambda_1' \lambda_2'}{(\lambda_1' + \lambda_2')^2} \right\}^{\frac{1}{2}} \|H^{-1} g_i\|_H = \beta \|H^{-1} g_i\|_H.$$

This implies the existence of a symmetric positive definite $m \times m$ matrix G_{Bi} such that

$$G_{Bi}^{-1} g_i = \tau_i \hat{C}^{-1} g_i$$

and

$$\|I - H^{\frac{1}{2}} G_{Bi}^{-1} H^{\frac{1}{2}}\| \leq \beta. \tag{3.3}$$

See Lemma 9 in [3], for example, for the existence of such a matrix G_{Bi}.

Now set $Q_{Bi}^{-1} = \theta_i G_{Bi}$; then

$$Q_{Bi}^{-1} g_i = \theta_i \tau_i \hat{C}^{-1} g_i,$$

and we know from (3.3) that all eigenvalues of the matrix $H^{\frac{1}{2}} Q_{Bi}^{-1} H^{\frac{1}{2}}$ lie in the interval $[\theta_i(1-\beta), \theta_i(1+\beta)]$.

To prove result (ii), let ϕ be an eigenvector of the matrix $Q_{Bi}^{-1} C$ corresponding to the eigenvalue λ. Then we can write

$$(C\phi, \phi) = \lambda (Q_{Bi} \phi, \phi),$$

or equivalently,

$$(\hat{A}^{\frac{1}{2}} A^{-1} \hat{A}^{\frac{1}{2}} (\hat{A}^{-\frac{1}{2}} B\phi), (\hat{A}^{-\frac{1}{2}} B\phi)) = \lambda (Q_{Bi}\phi, \phi).$$

Using inequalities (2.6) and (2.8), we immediately derive

$$\lambda_0^{-1} (\hat{A}^{-\frac{1}{2}} B\phi, \hat{A}^{-\frac{1}{2}} B\phi) \leq \lambda (Q_{Bi}\phi, \phi) \leq (\hat{A}^{-\frac{1}{2}} B\phi, \hat{A}^{-\frac{1}{2}} B\phi).$$

This can be written as

$$\lambda_0^{-1} (H\phi, \phi) \leq \lambda (Q_{Bi}\phi, \phi) \leq (H\phi, \phi).$$

Note that $Q_{Bi}^{-1} H$ has the same eigenvalues as the matrix $H^{\frac{1}{2}} Q_{Bi}^{-1} H^{\frac{1}{2}}$; thus by (3.3) we have

$$\lambda_0^{-1} \theta_i (1-\beta)(Q_{Bi}\phi, \phi) \leq \lambda (Q_{Bi}\phi, \phi) \leq \theta_i (1+\beta)(Q_{Bi}\phi, \phi),$$

which yields the desired eigenvalue bound. ∎

The two functions $F(z)$ and $\varphi(z)$ to be introduced below and their properties are very helpful in achieving some sharper estimates in the subsequent convergence rate analysis. $F(z)$ is defined for two given positive numbers $\alpha, \gamma \in (0,1)$ as follows:

$$F(z) = \frac{1}{2}\Big(az + b + \sqrt{(az+b)^2 - 4z}\Big), \quad z \in [0,1),$$

where $a = (1+\gamma)^2 + \gamma^2/\alpha$ and $b = \alpha\gamma^2 + (1-\gamma)^2$, and it has the following properties.

LEMMA 3.3. *The function F(z) can be bounded below and above as follows:*

$$\alpha\gamma^2 + (1-\gamma)^2 \le F(z) \le F(\alpha^2) = \Big(|c(\gamma,\alpha)| + \sqrt{c(\gamma,\alpha)^2 + 4\alpha}\Big)^2/4 \tag{3.4}$$

for all $z \in [0,\ \alpha^2]$. *Here* $c(\gamma,\alpha)$ *is as given in Theorem* 2.1.

Proof. Set $f(z) = az + b$. Then

$$F(z) = \frac{1}{2}[f(z) + \sqrt{f^2(z) - 4z}].$$

Moreover, we have

$$f(\alpha^2) = \alpha^2(1+\gamma)^2 + 2\alpha\gamma^2 + (1-\gamma)^2 = c(\gamma,\alpha)^2 + 2\alpha;$$

therefore

$$\sqrt{f^2(\alpha^2) - 4\alpha^2} = \sqrt{[f(\alpha^2) - 2\alpha][f(\alpha^2) + 2\alpha]} = |c(\gamma,\alpha)|\sqrt{c(\gamma,\alpha)^2 + 4\alpha}.$$

Note that $f(\alpha^2)$ can be written as

$$f(\alpha^2) = \frac{1}{2}c(\gamma,\alpha)^2 + \frac{1}{2}\{c(\gamma,\alpha)^2 + 4\alpha\};$$

then

$$F(\alpha^2) = \frac{1}{2}[f(\alpha^2) + \sqrt{f^2(\alpha^2) - 4\alpha^2}] = \left(\frac{|c(\gamma,\alpha)| + \sqrt{c(\gamma,\alpha)^2 + 4\alpha}}{2}\right)^2.$$

It is easy to see that (3.4) is equivalent to

$$F(0) \le F(z) \le F(\alpha^2),$$

so it suffices to prove that $F(z)$ is a real and monotone increasing function in the interval $[0,\ 1)$. First we see that

$$\begin{aligned} ab &= [(1+\gamma)^2 + \gamma^2/\alpha]\,[\alpha\gamma^2 + (1-\gamma)^2] \\ &= \alpha\gamma^2(1+\gamma)^2 + (1-\gamma^2)^2 + \gamma^4 + \frac{\gamma^2(1-\gamma)^2}{\alpha} \\ &= 1 + \left[\sqrt{\alpha}\gamma(1+\gamma) - \frac{\gamma(1-\gamma)}{\sqrt{\alpha}}\right]^2; \end{aligned}$$

thus $ab \ge 1$, and

$$(az+b)^2 - 4z = (az + 2\sqrt{z} + b)\left[\left(\sqrt{az} - \frac{1}{\sqrt{a}}\right)^2 + \frac{ab-1}{a}\right] \ge 0,$$

which indicates that $F(z)$ is real in the interval $[0,\ 1)$.

On the other hand, taking the derivative of F, we have

$$F'(z) = \frac{f'(z)[f(z) + \sqrt{f^2(z) - 4z}] - 2}{2\sqrt{f^2(z) - 4z}}, \quad z \in [0,\ 1);$$

then the condition that $F'(z) \geq 0$ is equivalent to

$$f'(z)[\sqrt{f^2(z) - 4z}] \geq 2 - f'(z)f(z), \quad z \in [0,\ 1). \tag{3.5}$$

Using $ab \geq 1$, we obtain (note that $f'(z) = a$)

$$z[f'(z)]^2 - f(z)f'(z) + 1 = a^2 z - a(az + b) + 1 = 1 - ab \leq 0, \quad z \in [0,\ 1).$$

This implies

$$[f'(z)]^2[f^2(z) - 4z] \geq [2 - f'(z)f(z)]^2, \quad z \in [0,\ 1),$$

which guarantees the inequality (3.5). (Note that $f'(z)\sqrt{f^2(z) - 4z} \geq 0$.) □

LEMMA 3.4. *Let γ be defined as in Theorem* 2.1 *and $\varphi(z) = \alpha z^2 + (1 - z)^2$; then*

$$\varphi(z) \leq \varphi(\gamma) \quad \textit{for all} \quad z \in \left[\frac{1 - \beta}{2\lambda_0},\ \frac{1 + \beta}{2}\right].$$

Proof. We can directly verify that

$$\varphi'(z) \begin{cases} < 0, & z < (1 + \alpha)^{-1}; \\ = 0, & z = (1 + \alpha)^{-1}; \\ > 0, & z > (1 + \alpha)^{-1}. \end{cases}$$

So the maximum value of $\varphi(z)$ is

$$\max\left\{\varphi\left(\frac{1 - \beta}{2\lambda_0}\right),\ \varphi\left(\frac{1 + \beta}{2}\right)\right\}.$$

By the direct calculations we have

$$\varphi\left(\frac{1 - \beta}{2\lambda_0}\right) = 1 - \frac{1 - \beta}{\lambda_0} + \frac{(1 + \alpha)(1 - \beta)^2}{4\lambda_0^2}$$

and

$$\varphi\left(\frac{1 + \beta}{2}\right) = 1 - (1 + \beta) + \frac{(1 + \alpha)(1 + \beta)^2}{4}.$$

Thus

$$\varphi\left(\frac{1 - \beta}{2\lambda_0}\right) - \varphi\left(\frac{1 + \beta}{2}\right) = \left[1 - \frac{1 + \alpha}{4}\left(1 + \beta + \frac{1 - \beta}{\lambda_0}\right)\right]\left[(1 + \beta) - \frac{1 - \beta}{\lambda_0}\right].$$

Note that $\lambda_0 \geq 1$ and $\alpha < 1$; hence

$$\frac{1 - \beta}{\lambda_0} \leq 1 - \beta \leq 1 + \beta$$

and

$$\frac{1+\alpha}{4}\left(1+\beta+\frac{1-\beta}{\lambda_0}\right) \le \frac{1+\alpha}{4}(1+\beta+1-\beta) < 1,$$

and we have

$$\varphi\left(\frac{1-\beta}{2\lambda_0}\right) - \varphi\left(\frac{1+\beta}{2}\right) \ge 0. \tag{3.6}$$

So $\varphi(z)$ reaches its maximum at $z = (1-\beta)/(2\lambda_0)$. By the definition of γ it is easy to see that

$$\frac{1-\beta}{2\lambda_0} \ge \gamma;$$

this and the monotonicity of φ implies the desired estimate of Lemma 3.4. ∎

The following spectral bounds will be directly used in the spectral estimates of the propagation matrix associated with Algorithm 2.1.

LEMMA 3.5. *Let Q be a given symmetric positive definite matrix with its eigenvalues lying in the interval $[\frac{\theta_i(1-\beta)}{\lambda_0}, \theta_i(1+\beta)]$ (cf. Lemma 3.2(ii)), and F_i is a matrix given by*

$$F_i = \begin{pmatrix} \alpha_i(I+Q) & -\sqrt{\alpha}Q \\ \sqrt{\alpha}^{-1}\alpha_i Q & (I-Q) \end{pmatrix}.$$

Then the spectrum of F_i is bounded by ρ_0 (defined in (2.7)), i.e., $\|F_i\| \le \rho_0$.

Proof. Let $\{\lambda_j\}_{j=1}^m$ be the positive eigenvalues of the matrix Q. It is easy to verify that

$$\|F_i\| = \max_{1\le j\le m} \left\| \begin{pmatrix} \alpha_i(1+\lambda_j) & -\sqrt{\alpha}\lambda_j \\ \sqrt{\alpha}^{-1}\alpha_i\lambda_j & 1-\lambda_j \end{pmatrix} \right\|. \tag{3.7}$$

To estimate $\|F_i\|$, it suffices to estimate the maximum eigenvalue of the matrix $\mathcal{F}_i^t\mathcal{F}_i$ with

$$\mathcal{F}_i = \begin{pmatrix} \alpha_i(1+\lambda_j) & -\sqrt{\alpha}\lambda_j \\ \sqrt{\alpha}^{-1}\alpha_i\lambda_j & 1-\lambda_j \end{pmatrix}.$$

The determinant of the matrix $\mathcal{F}_i^t\mathcal{F}_i$ can be simplified as follows:

$$\begin{aligned}
&[\alpha_i^2(1+\beta_j)^2 + \alpha^{-1}\alpha_i^2\beta_j^2]\,[(1-\beta_j)^2 + \alpha\beta_j^2] - \{\sqrt{\alpha}^{-1}\alpha_i\beta_j[1-\beta_j-\alpha(1+\beta_j)]\}^2 \\
&= \alpha_i^2(1-\beta_j^2)^2 + \alpha\alpha_i^2\beta_j^2(1+\beta_j)^2 + \alpha^{-1}\alpha_i^2\beta_j^2(1-\beta_j)^2 + \alpha_i^2\beta_j^4 \\
&\quad -\alpha^{-1}\alpha_i^2\beta_j^2[(1-\beta_j)^2 - 2\alpha(1-\beta_j^2) + \alpha^2(1+\beta_j)^2] \\
&= \alpha_i^2(1-\beta_j^2)^2 + \alpha\alpha_i^2\beta_j^2(1+\beta_j)^2 + \alpha^{-1}\alpha_i^2\beta_j^2(1-\beta_j)^2 + \alpha_i^2\beta_j^4 \\
&\quad -\alpha^{-1}\alpha_i^2\beta_j^2(1-\beta_j)^2 + 2\alpha_i^2\beta_j^2(1-\beta_j^2) - \alpha\alpha_i^2\beta_j^2(1+\beta_j)^2 \\
&= \alpha_i^2[(1-\beta_j^2)^2 + \beta_j^4 + 2\beta_j^2(1-\beta_j^2)] = \alpha_i^2[(1-\beta_j^2) + \beta_j^2]^2 = \alpha_i^2;
\end{aligned}$$

hence the characteristic equation of $\mathcal{F}_i^t\mathcal{F}_i$ is

$$\lambda^2 - [\alpha_i^2(1+\lambda_j)^2 + \alpha^{-1}\alpha_i^2\lambda_j^2 + (1-\lambda_j)^2 + \alpha\lambda_j^2]\lambda + \alpha_i^2 = 0.$$

Then the desired maximum eigenvalue is

$$\lambda^* = \left(f(\alpha_i, \lambda_j) + \sqrt{f^2(\alpha_i, \lambda_j) - 4\alpha_i^2} \right)/2 \tag{3.8}$$

with $f(\alpha_i, z)$ defined by

$$f(\alpha_i, z) = \alpha_i^2(1+z)^2 + \alpha^{-1}\alpha_i^2\, z^2 + (1-z)^2 + \alpha\, z^2.$$

For a fixed α_i, the equation $f'(\alpha_i, z) = 0$ has a unique solution:

$$z = \beta_0 \equiv \frac{\alpha(1-\alpha_i^2)}{\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha}.$$

Moreover, we have $f'(\alpha_i, z) < 0$ for $z < \beta_0$ and $f'(\alpha_i, z) > 0$ for $z > \beta_0$. Thus using the assumption on the range of the eigenvalues of Q, we have

$$\max_{1\le j\le m} \{f(\alpha_i, \lambda_j)\} \le \max\left\{ f\left(\alpha_i, \frac{\theta_i(1-\beta)}{\lambda_0}\right), f(\alpha_i, \theta_i(1+\beta)) \right\}. \tag{3.9}$$

Noting that

$$\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha \le \alpha(1+\alpha)(1+\alpha_i) < 2\alpha(1+\alpha_i),$$

it follows from Lemma 3.1 that

$$\theta_i = \frac{1-\sqrt{1-\omega_i}}{2} \le \frac{1-\alpha_i}{2} \le \frac{\alpha(1-\alpha_i^2)}{\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha}. \tag{3.10}$$

Using this, one can verify directly that

$$f(\alpha_i, \theta_i(1-\beta)) \ge f(\alpha_i, \theta_i(1+\beta)),$$

which, with the fact that $\lambda_0 \ge 1$, yields

$$f\left(\alpha_i, \frac{\theta_i(1-\beta)}{\lambda_0}\right) \ge f(\alpha_i, \theta_i(1+\beta)). \tag{3.11}$$

On the other hand, Lemma 3.1 implies that $\sqrt{1-\omega_i} \le \sqrt{1-\lambda_0^{-1}}$; hence

$$\theta_i = \frac{1-\sqrt{1-\omega_i}}{2} \ge \frac{1-\sqrt{1-\lambda_0^{-1}}}{2}$$

or

$$\frac{\theta_i(1-\beta)}{\lambda_0} \ge \frac{(1-\beta)}{2\lambda_0}\left(1-\sqrt{1-\lambda_0^{-1}}\right) = \gamma$$

with the γ given in Theorem 2.1. Therefore,

$$f\left(\alpha_i, \frac{\theta_i(1-\beta)}{\lambda_0}\right) \le f(\alpha_i, \gamma);$$

this together with (3.9) and (3.11) leads to

$$f(\alpha_i, \lambda_j) \le f(\alpha_i, \gamma), \quad j = 1, \ldots, m. \tag{3.12}$$

By (3.8), (3.12), and the definitions of $f(\alpha_i, \gamma)$ and $F(z)$, we have $\lambda^* \le F(\alpha_i^2)$. This result together with (3.7), Lemma 3.1, and the second inequality of Lemma 3.3 implies $\|F_i\| \le \rho_0$. □

With the help of Lemmas 3.1–3.5 above, we are now ready to show the convergence results in Theorems 2.1 and 2.2.

Proof of Theorem 2.1. As is true for classical iterative methods, the convergence proofs for most existing inexact Uzawa-type iterative methods are carried out with the natural error vectors $e_i^x = x - x_i$ and $e_i^y = y - y_i$ (cf. [3], [4], [17]). But this traditional analysis seems to be very difficult to follow in our current case with variable relaxation parameters, which is much more complicated technically. It is essential that we shall first estimate the residual f_i instead of the error vector e_i^x. Clearly, the residuals f_i and g_i can be represented in terms of e_i^x and e_i^y:

$$f_i = Ae_i^x + Be_i^y, \quad g_i = -B^t e_{i+1}^x. \tag{3.13}$$

By (2.3) and (3.13) we have

$$A^{\frac{1}{2}} e_{i+1}^x = A^{\frac{1}{2}}(e_i^x - \omega_i \hat{A}^{-1} f_i) = (I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) A^{-\frac{1}{2}} f_i - A^{-\frac{1}{2}} B e_i^y. \tag{3.14}$$

Using (2.4), Lemma 3.2(i), and (3.14) we obtain

$$\begin{aligned} A^{-\frac{1}{2}} B e_{i+1}^y &= A^{-\frac{1}{2}} B(e_i^y - \theta_i \tau_i \hat{C}^{-1} g_i) = A^{-\frac{1}{2}} B(e_i^y + Q_{Bi}^{-1} B^t e_{i+1}^x) \\ &= A^{-\frac{1}{2}} B[e_i^y + Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} ((I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) A^{-\frac{1}{2}} f_i - A^{-\frac{1}{2}} B e_i^y)] \\ &= A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} (I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) A^{-\frac{1}{2}} f_i \\ &\quad + (I - A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}}) A^{-\frac{1}{2}} B e_i^y, \end{aligned} \tag{3.15}$$

while using (3.14) and (3.15) we have

$$\begin{aligned} A^{-\frac{1}{2}} f_{i+1} &= A^{\frac{1}{2}} e_{i+1}^x + A^{-\frac{1}{2}} B e_{i+1}^y \\ &= (I + A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}})(I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) A^{-\frac{1}{2}} f_i \\ &\quad - (A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}}) A^{-\frac{1}{2}} B e_i^y. \end{aligned} \tag{3.16}$$

Now let

$$B^t A^{-\frac{1}{2}} = U \Sigma V^t \tag{3.17}$$

with $\Sigma = (\Sigma_0 \ 0)$ being the singular value decomposition of the matrix $B^t A^{-\frac{1}{2}}$. As usual, U is an orthogonal $m \times m$ matrix and V is an orthogonal $n \times n$ matrix. The diagonal entries of the matrix Σ_0 are the singular values of $B^t A^{-\frac{1}{2}}$. Define

$$E_i^{xy} = \sqrt{\alpha} V^t A^{-\frac{1}{2}} f_i, \quad E_i^y = \Sigma^t U^t e_i^y.$$

By (3.15) and (3.16), we obtain

$$\begin{aligned} E_{i+1}^{xy} &= (I + V^t A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} V) V^t (I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) V E_i^{xy} \\ &\quad - \sqrt{\alpha} (V^t A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} V) E_i^y \end{aligned} \tag{3.18}$$

and

$$E_{i+1}^y = \frac{1}{\sqrt{\alpha}}(V^t A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} V) V^t (I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) V E_i^{xy} + (I - V^t A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} V) E_i^y. \tag{3.19}$$

Set

$$Q_{1i} \equiv V^t (I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}) V$$

and

$$Q_{2i} \equiv \Sigma^t U^t Q_{Bi}^{-1} U \Sigma = V^t A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}} V\,;$$

then the propagation relations (3.18) and (3.19) may be written in the matrix form

$$\begin{pmatrix} E_{i+1}^{xy} \\ E_{i+1}^{y} \end{pmatrix} = \begin{pmatrix} (I + Q_{2i}) Q_{1i} & -\sqrt{\alpha} Q_{2i} \\ \sqrt{\alpha}^{-1} Q_{2i} Q_{1i} & (I - Q_{2i}) \end{pmatrix} \begin{pmatrix} E_i^{xy} \\ E_i^{y} \end{pmatrix}. \tag{3.20}$$

Let E_i^{0y} and Q_{2i}^0 denote the nonzero part of E_i^y and Q_{2i}, respectively, namely,

$$E_i^{0y} = \Sigma_0 U^t e_i^y, \quad Q_{2i}^0 = \Sigma_0 U^t Q_{Bi}^{-1} U \Sigma_0,$$

and set $\hat{Q}_{2i} = (Q_{2i}^0,\ 0)^t$. Then we have from (3.20) that

$$\begin{pmatrix} E_{i+1}^{xy} \\ E_{i+1}^{0y} \end{pmatrix} = \begin{pmatrix} (I + Q_{2i}) Q_{1i} & -\sqrt{\alpha} \hat{Q}_{2i} \\ \sqrt{\alpha}^{-1} \hat{Q}_{2i}^t Q_{1i} & (I - Q_{2i}^0) \end{pmatrix} \begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}. \tag{3.21}$$

Next we estimate the spectrum of the propagation matrix in (3.21). We first consider two cases: $f_i = 0$; $f_i \neq 0$ but $\alpha_i = 0$. Then we have by the definition of E_i^{xy} and α_i that

$$Q_{1i} E_i^{xy} = 0 \quad \text{for} \quad f_i = 0 \quad \text{or} \quad \alpha_i = 0.$$

So we can write (3.21) as

$$\begin{pmatrix} E_{i+1}^{xy} \\ E_{i+1}^{0y} \end{pmatrix} = \begin{pmatrix} 0 & -\sqrt{\alpha} \hat{Q}_{2i} \\ 0 & (I - Q_{2i}^0) \end{pmatrix} \begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix} \equiv F_{0i} \begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}.$$

For the case that $f_i \neq 0$ but $\alpha_i = 0$, an estimate of the norm $\|F_{0i}\|$ can be obtained directly later on, so we consider only the case that $f_i = 0$ at the moment. Since

$$\begin{aligned} F_{0i}^t F_{0i} &= \begin{pmatrix} 0 & 0 \\ -\sqrt{\alpha} \hat{Q}_{2i}^t & (I - Q_{2i}^0) \end{pmatrix} \begin{pmatrix} 0 & -\sqrt{\alpha} \hat{Q}_{2i} \\ 0 & (I - Q_{2i}^0) \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ 0 & \alpha (Q_{2i}^0)^2 + (I - Q_{2i}^0)^2 \end{pmatrix}, \end{aligned}$$

it suffices to estimate the maximum eigenvalue of the matrix

$$Q_{0i} = \alpha (Q_{2i}^0)^2 + (I - Q_{2i}^0)^2. \tag{3.22}$$

Using (1.2) and (3.17), we have

$$Q_{Bi}^{-1} C = Q_{Bi}^{-1} U \Sigma V^t V \Sigma^t U^t = Q_{Bi}^{-1} U \Sigma_0^2 U^t = (\Sigma_0 U^t)^{-1} Q_{2i}^0 (\Sigma_0 U^t). \tag{3.23}$$

Thus the matrix Q_{2i}^0 has the same eigenvalues as the matrix $Q_{Bi}^{-1}C$, and Lemma 3.2(ii) implies that the maximum eigenvalue of the matrix Q_{0i} defined in (3.22) is bounded above by the maximum of the function

$$\varphi(z) = \alpha z^2 + (1-z)^2, \quad z \in \left[\frac{(1-\beta)}{2\lambda_0}, \frac{(1+\beta)}{2}\right].$$

Here we have used the fact that $\theta_i = \frac{1}{2}$ for $f_i = 0$ by definition. Using (3.22), (3.4), and Lemmas 3.3 and 3.4 we have

$$\|F_{0i}\|^2 \le \alpha\gamma^2 + (1-\gamma)^2 \le F(\alpha^2) = \rho_0^2 \quad (\text{when } \ f_i = 0). \tag{3.24}$$

Next, we consider the case that $f_i \neq 0$ and $\alpha_i > 0$. Write (3.21) in the form

$$\begin{pmatrix} E_{i+1}^{xy} \\ E_{i+1}^{0y} \end{pmatrix} = \begin{pmatrix} \alpha_i(I+Q_{2i}) & -\sqrt{\alpha}\hat{Q}_{2i} \\ \sqrt{\alpha}^{-1}\alpha_i\hat{Q}_{2i}^t & (I-Q_{2i}^0) \end{pmatrix} \begin{pmatrix} \alpha_i^{-1}Q_{1i} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}.$$

By the definitions of Q_{1i}, E_i^{xy}, and α_i, we have (note that V^t is an orthogonal matrix)

$$\begin{aligned} \|\alpha_i^{-1}Q_{1i}E_i^{xy}\|^2 &= \|\alpha_i^{-1}\sqrt{\alpha}V^t(I-\omega_i A^{\frac{1}{2}}\hat{A}^{-1}A^{\frac{1}{2}})A^{-\frac{1}{2}}f_i\|^2 \\ &= \alpha_i^{-2}\alpha\|(I-\omega_i A^{\frac{1}{2}}\hat{A}^{-1}A^{\frac{1}{2}})A^{-\frac{1}{2}}f_i\|^2 \\ &= \alpha_i^{-2}\alpha\alpha_i^2\|A^{-\frac{1}{2}}f_i\|^2 \\ &= \|\sqrt{\alpha}V^tA^{-\frac{1}{2}}f_i\|^2 = \|E_i^{xy}\|^2. \end{aligned}$$

Thus

$$\begin{aligned} \left\|\begin{pmatrix} \alpha_i^{-1}Q_{1i} & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}\right\| &= \left\|\begin{pmatrix} \alpha_i^{-1}Q_{1i}E_i^{xy} \\ E_i^{0y} \end{pmatrix}\right\| \\ &= \left(\|\alpha_i^{-1}Q_{1i}E_i^{xy}\|^2 + \|E_i^{0y}\|^2\right)^{\frac{1}{2}} \\ &= \left\|\begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}\right\|. \end{aligned}$$

Therefore

$$\left\|\begin{pmatrix} E_{i+1}^{xy} \\ E_{i+1}^{0y} \end{pmatrix}\right\| \le \left\|\begin{pmatrix} \alpha_i(I+Q_{2i}) & -\sqrt{\alpha}\hat{Q}_{2i} \\ \sqrt{\alpha}^{-1}\alpha_i\hat{Q}_{2i}^t & (I-Q_{2i}^0) \end{pmatrix}\right\| \left\|\begin{pmatrix} E_i^{xy} \\ E_i^{0y} \end{pmatrix}\right\|.$$

It is clear that

$$\begin{pmatrix} \alpha_i(I+Q_{2i}) & -\sqrt{\alpha}\hat{Q}_{2i} \\ \sqrt{\alpha}^{-1}\alpha_i\hat{Q}_{2i}^t & (I-Q_{2i}^0) \end{pmatrix} = \begin{pmatrix} \alpha_i(I+Q_{2i}^0) & 0 & -\sqrt{\alpha}Q_{2i}^0 \\ 0 & \alpha_i I & 0 \\ \sqrt{\alpha}^{-1}\alpha_i Q_{2i}^0 & 0 & (I-Q_{2i}^0) \end{pmatrix}.$$

Let F_i be the matrix defined in Lemma 3.5 but with Q replaced by Q_{2i}^0; then we have

$$\left\|\begin{pmatrix} \alpha_i(I+Q_{2i}) & -\sqrt{\alpha}\hat{Q}_{2i} \\ \sqrt{\alpha}^{-1}\alpha_i\hat{Q}_{2i}^t & (I-Q_{2i}^0) \end{pmatrix}\right\| = \left\|\begin{pmatrix} \alpha_i I & 0 \\ 0 & F_i \end{pmatrix}\right\| = \max\{\alpha_i, \|F_i\|\} \le \max\{\alpha, \|F_i\|\}.$$

Noting that $\alpha \le \rho_0$ by the definition of ρ_0 and $|c(\gamma,\alpha)| \ge 0$, the desired estimate now follows from Lemma 3.5.

For the case that $f_i \neq 0$ and $\alpha_i = 0$, F_{0i} has the same form as F_i. Thus $\|F_{0i}\| \leq \rho_0$ by Lemma 3.5. This proves (2.7) for all possible cases.

Finally we show (2.9). We first claim that

$$|1-\gamma-\alpha(1+\gamma)| < 1-\alpha. \tag{3.25}$$

In fact, since

$$\lambda_0 \geq \kappa_1 = \frac{1+\alpha}{1-\alpha},$$

we have

$$\sqrt{1-\frac{1}{\lambda_0}} \geq \sqrt{\frac{2\alpha}{1+\alpha}} \geq \alpha.$$

Thus

$$\gamma = \frac{1-\beta}{2\lambda_0}\left(1-\sqrt{1-\frac{1}{\lambda_0}}\right) < 1-\alpha,$$

which implies (3.25) using $\gamma > 0$ and $\alpha < 1$. Now by (3.25) and the definition of ρ_0 in (2.7)

$$\rho_0 < \frac{|1-\gamma-\alpha(1+\gamma)|+(1+\alpha)}{2} = \begin{cases} \frac{1-\alpha-\gamma(1+\alpha)+(1+\alpha)}{2}, & 0<\gamma\leq\frac{1-\alpha}{1+\alpha}, \\ \frac{\gamma(1+\alpha)-(1-\alpha)+(1+\alpha)}{2}, & \frac{1-\alpha}{1+\alpha}<\gamma<1-\alpha\,. \end{cases}$$

This completes the proof of Theorem 2.1. □

Proof of Theorem 2.2. For ease of notation, we let

$$\tilde{Q}_{1i} = I - \omega_i A^{\frac{1}{2}} \hat{A}^{-1} A^{\frac{1}{2}}, \quad \tilde{Q}_{2i} = A^{-\frac{1}{2}} B Q_{Bi}^{-1} B^t A^{-\frac{1}{2}}.$$

Then (3.16) can be written as (replacing i by $i-1$)

$$A^{-\frac{1}{2}} f_i = (I+\widetilde{Q}_{2i})\tilde{Q}_{1i} A^{-\frac{1}{2}} f_{i-1} - \widetilde{Q}_{2i} A^{-\frac{1}{2}} B e_{i-1}^y.$$

Applying Young's inequality, we obtain for any positive η that

$$\|A^{-\frac{1}{2}} f_i\|^2 \leq (1+\eta)\|(I+\widetilde{Q}_{2i})\widetilde{Q}_{1i} A^{-\frac{1}{2}} f_{i-1}\|^2 + (1+\eta^{-1})\|\widetilde{Q}_{2i} A^{-\frac{1}{2}} B e_{i-1}^y\|^2. \tag{3.26}$$

By the proof of Theorem 2.1 we know that $\widetilde{Q}_{2i}$ has the same positive eigenvalues as the matrix $Q_{Bi}^{-1}C$. Hence, Lemma 3.2(ii) infers that the eigenvalues of $\widetilde{Q}_{2i}$ lie in the interval [0, 1], namely,

$$\|\widetilde{Q}_{2i}\| \leq 1, \quad \|I+\widetilde{Q}_{2i}\| \leq 2;$$

combining with (3.26) and Lemma 3.1, this leads to

$$\begin{aligned} \|A^{-\frac{1}{2}} f_i\|^2 &\leq (1+\eta)4\alpha^2 \|A^{-\frac{1}{2}} f_{i-1}\|^2 + (1+\eta^{-1})\|A^{-\frac{1}{2}} B e_{i-1}^y\|^2 \\ &= 4\alpha(1+\eta)\|\sqrt{\alpha} A^{-\frac{1}{2}} f_{i-1}\|^2 + (1+\eta^{-1})\|e_{i-1}^y\|_C^2\,; \end{aligned}$$

taking $\eta = (4\alpha)^{-1}$ and using Theorem 2.1, we have

$$\|A^{-\frac{1}{2}} f_i\| \leq \sqrt{1+4\alpha}\,\rho^{i-1}|||E_0|||.$$

Now Theorem 2.2 follows immediately from the identity $A^{\frac{1}{2}}e_i^x = A^{-\frac{1}{2}}f_i - A^{-\frac{1}{2}}Be_i^y$, the triangle inequality, and Theorem 2.1. □

We end this section with some remarks on the selection of the parameter θ_i in Algorithm 2.1. As we see, the parameter θ_i has been used in the convergence rate analysis (cf. the inequality (3.10)). We next illustrate in a more direct manner why we have to introduce such a parameter and why we suggest choosing θ_i using (2.5). It is easy to find out from the proof of Theorem 2.1 that the sufficient and necessary condition for Algorithm 2.1 to converge is $\|F_i\| < 1$, where F_i is essentially the propagation matrix of Algorithm 2.1. This is equivalent to the condition that $\lambda^* < 1$ (cf. 3.8), that is,

$$\sqrt{f^2(\alpha_i, \lambda_j) - 4\alpha_i^2} < 2 - f(\alpha_i, \lambda_j)$$

or

$$f^2(\alpha_i, \lambda_j) - 4\alpha_i^2 < 4 - 4f(\alpha_i, \lambda_j) + f^2(\alpha_i, \lambda_j), \quad f(\alpha_i, \lambda_j) \le 2.$$

Namely,

$$f(\alpha_i, \lambda_j) < 1 + \alpha_i^2.$$

By the definition of $f(\alpha_i, \lambda_j)$, this condition is equivalent to

$$0 < \lambda_j < \frac{2\alpha(1-\alpha_i^2)}{\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha}. \tag{3.27}$$

From Lemma 3.2(ii) and (3.23) we know that $\lambda_j \in [\theta_i(1-\beta)/\lambda_0, \theta_i(1+\beta)]$. Clearly (3.27) holds if θ_i is chosen such that

$$0 < \theta_i < \frac{2\alpha(1-\alpha_i^2)}{(\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha)(1+\beta)}. \tag{3.28}$$

But since the paramaters α, β, and α_i are not easily computable, it is impractical to choose θ_i using the criterion (3.28). To find a more practical way of choosing θ_i, we further relax the condition (3.27). By Lemma 3.1, we know $\alpha_i \le \alpha$; hence

$$\alpha\alpha_i^2 + \alpha_i^2 + \alpha^2 + \alpha = (1+\alpha)\,\alpha\,\left(1 + \frac{\alpha_i}{\alpha}\alpha_i\right) < 2\alpha(1+\alpha_i), \tag{3.29}$$

so (3.27) is still satisfied if

$$0 < \lambda_j \le 1 - \alpha_i, \quad j = 1, \ldots, m. \tag{3.30}$$

For this we need to choose θ_i such that

$$0 < \theta_i(1+\beta) \le 1 - \alpha_i, \quad j = 1, \ldots, m; \tag{3.31}$$

this, with the relation $\alpha_i < \sqrt{1-\omega_i}$ from Lemma 3.1, yields the following selection criterion for θ_i:

$$\theta_i \le \frac{1 - \sqrt{1-\omega_i}}{2}. \tag{3.32}$$

Namely, any positive θ_i satisfying (3.32) guarantees the convergence of Algorithm 2.1. However, using (3.8) and the monotone decreasing property of $f(\alpha_i, z)$ for $z < \beta_0$ we

know that the larger the parameter θ_i is, the faster Algorithm 2.1 converges, namely, the choice

$$\theta_i < \frac{1-\sqrt{1-\omega_i}}{2} \quad \left(\leq \frac{1-\alpha_i}{2} \leq \beta_0\right)$$

will result in a convergence slower than the equality case. This is why we choose the equality case for θ_i in Theorem 2.1.

Note that the condition (3.32) is very conservative and it is obtained under the worst case: $\alpha \to 1^-$ (cf. (3.29)) and $\beta \to 1^-$ (cf. (3.31)). Therefore the choice

$$\theta_i > \frac{1-\sqrt{1-\omega_i}}{2}$$

is also possible. We omit the detailed discussion about this possibility here.

Finally, we add the additional observation that when α is small the condition (3.27) becomes $0 < \lambda_j < 2$ (the last term of (3.27) tends to 2^- as $\alpha \to 0$), which is satisfied if $\theta_i(1+\beta) < 2$ or $\theta_i \leq 1$. Thus we can take $\theta_i = \omega_i \leq 1$ to speed up the convergence of Algorithm 2.1 in this case.

Summarizing the above, and noting that

$$0.25\omega_i < \frac{1-\sqrt{1-\omega_i}}{2} < 0.5\omega_i\,,$$

we can conclude that the convergence of Algorithm 2.1 will speed up in the following order:

$$\theta_i = 0.25\omega_i\,,\ \frac{1-\sqrt{1-\omega_i}}{2},\ 0.5\omega_i\,,\ \omega_i$$

in the case that Algorithm 2.1 converges with $\theta_i = 0.5\omega_i$ and ω_i. This matches well with our numerical results; see Tables 4.1 and 4.2.

4. Numerical experiments. In this section, we apply our new Algorithm 2.1 of section 2, Algorithm 1.1 of [4], and the preconditioned MINRES method [18] to solve the two-dimensional generalized Stokes problem and a system of purely algebraic equations. Let Ω be the unit square in R^2, and $L_0^2(\Omega)$ be the set of all square integrable functions with zero mean values over Ω, and let $H^1(\Omega)$ be the usual Sobolev space of order one. The space $H_0^1(\Omega)$ consists of those functions in $H^1(\Omega)$ with vanishing traces on $\partial\Omega$.

Our first example is the generalized Stokes problem whose variational formulation reads as follows: Find $(u,p) \in (H_0^1(\Omega))^2 \times L_0^2(\Omega)$ such that

$$(\mu(x)\nabla u, \nabla v) - (p, \nabla\cdot v) = (f,v), \quad \text{for all } v \in (H_0^1(\Omega))^2, \tag{4.1}$$

$$(q, \nabla\cdot u) = (q,g), \quad \text{for all } q \in L_0^2(\Omega), \tag{4.2}$$

where $f \in (L^2(\Omega))^2$, $g \in L^2(\Omega)$, and $\mu \in L^\infty(\Omega)$ with $\mu(x) \geq c > 0$ almost everywhere in Ω.

We use one of the well-known conforming Taylor–Hood elements, which have been widely used in engineering, to solve the system (4.1)–(4.2). For any positive integer N, a triangulation $\mathcal{T}^h$ of Ω is obtained by dividing Ω into $N \times N$ subsquares with side lengths of $h = 1/N$. Let $X_h \subset H_0^1(\Omega)$ and $M_h \subset H^1(\Omega) \cap L_0^2(\Omega)$ be the usual continuous Q_2 and Q_1 finite element spaces defined on $\mathcal{T}^h$, respectively

TABLE 4.1
Number of iterations for Algorithm 2.1.

N	$\theta_i = \omega_i^{-1}$	$\theta_i = 1$	$\theta_i = \omega_i$	$\theta_i = 0.5\omega_i$	$\theta_i = \frac{1-\sqrt{1-\omega_i}}{2}$	$\theta_i = 0.25\omega_i$
8	638	203	35	39	41	46
16	154	44	36	41	42	46
32	153	45	36	40	42	46
48	154	45	37	40	41	47
64	154	44	36	41	42	46

TABLE 4.2
Number of iterations for Algorithm 1.1 *(left) and the MINRES method (right).*

N	8	16	32	48	64
Alg. 1.1	917	300	58	93	95

N	8	16	32	48	64
MINRES	63	55	51	50	50

N	8	16	32	48	64
Alg. 1.1	92	85	76	75	75

N	8	16	32	48	64
MINRES	56	65	65	66	66

(cf. [6, 10]). The total number of unknowns for this finite element is $n+m = [2(2N-1)^2] + [(N+1)^2 - 1]$; e.g., the total unknowns are 36482 for $N = 64$. The finite element approximation of the above Stokes system can be formulated as follows: Find $(u_h, p_h) \in X_h^2 \times M_h$ such that

$$(\mu(x)\nabla u_h, \nabla v) - (p_h, \nabla\cdot v) = (f, v), \quad \text{for all } v \in X_h^2, \tag{4.3}$$

$$(q, \nabla\cdot u_h) = (q, g), \quad \text{for all } q \in M_h. \tag{4.4}$$

It is known that the inf-sup condition is satisfied by the pair (X_h^2, M_h) (see [6]), thus the Schur complement matrix $C = B^t A^{-1} B$ associated with the system (4.3)–(4.4) has a condition number independent of h. As in [5], [18], we take the variable coefficient μ to be $\mu = 1 + x_1x_2 + x_1^2 - x_2^2/2$. We know that the corresponding matrix A is block diagonal with two copies of a discrete Laplace operator on the diagonal if $\mu = 1$, and so it can be solved by the fast Poisson solver. Therefore it is natural to choose this fast solver $\hat{A}$ as the preconditioner of A. In fact, the matrix $\hat{A}^{-1}A$ is well-conditioned since we have

$$0.5\,(\hat{A}z, z) \le (Az, z) \le 2.5\,(\hat{A}z, z). \tag{4.5}$$

Thus the matrix $B^t\hat{A}^{-1}B$ is also well-conditioned. In fact, it is spectrally equivalent to $h^2 I$ (cf. [19]); that is, we can choose $\hat{C} = h^2 I$.

In most applications, the condition numbers κ_1 and κ_2 are not very large; otherwise all iterative methods for the saddle-point problems perform without any essential difference. It is clear that the parameter ω_i has a small range in this case, and we can roughly estimate the maximum and minimum eigenvalues of the matrix $\hat{A}^{-1}A$ based on several values of ω_i. In fact, when the system (4.3)–(4.4) is solved by Algorithm 2.1 with these preconditioners, the computational results (set $\theta_i = 1$) indicate that the parameter ω_i lies between 0.46 and 0.93 for $1 \le i \le 4$, which reflects roughly the range of the eigenvalues of the matrix $\hat{A}^{-1}A$.

In order to see whether assumption (2.6) is necessary for the convergence of Algorithm 2.1, we do not scale the preconditioner $\hat{A}$, so condition (2.6) is violated. The numerical results show that our Algorithm 2.1 converges well; the number of iterations is listed in Table 4.1. Note that all the initial guesses for the algorithms tested in this section are taken to be zero and the algorithms are terminated when

the following relative error reaches 1.0×10^{-5}:

$$\varepsilon = \frac{\|Mu_i - b\|}{\|Mu_0 - b\|},$$

where M and $b = (b_1\ b_2)^t$ are the coefficient matrix and the right-hand side vector of the algebraic system corresponding to (4.3)–(4.4) and $u_i = (x_i\ y_i)^t$ is the ith iterate of the algorithms to be tested. Here we take the vector $b = Mu$ with the solution $u = (x\ y)^t$, and x and y are two vectors with all components being 1.0 and 0.5, respectively. From Table 4.1 we can see the importance of choosing a different θ_i other than $\theta_i = \omega_i^{-1}$. Also, one can find out that the convergence of Algorithm 2.1 is nearly independent of the mesh size h.

The inexact Uzawa Algorithm 1.1 is convergent if the two preconditioners for A and C satisfy the conditions (3.2) and (2.3) of [4]. Using (4.5), one can verify that these two conditions are indeed satisfied if we take the two preconditioners to be $2.5\hat{A}$ and $2I$ for A and C, respectively. Thus, we can also apply Algorithm 1.1 to solve the system (4.3)–(4.4). However, the convergence is a bit slow; see Table 4.2 (upper left). When the preconditioner $2I$ for C is replaced by $h^2 I$, which is spectrally equivalent to C (cf. [19]), Algorithm 1.1 converges slightly faster; see Table 4.2 (lower left). The main reason for the slow convergence in this case is that the parameter γ defined by (2.4) of [4] is close to one. Also it is difficult to achieve an accurate estimate on this parameter γ because of the difficulty of estimating the maximum eigenvalue of the matrix $\hat{C}^{-1}C$.

Then we applied the preconditioned MINRES method (cf. [16], [18]) with a block diagonal preconditioner with diagonal blocks being $\hat{A}$ and $\hat{C} = 0.01I$ or $\hat{C} = h^2 I$ (spectrally equivalent to C; cf. [19]) to solve the system (4.3)–(4.4). The number of iterations is listed in the upper right of Table 4.2 for $\hat{C} = 0.01I$ and in the lower right for $\hat{C} = h^2 I$. We remark that different constant scalings for $\hat{C}$ affect the convergence of the MINRES method greatly; see the comments at the end of this section.

Our second example is a system of purely algebraic equations. We define the matrices $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times m}$ $(n \geq m)$ in (1.1) as follows:

$$a_{ij} = \begin{cases} i+1, & i = j, \\ 1, & |i-j| = 1, \\ 0, & \text{otherwise;} \end{cases} \qquad b_{ij} = \begin{cases} j, & i = j+n-m, \\ 0, & \text{otherwise.} \end{cases}$$

The preconditioners $\hat{A} = (\hat{a}_{ij})_{n\times n}$ and $\hat{C} = (\hat{c}_{ij})_{m\times m}$ are defined by

$$\hat{a}_{ij} = \begin{cases} i+2, & i = j\,, \\ 0, & i \neq j\,; \end{cases} \qquad \hat{c}_{ij} = \begin{cases} k\,(i^2+3), & i = j\,, \\ 0, & i \neq j\,, \end{cases}$$

where k is a scaling constant. The right-hand side vectors f and g are taken such that the exact solutions x and y are both vectors with all components being 1.

Assumption (2.6) is violated again with this example. However, Algorithm 2.1 still converges well; see the number of iterations listed in Table 4.3. The convergence of Algorithm 1.1 and the preconditioned MINRES method with two different scaling constants, $k = 1, 1/200$, are reported in Tables 4.4 and 4.5.

TABLE 4.3
Number of iterations for Algorithm 2.1.

n	m	$\theta_i = \omega_i^{-1}$	$\theta_i = 1$	$\theta_i = \omega_i$	$\theta_i = 0.5\omega_i$	$\theta_i = \frac{1-\sqrt{1-\omega_i}}{2}$	$\theta_i = 0.25\omega_i$
200	150	15	15	15	17	19	38
400	300	16	16	16	17	18	38
800	600	17	17	17	18	18	38
1600	1200	17	17	17	17	18	39

TABLE 4.4
Iterations for Algorithm 1.1 *with different scalings:* $k = 1,\ 1/200$.

n	200	400	800	1600
m	150	300	600	1200
$k = 1$	1892	3759	> 5000	> 5000

n	200	400	800	1600
m	150	300	600	1200
$k = 1/200$	diverge	24	34	71

TABLE 4.5
Iterations for the preconditioned MINRES method with different scalings $k = 1,\ 1/200$.

n	200	400	800	1600
m	150	300	600	1200
$k = 1$	33	35	38	39

n	200	400	800	1600
m	150	300	600	1200
$k = 1/200$	22	22	22	23

From the above numerical results and many more tests we have not reported here, one can observe that different scalings for the preconditioner $\hat{C}$ greatly affect the convergence of Algorithm 1.1 and the preconditioned MINRES method. For example, Algorithm 1.1 converges (slowly) when the scaling constant $k = 1$, but it may diverge (the errors do not decrease) when $k = 1/200$; see Table 4.4. Such behaviors also happen for the preconditioned MINRES method (cf. [16], [18] and also see Table 4.5), whose convergence rate is known to depend on the ratio $\lambda_{\min}/\lambda'_{\min}$, where $\lambda_{\min}$ and $\lambda'_{\min}$ are, respectively, the minimal eigenvalues of $\hat{A}^{-1}A$ and $\hat{C}^{-1}H$ with $H = B^t\hat{A}^{-1}B$ (cf. [18]). So it is important for these algorithms to have good a priori estimates on the minimum or maximum eigenvalues of the matrix $\hat{C}^{-1}C$ or $\hat{C}^{-1}H$ in order to find an effective scaling for the preconditioner $\hat{C}$. But such a priori estimates are usually very difficult to achieve in practical applications, even when $\hat{C}^{-1}C$ is well-conditioned; e.g., this is the case with our first example; see the system (4.3)–(4.4). One of the advantages of our Algorithm 2.1 is to have relieved such a troublesome estimate, and different scalings for the preconditioner $\hat{C}$ do not affect the convergence of our Algorithm 2.1, which is easily seen from the algorithm itself.

Acknowledgments. The authors wish to thank the anonymous referees for many constructive comments that improved the paper greatly.

REFERENCES

[1] K. ARROW, L. HURWICZ, AND H. UZAWA, *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, 1958.
[2] O. AXELSSON, *Numerical algorithms for indefinite problems*, in Elliptic Problem Solvers, Academic Press, New York, 1984, pp. 219–232.
[3] R. BANK, B. WELFERT, AND H. YSERENTANT, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.
[4] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[5] J. Bramble and J. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–18.
[6] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
[7] X. Chen, *On preconditioned Uzawa methods and SOR methods for saddle-point problems*, J. Comput. Appl. Math., 100 (1998), pp. 207–224.
[8] Z. Chen, Q. Du, and J. Zou, *Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients*, SIAM J. Numer. Anal., 37 (2000), pp. 1542–1570.
[9] Z. Chen and J. Zou, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM J. Control Optim., 37 (1999), pp. 892–910.
[10] P. Ciarlet, *Basic error estimates for elliptic problems*, in Handbook of Numerical Analysis, Vol. II, P. Ciarlet and J.-L. Lions, eds., North-Holland, Amsterdam, 1991, pp. 17–351.
[11] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.
[12] V. Girault and P. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer-Verlag, Berlin, 1986.
[13] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Math. 9, SIAM, Philadelphia, 1989.
[14] Q. Hu, G. Liang, and P. Sun, *Solving parabolic problems by domain decomposition methods with Lagrangian multipliers*, Math. Numer. Sin., 22 (2000), pp. 241–256.
[15] Y. Keung and J. Zou, *Numerical identifications of parameters in parabolic systems*, Inverse Problems, 14 (1998), pp. 83–100.
[16] A. Klawonn, *An optimal preconditioner for a class of saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 540–552.
[17] W. Queck, *The convergence factor of preconditioned algorithms of the Arrow–Hurwicz type*, SIAM J. Numer. Anal., 26 (1989), pp. 1016–1030.
[18] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.
[19] A. Wathen and D. Silvester, *Fast iterative solution of stabilised Stokes systems. Part* I: *Using simple diagonal preconditioners*, SIAM J. Numer. Anal., 30 (1993), pp. 630–649.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 339–348
© 2001 Society for Industrial and Applied Mathematics

SPECTRAL ANALYSIS OF (SEQUENCES OF) GRAPH MATRICES*

ANTONIO FRANGIONI† AND STEFANO SERRA CAPIZZANO‡

Abstract. We study the extreme singular values of incidence graph matrices, obtaining lower and upper estimates that are asymptotically tight. This analysis is then used for obtaining estimates on the spectral condition number of some weighted graph matrices. A short discussion on possible preconditioning strategies within interior-point methods for network flow problems is also included.

Key words. graph matrices, conditioning, preconditioning

AMS subject classifications. 05C50, 15A12, 65F10

PII. S089547989935366X

1. Introduction.

We study graph matrices coming from the application of interior-point methods [17, 14], which have grown a well-established reputation as efficient algorithms for large-scale problems. In these methods, at each step we have to solve linear systems of the form

$$E\Theta E^T x = b, \tag{1.1}$$

where E is an $n \times m$ matrix and Θ is an $m \times m$ diagonal positive definite matrix. In most general-purpose solvers, these linear systems are solved by means of direct methods, typically the Cholesky decomposition preceded by a heuristic reordering of the columns of E aimed at minimizing the "fill-in" [17]. We are interested in the possibility of using iterative methods instead. This can be beneficial in practice, especially in cases when E is a sparse structured matrix [7] such as the node-arc incidence matrix of a graph [15, 16]. However, these approaches can be competitive only if the rate of convergence of the iterative method is sufficiently high. This motivates our study of the extreme singular values of E and of the spectral behavior of $E\Theta E^T$ since the convergence rate of iterative methods largely depends on the conditioning of the matrix. This analysis may have an interest for the development of preconditioners [15, 16] for the numerical solution to (1.1) through a preconditioned conjugate gradient (PCG) method (for the convergence theory of the PCG method, refer to [3]).

The paper is organized as follows. In section 2 we study the spectral properties (extremal behavior and conditioning) of EE^T when E is the node-arc incidence matrix of a directed graph. In section 3 we extend the analysis to "weighted" matrices of the form $E\Theta E^T$. Finally, in section 4 the connections between this analysis and some possible preconditioning strategies are briefly discussed.

2. Graph matrices.

Let $H \equiv H_n = (\mathcal{U}_n, \mathcal{V}_n)$ be a *directed* graph with n nodes $\mathcal{U}_n = \{u_1, \ldots, u_n\}$ and m arcs $\mathcal{V}_n = \{v_1, \ldots, v_m\}$; its node-arc incidence matrix $E \equiv E_n = E(H_n)$ is the $n \times m$ matrix such that $E_{ij} = 1$ if v_j emanates from u_i, $E_{ij} = -1$ if v_j terminates at u_i, and $E_{ij} = 0$ otherwise.

*Received by the editors June 8, 1999; accepted for publication (in revised form) by L. El Ghaoui December 5, 2000; published electronically August 8, 2001.
http://www.siam.org/journals/simax/23-2/35366.html

†Dipartimento di Informatica, Corso Italia 40, 56100 Pisa, Italy (frangio@di.unipi.it).
‡Dipartimento di Chimica, Fisica e Matematica, Via Valleggio 11, 22100 Como, Italy (serra@mail.dm.unipi.it).

Here we analyze the spectral properties of sequences of matrices $\{E_n E_n^T\}_n$. Clearly, this both requires and implies the study of the spectra of (sequences of) graph matrices $\{E_n\}_n$. This analysis has an interest of its own, as demonstrated by the literature on the subject [1, 9]. However, the usual approach has most often been of strongly combinatorial flavor and for a fixed graph size n. By contrast, our analysis focuses on asymptotical results, for which little or no previous work seems to have been published.

2.1. Preliminary results. The EE^T matrix that we study is closely related to the *Laplacian* of an *undirected* graph $\bar{H} \equiv \bar{H}_n = (\bar{\mathcal{U}}_n, \bar{\mathcal{V}}_n)$ [1], i.e., the $n \times n$ matrix $L \equiv L_n = L(\bar{H}_n)$ such that L_{ii} is the degree (number of incident arcs) of node u_i and L_{ij} for $i \neq j$ is -1 if the arc (i,j) belongs to $\bar{\mathcal{V}}_n$ and zero otherwise. It is easy to prove the following relation between L and EE^T.

PROPOSITION 2.1. *Given an undirected graph $\bar{H} = (\bar{\mathcal{U}}, \bar{\mathcal{V}})$, the directed graph $H = (\bar{\mathcal{U}}, \mathcal{V})$ with $\mathcal{V} = \{ (i,j) : (i,j) \in \bar{\mathcal{V}},\ i < j \}$ has $E(H)E(H)^T = L(\bar{H})$.*

In other words, the Laplacian of an undirected graph $\bar{H}$ can be obtained as $E(H)E(H)^T$, where H is the directed graph obtained from $\bar{H}$ by orienting each arc in such a way that the head node is smaller than the tail node (with any fixed ordering of $\bar{\mathcal{U}}$). Conversely, the $E(H)E(H)^T$ matrix of a generic directed graph H can be seen as being obtained from the Laplacians of *two* undirected graphs.

PROPOSITION 2.2. *Given a directed graph $H = (\mathcal{U}, \mathcal{V})$, the two undirected graphs $\bar{H}_1 = (\mathcal{U}, \bar{\mathcal{V}}_1)$ and $\bar{H}_2 = (\mathcal{U}, \bar{\mathcal{V}}_2)$ with*

$$\bar{\mathcal{V}}_1 = \{ (i,j) : (i,j) \in \mathcal{V},\ i < j \},$$

$$\bar{\mathcal{V}}_2 = \{ (i,j) : (i,j) \in \mathcal{V},\ j < i \}$$

are such that $E(H)E(H)^T = L(\bar{H}_1) + L(\bar{H}_2)$.

Therefore, for the purpose of the analysis of the $E(H)E(H)^T$ matrices, a directed graph H can be seen as the composition of two undirected graphs. One of the two graphs contains (as undirected edges) the arcs having a head node smaller than the tail node, while the other graph contains (as undirected edges) the arcs having a head node larger than the tail node.

Thus, Laplacians of undirected graphs and $E(H)E(H)^T$ matrices of directed graphs can be related through appropriate (de)orientation of the arcs. We will therefore be able to exploit some interesting results about the spectra of Laplacians such as the following.

THEOREM 2.3 (see [1]). *For any undirected graph $\bar{H}$, $\lambda_{\max}(L(\bar{H})) \leq n$.*

It is immediate to verify that summing all the rows of E_n gives the null vector. This proves that $\lambda_{\min}(E_n E_n^T) = 0$ and therefore $\sigma_{\min}(E_n) = 0$ if $m \geq n$. However, if H_n is a connected graph, then the matrix obtained by E_n by eliminating any row has full rank. If H_n has k maximal connected components, then $E_n = E(H_n)$ is a block diagonal matrix with k blocks; the minimal (maximal) singular value of E_n is the minimum (maximum) among the minimal (maximal) singular values of the submatrices associated to the connected components. Hence, we can restrict our analysis to *connected* graphs. Note that $E_n E_n^T$ has exactly k zero eigenvalues: by deleting k appropriate rows of E_n (one for each of the connected components), one can always obtain a matrix with no zero singular values.

We can always reorder the nodes and the arcs in such a way that the square submatrix $S \equiv S_n = S(H_n)$ made of the first $n-1$ rows of E is nonsingular. In fact,

S_n is the node-arc incidence matrix of a spanning tree of H_n less one row, for which the following results hold.

PROPOSITION 2.4 (see [9]). *S_n is nonsingular and totally unimodular, i.e., the determinant of each square submatrix belongs to $\{\pm 1, 0\}$.*

PROPOSITION 2.5 (see [9]). *The entries of S_n^{-1} belong to $\{\pm 1, 0\}$.*

2.2. Conditioning of trees. We start by studying the special case when H is a tree, i.e., $m = n - 1$ (H is connected). We do not require the arcs to have a specific orientation since the matrix $E^{'} = E(H^{'})$, corresponding to the *directed* tree $H^{'}$ obtained from H by reorienting the arcs, can be obtained from E by right multiplication for an $m \times m$ diagonal $\{-1, +1\}$ matrix. By the singular value decomposition [4, 12], E and $E^{'}$ have the same set of singular values; therefore, from the spectral viewpoint the directed tree $H^{'}$ can be considered a special representative of an equivalence class.

THEOREM 2.6. *The conditioning of S_n and E_n satisfies*

$$\overline{\delta}(n-1)^{1/2} \leq \kappa_2(S_n) \leq \sqrt{2n}(n-1),$$

$$\frac{\overline{\delta}}{\sigma_{n-2}(S_n)} \leq \kappa_2(E_n) \leq \sqrt{2n}(n-1),$$

where $\overline{\delta} = \sqrt{\delta(H_n)}$ and $\delta(H_n)$ is the maximum degree among all nodes in H_n. Indeed

R1. $\sigma_{\min}(S_n) \leq (n-1)^{-1/2}$;
R2. $\sigma_{\min}(S_n) \geq (n-1)^{-1}$;
R3. $\sigma_{\max}(S_n) \leq (2n)^{1/2}$;
R4. $\sigma_{\max}(S_n) \geq \overline{\delta} \geq \sqrt{2}$.

Proof.

Part R1. By the singular value decomposition of S_n

$$\sigma_{\min}(S_n) = \inf_{\|x\|_2 > 0} \frac{\|x^T S_n\|_2}{\|x\|_2} \leq \frac{\|e^T S_n\|_2}{\|e\|_2} = \frac{1}{(n-1)^{1/2}},$$

where e is the vector of all ones.

Part R2. By Proposition 2.5, $|[S_n^{-1}]_{i,j}| \leq 1$, hence the entries of $B_n = S_n^{-T} S_n^{-1}$ cannot exceed $n - 1$. Therefore, $\|B_n\|_1 \leq (n-1)^2$; since B_n is positive definite, its maximal eigenvalue coincides with its spectral norm and is less than its $\|\cdot\|_1$ norm, hence

$$\lambda_{\max}(B_n) \leq (n-1)^2,$$

but $\lambda_{\max}(B_n) = \lambda_{\max}(S_n^{-1})^2 = (1/\lambda_{\min}(S_n))^2$.

Part R3. From Proposition 2.2, we know that there exist two undirected graphs $\bar{H}_1$ and $\bar{H}_2$ such that $E_n E_n^T = L(\bar{H}_1) + L(\bar{H}_2)$. Thus, using Theorem 2.3 and the fact that S_n is a submatrix of E_n,

$$\sigma_{\max}^2(S_n) \leq \sigma_{\max}^2(E_n) = \lambda_{\max}(E_n E_n^T) \leq \lambda_{\max}(L(\bar{H}_1)) + \lambda_{\max}(L(\bar{H}_1)) \leq 2n\ .$$

Part R4. Let u_h be one of the nodes with maximum degree: it is always possible to reorient the arcs in such a way that u_h becomes the root, i.e., it only has outgoing arcs. Then, let e_h be the hth vector of the canonical basis; by the singular value decomposition of S_n

$$\sigma_{\max}(S_n) = \sup_{\|x\|_2 > 0} \frac{\|x^T S_n\|_2}{\|x\|_2} \geq \frac{\|e_h^T S_n\|_2}{\|e_h\|_2} = \frac{\sqrt{\delta(H_n)}}{1}\ .$$

Note that $\delta(H_n) \geq 2$ since H_n is connected.

The bounds on the condition numbers of S_n are simple consequences of **R1**–**R4** above. For the minimal and the maximal singular values of E_n, as well as its asymptotic conditioning, note that S_n is a submatrix of E_n. We can apply a rewording of the Cauchy interlacing theorem that holds for non-Hermitian matrices [8]. In particular, the following relations hold:

$$\sigma_{n-2}(S_n) \geq \sigma_{\min}(E_n) \geq \sigma_{\min}(S_n), \tag{2.1}$$

$$\sigma_{\max}(S_n) \leq \sigma_{\max}(E_n) \leq \sqrt{2n}. \qquad \square \tag{2.2}$$

The estimates **R1**–**R4** are, up to positive constants, tight: the following special structures are the "extremes" that prove it.

2.2.1. Linear trees. H_n is a linear tree if it is a path, i.e., each node but two has exactly two incident arcs. We can assume that the path is oriented from the root to the unique leaf and that the nodes are ordered accordingly; thus, we obtain a bidiagonal matrix E_n. The corresponding S_n is the $(n-1)\times(n-1)$ square Toeplitz matrix generated by the symbol $f(x) = 1 - e^{\mathbf{i}x}$ [18, 6]. $f(x)$ is weakly sectorial [6] and has a zero of order 1; therefore, the analysis in [6] shows that

$$\sigma_{\min}(S_n) \sim n^{-1},$$

$$\sigma_{\max}(S_n) \leq \|f\|_\infty = 2,$$

$$\lim_{n\to\infty} \sigma_{\max}(S_n) = \|f\|_\infty = 2\ .$$

Hence, **R2** and **R4** are tight (up to suitable multiplicative constants) for linear trees. These estimates can even be refined a little bit by studying the matrix $S_n^T S_n$. Direct calculation shows that

$$S_n^T S_n = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} = T_{n-1} - e_{n-1}e_{n-1}^T, \tag{2.3}$$

where T_{n-1} is the $(n-1)\times(n-1)$ Toeplitz matrix generated by the symbol $f(x) = 2 - 2\cos(x)$. T_{n-1} belongs to the τ algebra [5], so that its eigenvalues are explicitly known:

$$\lambda_{\min}(T_{n-1}) = 4\sin^2\left(\frac{\pi}{2n}\right), \quad \lambda_{\max}(T_{n-1}) = 4\sin^2\left(\frac{\pi(n-1)}{2n}\right)\ .$$

Note that $S_n^T S_n \leq T_{n-1}$ in the sense of the partial ordering of the Hermitian matrices; hence,

$$\sigma_{\min}(S_n) \leq \pi/n, \quad \sigma_{\max}(S_n) = 2 - \epsilon_n \quad \text{with } \epsilon_n \geq \pi/n\ .$$

Finally, since $E_n^T E_n = T_{n-1}$, we have

$$\sigma_{\min}(E_n) = 2\sin\left(\frac{\pi}{2n}\right), \quad \sigma_{\max}(E_n) = 2\sin\left(\frac{\pi(n-1)}{2n}\right)\ .$$

Remark 2.1. Observe that the constant $\|f\|_\infty = 2$ is exactly the maximum node degree of a linear tree. Therefore, in the case of linear trees the lower bound in **R4** is not tight and it can be replaced by $\delta(H_n)$ minus an asymptotically small quantity.

2.2.2. Star trees. In the opposite direction, we have "concentrated" trees, the most concentrated one being the "star" tree where the root has $n-1$ sons. Let us choose any ordering for the nodes where the first node is the root, and let us order the arcs according to the chosen order of the nodes. The resulting E_n is not lower triangular, but the corresponding S_n has the following interesting structure:

$$S_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & -1 & 0 \end{bmatrix} . \tag{2.4}$$

This structure is close to that of the Frobenius matrices [4], and it is easy to prove that the characteristic polynomial is $p(\lambda) = (1+\lambda^n)/(1+\lambda)$. However, S_n is "highly nonnormal" [4]; therefore the fact that all its eigenvalues have unitary modulus does not tell anything about its conditioning. As in the previous case, we can extract information on S_n by studying the matrix

$$S_n S_n^T = I_{n-1} + V_n, \quad \text{where} \quad V_n = [f|e_1] \cdot [e_1|g]^T$$

and $g = -\sum_{j=2}^{n-1} e_j$, $f = (n-2)e_1 + g$. Since V_n has rank two, $S_n S_n^T$ has eigenvalues 1 with multiplicity $n-3$ plus two other values that can be explicitly calculated. The nonzero eigenvalues of V_n are those of the 2×2 matrix [4]

$$[e_1|g]^T \cdot [f|e_1] = \begin{bmatrix} n-2 & 1 \\ n-2 & 0 \end{bmatrix} .$$

Direct calculation yields

$$\lambda_{\max}(S_n S_n^T) = n - 2 + O(n^{-1}),$$

$$\lambda_{\min}(S_n S_n^T) = 2/(n-2) + O(n^{-2}) .$$

Therefore

$$\sigma_{\max}(S_n) = \sqrt{n - 2 + O(n^{-1})},$$

$$\sigma_{\min}(S_n) = \sqrt{\frac{2}{n-2} + O(n^{-2})} ,$$

proving that **R1** and **R3** are tight up to suitable multiplicative constants.

Remark 2.2. The root of a star tree has degree $n-1$; hence, $\sigma_{\max}(S_n) = \sqrt{n-2+O(n^{-1})} = \sqrt{\delta(H_n)} + O(1)$ proves that **R4** cannot be relaxed any further. Thus, $\overline{\delta}$ cannot be replaced by $\delta(H_n)$ as "substantially" done for linear trees (see Remark 2.1).

Remark 2.3. The case of "star" trees shows that the lower bound

$$\frac{\overline{\delta}}{\sigma_{n-2}(S_n)} \leq \kappa_2(E_n)$$

on the condition numbers in Theorem 2.6 is tight. Indeed $\overline{\delta} = \sqrt{n-1}$, $\sigma_{n-2}(S_n) = 1$, and $E_n E_n^T = I_{n-1} + ee^T$ so that $\sigma_{\min}(E_n) = 1$, $\sigma_{\max}(E_n) = \sqrt{n}$, and therefore $\kappa_2(E_n) = \sqrt{n}$ which is in good agreement with the bound.

Remark 2.4. Tightness of **R1**–**R4** does not imply that the upper estimates on the spectral conditioning of E_n in Theorem 2.6 are tight. In fact, "linear" trees have $O(n)$ condition numbers and "star" trees have $O(\sqrt{n})$ condition numbers, as opposed to the $O(n^{3/2})$ bound given in the theorem. Finally, notice that the conditioning of S_n and E_n are asymptotically the same for "linear" trees while for "star" trees there is a substantially different behavior since $\kappa_2(E_n) = \sqrt{n}$ while $\kappa_2(S_n)$ grows as n.

2.3. Conditioning of graphs. The results of the previous section can be used in order to evaluate the extremal behavior of the singular values of $E(H_n)$ when H_n is a generic graph with n nodes. Since trees have been analyzed before, we will reduce the case of connected graphs to the case of trees.

PROPOSITION 2.7. *Let H_n be a connected graph: then*

$$\sigma_{n-1}(E_n) \geq \max_{T \in T(H_n)} \sigma_{\min}(S_n(T)),$$

where $T(H_n)$ is the set of the spanning trees of H_n.

Proof. Let T be a generic spanning tree of H_n: reorder the nodes and the arcs of H_n in such a way that T is represented by the first $n-1$ columns of E_n, and $S_n(T)$ is represented by the first $n-1$ rows and columns. Therefore, we have

$$\begin{aligned}
\sigma_{n-1}(E_n) &= \sup_{\dim \mathrm{U}=n-1} \inf_{x \in U,\ \|x\|>0} \frac{\|x^T E_n\|_2}{\|x\|_2} \geq \inf_{y \in \mathcal{R}^{n-1},\ \|y\|>0} \frac{\|[y^T, 0] E_n\|_2}{\|[y,0]\|_2} \\
&= \inf_{y \in \mathcal{R}^{n-1},\ \|y\|>0} \frac{\sqrt{\|y^T S_n(T)\|_2^2 + \|w\|_2^2}}{\|y\|_2} \\
&\geq \inf_{y \in \mathcal{R}^{n-1},\ \|y\|>0} \frac{\|y^T S_n(T)\|_2}{\|y\|_2} = \sigma_{\min}(S_n(T)) \,. \qquad \square
\end{aligned}$$

From Proposition 2.7 and part **R2** of Theorem 2.6, we obtain

$$\sigma_{n-1}(E_n) \geq n^{-1}. \tag{2.5}$$

On the other hand, using Proposition 2.2 and Theorem 2.3 as in part **R3** of Theorem 2.6, one obtains

$$\sigma_{\max}(E_n) \leq (2n)^{1/2}. \tag{2.6}$$

As a consequence, the following theorem holds.

THEOREM 2.8. *$\kappa_2(E_n)$ grows at most as $n^{3/2}\sqrt{2}$.*

These bounds are asymptotically tight: linear trees realize (2.5), while (2.6) is realized by complete graphs. In fact, the matrix $E_n E_n^T$ corresponding to a complete graph H_n is the circulant matrix $2(nI - ee^T)$, whose maximal eigenvalue is $2n$ with multiplicity $n-1$ (all the nonzero vectors orthogonal to e are eigenvectors associated to the eigenvalue $2n$) and whose minimal eigenvalue is zero [10].

The bound on the condition number in Theorem 2.8 is asymptotically realized by a sequence of graphs H_n with two components: a star tree T_n^1 with $\lfloor n/2 \rfloor$ nodes and a linear tree T_n^2 with $\lceil n/2 \rceil$ nodes. The maximal singular value of E_n coincides with the one of $E(T_n^1)$, growing as $(n/2)^{1/2}$, while the minimal nonzero singular value

of E_n coincides with the one of $E(T_n^2)$, collapsing to zero as $(n/2)^{-1}$. Therefore, the spectral condition number behaves as $n^{3/2}$.

It is even possible to construct a sequence of trees having condition number asymptotic to $n^{3/2}$, answering in the positive to the question raised in Remark 2.4. Consider a sequence of trees $\hat{H}_{n+1}$ formed by the union of T_n^1 and T_n^2 with a new node u and the two arcs that join u with the roots of T_n^1 and T_n^2. We have

$$\sigma_{\max}(\hat{E}_{n+1}) \geq \sigma_{\max}(E(T_n^1)) = \sqrt{\lfloor n/2 \rfloor - 2 + O(n^{-1})} \sim n^{1/2} \ .$$

Let the order of the nodes and the arcs of $\hat{H}_{n+1}$ be such that the first rows and columns are related to the linear tree T_n^2:

$$\hat{E}_{n+1} = \left[\begin{array}{cccc|cccc|cc} 1 & 0 & \cdots & 0 & & & & & 0 & -1 \\ -1 & 1 & \cdots & 0 & & & & & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & & & & & \vdots & \vdots \\ 0 & \cdots & -1 & 1 & & & & & 0 & 0 \\ 0 & \cdots & 0 & -1 & & & & & 0 & 0 \\ & & & & 1 & 1 & \cdots & 1 & -1 & 0 \\ & & & & -1 & 0 & \cdots & 0 & 0 & 0 \\ & & & & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ & & & & 0 & \cdots & -1 & 0 & 0 & 0 \\ & & & & 0 & \cdots & 0 & -1 & 0 & 0 \\ \hline 0 & \cdots & & 0 & 0 & \cdots & & 0 & 1 & 1 \end{array}\right] \tag{2.7}$$

or, more compactly,

$$\hat{E}_{n+1} = \left[\begin{array}{cc|cc} E(T_n^2) & \mathrm{O} & 0 & -e_1 \\ \mathrm{O} & E(T_n^1) & -e_1 & 0 \\ \hline 0^T & 0^T & 1 & 1 \end{array}\right] .$$

We have already seen in section 2.2.1 that $E(T_n^2)^T E(T_n^2) = T_{\bar{n}}$, where $\bar{n} = \lceil n/2 \rceil - 1$ and T_h is the $h \times h$ Toeplitz matrix generated by the symbol $f(x) = 2 - 2\cos(x)$, having

$$\lambda_{\min}(T_h) = 4\sin^2\left(\frac{\pi}{2(h+1)}\right) \ .$$

Now, let $w \in \mathcal{R}^{\bar{n}}$ be the eigenvector of $T_{\bar{n}}$ corresponding to the minimum eigenvalue, and let $x \in \mathcal{R}^n$ be the vector $(1/\|w\|_2)[w|0]$ obtained by padding the normalized eigenvector with $\lfloor n/2 \rfloor + 1$ zeroes. Since the $(n+1) \times n$ matrix $\hat{E}_{n+1}$ has full column rank, we have

$$\sigma_{\min}(\hat{E}_{n+1}) = \sigma_n(\hat{E}_{n+1}) = \inf_{\|y\|_2 = 1} \|\hat{E}_{n+1} y\|_2$$

$$\leq \|\hat{E}_{n+1} x\|_2 = \frac{\|E(T_n^2) w\|_2}{\|w\|_2} = \frac{\sqrt{w^T E(T_n^2)^T E(T_n^2) w}}{\|w\|_2}$$

$$= \frac{\sqrt{w^T T_{\bar{n}} w}}{\|w\|_2} = 2\sin\left(\frac{\pi}{2\lceil n/2 \rceil}\right) \sim n^{-1},$$

and therefore $\kappa_2(\hat{E}_{n+1}) \sim n^{3/2}$.

3. Weighted graph matrices. We will now use the results of the previous section to study the spectral conditioning of (sequences of) weighted graph matrices $E_n\Theta E_n^T$. Let θ be the vector containing the diagonal elements of Θ. By considering the Rayleigh quotient

$$\frac{x^T(E_n\Theta E_n^T)x}{x^T(E_nE_n^T)x} = \frac{\sum_{i=1}^m y_i^2\theta_i}{\sum_{i=1}^m y_i^2}$$

at any $x \notin \mathrm{Ker}(E_nE_n^T)$, it is easy to see that

$$\lambda_{n-1}(E_n\Theta E_n^T) \geq \theta_{\min}\lambda_{n-1}(E_nE_n^T) = \theta_{\min}\sigma_{n-1}^2(E_n), \tag{3.1}$$

$$\lambda_{\max}(E_n\Theta E_n^T) \leq \theta_{\max}\lambda_{\max}(E_nE_n^T) = \theta_{\max}\sigma_{\max}^2(E_n), \tag{3.2}$$

where $\theta_{\min}$ and $\theta_{\max}$ are, respectively, the minimum and maximum elements of θ and where $\mathrm{Ker}(X)$ denotes the null space of a square matrix X. These estimates imply that the worst-case conditioning of $E_n\Theta E_n^T$ is in the order of $(\theta_{\max}/\theta_{\min})n^3$.

Other estimates of the condition number of $E_n\Theta E_n^T$ can be obtained through the "decomposition to spanning trees" of H_n. For any subgraph T of H_n, let us denote by $\mathcal{V}(T)$ the subset of $\mathcal{V}_n$ containing the arcs of T. Since the entries of θ (the diagonal elements of Θ) are also indexed by arcs, we will denote by $\theta(T)$ $[\Theta(T)]$ the subvector of θ (submatrix of Θ) relative to the arcs in $\mathcal{V}(T)$ and by $\theta_{\max}(T)$ and $\theta_{\min}(T)$ its minimum and maximum elements, respectively. Thus, for any T

$$E_n\Theta E_n^T \geq E(T)\Theta(T)E(T)^T \geq \theta_{\min}(T)E(T)E(T)^T$$

in the sense of the partial ordering of the Hermitian matrices. Clearly, one is interested in "maximal" subgraphs T of H_n, the obvious ones being spanning trees; therefore

$$\lambda_{n-1}(E_n\Theta E_n^T) \geq \max_{T\in T(H_n)} \theta_{\min}(T)\lambda_{n-1}(E(T)E(T)^T), \tag{3.3}$$

where $T(H_n)$ is the set of the spanning trees of H_n. The bound (3.3) can be strengthened by considering any set of disjoint spanning trees, i.e., the familiy

$$D(H_n) = \{D \subseteq T(H_n) \ : \ \mathcal{V}(T_1) \cap \mathcal{V}(T_2) = \emptyset \ \ \forall T_1, T_2 \in D\} \ . \tag{3.4}$$

For any $D \in D(H_n)$, one has

$$E_n\Theta E_n^T \geq \sum_{T\in D} E(T)\Theta(T)E(T)^T \geq \sum_{T\in D} \theta_{\min}(T)E(T)E(T)^T$$

and therefore

$$\lambda_{n-1}(E_n\Theta E_n^T) \geq \max_{D\in D(H_n)} \sum_{T\in D} \theta_{\min}(T)\lambda_{n-1}(E(T)E(T)^T) \ . \tag{3.5}$$

Note that the union of the subgraphs in D need not cover all the arc set $\mathcal{V}_n$ of H_n. Actually, one may replace $T(H_n)$ in (3.4) with the set $A(H_n)$ of all *acyclic* subgraphs of H_n, allowing for more terms in the sum of (3.5). Unfortunately, all acyclic subgraphs T which are not spanning trees have $\lambda_{n-1}(E(T)E(T)^T) = 0$, so that all the corresponding terms give no contribution to the bound.

Upper bounds on the maximum eigenvalue of $E_n\Theta E_n^T$ can be obtained with similar techniques. Consider the family of disjoint acyclic subgraphs covering all $\mathcal{V}_n$

$$C(H_n) = \left\{ C \subseteq A(H_n) \ : \ \mathcal{V}(T_1) \cap \mathcal{V}(T_2) = \emptyset \ \ \forall T_1, T_2 \in C \ , \ \bigcup_{T \in C} \mathcal{V}(T) = \mathcal{V}_n \right\} .$$

Clearly, one has

$$\lambda_{\max}(E_n\Theta E_n^T) \leq \min_{C \in C(H_n)} \sum_{T \in C} \theta_{\max}(T) \lambda_{\max}(E(T)E(T)^T) \ .$$

The above estimates can be useful when designing preconditioners for the solution of (1.1) through a PCG method, as briefly discussed in the next section.

4. Conditioning and preconditioning. In this section we briefly discuss how the analysis of the previous paragraphs is related to the study of preconditioners for the solution of (1.1) through a PCG method. In the following, we will assume that one row (for each connected component of H_n) has been deleted from E_n, so that (1.1) is a full-rank system.

In the literature, tree-based preconditioners have been shown to be quite successful, in practice, for the solution of (1.1) within interior-point approaches to linear min-cost network flow problems [15, 16]. These preconditioners are chosen as the matrices $S_nS_n^T$, where $S_n = S(T)$ corresponds to some spanning tree T of H_n, usually an (approximate) maximum-weight spanning tree, the weight of arc v_i being θ_i. Besides working well in practice, this choice has a clear rationale from the analysis of interior-point methods in that, if the optimal solution of the underlying problem is unique, then the weights θ_i tend to zero on all arcs but those corresponding to the basic optimal solution [17] that form a spanning tree. However, another rationale for this choice is given by (3.3). In fact, it is well known that, in practice, having small eigenvalues is what hurts most the performance of a PCG method. Thus, spanning trees T with large $\theta_{\min}(T)$ are presumably a good choice since

$$\kappa_2((E_n(T)E_n^T(T))^+ E_n\Theta_n E_n^T) \geq \theta_{\min}(T)$$

with X^+ denoting the pseudoinverse of Moore–Penrose of a matrix X (see, e.g., [12]). Interestingly, the Kruskal algorithm that is typically used for computing the maximum-weight T also gives the tree with largest $\theta_{\min}(T)$. This may be used to provide a more sophisticated convergence analysis for these methods.

Furthermore, (3.1) and (3.2) clearly imply that, using $E_nE_n^T$ as a preconditioner for (1.1), the spectral conditioning of the preconditioned matrix is limited by $\theta_{\max}/\theta_{\min}$. If the entries of θ would belong to a bounded interval $[r, R]$, with r and R positive constants independent on n, then $E_nE_n^T$ would be an optimal preconditioner for (1.1) [3], i.e., the number of PCG iterations required to achieve any chosen accuracy would be independent on n. Actually, it can be shown that the asymptotic behavior of the spectra of the preconditioner $E_nE_n^T$ describes the asymptotic behavior of the spectra of $E_n\Theta_nE_n^T$ for any "nondegenerating" sequence of positive $m(n)$ vectors θ_n. Unfortunately, $\theta_{\max}/\theta_{\min}$ grows very fast during the iterations of the interior-point methods. However, blending a preconditioning technique using $E_nE_n^T$ and a (classical) $O(\sqrt{n})$ adaptive updating has recently led to an $O(\frac{n^3}{\log n}L)$ interior-point method for linear programming [2]. It is conceivable that similar techniques could be used to keep the quantity $\theta_{\max}/\theta_{\min}$ bounded.

However, it is still not clear how to exploit the structure of E_n in order to devise a fast algorithm for solving linear systems involving the matrix $E_n E_n^T$. In the specific case of local graphs [11], which generalize the idea of grid graphs, the self-similarity of the matrices E_n and $E_{n'}$, with $n' \sim \theta n$, $\theta \in (0,1)$ independent of n, suggests the use of an algebraic multigrid method [13] since the matrix $E_{n'}$ can be interpreted as a coarse grid version of the original matrix E_n.

Acknowledgment. We are grateful to Claudio Gentile for his contribution to a part of Theorem 2.6.

REFERENCES

[1] W.N. Anderson and T. Morley, *Eigenvalues of the Laplacian of a graph*, Linear and Multilinear Algebra, 18 (1985), pp. 141–145.

[2] K.M. Anstreicher, *Linear programming in $O(\frac{n^3}{\log n} L)$ operations*, SIAM J. Optim., 9 (1999), pp. 803–812.

[3] O. Axelsson and G. Lindskög, *The rate of convergence of the preconditioned conjugate gradient method*, Numer. Math., 52 (1986), pp. 499–523.

[4] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[5] D. Bini and M. Capovani, *Spectral and computational properties of band symmetric Toeplitz matrices*, Linear Algebra Appl., 52/53 (1983), pp. 99–125.

[6] A. Böttcher and S. Grudsky, *On the condition numbers of large semi-definite Toeplitz matrices*, Linear Algebra Appl., 279 (1998), pp. 285–301.

[7] J. Castro, *A specialized interior-point algorithm for multicommodity network flows*, SIAM J. Optim., 10 (2000), pp. 852–877.

[8] R. Courant and D. Hilbert, *Methods of Mathematical Physics* I, Interscience, New York, 1953.

[9] D. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs*, Academic Press, New York, 1979.

[10] P. Davis, *Circulant Matrices*, John Wiley and Sons, New York, 1979.

[11] A. Frangioni and S. Serra Capizzano, *Matrix-Valued Linear Positive Operators and Applications to Graph Optimization*, TR 04/99, 1999, Dip. di Informatica, Università di Pisa, Pisa, Italy.

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[13] W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, Berlin, 1985.

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

[15] M.G.C. Resende and G. Veiga, *An implementation of the dual affine scaling algorithm for minimum-cost flow on bipartite uncapacitated networks*, SIAM J. Optim., 3 (1993), pp. 516–537.

[16] L.F. Portugal, M.G.C. Resende, G. Veiga, and J.J. Júdice, *A truncated primal-infeasible dual-feasible network interior point method*, Networks, 35 (2000), pp. 91–108.

[17] T. Roos, T. Terlaky, and J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley and Sons, Chichester, UK, 1997.

[18] S. Serra, *On the extreme eigenvalues of Hermitian (block) Toeplitz matrices*, Linear Algebra Appl., 270 (1998), pp. 109–129.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 349–367
© 2001 Society for Industrial and Applied Mathematics

A STRATIFICATION OF THE SET OF NORMAL MATRICES*

MARKO HUHTANEN†

Abstract. In this paper we consider the set of normal matrices $\mathcal{N} \subset \mathbb{C}^{n\times n}$ as a stratified submanifold of $\mathbb{R}^{2n^2}$. Based on the Toeplitz decomposition, we construct a stratification of $\mathcal{N}$ with the strata of dimension $n^2 + j$ for $1 \le j \le n$. The stratum of the maximal dimension $n^2 + n$ is readily parametrizable since the Toeplitz decomposition $Z = H + iK$ of a generic $Z \in \mathcal{N}$ equals $Z = H + ip(H)$ for a polynomial p with real coefficients. Using this, it is possible to approach computational tasks involving normal matrices in a new way. To give an example, we consider the problem of approximating eigenvalues of a large, possibly sparse, normal matrix Z. In particular, we generalize the Hermitian Lanczos method to normal matrices.

Key words. normal matrix, stratified submanifold, Toeplitz decomposition, Hermitian Lanczos method

AMS subject classifications. 65F, 15A57

PII. S0895479899357085

1. Introduction.

The set of normal matrices, denoted by $\mathcal{N} \subset \mathbb{C}^{n\times n}$, is a rich class of matrices well suited for numerical computations. To give an example of the computational well behavior, extreme sensitivity of eigenvalues and eigenvectors does not occur among the set of normal matrices. The ways to characterize normality are a "rich set" as well. So far there exist about ninety equivalent conditions for a matrix to be normal, collected in [14] by Grone et al. and in [8] by Elsner and Ikramov. The classical definition of normality for a matrix $Z \in \mathbb{C}^{n\times n}$, or condition 0 as taken in [14], is based on the algebraic relation

$$ZZ^* - Z^*Z = 0 \tag{1.1}$$

for Z. Instead of considering different equivalent characterizations of normality, in this paper we study $\mathcal{N}$ as a set. More precisely, we view the set of normal matrices as a stratified submanifold of $\mathbb{R}^{2n^2}$.

It is not difficult to verify that $\mathcal{N}$ is a stratified submanifold of $\mathbb{R}^{2n^2}$. Instead, it is more difficult to construct a stratification of $\mathcal{N}$ that would be structure revealing as well as concrete enough to be useful in practical problems. Consequently, the purpose of this paper is to introduce a stratification that would, at least to some extent, have these properties. To this end we take as a starting point the Toeplitz decomposition, also called the Cartesian decomposition, of $Z \in \mathbb{C}^{n\times n}$ defined via

$$Z = H + iK, \text{ where } H = \frac{1}{2}(Z + Z^*) \text{ and } K = \frac{1}{2i}(Z - Z^*). \tag{1.2}$$

Clearly both H and K belong to the set of Hermitian matrices $\mathcal{H}$. As is well known, Z is normal if and only if H and K commute; see, e.g., condition 21 in [14]. Thus, normality of Z renders H and K in the Toeplitz decomposition strongly interdependent. Using this property, a way to achieve a simple stratification of the set of normal

*Received by the editors December 16, 1999; accepted for publication (in revised form) by A. Edelman September 4, 2000; published electronically August 8, 2001.
http://www.siam.org/journals/simax/23-2/35708.html
†Institute of Mathematics, Helsinki University of Technology, Espoo, FIN-02150, Finland (Marko.Huhtanen@hut.fi).

matrices is to first fix j, with $1 \leq j \leq n$, and constrain $H \in \mathcal{H}$ to have exactly j distinct eigenvalues. Then consider all the possible Hermitian K that commute with H. With this construction we obtain a stratification of $\mathcal{N}$ with the strata of dimension $n^2 + j$ for $1 \leq j \leq n$. Thus, the smallest dimension $n^2 + 1$ occurs when H is restricted to be a scalar multiple of the identity, i.e., $H = sI$ for $s \in \mathbb{R}$. The maximal dimension $n^2 + n$ corresponds to the case when H varies among the set of nonderogatory Hermitian matrices.

What makes the described stratification of potential use in practice is the property that we obtain a very simple parametrization for the stratum of the maximal dimension. That is, denoting by $\mathcal{H}_n$ the set of nonderogatory Hermitian matrices, the stratum of dimension $n^2 + n$ is given by the injective mapping

$$(H, \alpha_0, \ldots, \alpha_{n-1}) \to H + i \sum_{k=0}^{n-1} \alpha_k H^k \tag{1.3}$$

from $\mathcal{H}_n \times \mathbb{R}^n$ to $\mathcal{N}$. The image of this mapping is an open dense subset of $\mathcal{N}$ in the norm topology inherited from $\mathbb{R}^{2n^2}$. This is based on the fact that for a normal $Z = H + iK$ the parts H and K are simultaneously diagonalizable by a unitary matrix. Therefore a slight perturbation of the eigenvalues of H yields an element $\hat{H}$ of $\mathcal{H}_n$. Then it remains to find a polynomial p with real coefficients such that $p(\hat{H})$ is close to K. Thus, a generic normal matrix Z is of the form $Z = H + ip(H)$ for $H \in \mathcal{H}_n$ and for a polynomial p with real coefficients. In particular, the set of normal matrices can be characterized as

$$\operatorname{clos}\{H + ip(H) : H \in \mathcal{H},\ p \text{ is a polynomial with real coefficients}\}. \tag{1.4}$$

With the parametrization (1.3) of an open dense subset of $\mathcal{N}$ a number of computational problems involving normal matrices can be approached in a new way. Consider, for instance, the problem of finding a few (or all) eigenvalues and the corresponding eigenvectors of a large, possibly sparse $Z = H + iK = H + ip(H) \in \mathcal{N}$ by using Krylov subspace methods. With the Toeplitz decomposition this problem can be divided into simpler, independent parts by approximating eigenvalues of H accompanied by finding p. Once this is accomplished, approximations to eigenvalues of Z are obtained by simply applying the spectral mapping theorem with p to those eigenvalues of H that have been computed. The eigenvectors of Z are obtained directly from the corresponding eigenvectors of H. Obviously here the key is that for a Hermitian matrix, for solving some of its eigenvalues, there exists a large variety of techniques as well as a lot of different preconditioning strategies. There are several analogous routes to approximate p.

For a large, possibly sparse, Hermitian matrix the most inexpensive overall technique for finding eigenvalues is the Hermitian Lanczos method; see, e.g., [21]. It turns out that if the Hermitian Lanczos method is used to approximate eigenvalues of H for $Z = H + ip(H) \in \mathcal{N}$, then an approximation to p is obtained without *any* significant additional cost. This leads to an approximation to eigenvalues of Z as just described. We call the resulting scheme the Hermitian Lanczos method for normal matrices. As opposed to standard iterative schemes, we find this approach particularly interesting since it is *structure preserving*, that is, at every stage the approximations remain normal thereby yielding an orthonormal eigenvector basis at every step. It is well known that with the standard Arnoldi method the normality of approximations is typically

lost; see [17]. Further, the introduced algorithm preserves all the essential properties of the Hermitian Lanczos method. For instance, without reorthogonalizations, only a 3 term recurrence is needed. Consequently, with the proposed method eigenvalues of Z can be approximated by storing *only* 3 vectors.

In the decomposition $Z = H + ip(H)$ of $Z \in \mathcal{N}$ the way p can be approximated by a low degree polynomial on the spectrum of H becomes an important factor. In particular, if p happens to be of low degree, then the proposed algorithm reduces, in essence, to computing approximations to eigenvalues of H. Consequently, the significance of the behavior of p leads, in a natural way, to an introduction of an adjustable parameter yielding an analogous representation for Z. That is, consider a rotation $e^{i\theta}Z$ of Z with $\theta \in [0, 2\pi)$. Except for a finite number of θ, this rotation possesses a representation $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$ with $H_\theta \in \mathcal{H}$ and for a polynomial p_θ with real coefficients. The degree as well as the behavior of p and p_θ on the corresponding spectra can, however, be very different. Consequently, the respective approximation problems for Z and $e^{i\theta}Z$ arising from the proposed method can be completely dissimilar.

The paper is organized as follows. In section 2 we present a stratification of $\mathcal{N}$ and construct a simple parametrization for an open dense set of $\mathcal{N}$. In section 3 we outline examples of computational problems where the presented parametrization can be of use. In particular, we generalize the Hermitian Lanczos method to normal matrices such that all the essential properties of the original algorithm are preserved. Finally, we compare the introduced method with the Arnoldi method.

2. A stratification of the set of normal matrices. So far there exist about ninety equivalent conditions for a matrix to be normal, collected in [14] by Grone et al. and in [8] by Elsner and Ikramov. The standard definition of normality for $Z \in \mathbb{C}^{n\times n}$, or the so-called condition 0, deals with the equation given by the self-commutator

$$[Z, Z^*] = ZZ^* - Z^*Z = 0. \tag{2.1}$$

In particular, starting from this, algebraic as well as differential geometric interpretations come naturally. As to the algebraic geometric point of view, obviously the elements of the matrix $[Z, Z^*]$ are not, because of the complex conjugation, polynomials with respect to complex variables $\{Z_{i,j}\}_{i,j=1}^n$ denoting the elements of Z. However, they are polynomials with respect to the real and imaginary parts of $\{Z_{i,j}\}_{i,j=1}^n$, and, consequently, it is useful to regard the set of all complex $n \times n$ matrices as the real vector space $\mathbb{R}^{2n^2}$. If $Z = X + iY$, where $X, Y \in \mathbb{R}^{n\times n}$ denote the real and imaginary parts of Z, respectively, then

$$[Z, Z^*] = XX^T - X^TX + YY^T - Y^TY + i(YX^T - X^TY + Y^TX - XY^T). \tag{2.2}$$

Requiring $[Z, Z^*] = 0$, the real part gives, because of symmetry, $(n^2 + n)/2$ polynomial equations of degree 2. Analogously, the imaginary part gives, because of skew-symmetry, $(n^2 - n)/2$ polynomial equations of degree 2. These, in all n^2 homogeneous polynomials, define an affine variety in $\mathbb{R}^{2n^2}$. Obviously $\mathcal{H}$ and $\mathcal{S}$, the set of Hermitian and the set of skew-Hermitian matrices, respectively, are both n^2 dimensional subspaces of $\mathbb{R}^{2n^2}$ contained in $\mathcal{N}$, the set of normal matrices. These two subspaces are the building blocks of the constructed stratification of $\mathcal{N}$ in this paper.

A stratification Σ of a subset X of a manifold M is a partition of X into submanifolds of M, called the strata, which satisfies the local finiteness condition. That is to say, every point in X has a neighborhood in M that meets only finitely many strata. If $X \subset M$ can be stratified, X is called a stratified submanifold of M. For further definitions and properties of stratified submanifolds, see, e.g., [1] or [11].

PROPOSITION 2.1. *$\mathcal{N}$ is a connected star-shaped stratified submanifold of $\mathbb{R}^{2n^2}$.*

Proof. It is obvious that $\mathcal{N}$ is connected as each of its elements is linearly path connected to the zero matrix. By this argument $\mathcal{N}$ is obviously star-shaped. The set of normal matrices is the image of the mapping $(U, D) \to UDU^*$ (or the equivalent real mapping constructed in an obvious way). By the stratification theorem [15], $\mathcal{N}$ is a stratified submanifold of $\mathbb{R}^{2n^2}$ as this mapping is real analytic and proper (compact sets have compact preimages). ∎

As to how to actually construct a stratification of $\mathcal{N}$, there are several routes. For practical purposes we have chosen to start from the Toeplitz decomposition since it turns out that in this way we obtain a very simple parametrization for the stratum of the maximal dimension. For that purpose, let $Z = H + iK$ denote the Toeplitz decomposition of a matrix Z as defined in (1.2). We call H and iK the Hermitian and skew-Hermitian parts of Z, respectively. Let $C(Z) = \{B \in \mathbb{C}^{n\times n} : BZ = ZB\}$ denote the centralizer of Z. It is obvious that $C(Z)$ is a subspace of $\mathbb{C}^{n\times n}$.

In what follows we will either consider real or complex dimension. This will be indicated by an inclusion either in $\mathbb{R}^{2n^2}$ or in $\mathbb{C}^{n\times n}$, respectively.

LEMMA 2.2. *Assume $M \in \mathcal{H}$ and the dimension of $C(M) \subset \mathbb{C}^{n\times n}$ is l. Then the dimension of $\mathcal{S} \bigcap C(M) \subset \mathbb{R}^{2n^2}$ is l.*

Proof. It is obvious that $\mathcal{S} \bigcap C(M)$ is a subspace of $\mathbb{R}^{2n^2}$ so that the dimension is well defined. Assuming that the dimension of $C(M) \subset \mathbb{C}^{n\times n}$ is l, suppose for $S_j \in \mathcal{S} \bigcap C(M) \subset \mathbb{R}^{2n^2}$, $j = 1, \ldots, l+1$, that there do not exist $\alpha_j \in \mathbb{R}$ for $j = 1, \ldots, l+1$, of which at least one is nonzero such that $\sum_{j=1}^{l+1} \alpha_j S_j = 0$. Clearly, then there cannot exist any $\beta_j \in \mathbb{C}$, $j = 1, \ldots, l+1$, of which at least one is nonzero such that $\sum_{j=1}^{l+1} \beta_j S_j = 0$ either. This contradicts the assumption. Thus, the dimension of $\mathcal{S} \bigcap C(M) \subset \mathbb{R}^{2n^2}$ is at most l. Assume then that S_j, for $j = 1, \ldots, l-1$, is a basis of $\mathcal{S} \bigcap C(M) \subset \mathbb{R}^{2n^2}$ and $Z = H + iK \in C(M) \subset \mathbb{C}^{n\times n}$. Since $MZ - ZM = 0$, then taking the adjoint yields $Z^*M - MZ^* = 0$, so that $H, K \in C(M)$ also. Therefore, for some $\alpha_j \in \mathbb{R}$ holds $\sum_{j=1}^{l-1} \alpha_j S_j = iH$ and for some $\beta_j \in \mathbb{R}$ holds $\sum_{j=1}^{l-1} \beta_j S_j = iK$. Thus, $\sum_{j=1}^{l-1}(-i\alpha_j + \beta_j)S_j = Z$ and, consequently, S_j, for $j = 1, \ldots, l-1$, is a basis of $C(M) \subset \mathbb{C}^{n\times n}$ as well. This, however, contradicts the assumption that the dimension of $C(M) \subset \mathbb{C}^{n\times n}$ equals l and the claim follows. ∎

Let $(k_1, \ldots, k_j) \in \mathbb{N}^j$ be a partition of $n \in \mathbb{N}$, that is, $\sum_{l=1}^{j} k_j = n$. For a partition of n we denote by $\mathcal{H}(k_1, \ldots, k_j) \subset \mathcal{H}$ those Hermitian matrices $H \in \mathbb{C}^{n\times n}$ that have *exactly* j distinct eigenvalues with multiplicities $k_1, \ldots, k_j$. Obviously this is independent on the ordering of these indices. To give an example, $\mathcal{H}(1, 2, 3) \in \mathbb{C}^{6\times 6}$ is the same as $\mathcal{H}(3, 1, 2)$. For simplicity, let $\mathcal{H}_n$ denote the set of nonderogatory Hermitian matrices.

PROPOSITION 2.3. *$\mathcal{H}(k_1, \ldots, k_j) \subset \mathbb{R}^{2n^2}$ is a smooth manifold of dimension $n^2 + \sum_{l=1}^{j}(1 - k_l^2)$.*

Proof. Assume $k_1, \ldots, k_j$ are in decreasing order and let j_0 be the largest subindex for which $k_{j_0} > 1$.

The dimension of the set of unitary matrices $\mathcal{U}$ as a real smooth manifold is n^2. Fix an element $U \in \mathcal{U}$ and consider those unitary matrices V obtained via

$$V = U \begin{bmatrix} V_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & V_{j_0} & 0 \\ 0 & \dots & 0 & \Phi \end{bmatrix} \tag{2.3}$$

for $V_l \in \mathbb{C}^{k_l \times k_l}$, with $1 \leq l \leq j_0$, unitary and $\Phi = \operatorname{diag}(\theta_1, \dots, \theta_{n-p})$, where $p = \sum_{l=1}^{j_0} k_l$ and $|\theta_l| = 1$ for $1 \leq l \leq n-p$. This set is a smooth manifold of real dimension $\sum_{l=1}^{j_0} k_l^2 + (n-p)$ since the set of unitary matrices in each $\mathbb{C}^{k_l \times k_l}$ is k_l^2 dimensional and the product space of $n-p$ unit circles is an $(n-p)$ dimensional manifold.

Let X be the open set in $\mathbb{R}^j$ defined as the complement of the inverse image of 0 of the function $(\lambda_1, \dots, \lambda_j) \to \prod_{1 \leq s < t \leq j} (\lambda_s - \lambda_t)$ from $\mathbb{R}^j$ to $\mathbb{R}$. That is, X is such that $\lambda_s \neq \lambda_t$ whenever $s \neq t$. Identify X with those diagonal matrices D that have the first j_0 blocks equaling eigenvalues in each block of size k_l only, for $1 \leq l \leq j_0$, while the remaining $n-p$ eigenvalues are all pairwise different as well as different from those in the first j_0 blocks. Clearly, the set $\mathcal{H}(k_1, \dots, k_j)$ equals the image of the mapping $(U, D) \to UDU^*$ from $\mathcal{U} \times X$.

For a fixed $D \in X$ and for arbitrary $U, V \in \mathcal{U}$ there holds $UDU^* = VDV^*$ if and only if U and V are related as in (2.3). Consequently, the dimension of the image of the mapping $(U, D) \to UDU^*$ is

$$n^2 - \sum_{l=1}^{j_0} k_l^2 - (n-p) + j = n^2 - \sum_{l=1}^{j_0} k_l^2 + j_0 = n^2 + \sum_{l=1}^{j} (1 - k_l^2)$$

and the claim follows as it is obviously smooth as well. ∎

PROPOSITION 2.4. *If $H \in \mathcal{H}(k_1, \dots, k_j)$, then the dimension of $\mathcal{S} \bigcap C(H) \subset \mathbb{R}^{2n^2}$ is $\sum_{l=1}^{j} k_l^2$.*

Proof. The subindex j_0 is defined as in the proof of Proposition 2.3. Let $H = U\Lambda U^*$ be a diagonalization of H by a unitary similarity. Then $BH = HB$ is equivalent to $U^*BU\Lambda = \Lambda U^*BU$, that is, we can consider the centralizer of Λ and then use this unitary similarity to get the centralizer of H. A block, say, of size k_l has the centralizer of dimension k_l^2 as a subspace of $\mathbb{C}^{k_l \times k_l}$ simply because all the matrices of respective size commute with this block. The corresponding skew-Hermitian subspace is k_l^2 dimensional in $\mathbb{R}^{2k_l^2}$ by Lemma 2.2. Thus continuing in this manner we obtain a subspace of dimension $\sum_{l=1}^{j_0} k_l^2$. Then the remaining eigenvalues are all different, i.e., this block is nonderogatory. Thus, for this block the dimension of the centralizer is $n-p$ [16, p. 275] in $\mathbb{C}^{(n-p)\times(n-p)}$, with $p = \sum_{l=1}^{j_0} k_l$. Consequently, using again Lemma 2.2 with this block we obtain the claim after an addition. ∎

Now $\mathcal{N}$ can be stratified as follows. For $1 \leq j \leq n$, let $\mathcal{N}_j$ denote those $N = H + iK \in \mathcal{N}$ whose Hermitian part H has exactly j distinct eigenvalues (with no restrictions to multiplicities). Obviously $\mathcal{N} = \bigcup_{j=1}^{n} \mathcal{N}_j$. The following shows that this provides a stratification of $\mathcal{N}$ as well.

THEOREM 2.5. *For each $1 \leq j \leq n$ the set $\mathcal{N}_j \subset \mathbb{R}^{2n^2}$ is a smooth manifold of dimension $n^2 + j$.*

Proof. Consider $\mathcal{H}(k_1, \dots, k_j)$ and define

$$\mathcal{N}(k_1, \dots, k_j) = \{N = H + iK \in \mathcal{N} : H \in \mathcal{H}(k_1, \dots, k_j)\}. \tag{2.4}$$

Then $\mathcal{N}_k = \bigcup \mathcal{N}(k_1, \ldots, k_j)$ such that all these components are disjoint, i.e., all the partitions differ (modulo ordering). Consider now a single component $\mathcal{N}(k_1, \ldots, k_j)$. By Proposition 2.3 $\mathcal{H}(k_1, \ldots, k_j)$ is a smooth manifold of dimension $n^2 + \sum_{l=1}^{j}(1-k_l^2)$. In Proposition 2.4 we demonstrated that for a fixed $H \in \mathcal{H}(k_1, \ldots, k_j)$ the dimension of $\mathcal{S} \bigcap C(H) \subset \mathbb{R}^{2n^2}$ is $\sum_{l=1}^{j} k_l^2$. Thus $n^2 + \sum_{l=1}^{j}(1 - k_l^2) + \sum_{l=1}^{j} k_l^2 = n^2 + j$. If $H = U\Lambda U^* \in \mathcal{H}(k_1, \ldots, k_j)$, then in the proof of Proposition 2.4 we showed that the centralizer of H was the direct sum of full matrix algebras

$$U(\mathcal{M}_{k_1} \oplus \cdots \oplus \mathcal{M}_{k_{j_0}} \oplus \mathbb{C} \oplus \cdots \oplus \mathbb{C})U^*. \tag{2.5}$$

In particular, $C(H)$ is independent on the numerical values of the eigenvalues of Λ as long as they are constrained to have fixed multiplicities in this order such that $\Lambda \in \mathcal{H}(k_1, \ldots, k_j)$. Further, the computation of the skew-Hermitian part from (2.5) is a smooth operation. Thus, when U and Λ vary smoothly, the smoothness of the structure follows. ∎

The set of nonderogatory Hermitian matrices $\mathcal{H}_n$ is of particular interest for the following reason.

PROPOSITION 2.6. *Assume $H \in \mathcal{H}_n$ and p is a polynomial with real coefficients. Then $H + ip(H)$ is normal. Conversely, every normal matrix with $H \in \mathcal{H}_n$ as its Hermitian part is of this form for a polynomial p with real coefficients.*

Proof. The first claim is obvious. For the converse, let K be Hermitian. By condition 21 in [14] $H + iK$ is normal if and only if $K \in C(H)$. Since H is nonderogatory, $K = p(H)$ for a polynomial p; see, e.g., [16, p. 275]. If p is not real, then taking the real part $p_{\Re}$ of p (i.e., $p = p_{\Re} + ip_{\Im}$, where both $p_{\Re}$ and $p_{\Im}$ are polynomials with real coefficients) gives the claim. ∎

Now $\mathcal{N}_n$, that is, the set of normal matrices having a nonderogatory Hermitian part, has the following property.

THEOREM 2.7. *$\mathcal{N}_n$ is an open dense subset of $\mathcal{N}$ in the norm topology inherited from $\mathbb{R}^{2n^2}$.*

Proof. Let $\{\lambda_k(A)\}_{k=1}^{n}$ denote the eigenvalues of a $A \in \mathbb{C}^{n\times n}$, counting multiplicities, arranged in decreasing order in modulus. The function $A = H + iK \to \prod_{k=1}^{n-1}(\lambda_k(H) - \lambda_{k+1}(H))$ is continuous from $\mathbb{C}^{n\times n}$ to $\mathbb{R}$. Thereby the inverse image of 0 for this function is a closed set in $\mathbb{C}^{n\times n}$. Since we use the inherited topology, its intersection with $\mathcal{N}$, which obviously equals $\mathcal{N}\backslash\mathcal{N}_n$, is a closed set in $\mathcal{N}$.

Assume $Z \in \mathcal{N}\backslash\mathcal{N}_n$. We need to show that there is $Z_\epsilon \in \mathcal{N}_n$ arbitrarily close to Z. Let $Z = H + iK$ be the Toeplitz decomposition of Z. Assume H has j_0 eigenvalues $\lambda_1, \ldots, \lambda_{j_0}$ with multiplicities strictly larger than one. Let $\hat{\Lambda}$ denote the remaining eigenvalues. Let U be a unitary matrix diagonalizing H and K simultaneously such that the diagonal blocks corresponding to $\lambda_1, \ldots, \lambda_{j_0}$ come first. As reasoned in the proof of Proposition 2.4, then all normal matrices with H as their Hermitian part are of the form

$$U \begin{bmatrix} \Lambda_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & \Lambda_{j_0} & 0 \\ 0 & \ldots & 0 & \hat{\Lambda} \end{bmatrix} U^* + iU \begin{bmatrix} S_1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & S_{j_0} & 0 \\ 0 & \ldots & 0 & \hat{S} \end{bmatrix} U^*, \tag{2.6}$$

where $S_1, \ldots, S_{j_0}$ and $\hat{S}$ are diagonal Hermitian matrices. Now perturb each diagonal element in each $\Lambda_1, \ldots, \Lambda_{j_0}$ slightly to get a nonderogatory Hermitian matrix $U(\Lambda + \epsilon)U^*$. This resulting perturbed matrix Z_ϵ of Z remains also normal since its Hermitian

part commutes with its skew-Hermitian part and the claim follows as the perturbation can be made arbitrarily small. ∎

In particular, denoting by clos$S$ the closure of a set S, the set of normal matrices can be characterized as follows.

COROLLARY 2.8. clos$\mathcal{N}_n = \mathcal{N}$ *and, in particular,*

$$\text{clos}\{H + ip(H) : H \in \mathcal{H},\ p \text{ is a polynomial with real coefficients}\} = \mathcal{N} \tag{2.7}$$

and $\{p(H) : H \in \mathcal{H},\ p \text{ is a polynomial}\} = \mathcal{N}$.

Proof. Only $\{p(H) : H \in \mathcal{H},\ p \text{ is a polynomial}\} = \mathcal{N}$ needs to be shown. For that purpose, draw lines through each pair of eigenvalues of a normal matrix Z. Then take a rotation $e^{i\theta}Z$ of Z such that there are no vertical lines. Denote by H_θ the resulting Hermitian part of $e^{i\theta}Z$. Then construct a polynomial p_θ, via interpolation, such that $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$. From this the polynomial is readily available after multiplying both sides by $e^{-i\theta}$. ∎

An immediate question is, how about $\mathcal{L}(H)$, the set of bounded linear operators on a separable Hilbert space H? Using an adaptation of the "box construction" of the proof of the theorem of Weyl–von Neumann–Berg [6] it can be shown that (2.7) does hold. However, being quite technical, we omit it.

For $\mathcal{N}_n$ we obtain a smooth structure as well as a parametrization in a simple manner from Proposition 2.6 and Theorem 2.7.

COROLLARY 2.9. $\mathcal{N}_n \subset \mathbb{R}^{2n^2}$ *is a smooth connected manifold of dimension* n^2+n *with the parametrization*

$$(H, \alpha_0, \ldots, \alpha_{n-1}) \to H + i\sum_{k=0}^{n-1} \alpha_k H^k \tag{2.8}$$

from $\mathcal{H}_n \times \mathbb{R}^n$ *to* $\mathcal{N}_n$.

Proof. Consider the mapping (2.8). It is bijective, since H is nonderogatory and the polynomial is of degree $n-1$ at most. Also it is clearly smooth. As to the connectedness, suppose $N_1, N_2 \in \mathcal{N}_n$. Thus, $N_1 = U\Lambda_1 U^* + iUp(\Lambda_1)U^*$ and $N_2 = V\Lambda_2 V^* + iVq(\Lambda_2)V^*$ for some $U, V \in \mathcal{U}$ and some polynomials p and q with real coefficients. Since $\mathcal{U}$ is path connected (every unitary Q is of the form $Q = e^{iE}$ for a Hermitian matrix E; thus $Q_t = e^{itE}$, for $0 \le t \le 1$, connects Q to the identity matrix), V can be connected with a path to U. Since Λ_1 and Λ_2 are both sets with n distinct elements, they can be transformed smoothly to one another such that the amount of distinct points remains equal to n during the process. And finally, the coefficients of p and q can be smoothly transformed to one another. ∎

Also the stratum $\mathcal{N}_1$, which consists of matrices $sI + iK$ for $s \in \mathbb{R}$ and $K \in \mathcal{H}$ and is the stratum of the least dimension $n^2 + 1$, is connected. The manifold $\mathcal{N}(k_1, \ldots, k_j)$ in (2.4) is not connected unless $k_1 = k_2 = \cdots = k_j$ and $\sum_{l=1}^{j} k_l = n$. In case there are varying multiplicities, individual components of $\mathcal{N}(k_1, \ldots, k_j)$ differ only in the ordering of the eigenvalues of the Hermitian part with the corresponding multiplicities. The reason that they cannot be connected to one another is that when trying to move from one component to another, i.e., when trying to change the ordering of the multiplicities of the eigenvalues of $H \in \mathcal{H}(k_1, \ldots, k_j)$ on $\mathbb{R}$, some eigenvalues of H will coalesce. This in turn means that the matrix has entered into another manifold $\mathcal{N}(p_1, \ldots, p_l)$ with the indices $p_1, \ldots, p_l$ corresponding to the arisen coalescence. While clos$\mathcal{N}_n = \mathcal{N}$, the other strata have the following property when taking the closure.

PROPOSITION 2.10. *Let* $\{p_s^1\},\{p_s^2\},\ldots,\{p_s^j\}$ *be a partition of* $\{p_1,\ldots,p_l\}$ *such that* $\sum_s p_s^m \leq k_m$ *for* $1 \leq m \leq j$. *Then* $\mathcal{N}(k_1,\ldots,k_j) \subset \mathrm{clos}\mathcal{N}(p_1,\ldots,p_l)$.

Proof. Take $Z \in \mathcal{N}(k_1,\ldots,k_j)$ and assume Z has been decomposed as in (2.6). It should be obvious how an element of $\mathcal{N}(p_1,\ldots,p_l)$ close to Z is now constructed: Slightly vary each block $\Lambda_1,\ldots,\Lambda_j$ appropriately so as to get the right amount of eigenvalues with multiplicities $p_1,\ldots,p_l$. ∎

Thus, generically, one can expect a normal matrix Z to be of the form $H+ip(H)$ for a Hermitian matrix H and a polynomial p with real coefficients. As to applications, the degree of p becomes an important factor. Overstating somewhat, the smaller the degree of p, the more Hermitian Z can be regarded as far as Krylov methods are concerned. In the following section we describe problems where this approach can be useful.

3. Applications to problems involving normal matrices. With a readily available parametrization for an open dense set of $\mathcal{N}$ it is possible to solve computational problems involving normal matrices in a new way. To demonstrate this we outline an approach for two well-known examples, the eigenvalue problem and the problem of finding a closest normal approximant to a matrix $A \in \mathbb{C}^{n\times n}$.

Instead of using the Hermitian part of $Z \in \mathcal{N}$, the computations can be performed with the skew-Hermitian part of Z. That is, a generic $Z = H + iK \in \mathcal{N}$ can be represented either as $Z = H + ip(H)$ or as $Z = q(K) + iK$ for polynomials p and q with real coefficients. And more generally, for a rotated $e^{i\theta}Z$, with a $\theta \in [0, 2\pi)$, this type of representation exists except for some very exceptional values of θ. To this end, let us denote by H_θ the Hermitian part of $e^{i\theta}Z$.

PROPOSITION 3.1. *Assume* $Z \in \mathbb{C}^{n\times n}$ *is normal. Then, for* θ *belonging to an open dense subset of* $[0, 2\pi)$, *there holds* $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$ *for a polynomial* p_θ *with real coefficients.*

Proof. Draw lines through each pair of eigenvalues of Z. For each rotation for which there are no vertical lines one can construct a polynomial p_θ such that $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$ by interpolation as in the proof of Corollary 2.8. ∎

Recall that H_θ arises while approximating the field of values of a (not necessarily normal) matrix Z. Namely then, for a finite number of different θ, one computes the largest eigenvalue of H_θ and intersects certain half-planes defined on the basis of these eigenvalues; see, e.g., [16, Thms. 1.5.12, 1.5.14]. For a normal $Z \in \mathbb{C}^{n\times n}$ the rotation is of interest for computational purposes as the polynomial p_θ in the representation $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$ obviously depends on the rotation parameter θ. Let us illustrate this with a simple example.

Example 1. Assume $Z = H + iK$ is normal and $\sigma(Z)$ lies on the parabola $x = y^2$ such that $\sigma(K)$ is indefinite. If Z can be presented as $H + ip(H)$ with a polynomial with real coefficients, then the degree of p can be quite high, even n. Whereas for $e^{i\frac{\pi}{2}}Z$, the representation $e^{i\frac{\pi}{2}}Z = H_{\frac{\pi}{2}} + ip_{\frac{\pi}{2}}(H_{\frac{\pi}{2}})$ is obtained for a polynomial $p_{\frac{\pi}{2}}$ of degree 2 only.

DEFINITION 3.2. *Let* $Z \in \mathbb{C}^{n\times n}$ *and* $\theta \in [0, 2\pi)$. *Then* $Z = e^{-i\theta}H_\theta + ie^{-i\theta}K_\theta$ *is the rotated Toeplitz decomposition of* Z *by the angle* θ.

In this decomposition the parts are (typically) not Hermitian matrices. This is obviously irrelevant as all the computational aspects are analogous for H_θ and $e^{-i\theta}H_\theta$.

3.1. Correcting the Francis shifted QR for normal matrices. Let us first consider "small problems," that is, problems in which all the dense matrix manipulations are feasible.

Although in practice the Francis shifted QR is very reliable for finding eigenvalues, there are, even among $\mathcal{N}$, matrices for which the convergence can fail; see [4, 5] by Batterson and [7, p. 173] by Demmel. The simplest example from [5] is the following.

Example 2. The Francis shifted QR for the unitary matrix

$$Z = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.1}$$

fails to converge as the "decoupling" into smaller matrices does not take place.

Fortunately, roundoff typically results in convergence; for an example, see [5].

With the representation of Proposition 3.1 we can introduce a simple trick to make QR convergence for normal matrices. Namely, for Hermitian matrices the Francis shifted QR converges globally, by considering, if necessary, the symmetric matrix

$$\begin{bmatrix} B & -C \\ C & B \end{bmatrix}$$

obtained from a Hermitian $H = B + iC$, where C and B are the real and imaginary parts of H. For symmetric matrices the global convergence has been demonstrated in [5]. This result can be combined with a representation $e^{i\theta}Z = H_\theta + ip_\theta(H_\theta)$ for $Z \in \mathcal{N}$ as follows. For that purpose define, as usual, the inner-product among $\mathbb{C}^{n\times n}$ via setting

$$(A, B) := \mathrm{Trace}(B^*A). \tag{3.2}$$

Then solve, using the induced norm,

$$\min_{p\in\mathcal{P}} \|K_\theta - p(H_\theta)\| \tag{3.3}$$

for a value of $\theta \in [0, 2\pi)$ that yields zero. Thereafter the spectrum of Z is obtained by applying the spectral mapping theorem with H_θ.

Algorithm 1 (for every normal matrix converging QR eigenvalue algorithm).

Step 1. *For* $\theta \in [0, 2\pi)$ *compute the eigenvalues of* H_θ *with the Francis shifted QR.*

Step 2. *Compute* p *realizing* $\min_{p\in\mathcal{P}} \|K_\theta - p(H_\theta)\|$.

Step 3. *If the minimum is zero, use the spectral mapping theorem with* $q(\lambda) = e^{-i\theta}(\lambda + ip(\lambda))$ *to find the spectrum of* Z. *If the minimum is not zero, go to Step* 1.

A way to realize the second step is the following. We do not claim it to be optimal and we describe it *merely* because it is the closest approach to sparse methods which is a primary reason, aside from a simple stratification of $\mathcal{N}$, for the introduction of a representation $Z = e^{-i\theta}H_\theta + ie^{-i\theta}p_\theta(H_\theta)$. Namely, consider the linear mapping in $\mathbb{C}^{n\times n}$ defined via $A \to H_\theta A$. Now compute the Hessenberg representation for this starting from the identity matrix. We thus obtain a Krylov subspace

$$\mathcal{K}(H_\theta; I) = \mathrm{span}\{I, H_\theta, H_\theta^2, \ldots, H_\theta^{n-1}\}. \tag{3.4}$$

Orthogonalizing these matrices with the Gram–Schmidt process yields a Hessenberg representation for the mapping $A \to H_\theta A$ restricted to the Krylov subspace (3.4). Then, after expanding K_θ in this basis, it is straightforward to compute the polynomial p_θ by using the elements of the Hessenberg representation; see Algorithm 2 below.

Example 2 (continued). We execute Algorithm 1 with the matrix A in (3.1). It is natural to start with $\theta = 0$. This is a "rare" value of θ as one needs to choose another θ to make the minimizing problem of Step 2 equal zero. We thus return to Step 1 and pick $e^{i\pi/8}$, for instance. With this value zero is attained at Step 2 so that the convergence takes place.

3.2. A Hermitian Lanczos method for normal matrices. Consider the problem of computing an approximation to some eigenvalues of a *large*, possibly sparse, normal matrix Z. It is reasonable to assume Z to be generic in the sense that $Z = H + iK = H + ip(H)$ for $H \in \mathcal{H}$ and for a polynomial p with real coefficients. Or, based on some a priori information, a rotated Toeplitz decomposition of Z can be taken as a starting point so as to have $Z = e^{-i\theta}H_\theta + ie^{-i\theta}p_\theta(H_\theta)$. Below we will discuss alternatives for choosing a rotation parameter θ for this purpose.

Whenever $Z = H + ip(H)$, it is apparent that H is readily computable whereas p is not available. Proceeding with H and finding an eigenvector, say, x, of H yields an eigenvalue of Z related to z after evaluating Zx. This is clearly not a very practical approach as it gives eigenvalues of Z almost randomly. To avoid this, some information about p is needed and, consequently, p has to be approximated in some manner. A way to achieve this is to use the Hermitian Lanczos method [21] with H. Namely then, aside from an approximation to the spectrum of H, also an approximation to p is obtained. In particular, this is achieved without any relevant additional computational expenses. The derivation of the scheme is as follows.

The well-known classical Arnoldi method [2] for the eigenvalue approximation starts from the construction of a Krylov subspace with $A \in \mathbb{C}^{n\times n}$ and a vector $b \in \mathbb{C}^n$. Then a monic polynomial $\hat{p}_j$ of degree j is computed with the property that $\|\hat{p}_j(A)b\|$ is minimized over all monic polynomials of degree j, that is,

$$\|\hat{p}_j(A)b\| = \min_{\gamma_1,\ldots,\gamma_j\in\mathbb{C}} \left\| \left(A^j - \sum_{i=1}^{j} \gamma_i A^{j-i} \right) b \right\|. \tag{3.5}$$

The roots of the polynomial $\hat{p}_j$ are then taken as approximate eigenvalues of A. For more information on the Arnoldi method for the eigenvalue problems, see, e.g., [21]. An eigenvalue approximation for $Z = H + ip(H)$ involving the Hermitian part H of Z can be obtained as follows. Compute polynomials $\hat{p}_j$ and p_j via

$$\|\hat{p}_j(H)b\| = \min_{\gamma_1,\ldots,\gamma_j\in\mathbb{C}} \left\| \left(H^j - \sum_{i=1}^{j} \gamma_i H^{j-i} \right) b \right\| \tag{3.6}$$

and

$$\|Kb - p_j(H)b\| = \min_{\gamma_1,\ldots,\gamma_j\in\mathbb{R}} \left\| \left(K - \sum_{i=1}^{j} \gamma_i H^{j-i} \right) b \right\|. \tag{3.7}$$

That is, $\hat{p}_j$ and p_j are constructed to approximate the eigenvalues of H and the polynomial p, respectively. Then an eigenvalue approximation for Z is obtained by applying the spectral mapping theorem with the polynomial $z + ip_j(z)$ to the eigenvalue approximation obtained with $\hat{p}_j$ for H in (3.6). Obviously (3.7) can be replaced with the approximation

$$\|Zb - p_j(H)b\| = \min_{\gamma_1,\ldots,\gamma_j\in\mathbb{C}} \left\| \left(Z - \sum_{i=1}^{j} \gamma_i H^{j-i} \right) b \right\| \tag{3.8}$$

instead.

Finding the roots of $\hat{p}_j$ and the computation of p_j can be realized as follows. For the roots of $\hat{p}_j$ the Hermitian Lanczos method is employed by computing

$$T_j := Q_j^* H Q_j = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ 0 & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{j-1} & \beta_{j-1} \\ & & & \beta_{j-1} & \alpha_j \end{bmatrix} \tag{3.9}$$

with the Hermitian $H = \frac{1}{2}(Z + Z^*)$ using an initial vector $q_0 = \frac{b}{\|b\|} \in \mathbb{C}^n$. The matrix $Q_j \in \mathbb{C}^{n\times j}$ has orthonormal columns spanning the Krylov subspace $\mathcal{K}_j(H; q_0) = \text{span}\{q_0, Hq_0, H^2q_0, \ldots, H^{j-1}q_0\}$. As is well known, the eigenvalues of (3.9) yield the roots of $\hat{p}_j$; see, e.g., [21]. For finding p_j one needs to compute the Fourier coefficients of Kb in the constructed basis of $\mathcal{K}_j(H; q_0)$ and collect the coefficient accordingly from the elements of (3.9).

Denoting by $\sigma(A)$ the spectrum of a matrix $A \in \mathbb{C}^{n\times n}$, we obtain the following algorithm.

ALGORITHM 2 (a Hermitian Lanczos method for normal matrices).
for $Z = H + iK \in \mathbb{C}^{n\times n}$ and a vector q_0 of unit length set $\beta_{-1} = 0$, $p(\lambda) = (Kq_0, q_0)$, $p_{-1}(\lambda) = 0$ and $p_0(\lambda) = 1$.
for $j = 1$ to k
  compute with H using the Hermitian Lanczos method α_j and β_j and q_j
  compute $p_j(\lambda) = \frac{1}{\beta_j}\lambda p_{j-1}(\lambda) - \frac{\alpha_j}{\beta_j}p_{j-1}(\lambda) - \frac{\beta_{j-1}}{\beta_j}p_{j-2}(\lambda)$
  compute $\gamma_j = (Kq_0, q_j)$
  set $p(\lambda) = p(\lambda) + \gamma_j p_j(\lambda)$
  compute $\sigma(T_j) + ip(\sigma(T_j))$
end for

Remark 1. This is a structure preserving Krylov subspace method. Namely, at each step the approximant for Z is $T_j + ip(T_j)$ restricted to

$$\mathcal{K}_j(H; q_0) = \text{span}\{q_0, Hq_0, \ldots, H^{j-1}q_0\}$$

which is *normal*. Consequently, all the approximative eigenvectors can be orthogonalized against each other to give an orthonormal basis. We stress that the standard Arnoldi method does not preserve normality [17].

Remark 2. For computing eigenvalue approximations (in exact arithmetic) only 3 vectors need to be stored as both finding the elements of T_j and updating the polynomial involves a single 3 term recurrence. Note that the cost resulting from updating the polynomial p as well as the additional memory requirements are negligible. These are very attractive properties. Of course, as with the Hermitian Lanczos method, finite precision means that rounding errors can contaminate the results after a number of steps. One alternative to try to avoid this is to use full reorthogonalization which means saving all the q_k's, or something in between, that is, using selective orthogonalization. See [7] for a thorough discussion of this subject for the Hermitian Lanczos method.

Remark 3. This approach to computing eigenvalues of a normal Z is directly related to the polynomial p in representation $Z = H + ip(H)$. If p is of low degree, then finding eigenvalues of Z reduces essentially to that of finding eigenvalues of H. For instance, if the degree of p is two so that $\sigma(Z)$ lies on a parabola $y = ax^2 + bx + c$

with $a, b, c \in \mathbb{R}$, then for $j \geq 3$ *all* the approximative eigenvalues will lie on this parabola. This is obviously a very attractive feature and completely analogous with the property that for a Hermitian matrix the approximants converge on the real axis. The standard Arnoldi method will not "detect" the parabola before the nth step.

Remark 4. Computing approximations with Algorithm 2 allows one to use the Kaniel–Paige convergence theory [12] with normal matrices in the simplest form as follows. If $Z = H + ip(H)$ with a polynomial p of degree K, then, as soon as $j \geq K + 1$, Algorithm 2 has detected p. At the same time, the extreme eigenvalues of H are bounded according to the Kaniel–Paige theory. Thus, combining these bounds with knowing p makes it completely straightforward to bound the error of the approximations of Algorithm 2 for the right- and leftmost eigenvalues of Z.

Remark 5. A more realistic situation is that, instead of being a low degree polynomial, p is *well approximated* by a reasonably low degree polynomial on the spectrum of H. If this is the case, then after the corresponding number of steps p is well approximated with Algorithm 2. We will demonstrate with an example that the convergence behavior of Algorithm 2 can be expected to be very good then.

Remark 6. Finding a good rotation parameter $\theta \in [0, 2\pi)$ for a rotated Toeplitz decomposition can make a big difference as demonstrated in Example 1. A way to find one is to test with a few rotations, with a small k, how the corresponding minimization problem (3.7) (or (3.8)) does behave. Based on this, the rotation giving the smallest value will then be chosen. For this purpose, see Algorithm 3 below.

Remark 7. Since the rotation affects the convergence, it is possible to try to benefit fully from it. Namely, instead of trying to find an optimal θ as suggested in Remark 6, computing approximations with different θ can yield a good overall picture of $\sigma(Z)$.

Remark 8. As opposed to the power method, for obvious reasons we are tempted to call this method a *fractional power* Krylov subspace method for Z. Namely, if $Z = H + ip(H)$ for a polynomial p of degree $l \geq 1$, then the fractional power can be considered to be $1/l$.

Remark 9. Note that all the minimization problems giving rise to this algorithm involve *commuting* normal matrices so that max-min property holds [13]. Thus, in this respect the approximation problems (3.6) and (3.8) are equivalent.

Remark 10. One can easily give examples where Algorithm 2 beats the Arnoldi method when measured in the number of steps taken. And this is even achieved with the *fixed* amount of storage of only 3 vectors. Thus, it is a very good question to ask how this can be possible at all. An answer is that Algorithm 2 is tuned to find out about the structure of normality of Z, that is, of the polynomial p, by using the information provided by the Krylov subspaces $\mathcal{K}_j(H; q_0)$ in a twofold way. This is not possible with the standard Arnoldi method as, being a more general method, it does not exploit normality in any reasonable way.

If certain interior eigenvalues of Z are being computed, then the computation should be divided into two parts. First, one needs to find an approximation to p. One way to achieve this is to use Algorithm 2. With H it is possible to use preconditioning techniques in order to locate its interior eigenvalues. Clearly these tasks can be performed completely independently and in parallel. Thereafter using the spectral mapping theorem in an obvious manner yields an eigen-approximation.

For finding a good "local" approximation to p, that is, an approximation that is meant to be particularly accurate over a certain part of $\sigma(H)$, it is possible to use preconditioning. The most elementary approach is to use an inverse iteration type of

algorithms with a translated Z. In this approach a priori knowledge of the location of searched eigenvalues should be available. We do not consider preconditioning in this paper.

For choosing an appropriate $\theta \in [0, 2\pi)$ for Algorithm 2, the following algorithm can be used. It simply monitors (3.7) up to $k_{\max}$, stopping earlier if a given tolerance is attained. Note that the elements of (3.9) need not be saved (except for those that are used for computing the latest q_j, of course).

ALGORITHM 3 (monitoring the rotation parameter θ).
choose θ and a vector q_0 of unit length and set $e^{i\theta}Z = H_\theta + iK_\theta$ and $q = K_\theta q_0$
for $j = 1$ to $k_{\max}$
compute with H_θ using the Hermitian Lanczos q_j and $q = q - (q, q_j)q_j$
if $\|q\| < tol$, break, end
end for

Let us now consider examples. Regarding the proposed method for normal matrices an overall method we compare it with the Arnoldi method. We do not reorthogonalize with Algorithm 2. In some respects it is hard to make the comparison fair. For instance, Algorithm 2, with no reorthogonalization, consumes only a fixed amount of memory. Thus, from this point of view it is not clear whether we should run the methods equally "far." All the computations are performed with `matlab` [19]. In all the examples the initial vector is randomly generated.

Example 3. It is easy to construct examples in which the overall convergence of the Arnoldi method is almost disastrous compared with Algorithm 2. A way to achieve this is to consider examples in which the spectrum Z lies on a polynomial curve of low degree making "violent curves." Namely then, the Arnoldi method places approximations between curves while Algorithm 2 is very quickly forced to follow the polynomial form of the spectrum. For an example we take H to be a 200×200 diagonal matrix with the eigenvalues chosen randomly from a normal Gaussian distribution. We form $Z = H + ip(H)$, where $p(x) = 2x^4 - 35x^2 - 2$. We compute 10 and 19 steps using the Arnoldi method as well as Algorithm 2. The approximative eigenvalues are depicted in Figures 3.1 and 3.2.

Example 4. We take Z to be a direct sum $Z_1 \oplus Z_2$ of size 230×230 with the following diagonal matrices. $Z_1 = \frac{1}{2}A_1 + 5iA_2 \in \mathbb{C}^{30\times30}$, where A_1 and A_2 have diagonal elements chosen randomly from a normal Gaussian distribution. $Z_2 = H + iq(H)$ is generated in exactly the same way as in Example 3 except that $q(x) = -4x^4+18x^3+50x^2+30$. Thus, Z is normal but the polynomial p in the representation $Z = H + ip(H)$ is *not* of low degree as there is a "perturbation" Z_1. Despite the perturbation, Algorithm 2 finds the essential polynomial form of the spectrum very well. The approximative eigenvalues are depicted in Figures 3.3 and 3.4.

Example 5. We let H be a 200×200 diagonal matrix with eigenvalues chosen randomly from a normal Gaussian distribution. We form $Z = H + ip(H)$, where $p(x) = x^2$. Then we rotate Z by angles $e^{i\pi/8}$, $e^{i\pi/4}$, and $e^{i\pi/2}$. The approximations are in Figures 3.5–3.9. Recall that the Arnoldi approximants are rotation invariant so that we have depicted its approximations only once. Note that Algorithm 2 does fine at those parts of the spectrum that can be interpolated well with a low degree polynomial. As a result, the eigenvalue approximation differ correspondingly.

3.3. Closeness problems. Inexpensive approximative solutions to departure from normality in the sense of Henrici have been derived by Lee [18]. Another way to measure nonnormality is to find a closest normal matrix to $A \in \mathbb{C}^{n\times n}$. In the Frobenius norm this was solved by Gabriel [10] and Ruhe [20]. However, a computation of

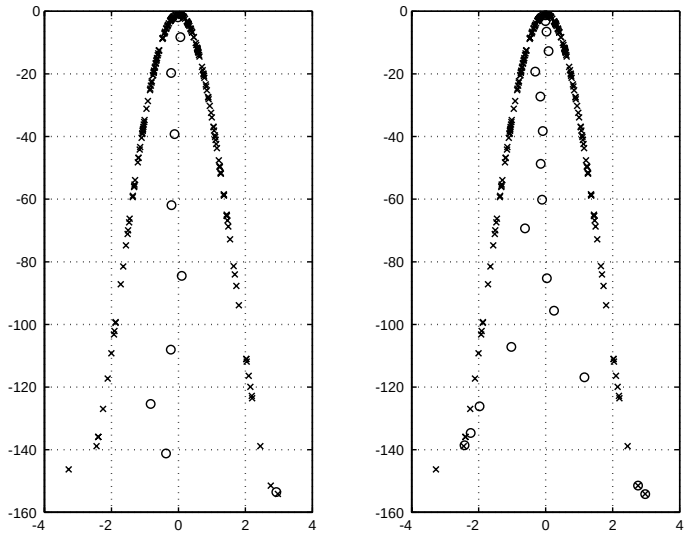


FIG. 3.1. *For Example* 3 *the Arnoldi approximants for* $H + ip(H)$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = 2x^4 - 35x^2 - 2$ *with* $j = 10$ *and* 19. x*'s denote the exact eigenvalues and* o*'s are approximations.*

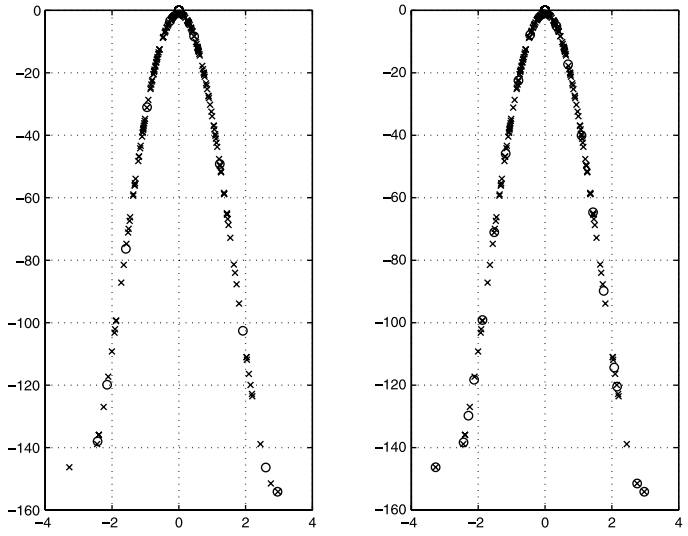


FIG. 3.2. *For Example* 3 *Algorithm* 2 *approximants for* $H + ip(H)$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = 2x^4 - 35x^2 - 2$ *with* $j = 10$ *and* 19. x*'s denote the exact eigenvalues and* o*'s are approximations.*

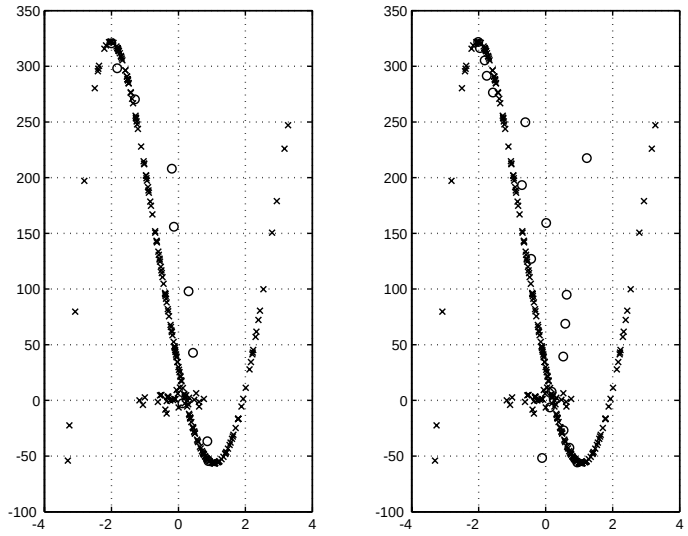


FIG. 3.3. *For Example* 4 *the Arnoldi approximants for a matrix of size* 230×230 *with* $j = 10$ *and* 19. x*'s denote the exact eigenvalues and* o*'s are approximations.*

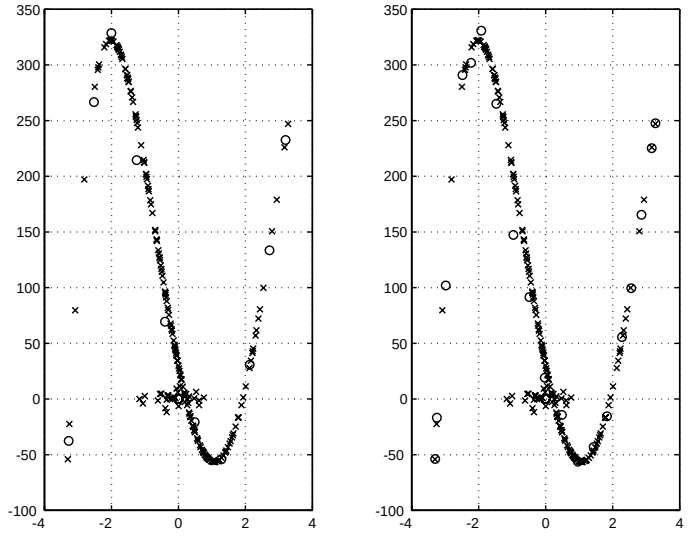


FIG. 3.4. *For Example* 4 *Algorithm* 2 *approximants for a matrix of size* 230×230 *with* $j = 10$ *and* 19. x*'s denote the exact eigenvalues and* o*'s are approximations.*

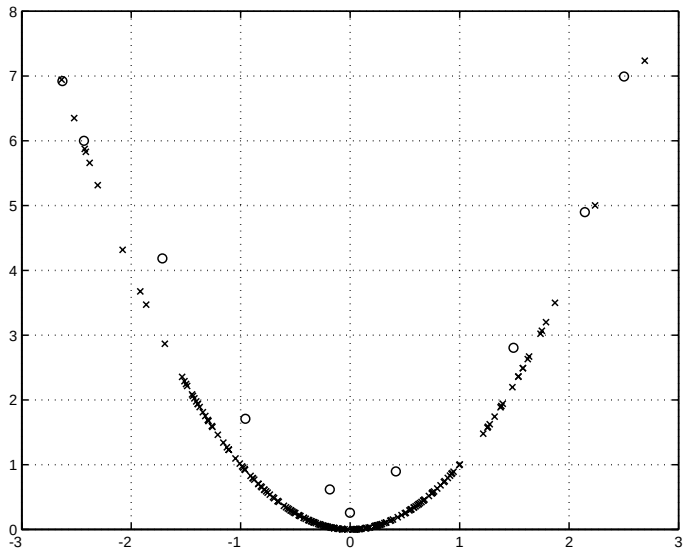


FIG. 3.5. *For Example* 5 *the Arnoldi approximants for* $H + ip(H)$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = x^2$ *with* $j = 10$. x*'s denote the exact eigenvalues and* o*'s are approximations.*

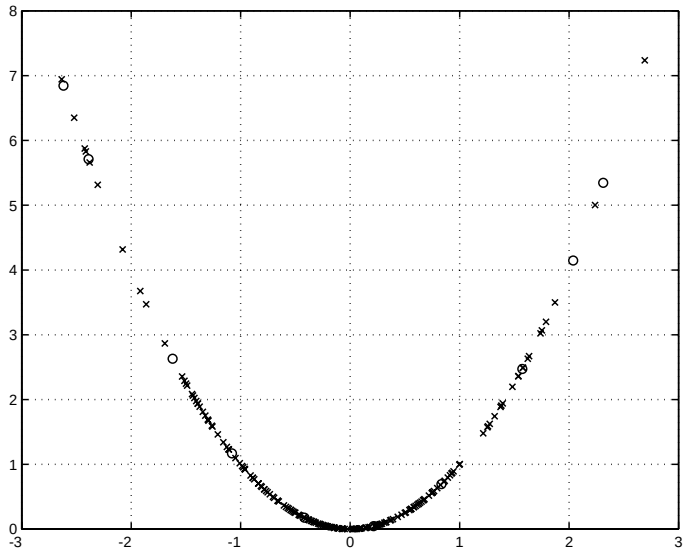


FIG. 3.6. *For Example* 5 *Algorithm* 2 *approximants for* $H + ip(H)$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = x^2$ *with* $j = 10$. x*'s denote the exact eigenvalues and* o*'s are approximations.*

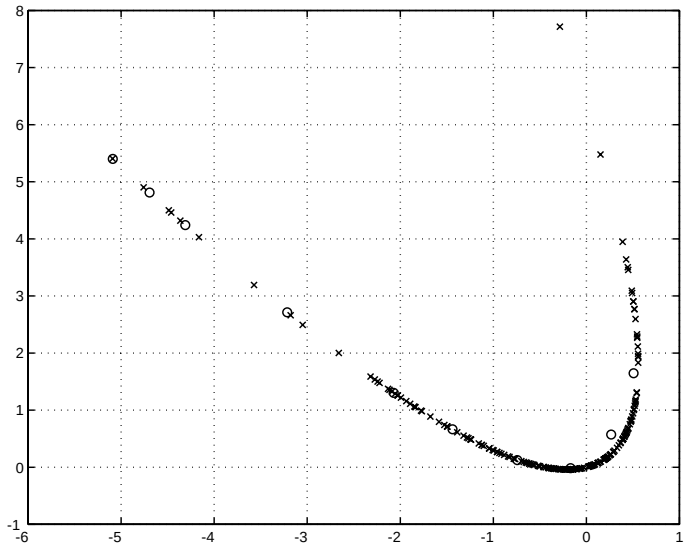


FIG. 3.7. *For Example* 5 *Algorithm* 2 *approximants for* $e^{i\pi/8}(H + ip(H))$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = x^2$ *with* $j = 10$. x*'s denote the exact eigenvalues and* o*'s are approximations.*

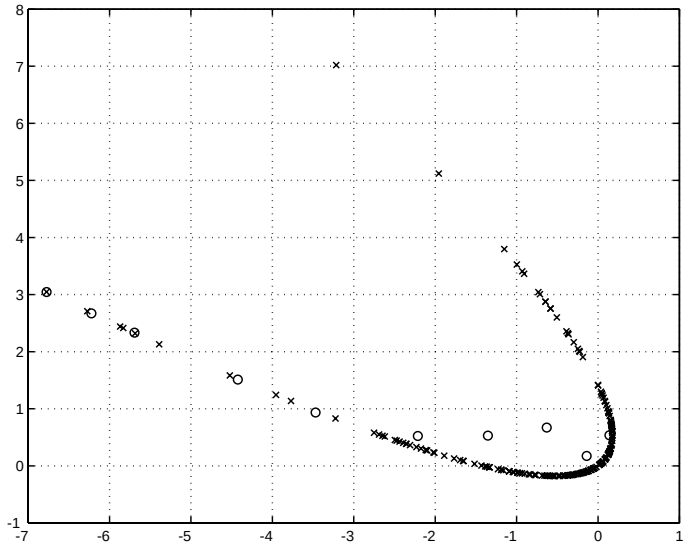


FIG. 3.8. *For Example* 5 *Algorithm* 2 *approximants for* $e^{i\pi/4}(H + ip(H))$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = x^2$ *with* $j = 10$. x*'s denote the exact eigenvalues and* o*'s are approximations.*

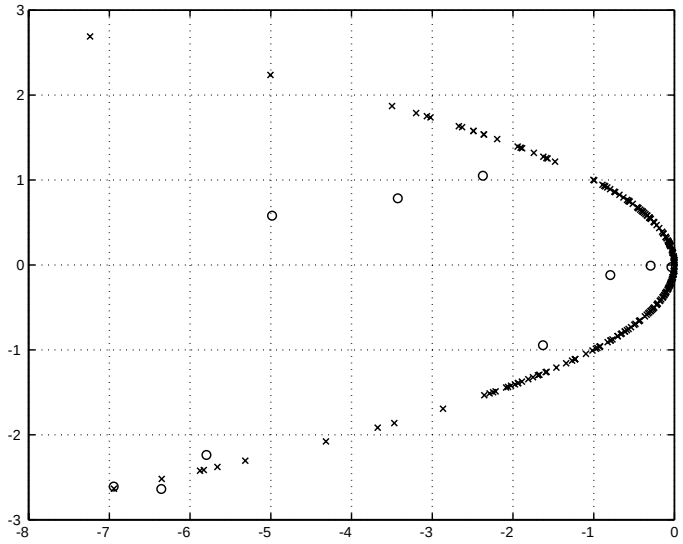


FIG. 3.9. *For Example* 5 *Algorithm* 2 *approximants for* $e^{i\pi/2}(H + ip(H))$ *with* H *a random diagonal Hermitian matrix of size* 200×200 *and* $p(x) = x^2$ *with* $j = 10$. x*'s denote the exact eigenvalues and* o*'s are approximations.*

an approximation is fairly expensive. As we have demonstrated, this problem can be stated as

$$\inf_{H\in\mathcal{H},\alpha_0,\ldots,\alpha_{n-1}\in\mathbb{R}} \left\| A - H - i\sum_{j=0}^{n-1} \alpha_j H^j \right\|_{\mathcal{F}} \tag{3.10}$$

or as

$$\min_{H\in\mathcal{H},\alpha_0,\ldots,\alpha_{n-1}\in\mathbb{C}} \left\| A - \sum_{j=0}^{n-1} \alpha_j H^j \right\|_{\mathcal{F}} \tag{3.11}$$

instead of introducing a minimization problem with constraints. In particular, approaching the problem with the parametrization (2.8) allows us to derive inexpensive approximative solutions to the problem of finding a closest normal matrix to A. One such solution is obtained by making an "initial guess" $H \in \mathcal{H}$ and then applying an Arnoldi type of iteration, that is, using matrix-vector products only. More precisely, one looks for

$$\min_{\alpha_1,\ldots,\alpha_k\in\mathbb{C}} \left\| \left(A - \sum_{i=1}^{k} \alpha_i H^{k-i} \right) b \right\| \tag{3.12}$$

with a vector $b \in \mathbb{C}^n$. Thus, generated approximations are of the form $p(H)$ for polynomials p of degree k at most. The algorithm for this purpose is easily adapted from Algorithm 2 and Algorithm 3 so that upper bounds for (3.12) are obtained by using very inexpensive computations as well as fixed storage.

An interesting problem is how to choose an H to start with. After choosing an H the computation of a normal approximant with (3.12) is relatively inexpensive

and, consequently, testing with a number of different initial guesses becomes feasible. Again, "natural" choices are, perhaps, H_θ from rotated Toeplitz decompositions for A with a few values of $\theta \in [0, 2\pi)$.

4. Conclusions. In this paper we have presented a stratification of the set of normal matrices. The stratification is constructed in such a way that the parametrization for the stratum having maximal dimension is readily available. The parametrization is simple enough to be of interest also in computational problems involving normal matrices. In particular, we have described how it is possible to approximate eigenvalues of a generic normal matrix Z by solving two approximation problems in a Krylov subspace for the Hermitian part of Z. The resulting algorithm for sparse matrices generalizes the Hermitian Lanczos method to apply to normal matrices preserving all the essential properties of the original Hermitian Lanczos method. The approximations are particularly good if, after a possible rotation, a low degree polynomial approximates the form of the spectrum of Z well.

Acknowledgment. We would like to thank Professor Alan Edelman for suggestions that improved and simplified significantly an earlier version of this paper.

REFERENCES

[1] V. I. Arnold, S. M. Gusein-Zade, and A. N. Varschenko, *Singularities of Differentiable Maps, Vol.* I, Monogr. Math. 82, Birkhäuser, Boston, 1985.
[2] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
[3] B. Aupetit, *A Primer on Spectral Theory*, Springer-Verlag, New York, 1991.
[4] S. Batterson, *Convergence of the shifted QR algorithm on* 3×3 *normal matrices*, Numer. Math., 58 (1990), pp. 341–352.
[5] S. Batterson, *Convergence of the Francis shifted QR algorithm on normal matrices*, Linear Algebra Appl., 207 (1994), pp. 181–195.
[6] I. D. Berg, *An extension of Weyl-von Neumann theorem normal operators*, Trans. Amer. Math. Soc., 160 (1971), pp. 365–371.
[7] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[8] L. Elsner and Kh. D. Ikramov, *Normal matrices: An update*, Linear Algebra Appl., 285 (1998), pp. 291–303.
[9] L. Elsner and M. H. C. Paardekooper, *On measures of nonnormality of matrices*, Linear Algebra Appl., 92 (1987), pp. 107–124.
[10] R. Gabriel, *The normal* $\triangle H$*-Matrices with connection to some Jacobi-like methods*, Linear Algebra Appl., 91 (1987), pp. 181–194.
[11] C. G. Gibson, K. Wirthmüller, A. A. Du Plessis, and E. J. N. Looijenga, *Topological Stability of Smooth Mappings*, Springer-Verlag, New York, 1975.
[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[13] A. Greenbaum and L. Gurvits, *Max-min properties of matrix factor norms*, SIAM J. Sci. Comput., 15 (1994), pp. 348–358.
[14] R. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, *Normal matrices*, Linear Algebra Appl., 87 (1987), pp. 213–225.
[15] R. Hardt, *Topological properties of subanalytic sets*, Trans. Amer. Math. Soc., 211 (1975), pp. 57–70.
[16] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[17] T. Huckle, *The Arnoldi method for normal matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 479–489.
[18] S. L. Lee, *Best available bounds for departure from normality*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 984–991.
[19] Mathworks, *Matlab*, Mathworks, Natick, MA, www.mathworks.com/products/matlab.
[20] A. Ruhe, *Closest normal matrix finally found!*, BIT, 27 (1987), pp. 585–598.
[21] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York, 1992.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 368–386
© 2001 Society for Industrial and Applied Mathematics

TWICE DIFFERENTIABLE SPECTRAL FUNCTIONS*

ADRIAN S. LEWIS† AND HRISTO S. SENDOV†

Abstract. A function F on the space of $n \times n$ real symmetric matrices is called *spectral* if it depends only on the eigenvalues of its argument. Spectral functions are just symmetric functions of the eigenvalues. We show that a spectral function is twice (continuously) differentiable at a matrix if and only if the corresponding symmetric function is twice (continuously) differentiable at the vector of eigenvalues. We give a concise and usable formula for the Hessian.

Key words. spectral function, twice differentiable, eigenvalue optimization, semidefinite program, symmetric function, perturbation theory

AMS subject classifications. 47A55, 15A18, 90C22

PII. S089547980036838X

1. Introduction.

In this paper we are interested in functions F of a symmetric matrix argument that are invariant under orthogonal similarity transformations:

$$F(U^T A U) = F(A) \qquad \text{for all orthogonal } U \text{ and symmetric } A.$$

Every such function can be decomposed as $F(A) = (f \circ \lambda)(A)$, where λ is the map that gives the eigenvalues of the matrix A and f is a symmetric function. (See the next section for more details.) We call such functions F *spectral functions* (or just functions of eigenvalues) because they depend only on the spectrum of the operator A. Classical interest in such functions arose from their important role in quantum mechanics [7, 20]. Nowadays they are an inseparable part of optimization [11] and matrix analysis [4, 5]. In modern optimization the key example is "semidefinite programming," where one encounters problems involving spectral functions like $\log\det(A)$, the largest eigenvalue of A, or the constraint that A must be positive definite.

There are many examples where a property of the spectral function F is actually equivalent to the corresponding property of the underlying symmetric function f. Among them are first-order differentiability [9], convexity [8], generalized first-order differentiability [9, 10], analyticity [26], and various second-order properties [25, 24, 23]. It is also worth mentioning the "Chevalley restriction theorem," which in this context identifies spectral functions that are polynomials with symmetric polynomials of the eigenvalues. Second-order properties of matrix functions are of great interest for optimization because the application of Newton's method, interior point methods [13], or second-order nonsmooth optimality conditions [19] requires that we know the second-order behavior of the functions involved in the mathematical model.

The standard reference for the behavior of the eigenvalues of a matrix subject to perturbations in a particular *direction* is [6]. Second-order properties of eigenvalue functions in a particular direction are derived in [25].

The problem that interests us in this paper is that of when a spectral function is twice differentiable (as a function of the matrix itself, rather than in a particular

*Received by the editors March 1, 2000; accepted for publication (in revised form) by M. Overton October 6, 2000; published electronically August 8, 2001. The research of the authors was supported by NSERC.

http://www.siam.org/journals/simax/23-2/36838.html

†Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (aslewis@math.uwaterloo.ca, hssendov@barrow.uwaterloo.ca).

direction) and when its Hessian is continuous. Analyticity is discussed in [26]: thus our result lies in some sense between the results in [9] and [26]. Smoothness properties of some special spectral functions (such as the largest eigenvalue) on certain manifolds are helpful in perturbation theory and Newton-type methods; see, for example, [15, 16, 18, 17, 22, 21, 14].

We show that a spectral function is twice (continuously) differentiable at a matrix if and only if the corresponding symmetric function is twice (continuously) differentiable at the vector of eigenvalues. Thus, in particular, a spectral function is $\mathcal{C}^2$ if and only if its restriction to the subspace of diagonal matrices is $\mathcal{C}^2$. For example, the Schatten p-norm of a symmetric matrix is the pth root of the function $\sum_i |\lambda_i|^p$ (where the λ_i's are the eigenvalues of the matrix). We see that this latter function is $\mathcal{C}^2$ for $p \geq 2$, although it is not analytic unless p is an even integer.

As part of our general result, we also give a concise and easy-to-use formula for the Hessian: the results in [26], for analytic functions, are rather implicit. The paper is self-contained and the results are derived essentially from scratch, making no use of complex-variable techniques as in [2], for example. In a parallel paper [12] we give an analogous characterization of those spectral functions that have a quadratic expansion at a point (but that may not be twice differentiable).

2. Notation and preliminary results. In what follows, S^n will denote the Euclidean space of all $n \times n$ symmetric matrices with inner product $\langle A, B\rangle = \operatorname{tr}(AB)$ and for $A \in S^n$, $\lambda(A) = (\lambda_1(A), \ldots, \lambda_n(A))$ will be the vector of its eigenvalues ordered in nonincreasing order. By O^n we will denote the set of all $n \times n$ orthogonal matrices. For any vector x in $\mathbb{R}^n$, $\operatorname{Diag} x$ will denote the diagonal matrix with the vector x on the main diagonal, and $\bar{x}$ will denote the vector with the same entries as x ordered in nonincreasing order, that is, $\bar{x}_1 \geq \bar{x}_2 \geq \cdots \geq \bar{x}_n$. Let $\mathbb{R}^n_\downarrow$ denote the set of all vectors x in $\mathbb{R}^n$ such that $x_1 \geq x_2 \geq \cdots \geq x_n$. Let also the operator $\operatorname{diag}\colon S^n \to \mathbb{R}^n$ be defined by $\operatorname{diag}(A) = (a_{11}, \ldots, a_{nn})$. Throughout this paper, $\{M_m\}_{m=1}^\infty$ will denote a sequence of symmetric matrices converging to 0, and $\{U_m\}_{m=1}^\infty$ will denote a sequence of orthogonal matrices. We describe sets in $\mathbb{R}^n$ and functions on $\mathbb{R}^n$ as *symmetric* if they are invariant under coordinate permutations. Thus $f : \mathbb{R}^n \to \mathbb{R}$ denotes a function, defined on an open symmetric set, with the property

$$f(x) = f(Px) \text{ for any permutation matrix } P \text{ and any } x \in \operatorname{domain} f.$$

We denote the gradient of f by ∇f or f' and the Hessian by $\nabla^2 f$ or f''. Vectors are understood to be column vectors unless stated otherwise. Whenever we denote by μ a vector in $\mathbb{R}^n_\downarrow$ we make the convention that

$$\mu_1 = \cdots = \mu_{k_1} > \mu_{k_1+1} = \cdots = \mu_{k_2} > \mu_{k_2+1} \ldots \mu_{k_r} \qquad (k_0 = 0,\ k_r = n).$$

Thus r is the number of distinct entries. We define a corresponding partition

$$I_1 := \{1, 2, \ldots, k_1\}, \quad I_2 := \{k_1 + 1, k_1 + 2, \ldots, k_2\}, \ldots, \quad I_r := \{k_{r-1} + 1, \ldots, k_r\},$$

and we call these sets *blocks*. We denote the standard basis in $\mathbb{R}^n$ by $e^1, e^2, \ldots, e^n$, and e is the vector with all entries equal to 1. We also define corresponding matrices

$$X_l := [e^{k_{l-1}+1}, \ldots, e^{k_l}] \quad \text{for all } l = 1, \ldots, r.$$

For an arbitrary matrix A, A^i will denote its ith row (a row vector), and $A^{i,j}$ will denote its (i, j)th entry. Finally, we say that a vector a is *block refined* by a vector b

if

$$b_i = b_j \quad \text{implies} \quad a_i = a_j \qquad \text{for all } i, j.$$

We need the following result.

LEMMA 2.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a symmetric function, twice differentiable at the point $\mu \in \mathbb{R}^n_\downarrow$, and let P be a permutation matrix such that $P\mu = \mu$. Then*

(i) $\nabla f(\mu) = P^T \nabla f(\mu)$ *and*

(ii) $\nabla^2 f(\mu) = P^T \nabla^2 f(\mu) P$.

In particular we have the representation

$$\nabla^2 f(\mu) = \begin{pmatrix} a_{11}E_{11} + b_{k_1}J_1 & a_{12}E_{12} & \cdots & a_{1r}E_{1r} \\ a_{21}E_{21} & a_{22}E_{22} + b_{k_2}J_2 & \cdots & a_{2r}E_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}E_{r1} & a_{r2}E_{r2} & \cdots & a_{rr}R_{rr} + b_{k_r}J_r \end{pmatrix},$$

where the E_{uv} are matrices of dimensions $|I_u| \times |I_v|$ with all entries equal to one, $(a_{ij})_{i,j=1}^r$ is a real symmetric matrix, $b := (b_1, \ldots, b_n)$ is a vector which is block refined by μ, and J_u is an identity matrix of the same dimensions as E_{uu}.

Proof. Just apply the chain rule twice to the equality $f(\nu) = f(P\nu)$ in order to get parts (i) and (ii). To deduce the block structure of the Hessian, consider the block structure of permutation matrices P such that $P\mu = \mu$: then, when we permute the rows and the columns of the Hessian in the way defined by P, it must stay unchanged. □

Using the notation of this lemma, we define the matrix

$$B := \nabla^2 f(\mu) - \operatorname{Diag} b = (a_{ij}E_{ij})_{i,j=1}^r. \tag{2.1}$$

NOTE 2.2. *We make the convention that if the ith diagonal block in the above representation has dimensions 1×1, then we set $a_{ii} = 0$ and $b_{k_i} = f''_{k_i k_i}(\mu)$. Otherwise the value of b_{k_i} is uniquely determined as the difference between a diagonal and an off-diagonal element of this block. Note also that the matrix B and the vector b depend on the point μ and the function f.*

LEMMA 2.3. *For $\mu \in \mathbb{R}^n_\downarrow$ and a sequence of symmetric matrices $M_m \to 0$ we have that*

$$\lambda(\operatorname{Diag}\mu + M_m)^T = \mu^T + \big(\lambda(X_1^T M_m X_1)^T, \ldots, \lambda(X_r^T M_m X_r)^T\big) + o(\|M_m\|).$$

Proof. Combine Lemma 5.10 in [10] and Theorem 3.12 in [3]. □

The following is our main technical tool.

LEMMA 2.4. *Let $\{M_m\}$ be a sequence of symmetric matrices converging to 0 such that $M_m/\|M_m\|$ converges to M. Let μ be in $\mathbb{R}^n_\downarrow$ and $U_m \to U \in O^n$ be a sequence of orthogonal matrices such that*

$$\operatorname{Diag}\mu + M_m = U_m\big(\operatorname{Diag}\lambda(\operatorname{Diag}\mu + M_m)\big)U_m^T \quad \text{for all } m = 1, 2, \ldots. \tag{2.2}$$

Then the following properties hold.

(i) *The orthogonal matrix U has the form*

$$U = \begin{pmatrix} V_1 & 0 & \cdots & 0 \\ 0 & V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_r \end{pmatrix},$$

where V_l is an orthogonal matrix with dimensions $|I_l| \times |I_l|$ for all l.

(ii) *If $i \in I_l$, then*

$$\lim_{m\to\infty} \frac{1 - \sum_{p\in I_l} \left(U_m^{i,p}\right)^2}{\|M_m\|} = 0.$$

(iii) *If i and j do not belong to the same block, then*

$$\lim_{m\to\infty} \frac{\left(U_m^{i,j}\right)^2}{\|M_m\|} = 0.$$

(iv) *If $i \in I_l$, then*

$$V_l^i \left(\operatorname{Diag} \lambda(X_l^T M X_l)\right) (V_l^i)^T = M^{i,i}.$$

(v) *If $i, j \in I_l$, and $p \notin I_l$, then*

$$\lim_{m\to\infty} \frac{U_m^{i,p} U_m^{j,p}}{\|M_m\|} = 0.$$

(vi) *For any indices $i \neq j$ such that $i, j \in I_l$,*

$$\lim_{m\to\infty} \frac{\sum_{p\in I_l} U_m^{i,p} U_m^{j,p}}{\|M_m\|} = 0.$$

(vii) *For any indices $i \neq j$ such that $i, j \in I_l$,*

$$V_l^i \big(\operatorname{Diag} \lambda(X_l^T M X_l)\big) (V_l^j)^T = M^{i,j}.$$

(viii) *For any three indices i, j, p in distinct blocks,*

$$\lim_{m\to\infty} \frac{U_m^{i,p} U_m^{j,p}}{\|M_m\|} = 0.$$

(ix) *For any two indices i, j such that $i \in I_l$, $j \in I_s$, where $l \neq s$,*

$$\lim_{m\to\infty} \left(\mu_{k_l} \frac{\sum_{p\in I_l} U_m^{i,p} U_m^{j,p}}{\|M_m\|} + \mu_{k_s} \frac{\sum_{p\in I_s} U_m^{i,p} U_m^{j,p}}{\|M_m\|} \right) = M^{i,j}.$$

Proof.

(i) After taking the limit in (2.2) we are left with

$$(\operatorname{Diag} \mu) U = U (\operatorname{Diag} \mu).$$

The described representation of the matrix U follows.

(ii) Let us denote

$$h_m = \left(\lambda(X_1^T M_m X_1)^T, \ldots, \lambda(X_r^T M_m X_r)^T\right)^T. \tag{2.3}$$

We use Lemma 2.3 in (2.2) to obtain

$$\operatorname{Diag} \mu + M_m = U_m (\operatorname{Diag} \mu) U_m^T + U_m (\operatorname{Diag} h_m) U_m^T + o(\|M_m\|)$$

and the equivalent form

$$U_m^T(\operatorname{Diag}\mu)U_m + U_m^T M_m U_m = \operatorname{Diag}\mu + \operatorname{Diag} h_m + o(\|M_m\|).$$

We now divide both sides of these equations by $\|M_m\|$ and rearrange:

$$\frac{\operatorname{Diag}\mu - U_m(\operatorname{Diag}\mu)U_m^T}{\|M_m\|} = -\frac{M_m}{\|M_m\|} + \frac{U_m(\operatorname{Diag} h_m)U_m^T}{\|M_m\|} + o(1) \tag{2.4}$$

and

$$\frac{\operatorname{Diag}\mu - U_m^T(\operatorname{Diag}\mu)U_m}{\|M_m\|} = \frac{U_m^T M_m U_m}{\|M_m\|} - \frac{\operatorname{Diag} h_m}{\|M_m\|} - o(1). \tag{2.5}$$

Notice that the right-hand sides of these equations converge to a finite limit as m increases to infinity. If we call the matrix limit of the right-hand side of the first equation L, then clearly the limit of the second equation is $-U^T L U$. We are now going to prove parts (ii) and (iii) together inductively by dividing the orthogonal matrix U_m into the same block structure as U. We begin by considering the first row of blocks of U_m.

Let i be an index in the first block, I_1. Then the limit of the (i,i)th entry in the matrix at the left-hand side of (2.4) is

$$\lim_{m\to\infty} \frac{\left(\mu_{k_1}\left(1 - \sum_{p\in I_1}\left(U_m^{i,p}\right)^2\right) - \sum_{s=2}^r \mu_{k_s}\sum_{p\in I_s}\left(U_m^{i,p}\right)^2\right)}{\|M_m\|} = L^{i,i}. \tag{2.6}$$

Now recall that

$$L^{i,i} = -M^{i,i} + V_1^i(\operatorname{Diag}\lambda(X_1^T M X_1))(V_1^i)^T,$$

and because V_1 is an orthogonal matrix, notice that

$$\begin{aligned}
\sum_{i\in I_1} L^{i,i} &= -\operatorname{tr}(X_1^T M X_1) + \sum_{i\in I_1} V_1^i(\operatorname{Diag}\lambda(X_1^T M X_1))(V_1^i)^T \\
&= -\operatorname{tr}(X_1^T M X_1) + \sum_{i\in I_1}\lambda_i(X_1^T M X_1)\sum_{j\in I_1}(V_1^{j,i})^2 \\
&= -\operatorname{tr}(X_1^T M X_1) + \sum_{i\in I_1}\lambda_i(X_1^T M X_1) \\
&= 0.
\end{aligned}$$

We now sum (2.6) over all i in I_1 to get

$$\lim_{m\to\infty} \frac{\left(\mu_{k_1}\left(|I_1| - \sum_{i,p\in I_1}\left(U_m^{i,p}\right)^2\right) - \sum_{s=2}^r \mu_{k_s}\sum_{i\in I_1,\, p\in I_s}\left(U_m^{i,p}\right)^2\right)}{\|M_m\|} = 0.$$

Notice here that the coefficients in front of the μ_{k_l}, $l = 1, 2, \ldots, r$, in the numerator sum up to zero. That is,

$$|I_1| - \sum_{i,p\in I_1}\left(U_m^{i,p}\right)^2 - \sum_{s=2}^r \sum_{i\in I_1,\, p\in I_s}\left(U_m^{i,p}\right)^2 = 0.$$

So let us choose a number α such that

$$(\mu + \alpha e)_{k_1} > 0 > (\mu + \alpha e)_{k_1+1}$$

and add α to every coordinate of the vector μ thus "shifting" it. The coordinates of the shifted vector that are in the first block are strictly bigger than zero, and the rest are strictly less than zero. By our comment above, the last limit remains true if we "shift" μ in this way. If we rewrite the last limit for the "shifted" vector, because all summands are positive, we immediately see that we must have

$$\lim_{m\to\infty} \frac{|I_1| - \sum_{i,p\in I_1} \left(U_m^{i,p}\right)^2}{\|M_m\|} = 0 \tag{2.7}$$

and

$$\lim_{m\to\infty} \frac{\sum_{i\in I_1,\, p\in I_s} \left(U_m^{i,p}\right)^2}{\|M_m\|} = 0 \quad \text{for all } s = 2, \ldots, r. \tag{2.8}$$

The first of these limits can be written as

$$\lim_{m\to\infty} \frac{\sum_{i\in I_1} \left(1 - \sum_{p\in I_1} \left(U_m^{i,p}\right)^2\right)}{\|M_m\|} = 0,$$

and because all the summands are positive, we conclude that

$$\lim_{m\to\infty} \frac{1 - \sum_{p\in I_1} \left(U_m^{i,p}\right)^2}{\|M_m\|} = 0 \quad \text{for all } i \in I_1.$$

The second of the limits implies immediately that

$$\lim_{m\to\infty} \frac{\left(U_m^{i,p}\right)^2}{\|M_m\|} = 0 \quad \text{for any } i \in I_1,\ p \notin I_1.$$

Thus we proved part (ii) for $i \in I_1$ and part (iii) for the cases specified above. Here is a good place to say a few more words about the idea of the proof. As we said, we divide the matrix U_m into blocks complying with the block structure of the vector μ (exactly as in part (i) for the matrix U). We proved parts (ii) and (iii) for the elements in the first row of blocks of this division. What we are going to do now is prove the same thing for the first *column* of blocks. In order to do this we fix an index i in I_1 and consider the (i,i)th entry in the matrix at the left-hand side of (2.5), and take the limit:

$$\lim_{m\to\infty} \frac{\mu_{k_1}\left(1 - \sum_{p\in I_1} \left(U_m^{p,i}\right)^2\right) - \sum_{s=2}^{r} \mu_{k_s} \sum_{p\in I_s} \left(U_m^{p,i}\right)^2}{\|M_m\|} = -(U^T L U)^{i,i}. \tag{2.9}$$

Using also the block-diagonal structure of the matrix U, we again have

$$\sum_{i\in I_1} (U^T L U)^{i,i} = \sum_{i\in I_1} L^{i,i} = 0.$$

So we proceed just as before in order to conclude that

$$\lim_{m\to\infty} \frac{1-\sum_{p\in I_1}\left(U_m^{p,i}\right)^2}{\|M_m\|}=0 \quad \text{for all } i\in I_1$$

and

$$\lim_{m\to\infty} \frac{\left(U_m^{p,i}\right)^2}{\|M_m\|}=0 \quad \text{for any } i\in I_1,\ p\notin I_1. \tag{2.10}$$

We are now ready for the second step of our induction. Let i be an index in I_2. Then the limit of the (i,i)th entry in the matrix at the left-hand side of (2.4) is

$$\lim_{m\to\infty}\frac{1}{\|M_m\|}\Big(-\mu_{k_1}\sum_{p\in I_1}\left(U_m^{i,p}\right)^2+\mu_{k_2}\Big(1-\sum_{p\in I_2}\left(U_m^{i,p}\right)^2\Big) -\sum_{s=3}^{r}\mu_{k_s}\sum_{p\in I_s}\left(U_m^{i,p}\right)^2\Big)=L^{i,i}.$$

Analogously to the above we have

$$\sum_{i\in I_2}L^{i,i}=0,$$

so summing the above limit over all i in I_2 we get

$$\lim_{m\to\infty}\frac{1}{\|M_m\|}\Big(-\mu_{k_1}\sum_{i\in I_2,\,p\in I_1}\left(U_m^{i,p}\right)^2+\mu_{k_2}\Big(|I_2|-\sum_{i,p\in I_2}\left(U_m^{i,p}\right)^2\Big) -\sum_{s=3}^{r}\mu_{k_s}\sum_{i\in I_2,\,p\in I_s}\left(U_m^{i,p}\right)^2\Big)=0.$$

We know from (2.10) that

$$\lim_{m\to\infty}\frac{\sum_{i\in I_2,\,p\in I_1}\left(U_m^{i,p}\right)^2}{\|M_m\|}=0.$$

So now we choose a number α such that

$$(\mu+\alpha e)_{k_2}>0>(\mu+\alpha e)_{k_2+1}$$

and as before exchange μ with its shifted version. Just as before we conclude that

$$\lim_{m\to\infty} \frac{1-\sum_{p\in I_2}\left(U_m^{i,p}\right)^2}{\|M_m\|}=0 \quad \text{for all } i\in I_2$$

and

$$\lim_{m\to\infty} \frac{\left(U_m^{i,p}\right)^2}{\|M_m\|}=0 \quad \text{for any } i\in I_2,\ p\notin I_2.$$

We repeat the same steps for the second column of blocks in the matrix U_m and so on inductively until we exhaust all the blocks. This completes the proof of parts (ii) and (iii).

(iv) For the proof of this part, one needs to consider the (i,i)th entry of the right-hand side of (2.4). Because the diagonal of the left-hand side converges to zero (by (ii) and (iii)), taking the limit proves the statement in this part.

(v) This part follows immediately from part (iii).

(vi) Taking the limit in (2.4) gives

$$\lim_{m\to\infty} -\sum_{s\neq l} \mu_{k_s} \frac{\sum_{p\in I_s} U_m^{i,p} U_m^{j,p}}{\|M_m\|} - \mu_{k_l} \frac{\sum_{p\in I_l} U_m^{i,p} U_m^{j,p}}{\|M_m\|} = L^{i,j},$$

where $L^{i,j}$ is the (i,j)th entry of the limit of the right-hand side of (2.4). Note that the coefficients of μ_{k_s} again sum up to zero:

$$\sum_{s\neq l} \sum_{p\in I_s} U_m^{i,p} U_m^{j,p} + \sum_{p\in I_l} U_m^{i,p} U_m^{j,p} = 0$$

because U_m is an orthogonal matrix. Now by part (v) we have

$$0 = \lim_{m\to\infty} -\sum_{s\neq l} \frac{\sum_{p\in I_s} U_m^{i,p} U_m^{j,p}}{\|M_m\|} = \lim_{m\to\infty} \frac{\sum_{p\in I_l} U_m^{i,p} U_m^{j,p}}{\|M_m\|}$$

as required, and moreover $L^{i,j} = 0$.

(vii) The statement of this part is the detailed way of writing the fact, proved in the previous part, that $L^{i,j} = 0$.

(viii) This part follows immediately from part (iii). (In fact the expression in part (v) is identical to the one in part (viii), reiterated with different index conditions for later convenience.)

(ix) We again take the limit of the (i,j)th entry of the matrices on both sides of (2.4):

$$\lim_{m\to\infty} \Bigg(-\sum_{t\neq l,s} \mu_{k_t} \frac{\sum_{p\in I_t} U_m^{i,p} U_m^{j,p}}{\|M_m\|} - \mu_{k_l} \frac{\sum_{p\in I_l} U_m^{i,p} U_m^{j,p}}{\|M_m\|} - \mu_{k_s} \frac{\sum_{p\in I_s} U_m^{i,p} U_m^{j,p}}{\|M_m\|} \Bigg) = L^{i,j}.$$

By part (viii) we have that all but the lth and the sth summand above converge to zero. On the other hand

$$\begin{aligned} L^{i,j} &= \lim_{m\to\infty} \Big(-\frac{M_m}{\|M_m\|} + \frac{U_m (\operatorname{Diag} h_m) U_m^T}{\|M_m\|} \Big)^{i,j} \\ &= -M^{i,j} + U^i \Big(\lim_{m\to\infty} \frac{\operatorname{Diag} h_m}{\|M_m\|} \Big) (U^j)^T \\ &= -M^{i,j} \end{aligned}$$

because U^i and U^j are rows in different blocks and $(\operatorname{Diag} h_m)/\|M_m\|$ converges to a diagonal matrix. □

Now we have all the tools to prove the main result of the paper.

3. Twice differentiable spectral functions. In this section we prove that a symmetric function f is twice differentiable at the point $\lambda(A)$ if and only if the corresponding spectral function $f \circ \lambda$ is twice differentiable at the matrix A.

Recall that the Hadamard product of two matrices $A = [A^{i,j}]$ and $B = [B^{i,j}]$ of the same size is the matrix of their elementwise products $A \circ B = [A^{i,j}B^{i,j}]$. Let the symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable at the point $\mu \in \mathbb{R}^n_\downarrow$, where, as before,

$$\mu_1 = \cdots = \mu_{k_1} > \mu_{k_1+1} = \cdots = \mu_{k_2} > \mu_{k_2+1} \ldots \mu_{k_r} \qquad (k_0 = 0,\, k_r = n).$$

We define the vector $b(\mu) = (b_1(\mu), \ldots, b_n(\mu))$ as in Lemma 2.1. Specifically, for any index i (say $i \in I_l$ for some $l \in \{1, 2, \ldots, r\}$), we define

$$b_i(\mu) = \begin{cases} f''_{ii}(\mu) & \text{if } |I_l| = 1, \\ f''_{pp}(\mu) - f''_{pq}(\mu) & \text{for any } p \neq q \in I_l. \end{cases}$$

Lemma 2.1 guarantees that the second case of this definition doesn't depend on the choice of p and q. We also define the matrix $\mathcal{A}(\mu)$:

$$\mathcal{A}^{i,j}(\mu) = \begin{cases} 0 & \text{if } \ i = j, \\ b_i(\mu) & \text{if } \ i \neq j \text{ but } i, j \in I_l, \\ \frac{f'_i(\mu) - f'_j(\mu)}{\mu_i - \mu_j} & \text{otherwise.} \end{cases} \tag{3.1}$$

Notice the similarity between this definition and classical divided difference constructions in Löwner theory (see [1, Chap. V], for example). For simplicity, when the argument is understood by the context, we will write just b_i and $\mathcal{A}^{i,j}$. The following lemma is Theorem 1.1 in [9].

LEMMA 3.1. *Let $A \in S^n$ and suppose $\lambda(A)$ belongs to the domain of the symmetric function $f : \mathbb{R}^n \to \mathbb{R}$. Then f is differentiable at the point $\lambda(A)$ if and only if $f \circ \lambda$ is differentiable at the point A. In that case we have the formula*

$$\nabla(f \circ \lambda)(A) = U\big(\operatorname{Diag} \nabla f(\lambda(A))\big)U^T$$

for any orthogonal matrix U satisfying $A = U\big(\operatorname{Diag} \lambda(A)\big)U^T$.

We recall some standard notions about twice differentiability. Consider a function F from S^n to $\mathbb{R}$. Its gradient at any point A (when it exists) is a linear functional on the Euclidean space S^n and thus can be identified with an element of S^n, which we denote $\nabla F(A)$. Thus ∇F is a map from S^n to S^n. When this map is itself differentiable at A we say F is *twice differentiable* at A. In this case we can interpret the Hessian $\nabla^2 F(A)$ as a symmetric, bilinear function from $S^n \times S^n$ into $\mathbb{R}$. Its value at a particular point $(H, Y) \in S^n \times S^n$ will be denoted $\nabla^2 F(A)[H, Y]$. In particular, for fixed H, the function $\nabla^2 F(A)[H, \cdot]$ is again a linear functional on S^n, which we consider an element of S^n, for brevity denoted by $\nabla^2 F(A)[H]$. When the Hessian is continuous at A we say F is *twice continuously differentiable* at A. In that case the following identity holds:

$$\nabla^2 F(A)[H, H] = \frac{d^2}{dt^2} F(A + tH)\bigg|_{t=0}.$$

The next theorem is a preliminary version of our main result.

THEOREM 3.2. *The symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at the point $\mu \in \mathbb{R}^n_\downarrow$ if and only if $f \circ \lambda$ is twice differentiable at the point* $\operatorname{Diag}\mu$. *In that case the Hessian is given by*

$$\nabla^2(f \circ \lambda)(\operatorname{Diag}\mu)[H] = \operatorname{Diag}\big(\nabla^2 f(\mu)(\operatorname{diag} H)\big) + \mathcal{A} \circ H. \tag{3.2}$$

Hence

$$\nabla^2(f \circ \lambda)(\operatorname{Diag}\mu)[H, H] = \nabla^2 f(\mu)[\operatorname{diag} H, \operatorname{diag} H] + \langle \mathcal{A}, H \circ H \rangle.$$

Proof. It is easy to see that f must be twice differentiable at the point μ whenever $f \circ \lambda$ is twice differentiable at $\operatorname{Diag}\mu$ because by restricting $f \circ \lambda$ to the subspace of diagonal matrices we get the function f. So the interesting case is the other direction. Let f be twice differentiable at the point $\mu \in \mathbb{R}^n_\downarrow$ and suppose on the contrary that either $f \circ \lambda$ is not twice differentiable at the point $\operatorname{Diag}\mu$, or (3.2) fails. Define a linear operator Δ by

$$\Delta(H) := \operatorname{Diag}\big((\nabla^2 f(\mu)(\operatorname{diag} H)\big) + \mathcal{A} \circ H.$$

(Lemma 3.1 tells us that $f \circ \lambda$ is at least differentiable around $\operatorname{Diag}\mu$.) So, for this linear operator Δ there is an $\epsilon > 0$ and a sequence of symmetric matrices $\{M_m\}_{m=1}^\infty$ converging to 0 such that

$$\frac{\|\nabla(f \circ \lambda)(\operatorname{Diag}\mu + M_m) - \nabla(f \circ \lambda)(\operatorname{Diag}\mu) - \Delta(M_m)\|}{\|M_m\|} > \epsilon$$

for all $m = 1, 2, \ldots$. Without loss of generality we may assume that the sequence $\{M_m\}_{m=1}^\infty$ is such that $M_m/\|M_m\|$ converges to a matrix M because some subsequence of $\{M_m\}_{m=1}^\infty$ surely has this property. Let $\{U_m\}_{m=1}^\infty$ be a sequence of orthogonal matrices such that

$$\operatorname{Diag}\mu + M_m = U_m\big(\operatorname{Diag}\lambda(\operatorname{Diag}\mu + M_m)\big)U_m^T \quad \text{for all} \quad m = 1, 2, \ldots.$$

Without loss of generality we may assume that $U_m \to U \in O^n$, or otherwise we will just take subsequences of $\{M_m\}_{m=1}^\infty$ and $\{U_m\}_{m=1}^\infty$. The above inequality shows that for every m there corresponds a pair (or more precisely at least one pair) of indices (i, j) such that

$$\frac{|\big(\nabla(f \circ \lambda)(\operatorname{Diag}\mu + M_m) - \operatorname{Diag}\nabla f(\mu) - \Delta(M_m)\big)^{i,j}|}{\|M_m\|} > \frac{\epsilon}{n}. \tag{3.3}$$

So at least for one pair of indices, call it again (i, j), we have infinitely many numbers m for which (i, j) is the corresponding pair, and because if necessary we can again take a subsequence of $\{M_m\}_{m=1}^\infty$ and $\{U_m\}_{m=1}^\infty$, we may assume without loss of generality that there is a pair of indices (i, j) for which the last inequality holds for all $m = 1, 2, \ldots$. Define the symbol h_m again by (2.3). Notice that using Lemma 3.1, Lemma 2.3, and the fact that ∇f is differentiable at μ we get

$$\begin{aligned}
&\nabla(f \circ \lambda)(\operatorname{Diag}\mu + M_m) = U_m\big(\operatorname{Diag}\nabla f(\lambda(\operatorname{Diag}\mu + M_m))\big)U_m^T \\
&= U_m\big(\operatorname{Diag}\nabla f(\mu + h_m + o(\|M_m\|))\big)U_m^T \\
&= U_m(\operatorname{Diag}(\nabla f(\mu) + \nabla^2 f(\mu)h_m + o(\|M_m\|)))U_m^T \\
&= U_m(\operatorname{Diag}\nabla f(\mu))U_m^T + U_m(\operatorname{Diag}(\nabla^2 f(\mu)h_m))U_m^T + o(\|M_m\|).
\end{aligned} \tag{3.4}$$

We consider three cases. In every case we are going to show that the left-hand side of inequality (3.3) actually converges to zero, which contradicts the assumption.

Case I. If $i = j$, then using (3.4) the left-hand side of inequality (3.3) is less than or equal to

$$\frac{|U_m^i\big(\operatorname{Diag}\nabla f(\mu)\big)(U_m^i)^T - f_i'(\mu)|}{\|M_m\|} + \frac{|U_m^i\big(\operatorname{Diag}\nabla^2 f(\mu)h_m\big)(U_m^i)^T - \big(\nabla^2 f(\mu)(\operatorname{diag} M_m)\big)_i|}{\|M_m\|} + o(1).$$

We are going to show that each summand approaches zero as m goes to infinity. Assume that $i \in I_l$ for some $l \in \{1, \dots, r\}$. Using the fact that the vector μ block refines the vector $\nabla f(\mu)$ (Lemma 2.1, part (i)) the first term can be written as

$$\frac{1}{\|M_m\|}\left| f_{k_l}'(\mu)\left(1 - \sum_{p\in I_l}\big(U_m^{i,p}\big)^2\right) - \sum_{s:s\neq l} f_{k_s}'(\mu)\sum_{p\in I_s}\big(U_m^{i,p}\big)^2\right|.$$

We now apply Lemma 2.4, parts (ii) and (iii) to the last expression.

We concentrate now on the second term above. Using the notation of (2.1) (that is, $\nabla^2 f(\mu) = B + \operatorname{Diag} b$) this term is less than or equal to

$$\frac{|U_m^i\big(\operatorname{Diag}\,(Bh_m)\big)(U_m^i)^T - \big(B(\operatorname{diag} M_m)\big)_i|}{\|M_m\|} + \frac{|U_m^i\big(\operatorname{Diag}\,((\operatorname{Diag} b)h_m)\big)(U_m^i)^T - \big((\operatorname{Diag} b)(\operatorname{diag} M_m)\big)_i|}{\|M_m\|}. \tag{3.5}$$

As m approaches infinity, we have that $U_m^i \to U^i$. We define the vector h to be

$$h := \lim_{m\to\infty}\frac{h_m}{\|M_m\|} = \big(\lambda(X_1^T M X_1)^T, \dots, \lambda(X_r^T M X_r)^T\big)^T.$$

So taking limits, expression (3.5) turns into

$$|U^i\big(\operatorname{Diag}\,(Bh)\big)(U^i)^T - \big(B(\operatorname{diag} M)\big)_i| + |U^i\big(\operatorname{Diag}\,((\operatorname{Diag} b)h)\big)(U^i)^T - \big((\operatorname{Diag} b)(\operatorname{diag} M)\big)_i|.$$

We are going to investigate each term in this sum separately and show that they are both actually equal to zero. For the first, we use the block structure of the matrix B (see Lemma 2.1) and the block structure of the vector h to obtain

$$(Bh)_j = \sum_{s=1}^{r} a_{qs}\operatorname{tr}\,(X_s^T M X_s) \quad \text{when } j \in I_q.$$

Using the fact that $i \in I_l$ and that V_l is orthogonal we get

$$\begin{aligned} U^i\big(\operatorname{Diag}\,(Bh)\big)(U^i)^T &= \big(V_l^i X_l^T\big)\big(\operatorname{Diag}\,(Bh)\big)\big(X_l(V_l^i)^T\big) \\ &= V_l^i\big(X_l^T(\operatorname{Diag}\,(Bh))X_l\big)(V_l^i)^T \\ &= \left(\sum_{s=1}^{r} a_{ls}\operatorname{tr}\,(X_s^T M X_s)\right)\left(\sum_{s=1}^{|I_l|}(V_l^{i,s})^2\right) \end{aligned}$$

$$= \sum_{s=1}^{r} a_{ls} \mathrm{tr}\,(X_s^T M X_s)$$
$$= (B \mathrm{diag}\, M)_i,$$

which shows that the first term is zero. For the second term, we use the block structure of the vector b to write

$$(\mathrm{Diag}\, b)h = \big(b_{k_1}\lambda(X_1^T M X_1)^T, \ldots, b_{k_r}\lambda(X_r^T M X_r)^T\big)^T.$$

In the next to the last equality below we use part (iv) of Lemma 2.4:

$$\begin{aligned} U^i\big(\mathrm{Diag}\,((\mathrm{Diag}\, b)h)\big)(U^i)^T &= \big(V_l^i X_l^T\big)\big(\mathrm{Diag}\,((\mathrm{Diag}\, b)h)\big)\big(X_l(V_l^i)^T\big) \\ &= V_l^i\big(X_l^T(\mathrm{Diag}\,((\mathrm{Diag}\, b)h))X_l\big)(V_l^i)^T \\ &= V_l^i\big(\mathrm{Diag}\, b_{k_l}\lambda(X_l^T M X_l)\big)(V_l^i)^T \\ &= b_{k_l} M^{i,i} \\ &= \big((\mathrm{Diag}\, b)(\mathrm{diag}\, M)\big)_i. \end{aligned}$$

We can see now that the second term is also zero.

Case II. If $i \neq j$ but $i, j \in I_l$ for some $l \in \{1, 2, \ldots r\}$, then using (3.4) the left-hand side of inequality (3.3) becomes

$$\frac{|U_m^i\big(\mathrm{Diag}\,\nabla f(\mu)\big)(U_m^j)^T + U_m^i\big(\mathrm{Diag}\,(\nabla^2 f(\mu)h_m)\big)(U_m^j)^T - b_{k_l} M_m^{i,j}|}{\|M_m\|} + o(1).$$

Using the fact that μ block refines the vector $\nabla f(\mu)$, we can write the first summand above as

$$\frac{1}{\|M_m\|}\left(\sum_{s\neq l} f'_{k_s}(\mu) \sum_{p\in I_s} U_m^{i,p} U_m^{j,p} + f'_{k_l}(\mu) \sum_{p\in I_l} U_m^{i,p} U_m^{j,p}\right).$$

We use parts (v) and (vi) of Lemma 2.4 to conclude that this expression converges to zero. We are left with

$$\frac{|U_m^i\big(\mathrm{Diag}\,(\nabla^2 f(\mu)h_m)\big)(U_m^j)^T - b_{k_l} M_m^{i,j}|}{\|M_m\|}.$$

Substituting $\nabla^2 f(\mu) = B + \mathrm{Diag}\, b$ we get

$$\frac{|U_m^i\big(\mathrm{Diag}\,(Bh_m)\big)(U_m^j)^T + U_m^i\big(\mathrm{Diag}\,((\mathrm{Diag}\, b)h_m)\big)(U_m^j)^T - b_{k_l} M_m^{i,j}|}{\|M_m\|}.$$

Recall the notation from Lemma 2.1 used to denote the entries of the matrix B. Then the limit of the first summand above can be written as

$$\begin{aligned} \lim_{m\to\infty} \frac{U_m^i\big(\mathrm{Diag}\,(Bh_m)\big)(U_m^j)^T}{\|M_m\|} &= U^i\big(\mathrm{Diag}\,(Bh)\big)(U^j)^T \\ &= \sum_{s=1}^{r}\left(\left(\sum_{l=1}^{r} a_{sl}\,\mathrm{tr}\,(X_l^T M X_l)\right)\sum_{p\in I_s} U^{i,p}U^{j,p}\right) \\ &= 0 \end{aligned}$$

because clearly $\sum_{p\in I_s} U^{i,p}U^{j,p} = 0$ for all $s \in \{1,2,\dots,r\}$. We are left with the following limit:

$$\lim_{m\to\infty} \frac{|U^i_m\big(\operatorname{Diag}\left((\operatorname{Diag} b)h_m\right)\big)(U^j_m)^T - b_{k_l}M^{i,j}_m|}{\|M_m\|} = |U^i\big(\operatorname{Diag}\left((\operatorname{Diag} b)h\right)\big)(U^j)^T - b_{k_l}M^{i,j}|.$$

Using Lemma 2.4, part (vii) we observe that the right-hand side is zero.

Case III. If $i \in I_l$ and $j \in I_s$, where $l \neq s$, then using (3.4), the left-hand side of inequality (3.3) becomes (up to $o(1)$)

$$\frac{|U^i_m\big(\operatorname{Diag}\nabla f(\mu)\big)(U^j_m)^T + U^i_m\big(\operatorname{Diag}\nabla^2 f(\mu)h_m\big)(U^j_m)^T - \frac{f'_{k_l}(\mu)-f'_{k_s}(\mu)}{\mu_{k_l}-\mu_{k_s}}M^{i,j}_m|}{\|M_m\|}.$$

We start with the second term above. Its limit is

$$\lim_{m\to\infty} \frac{U^i_m\big(\operatorname{Diag}\left(\nabla^2 f(\mu)h_m\right)\big)(U^j_m)^T}{\|M_m\|} = U^i\big(\operatorname{Diag}\left(\nabla^2 f(\mu)h\right)\big)(U^j)^T = 0$$

because in our case U^i has nonzero coordinates where the entries of U^j are zero. We are left with

$$\lim_{m\to\infty} \left| \frac{U^i_m\big(\operatorname{Diag}\nabla f(\mu)\big)(U^j_m)^T}{\|M_m\|} - \frac{f'_{k_l}(\mu) - f'_{k_s}(\mu)}{\mu_{k_l}-\mu_{k_s}} \frac{M^{i,j}_m}{\|M_m\|} \right|. \tag{3.6}$$

We expand the first term in this limit:

$$\frac{U^i_m\big(\operatorname{Diag}\nabla f(\mu)\big)(U^j_m)^T}{\|M_m\|} = f'_{k_l}(\mu)\frac{\sum_{p\in I_l} U^{i,p}_m U^{j,p}_m}{\|M_m\|} + f'_{k_s}(\mu)\frac{\sum_{p\in I_s} U^{i,p}_m U^{j,p}_m}{\|M_m\|} + \sum_{t\neq l,s} f'_{k_t}(\mu)\frac{\sum_{p\in I_t} U^{i,p}_m U^{j,p}_m}{\|M_m\|}.$$

Using Lemma 2.4, part (viii) we see that the third summand above converges to zero as m goes to infinity. Part (ix) of the same lemma tells us that

$$\lim_{m\to\infty} \frac{M^{i,j}_m}{\|M_m\|} = \lim_{m\to\infty} \left(\mu_{k_l} \frac{\sum_{p\in I_l} U^{i,p}_m U^{j,p}_m}{\|M_m\|} + \mu_{k_s} \frac{\sum_{p\in I_s} U^{i,p}_m U^{j,p}_m}{\|M_m\|} \right).$$

In order to abbreviate the formulae we introduce the following notation:

$$\beta^l_m := \frac{\sum_{p\in I_l} U^{i,p}_m U^{j,p}_m}{\|M_m\|} \qquad \text{for all} \quad l = 1,2,\dots,r.$$

Substituting everything in (3.6) we get the following equivalent limit:

$$\lim_{m\to\infty} \left| \left(f'_{k_l}(\mu)\beta^l_m + f'_{k_s}(\mu)\beta^s_m \right) - \frac{f'_{k_l}(\mu) - f'_{k_s}(\mu)}{\mu_{k_l}-\mu_{k_s}} \left(\mu_{k_l}\beta^l_m + \mu_{k_s}\beta^s_m \right) \right|.$$

Simplifying we get

$$\lim_{m\to\infty} (\beta^l_m + \beta^s_m) \frac{f'_{k_s}(\mu)\mu_{k_l} - f'_{k_l}(\mu)\mu_{k_s}}{\mu_{k_l}-\mu_{k_s}}.$$

Notice now that

$$\sum_{l=1}^{r} \beta_m^l = 0 \quad \text{for all } m$$

because U_m is an orthogonal matrix and the numerator of the above sum is the product of its ith and the jth row. Next, Lemma 2.4, part (viii) says that

$$\lim_{m\to\infty} \sum_{t\neq l,s} \beta_m^t = 0,$$

so

$$\lim_{m\to\infty} (\beta_m^l + \beta_m^s) = 0,$$

which completes the proof. □

We are finally ready to give and prove the full version of our main result.

THEOREM 3.3. *Let A be an $n \times n$ symmetric matrix. The symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at the point $\lambda(A)$ if and only if the spectral function $f \circ \lambda$ is twice differentiable at the matrix A. Moreover, in this case the Hessian of the spectral function at the matrix A is*

$$\nabla^2(f \circ \lambda)(A)[H] = W\big(\operatorname{Diag}\big(\nabla^2 f(\lambda(A))\operatorname{diag}\tilde{H}\big) + \mathcal{A} \circ \tilde{H}\big)W^T,$$

where W is any orthogonal matrix such that $A = W\big(\operatorname{Diag}\lambda(A)\big)W^T$, $\tilde{H} = W^T H W$, and $\mathcal{A} = \mathcal{A}(\lambda(A))$ is defined by (3.1). *Hence*

$$\nabla^2(f \circ \lambda)(A)[H, H] = \nabla^2 f(\lambda(A))[\operatorname{diag}\tilde{H}, \operatorname{diag}\tilde{H}] + \langle \mathcal{A}, \tilde{H} \circ \tilde{H}\rangle.$$

Proof. Let W be an orthogonal matrix which diagonalizes A in an ordered fashion, that is,

$$A = W\big(\operatorname{Diag}\lambda(A)\big)W^T.$$

Let M_m be a sequence of symmetric matrices converging to zero, and let U_m be a sequence of orthogonal matrices such that

$$\operatorname{Diag}\lambda(A) + W^T M_m W = U_m\big(\operatorname{Diag}\lambda(\operatorname{Diag}\lambda(A) + W^T M_m W)\big)U_m^T.$$

Then using Lemma 3.1 we get

$$\begin{aligned}
&\nabla(f \circ \lambda)(A + M_m)\\
&= \nabla(f \circ \lambda)\big(W(\operatorname{Diag}\lambda(A) + W^T M_m W)W^T\big)\\
&= \nabla(f \circ \lambda)\big(WU_m(\operatorname{Diag}\lambda(\operatorname{Diag}\lambda(A) + W^T M_m W))U_m^T W^T\big)\\
&= WU_m\big(\operatorname{Diag}\nabla f(\lambda(\operatorname{Diag}\lambda(A) + W^T M_m W))\big)U_m^T W^T.
\end{aligned}$$

We also have that

$$\nabla(f \circ \lambda)(A) = W\big(\operatorname{Diag}\nabla f(\lambda(A))\big)W^T,$$

and $W^T M_m W \to 0$, as m goes to infinity. Because W is an orthogonal matrix we have $\|WXW^T\| = \|X\|$ for any matrix X. It is now easy to check the result by Theorem 3.2. □

4. Continuity of the Hessian. Suppose now that the symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable in a neighborhood of the point $\lambda(A)$ and that its Hessian is continuous at the point $\lambda(A)$. Then Theorem 3.3 shows that $f \circ \lambda$ must be twice differentiable in a neighborhood of the point A, and in this section we are going to show that $\nabla^2(f \circ \lambda)$ is also continuous at the point A.

We define a basis, $\{H_{ij}\}$, on the space of symmetric matrices. If $i \neq j$ all the entries of the matrix H_{ij} are zeros, except the (i,j)th and (j,i)th, which are one. If $i = j$ we have one only on the (i,i)th position. It suffices to prove that the Hessian is continuous when applied to any matrix of the basis. We begin with a lemma.

LEMMA 4.1. *Let $\mu \in \mathbb{R}^n_\downarrow$ be such that*

$$\mu_1 = \cdots = \mu_{k_1} > \mu_{k_1+1} = \cdots = \mu_{k_2} > \mu_{k_2+1} \ldots \mu_{k_r} \qquad (k_0 = 0,\, k_r = n),$$

and let the symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable at the point μ. Let $\{\mu^m\}_{m=1}^\infty$ be a sequence of vectors in $\mathbb{R}^n$ converging to μ. Then

$$\lim_{m\to\infty} \nabla^2(f \circ \lambda)(\operatorname{Diag}\mu^m) = \nabla^2(f \circ \lambda)(\operatorname{Diag}\mu).$$

Proof. For every m there is a permutation matrix P_m such that $P_m^T \mu^m = \overline{\mu^m}$. (See the beginning of section 2 for the meaning of the bar above a vector.) But there are finitely many permutation matrices (namely, $n!$) so we can form $n!$ subsequences of $\{\mu^m\}$ such that any two vectors in a particular subsequence can be ordered in descending order by the same permutation matrix. If we prove the lemma for every such subsequence we will be done. So without loss of generality we may assume that $P^T \mu^m = \overline{\mu^m}$ for every m and some fixed permutation matrix P. Clearly, for all large enough m, we have

$$\mu^m_{k_1} > \mu^m_{k_1+1}, \quad \mu^m_{k_2} > \mu^m_{k_2+1}, \ldots, \mu^m_{k_{r-1}} > \mu^m_{k_{r-1}+1}.$$

Consequently the matrix P is block-diagonal with permutation matrices on the main diagonal, and dimensions matching the block structure of μ, so $P\mu = \mu$. Consider now the block structure of the vectors $\{\overline{\mu^m}\}$. Because there are finitely many different block structures, we can divide this sequence into subsequences such that the vectors in a particular subsequence have the same block structure. If we prove the lemma for each subsequence we will be done. So without loss of generality we may assume that the vectors $\{\overline{\mu^m}\}$ have the same block structure for every m. Next, using the formula for the Hessian in Theorem 3.3 we have

$$\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu^m)[H_{ij}] = P\big(\operatorname{Diag}\big(\nabla^2 f(\overline{\mu^m})\operatorname{diag}(P^T H_{ij} P)\big) + \mathcal{A}(\overline{\mu^m}) \circ (P^T H_{ij} P)\big)P^T,$$

and Lemma 2.1 together with Theorem 3.2 gives us

$$\begin{aligned}\nabla^2(f \circ \lambda)(\operatorname{Diag}\mu)[H_{ij}] &= \operatorname{Diag}\big(\nabla^2 f(\mu)\operatorname{diag} H_{ij}\big) + \mathcal{A}(\mu) \circ H_{ij} \\ &= P\big(\operatorname{Diag}\big(\nabla^2 f(\mu)\operatorname{diag}(P^T H_{ij} P)\big) \\ &\quad + \mathcal{A}(\mu) \circ (P^T H_{ij} P)\big)P^T.\end{aligned}$$

These equations show that without loss of generality it suffices to prove the lemma only in the case when all vectors $\{\mu^m\}$ are ordered in descending order, that is, the vectors μ^m all block refine the vector μ. In that case we have

$$\nabla^2(f \circ \lambda)(\operatorname{Diag}\mu^m)[H_{ij}] = \operatorname{Diag}\big(\nabla^2 f(\mu^m)\operatorname{diag} H_{ij}\big) + \mathcal{A}(\mu^m) \circ H_{ij}$$

and

$$\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu)[H_{ij}] = \operatorname{Diag}\left(\nabla^2 f(\mu)\operatorname{diag} H_{ij}\right) + \mathcal{A}(\mu)\circ H_{ij}.$$

We consider four cases.

Case I. If $i=j$, then

$$\begin{aligned}\lim_{m\to\infty}\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu^m)[H_{ij}] &= \lim_{m\to\infty}\operatorname{Diag}\left(\nabla^2 f(\mu^m)e^i\right)\\ &= \operatorname{Diag}\left(\nabla^2 f(\mu)e^i\right)\\ &= \nabla^2(f\circ\lambda)(\operatorname{Diag}\mu)[H_{ij}]\end{aligned}$$

just because $\nabla^2 f(\cdot)$ is continuous at μ.

Case II. If $i\neq j$ but belong to the same block for μ^m, then i, j will be in the same block of μ as well and we have

$$\begin{aligned}\lim_{m\to\infty}\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu^m)[H_{ij}] &= \lim_{m\to\infty} b_i(\mu^m)H_{ij}\\ &= b_i(\mu)H_{ij}\\ &= \nabla^2(f\circ\lambda)(\operatorname{Diag}\mu)[H_{ij}]\end{aligned}$$

again because $\nabla^2 f(\cdot)$ is continuous at μ.

Case III. If i and j belong to different blocks of μ^m but to the same block of μ, then

$$\lim_{m\to\infty}\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu^m)[H_{ij}] = \lim_{m\to\infty}\frac{f_i'(\mu^m)-f_j'(\mu^m)}{\mu_i^m-\mu_j^m}H_{ij}$$

and

$$\nabla^2(f\circ\lambda)(\operatorname{Diag}\mu)[H_{ij}] = b_i(\mu)H_{ij}.$$

So we have to prove that

$$\lim_{m\to\infty}\frac{f_i'(\mu^m)-f_j'(\mu^m)}{\mu_i^m-\mu_j^m} = f_{ii}''(\mu)-f_{ij}''(\mu).$$

(See the definition of $b_i(\mu)$ in the beginning of section 3.) For every m we define the vectors $\dot\mu^m$ and $\ddot\mu^m$ coordinatewise as follows:

$$\dot\mu_p^m = \begin{cases}\mu_p^m, & p\neq i,\\ \mu_j^m, & p=i,\end{cases}\qquad \ddot\mu_p^m = \begin{cases}\mu_p^m, & p\neq i,j,\\ \mu_j^m, & p=i,\\ \mu_i^m, & p=j.\end{cases}$$

Because $\mu_i=\mu_j$ we conclude that both sequences $\{\dot\mu^m\}_{m=1}^\infty$ and $\{\ddot\mu^m\}_{m=1}^\infty$ converge to μ because $\{\mu^m\}_{m=1}^\infty$ does so. Below we are applying the mean value theorem twice:

$$\begin{aligned}\frac{f_i'(\mu^m)-f_j'(\mu^m)}{\mu_i^m-\mu_j^m} &= \frac{f_i'(\mu^m)-f_i'(\dot\mu^m)+f_i'(\dot\mu^m)-f_j'(\mu^m)}{\mu_i^m-\mu_j^m}\\ &= \frac{(\mu_i^m-\mu_j^m)f_{ii}''(\xi^m)+f_i'(\dot\mu^m)-f_j'(\mu^m)}{\mu_i^m-\mu_j^m}\end{aligned}$$

$$= f''_{ii}(\xi^m) + \frac{f'_i(\dot{\mu}^m) - f'_i(\ddot{\mu}^m) + f'_i(\ddot{\mu}^m) - f'_j(\mu^m)}{\mu_i^m - \mu_j^m}$$

$$= f''_{ii}(\xi^m) + \frac{(\mu_j^m - \mu_i^m) f''_{ij}(\eta^m) + f'_i(\ddot{\mu}^m) - f'_j(\mu^m)}{\mu_i^m - \mu_j^m}$$

$$= f''_{ii}(\xi^m) - f''_{ij}(\eta^m),$$

where ξ^m is a vector between μ^m and $\dot{\mu}^m$, and η^m is a vector between $\dot{\mu}^m$ and $\ddot{\mu}^m$. Consequently, $\xi^m \to \mu$ and $\eta^m \to \mu$. Notice that vector $\ddot{\mu}^m$ is obtained from μ^m by swapping the ith and the jth coordinate. Then using the first part of Lemma 2.1 we see that $f'_i(\ddot{\mu}^m) = f'_j(\mu^m)$. Finally we just have to take the limit above and use again the continuity of the Hessian of f at the point μ.

Case IV. If i and j belong to different blocks of μ^m and to different blocks of μ, then

$$\lim_{m\to\infty} \nabla^2(f \circ \lambda)(\operatorname{Diag} \mu^m)[H_{ij}] = \lim_{m\to\infty} \frac{f'_i(\mu^m) - f'_j(\mu^m)}{\mu_i^m - \mu_j^m} H_{ij}$$

$$= \frac{f'_i(\mu) - f'_j(\mu)}{\mu_i - \mu_j} H_{ij}$$

$$= \nabla^2(f \circ \lambda)(\operatorname{Diag} \mu)[H_{ij}]$$

because $\nabla f(\cdot)$ is continuous at μ and the denominator is never zero. □

Now we are ready to prove the main result of this section.

THEOREM 4.2. *Let A be an $n \times n$ symmetric matrix. The symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable at the point $\lambda(A)$ if and only if the spectral function $f \circ \lambda$ is twice continuously differentiable at the matrix A.*

Proof. We know that $f \circ \lambda$ is twice differentiable at A if and only if f is twice differentiable at $\lambda(A)$, so what is left to prove is the continuity of the Hessian. Suppose that f is twice continuously differentiable at $\lambda(A)$ and that $f \circ \lambda$ is not twice continuously differentiable at A, that is, the Hessian $\nabla^2(f \circ \lambda)$ is not continuous at A. Take a sequence, $\{A_m\}_{m=1}^{\infty}$, of symmetric matrices converging to A such that for some $\epsilon > 0$ we have

$$\|\nabla^2(f \circ \lambda)(A_m) - \nabla^2(f \circ \lambda)(A)\| > \epsilon$$

for all m. Let $\{U_m\}_{m=1}^{\infty}$ be a sequence of orthogonal matrices such that

$$A_m = U_m \big(\operatorname{Diag} \lambda(A_m)\big) U_m^T.$$

Without loss of generality we may assume that $U_m \to U$, where U is orthogonal and then

$$A = U \big(\operatorname{Diag} \lambda(A)\big) U^T.$$

(Otherwise we take subsequences of $\{A_m\}$ and $\{U_m\}$.) Using the formula for the Hessian given in Theorem 3.3 and Lemma 4.1 we can easily see that

$$\lim_{m\to\infty} \nabla^2(f \circ \lambda)(A_m)[H] = \nabla^2(f \circ \lambda)(A)[H]$$

for every symmetric H. This is a contradiction.

The other direction follows from the chain rule after observing

$$f(x) = (f \circ \lambda)(\operatorname{Diag} x).$$

This completes the proof. □

5. Example and conjecture. As an example, suppose we require the second directional derivative of the function $f \circ \lambda$ at the point A in the direction B. That is, we want to find the second derivative of the function

$$g(t) = (f \circ \lambda)(A + tB)$$

at $t = 0$. Let W be an orthogonal matrix such that $A = W(\operatorname{Diag}\lambda(A))W^T$. Let $\tilde{B} = W^T B W$. We differentiate twice:

$$g''(t) = \nabla^2 (f \circ \lambda)(A + tB)[B, B].$$

Using Lemma 3.1 and Theorem 3.3 at $t = 0$ we get

$$\begin{aligned}
g(0) &= f(\lambda(A)),\\
g'(0) &= \operatorname{tr}\left(\tilde{B}\operatorname{Diag}\nabla f(\lambda(A))\right),\\
g''(0) &= \nabla^2 (f \circ \lambda)(\lambda(A))[\operatorname{diag}\tilde{B}, \operatorname{diag}\tilde{B}] + \langle \mathcal{A}, \tilde{B} \circ \tilde{B}\rangle\\
&= \sum_{i,j=1}^{n} f''_{ij}(\lambda(A))(\tilde{B}^{i,i})(\tilde{B}^{j,j}) + \sum_{\substack{i\neq j\\ \lambda_i = \lambda_j}} b_i(\tilde{B}^{i,j})^2\\
&\quad + \sum_{\substack{i,j\\ \lambda_i \neq \lambda_j}} \frac{f'_i(\lambda(A)) - f'_j(\lambda(A))}{\lambda_i(A) - \lambda_j(A)}(\tilde{B}^{i,j})^2.
\end{aligned}$$

In principle, if the function f is analytic, this second directional derivative can also be computed using the implicit formulae from [26]. Some work shows that the answers agree.

As a final illustration, consider the classical example of the power series expansion of a simple eigenvalue. In this case we consider the function f given by

$$f(x) = \bar{x}_k := \text{the } k\text{th largest entry in } x$$

and the matrix

$$A = \operatorname{Diag}\mu,$$

where $\mu \in \mathbb{R}^n_{\downarrow}$ and

$$\mu_{k-1} > \mu_k > \mu_{k+1}.$$

Then we have

$$f'(\mu) = e^k \quad \text{and} \quad f''(\mu) = 0,$$

so for the function $g(t) = \lambda_k(\operatorname{Diag}\mu + tB)$ our results show the following formulae (familiar in perturbation theory and quantum mechanics):

$$\begin{aligned}
g(0) &= \mu_k,\\
g'(0) &= B^{k,k},\\
g''(0) &= \sum_{j\neq k} \frac{1}{\mu_k - \mu_j}(B^{k,j})^2 + \sum_{i\neq k} \frac{-1}{\mu_i - \mu_k}(B^{i,k})^2\\
&= 2\sum_{j\neq k} \frac{1}{\mu_k - \mu_j}(B^{k,j})^2.
\end{aligned}$$

This agrees with the result in [6, p. 92].

We conclude with the following natural conjecture.

CONJECTURE 5.1. *A spectral function $f \circ \lambda$ is k-times differentiable at the matrix A if and only if its corresponding symmetric function f is k-times differentiable at the point $\lambda(A)$. Moreover, $f \circ \lambda$ is $\mathcal{C}^k$ if and only if f is $\mathcal{C}^k$.*

REFERENCES

[1] R. BHATIA, *Matrix Analysis,* Springer-Verlag, New York, 1997.

[2] R. BHATIA AND K. B. SINHA, *Derivations, derivatives and chain rules*, Linear Algebra Appl., 302-303 (1999), pp. 231–244.

[3] J.-B. HIRIART-URRUTY AND D. YE, *Sensitivity analysis of all eigenvalues of a symmetric matrix*, Numer. Math., 70 (1992), pp. 45–72.

[4] R.A. HORN AND C.R. JOHNSON, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, 1985.

[5] R.A. HORN AND C.R. JOHNSON, *Topics in Matrix Analysis*, 1st ed., Cambridge University Press, Cambridge, 1991; paperback edition with corrections, 1994.

[6] T. KATO, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, 1976.

[7] E.C. KEMBLE, *The Fundamental Principles of Quantum Mechanics*, Dover, New York, 1958.

[8] A.S. LEWIS, *Convex analysis on the Hermitian matrices*, SIAM J. Optim., 6 (1996), pp. 164–177.

[9] A.S. LEWIS, *Derivatives of spectral functions*, Math. Oper. Res., 21 (1996), pp. 576–588.

[10] A.S. LEWIS, *Nonsmooth analysis of eigenvalues*, Math. Programming, 84 (1999), pp. 1–24.

[11] A.S. LEWIS AND M.L. OVERTON, *Eigenvalue optimization*, Acta Numer., 5 (1996), pp. 149–190.

[12] A.S. LEWIS AND H.S. SENDOV, *Quadratic expansions of spectral functions*, Technical Report CORR 2000-41, University of Waterloo, 2000; Linear Algebra Appl., submitted.

[13] Y. NESTEROV AND A. NEMIROVSKII, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[14] F. OUSTRY, *The $\mathcal{U}$-Lagrangian of the maximum eigenvalue function*, SIAM J. Optim., 9 (1999), pp. 526–549.

[15] M.L. OVERTON, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[16] M.L. OVERTON, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[17] M.L. OVERTON AND R.S. WOMERSLEY, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 667–718.

[18] M.L. OVERTON AND X. YE, *Towards second-order methods for structured nonsmooth optimization*, in Advances in Optimization and Numerical Analysis, S. Gomez and J.-P. Hennart, eds., Kluwer, Amsterdam, 1994, pp. 97–109.

[19] R.T. ROCKAFELLAR AND R.J.-B. WETS, *Variational Analysis*, Springer-Verlag, Berlin, 1996.

[20] L.I. SCHIFF, *Quantum Mechanics*, McGraw-Hill, New York, 1955.

[21] A. SHAPIRO, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming, 77 (1997), pp. 301–320.

[22] A. SHAPIRO AND M.K.H. FAN, *On eigenvalue optimization*, SIAM J. Optim., 5 (1995), pp. 552–569.

[23] M. TORKI, *First- and second-order epi-differentiability in eigenvalue optimization*, J. Math. Anal. Appl., 234 (1999), pp. 391–416.

[24] M. TORKI, *Valeurs propres de matrices symétriques: Analyses de la sensibilité d'ordre supérieur et formulations variationelles pour leur estimation*, Ph.D. thesis, Paul Sabatier University, Toulouse, France, 1999.

[25] M. TORKI, *Second-order directional derivatives of all eigenvalues of a symmetric matrix*, Nonlinear Anal., to appear.

[26] N.-K. TSING, M.K.H. FAN, AND E.I. VERRIEST, *On analyticity of functions involving eigenvalues*, Linear Algebra Appl., 207 (1994), pp. 159–180.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 387–424
© 2001 Society for Industrial and Applied Mathematics

PERTURBATION ANALYSIS OF HAMILTONIAN SCHUR AND BLOCK-SCHUR FORMS*

M. KONSTANTINOV†, V. MEHRMANN‡, AND P. PETKOV§

Abstract. In this paper we present a complete perturbation analysis for the Hamiltonian Schur form of a Hamiltonian matrix under similarity transformations with unitary symplectic matrices. Both linear asymptotic and nonlinear perturbation bounds are presented. The same analysis is also carried out for two less condensed block-Schur forms. It suggests that the block forms are less sensitive to perturbations. The analysis is based on the technique of splitting operators and Lyapunov majorants as well as on a representation of the symplectic unitary group which is convenient for perturbation analysis of condensed forms. As a corollary, a perturbation bound for the stable invariant subspace of Hamiltonian matrices is obtained. Finally, given an ε-perturbation in the initial Hamiltonian matrix, the perturbations in the Hamiltonian Schur form, and the unitary symplectic basis are constructed in the form of a power series expansion in ε.

Key words. Hamiltonian Schur form, block-Schur form, Riccati equation, unitary symplectic group, perturbation analysis, splitting operators

AMS subject classifications. 15A21, 93B35, 93C73

PII. S0895479898342420

1. Introduction.

The computation of the eigenvalues and invariant subspaces (in particular the stable invariant subspace) of Hamiltonian matrices is an important problem in many applications such as linear quadratic optimal control and H_∞ control, as well as in the solution of continuous-time algebraic Riccati equations [20, 23, 30, 34].

It is known [26] that for a Hamiltonian matrix H with no eigenvalues on the imaginary axis there exists a unitary symplectic matrix U such that $\Sigma = U^{\mathrm{H}}HU = \left[\begin{smallmatrix} T & R \\ 0 & -T^{\mathrm{H}} \end{smallmatrix}\right]$, where R is Hermitian and T is upper triangular with all eigenvalues in the left half plane. A matrix of the form Σ is called a *Hamiltonian Schur form* of H. Under some further conditions (see [21, 22]), such a form also exists if the Hamiltonian matrix has eigenvalues on the imaginary axis.

The computation of the Hamiltonian Schur form is a highly structured problem and the analysis, as well as the corresponding numerical method, should reflect this structure in the maximal possible way. This is important for increasing the efficiency of the numerical method and the accuracy of the computed solution.

An open problem suggested in [26] was to construct a method that is backwards stable of complexity $O(n^3)$ and fully exploits the Hamiltonian structure. Many approaches have been made but, except for some important special cases [6], it is still an open problem to construct such a method. It was shown in [2] that here a fundamental difficulty occurs, in comparison with the transformation into standard Schur form.

*Received by the editors July 27, 1998; accepted for publication (in revised form) by J. Varah March 25, 2001; published electronically September 7, 2001. This work was supported by grant 5/72 764 of VW Stiftung.

http://www.siam.org/journals/simax/23-2/34242.html

†University of Architecture and Civil Engineering, 1 Hr. Smirnenski Blvd., 1421 Sofia, Bulgaria (fte@uacg.acad.bg).

‡Institut für Mathematik, MA 4-5, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, Germany (mehrmann@math.tu-berlin.de).

§Department of Automatics, Technical University of Sofia, 1756 Sofia, Bulgaria (php@mbox.digsys.bg).

This is due to the fact that the group of unitary symplectic $2n \times 2n$ matrices, being an algebraic variety of real dimension only $2n^2$, is much "smaller" than the group of unitary $2n \times 2n$ matrices, which is of real dimension $4n^2$. Several new approaches have been developed which partially (but not yet completely) overcome this difficulty; see [1, 3, 4].

An important issue, when considering structure-preserving algorithms for structured problems, is the analysis of the influence of perturbations that are also structured. This includes determining the sensitivity of the problem (finding condition estimates in particular) and studying the accuracy of the numerical methods. In the case of the Hamiltonian eigenvalue problem, one has to study the sensitivity of the Hamiltonian Schur form under the influence of Hamiltonian perturbations. In particular, it is important to analyze the sensitivity of the stable invariant subspace, which is often the desired object when computing the Hamiltonian Schur form. This is essential for the applications arising in linear quadratic and H_∞ control problems [23, 38].

The perturbation problem for eigensystems under general perturbations is well understood since the fundamental work of Stewart [35] (see also [36]) and some recent developments in [16, 29]. However, one may expect that the results will be different if the perturbations are structured, but a structured perturbation analysis for the Hamiltonian Schur form has not been carried out previously.

In this paper we call a bound *asymptotic* if it holds for perturbations in the data tending to zero, and we call a bound *nonlocal* if it holds for perturbations in the data which belong to a finite (although possibly small) domain.

We present a complete perturbation analysis for both the Hamiltonian Schur form and the corresponding unitary symplectic transformation matrix. We derive nonlocal nonlinear perturbation bounds based on the technique of splitting operators [16, 29] (see Appendix C) and analyze in detail asymptotic perturbation bounds.

The same analysis is done for other less condensed block-Schur forms of Hamiltonian matrices. An estimate for the sensitivity of the stable invariant subspace is obtained as a corollary. Finally, we construct power series expansions for the perturbed Hamiltonian Schur form as well as for the transformation matrix.

We use the following notation: $\mathcal{F}$ is the field of real or complex numbers, i.e., $\mathcal{F} = \mathcal{R}$ or $\mathcal{F} = \mathcal{C}$; $\mathcal{R}_+ = [0, \infty)$; $\imath = \sqrt{-1}$; $\mathcal{C}_-$ and $\mathcal{C}_+$ are the open left and right half of the complex plane $\mathcal{C}$; $\mathcal{F}^{m\times n}$ is the set of $m \times n$ matrices over $\mathcal{F}$, $\mathcal{F}^{n\times 1} = \mathcal{F}^n$; $\|.\|_2$ and $\|.\|_{\mathrm{F}}$ are the spectral and Frobenius norms in $\mathcal{F}^{m\times n}$ (we use $\|.\|$ for the Euclidean norm of vectors as well as for matrices when the particular norm is not specified); $\mathrm{Range}(A)$ is the range of the matrix A; $\mathrm{Inv}(A)$ is an invariant subspace of the matrix A; $\mathrm{gap}(M, N)$ is the gap between the subspaces M and N; $\mathrm{spect}(A)$ is the collection of n eigenvalues of the matrix $A \in \mathcal{C}^{n\times n}$, counted according to their algebraic multiplicities; $\mathrm{spect}_-(A)$, $\mathrm{spect}_+(A)$ are the collection of eigenvalues of A in the open left or right half plane, respectively; $\mathrm{rad}(A)$ is the spectral radius of A; I_n is the $n \times n$ identity matrix; $e_k \in \mathcal{R}^{n\times 1}$ is the kth column of I_n; $0_{m\times n}$ is the zero $m \times n$ matrix (if the size is clear from the context we write 0 for the zero matrix); $\mathbf{vec}(A) \in \mathcal{F}^{mn}$ is the columnwise vector representation of the matrix $A \in \mathcal{F}^{m\times n}$; $\Pi_n \in \mathcal{R}^{n^2\times n^2}$ is the vec-permutation matrix, such that $\mathbf{vec}(X^\top) = \Pi_n \mathbf{vec}(X)$ for $X \in \mathcal{C}^{n\times n}$; $A^\top$ and A^{H} are the transpose and complex conjugate transpose of A; $A \otimes B = [a_{ij}B]$ is the Kronecker product of the matrices $A = [a_{ij}]$ and B. If $X, Y \in \mathcal{F}^{n\times n}$, we set $\mathcal{U}[X, Y] = \left[\begin{smallmatrix} X & Y \\ -Y & X \end{smallmatrix}\right] \in \mathcal{F}^{2n\times 2n}$.

We also use the following notation for different sets of square matrices:

- $\mathbf{GL}(n) \subset \mathcal{C}^{n\times n}$ – the group of nonsingular matrices;
- $\mathbf{U}(n) \subset \mathbf{GL}(n)$ – the group of unitary matrices V, $VV^{\mathrm{H}} = I_n$;
- $\mathbf{Her}(n) \subset \mathcal{C}^{n\times n}$ – the set of Hermitian matrices $B = B^{\mathrm{H}}$;
- $\mathbf{T}(n) \subset \mathcal{C}^{n\times n}$ – the set of upper triangular matrices;
- $\mathbf{S}(2n) \subset \mathbf{GL}(2n)$ – the group of symplectic matrices U, $U^{\mathrm{H}} J_{2n} U = J_{2n}$, where $J_{2n} = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$;
- $\mathbf{US}(2n) = \mathbf{U}(2n) \cap \mathbf{S}(2n)$ – the group of unitary symplectic matrices;
- $\mathbf{Ham}(2n) \subset \mathcal{C}^{2n\times 2n}$ – the set of Hamiltonian matrices H, $J_{2n}H = (J_{2n}H)^{\mathrm{H}}$;
- $\mathbf{Ham}_0(2n) \subset \mathbf{Ham}(2n)$ – the set of Hamiltonian matrices with no eigenvalues on the imaginary axis.

The projectors in $\mathcal{C}^{n\times n}$ onto the strictly lower triangular, diagonal, and strictly upper triangular parts of a matrix are denoted by **low**, **diag**, and **up**, respectively. With this notation we have $\mathbf{up}(X) = (\mathbf{low}(X^{\top}))^{\top}$, $\mathbf{T}(n) = \mathrm{Range}(\mathbf{diag} + \mathbf{up})$, and

$$\mathbf{low}(X) = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} e_i e_i^{\top} X e_j e_j^{\top}, \qquad \mathbf{diag}(X) = \sum_{i=1}^{n} e_i e_i^{\top} X e_i e_i^{\top}.$$

The compressed vectorizations of the operations **low**, **diag**, and **up** are denoted by $\mathbf{lvec} : \mathcal{C}^{n\times n} \to \mathcal{C}^{\ell}$, $\mathbf{dvec} : \mathcal{C}^{n\times n} \to \mathcal{C}^{n}$, and $\mathbf{uvec} : \mathcal{C}^{n\times n} \to \mathcal{C}^{\ell}$, where $\ell = n(n-1)/2$. Thus for $X = [x_{ij}] \in \mathcal{C}^{n\times n}$ we have

$$\begin{aligned} \mathbf{lvec}(X) &= [x_{2,1}, \ldots, x_{n,1}, x_{3,2}, \ldots, x_{n,2}, \ldots, x_{n,n-1}]^{\top} = \Omega\, \mathbf{vec}(X) \in \mathcal{C}^{\ell}, \\ \mathbf{dvec}(X) &= [x_{1,1}, \ldots, x_{n,n}]^{\top} = \theta\, \mathbf{vec}(X) \in \mathcal{C}^{n}, \\ \mathbf{uvec}(X) &= [x_{1,2}, \ldots, x_{1,n}, x_{2,3}, \ldots, x_{2,n}, \ldots, x_{n-1,n}]^{\top} = \Omega\Pi_n \mathbf{vec}(X) \in \mathcal{C}^{\ell}, \end{aligned} \tag{1.1}$$

where

$$\begin{aligned} \Omega &:= [\mathrm{diag}(\Omega_1, \ldots, \Omega_{n-1}), 0_{\ell\times n}] \in \mathcal{R}^{\ell\times n^2}, \\ \theta &:= \mathrm{diag}\left(e_1^{\top}, \ldots, e_n^{\top}\right) \in \mathcal{R}^{n\times n^2}, \\ \Omega_k &:= [0_{(n-k)\times k}, I_{n-k}] \in \mathcal{R}^{(n-k)\times n}. \end{aligned}$$

The abbreviation ":=" stands for "equal by definition." We note finally that $\overline{A}$ is not used for the complex conjugate of A.

2. Condensed forms for Hamiltonian matrices. In this section we consider three condensed forms for Hamiltonian $2n \times 2n$ matrices under the similarity action of the unitary symplectic and unitary transformation groups $\mathbf{US}(2n)$ and $\mathbf{U}(n)$.

Consider the Hamiltonian matrix

$$H := \begin{bmatrix} A & B \\ C & -A^{\mathrm{H}} \end{bmatrix} \in \mathbf{Ham}(2n); \quad B, C \in \mathbf{Her}(n). \tag{2.1}$$

If $H \in \mathbf{Ham}_0(2n)$, then the spectrum of H splits in two parts, $\mathrm{spect}(H) = \mathrm{spect}_-(H) \cup \mathrm{spect}_+(H)$, where $\mathrm{spect}_-(H) \subset \mathcal{C}_-$ and $\mathrm{spect}_+(H) \subset \mathcal{C}_+$. The elements of $\mathrm{spect}_-(H)$ are called the *stable* eigenvalues of H, and the associated invariant subspace is the *stable* H-invariant subspace. It was shown in [26] that in this case there exists a matrix

$$U = \mathcal{U}[V, W] := \begin{bmatrix} V & W \\ -W & V \end{bmatrix} \in \mathbf{US}(2n); \quad V, W \in \mathcal{C}^{n\times n}$$

such that

$$\Sigma := U^{\mathrm{H}} H U = \begin{bmatrix} T & R \\ 0 & -T^{\mathrm{H}} \end{bmatrix} \in \mathbf{Ham}(2n); \quad T \in \mathbf{T}(n), R \in \mathbf{Her}(n), \tag{2.2}$$

where $\mathrm{spect}(T) = \mathrm{spect}\_(H)$. (For properties of unitary symplectic matrices $\mathcal{U}[V, W]$ see Appendix A.) An analogous form may also exist if H has purely imaginary eigenvalues; see [22]. A complete analysis of this case is not yet available. The uniqueness of the stable invariant subspaces for this case has been studied in [24], and a partial analysis on the stability of such subspaces under perturbations follows from the results in [31]. In general the above Hamiltonian Schur forms are not unique unless additional restrictions on the blocks of Σ are imposed.

DEFINITION 2.1. *A matrix Σ as in* (2.2)*, where T has no eigenvalues in $\mathcal{C}_+$, is called a* Hamiltonian Schur form *of H under the action of $\mathbf{US}(2n)$. The columns of U form a* symplectic Schur basis *of H. The set of all matrices of the form* (2.2) *is called the* set of Hamiltonian Schur forms *of $\mathbf{Ham}(2n)$ under the action of $\mathbf{US}(2n)$ and is denoted by $\mathbf{HSF}(2n)$.*

The *stabilizer* of $\Sigma \in \mathbf{HSF}(2n)$ relative to $\mathbf{US}(2n)$ is denoted by

$$\mathbf{Stab}(\Sigma, \mathbf{US}(2n)) := \{U \in \mathbf{US}(2n) : U^{\mathrm{H}} \Sigma U \in \mathbf{HSF}(2n)\}.$$

Similarly, for $T \in \mathbf{T}(n)$ we denote by $\mathbf{Stab}(T, \mathbf{U}(n)) \subset \mathbf{U}(n)$ the stabilizer of T relative to $\mathbf{U}(n)$.

PROPOSITION 2.2. *If $H \in \mathbf{Ham}_0(2n)$, then the stabilizer of $\Sigma \in \mathbf{HSF}(2n)$ as in* (2.2) *is*

$$\mathbf{Stab}(\Sigma, \mathbf{US}(2n)) = \{\mathrm{diag}(V, V) : \ V \in \mathbf{Stab}(T, \mathbf{U}(n))\}. \tag{2.3}$$

Proof. If $U = \mathrm{diag}(V, V)$ with $V \in \mathbf{Stab}(T, \mathbf{U}(n))$, then $U \in \mathbf{Stab}(\Sigma, \mathbf{US}(2n))$. Suppose now that $U = \mathcal{U}[V, W] \in \mathbf{Stab}(\Sigma, \mathbf{US}(2n))$ and let $\widetilde{T} \in \mathbf{T}(n)$ be the upper left block of $\widetilde{\Sigma} := U^{\mathrm{H}} \Sigma U \in \mathbf{HSF}(2n)$. It follows from $\Sigma U = U \widetilde{\Sigma}$ that W satisfies the Sylvester equation $T^{\mathrm{H}} W + W \widetilde{T} = 0$. Since both matrices T and $\widetilde{T}$ are stable it follows from the theory of Sylvester equations [8] that $W = 0$. ∎

Proposition 2.2 implies that $\mathbf{Stab}(\Sigma, \mathbf{US}(2n))$ is isomorphic to $\mathbf{Stab}(T, \mathbf{U}(n))$. In the generic case when H (and hence T) has n distinct stable eigenvalues, then except for similarity transformations with unitary diagonal matrices, the stabilizer $\mathbf{Stab}(\Sigma, \mathbf{US}(2n))$ contains $n(n-1)/2$ substantially different elements. They correspond to the matrices $V_{ij} \in \mathbf{Stab}(T, \mathbf{U}(n))$, whose similarity action interchanges the eigenvalues t_{ii} and t_{jj} of the matrix $T = [t_{ij}]$ along its diagonal.

The Hamiltonian Schur form (2.2) is only a condensed form and not a canonical one in the strict sense, unless some additional restrictions on T are imposed; see [24]. For a discussion on the standard Schur canonical form, see [33].

In many applications (e.g., in the computation of stabilizing solutions of algebraic Riccati equations) one is interested in the stable invariant subspace of H, and it suffices to have a condensed form like (2.2) relative to transformations from $\mathbf{US}(2n)$ but without the restriction $T \in \mathbf{T}(n)$. Therefore we also consider the less condensed *Hamiltonian block-Schur form*

$$\widehat{\Sigma} := \widehat{U}^{\mathrm{H}} H \widehat{U} = \begin{bmatrix} \widehat{T} & \widehat{R} \\ 0 & -\widehat{T}^{\mathrm{H}} \end{bmatrix} \in \mathbf{Ham}(2n), \tag{2.4}$$

relative to $\mathbf{US}(2n)$, where

$$\widehat{R} \in \mathbf{Her}(n), \quad \widehat{U} := \mathcal{U}[\widehat{V}, \widehat{W}] \in \mathbf{US}(2n)$$

and $\widehat{T}$ has no eigenvalues in $\mathcal{C}_+$. Note that if $H \in \mathbf{Ham}_0(2n)$, then $\mathrm{spect}(\widehat{T}) = \mathrm{spect}_-(H)$.

A Hamiltonian matrix may be transformed into Hamiltonian block-Schur form if and only if it may be transformed into Hamiltonian Schur form. Indeed, the Hamiltonian Schur form is also a Hamiltonian block-Schur form. In turn, if H admits a Hamiltonian block-Schur form $\widehat{\Sigma}$, then a further symplectic unitary transformation with a matrix $\mathrm{diag}(V, V)$ with $V \in \mathbf{U}(n)$, such that $V^{\mathrm{H}}\widehat{T}V \in \mathbf{T}(n)$, reduces the Hamiltonian block-Schur form $\widehat{\Sigma}$ into a Hamiltonian Schur form Σ.

We also consider the least condensed *block-Schur form*

$$\overline{\Sigma} := \overline{U}^{\mathrm{H}} H \overline{U} = \begin{bmatrix} \overline{T} & \overline{R} \\ 0 & \overline{S} \end{bmatrix} \in \mathcal{C}^{2n\times 2n}, \quad \overline{U} \in \mathbf{U}(2n), \tag{2.5}$$

of H relative to transformations from $\mathbf{U}(2n)$, where the matrix $\overline{T} \in \mathcal{C}^{n\times n}$ has no eigenvalues in $\mathcal{C}_+$. If $H \in \mathbf{Ham}_0(2n)$, then this implies $\mathrm{spect}(\overline{T}) = \mathrm{spect}_-(H)$, but it may happen that the matrix $\overline{\Sigma}$ is not Hamiltonian. Such a block-Schur form always exists as a consequence of Schur's theorem [9] and may be computed, for example, by the algorithm proposed in [28].

The standard Schur form of H relative to $\mathbf{U}(2n)$ is as $\overline{\Sigma}$ in (2.5) with the additional requirement $\overline{T}, \overline{S} \in \mathbf{T}(n)$. In this case the linear and nonlinear perturbation bounds from [16] are applicable.

As the following analysis suggests, the block-Schur forms $\widehat{\Sigma}$, $\overline{\Sigma}$ and the associated transformation matrices $\widehat{U}$, $\overline{U}$ may be much less sensitive to perturbations than the corresponding matrices Σ and U in the Hamiltonian Schur form. So if the structure of the upper left block in the condensed form is not important, it is preferable to work with the forms (2.4) or (2.5). Some numerical algorithms for computing the stable H-invariant subspace, however, lead to a Hamiltonian Schur form or to a standard Schur form rather than to the forms (2.4) or (2.5). This is because the triangular form is usually needed for the eigenvalue ordering. An exception is the multishift-method from [1]. Also, methods based on the matrix sign function [7, 15] and some related methods implicitly compute block-Schur forms rather than the Hamiltonian Schur form.

3. Perturbation theory for condensed forms. In this section we formulate the basic problems in the perturbation analysis for the three condensed forms that we introduced in section 2.

3.1. Hamiltonian Schur form. Our main assumptions when studying the Hamiltonian Schur form (2.2) are the following:

A1 *The matrix H has no imaginary eigenvalues, i.e., $H \in \mathbf{Ham}_0(2n)$.*

A2 *The matrix H has distinct eigenvalues.*

Assumption A2 seems restrictive but in a sense is necessary. Indeed, if H has multiple eigenvalues, then the perturbations in the symplectic Schur basis U may be discontinuous functions of the perturbations in H (see Appendix B for a detailed analysis of this phenomenon).

We consider two types of structured perturbations. These are general Hamiltonian perturbations,

$$\delta H := \begin{bmatrix} \delta A & \delta B \\ \delta C & -\delta A^{\mathrm{H}} \end{bmatrix} \in \mathbf{Ham}(2n) \tag{3.1}$$

with $\varepsilon := \|\delta H\|_{\mathrm{F}}$, and one-parameter families of perturbations,

$$\delta H = \varepsilon H_1 := \varepsilon \begin{bmatrix} A_1 & B_1 \\ C_1 & -A_1^{\mathrm{H}} \end{bmatrix} \in \mathbf{Ham}(2n). \tag{3.2}$$

Here $\varepsilon \geq 0$ is a (usually small) parameter. In the latter case we assume that $\|H_1\|_{\mathrm{F}} = 1$ so that the F-norm of εH_1 is ε. Families of one-parameter matrix perturbations are considered in more detail in Appendix B. Note that we do not require that assumptions A1 and/or A2 hold for the perturbed matrix $\widetilde{H} := H + \delta H$.

For matrices $H \in \mathbf{Ham}_0(2n)$ the Hamiltonian Schur form exists. If we require $\widetilde{H} := H + \delta H$ also to be in $\mathbf{Ham}_0(2n)$, this would imply restrictions on the norm ε of δH as follows. The quantity

$$d_0(H) := \min\{\|G\|_{\mathrm{F}} : G \in \mathbf{Ham}(2n),\ H + G \notin \mathbf{Ham}_0(2n)\}$$

may be interpreted as the *distance* from H to the set of Hamiltonian matrices with eigenvalues on the imaginary axis. If $\varepsilon < d_0(H)$, then $\widetilde{H} \in \mathbf{Ham}_0(2n)$ and there exists a Hamiltonian Schur form $\widetilde{\Sigma}$ of $\widetilde{H}$. However, the Hamiltonian Schur form may also exist when $\varepsilon \geq d_0(H)$ and $\widetilde{H}$ has imaginary eigenvalues. In the following analysis we determine a quantity $\varepsilon_0 > 0$ such that the Hamiltonian Schur form $\widetilde{\Sigma}$ of $\widetilde{H}$ exists provided that $\varepsilon \leq \varepsilon_0$. Thus we avoid the computation (or estimation) of $d_0(H)$. Whether our ε_0 is smaller than $d_0(H)$ remains an open question.

Suppose that the Hamiltonian Schur form $\widetilde{\Sigma}$ of $\widetilde{H}$ exists and denote by

$$\widetilde{U} := U + \delta U = \begin{bmatrix} V + \delta V & W + \delta W \\ -W - \delta W & V + \delta V \end{bmatrix} \in \mathbf{US}(2n) \tag{3.3}$$

the corresponding transformation matrix. Then the matrix $\widetilde{\Sigma} = \widetilde{U}^{\mathrm{H}} \widetilde{H} \widetilde{U}$ is

$$\widetilde{\Sigma} := \Sigma + \delta\Sigma = \begin{bmatrix} T + \delta T & R + \delta R \\ 0 & -(T + \delta T)^{\mathrm{H}} \end{bmatrix} \in \mathbf{HSF}(2n), \tag{3.4}$$

where $\mathrm{spect}(T + \delta T) \subset \mathcal{C}_-$ provided that $\widetilde{H} \in \mathbf{Ham}_0(2n)$. If we set

$$Z := U^{\mathrm{H}} \delta U := \mathcal{U}[X, Y] = \begin{bmatrix} X & Y \\ -Y & X \end{bmatrix};\ X, Y \in \mathcal{C}^{n \times n}, \tag{3.5}$$

then

$$I_{2n} + Z = U^{\mathrm{H}} \widetilde{U} \in \mathbf{U}(2n) \tag{3.6}$$

and $\delta V = VX - WY$, $\delta W = WX + VY$. By (3.6) we obtain $Z + Z^{\mathrm{H}} + ZZ^{\mathrm{H}} = 0$, and hence we have the conditions for unitarity

$$X + X^{\mathrm{H}} + XX^{\mathrm{H}} + YY^{\mathrm{H}} = 0, \qquad (I_n + X)Y^{\mathrm{H}} = Y(I_n + X^{\mathrm{H}}). \tag{3.7}$$

If $\widetilde{H} \in \mathbf{Ham}_0(2n)$, then the columns of the matrix $\left[\begin{smallmatrix} V+\delta V \\ -W-\delta W \end{smallmatrix}\right]$ span the stable $\widetilde{H}$-invariant subspace $\mathrm{Inv}_-(\widetilde{H})$.

In what follows we determine a constant $\varepsilon_0 > 0$ such that the perturbed matrix $\widetilde{H}$ has a Hamiltonian Schur form provided that $\varepsilon \le \varepsilon_0$. Then we derive estimates for $\|X\|_{\mathrm{F}}$ and $\|Y\|_{\mathrm{F}}$ of the form

$$\|Y\|_{\mathrm{F}} \le p(\varepsilon), \qquad \|X\|_{\mathrm{F}} \le q(\varepsilon), \tag{3.8}$$

where $\varepsilon \in [0, \varepsilon_0]$ and $p, q : [0, \varepsilon_0] \to \mathcal{R}_+$ are continuous nondecreasing functions such that $p(0) = q(0) = 0$. Using the bounds p and q we obtain bounds for $\|\delta U\|_{\mathrm{F}}$. Finally $\|\delta\Sigma\|_{\mathrm{F}}$ is estimated via q and ε.

Due to the nonuniqueness of the Hamiltonian Schur form, the estimates (3.8) are not valid for all transformation matrices $\widetilde{U}$. They rather hold true for at least one $\widetilde{U}$ which transforms $\widetilde{H}$ to Hamiltonian Schur form $\widetilde{\Sigma}$. This is a common situation in perturbation problems with nonunique solution.

To be precise, let us introduce the concept of a minimal perturbation. Let δH be a fixed small perturbation so that $\widetilde{H} = H + \delta H$ has a Hamiltonian Schur form. Consider the set $\mathbf{M} \subset \mathcal{C}^{2n\times 2n}$ of all δU such that $\widetilde{U} = U + \delta U \in \mathbf{U}(2n)$ and $\widetilde{U}^{\mathrm{H}}\widetilde{H}\widetilde{U} \in \mathbf{HSF}(2n)$. We are interested in those perturbations δU which are small together with δH. It is necessary to restrict ourselves to these perturbations, because $\mathbf{M}$ always contains elements of large norm. Indeed, if $\delta U \in \mathbf{M}$ is small, then the matrix $-2U - \delta U$ is also in $\mathbf{M}$ but has an F-norm close to $2\sqrt{n}$.

Since $\mathbf{M}$ is a compact set, there exists $\delta U_0 \in \mathbf{M}$ such that

$$\|\delta U_0\|_{\mathrm{F}} = \min\{\|\delta U\|_{\mathrm{F}} : \delta U \in \mathbf{M}\}.$$

The perturbations δU in U and $\delta\Sigma = \widetilde{U}^{\mathrm{H}}\widetilde{H}\widetilde{U} - U^{\mathrm{H}}HU$ in Σ are said to be *minimal* if $\|\delta U\|_{\mathrm{F}} = \|\delta U_0\|_{\mathrm{F}}$. In view of this, the bounds (3.8) are valid for minimal perturbations.

3.2. Hamiltonian block-Schur form and stable invariant subspaces. For the Hamiltonian block-Schur form (2.4) we have an analogous perturbation problem. In this case we require only assumption A1 to be fulfilled. (Assumption A2 is not needed here since the matrix $\widetilde{T}$ in (2.4) is not necessarily upper triangular.) Since the Hamiltonian Schur and Hamiltonian block-Schur forms of H exist simultaneously, the inequality $\varepsilon < \varepsilon_0$ (see section 3.1) guarantees that the Hamiltonian block-Schur form of $\widetilde{H} = H + \delta H$ also exists. In the perturbation analysis of the Hamiltonian block-Schur form, presented below, we find a quantity $\widehat{\varepsilon}_0$ for which this form exists provided $\varepsilon \le \widehat{\varepsilon}_0$. In this case the corresponding perturbation bounds of type (3.8) are valid for $\varepsilon \in [0, \widehat{\varepsilon}_0]$.

Suppose that the Hamiltonian block-Schur form $\widehat{\Sigma}$ of H exists. Denote by

$$\widehat{U} + \delta\widehat{U} := \begin{bmatrix} \widehat{V} + \delta\widehat{V} & \widehat{W} + \delta\widehat{W} \\ -\widehat{W} - \delta\widehat{W} & \widehat{V} + \delta\widehat{V} \end{bmatrix} \in \mathbf{US}(2n)$$

the transformation matrix such that

$$\widehat{\Sigma} + \delta\widehat{\Sigma} := (\widehat{U} + \delta\widehat{U})^{\mathrm{H}}\widetilde{H}(\widehat{U} + \delta\widehat{U}) = \begin{bmatrix} \widehat{T} + \delta\widehat{T} & \widehat{R} + \delta\widehat{R} \\ 0 & -(\widehat{T} + \delta\widehat{T})^{\mathrm{H}} \end{bmatrix}, \quad \delta\widehat{R} \in \mathbf{Her}(n),$$

where $\mathrm{spect}(\widehat{T} + \delta\widehat{T}) \cap \mathcal{C}_+ = \emptyset$. If we set $\widehat{Z} := \widehat{U}^{\mathrm{H}}\delta\widehat{U} := \mathcal{U}[\widehat{X}, \widehat{Y}]$, then $\widehat{X}$ and $\widehat{Y}$ satisfy the conditions (3.7).

Suppose that $\widetilde{H} \in \mathbf{Ham}_0(2n)$. Then the perturbation analysis for the Hamiltonian block-Schur form (2.4) also gives estimates for the gap (see [36])

$$\gamma := \mathrm{gap}(\mathrm{Inv}_-(H), \mathrm{Inv}_-(\widetilde{H}))$$

between the stable H-invariant subspace $\mathrm{Inv}_-(H)$ and the stable $\widetilde{H}$-invariant subspace $\mathrm{Inv}_-(\widetilde{H})$. The gap γ is the smallest distance from a vector in $\mathrm{Inv}_-(H)$ of unit length to its projection onto $\mathrm{Inv}_-(\widetilde{H})$. Since

$$\begin{aligned}\gamma &= \mathrm{gap}(\widehat{U}^{\mathrm{H}}\mathrm{Inv}_-(H), \widehat{U}^{\mathrm{H}}\mathrm{Inv}_-(\widetilde{H})) \\ &= \mathrm{gap}\left(\mathrm{Range}\begin{bmatrix} I_n \\ 0 \end{bmatrix}, (I_{2n}+\widehat{Z})\mathrm{Range}\begin{bmatrix} I_n \\ 0 \end{bmatrix}\right)\end{aligned}$$

and $\widehat{X}$, $\widehat{Y}$ satisfy (3.7), we get $\gamma = \|\Theta\|_2$, where

$$\Theta := \begin{bmatrix} \Theta_1 & \Theta_2 \\ \Theta_2 & -\Theta_1 \end{bmatrix}; \quad \Theta_1 := \widehat{Y}\widehat{Y}^{\mathrm{H}},\ \Theta_2 := (I_n+\widehat{X})\widehat{Y}^{\mathrm{H}}.$$

Using (3.7) we obtain $\Theta^2 = \Theta\Theta^{\mathrm{H}} = \mathrm{diag}\left(\Theta_1^2+\Theta_2^2, \Theta_1^2+\Theta_2^2\right)$. Hence,

$$\|\Theta\|_2 = \sqrt{\|\Theta_1^2+\Theta_2^2\|_2} \le \sqrt{\|\Theta_1\|_2^2+\|\Theta_2\|_2^2}$$

and, since $\|\Theta_1\|_2 = \|\widehat{Y}\|_2^2$, it follows from $(I_n+\widehat{X})(I_n+\widehat{X}^{\mathrm{H}}) = I_n - \widehat{Y}\widehat{Y}^{\mathrm{H}}$ that

$$\|I_n+\widehat{X}\|_2 = \sqrt{\left\|I_n-\widehat{Y}\widehat{Y}^{\mathrm{H}}\right\|_2} = \sqrt{1-\sigma_{\min}^2(\widehat{Y})},$$

where $\sigma_{\min}(\widehat{Y})$ is the minimal singular value of $\widehat{Y}$. Hence,

$$\|\Theta_2\|_2 \le \|\widehat{Y}\|_2\sqrt{1-\sigma_{\min}^2(\widehat{Y})}$$

and

$$\gamma = \|\Theta\|_2 \le \|\widehat{Y}\|_2\sqrt{1+\|\widehat{Y}\|_2^2-\sigma_{\min}^2(\widehat{Y})} \le \|\widehat{Y}\|_2\sqrt{1+\|\widehat{Y}\|_2^2}. \tag{3.9}$$

We see that the gap between the stable invariant subspaces of H and $\widetilde{H}$ is bounded from above by a quantity, which depends only on $\widehat{Y}$ and is of asymptotic order $\|\widehat{Y}\|_2$ for small $\|\widehat{Y}\|_2$.

It follows from this analysis that the sensitivity of the symplectic Schur basis U in the Hamiltonian Schur form and the sensitivity of the stable H-invariant subspace $\mathrm{Inv}_-(H)$ to perturbations $H \to H+\delta H$ may be different. The reason is that for small ε the norm of X (and hence the norm of δU) may not be small, while at the same time $\|\widehat{Y}\|_2$, which governs the gap γ, remains small.

The symplectic Schur basis for the Hamiltonian Schur form may be sensitive to perturbations if H has any close eigenvalues. Thus the Hamiltonian Schur form Σ may not be relevant for the investigation of the sensitivity of the stable H-invariant subspace. To study the sensitivity of this subspace we need to use the sensitivity estimates for the Hamiltonian block-Schur form (2.4) or the block-Schur form (2.5).

3.3. Block-Schur form. In this subsection we shall need some facts about finite *collections* $\lambda := \{\lambda_1, \ldots, \lambda_n\}$, i.e., sets with possibly repeated elements, which are very useful in describing and analyzing matrix spectra. Note that from a set-theoretical point of view a collection is indistinguishable from the set of its disjoint elements.

Let $l = \{l_1, \ldots, l_m\}$ be the set of disjoint elements in λ. Then λ may be represented by k pairs $(l_1, k_1), \ldots, (l_m, k_m)$, where k_i is the number of l_i's in λ.

The following operations with finite collections may be introduced. The empty collection $\emptyset$ is the standard empty set. The number of elements in the collection λ is denoted by $\#\lambda$. The collection λ' is a subcollection of λ, denoted as $\lambda' \subset \lambda$, if for each pair (l'_i, k'_i), associated with λ', there is a pair (l_i, k_i), associated with λ and such that $l_i = l'_i$ and $k_i \geq k'_i$. The union $\lambda \vee \lambda'$ of two collections λ and λ' is the collection containing all elements from λ and λ'. The intersection $\lambda \wedge \lambda'$ of λ and λ' is the collection containing the joint elements $\lambda_i = \lambda'_i$ from λ and λ', each one taken with multiplicity $\min\{k_i, k'_i\}$. Note that the union and intersection of two collections are different from the corresponding operations for sets.

The Hamiltonian matrix H is similar to $-H^{\mathrm{H}}$. Hence, the spectrum $\mathrm{spect}(H)$ of H may be represented as the union $S_- \vee S_0 \vee S_+$ of three disjoint collections, where the elements of S_- (if any) are from $\mathcal{C}_-$, the collection S_+ is symmetric to S_- relative to the imaginary axis, and the elements of S_0 (if any) are purely imaginary.

The perturbation analysis of the block-Schur form (2.5) is done under the following assumption.

A3 *The spectrum* $\mathrm{spect}(H)$ *of* $H \in \mathbf{Ham}(2n)$ *may be represented as the union* $\Lambda_- \vee \Lambda_+$ *of two disjoint collections* Λ_- *and* Λ_+ *such that* $\#\Lambda_- = n$ *and* Λ_- *contains no elements from* C_+.

Note that $S_- \subset \Lambda_-$, $S_+ \subset \Lambda_+$, and the collections Λ_- and Λ_+ may contain (an equal number of) imaginary elements.

In view of A3 we may always assume that the matrix $\overline{T}$ is chosen so that $\mathrm{spect}(\overline{T}) = \Lambda_-$. Thus the matrix $\overline{T}$ is stable if and only if $H \in \mathbf{Ham}_0(2n)$ (i.e., if and only if $S_0 = \emptyset$).

Whether assumption A3 holds for a particular Hamiltonian matrix H depends only on the imaginary part S_0 of the spectrum of H. For example, assumption A3 is fulfilled if H has no imaginary eigenvalues. At the same time A3 may be valid also if the imaginary part S_0 of the collection $\mathrm{spect}(H)$ is nonempty but has a certain special structure as described below.

Let $r := \#S_0 = 2(n - \#S_+)$. If the collection S_0 is not empty, then we have $r \geq 2$ and $S_0 = \{\imath\alpha_1, \imath\alpha_2, \ldots, \imath\alpha_r\}$. Here $\alpha := \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is a real collection with each element participating an even number of times, e.g., $\alpha_1 = \alpha_2, \ldots, \alpha_{r-1} = \alpha_r$. Let the disjoint elements of α be $l_1, \ldots, l_m$ with multiplicities $k_1 \leq \cdots \leq k_m$, respectively, where $k_1 + \cdots + k_m = r$. We have $m \leq r/2$ since k_i are positive even numbers.

Then assumption A3 is valid if and only if either $H \in \mathbf{Ham}_0(2n)$ (which is assumption A1) or the following holds.

A4 *The number* $r/2$ *is even and there is a positive integer* $p < m$ *such that* $k_1 + \cdots + k_p = r/2$.

Thus either A1 or A4 must hold in order to guarantee that we can create the block-Schur form in a specific way such that for every purely imaginary eigenvalue all of its multiplicity is in only one of the diagonal blocks.

Example 1. For $\alpha = \{1, 1, 0, 0\}$ we have $r = 4$, $m = 2$, $r_1 = r_2 = 2$ and assumption A4 holds with $p = 1$. For $\alpha = \{1, 1, -1, -1, 0, 0, 0, 0\}$ we have $r = 8$, $m = 3$, $r_1 = r_2 = 2$, $r_3 = 4$ and assumption A4 holds with $p = 2$. For $\alpha = \{1, 1, 1, 1\}$

we have $m = 1$, a positive integer $p < 1$ does not exist, and assumption A4 does not hold. For $\alpha = \{1, 1, -1, -1, 0, 0\}$ we have $r = 6$, the number $r/2 = 3$ is odd, and assumption A4 does not hold. For $\alpha = \{1, 1, 0, 0, 0, 0, 0, 0\}$ we have $r = 8$, $m = 2$, $r_1 = 2$, $r_2 = 6$ and assumption A4 does not hold.

Assumption A4 relaxes the restrictions on H. At the same time for the block-Schur form of a Hamiltonian matrix the transformation matrix $\overline{U}$ in general is not symplectic. Consider the following example.

Example 2. The matrix

$$H = \begin{bmatrix} 0 & K_2 \\ -K_2 & 0 \end{bmatrix} \in \mathbf{Ham}(4), \quad K_2 := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

has spectrum $\{\imath, \imath, -\imath, -\imath\}$ coinciding with S_0 with $r = 4$, $m = 2$, $k_1 = k_2 = 2$ and satisfies assumption A4. A block-Schur form of H is

$$\overline{\Sigma} = \overline{U}^{\mathrm{H}} H \overline{U} = \mathrm{diag}(\imath I_2, -\imath I_2),$$

where

$$\overline{U} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_2 & I_2 \\ \imath K_2 & -\imath K_2 \end{bmatrix} \in \mathbf{U}(4).$$

At the same time there exists no symplectic transformation to block-Schur form; see [22].

For the block-Schur form it is not necessary to consider Hamiltonian perturbations, since the Hamiltonian structure of H is not preserved under general unitary transformations. So in this case we assume that the perturbation in H is

$$\delta\overline{H} := \begin{bmatrix} \delta\overline{H}_{11} & \delta\overline{H}_{12} \\ \delta\overline{H}_{21} & \delta\overline{H}_{22} \end{bmatrix} \in \mathcal{C}^{2n \times 2n}.$$

Let

$$\overline{\Sigma} + \delta\overline{\Sigma} = (\overline{U} + \delta\overline{U})^{\mathrm{H}} (H + \delta\overline{H})(\overline{U} + \delta\overline{U}) = \begin{bmatrix} \overline{T} + \delta\overline{T} & \overline{R} + \delta\overline{R} \\ 0 & \overline{S} + \delta\overline{S} \end{bmatrix}$$

be the block-Schur form of the perturbed matrix $H + \delta\overline{H}$. Setting

$$\overline{Z} := \overline{U}^{\mathrm{H}} \delta\overline{U} := \begin{bmatrix} \overline{Z}_{11} & \overline{Z}_{12} \\ \overline{Z}_{21} & \overline{Z}_{22} \end{bmatrix}, \quad \overline{Z}_{ij} \in \mathcal{C}^{n \times n},$$

we have $I_{2n} + \overline{Z} \in \mathbf{U}(2n)$. Hence, for $i = 1, 2$ and $j \neq i$ we have that

$$\overline{Z}_{ii} + \overline{Z}_{ii}^{\mathrm{H}} + \overline{Z}_{ii}\overline{Z}_{ii}^{\mathrm{H}} + \overline{Z}_{ji}\overline{Z}_{ji}^{\mathrm{H}} = 0, \quad (I_n + \overline{Z}_{11})\overline{Z}_{21}^{\mathrm{H}} + \overline{Z}_{12}(I_n + \overline{Z}_{22}^{\mathrm{H}}) = 0. \tag{3.10}$$

In the following we derive nonlocal perturbation bounds for the F-norms of the matrices $\overline{Z}_{ij}$ and hence of $\overline{Z}$. In view of $\|\delta\overline{U}\|_{\mathrm{F}} = \|\overline{Z}\|_{\mathrm{F}}$ this gives the desired perturbation bounds for the Schur basis $\overline{U}$ and the block-Schur form $\overline{\Sigma}$ as functions of $\overline{\varepsilon} := \|\delta\overline{H}\|_{\mathrm{F}}$. These bounds are valid for $\overline{\varepsilon} \in [0, \overline{\varepsilon}_0]$, where $\overline{\varepsilon}_0$ is a positive constant.

4. Basic relations for perturbation analysis. In this section we derive the basic relations necessary for the perturbation analysis of the condensed forms, described in sections 2 and 3. We recall that the condensed forms are considered under the assumptions listed in the table below.

Condensed form	Assumptions
Hamiltonian Schur	A1 and A2
Hamiltonian block-Schur	A1
Block-Schur	A3

4.1. Hamiltonian Schur form. Consider the Hamiltonian matrix (2.1) and the Hamiltonian perturbation (3.1) such that the perturbed matrix $\widetilde{H} = H + \delta H$ admits a Hamiltonian Schur form. Recall that $\widetilde{U}$ in (3.3) is the matrix which transforms $\widetilde{H}$ into Hamiltonian Schur form (3.4). Set

$$E := \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} = U^{\mathrm{H}}\delta H U \in \mathbf{Ham}(2n),$$
$$F := \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = U^{\mathrm{H}}\delta H \widetilde{U} \in \mathcal{C}^{2n\times 2n},$$
$$G := \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \widetilde{U}^{\mathrm{H}}\delta H \widetilde{U} \in \mathbf{Ham}(2n),$$

where the matrices E, F, G are partitioned conformally with H. Then

$$\begin{aligned} &E_{11} = V^{\mathrm{H}}\delta A V - V^{\mathrm{H}}\delta B W - W^{\mathrm{H}}\delta C V - W^{\mathrm{H}}\delta A^{\mathrm{H}} W, \quad E_{22} = -E_{11}^{\mathrm{H}}, \\ &E_{21} = W^{\mathrm{H}}\delta A V - W^{\mathrm{H}}\delta B W + V^{\mathrm{H}}\delta C V + V^{\mathrm{H}}\delta A^{\mathrm{H}} W, \\ &E_{12} = V^{\mathrm{H}}\delta A W + V^{\mathrm{H}}\delta B V - W^{\mathrm{H}}\delta C W + W^{\mathrm{H}}\delta A^{\mathrm{H}} V, \end{aligned} \tag{4.1}$$

and since $\widetilde{H}\widetilde{U} = \widetilde{U}\widetilde{\Sigma}$, we have that

$$(\Sigma + E)(I_{2n} + Z) = (I_{2n} + Z)(\Sigma + \delta\Sigma), \tag{4.2}$$

where the matrix $Z = U^{\mathrm{H}}\delta U$ is defined by (3.5). We may rewrite (4.2) in two equivalent forms, namely,

$$\Sigma Z - Z\Sigma = (I_{2n} + Z)\delta\Sigma - F \tag{4.3}$$

and

$$\delta\Sigma = (I_{2n} + Z^{\mathrm{H}})(\Sigma Z - Z\Sigma) + G. \tag{4.4}$$

Equations (4.3), (4.4), and (3.7) are the basic relations that we use to determine the blocks X and Y in Z. From (4.3) and the fact that $\widehat{\Sigma} \in \mathbf{HSF}(2n)$ we obtain $\widetilde{\Sigma}_{11} = T + \delta T \in \mathbf{T}(n)$ (which is equivalent to $\delta T \in \mathbf{T}(n)$) and furthermore that $\widetilde{\Sigma}_{21} = 0$.

For the $(1,1)$ block in (4.3) we obtain

$$TX - XT = RY + (I_n + X)\delta T - F_{11}. \tag{4.5}$$

To show that $\widetilde{\Sigma}_{11} \in \mathbf{T}(n)$ we apply the **low** projector on both sides of (4.5), keeping in mind that $\mathbf{low}(\delta T) = 0$. We obtain

$$\mathbf{low}(TX - XT) = \mathbf{low}(RY + X\delta T - F_{11}). \tag{4.6}$$

The equation for the $(2,1)$ block in (4.3) is

$$\mathcal{L}(Y) := T^{\mathrm{H}}Y + YT = -Y\delta T - F_{21}. \tag{4.7}$$

Thus the equation for the matrix X is obtained by putting the $(1,1)$ block of $\widetilde{\Sigma}$ into upper triangular (Schur) form, while the equation for the matrix Y is derived by zeroing the $(2,1)$ block of $\widetilde{\Sigma}$.

The equation for the $(2,2)$ block in (4.3) yields

$$\mathbf{low}(TX^{\mathrm{H}} - X^{\mathrm{H}}T) = -\mathbf{low}\left((R+\delta R)Y^{\mathrm{H}} + \delta T(I_n + X^{\mathrm{H}}) + F_{22}^{\mathrm{H}}\right), \tag{4.8}$$

where in general $F_{22} \neq -F_{11}^{\mathrm{H}}$.

The $(1,2)$ block in (4.3) gives

$$RX - XR + TY + YT^{\mathrm{H}} = (I_n + X)\delta R - Y\delta T^{\mathrm{H}} - F_{12}. \tag{4.9}$$

We will also use the identity

$$\delta T = \left[I_n + X^{\mathrm{H}}, -Y^{\mathrm{H}}\right] \begin{bmatrix} TX - XT - RY \\ T^{\mathrm{H}}Y + YT \end{bmatrix} + G_{11} \tag{4.10}$$

for the $(1,1)$ block in (4.4).

An important observation from (4.6) and the upper triangular form of T is that the matrices $\mathbf{low}(TX)$, $\mathbf{low}(XT)$, and hence $\mathbf{low}(TX-XT)$ depend only on $\mathbf{low}(X)$; see [16]. Taking the **lvec** operation on both sides of (4.6) we obtain

$$\begin{aligned} M\,\mathbf{lvec}(X) &= \Omega\,\mathbf{vec}(RY + X\delta T - F_{11}) \\ &= \Omega(I_n \otimes R)\,\mathbf{vec}(Y) \\ &\quad + \Omega\,(\delta T^{\top} \otimes I_n)\,\Omega^{\top}\,\mathbf{lvec}(X) - \Omega\,\mathbf{vec}(F_{11}), \end{aligned} \tag{4.11}$$

where

$$M := \Omega(I_n \otimes T - T^{\top} \otimes I_n)\Omega^{\top} \in \mathcal{C}^{\ell\times\ell} \tag{4.12}$$

is the matrix of the linear operator $\mathbf{lvec}(X) \to \mathbf{lvec}(TX - XT)$ (see [16]), and Ω is as in (1.1). The eigenvalues of the matrix M are $\lambda_{ij} := t_{ii} - t_{jj}$, $i > j$. Hence, M is nonsingular if and only if the matrix $T = [t_{ij}]$ has distinct eigenvalues, which is the case according to assumption A2.

Example 3. For $n = 4$ the matrix M in (4.12) is

$$M = \begin{bmatrix} \lambda_{21} & t_{23} & t_{24} & 0 & 0 & 0 \\ 0 & \lambda_{31} & t_{34} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{41} & 0 & 0 & 0 \\ 0 & -t_{12} & 0 & \lambda_{32} & t_{34} & 0 \\ 0 & 0 & -t_{12} & 0 & \lambda_{42} & 0 \\ 0 & 0 & -t_{13} & 0 & -t_{23} & \lambda_{43} \end{bmatrix}.$$

4.2. Hamiltonian block-Schur form. If we perturb H to $H + \delta H$, then the matrices $\widehat{U}$, $\widehat{\Sigma}$, and $\widehat{T}$ are perturbed to $\widehat{U} + \delta\widehat{U} = \widehat{U}(I_{2n} + \widehat{Z})$, $\widehat{\Sigma} + \delta\widehat{\Sigma}$, and $\widehat{T} + \delta\widehat{T}$. Here the matrices $\widehat{T}$, $\delta\widehat{T}$, and $\widehat{T} + \delta\widehat{T}$ may have nonzero elements below the diagonal.

Hence, in this case we have only an equation of the form (4.7) and the unitarity conditions (3.7), i.e.,

$$\widehat{\mathcal{L}}(\widehat{Y}) := \widehat{T}^{\mathrm{H}}\widehat{Y} + \widehat{Y}\widehat{T} = -\widehat{Y}\delta\widehat{T} - \widehat{F}_{21}, \quad (I_n + \widehat{X})(I_n + \widehat{X}^{\mathrm{H}}) = I_n - \widehat{Y}\widehat{Y}^{\mathrm{H}}, \tag{4.13}$$

where $\widehat{E} := \widehat{U}^{\mathrm{H}}\delta H\widehat{U}$, $\widehat{F} := \widehat{E}(I_{2n} + \widehat{Z})$, $\widehat{G} := (I_{2n} + \widehat{Z}^{\mathrm{H}})\widehat{F}$. As we will show in Appendix A we may set

$$\widehat{X} = \left(I_n - \widehat{Y}\widehat{Y}^{\mathrm{H}}\right)^{\frac{1}{2}} P(\widehat{Y})Q^{\mathrm{H}}(\widehat{Y}) - I_n = P(\widehat{Y})\left(I_n - D^2(\widehat{Y})\right)Q^{\mathrm{H}}(\widehat{Y}) - I_n, \tag{4.14}$$

where $\widehat{Y} = P(\widehat{Y})D(\widehat{Y})Q^{\mathrm{H}}(\widehat{Y})$ is a decomposition of the form (A.5) in Appendix A.

4.3. Block-Schur form. If we consider the block-Schur form (2.5), then we obtain the following identities analogous to (4.2), (4.3), and (4.4):

$$\overline{\Sigma}\,\overline{Z} - \overline{Z}\,\overline{\Sigma} = (I_{2n} + \overline{Z})\delta\overline{Z} - \overline{F}, \quad \delta\overline{\Sigma} = (I_{2n} + \overline{Z}^{\mathrm{H}})(\overline{\Sigma}\,\overline{Z} - \overline{Z}\,\overline{\Sigma}) + \overline{G}, \tag{4.15}$$

where $\overline{E} := \overline{U}^{\mathrm{H}}\delta\overline{H}\,\overline{U}$, $\overline{F} := \overline{E}(I_{2n} + \overline{Z})$, $\overline{G} := (I_{2n} + \overline{Z}^{\mathrm{H}})\overline{F}$. Furthermore,

$$\overline{S}\,\overline{Z}_{21} - \overline{Z}_{21}\overline{T} = \overline{Z}_{21}\delta\overline{T} - \overline{F}_{21}, \tag{4.16}$$

analogous to (4.7) while the unitarity conditions are given by (3.10).

5. First order perturbation analysis. In this section we present a detailed first order (or asymptotic) perturbation analysis of the condensed forms for Hamiltonian matrices, giving the first order terms of the corresponding perturbation bounds. The first order approximations are used then in the next section to derive higher order approximations.

The asymptotic perturbation analysis produces relations of the form

$$\|\delta\Sigma\|_{\mathrm{F}} \le K\varepsilon + O(\varepsilon^2), \qquad \varepsilon \to 0, \tag{5.1}$$

where $\varepsilon = \|\delta H\|_{\mathrm{F}}$ and K is the *absolute condition number* of Σ relative to perturbations in H (for other types of first order bounds, see [19]). Since the $O(\varepsilon^2)$-term in (5.1) is usually not known, bounds of this type are applied in the chopped form $\|\delta\Sigma\|_{\mathrm{F}} \le K\varepsilon$, neglecting second and higher order terms. This must be done very carefully since the quantity $K\varepsilon$ may in fact underestimate the actual perturbation $\|\delta\Sigma\|_{\mathrm{F}}$. The underestimation may occur when, for example, the first partial derivatives of the mapping $\delta H \mapsto \|\delta\Sigma(\delta H)\|$ in the elements of δH are nonnegative and the Hessian of the same mapping is a positive definite matrix (if the Hessian exists and is a continuous function). Due to the lack of a general algorithm for computing the Hamiltonian Schur form in the case of eigenvalues on the imaginary axis, it is not an easy task to construct such an example. For the related problem of quadratic matrix equations, examples that show the underestimation of the perturbations by the linear bound are given in [17].

5.1. Hamiltonian Schur form. It may be shown (see Proposition 7.1) that for ε small enough the matrices Z in (3.5) and $\delta\Sigma$ in (3.4) are analytic functions in ε that vanish for $\varepsilon = 0$, i.e.,

$$Z = \sum_{k=1}^{\infty} \varepsilon^k Z_k, \qquad \delta\Sigma = \sum_{k=1}^{\infty} \varepsilon^k \Sigma_k = \sum_{k=1}^{\infty} \varepsilon^k \begin{bmatrix} T_k & R_k \\ 0 & -T_k^{\mathrm{H}} \end{bmatrix}, \tag{5.2}$$

where $Z_k := \mathcal{U}[X_k, Y_k]$ and $T_k \in \mathbf{T}(n)$, $R_k \in \mathbf{Her}(n)$. Hence, the first order perturbations $X_0 := \varepsilon X_1$ and $Y_0 := \varepsilon Y_1$ satisfy the basic equations

$$M\,\mathbf{lvec}(X_0) = \Omega((I_n \otimes R)\mathbf{vec}(Y_0) - \mathbf{vec}(E_{11})), \tag{5.3}$$

$$T^{\mathrm{H}} Y_0 + Y_0 T = -E_{21}, \tag{5.4}$$

and the first order approximations to conditions for unitarity (3.7) are

$$X_0 + X_0^{\mathrm{H}} = 0, \qquad Y_0 = Y_0^{\mathrm{H}}, \tag{5.5}$$

where M is given in (4.12) and $E_{21} \in \mathbf{Her}(n)$ according to (4.1).

We can solve (5.3)–(5.5) as follows. Equation (5.4) is a Hermitian Lyapunov equation with Lyapunov operator $\mathcal{L}$, defined by $\mathcal{L}(Y) := T^{\mathrm{H}} Y + YT$. Since T is a stable matrix, the operator $\mathcal{L}$ is invertible [8]. Thus (5.4) has a unique solution $Y_0 = -\mathcal{L}^{-1}(E_{21}) \in \mathbf{Her}(n)$, which satisfies the second equation in (5.5) as well. The matrix Y_0 is then substituted in (5.3), and the compressed lower part of X_0 becomes

$$\mathbf{lvec}(X_0) = M^{-1}\Omega((I_n \otimes R)\mathbf{vec}(Y_0) - \mathbf{vec}(E_{11})). \tag{5.6}$$

To satisfy the first equation in (5.5) we choose X_0 to be the following skew-Hermitian matrix with zero main diagonal (this minimizes $\|\delta U\|_{\mathrm{F}}$ in a first order approximation):

$$\mathbf{low}(X_0) = \mathbf{vec}^{-1}\left(\Omega^{\top}\mathbf{lvec}(X_0)\right), \qquad \mathbf{diag}(X_0) = 0, \qquad \mathbf{up}(X_0) = -\mathbf{low}^{\mathrm{H}}(X_0).$$

It is instructive to see how the first order terms in the equations for the $(2,2)$ and $(1,2)$ block in (4.3) look. Since $E_{22} = -E_{11}^{\mathrm{H}}$, the first order part of (4.8) is

$$\mathbf{low}(TX_0^{\mathrm{H}} - X_0^{\mathrm{H}} T) = -\mathbf{low}(RY_0^{\mathrm{H}} - E_{11}), \tag{5.7}$$

and, since $X_0^{\mathrm{H}} = -X_0$, we see that (5.7) is fulfilled. Equation (4.9) yields the following first order relation for $R_0 := \varepsilon R_1$:

$$R_0 = RX_0 - X_0 R + TY_0 + Y_0 T^{\mathrm{H}} + E_{12}.$$

Since $Y_0, E_{12}, R \in \mathbf{Her}(n)$ and X_0 is skew-Hermitian, it follows that necessarily $R_0 \in \mathbf{Her}(n)$.

In most applications only information on the norms of the perturbations in the data is available. For this reason we now derive first order bounds for the F-norms of X_0, Y_0 and δU, $\delta\Sigma$ in terms of the quantities

$$\varepsilon_{ij} := \|E_{ij}\|_{\mathrm{F}}, \qquad \varepsilon := \|\delta H\|_{\mathrm{F}} = \sqrt{2\varepsilon_{11}^2 + \varepsilon_{12}^2 + \varepsilon_{21}^2}; \tag{5.8}$$

see (4.1). We have $\|Y_0\|_{\mathrm{F}} \le l\varepsilon_{21}$, which together with (5.6) yields

$$\|X_0\|_{\mathrm{F}} = \sqrt{2}\,\|\mathbf{lvec}(X_0)\| \le \sqrt{2}\,(r\|Y_0\|_{\mathrm{F}} + m\varepsilon_{11}) \le \sqrt{2}\,(lr\varepsilon_{21} + m\varepsilon_{11}).$$

Here we have made the substitutions

$$l := \max_{Y \neq 0} \frac{\|\mathcal{L}^{-1}(Y)\|_{\mathrm{F}}}{\|Y\|_{\mathrm{F}}} = \|L^{-1}\|_2, \qquad L := I_n \otimes T^{\mathrm{H}} + T^{\top} \otimes I_n,$$

$$m := \left\|M^{-1}\Omega\right\|_2 = \left\|M^{-1}\right\|_2, \qquad r := \left\|M^{-1}\Omega(I_n \otimes R)\right\|_2. \tag{5.9}$$

The norm of δU, which is in first order approximation equal to the norm of $Z_0 := \mathcal{U}[X_0, Y_0]$, can now be estimated as

$$\begin{aligned} \|\delta U\|_{\mathrm{F}} &= \sqrt{2\left(\|X_0\|_{\mathrm{F}}^2 + \|Y_0\|_{\mathrm{F}}^2\right)} + O(\varepsilon^2) \\ &\leq \sqrt{4(lr\varepsilon_{21} + m\varepsilon_{11})^2 + 2(l\varepsilon_{21})^2} + O(\varepsilon^2). \end{aligned} \tag{5.10}$$

In turn, the norm of the perturbation $\delta\Sigma$ can be estimated via (4.4) and (5.10) as

$$\begin{aligned} \|\delta\Sigma\|_{\mathrm{F}} &\leq \|\Sigma Z_0 - Z_0\Sigma\|_{\mathrm{F}} + \varepsilon + O(\varepsilon^2) \leq s\|Z_0\|_{\mathrm{F}} + \varepsilon + O(\varepsilon^2) \\ &\leq s\sqrt{4(lr\varepsilon_{21} + m\varepsilon_{11})^2 + 2(l\varepsilon_{21})^2} + \varepsilon + O(\varepsilon^2), \end{aligned} \tag{5.11}$$

where

$$s := \left\| I_{2n} \otimes \Sigma - \Sigma^{\top} \otimes I_{2n} \right\|_2. \tag{5.12}$$

Estimates in terms of ε are obtained, taking into account that $2\varepsilon_{11}^2 + \varepsilon_{21}^2 \leq \varepsilon^2$. The maximum of the estimate $\sqrt{2}\,(lr\varepsilon_{21} + m\varepsilon_{11})$ for $\|X_0\|_{\mathrm{F}}$ under the constraint $2\varepsilon_{11}^2 + \varepsilon_{21}^2 = \varepsilon^2$ is obtained as follows. For a vector $b \in \mathcal{C}^n$ and a positive definite matrix $P \in \mathcal{C}^{n\times n}$ one has

$$\max\left\{|b^{\mathrm{H}}y| : x \in \mathcal{C}^n, y^{\mathrm{H}}Py = \varepsilon^2\right\} = \varepsilon\|P^{-1/2}c\|_2, \tag{5.13}$$

where $\varepsilon > 0$. Setting $y := [\varepsilon_{11}, \varepsilon_{21}]^{\top}$, $b := \sqrt{2}[m, lr]^{\top}$, and $P := \mathrm{diag}(2, 1)$ we get

$$\|X_0\|_{\mathrm{F}} \leq \varepsilon\|P^{-1/2}b\|_2 = \varepsilon\sqrt{m^2 + 2l^2r^2}.$$

Returning to the perturbations δU and $\delta\Sigma$, it follows from (5.10) and (5.11) that

$$\|\delta U\|_{\mathrm{F}} \leq \varepsilon K_U + O(\varepsilon^2), \tag{5.14}$$

$$\|\delta\Sigma\|_{\mathrm{F}} \leq \varepsilon K_\Sigma + O(\varepsilon^2),\ K_\Sigma := 1 + sK_U, \tag{5.15}$$

where the quantity K_U, depending on l, m, and r, is determined as

$$K_U := \max\left\{\sqrt{4(lre_{21} + me_{11})^2 + 2(le_{21})^2} : 2e_{11}^2 + e_{21}^2 = 1\right\},$$

with

$$e_{ij} := \frac{\varepsilon_{ij}}{\varepsilon}. \tag{5.16}$$

The explicit expression for K_U is obtained using the fact that for matrices $Q, P \in \mathbf{Her}(n)$ with Q nonnegative and P positive definite one has

$$\max\left\{y^{\mathrm{H}}Qy : x \in \mathcal{C}^n, y^{\mathrm{H}}Py = \varepsilon^2\right\} = \varepsilon\|P^{-1}Q\|_2. \tag{5.17}$$

For $Q := 2\left[\begin{smallmatrix} 2m^2 & 2lmr \\ 2lmr & l^2(1+2r^2) \end{smallmatrix}\right]$ and y, P as above it follows that $K_U = \sqrt{\|P^{-1}Q\|_2}$, i.e.,

$$K_U = \sqrt{m^2 + l^2(1 + 2r^2) + \sqrt{(m^2 - l^2(1 + 2r^2))^2 + 8l^2m^2r^2}}. \tag{5.18}$$

The quantities K_U and K_Σ are estimates for the absolute condition numbers for the symplectic Schur basis U and for the Hamiltonian Schur form Σ, respectively.

5.2. Hamiltonian block-Schur form. Consider the form (2.4). Using the notation

$$\widehat{\varepsilon}_{ij} := \|\widehat{E}_{ij}\|_{\mathrm{F}}, \qquad 2\widehat{\varepsilon}_{11}^2 + \widehat{\varepsilon}_{21}^2 + \widehat{\varepsilon}_{12}^2 = \varepsilon^2, \qquad \varepsilon = \|\widehat{E}\|_{\mathrm{F}} = \|\delta H\|_{\mathrm{F}},$$

(4.13) and (4.14) give $\widehat{\mathcal{L}}(\widehat{Y}_0) = -\widehat{E}_{21}$ and $\widehat{X}_0 = P(\widehat{Y}_0)Q^{\mathrm{H}}(\widehat{Y}_0) - I_n$, where $\widehat{Z}_0 := \mathcal{U}[\widehat{X}_0, \widehat{Y}_0]$ is the first order approximation to $\widehat{Z}$. Since $\widehat{E}_{21} \in \mathbf{Her}(n)$ we have $\widehat{Y}_0 = -\widehat{\mathcal{L}}^{-1}(\widehat{E}_{21}) \in \mathbf{Her}(n)$ and $P(\widehat{Y}_0) = Q(\widehat{Y}_0)$; see (A.6) from Appendix A. Hence, $\widehat{X}_0 = 0$.

Set $\widehat{L} := I_n \otimes \widehat{T}^{\mathrm{H}} + \widehat{T}^{\top} \otimes I_n$ and

$$\widehat{l} := \max_{Y \neq 0} \frac{\|\widehat{\mathcal{L}}^{-1}(Y)\|_{\mathrm{F}}}{\|Y\|_{\mathrm{F}}} = \|\widehat{L}^{-1}\|_2, \qquad \widehat{s} := \left\| I_{2n} \otimes \widehat{\Sigma} - \widehat{\Sigma}^{\top} \otimes I_{2n} \right\|_2^{-1}.$$

The matrices $\widehat{L}$, L and $\widehat{\Sigma}$, Σ, respectively, are unitarily similar, which implies that $\widehat{l} = l$ and $\widehat{s} = s$. Therefore $\|\widehat{Y}_0\|_{\mathrm{F}} \leq l\widehat{\varepsilon}_{21}$ and

$$\|\delta\widehat{U}\|_{\mathrm{F}} \leq \sqrt{2}\,\|\widehat{Y}_0\|_{\mathrm{F}} + O(\varepsilon^2) \leq (\sqrt{2}\,l)\widehat{\varepsilon}_{21} + O(\varepsilon^2), \tag{5.19}$$

$$\|\delta\widehat{\Sigma}\|_{\mathrm{F}} \leq (1 + \sqrt{2}\,ls)\widehat{\varepsilon}_{21} + O(\varepsilon^2). \tag{5.20}$$

Since the equality $\widehat{\varepsilon}_{21} = \varepsilon$ is possible, it follows that the quantities $K_{\widehat{U}} := \sqrt{2}\,l$ and $K_{\widehat{\Sigma}} := 1 + \sqrt{2}\,sl$ are estimates of the absolute condition numbers of the symplectic basis $\widehat{U}$ and the Hamiltonian block-Schur form $\widehat{\Sigma}$, respectively.

The condition numbers $K_{\widehat{U}}$ and $K_{\widehat{\Sigma}}$ for the Hamiltonian block-Schur form can be much smaller than the corresponding numbers K_U and K_Σ for the Hamiltonian Schur form. The reason is that when dealing with the Hamiltonian block-Schur form we do not transform the $(1,1)$ block of $H + \delta H$ into upper triangular form, which leaves more freedom in $\widehat{X}_0$ in comparison with X_0. In fact $\widehat{X}_0$ may be chosen as zero.

In a first order approximation the gap γ (see (3.9)) between the stable invariant subspaces of H and $\widetilde{H}$ may be estimated as

$$\gamma \leq \|\widehat{Y}_0\|_2 + O(\varepsilon^2) \leq l_2\|\widehat{E}_{21}\|_2 + O(\varepsilon^2), \tag{5.21}$$

where

$$l_2 := \max_{Y \neq 0} \frac{\|\widehat{\mathcal{L}}^{-1}(Y)\|_2}{\|Y\|_2} = \|\widehat{\mathcal{L}}^{-1}(I_n)\|_2. \tag{5.22}$$

Note that l_2 is invariant under the action of the stabilizer of $\widehat{\Sigma}$ in $\mathbf{US}(2n)$. Indeed, following the proof of Proposition 2.2 we see that this stabilizer consists of matrices $\mathrm{diag}(V, V)$, $V \in \mathbf{U}(n)$. Hence, the norms of the operators $\widehat{\mathcal{L}}$ and $\widehat{\mathcal{L}}^{-1}$, induced by any unitarily invariant norm in $\mathcal{C}^{n\times n}$, are the same for each $\widehat{T}$ with $\mathrm{spect}(\widehat{T}) = \mathrm{spect}_-(H)$ in (2.4). Thus the *condition number* for the gap is

$$l_2 = \|\Gamma\|_2, \tag{5.23}$$

where the matrix $\Gamma = \mathcal{L}^{-1}(I_n)$ solves the stable Lyapunov equation $T^{\mathrm{H}}\Gamma + \Gamma T = I_n$; see [11].

The linear term $l_2\|\widehat{E}_{21}\|_2$ in the estimate (5.21), (5.23) coincides with the linear term in the estimate for the gap from [35].

5.3. Block-Schur form. Set

$$\overline{\varepsilon} := \|\overline{H}\|_{\mathrm{F}} = \|\overline{E}\|_{\mathrm{F}}, \qquad \overline{\varepsilon}_{ij} := \|\overline{E}_{ij}\|_{\mathrm{F}},$$

and let $\overline{Z}^0 = [\overline{Z}^0_{ij}]$ be the first order approximation to $\overline{Z}$, where $\overline{Z}^0_{ij} \in \mathcal{C}^{n\times n}$. Within first order terms in $\overline{\varepsilon}$ the equation for zeroing the (2,1) block of $\overline{\Sigma} + \delta\overline{\Sigma}$ is

$$\overline{\mathcal{L}}(\overline{Z}^0_{21}) := \overline{S}\,\overline{Z}^0_{21} - \overline{Z}^0_{21}\overline{T} = -\overline{E}_{21}.$$

Hence, $\overline{Z}^0_{21} = -\overline{\mathcal{L}}^{-1}(\overline{E}_{21})$, where the operator $\overline{\mathcal{L}}$ is invertible in view of assumption A3 and the requirement that $\mathrm{spect}(\overline{T}) = \Lambda_-$.

The first order approximations to the conditions for unitarity (3.10) are

$$(\overline{Z}^0_{ii})^{\mathrm{H}} + \overline{Z}^0_{ii} = 0, \qquad \overline{Z}^0_{12} = -(\overline{Z}^0_{21})^{\mathrm{H}}.$$

In order to have a solution $\overline{Z}^0$ of smallest F-norm we choose $\overline{Z}^0_{11} = \overline{Z}^0_{22} = 0$. Hence,

$$\|\delta\overline{U}\|_{\mathrm{F}} \le \sqrt{2}\left\|\overline{Z}^0_{21}\right\|_{\mathrm{F}} + O(\overline{\varepsilon}^2) \le (\sqrt{2}\,\overline{l})\overline{\varepsilon}_{21} + O(\overline{\varepsilon}^2), \tag{5.24}$$

$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le (1 + \sqrt{2}\,\overline{l}\,\overline{s})\overline{\varepsilon}_{21} + O(\overline{\varepsilon}^2), \tag{5.25}$$

where

$$\overline{l} := \max_{Y\neq 0} \frac{\|\overline{\mathcal{L}}^{-1}(Y)\|_{\mathrm{F}}}{\|Y\|_{\mathrm{F}}} = \|\overline{L}^{-1}\|_2, \qquad \overline{s} := \left\|I_{2n}\otimes\overline{\Sigma} - \overline{\Sigma}^{\top}\otimes I_{2n}\right\|_2, \tag{5.26}$$

and $\overline{L} := I_n \times \overline{S} - \overline{T}^{\top} \otimes I_n$. The quantities $K_{\overline{U}} := \sqrt{2}\,\overline{l}$ and $K_{\overline{\Sigma}} := 1 + \sqrt{2}\,\overline{l}\overline{s}$ are the absolute condition numbers of the unitary basis $\overline{U}$ and of the block-Schur form $\overline{\Sigma}$, respectively.

5.4. Summary of the first order perturbation analysis. We summarize the results of the first order perturbation analysis in the following theorem.

THEOREM 5.1. *Consider a Hamiltonian matrix as in* (2.1) *and a perturbation as in* (3.1).

1. *For the Hamiltonian Schur form we have*

$$\|\delta U\|_{\mathrm{F}} \le \varepsilon K_U + O(\varepsilon^2),$$
$$\|\delta\Sigma\|_{\mathrm{F}} \le \varepsilon K_\Sigma + O(\varepsilon^2),\ K_\Sigma := 1 + sK_U,$$

   where K_U is given by (5.18).
2. *For the Hamiltonian block-Schur form we have*

$$\|\delta\widehat{U}\|_{\mathrm{F}} \le \sqrt{2}\,\|\widehat{Y}_0\|_{\mathrm{F}} + O(\varepsilon^2) \le (\sqrt{2}\,l)\widehat{\varepsilon}_{21} + O(\varepsilon^2),$$
$$\|\delta\widehat{\Sigma}\|_{\mathrm{F}} \le (1 + \sqrt{2}\,ls)\widehat{\varepsilon}_{21} + O(\varepsilon^2)$$

   with l, s given by (5.9) *and* (5.12).
3. *For the gap between the stable invariant subspaces we have*

$$\gamma \le \|\widehat{Y}_0\|_2 + O(\varepsilon^2) \le l_2\|\widehat{E}_{21}\|_2 + O(\varepsilon^2)$$

   with l_2 given by (5.23).

4. *For the block-Schur form we have*

$$\|\delta\overline{U}\|_{\mathrm{F}} \le \sqrt{2}\left\|\overline{Z}_{21}^{0}\right\|_{\mathrm{F}} + O(\bar{\varepsilon}^2) \le (\sqrt{2}\,\bar{l})\bar{\varepsilon}_{21} + O(\bar{\varepsilon}^2),$$
$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le (1+\sqrt{2}\,\bar{l}\,\bar{s})\bar{\varepsilon}_{21} + O(\bar{\varepsilon}^2),$$

where $\bar{l}, \bar{s}$ *are given by* (5.26).

Note that the bounds (5.24), (5.25) for the block-Schur form may be compared with the bounds (5.19), (5.20) for the Hamiltonian block-Schur form as follows. The bounds (5.24), (5.25) are valid under assumption 3 of Theorem 5.1, where H may have purely imaginary eigenvalues and hence the bounds (5.19), (5.20) do not hold. If $H \in \mathbf{Ham}_0(2n)$, then the bounds for the Hamiltonian block-Schur and block-Schur form are identical.

6. Nonlocal perturbation analysis. In this section we derive nonlinear, nonlocal perturbation bounds for the Hamiltonian Schur and block-Schur forms of H as functions of the quantities ε_{ij}, $\widehat{\varepsilon}_{ij}$, $\bar{\varepsilon}_{ij}$. For this purpose we rewrite the perturbation problem as an equivalent operator equation for the blocks of the matrices Z, $\widehat{Z}$, and $\overline{Z}$.

6.1. Hamiltonian Schur form. The blocks X and Y in Z contain $4n^2$ real elements. For these elements we have $n(n-1)/2$ complex (or $n(n-1)$ real) equations from (4.5), n^2 complex (or $2n^2$ real) equations from (4.6), and n^2 real equations from (3.7). Thus at this stage we face the more general problem of perturbation analysis for transformation matrices U from the set $\mathbf{S}^*(2n)$ of matrices of the form $M = \mathcal{U}[V,\ (I_n - VV^{\mathrm{H}})^{1/2}N]$ with $N \in \mathbf{U}(n)$ and $\|V\|_2 \le 1$; see Appendix A. In the following we construct an operator equation for X, Y and, using the Schauder fixed point principle [14, 25], we show that it has at least one solution in a closed convex set $\Delta(\varepsilon) \subset \mathbf{S}^*(2n)$. The diameter of $\Delta(\varepsilon)$ tends to zero together with ε, and this gives us the desired perturbation bounds. An additional canonical projection of the resulting matrix $I_{2n} + Z$ with $Z = \mathcal{U}[X, Y]$ into the group of unitary matrices proves that the bounds are valid for the original problem as well.

Set

$$x_1 := \mathbf{lvec}(X) \in \mathcal{C}^{\ell},$$
$$x_2 := \mathbf{dvec}(X) \in \mathcal{C}^{n},$$
$$x_3 := \mathbf{uvec}(X) \in \mathcal{C}^{\ell},$$
$$x_4 := \mathbf{vec}(Y) \in \mathcal{C}^{n^2},$$

and $x := \left[x_1^{\top}, x_2^{\top}, x_3^{\top}, x_4^{\top}\right]^{\top} \in \mathcal{C}^{2n^2}$. We have

$$\|X\|_{\mathrm{F}} = \sqrt{\|x_1\|^2 + \|x_2\|^2 + \|x_3\|^3}, \qquad \|Y\|_{\mathrm{F}} = \|x_4\|,$$

and

$$\|Z\|_{\mathrm{F}} = \sqrt{2\left(\|X\|_{\mathrm{F}}^2 + \|Y\|_{\mathrm{F}}^2\right)} = \sqrt{2}\,\|x\|.$$

Recalling that $\|\delta H\|_{\mathrm{F}} = \varepsilon$ and using (4.10), we may show that (4.7), (4.11), and (3.7) are equivalent to the operator equation $x = \Phi(x, \varepsilon)$, where the components of

the operator $\Phi := [\Phi_1^\top, \Phi_2^\top, \Phi_3^\top, \Phi_4^\top]^\top : \mathcal{C}^{2n^2} \to \mathcal{C}^{2n^2}$ are given by the relations

$$\begin{aligned}
\Phi_1(x,\varepsilon) &:= M^{-1}\Omega\left(\left(\delta T^\top \otimes I_n\right)\Omega^\top x_1 + (I_n \otimes R)x_4 - \mathbf{vec}(F_{11})\right),\\
\Phi_2(x) &:= -0.5\,\mathbf{dvec}(XX^{\mathrm{H}} + YY^{\mathrm{H}}),\\
\Phi_3(x) &:= -\overline{x}_1 - \mathbf{uvec}(XX^{\mathrm{H}} + YY^{\mathrm{H}}),\\
\Phi_4(x,\varepsilon) &:= -L^{-1}\left(\left(\delta T^\top \otimes I_n\right)x_4 + \mathbf{vec}(F_{21})\right).
\end{aligned} \tag{6.1}$$

Here $\overline{x}_1$ is the complex conjugate of x_1, the matrix M is as in (4.12), and the matrix L is as in (5.9). It is assumed also that X has a real main diagonal; see [16]. To get tighter bounds it is better to work with the Hamiltonian perturbation matrix E instead of with the general perturbation matrix F. Since $F = E(I_{2n} + Z)$ we obtain $F_{i1} = E_{i1}(I_n + X) - E_{i2}Y$ and

$$\|F_{i1}\|_{\mathrm{F}} \le \varepsilon_{i1} + \varepsilon_{i2}\|x_4\|;\ i = 1, 2, \tag{6.2}$$

where $\varepsilon_{11} = \varepsilon_{22}$ and $2\varepsilon_{11}^2 + \varepsilon_{21}^2 + \varepsilon_{12}^2 = \varepsilon^2$; see (5.8), (4.1).

Due to (4.4) we have

$$\begin{aligned}
\|\delta\Sigma\|_{\mathrm{F}} &= \sqrt{2\|\delta T\|_{\mathrm{F}}^2 + \|\delta R\|_{\mathrm{F}}^2} = \left\|\left(I_{2n} + Z^{\mathrm{H}}\right)(\Sigma Z - Z\Sigma) + G\right\|_{\mathrm{F}}\\
&\le \|\Sigma Z - Z\Sigma\|_{\mathrm{F}} + \varepsilon \le s\|Z\|_{\mathrm{F}} + \varepsilon = \sqrt{2}\,s\|x\| + \varepsilon.
\end{aligned}$$

Keeping in mind that $G \in \mathbf{Ham}(2n)$, it follows from (4.10) and the results in [16] that

$$\begin{aligned}
\|\delta T\|_{\mathrm{F}} &\le \left\|\begin{bmatrix} TX - XT - RY \\ T^{\mathrm{H}}Y + YT \end{bmatrix}\right\|_{\mathrm{F}} + \|G_{11}\|_{\mathrm{F}}\\
&= \left\|S_1\begin{bmatrix} \mathbf{vec}(X) \\ \mathbf{vec}(Y) \end{bmatrix}\right\| + \|G_{11}\|_{\mathrm{F}} \le s_1\sqrt{\|X\|_{\mathrm{F}}^2 + \|Y\|_{\mathrm{F}}^2} + \varepsilon/\sqrt{2}\\
&= s_1\|x\| + \varepsilon/\sqrt{2},
\end{aligned}$$

where

$$s_1 := \|S_1\|_2, \qquad S_1 := \begin{bmatrix} I_n \otimes T - T^\top \otimes I_n & -(I_n \otimes R) \\ 0_{n^2\times n^2} & I_n \otimes T^{\mathrm{H}} + T^\top \otimes I_n \end{bmatrix}. \tag{6.3}$$

It follows from (6.3) and (2.2) that $s_1 \le s$.

Let $\xi := [\xi_1, \xi_2, \xi_3, \xi_4]^\top \in \mathcal{R}_+^4$ and

$$c_n := \sqrt{(n-1)/(2n)}.$$

If $\|x_i\| \le \xi_i$, then using (6.3), (6.2), and the estimates from [16] we get

$$\begin{aligned}
\|\Phi_1(x,\varepsilon)\|_{\mathrm{F}} &\le m\|\delta T\|_{\mathrm{F}}\,\xi_1 + r\xi_4 + m(\varepsilon_{11} + \varepsilon_{12}\xi_4)\\
&\le m(s_1\|\xi\| + \varepsilon/\sqrt{2})\xi_1 + (r + m\varepsilon_{12})\xi_4 + m\varepsilon_{11} =: f_1(\xi,\varepsilon),\\
\|\Phi_2(x)\|_{\mathrm{F}} &\le 0.5\,\|\xi\|^2 =: f_2(\xi),\\
\|\Phi_3(x)\|_{\mathrm{F}} &\le \xi_1 + c_n\|\xi\|^2 =: f_3(\xi),\\
\|\Phi_4(x,\varepsilon)\|_{\mathrm{F}} &\le l\|\delta T\|_{\mathrm{F}}\,\xi_4 + l(\varepsilon_{21} + \varepsilon_{11}\xi_4)\\
&\le l(s_1\|\xi\| + \varepsilon/\sqrt{2} + \varepsilon_{11})\xi_4 + l\varepsilon_{21} =: f_4(\xi,\varepsilon),
\end{aligned} \tag{6.4}$$

where $\varepsilon_{ij} = \varepsilon\, e_{ij}$ and $2e_{11}^2 + e_{21}^2 + e_{12}^2 = 1$; see (5.16).

Note that in the trivial case that $e_{11} = e_{21} = 0$ we may choose $\xi = 0$, which corresponds to $x = 0$. Indeed, here the matrix $\Sigma + \delta\Sigma$ is already in Hamiltonian Schur form and there is nothing to transform. So we assume further that at least one of the quantities e_{11} or e_{21} is positive.

Consider the function $f := [f_1, f_2, f_3, f_4]^\top : \mathcal{R}_+^4 \times \mathcal{R}_+ \to \mathcal{R}_+^4$, defined by (6.4). Let $J(\xi, \varepsilon) := f'_\xi(\xi, \varepsilon)$, $\xi \neq 0$, be the Jacobi matrix of f relative to ξ, and set

$$J(0, \varepsilon) := \lim_{\xi \to 0} J(\xi, \varepsilon) = \begin{bmatrix} (m/\sqrt{2})\varepsilon & 0 & 0 & r + m\varepsilon_{12} \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & l\left(\varepsilon/\sqrt{2} + \varepsilon_{11}\right) \end{bmatrix}.$$

The matrix $J(\xi, \varepsilon)$ has nonnegative elements, which are continuous functions of ξ and ε. The spectral radius of $J(0, \varepsilon)$ is

$$\operatorname{rad}(J(0, \varepsilon)) = \max\left\{(m/\sqrt{2})\varepsilon, l(\varepsilon/\sqrt{2} + \varepsilon_{11})\right\},$$

and hence $\operatorname{rad}(J(\xi, \varepsilon)) \to 0$ for $\xi \to 0$ and $\varepsilon \to 0$.

The function f is a vector Lyapunov majorant for the operator Φ; see [10, 18]. We recall that a function f of nonnegative arguments is a Lyapunov majorant for Φ if the following conditions are satisfied:

- For $\|x_i\| \leq \xi_i$ the inequalities $\|\Phi_i(x, \varepsilon)\|_{\mathrm{F}} \leq f_i(\xi, \varepsilon)$ are fulfilled.
- The function f is nonnegative, continuously differentiable, and nondecreasing in all its arguments, $f(0, 0) = 0$ and $\operatorname{rad}(J(0, 0)) < 1$.

In addition we have $\|f(\xi, \varepsilon)\| \to \infty$ as $\|\xi\| \to \infty$, and

$$\|f(\xi, \varepsilon)\| \geq \sqrt{(m\varepsilon_{11})^2 + (l\varepsilon_{21})^2} > 0.$$

Therefore, according to the technique of Lyapunov majorants (see [10]), and using the fact that $f(\xi, \varepsilon)$ is algebraic in ξ and ε, there exists a number $\varepsilon_0 > 0$ with the following properties:

- For $\varepsilon \leq \varepsilon_0$ the system of equations $\xi = f(\xi, \varepsilon)$ has a solution $\xi = \xi(\varepsilon)$, which is continuous and nondecreasing in ε and satisfies $\xi(0) = 0$, $\operatorname{rad}(J(\xi(\varepsilon), \varepsilon)) < 1$ for $\varepsilon < \varepsilon_0$, and $\operatorname{rad}(J(\xi(\varepsilon_0), \varepsilon_0)) = 1$.
- The function $\xi : [0, \varepsilon_0] \to \mathcal{R}_+^4$ is algebraic, differentiable on the interval $[0, \varepsilon_0)$, and its derivative ξ' satisfies

$$\xi'(\varepsilon) = (I_4 - J(\xi(\varepsilon), \varepsilon))^{-1} f'_\varepsilon(\xi(\varepsilon), \varepsilon)$$

  for $\varepsilon < \varepsilon_0$ and $\|\xi'(\varepsilon)\| \to \infty$ as $\varepsilon \to \varepsilon_0$.

Thus the critical value ε_0 for ε may be determined by solving the system

$$\xi = f(\xi, \varepsilon), \qquad \det(I_4 - J(\xi, \varepsilon)) = 0$$

of five algebraic equations for the five unknowns $\xi_1, \ldots, \xi_4, \varepsilon$.

Let $\Delta(\varepsilon) \subset \mathcal{C}^{2n^2}$ be the set of all x such that $\|x_i\| \leq \xi_i(\varepsilon)$ for $\varepsilon \leq \varepsilon_0$. In view of (6.4) the operator Φ maps the convex, compact set $\Delta(\varepsilon)$ into itself. According to the Schauder fixed point principle (see, e.g., [14, 25]), there exists a solution $x \in \Delta(\varepsilon)$ of the operator equation $x = \Phi(x, \varepsilon)$. Thus we have the estimates

$$\|Y\|_{\mathrm{F}} \leq \xi_4(\varepsilon), \qquad \|X\|_{\mathrm{F}} \leq \sqrt{\xi_1^2(\varepsilon) + \xi_2^2(\varepsilon) + \xi_3^2(\varepsilon)},$$

and

$$\|\delta U\|_{\mathrm{F}} = \|Z\|_{\mathrm{F}} = \sqrt{2}\,\|x\| \le \sqrt{2}\,\|\xi(\varepsilon)\|, \tag{6.5}$$

$$\|\delta\Sigma\|_{\mathrm{F}} \le s\|Z\|_{\mathrm{F}} + \varepsilon \le s\sqrt{2}\,\|\xi(\varepsilon)\| + \varepsilon. \tag{6.6}$$

After some calculations the system $\xi = f(\xi,\varepsilon)$ in view of (6.4) yields an algebraic equation of 8th degree for $\|\xi\|$ of the form

$$\sum_{i=0}^{8} a_i(\varepsilon) z^i = 0, \tag{6.7}$$

where the coefficients $a_i(\varepsilon)$ are given by

$$\begin{aligned}
a_0(\varepsilon) &:= 2\nu^2(\varepsilon) + (l\varepsilon_{21})^2\mu^2(\varepsilon),\\
a_1(\varepsilon) &:= -2lms_1\left(2\varepsilon_{11}\nu(\varepsilon) + l\varepsilon_{21}^2\mu(\varepsilon)\right),\\
a_2(\varepsilon) &:= -\lambda^2(\varepsilon)\mu^2(\varepsilon) + 2c_n\lambda(\varepsilon)\mu(\varepsilon)\nu(\varepsilon) + (lms_1)^2\left(2\varepsilon_{11}^2 + \varepsilon_{21}^2\right),\\
a_3(\varepsilon) &:= 2\lambda(\varepsilon)\mu(\varepsilon)(\omega(\varepsilon) - c_n lms_1\varepsilon_{11}) - 2c_n\nu(\varepsilon)\omega(\varepsilon),\\
a_4(\varepsilon) &:= d_n\lambda^2(\varepsilon)\mu^2(\varepsilon) + 2c_n lms_1(s_1\nu(\varepsilon) + \varepsilon_{11}\omega(\varepsilon))\\
&\quad - \omega^2(\varepsilon) - 2lms_1^2\lambda(\varepsilon)\mu(\varepsilon),\\
a_5(\varepsilon) &:= 2\omega(\varepsilon)\left(lms_1^2 - d_n\lambda(\varepsilon)\mu(\varepsilon)\right) - 2c_n(lms_1)^2\varepsilon_{11},\\
a_6(\varepsilon) &:= d_n\left(\omega^2(\varepsilon) + 2lms_1^2\lambda(\varepsilon)\mu(\varepsilon)\right) - (lm)^2 s_1^4,\\
a_7(\varepsilon) &:= -2d_n lms_1^2\omega(\varepsilon),\\
a_8(\varepsilon) &:= d_n(lm)^2 s_1^4
\end{aligned} \tag{6.8}$$

and

$$\begin{aligned}
\lambda(\varepsilon) &:= 1 - l\left(\varepsilon/\sqrt{2} + \varepsilon_{11}\right),\\
\mu(\varepsilon) &:= 1 - (m/\sqrt{2})\varepsilon,\\
d_n &:= (3n-2)/(4n),\\
\nu(\varepsilon) &:= m\varepsilon_{11}\lambda(\varepsilon) + l\varepsilon_{21}(r + m\varepsilon_{12}),\\
\omega(\varepsilon) &:= s_1(l\mu(\varepsilon) + m\lambda(\varepsilon)).
\end{aligned}$$

The discriminant of (6.7) is an algebraic polynomial in ε, whose smallest positive root is the number ε_0. There is no explicit formula for the solution $\|\xi\|$ of (6.7) as a function of ε or for determining ε_0. Computable bounds for these quantities are derived below.

If we represent the solution $\|\xi(\varepsilon)\|$ as a power series in ε, then after some elementary calculations we get

$$\|\xi(\varepsilon)\| = \alpha_1\varepsilon + \alpha_2\varepsilon^2 + O(\varepsilon^3),$$

where the coefficients $\alpha_1 \le K_U/\sqrt{2}$ and α_2 are determined from

$$\begin{aligned}
\alpha_1 &:= \sqrt{2(me_{11} + lre_{21})^2 + (le_{21})^2},\\
\alpha_2 &:= \frac{\beta_{-1}}{\alpha_1} + \beta_0 + \beta_1\alpha_1 + \beta_2\alpha_1^2,
\end{aligned} \tag{6.9}$$

and

$$\beta_{-1} := 2lm(me_{11} + lre_{21})\left(e_{12}e_{21} - e_{11}^2 - e_{11}/\sqrt{2}\right) - (m/\sqrt{2})(le_{21})^2,$$

$$\beta_0 := -lms_1\left(2me_{11}^2 + 2lre_{11}e_{21} + le_{21}^2\right), \tag{6.10}$$

$$\beta_1 := c_n(me_{11} + lre_{21}) + le_{11} + (l+m)/\sqrt{2},\ \beta_2 := s_1(l+m).$$

Hence,

$$\|\delta U\|_{\mathrm{F}} \le (\sqrt{2}\,\alpha_1)\varepsilon + (\sqrt{2}\,\alpha_2)\varepsilon^2 + O(\varepsilon^3), \tag{6.11}$$

$$\|\delta\Sigma\|_{\mathrm{F}} \le (1 + \sqrt{2}\,s\alpha_1)\varepsilon + (\sqrt{2}\,s\alpha_2)\varepsilon^2 + O(\varepsilon^3). \tag{6.12}$$

We may bound f from above to get slightly less sharp bounds $\|x_i\| \le \eta_i(\varepsilon)$, which are easier to compute (these new bounds will differ from $\xi_i(\varepsilon)$ only by $O(\varepsilon^2)$ terms). It follows from the second and third equations of the system $\xi = f(\xi, \varepsilon)$ and from (6.4) that

$$\xi_3 = \xi_1 + 2c_n\xi_2$$

and

$$\xi_2 = \xi_1^2 + 2c_n\xi_1\xi_2 + (0.5 + 2c_n^2)\xi_2^2 + 0.5\,\xi_4^2.$$

Hence,

$$\xi_1 \le \sqrt{\xi_2} = \|\xi\|/\sqrt{2}$$

(because of the definition of f_2) and, since $\xi_4 \le \|\xi\|$, we obtain $\xi_1\|\xi\| \le \|\xi\|^2/\sqrt{2}$ and $\xi_4\|\xi\| \le \|\xi\|^2$. Thus we get the majorant system for the vector η with elements $\eta_i \ge \xi_i$,

$$\begin{aligned}
\eta_1 &= (m/\sqrt{2})\left(s_1\|\eta\|^2 + \varepsilon\eta_1\right) + (r + m\varepsilon_{12})\eta_4 + m\varepsilon_{11},\\
\eta_2 &= 0.5\|\eta\|^2,\\
\eta_3 &= \eta_1 + c_n\|\eta\|^2,\\
\eta_4 &= l\left(s_1\|\eta\|^2 + \left(\varepsilon/\sqrt{2} + \varepsilon_{11}\right)\eta_4\right) + l\varepsilon_{21}.
\end{aligned}$$

This system yields the following biquadratic equation in $\|\eta\|$

$$\alpha(\varepsilon)\|\eta\|^4 - \beta(\varepsilon)\|\eta\|^2 + a_0(\varepsilon) = 0, \tag{6.13}$$

where

$$\begin{aligned}
\alpha(\varepsilon) &:= \mu^2(\varepsilon)\left((ls_1)^2 + d_n\lambda^2(\varepsilon)\right) + 2\nu_1(\varepsilon)(\nu_1(\varepsilon) + c_n\lambda(\varepsilon)\mu(\varepsilon)),\\
\beta(\varepsilon) &:= \mu^2(\varepsilon)\left(\lambda^2(\varepsilon) - 2l^2s_1\varepsilon_{21}\right) - 2\nu(\varepsilon)(2\nu_1(\varepsilon) + c_n\lambda(\varepsilon)\mu(\varepsilon)),\\
\nu_1(\varepsilon) &:= s_1\left(m\lambda(\varepsilon)/\sqrt{2} + l(r + m\varepsilon_{12})\right),
\end{aligned} \tag{6.14}$$

and $a_0(\varepsilon)$ is defined from (6.8). We choose the smaller positive root of equation (6.13),

$$\|\eta(\varepsilon)\| = \sqrt{\frac{2a_0(\varepsilon)}{\beta(\varepsilon) + \sqrt{\beta^2(\varepsilon) - 4\alpha(\varepsilon)a_0(\varepsilon)}}},\ \varepsilon \in [0, \varepsilon_1], \tag{6.15}$$

which is of order $O(\varepsilon)$. Here $\varepsilon_1 < \varepsilon_0$ is the smallest positive root of the equation

$$\beta^2(\varepsilon) = 4\alpha(\varepsilon)a_0(\varepsilon). \tag{6.16}$$

The quantity $\varepsilon_1 > 0$ is well defined. Indeed, $\beta(0) = 1$, $a_0(0) = 0$, and 0 is not a root of (6.16). A direct computation shows that $\varepsilon' := (l(1/\sqrt{2} + e_{11}))^{-1} > 0$ is a root of (6.16), so the equation has at least one positive root. Since $\beta(\varepsilon') < 0$, we see that $\varepsilon_1 < \varepsilon'$.

As a result, using (4.4) and (5.12), we obtain the rigorous and easily computable bounds

$$\|\delta U\|_{\mathrm{F}} \le \sqrt{2}\,\|\eta(\varepsilon)\|, \tag{6.17}$$

$$\|\delta\Sigma\|_{\mathrm{F}} \le \varepsilon + s\sqrt{2}\,\|\eta(\varepsilon)\|. \tag{6.18}$$

It may be shown that $\|\eta(\varepsilon)\| = \alpha_1\varepsilon + O(\varepsilon^2) = \|\xi(\varepsilon)\| + O(\varepsilon^2)$. Hence, the bounds (6.17), (6.18) coincide with (6.5), (6.6) within first order terms of magnitude relative to the small parameter ε.

The above perturbation analysis for the Hamiltonian Schur form solves the perturbation problem for transformation matrices from $\mathbf{S}^*(2n)$ rather than from $\mathbf{US}(2n)$. However, the bounds are the same for matrices from $\mathbf{US}(2n)$. Indeed, let $\widetilde{U} = \mathcal{U}[V, W](I_{2n} + Z) \in \mathbf{S}^*(2n)$, where $Z = \mathcal{U}[X, Y]$, be the matrix for which the derived nonlinear nonlocal perturbation bounds hold. In general $\widetilde{U} \notin \mathbf{US}(2n)$. In this case according to the parametrization (A.8) of $\mathbf{US}(2n)$, there exists a matrix $R \in \mathbf{U}(n)$ (not necessarily as in (2.2)) such that $I_{2n} + \mathcal{U}[X, YR] \in \mathbf{US}(2n)$, and hence

$$\widetilde{U}_c := \mathcal{U}[V, W](I_{2n} + \mathcal{U}[X, YR]) \in \mathbf{US}(2n).$$

The matrix $Y_R := YR$ satisfies the equation

$$\mathcal{L}_R(Y_R) := T^{\mathrm{H}}Y_R + Y_R(R^{\mathrm{H}}TR) = -(Y_R R^{\mathrm{H}}\delta T + F_{21})R.$$

It follows from (4.7) that the matrix representation $L_R := (R^{\mathrm{H}}TR)^{\mathrm{H}} \otimes I_n + I_n \otimes T^{\mathrm{H}}$ of the operator $\mathcal{L}_R$ is unitarily similar to the matrix $L := T^{\top} \otimes I_n + I_n \otimes T^{\mathrm{H}}$ of the operator $\mathcal{L}$. Indeed, we have $L_R = (R^{\top} \otimes I_n)L(R^{\top} \otimes I_n)^{\mathrm{H}}$, where $R^{\top} \otimes I_n \in \mathbf{U}(n^2)$. Hence,

$$\max_{Y \neq 0} \frac{\|\mathcal{L}_R^{-1}(Y)\|_{\mathrm{F}}}{\|Y\|_{\mathrm{F}}} = \left\|L_R^{-1}\right\|_2 = \|L^{-1}\|_2 = l,$$

and the above perturbation bounds hold for X and Y_R as well.

6.2. Hamiltonian block-Schur form. The nonlocal perturbation analysis for the Hamiltonian block-Schur form (2.4) is easier than the analysis of the Hamiltonian Schur form, because we do not have to make the $(1, 1)$ block of $\widetilde{\Sigma}$ upper triangular. This additional freedom in the transformation matrix explains why the form (2.4) may be less sensitive to perturbations in comparison with the form (2.2).

To simplify the presentation we assume $X \in \mathbf{Her}(n)$ in (3.7), which implies

$$2\widehat{X} + \widehat{X}^2 + \widehat{Y}\widehat{Y}^{\mathrm{H}} = 0, \qquad \widehat{\mathcal{L}}(\widehat{Y}) := \widehat{T}^{\mathrm{H}}\widehat{Y} + \widehat{Y}\widehat{T} = -\widehat{Y}\delta\widehat{T} - \widehat{F}_{21}. \tag{6.19}$$

Set $\widehat{x}_1 := \mathbf{vec}(\widehat{X})$, $\widehat{x}_2 := \mathbf{vec}(\widehat{Y}) \in \mathcal{C}^n$. Then the system of matrix equations (6.19) may be written as an equivalent operator equation $\widehat{x} = \widehat{\Phi}(\widehat{x}, \varepsilon)$, where $\widehat{x} := [\widehat{x}_1^\top, \widehat{x}_2^\top]^\top \in \mathcal{C}^{2n^2}$, $\widehat{\Phi} := [\widehat{\Phi}_1^\top, \widehat{\Phi}_2^\top]^\top : \mathcal{C}^{2n^2} \to \mathcal{C}^{2n^2}$, and

$$\widehat{\Phi}_1(\widehat{x}) := -0.5\,\mathbf{vec}(\widehat{X}^2 + \widehat{Y}\widehat{Y}^{\mathrm{H}}),$$
$$\widehat{\Phi}_2(\widehat{x}, \varepsilon) := -\widehat{L}^{-1}\left(\left(\delta\widehat{T}^\top \otimes I_n\right)\widehat{x}_2 + \mathbf{vec}(\widehat{F}_{21})\right).$$

Let $\widehat{\xi} := [\widehat{\xi}_1, \widehat{\xi}_2]^\top \in \mathcal{R}_+^2$ and $\|\widehat{x}_i\| \le \widehat{\xi}_i$. Then we have

$$\|\widehat{\Phi}_1(\widehat{x}, \varepsilon)\|_{\mathrm{F}} \le 0.5\,\|\widehat{\xi}\|^2 =: \widehat{f}_1(\widehat{\xi}),$$
$$\|\widehat{\Phi}_2(\widehat{x}, \varepsilon)\|_{\mathrm{F}} \le l(s_1\|\widehat{\xi}\| + \varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11})\widehat{\xi}_2 + l\widehat{\varepsilon}_{21} =: \widehat{f}_2(\widehat{\xi}, \varepsilon),$$

where $\widehat{\varepsilon}_{ij} := \varepsilon\widehat{e}_{ij}$. We may assume that $\widehat{e}_{21} > 0$, since otherwise $\widehat{\Sigma} + \delta\widehat{\Sigma}$ is in Hamiltonian block-Schur form, and we have the trivial case $\widehat{\xi} = 0$ and $\widehat{x} = 0$.

As in section 6.1 it can be shown that the function $\widehat{f} := [\widehat{f}_1, \widehat{f}_2]^\top : \mathcal{R}_+^2 \times \mathcal{R}_+ \to \mathcal{R}_+^2$ is a vector Lyapunov majorant for the operator $\widehat{\Phi}$. Hence, there exists $\widehat{\varepsilon}_0 > 0$ such that for $\varepsilon \le \widehat{\varepsilon}_0$ the system of algebraic equations $\widehat{\xi} = \widehat{f}(\widehat{\xi}, \varepsilon)$ has a continuous solution $\widehat{\xi} = \widehat{\xi}(\varepsilon)$, differentiable in $\varepsilon < \widehat{\varepsilon}_0$ and such that $\widehat{\xi}(0) = 0$.

Let $\widehat{\Delta}(\varepsilon) \subset \mathcal{C}^{2n^2}$ be the set of all $\widehat{x}$ with $\|\widehat{x}_i\| \le \widehat{\xi}(\varepsilon)$ for $\varepsilon \le \widehat{\varepsilon}_0$. Then there exists a solution $\widehat{x} \in \widehat{\Delta}(\varepsilon)$ of the operator equation $\widehat{x} = \widehat{\Phi}(\widehat{x}, \varepsilon)$. Therefore we have the estimates

$$\|\delta\widehat{U}\|_{\mathrm{F}} = \|\widehat{Z}\|_{\mathrm{F}} = \sqrt{2}\,\|\widehat{x}\| \le \sqrt{2}\,\|\widehat{\xi}(\varepsilon)\|, \tag{6.20}$$
$$\|\delta\widehat{\Sigma}\|_{\mathrm{F}} \le s\|\widehat{Z}\|_{\mathrm{F}} + \varepsilon \le s\sqrt{2}\,\|\widehat{\xi}(\varepsilon)\| + \varepsilon. \tag{6.21}$$

The system $\widehat{\xi} = \widehat{f}(\widehat{\xi}, \varepsilon)$ yields an algebraic equation of 6th degree for $\|\widehat{\xi}\|$,

$$\sum_{i=0}^{6} \widehat{a}_i \|\widehat{\xi}\|^i = 0, \tag{6.22}$$

where

$$\widehat{a}_0(\varepsilon) := (l\widehat{\varepsilon}_{21})^2, \quad \widehat{a}_1(\varepsilon) := 0, \quad \widehat{a}_2(\varepsilon) := -\widehat{\lambda}^2(\varepsilon), \quad \widehat{a}_3(\varepsilon) := 2ls_1\widehat{\lambda}(\varepsilon),$$
$$\widehat{a}_4(\varepsilon) := 0.25\widehat{\lambda}^2 - (ls_1)^2, \quad \widehat{a}_5(\varepsilon) := -0.5\,ls_1\widehat{\lambda}(\varepsilon), \quad \widehat{a}_6(\varepsilon) := 0.25(ls_1)^2,$$

and $\widehat{\lambda}(\varepsilon) := 1 - l(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11})$. The smallest positive root of the discriminant of (6.22) is the number $\widehat{\varepsilon}_0$. If we represent the solution $\|\widehat{\xi}\|$ of (6.22) as a power series in ε, then we get

$$\|\widehat{\xi}(\varepsilon)\| = l\widehat{\varepsilon}_{21} + l^2\widehat{\varepsilon}_{21}(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11} + ls_1\widehat{\varepsilon}_{21}) + O(\varepsilon^3).$$

Hence,

$$\|\delta\widehat{U}\|_{\mathrm{F}} \le \sqrt{2}\,l\widehat{\varepsilon}_{21} + \sqrt{2}\,l^2\widehat{\varepsilon}_{21}(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11} + ls_1\widehat{\varepsilon}_{21}) + O(\varepsilon^3), \tag{6.23}$$
$$\|\delta\widehat{\Sigma}\|_{\mathrm{F}} \le (1 + \sqrt{2}\,ls)\widehat{\varepsilon}_{21} + \sqrt{2}\,l^2s\widehat{\varepsilon}_{21}(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11} + ls_1\widehat{\varepsilon}_{21}) + O(\varepsilon^3). \tag{6.24}$$

Again, we may bound $\widehat{f}$ from above using the inequality $\|\widehat{\xi}\|\widehat{\xi}_2 \le \|\widehat{\xi}\|^2$ in order to get slightly less sharp, but easily computable, perturbation bounds. This gives a new majorant system for $\widehat{\eta} := [\widehat{\eta}_1, \widehat{\eta}_2]^\top \in \mathcal{R}_+^2$ with $\widehat{\eta}_i \ge \widehat{\xi}_i$, namely,

$$\widehat{\eta}_1 = 0.5\,\|\widehat{\eta}\|^2, \qquad \widehat{\eta}_2 = l\left(s_1\|\widehat{\eta}\|^2 + (\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11})\widehat{\eta}_2\right) + l\widehat{\varepsilon}_{21}.$$

The quantity $\|\widehat{\eta}\|$ satisfies the biquadratic equation

$$\widehat{\alpha}(\varepsilon)\|\widehat{\eta}\|^4 - \widehat{\beta}(\varepsilon)\|\widehat{\eta}\|^2 + \widehat{a}_0(\varepsilon) = 0, \tag{6.25}$$

where

$$\widehat{\alpha}(\varepsilon) := 0.25\,\widehat{\lambda}^2(\varepsilon) + (ls_1)^2, \qquad \widehat{\beta}(\varepsilon) := \widehat{\lambda}^2(\varepsilon) - 2l^2 s_1 \widehat{\varepsilon}_{21}. \tag{6.26}$$

The positive root of (6.25) of order $O(\varepsilon)$ is

$$\|\widehat{\eta}(\varepsilon)\| = \sqrt{\frac{2\widehat{a}_0(\varepsilon)}{\widehat{\beta}(\varepsilon) + \sqrt{\widehat{\beta}^2 - 4\widehat{\alpha}(\varepsilon)\widehat{a}_0(\varepsilon)}}}, \quad \varepsilon \le \widehat{\varepsilon}_1, \tag{6.27}$$

where $\widehat{\varepsilon}_1 < \widehat{\varepsilon}_0$ is the smallest positive root of the equation $\widehat{\beta}^2(\varepsilon) = 4\widehat{\alpha}(\varepsilon)\widehat{a}_0(\varepsilon)$. Thus the perturbation bounds for the Hamiltonian block-Schur form becomes

$$\|\delta\widehat{U}\|_{\mathrm{F}} \le \sqrt{2}\,\|\widehat{\eta}(\varepsilon)\|, \tag{6.28}$$

$$\|\delta\widehat{\Sigma}\|_{\mathrm{F}} \le s\sqrt{2}\,\|\widehat{\eta}(\varepsilon)\| + \varepsilon. \tag{6.29}$$

Note that $\|\widehat{\eta}(\varepsilon)\| = l\widehat{\varepsilon}_{21} + O(\varepsilon^2) = \|\widehat{\xi}(\varepsilon)\| + O(\varepsilon^2)$ and hence the bounds (6.20), (6.21) and (6.28), (6.29) coincide within first order terms relative to ε.

Using (3.9), a nonlocal bound for the gap γ is straightforward, i.e.,

$$\gamma \le \widehat{\eta}_2(\varepsilon)\sqrt{1 + \widehat{\eta}_2^2(\varepsilon)}, \qquad \widehat{\eta}_2(\varepsilon) := \frac{l\left(s_1\|\widehat{\eta}(\varepsilon)\|^2 + \widehat{\varepsilon}_{21}\right)}{\widehat{\lambda}(\varepsilon)}. \tag{6.30}$$

As in section 6.1 the presented perturbation analysis solves the perturbation problem for transformation matrices from $\mathbf{S}^*(2n)$ rather than from $\mathbf{US}(2n)$. The same arguments as before imply that the bounds are the same for matrices from $\mathbf{US}(2n)$.

It can be shown that the given estimates for the gap are identical in first order with those in [35] (see also Theorem 2.8 from [36]). As nonlinear expressions these estimates are alternative to the estimates for the gap, i.e., one or the other may give better results depending on H and δH.

6.3. Block-Schur form. The nonlocal perturbation analysis for the block-Schur form (2.4) makes sense even when the matrix H has imaginary eigenvalues as described in assumption A4. In this case we cannot use unitary symplectic transformations, since they yield $\overline{S} = -\overline{T}^{\mathrm{H}}$ and the Lyapunov operator $\overline{\mathcal{L}}$ would be singular.

For the sake of simplicity we take $\overline{Z}_{11}, \overline{Z}_{22} \in \mathbf{Her}(n)$, which implies

$$\begin{aligned} &2\overline{Z}_{ii} + \overline{Z}_{ii}^2 + \overline{Z}_{ji}\overline{Z}_{ji}^{\mathrm{H}} = 0, \qquad (I_n + \overline{Z}_{11})\overline{Z}_{21}^{\mathrm{H}} + \overline{Z}_{12}(I_n + \overline{Z}_{22}) = 0, \\ &\overline{\mathcal{L}}(\overline{Z}_{21}) := \overline{S}\,\overline{Z}_{21} - \overline{Z}_{21}\overline{T} = \overline{Z}_{21}\delta\overline{T} - \overline{F}_{21}. \end{aligned} \tag{6.31}$$

In the following analysis we use the representation (4.15), which yields

$$\delta\overline{T} = [I_n + \overline{Z}_{11}, \overline{Z}_{12}] \begin{bmatrix} \overline{T}\,\overline{Z}_{11} - \overline{Z}_{11}\overline{T} + \overline{R}\,\overline{Z}_{21} \\ \overline{S}\,\overline{Z}_{21} - \overline{Z}_{21}\overline{T} \end{bmatrix}$$

and $\|\delta\overline{T}\|_{\mathrm{F}} \le \overline{s}_1\sqrt{\|\overline{Z}_{11}\|_{\mathrm{F}}^2 + \|\overline{Z}_{21}\|_{\mathrm{F}}^2}$, where

$$\overline{s}_1 := \left\| \begin{bmatrix} I_n \otimes \overline{T} - \overline{T}^{\top} \otimes I_n & I_n \otimes \overline{R} \\ 0_{n^2 \times n^2} & I_n \otimes \overline{S} - \overline{T}^{\top} \otimes I_n \end{bmatrix} \right\|_2. \tag{6.32}$$

Set $\overline{x}_1 := \mathbf{vec}(\overline{Z}_{11})$, $\overline{x}_2 := \mathbf{vec}(\overline{Z}_{12})$, $\overline{x}_3 := \mathbf{vec}(\overline{Z}_{22})$, $\overline{x}_4 := \mathbf{vec}(\overline{Z}_{21}) \in \mathcal{C}^n$. Then the system (6.31) is equivalent to the operator equation $\overline{x} = \overline{\Phi}(\overline{x}, \overline{\varepsilon})$, where $\overline{x} := [\overline{x}_1^\top, \overline{x}_2^\top, \overline{x}_3^\top, \overline{x}_4^\top]^\top \in \mathcal{C}^{4n^2}$, $\overline{\Phi} := [\overline{\Phi}_1^\top, \overline{\Phi}_2^\top, \overline{\Phi}_3^\top, \overline{\Phi}_4^\top]^\top : \mathcal{C}^{4n^2} \to \mathcal{C}^{4n^2}$, and

$$\begin{aligned}
\overline{\Phi}_1(\overline{x}) &:= -0.5\,\mathbf{vec}\left(\overline{Z}_{11}^2 + \overline{Z}_{12}\overline{Z}_{12}^{\mathrm{H}}\right),\\
\overline{\Phi}_2(\overline{x}) &:= -\mathbf{vec}\left((I_n + \overline{Z}_{11})\overline{Z}_{21}^{\mathrm{H}}(I_n + \overline{Z}_{22})^{-1}\right),\\
\overline{\Phi}_3(\overline{x}) &:= -0.5\,\mathbf{vec}\left(\overline{Z}_{22}^2 + \overline{Z}_{21}\overline{Z}_{21}^{\mathrm{H}}\right),\\
\overline{\Phi}_4(\overline{x}, \overline{\varepsilon}) &:= -\overline{L}^{-1}\left(\left(\delta\overline{T}^\top \otimes I_n\right)\overline{x}_4 + \mathbf{vec}(\overline{F}_{21})\right).
\end{aligned}$$

Let $\overline{\xi} := [\overline{\xi}_1, \overline{\xi}_2, \overline{\xi}_3, \overline{\xi}_4]^\top \in \mathcal{R}_+^4$ and $\|\overline{x}_i\| \le \overline{\xi}_i$. Since $\overline{F}_{21} = \overline{E}_{21}(I_n + \overline{Z}_{11}) + \overline{E}_{22}\overline{Z}_{21}$, we have $\|\overline{F}_{21}\|_{\mathrm{F}} \le \overline{\varepsilon}_{21} + \overline{\varepsilon}_{22}\overline{\xi}_4$ and

$$\begin{aligned}
&\|\overline{\Phi}_1(\overline{x})\|_{\mathrm{F}} \le 0.5\left(\overline{\xi}_1^2 + \overline{\xi}_2^2\right) =: \overline{f}_1(\overline{\xi}), \qquad \|\overline{\Phi}_2(\overline{x})\|_{\mathrm{F}} \le \frac{\overline{\xi}_4}{1 - \overline{\xi}_3} =: \overline{f}_2(\overline{\xi}),\\
&\|\overline{\Phi}_3(\overline{x})\|_{\mathrm{F}} \le 0.5\left(\overline{\xi}_3^2 + \overline{\xi}_4^2\right) =: \overline{f}_3(\overline{\xi}),\\
&\|\overline{\Phi}_4(\overline{x}, \overline{\varepsilon})\|_{\mathrm{F}} \le \overline{l}\left(\overline{s}_1\sqrt{\overline{\xi}_1^2 + \overline{\xi}_4^2} + \overline{\varepsilon} + \overline{\varepsilon}_{22}\right)\overline{\xi}_4 + \overline{l}\overline{\varepsilon}_{21} =: \overline{f}_4(\overline{\xi}, \overline{\varepsilon}),
\end{aligned}$$

where $\overline{\varepsilon}_{ij} := \overline{\varepsilon}\,\overline{e}_{ij}$. We may assume that $\overline{e}_{21} > 0$, since otherwise we have the trivial case $\overline{\xi} = 0$ and $\overline{x} = 0$.

The function $\overline{f} := [\overline{f}_1, \overline{f}_2, \overline{f}_3, \overline{f}_4]^\top : \mathcal{R}_+^4 \times \mathcal{R}_+ \to \mathcal{R}_+^4$ is a Lyapunov majorant for $\overline{\Phi}$. Hence, there exists $\overline{\varepsilon}_0 > 0$ such that for $\overline{\varepsilon} \le \overline{\varepsilon}_0$ the system $\overline{\xi} = \overline{f}(\overline{\xi}, \overline{\varepsilon})$ has a continuous solution $\overline{\xi} = \overline{\xi}(\overline{\varepsilon})$, differentiable in $\overline{\varepsilon} < \overline{\varepsilon}_0$ and such that $\overline{\xi}(0) = 0$. Therefore we have the estimates

$$\|\delta\overline{U}\|_{\mathrm{F}} = \|\overline{Z}\|_{\mathrm{F}} = \sqrt{2}\,\|\overline{x}\| \le \sqrt{2}\,\|\overline{\xi}(\overline{\varepsilon})\|, \tag{6.33}$$

$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le \overline{s}\|\overline{Z}\|_{\mathrm{F}} + \overline{\varepsilon} \le \overline{s}\sqrt{2}\,\|\overline{\xi}(\overline{\varepsilon})\| + \overline{\varepsilon}. \tag{6.34}$$

The system $\overline{\xi} = \overline{f}(\overline{\xi}, \overline{\varepsilon})$ yields, unfortunately, an algebraic equation of 24th degree for $\|\overline{\xi}\|$, which is not presented here. The smallest positive root of its discriminant is $\overline{\varepsilon}_0$.

The asymptotic representation of $\|\overline{\xi}(\overline{\varepsilon})\|$ is

$$\|\overline{\xi}(\overline{\varepsilon})\| = \sqrt{2}\,\overline{l}\overline{\varepsilon}_{21} + \sqrt{2}\,\overline{l}^2\overline{\varepsilon}_{21}(\overline{\varepsilon} + \overline{\varepsilon}_{22} + \overline{l}\overline{s}_1\overline{\varepsilon}_{21}) + O(\overline{\varepsilon}^3).$$

Hence,

$$\|\delta\overline{U}\|_{\mathrm{F}} \le 2\overline{l}\overline{\varepsilon}_{21} + 2\overline{l}^2\overline{\varepsilon}_{21}(\overline{\varepsilon} + \overline{\varepsilon}_{22} + \overline{l}\overline{s}_1\overline{\varepsilon}_{21}) + O(\overline{\varepsilon}^3), \tag{6.35}$$

$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le (1 + 2\overline{l}\overline{s})\overline{\varepsilon}_{21} + 2\overline{l}^2\overline{s}\overline{\varepsilon}_{21}(\overline{\varepsilon} + \overline{\varepsilon}_{22} + \overline{l}\overline{s}_1\overline{\varepsilon}_{21}) + O(\overline{\varepsilon}^3). \tag{6.36}$$

We will bound $\overline{f}$ from above to get computable bounds. There are many ways to do this. We bound the sums $\overline{\xi}_i^2 + \overline{\xi}_j^2$ in $\overline{f}_1$, $\overline{f}_3$ and the term $\overline{\xi}_4\sqrt{\overline{\xi}_1^2 + \overline{\xi}_4^2}$ in $\overline{f}_4$ by $\|\overline{\xi}\|^2$. Assuming that $\|\overline{\xi}\| \le 1$ we have $\overline{\xi}_3 \le 0.5$ and hence $\overline{f}_3(\overline{\xi}) \le 2\overline{\xi}_4$. This gives a new majorant system for $\overline{\eta}_i \ge \overline{\xi}_i$, namely,

$$\overline{\eta}_1 = 0.5\,\|\overline{\eta}\|^2, \quad \overline{\eta}_2 = 2\overline{\eta}_4, \quad \overline{\eta}_3 = \overline{\eta}_1, \quad \overline{\eta}_4 = \overline{l}\left(\overline{s}_1\|\overline{\eta}\|^2 + \overline{\varepsilon} + \overline{\varepsilon}_{22}\overline{\eta}_4\right) + \overline{l}\overline{\varepsilon}_{21},$$

which yields a biquadratic equation for $\|\overline{\eta}\|$,

$$\left(\overline{\lambda}^2(\overline{\varepsilon}) + 10(\bar{l}\bar{s}_1)^2\right)\|\overline{\eta}\|^4 - 2\left(\overline{\lambda}(\overline{\varepsilon}) - 10\bar{l}^2\overline{s}_1\overline{\varepsilon}_{21}\right)\|\overline{\eta}\|^2 + 10(\bar{l}\overline{\varepsilon}_{21})^2 = 0, \tag{6.37}$$

with

$$\overline{\lambda}(\overline{\varepsilon}) := 1 - \bar{l}(\overline{\varepsilon} + \overline{\varepsilon}_{22}). \tag{6.38}$$

The smaller positive root of (6.37) is

$$\|\overline{\eta}(\overline{\varepsilon})\| = \frac{\sqrt{5}\,\bar{l}\overline{\varepsilon}_{21}}{\sqrt{\overline{\lambda}(\overline{\varepsilon}) - 10\bar{l}^2\overline{s}_1\overline{\varepsilon}_{21} + \sqrt{\overline{\mu}(\overline{\varepsilon})}}}, \qquad \overline{\varepsilon} \in [0, \overline{\varepsilon}_1], \tag{6.39}$$

where

$$\overline{\mu}(\overline{\varepsilon}) := \overline{\lambda}(\overline{\varepsilon})\left(\overline{\lambda}(\overline{\varepsilon})\left(1 - 10(\bar{l}\overline{\varepsilon}_{21})^2\right) - 20\,\bar{l}^2\,\overline{s}_1\overline{\varepsilon}_{21}\right). \tag{6.40}$$

Here $\overline{\varepsilon}_1 = \min\{\varepsilon', \varepsilon''\}$, where ε' is the smallest positive root of the equation $\overline{\mu}(\overline{\varepsilon}) = 0$ and ε'' is the positive root of the equation $\|\overline{\eta}(\overline{\varepsilon})\| = 1$ (if any). Thus the perturbation bounds for the block-Schur form become

$$\|\delta\overline{U}\|_{\mathrm{F}} \le \|\overline{\eta}(\overline{\varepsilon})\|, \tag{6.41}$$

$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le \overline{s}\|\overline{\eta}(\overline{\varepsilon})\| + \overline{\varepsilon}. \tag{6.42}$$

6.4. Summary of the nonlocal perturbation analysis. We summarize the results of the nonlocal perturbation analysis in the next theorem.

THEOREM 6.1. *Given a Hamiltonian matrix as in* (2.1) *and perturbations of the form* (3.1), *the nonlocal perturbation bounds for the condensed forms of Hamiltonian matrices are as follows.*

1. *For the Hamiltonian Schur form we have*

$$\begin{aligned}\|\delta U\|_{\mathrm{F}} &\le (\sqrt{2}\,\alpha_1)\varepsilon + (\sqrt{2}\,\alpha_2)\varepsilon^2 + O(\varepsilon^3),\\ \|\delta\Sigma\|_{\mathrm{F}} &\le (1 + \sqrt{2}\,s\alpha_1)\varepsilon + (\sqrt{2}\,s\alpha_2)\varepsilon^2 + O(\varepsilon^3)\end{aligned}$$

and

$$\begin{aligned}\|\delta U\|_{\mathrm{F}} &\le \sqrt{2}\,\|\eta(\varepsilon)\|,\\ \|\delta\Sigma\|_{\mathrm{F}} &\le \varepsilon + s\sqrt{2}\,\|\eta(\varepsilon)\|,\end{aligned}$$

with coefficients given by (6.9), (6.10) *and* (6.15), (6.14), *respectively.*

2. *For the Hamiltonian block-Schur form we have*

$$\begin{aligned}\|\delta\widehat{U}\|_{\mathrm{F}} &\le \sqrt{2}\,l\widehat{\varepsilon}_{21} + \sqrt{2}\,l^2\widehat{\varepsilon}_{21}(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11} + ls_1\widehat{\varepsilon}_{21}) + O(\varepsilon^3),\\ \|\delta\widehat{\Sigma}\|_{\mathrm{F}} &\le (1 + \sqrt{2}\,ls)\widehat{\varepsilon}_{21} + \sqrt{2}\,l^2s\widehat{\varepsilon}_{21}(\varepsilon/\sqrt{2} + \widehat{\varepsilon}_{11} + ls_1\widehat{\varepsilon}_{21}) + O(\varepsilon^3)\end{aligned}$$

and

$$\begin{aligned}\|\delta\widehat{U}\|_{\mathrm{F}} &\le \sqrt{2}\,\|\widehat{\eta}(\varepsilon)\|,\\ \|\delta\widehat{\Sigma}\|_{\mathrm{F}} &\le s\sqrt{2}\,\|\widehat{\eta}(\varepsilon)\| + \varepsilon,\end{aligned}$$

with coefficients given by (6.27), (6.26).

3. *For the gap we have*

$$\gamma \le \widehat{\eta}_2(\varepsilon)\sqrt{1+\widehat{\eta}_2^2(\varepsilon)}, \qquad \widehat{\eta}_2(\varepsilon) := \frac{l\left(s_1\|\widehat{\eta}(\varepsilon)\|^2+\widehat{\varepsilon}_{21}\right)}{\widehat{\lambda}(\varepsilon)}$$

with coefficients given by (6.27), (6.26).

4. *For the block-Schur form we have*

$$\|\delta\overline{U}\|_{\mathrm{F}} \le 2\bar{l}\bar{\varepsilon}_{21} + 2\bar{l}^2\bar{\varepsilon}_{21}(\bar{\varepsilon}+\bar{\varepsilon}_{22}+\bar{l}\bar{s}_1\bar{\varepsilon}_{21}) + O(\bar{\varepsilon}^3),$$
$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le (1+2\bar{l}\bar{s})\bar{\varepsilon}_{21} + 2\bar{l}^2\bar{s}\bar{\varepsilon}_{21}(\bar{\varepsilon}+\bar{\varepsilon}_{22}+\bar{l}\bar{s}_1\bar{\varepsilon}_{21}) + O(\bar{\varepsilon}^3),$$

and

$$\|\delta\overline{U}\|_{\mathrm{F}} \le \|\overline{\eta}(\bar{\varepsilon})\|,$$
$$\|\delta\overline{\Sigma}\|_{\mathrm{F}} \le \bar{s}\|\overline{\eta}(\bar{\varepsilon})\| + \bar{\varepsilon}$$

with coefficients in (6.39), (6.38), *and* (6.40).

7. Power series expansions. An alternative way to obtain nonlinear perturbation bounds for the condensed forms of Hamiltonian matrices is based on power series expansions for the perturbations in the condensed forms of Hamiltonian matrices. We begin with the Hamiltonian Schur form and derive a recurrence for the computation of the perturbed matrix $\widetilde{U}$, and from this also for $\widetilde{\Sigma}$ of $\widetilde{H}$ when the perturbation $\delta H = \varepsilon H_1$ is given.

For ε small enough the matrices Z and $\delta\Sigma$ are analytic functions of ε, vanishing together with $\varepsilon = 0$. Substituting (5.2) in (4.3) and comparing the coefficients we get the recurrence relation

$$\Sigma Z_{k+1} - Z_{k+1}\Sigma = \Sigma_{k+1} - E_1 Z_k + \sum_{i=1}^{k} Z_i \Sigma_{k+1-i},$$
$$\Sigma_k = \sum_{i=0}^{k-1} Z_i^{\mathrm{H}}(\Sigma Z_{k-i} - Z_{k-i}\Sigma + E_1 Z_{k-i-1}), \quad Z_0 := I_n, \tag{7.1}$$

where $E_1 := U^{\mathrm{H}} H_1 U$. (Note that we have already constructed the first order approximations $Z_0 = \varepsilon Z_1$ and $\delta\Sigma_0 = \varepsilon\Sigma_1$.)

Taking the low operation in the $(1,1)$ block in (7.1), using the $(2,1)$ block, and keeping in mind that $\mathbf{low}(\Sigma_{k+1}) = 0$ and that the $(2,1)$ block in Σ_{k+1} vanishes, we get

$$M\,\mathbf{lvec}(X_{k+1}) = \mathbf{lvec}(RY_{k+1} + (N_k)_{11}), \qquad \mathcal{L}(Y_{k+1}) = N_{k,21}, \tag{7.2}$$

where M is as in (4.12) and $N_{k,ij}$ are the corresponding $n \times n$ blocks of the matrix

$$N_k = N_k(Z_1, \ldots, Z_k) := -E_1 Z_k + \sum_{i=1}^{k} Z_i \Sigma_{k+1-i}\,. \tag{7.3}$$

The second equation in (7.2) has a unique solution $Y_{k+1} = \mathcal{L}^{-1}(N_{k,21})$, which is then substituted in the first equation in (7.2).

At stage $k+1$ of the recurrence (7.1) we can determine the whole matrix Y_{k+1} and the $n(n-1)/2$ elements of the lower part $\mathbf{lvec}(X_{k+1})$ of X_{k+1}. To determine the remaining elements of X_{k+1} we use (3.7). Substituting the power series expansions for $X = X(\varepsilon)$ and $Y = Y(\varepsilon)$ we get

$$X_{k+1} + X_{k+1}^{\mathrm{H}} = -\sum_{i=1}^{k} \left(X_i X_{k+1-i}^{\mathrm{H}} + Y_i Y_{k+1-i}^{\mathrm{H}} \right) =: -\Psi_k(Z_1, \ldots, Z_k), \quad k \geq 1. \tag{7.4}$$

Taking the operations **up** and **diag** on both sides of (7.4) we obtain

$$\mathbf{up}(X_{k+1}) = -(\Psi_k + \mathbf{low}(X_{k+1}))^{\mathrm{H}}, \qquad \mathbf{diag}(X_{k+1}) = -0.5\,\mathbf{diag}(\Psi_k). \tag{7.5}$$

Thus the remaining part of the matrix X_{k+1} is determined. The presented approach is justified by the following proposition.

PROPOSITION 7.1. *There exists $\varepsilon^* > 0$ such that the power series expansions* (5.2) *are convergent for $\varepsilon \in [0, \varepsilon^*)$.*

Proof. As we have shown already, the perturbation problem for the Hamiltonian Schur form with $\delta H = \varepsilon H_1$ is equivalent to the operator equation $x = \Phi(x, \varepsilon)$, where Φ is as in (6.1). We have proved in section 6.1 that for each $\varepsilon \in [0, \varepsilon_0)$ a solution $x = x(\varepsilon)$ exists such that $\|x_i(\varepsilon)\| \leq \xi_i(\varepsilon)$. Since Φ is a polynomial in x and ε (in fact quadratic in x and affine in ε), it follows that the solution $x(\varepsilon)$ is analytic in ε. Hence, it may be represented as the sum of a convergent series $\sum_{k=1}^{\infty} \varepsilon^k c_k$, $c_k \in \mathcal{C}^{2n^2}$, in the powers of ε, starting from the first power, since $x(0) = 0$. The number $\varepsilon^* > 0$ is then the radius of convergence of the power series. ∎

Similar results hold for the Hamiltonian block-Schur and block-Schur forms as well.

8. A numerical example. In this section we present a numerical example which illustrates the accuracy and applicability of the derived linear and nonlinear perturbation bounds for the Hamiltonian Schur form. All computations are done in floating-point arithmetic with round-off unit $u = 2.22 \times 10^{-16}$.

Consider a sixth order Hamiltonian matrix which is already in Hamiltonian Schur form ($H = \Sigma$) with

$$T = \begin{bmatrix} -1 & 1 & 2 \\ 0 & -2 & -1 \\ 0 & 0 & -3 \end{bmatrix}, \qquad R = \begin{bmatrix} 2 & 4 & -3 \\ 4 & -2 & 1 \\ -3 & 1 & 5 \end{bmatrix}.$$

The Hamiltonian Schur form is perturbed to

$$\widetilde{\Sigma} = \Sigma + \delta\Sigma = \begin{bmatrix} T + \delta T & R + \delta R \\ 0 & -(T + \delta T)^T \end{bmatrix},$$

where $\delta T = 10^{-4}\delta T^0$, $\delta R = 10^{-4}\delta R^0$ and

$$\delta T^0 = \begin{bmatrix} 2 & -1 & 4 \\ 0 & 1 & -3 \\ 0 & 0 & -2 \end{bmatrix}, \qquad \delta R^0 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix}.$$

An orthogonal symplectic transformation with $\widetilde{U} = \mathcal{U}[\widetilde{V}, \widetilde{W}]$ is applied, where (to 15 digits)

$$\widetilde{V} = \begin{bmatrix} 0.99999997000000 & 0.00000002999999 & -0.00000003499999 \\ 0.00000002999999 & 0.99999994500001 & 0.00000003999999 \\ -0.00000003499999 & 0.00000003999999 & 0.99999991000002 \end{bmatrix},$$

$$\widetilde{W} = \begin{bmatrix} 0.00019999998750 & -0.00009999998450 & 0.00009999998000 \\ -0.00009999998450 & 0.00029999997650 & -0.00009999997550 \\ 0.00009999998000 & -0.00009999997550 & 0.00039999995650 \end{bmatrix},$$

so that $\widetilde{H} = \widetilde{U}\widetilde{\Sigma}\widetilde{U}^T$. This allows us to compute the exact perturbations δA, δB, δC in the Hamiltonian matrix, which produce the variations δT, δR in the Hamiltonian Schur form. As a result it is obtained that

$$\delta A = \begin{bmatrix} -0.00009988990302 & 0.00119979484054 & -0.00099992986154 \\ 0.00109979486304 & -0.00099952981408 & 0.00069982485308 \\ -0.00019990001756 & 0.00009980500359 & 0.00140077982183 \end{bmatrix},$$

$$\delta B = \begin{bmatrix} 0.00030014997894 & -0.00040036493293 & -0.00029986507103 \\ -0.00040036493292 & 0.00120041984391 & -0.00000014485496 \\ -0.00029986507103 & -0.00000014485496 & 0.00229954960709 \end{bmatrix},$$

$$\delta C = \begin{bmatrix} 0.00040010994597 & -0.00050032989395 & -0.00009989008602 \\ -0.00050032989395 & 0.00140038978993 & -0.00010014984896 \\ -0.00009989008602 & -0.00010014984896 & 0.00179961969405 \end{bmatrix}.$$

For the linear approximation U_{lin} of the perturbed orthogonal symplectic transformation $\widetilde{U}$ we have that

$$\|U_{\text{lin}}^T U_{\text{lin}} - I_6\|_2 = 2.72 \times 10^{-7}, \qquad \|U_{\text{lin}} - \widetilde{U}\|_2 = 7.54 \times 10^{-7},$$

and this transformation produces a linear approximation $\Sigma_{\text{lin}} = U_{\text{lin}}^T \widetilde{H} U_{\text{lin}}$ of the perturbed Hamiltonian Schur decomposition for which

$$T_{\text{lin}} = \begin{bmatrix} -0.99980023313949 & 0.99990016105472 & 2.00040110800902 \\ 0.00000026099100 & -1.99989881036519 & -1.00029891030407 \\ -0.00000024021200 & 0.00000042978581 & -3.00020034543892 \end{bmatrix},$$

$$R_{\text{lin}} = \begin{bmatrix} 2.00009442867804 & 3.99990387841553 & -3.00000058586934 \\ 3.99990387841554 & -1.99979566286183 & 1.00009722174880 \\ -3.00000058586934 & 1.00009722174880 & 4.99990085393051 \end{bmatrix}.$$

Comparing the exact perturbed matrices $T + \delta T$, $R + \delta R$ with their linear approximations T_{lin}, R_{lin}, respectively, we see that the errors in the linear approximations are of order ε^2, as expected. Instead of being a zero 3×3 matrix, the $(2,1)$ block of the matrix Σ_{lin} has the form

$$\begin{bmatrix} -0.00000010979360 & 0.00000032951121 & -0.00000011002986 \\ 0.00000032951121 & -0.00000039017789 & 0.00000014961474 \\ -0.00000011002986 & 0.00000014961474 & 0.00000038047439 \end{bmatrix}.$$

The elements of this block are of order ε^2, and its 2-norm is 6.42×10^{-7}.

TABLE 1
Exact and estimated quantities as functions of ε.

ε	$\|\delta U\|_{\mathrm{F}}$	u_{lin}	u_{nonlin}
5.25×10^{-8}	8.37×10^{-9}	5.55×10^{-7}	5.55×10^{-7}
5.25×10^{-7}	8.37×10^{-8}	5.55×10^{-6}	5.55×10^{-6}
5.25×10^{-6}	8.37×10^{-7}	5.55×10^{-5}	5.57×10^{-5}
5.25×10^{-5}	8.37×10^{-6}	5.55×10^{-4}	5.72×10^{-4}
5.25×10^{-4}	8.37×10^{-5}	5.55×10^{-3}	$*$
5.25×10^{-3}	8.37×10^{-4}	5.55×10^{-2}	$*$
ε	$\|\delta\Sigma\|_{\mathrm{F}}$	σ_{lin}	σ_{nonlin}
5.25×10^{-8}	8.94×10^{-9}	6.87×10^{-6}	6.87×10^{-6}
5.25×10^{-7}	8.94×10^{-8}	6.87×10^{-5}	6.88×10^{-5}
5.25×10^{-6}	8.94×10^{-7}	6.87×10^{-4}	6.89×10^{-4}
5.25×10^{-5}	8.94×10^{-6}	6.87×10^{-3}	7.08×10^{-3}
5.25×10^{-4}	8.94×10^{-5}	6.87×10^{-2}	$*$
5.25×10^{-3}	8.94×10^{-4}	6.87×10^{-1}	$*$

The gap between the perturbed and original stable invariant subspaces is $\gamma = 5.21 \times 10^{-4}$. If we compute the projection onto the approximate subspace spanned by the first three columns of the approximated matrix U_{lin}, then we obtain that $\gamma_0 = 5.22 \times 10^{-4}$.

To compare the linear and nonlinear estimates for perturbations of different size and to demonstrate the quality of our bounds, we computed the exact quantities related to the perturbation analysis along with their estimates for perturbations δH constructed as described above for

$$\delta T = 10^{-10+i} T^0, \quad \delta R = 10^{-10+i} R^0, \qquad i = 1, \ldots, 6.$$

In Table 1 we give the values of $\varepsilon = \|\delta H\|_{\mathrm{F}}$ along with the values of the exact and estimated quantities. The linear bounds for $\|\delta U\|_{\mathrm{F}}$ and $\|\delta\Sigma\|_{\mathrm{F}}$ are denoted by $u_{\mathrm{lin}}, \sigma_{\mathrm{lin}}$, and the nonlinear bounds by $u_{\mathrm{nonlin}}, \sigma_{\mathrm{nonlin}}$, respectively. The cases in which the corresponding nonlinear bounds do not exist are denoted by $*$.

Note that the theoretically obtained perturbation bounds are valid for the minimal perturbations in U and Σ. To construct minimal perturbations, however, is a difficult optimization problem. In this example δU and $\delta\Sigma$ are not constructed as minimal perturbations, but we see from the numerical results that the computed perturbation bounds are correct bounds nevertheless.

9. Conclusions and future research. We have presented a complete perturbation analysis of the Hamiltonian Schur form and of two less condensed block-Schur forms for Hamiltonian matrices H. The analysis is based on the technique of splitting operators [16, 29] and Lyapunov majorants [10, 18] as well a special representation of the unitary symplectic group $\mathbf{US}(2n)$; see Appendix A. The technique for perturbation analysis of the block-Schur form of a Hamiltonian matrix is applicable to the investigation of the sensitivity of the block-Schur form $S = \left[\begin{smallmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{smallmatrix}\right]$ of an arbitrary square matrix A, whose spectrum splits into two nonempty disjoint collections.

From the perturbation results for the condensed forms and the corresponding Schur bases we have also obtained bounds for the sensitivity of the stable H-invariant subspace when H has no imaginary eigenvalues. The sensitivity of the stable H-invariant subspace may be analyzed using the results from [35]; see also [36]. Our estimates and the estimates from [35] coincide within first order terms. As nonlinear expressions, both estimates are alternative.

The Hamiltonian Schur and Hamiltonian block-Schur forms of a Hamiltonian matrix are also Hamiltonian matrices. So in these two cases we have considered structured Hamiltonian perturbations in order to preserve the Hamiltonian structure of the condensed forms for the perturbed matrices and to remain in the group of unitary symplectic transformations. This approach corresponds to the use of structure-preserving methods for transforming Hamiltonian matrices into condensed form. Here we have also assumed that the initial Hamiltonian matrix has no imaginary eigenvalues. Thus various phenomena connected with the splitting of imaginary eigenvalues under small perturbations (such as round-off errors) have not been completely analyzed. Partial results follow from [31].

The case of general unstructured perturbations of Hamiltonian matrices is covered by the analysis of the block-Schur condensed form. In this case the transformations are unitary but not necessarily symplectic. Here we deal with Hamiltonian matrices which may have imaginary eigenvalues that split into two disjoint collections.

The perturbation bounds are tighter when given in terms of the quantities ε_{ij} from (5.8), (4.1), instead of ε; see also [35]. This, however, requires a knowledge about the norms of the blocks E_{ij} of the *transformed* perturbation E, which may not be available even if the norms of the perturbations δA, δB, and δC are known. In this case one should use bounds in terms of ε such as (5.14), (5.15) which are directly deducible from the bounds based on ε_{ij}, keeping in mind that $2\varepsilon_{11}^2 + \varepsilon_{21}^2 \leq \varepsilon^2$.

Appendix A. Unitary symplectic matrices. In this appendix we present parameterizations of the group of unitary symplectic matrices. The matrices from $\mathbf{US}(2n)$ are of the form

$$U = \mathcal{U}[V, W] := \begin{bmatrix} V & W \\ -W & V \end{bmatrix} \in \mathcal{C}^{2n\times 2n}, \tag{A.1}$$

where $V, W \in \mathcal{C}^{n\times n}$ and

$$VV^{\mathrm{H}} + WW^{\mathrm{H}} = I_n, \qquad VW^{\mathrm{H}} = WV^{\mathrm{H}}. \tag{A.2}$$

Since there are n^2 independent real scalar equations in each of the equations (A.2), the real dimension of $\mathbf{US}(2n)$ is $4n^2 - 2n^2 = 2n^2$ in contrast to the real dimension of both $\mathbf{U}(2n)$ and $\mathbf{Her}(2n)$, which is $4n^2$. This observation allows us to obtain different representations of the group $\mathbf{US}(2n)$.

For $V \in \mathcal{C}^{n\times n}$ set $F(V) := (I_n + VV^{\mathrm{H}})^{-1/2}$. If $M \in \mathbf{Her}(n)$ and $N \in \mathbf{U}(n)$, then the matrix $\mathcal{U}[F(M)N, MF(M)N]$ is unitary symplectic and depends on $2n^2$ real parameters (the elements of M and N). Hence, the group $\mathbf{US}(2n)$ may be generated as

$$\mathbf{US}(2n) := \{\mathcal{U}[F(M)N, MF(M)N] : M \in \mathbf{Her}(n), N \in \mathbf{U}(n)\}. \tag{A.3}$$

(Using the dual conditions $V^{\mathrm{H}}V + W^{\mathrm{H}}W = I_n$, $W^{\mathrm{H}}V = V^{\mathrm{H}}W$ and interchanging V and W we get three more equivalent representations.)

We also use matrices of the form (A.1), for which V and W satisfy only the first equation in (A.2), but not necessarily the second. Let $\mathbf{M}_n \subset \mathcal{C}^{n\times n} \times \mathcal{C}^{n\times n}$ be the set of all pairs (V, W), such that the rows of the matrix $[V, W]$ are orthonormal. Setting $G(V) := (I_n - VV^{\mathrm{H}})^{1/2}$ it follows that

$$\mathbf{M}_n = \{(V, G(V)N) : \|V\|_2 \leq 1, N \in \mathbf{U}(n)\}, \tag{A.4}$$

and hence the real dimension of $\mathbf{M}_n$ is $3n^2$.

The set $\mathbf{S}^*(2n)$ of all matrices of the form (A.1), which satisfy the first equation in (A.2), is isomorphic to $\mathbf{M}_n$ and according to (A.4) may be represented as

$$\mathbf{S}^*(2n) := \{\mathcal{U}[V, G(V)N] : N \in \mathbf{U}(n), \|V\|_2 \leq 1\}.$$

Note that $\mathbf{US}(2n) \subset \mathbf{S}^*(2n)$, but $\mathbf{S}^*(2n)$ is not a subset of $\mathbf{S}(2n)$ and is not a (multiplicative) group, since in general $U_1, U_2 \in \mathbf{S}^*(2n)$ does not imply $U_1 U_2 \in \mathbf{S}^*(2n)$. Furthermore, $\mathbf{S}^*(2n)$ contains matrices of rank n, which is the minimum possible rank, since obviously $U \in \mathbf{S}^*(2n)$ implies $\operatorname{rank}(U) \geq n$.

Example 4. Let $U = \mathcal{U}[V, W]$, where $V = \frac{1}{\sqrt{2}}\Theta$, $W = \imath V$, and $\Theta \in \mathbf{U}(n)$. Then

$$UU^{\mathrm{H}} = \begin{bmatrix} I_n & \imath I_n \\ -\imath I_n & I_n \end{bmatrix} = \begin{bmatrix} I_n & \imath I_n \\ 0 & I_n \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -\imath I_n & I_n \end{bmatrix},$$

and hence U is of rank n.

If $U \in \mathbf{S}^*(2n)$, then $UJ_{2n}U^{\mathrm{H}} = J_{2n} + \operatorname{diag}(\Psi, \Psi)$ and

$$UU^{\mathrm{H}} = I_{2n} + \begin{bmatrix} 0 & \Psi^{\mathrm{H}} \\ \Psi & 0 \end{bmatrix},$$

where $\Psi := VW^{\mathrm{H}} - WV^{\mathrm{H}}$. Hence, we have

$$\mathbf{S}^*(2n) \cap \mathbf{S}(2n) = \mathbf{S}^*(2n) \cap \mathbf{U}(2n) = \mathbf{US}(2n).$$

In our analysis we construct matrices $U = \mathcal{U}[V, W] \in \mathbf{S}^*(2n)$ and then transform them to $\mathcal{U}[V, WN]$ with $N \in \mathbf{U}(n)$ in order to get $\mathcal{U}[V, WN] \in \mathbf{US}(2n)$. Although $\mathbf{S}^*(2n)$ contains $\mathbf{US}(2n)$ as a subset, it is not a priori clear whether for each $(V, W) \in \mathbf{M}_n$ there exists $N \in \mathbf{U}(n)$, such that $\mathcal{U}[V, WN] \in \mathbf{US}(2n)$. We show that such N always exists and give an explicit expression for N.

We can introduce an equivalence relation for matrices in $\mathbf{S}^*(2n)$ by saying that $U = \mathcal{U}[V, W]$ and $U' = \mathcal{U}[V, W']$ are *equivalent* if there exists $N \in \mathbf{U}(n)$ such that $W' = WN$. For this equivalence relation we can study the canonical sets, i.e., the set that contains exactly one representative of each equivalence class.

In the following proposition we show that $\mathbf{US}(2n)$ is a canonical set of $\mathbf{S}^*(2n)$ and give in the proof an explicit expression for the canonical form U_c, i.e., the representative of each equivalence class in $U \in \mathbf{S}^*(2n)$.

PROPOSITION A.1. *The group* $\mathbf{US}(2n)$ *is a canonical set for* $\mathbf{S}^*(2n)$.

Proof. Every matrix $U = \mathcal{U}[V, W] \in \mathbf{S}^*(2n)$ may be represented in the form $U = \mathcal{U}[V, G(V)N]$. If $VW^{\mathrm{H}} = WV^{\mathrm{H}}$, then already $U \in \mathbf{US}(2n)$ and we may set $U_c = U$. In the general case $VW^{\mathrm{H}} \neq WV^{\mathrm{H}}$ we construct $R \in \mathbf{U}(n)$ such that $U_c := \mathcal{U}[V, G(V)NR] \in \mathbf{US}(2n)$. For this purpose R must satisfy $V(WR)^{\mathrm{H}} = WRV^{\mathrm{H}}$ or $V(NR)^{\mathrm{H}}G(V) = G(V)NRV^{\mathrm{H}}$. To construct R we perform a factorization

$$V = P(V)D(V)Q^{\mathrm{H}}(V), \tag{A.5}$$

with $P(V), Q(V) \in \mathbf{U}(n)$ and $D(V)$ diagonal, but in contrast to the usual singular value decomposition [9], we do not require the diagonal elements of $D(V)$ to be nonnegative or ordered in a decreasing way (see [5] for the analysis and numerical computation of such decompositions). To obtain that the factorization is unique, the freedom in such decompositions is resolved by requiring that $P(V)Q^{\mathrm{H}}(V)$ is closest to I_n. Set

$$\begin{aligned} \mu(V) &:= \left\|P(V)Q^{\mathrm{H}}(V) - I_n\right\| \\ &= \min\left\{\left\|PQ^{\mathrm{H}} - I_n\right\| : P^{\mathrm{H}}VQ \text{ diagonal};\ P, Q \in \mathbf{U}(n)\right\}. \end{aligned} \tag{A.6}$$

The minimum is taken over a compact subset of $\mathbf{U}(n) \times \mathbf{U}(n)$ and is hence achieved. Then we obtain $G(V) = P(V)G(D(V))P^{\mathrm{H}}(V)$ and the equation for NR becomes

$$\begin{aligned} &P(V)D(V)Q^{\mathrm{H}}(V)(NR)^{\mathrm{H}}P(V)G(D(V))P^{\mathrm{H}}(V) \\ &= P(V)G(D(V))P^{\mathrm{H}}(V)NRQ(V)D(V)P^{\mathrm{H}}(V). \end{aligned}$$

Setting $NR = P(V)\Lambda Q^{\mathrm{H}}(V)$, where $\Lambda \in \mathbf{U}(n)$, it follows that Λ satisfies

$$G(D(V))\Lambda D(V) = D(V)\Lambda^{\mathrm{H}}G(D(V)). \tag{A.7}$$

Suppose that $D(V)$ has n_0 zero eigenvalues, n_1 eigenvalues of modulus 1, and k pairwise distinct eigenvalues $d_1, \ldots, d_k$ $(0 < |d_i| < 1)$ with algebraic multiplicities $\nu_1, \ldots, \nu_k$ $(n_0 + n_1 + \nu_1 + \cdots + \nu_k = n)$. Then the solution set of (A.7) is isomorphic to

$$\mathbf{U}(n_0) \times \mathbf{U}(n_1) \times \mathbf{U}(\nu_1) \cap \mathbf{Her}(\nu_1) \times \cdots \times \mathbf{U}(\nu_k) \cap \mathbf{Her}(\nu_k).$$

Generically we have $n_0 = n_1 = 0$ and $\sigma_1 = \cdots = \sigma_n = 1$, and the solutions of (A.7) are of the form $\Lambda = \mathrm{diag}(\pm 1, \ldots, \pm 1)$.

In order to get a representation which is generically unique, we choose $\Lambda = I_n$, and then we have $NR = P(V)Q^{\mathrm{H}}(V)$ and hence

$$U_c := \mathcal{U}[V, G(V)P(V)Q^{\mathrm{H}}(V)] \in \mathbf{US}(2n),$$

which finishes the proof. □

From the previous analysis we obtain the following new parametrizations of the group $\mathbf{US}(2n)$:

$$\begin{aligned} \mathbf{US}(2n) &= \left\{ \mathcal{U}\left[V, \left(I_n - VV^{\mathrm{H}}\right)^{1/2} P(V)Q^{\mathrm{H}}(V) \right] : \|V\|_2 \le 1 \right\} \\ &= \left\{ \mathcal{U}\left[\left(I_n - WW^{\mathrm{H}}\right)^{1/2} P(W)Q^{\mathrm{H}}(W), W \right] : \|W\|_2 \le 1 \right\}. \end{aligned} \tag{A.8}$$

The parametrization (A.8) is more convenient than (A.3) in analyzing the similarity action of $\mathbf{US}(2n)$ on the set of Hamiltonian matrices.

Example 5. Consider the matrix U from Example 4. We have $N = \imath\Theta$ and $P(V)Q^{\mathrm{H}}(V) = \Theta$. Hence, the canonical form of U is $U_c = \mathcal{U}[V, V] \in \mathbf{US}(2n)$, with $V = \Theta/\sqrt{2}$.

The minimal distance $\mu(V)$ in (A.6) is a characteristic of the matrix V, revealing the sensitivity of the factors P, Q in the decomposition (A.5). If V is normal, then we have $P(V) = Q(V)$ and $\mu(V) = 0$. But μ may be discontinuous for derogatory matrices V. Also it is possible that μ is continuous at a point V, where P and Q are discontinuous. Furthermore it is worth mentioning that P and Q may be less sensitive to perturbations than the matrices of left and right singular vectors in the singular value decomposition (SVD) of V. For a detailed analysis of these factorizations, see [5]. All these instructive facts are illustrated in the following examples. To compare with the standard SVD, introduce μ_0 analogous to μ as $\mu_0(V) := \left\| P_0(V)Q_0^{\mathrm{H}}(V) - I_n \right\|$, where P_0, Q_0 are those unitary factors in the SVD of V for which the minimum is achieved. Obviously $\mu(V) \le \mu_0(V)$, i.e., the diagonal decomposition is less sensitive to perturbations than the standard SVD.

Example 6. Let $V = I_2 = PDQ^{\mathrm{H}}$ with $P = D = Q = I_2$ be perturbed to $V + \varepsilon F = \left[\begin{smallmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{smallmatrix}\right]$. Then both pairs (P, Q) and (P_0, Q_0) jump from (I_2, I_2) to (Θ, Θ),

where $\Theta := (I_2 \pm J_2)/\sqrt{2}$, being discontinuous as functions of ε for $\varepsilon = 0$. At the same time $\mu(V + \varepsilon F) = \mu_0(V + \varepsilon F) = 0$, i.e., the functions $\varepsilon \mapsto \mu(V + \varepsilon F), \mu_0(V + \varepsilon F)$ are constant and hence continuous.

Example 7. Let $V = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$ be perturbed to $V + \varepsilon F = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -\varepsilon \end{smallmatrix}\right]$, $\varepsilon > 0$. Then the pair $(P, Q) = (I_2, I_2)$ remains unchanged, but the pair (P_0, Q_0) jumps from (I_2, I_2) to $(\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right], I_2)$. Hence, the function $\varepsilon \mapsto \mu(V + \varepsilon F) = 0$ is constant, but the function $\varepsilon \mapsto \mu_0(V + \varepsilon F)$ is discontinuous at $\varepsilon = 0$.

Example 8. Let $V = 0_{2\times 2}$ be perturbed to $V + \varepsilon F = \left[\begin{smallmatrix} 0 & 0 \\ \varepsilon & 0 \end{smallmatrix}\right]$, $\varepsilon > 0$. Then both pairs $(P, Q) = (P_0, Q_0)$ jump from (I_2, I_2) to $(\pm J_2, I_2)$. Hence, the functions $\varepsilon \mapsto \mu(V + \varepsilon F) = \mu_0(V + \varepsilon F)$ are discontinuous at $\varepsilon = 0$.

Appendix B. One-parameter families of perturbations. In the sensitivity analysis of the Hamiltonian Schur form of H we study the case when the minimally perturbed matrices are analytic functions in ε that vanish at $\varepsilon = 0$. Here we impose the additional assumption that H (and hence T) has distinct eigenvalues for the following reasons.

The case of multiple eigenvalues is more complicated and may lead to nonanalyticity or even discontinuity of some of the involved quantities. Indeed (see [37]), if a defective matrix $M_0 \in \mathcal{C}^{n\times n}$ is perturbed to $M_0 + \varepsilon M_1$, where ε is a small parameter, then some eigenvalues of $M_0 + \varepsilon M_1$ may depend on fractional powers $\varepsilon^{p/q}$ of ε, being nondifferentiable in $\varepsilon = 0$ and hence nonanalytical in a neighborhood of $\varepsilon = 0$. If M_0 is nonderogatory (i.e., if each eigenvalue of M_0 is involved in only one Jordan block), then the minimal perturbations in the corresponding eigenvectors and principal vectors, as well as in the Schur vectors of $M_0 + \varepsilon M_1$, will depend on $\varepsilon^{p/q}$ as well. If M_0 is derogatory, then not only the minimal perturbations in some eigenvectors and principal vectors but also the minimal perturbations in some of the Schur vectors of $M_0 + \varepsilon M_1$ may be discontinuous functions of ε. In this case the modal basis, which yields the Jordan canonical form, and the Schur basis, which yields the Schur form, have equally unpleasant behavior, and this is true also for normal matrices [27]. It should be emphasized that the Schur form is continuous as a function of the perturbations but that the Schur basis may be a discontinuous function of the same perturbations. The bases of singular vectors may also be discontinuous functions of the perturbations (see Examples 6–8).

To give a quantitative expression for the sensitivity of canonical forms, let $\mathbf{CF} \subset \mathcal{C}^{n\times n}$ be a set of condensed forms for the similarity action of a group $\Gamma \subset \mathbf{GL}(n)$ on $\mathcal{C}^{n\times n}$. To avoid trivial results let us assume that Γ is large enough, e.g., $\mathbf{U}(n) \subset \Gamma$. Then we may assume that $\mathbf{CF} \subset \mathbf{T}(n)$. For a fixed $M \in \mathcal{C}^{n\times n}$ denote by

$$\mathbf{B}(M) := \{V \in \Gamma : V^{-1}MV \in \mathbf{CF}\}$$

the set of canonical bases for M (i.e., the set of all matrices from Γ, which transform M into a condensed form), and by

$$\mathbf{C}(M) := \{V^{-1}MV : V \in B(M)\}$$

the set of condensed forms of M (in the case of canonical forms, $\mathbf{C}(M)$ should contain exactly one element). For $\mathbf{B}_1, \mathbf{B}_2 \subset \mathcal{C}^{n\times n}$ let

$$d(\mathbf{B}_1, \mathbf{B}_2) := \inf\{\|M_1 - M_2\| : M_i \in \mathbf{B}_i\}$$

be the distance between $\mathbf{B}_1$ and $\mathbf{B}_2$.

The sensitivity of the condensed forms of M may then be measured by the quantity

$$\omega(\varepsilon) := \max\{d(\mathbf{C}(M), \mathbf{C}(M+N)) : \|N\| \leq \varepsilon\}, \quad \varepsilon \geq 0.$$

Obviously $\omega(0) = 0$. For the transformation group $\Gamma = \mathbf{U}(n)$ the function $\omega : \mathcal{R}_+ \to \mathcal{R}_+$ is continuous and satisfies $\omega(\varepsilon) = O(\varepsilon^{1/k})$, $\varepsilon \to 0$, where k is the index of nilpotency of $\mathbf{up}(M_c)$ and $M_c \in \mathbf{C}(M)$; see [9].

The function ω is discontinuous for Jordan canonical forms, where $\mathbf{CF}$ is the set of bidiagonal matrices with elements 1 or 0 on the super diagonal. However, the ones in the super diagonal are introduced for purely theoretical purposes to make the form canonical under the action of the general linear group. But because of that the transformation matrices may have arbitrary large norms, which makes the standard Jordan canonical form not very suitable for computations in finite arithmetic. For computational purposes it is better to use either upper triangular forms or variants of the Jordan canonical form with no restrictions on the sizes of the elements on the super diagonal; see, e.g., [12, 13, 32] and [30].

In order to analyze the sensitivity of the transformation matrices let $N_1, N_2 \in \mathcal{C}^{n\times n}$ and define a function b via

$$b(\varepsilon) := \max_{\|N_i\| \leq \varepsilon} \min_{V_i \in B(M+N_i)} \|V_1 - V_2\|.$$

The expression $b(\varepsilon)$ may be used to analyze the sensitivity of the canonical basis V of M relative to perturbations of size ε. We have $b(0) = 0$. The function b will be discontinuous at the point $\varepsilon = 0$ for both transformation groups $\Gamma = \mathbf{U}(n)$ and $\Gamma = \mathbf{GL}(n)$ in the case when the matrix M is derogatory.

The sensitivity of Schur and Hamiltonian Schur forms is illustrated in the next three examples.

Example 9. Let the nonderogatory matrix $M = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ in Schur form $T = M$ and with Schur basis $V = I_2$ be perturbed to $\widetilde{M} = \left[\begin{smallmatrix} 0 & 1 \\ \varepsilon & 0 \end{smallmatrix}\right]$, with $\varepsilon > 0$. Then the Schur form $\widetilde{T}$ of $\widetilde{M}$ is $\widetilde{T} = \widetilde{V}^{\mathrm{H}}\widetilde{M}\widetilde{V} = \left[\begin{smallmatrix} \varepsilon^{1/2} & 1-\varepsilon \\ 0 & -\varepsilon^{1/2} \end{smallmatrix}\right]$, where $\widetilde{V} = (1+\varepsilon)^{-1/2}\left[\begin{smallmatrix} 1 & -\varepsilon^{1/2} \\ \varepsilon^{1/2} & 1 \end{smallmatrix}\right]$. For $\varepsilon \to 0$, the minimal perturbations satisfy

$$\|\widetilde{V} - I_2\|_2 = \sqrt{2}\left(1 - (1+\varepsilon)^{-1/2}\right)^{1/2} = \varepsilon^{1/2} + O(\varepsilon^{3/2}),$$
$$\|\widetilde{T} - T\|_2 = \left(\varepsilon + \varepsilon^2/2 + \left(\varepsilon + \varepsilon^2/4\right)^{1/2}\right)^{1/2} = \varepsilon^{1/2} + O(\varepsilon).$$

Hence, the sensitivities of both the Schur form and the Schur basis are of asymptotic order $\varepsilon^{1/2}$ for small ε.

Example 10. Consider the derogatory matrix $M = 0$ in Schur form $T = M$ and with Schur basis $V = I_2$. Taking $N_1 = \left[\begin{smallmatrix} 0 & 0 \\ \varepsilon & 0 \end{smallmatrix}\right]$, $N_2 = N_1^T$ with $\varepsilon > 0$, we get

$$\mathbf{B}(N_1) = \left\{ \pm \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \pm \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}, \qquad \mathbf{B}(N_2) = \left\{ \pm I_2, \pm \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right\}.$$

The distance between $\mathbf{B}(N_1)$ and $\mathbf{B}(N_2)$, measured in $\|\ \|_2$, is $\sqrt{2}$. Hence, $b(0) = 0$ and $b(\varepsilon) = \sqrt{2}$ for $\varepsilon > 0$, i.e., the function b is discontinuous at $\varepsilon = 0$, which means that the transformation matrices are infinitely sensitive. At the same time we have $\omega(\varepsilon) = \varepsilon$, i.e., the Schur form is of minimal sensitivity.

Example 11. Consider the derogatory matrix $H = \mathrm{diag}(-I_2, I_2)$ which is in Hamiltonian Schur form with Hamiltonian Schur basis $U = I_4$. Let, as in Example 10, $\delta H = \mathrm{diag}(N_1, -N_2)$. In view of (2.3) and Example 10 there are four matrices $\widetilde{U}_{1,2,3,4} = \mathrm{diag}(\pm J_2, \pm J_2)$ which transform $\widetilde{H}$ into Hamiltonian Schur form so that the perturbation $\delta U_i = \widetilde{U}_i - I_4$ in U_i is minimal. For all of them $\|\delta U_i\|_2 = \sqrt{2}$ and $\|\delta U_i\|_{\mathrm{F}} = 2\sqrt{2}$. Hence, the symplectic Schur basis of the matrix H is discontinuous due to the derogatory structure of H. Note, however, that despite this discontinuity, the stable invariant subspace is not altered by these perturbations.

Appendix C. The technique of splitting operators. The technique of splitting operators [16, 18, 29] allows us to solve efficiently the perturbation problem for Schur and generalized Schur forms, as well as other problems in linear algebra and control theory. It is based on the following idea. Let a matrix problem with nominal solution I be given and let $I + X$ be the solution of the corresponding perturbed problem. Let an invertible linear *operator* $X \mapsto \mathcal{L}(X) = LX - XL$ and a nonlinear operator $X \mapsto F(X)$ be given, where L is upper triangular and $I + X$ is unitary. Suppose that the operator equation for X is $\mathcal{L}(X) = F(X)$, together with the unitarity condition $X + X^{\mathrm{H}} + XX^{\mathrm{H}} = 0$. Then it may be split into three operator equations for the strictly lower $\mathbf{low}(X)$, diagonal $\mathbf{diag}(X)$, and strictly upper $\mathbf{up}(X)$ parts of X. Application of a fixed point principle then allows us to estimate the norms of the fixed points of the operator $\Phi(\cdot) := \mathcal{L}^{-1}(F(\cdot))$ and gives the desired perturbation bound.

Acknowledgment. The authors would like to thank the referees of this paper for their valuable comments and suggestions during the revision process.

REFERENCES

[1] G. Ammar, P. Benner, and V. Mehrmann, *A multishift algorithm for the numerical solution of algebraic Riccati equations*, Electron. Trans. Numer. Anal., 1 (1993), pp. 33–48.
[2] G. Ammar and V. Mehrmann, *On Hamiltonian and symplectic Hessenberg forms*, Linear Algebra Appl., 149 (1991), pp. 55–72.
[3] P. Benner, V. Mehrmann, and H. Xu, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, J. Comput. Appl. Math., 86 (1997), pp. 17–43.
[4] P. Benner, V. Mehrmann, and H. Xu, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., 78 (1998), pp. 329–358.
[5] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. Nichols, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numer. Math., 60 (1991), pp. 1–40.
[6] R. Byers, *A Hamiltonian QR algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[7] R. Byers, C. He, and V. Mehrmann, *The matrix sign function method and the computation of invariant subspaces*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 615–632.
[8] F. Gantmacher, *Theory of Matrices*, Vol. 1, Chelsea, New York, 1959.
[9] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[10] E. Grebenikov and Y. Ryabov, *Constructive Methods for Analysis of Nonlinear Systems*, Nauka, Moscow, 1979.
[11] G. Hewer and C. Kenney, *The sensitivity of the stable Lyapunov equation*, SIAM J. Control Optim., 26 (1988), pp. 321–344.
[12] B. Kågström and A. Ruhe, *Algorithm* 560: *JNF, an algorithm for numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, 6 (1980), pp. 437–443.
[13] B. Kågström and A. Ruhe, *An algorithm for numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, 6 (1980), pp. 398–419.
[14] L. V. Kantorovich and G. P. Akilov, *Functional Analysis in Normed Spaces*, Pergamon, New York, 1964.

[15] C. Kenney and A. Laub, *The matrix sign function*, IEEE Trans. Automat. Control, 40 (1995), pp. 1330–1348.
[16] M. Konstantinov, P. Petkov, and N. Christov, *Nonlocal perturbation analysis of the Schur system of a matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 383–392.
[17] M. Konstantinov, P. Petkov, D. Gu, and I. Postlethwaite, *Perturbation Techniques for Linear Control Problems*, Tech. Report 95-7, Department of Engineering, Leicester University, Leicester, UK, 1995.
[18] M. Konstantinov, P. Petkov, D. Gu, and I. Postlethwaite, *Perturbation Analysis in Finite Dimensional Spaces*, Tech. Report 96–18, Department of Engineering, Leicester University, Leicester, UK, 1996.
[19] M. Konstantinov, M. Stanislavova, and P. Petkov, *Perturbation bounds and characterisation of the solution of the associated algebraic Riccati equation*, Linear Algebra Appl., 285 (1998), pp. 7–31.
[20] P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*, Oxford University Press, Oxford, UK, 1995.
[21] W.-W. Lin and T.-C. Ho, *On Schur Type Decompositions for Hamiltonian and Symplectic Pencils*, Tech. Report, Institute of Appl. Math., Nat. Tsing Hua University, Taiwan, 1990.
[22] 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.
[23] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Heidelberg, Germany, 1991.
[24] V. Mehrmann and H. Xu, *Lagrangian Invariant Subspaces of Hamiltonian Matrices*, Preprint SFB393/98-25, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, Chemnitz, Germany, 1998.
[25] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[26] C. Paige and C. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 14 (1981), pp. 11–32.
[27] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[28] R. V. Patel, Z. Lin, and P. Misra, *Computation of stable invariant subspaces of Hamiltonian matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 284–298.
[29] P. Petkov, N. Christov, and M. Konstantinov, *A new approach to the perturbation analysis of linear control problems*, in Proceedings of the 11th IFAC Congress, Tallinn, Estonia, Pergamon Press, Oxford, 1990, pp. 311–316.
[30] P. Petkov, N. Christov, and M. Konstantinov, *Computational Methods for Linear Control Systems*, Prentice-Hall, Hemel Hempstead, Herts, UK, 1991.
[31] A. Ran and L. Rodman, *Stability of invariant maximal semidefinite subspaces*. I, Linear Algebra Appl., 62 (1984), pp. 51–86.
[32] A. Ruhe, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.
[33] H. Shapiro, *A survey of canonical forms and invariants under similarity*, Linear Algebra Appl., 147 (1991), pp. 101–167.
[34] V. Sima, *Algorithms for Linear Quadratic Optimization*, Marcel Dekker, New York, 1996.
[35] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[36] G. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[37] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.
[38] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 425–442
© 2001 Society for Industrial and Applied Mathematics

A BACKWARD ERROR ANALYSIS OF A NULL SPACE ALGORITHM IN SPARSE QUADRATIC PROGRAMMING*

MARIO ARIOLI† AND LUCIA BALDINI‡

Abstract. We present a roundoff error analysis of a null space method for solving quadratic programming minimization problems. This method combines the use of a direct *LU* factorization of the constraints with an iterative solver on the corresponding null space. Numerical experiments are presented which give evidence of the good performance of the algorithm on sparse matrices.

Key words. augmented systems, sparse matrices, Gaussian factorization, roundoff

AMS subject classifications. 65F05, 65F10, 64F25, 65F50, 65G05

PII. S0895479800375977

1. Introduction.

Let $M \in \mathbb{R}^{n\times n}$ be a symmetric and positive semidefinite matrix, and let $A \in \mathbb{R}^{n\times m}, m \le n$, be a real full rank matrix, $q \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. The quadratic programming problem with equality constraints

$$\min_{A^T x = b} \frac{1}{2} x^T M x + q^T x \tag{1.1}$$

has a unique solution $\hat{x}$ if and only if $\mathrm{Ker}(A^T) \cap \mathrm{Ker}(M) = \{0\}$. Introducing the vector $u \in \mathbb{R}^m$ of the Lagrangian parameters the problem (1.1) is equivalent to the augmented system

$$\begin{bmatrix} M & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}. \tag{1.2}$$

The augmented matrix is invertible, and the solution $[\hat{x}^T, \hat{u}^T]^T$ is composed of the solution of problem (1.1) and the Lagrangian parameters of the gradient of the objective function at $\hat{x}$.

In this paper, we present a roundoff error analysis of a null space method which uses a mixture of direct and iterative solvers. The direct solver is based on the *LU* factorization of the matrix A. Similar error analyses are discussed in [8, 19] for the least squares problems, and in [13] for a different algorithm for quadratic programming problems. The use of iterative algorithms is not taken into account in [8] or [13]. In [4, 7, 21] similar algorithms are proposed for the constrained least squares problems: these papers focus on the convergence properties of the iterative methods and use block partitioning of $A^T = [S^T; N^T]$, where S^T is assumed square and nonsingular, to compute a null space basis. In [3] the error analysis is presented when the QR algorithm is used to factorize A.

In section 2 we present the details of the algorithm, and in section 3 we analyze its roundoff error propagation. In section 4 we introduce upper bounds for the error affecting the computed solution $\overline{x}$ and the computed Lagrangian parameters $\overline{u}$. It

*Received by the editors August 1, 2000; accepted for publication (in revised form) by Z. Strakoš January 22, 2001; published electronically September 7, 2001.
http://www.siam.org/journals/simax/23-2/37597.html
†Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, OX11 0QX, UK (m.arioli@rl.ac.uk).
‡Cap Gemini Ernst & Young, Via Torino 68, 20123 Milano, Italy (Lucia.Baldini@it.eyi.com).

should be pointed out that the augmented system (1.2) also gives the solution to the (dual) least squares problem

$$\min_u \frac{1}{2}(Au+q)^T M^{-1}(Au+q) + b^T u. \tag{1.3}$$

Thus, the results given in sections 3 and 4 are equally valid for this problem.

In section 5, we show the results of the numerical tests that we conducted on selected experiments, and in section 6 we give our conclusions.

In the following, we will denote the augmented matrix by $\mathfrak{A}$, and by E_1 and E_2 the matrices

$$E_1 = \begin{bmatrix} I_m \\ 0_{n-m,m} \end{bmatrix} \qquad \text{and} \qquad E_2 = \begin{bmatrix} 0_{m,n-m} \\ I_{n-m} \end{bmatrix}.$$

Let $fl(\cdot)$ denote the result of a floating point computation. We assume that

$$fl(\alpha \square \beta) = (\alpha \square \beta)(1+\delta(\square,\alpha,\beta)); \quad |\delta(\square,\alpha,\beta)| \le \varepsilon, \tag{1.4}$$

where α and β are floating point numbers, ε is the machine precision, and $\square$ is one of $+ - * /$. To a great extent modern computers have arithmetic that satisfies assumption (1.4) with the exception of some CRAY computers (e.g., CRAY2 and CRAY-YMP). Furthermore, we assume that the scalar products are accumulated using either extended precision arithmetic or the Kahan summation formula [22, 9] (for more details about these techniques we refer to [15, 20]). As a consequence of these assumptions, given x and y real vectors of dimension n, we have

$$fl(x^T y) = x^T y + x^T D y + s, \qquad |D| \le 3\varepsilon I, \quad |s| \le \mathcal{O}(n\varepsilon^2 |x|^T |y|).$$

Given an $n \times m$ matrix B of entries B_{ij} and an n-vector v of entries v_i, we will denote by $|B|$ and $|v|$ the matrix and the vector whose entries are the absolute values of the entries of B and v. Finally, for the sake of simplicity we will denote by $c_i,\ i = 0, 1, \ldots$, the constants that will be used in the expressions of the roundoff errors.

2. Algorithms. In this section, we take into account the null space classical algorithm, which is described in [14], for the solution of (1.1).

We choose a formulation of this algorithm which is based on the factorization of the augmented matrix, and in the next section we will show how this formulation enables us to give a roundoff error analysis of the algorithm.

The matrix A can be factorized in the following way:

$$PAQ = L \begin{bmatrix} U \\ 0 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_2 & I \end{bmatrix} \begin{bmatrix} U \\ 0 \end{bmatrix}, \tag{2.1}$$

where $U \in \mathbb{R}^{m\times m}$ is an upper triangular matrix, $L \in \mathbb{R}^{n\times n}$ is a nonsingular lower triangular matrix generated by the Gaussian algorithm applied to PAQ, and P, Q are permutation matrices that cope with numerical pivoting and sparsity [10, 17, 28]. For the sake of simplicity, we will omit P, Q in the following, assuming that M, A, q, and b have been consistently permuted. Let

$$\tilde{M} = L^{-1} M L^{-T}.$$

The augmented matrix $\mathfrak{A}$ can be factorized in the following way:

$$\mathfrak{A} = \begin{bmatrix} L & 0 \\ 0 & U^T \end{bmatrix} \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} & I_m \\ \tilde{M}_{12}^T & \tilde{M}_{22} & 0 \\ I_m & 0 & 0 \end{bmatrix} \begin{bmatrix} L^T & 0 \\ 0 & U \end{bmatrix}.$$

Let $Z = L^{-T}E_2$. The matrix Z is a nonorthonormal basis of the kernel of A^T. On the basis of the previous discussion, we obtain the following algorithm.

NULL SPACE ALGORITHM.

$$\begin{bmatrix} h \\ v \end{bmatrix} = \begin{bmatrix} -L^{-1}q \\ U^{-T}b \end{bmatrix},$$
$$h_1 = E_1^T h,$$
$$h_2 = E_2^T h.$$

Solve the block lower triangular system

$$\begin{bmatrix} I_m & 0 & 0 \\ \tilde{M}_{12}^T & \tilde{M}_{22} & 0 \\ \tilde{M}_{11} & \tilde{M}_{12} & I_m \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} v \\ h_2 \\ h_1 \end{bmatrix} \tag{2.2}$$

and let

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = L^{-T} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix},$$
$$u = U^{-1}z_3.$$

It follows directly from (2.1) that $x_2 = z_2$. Frequently, the product $L^{-1}ML^{-T}$ cannot be performed directly because the complexity would be too high ($\mathcal{O}(n^3)$) or because the resulting matrix would be fairly dense despite the sparsity of M. Nevertheless, in some cases the product can be performed successfully, obtaining a sparse result. Alternatively, in solving the triangular system (2.2), we can perform the product of a submatrix of $\tilde{M}$ by a vector in the following way:

$$\tilde{M}_{ij}y = E_i^T L^{-1}(M(L^{-T}E_j y)) \qquad \text{with } (i,j) = (1,2). \tag{2.3}$$

This approach has the advantage of performing backward and forward substitution for triangular matrices.

In the null space algorithm, we also need to solve $\tilde{M}_{22}x_2 = p$. We have two alternative ways of proceeding. If $n - m$ is small (the number of constraints is very close to the number of unknowns) or $\tilde{M}_{22}$ is still sparse, we can explicitly compute $\tilde{M}_{22}$ using the algorithm (2.3) and then solve the system using the Cholesky factorization. Otherwise, we can solve the linear system $\tilde{M}_{22}x_2 = p$ using a conjugate gradient algorithm (see [18]) without explicitly computing $\tilde{M}_{22}$ and using (2.3) to perform the matrix by vector products.

3. Roundoff error analysis. In the following we will denote the computed values of the corresponding variables by $\bar{z_1}$, $\bar{z_2}$, $\bar{z_3}$, $\bar{x}$, and $\bar{u}$. The roundoff properties of the Gaussian elimination with partial pivoting (GEPP) are very well known. In [28] and [20, Chapter 9] it is shown that the computed $\bar{L}$ and $\bar{U}$ satisfy the following equation:

$$(A + \delta A) = \bar{L} \begin{bmatrix} \bar{U} \\ 0 \end{bmatrix} \tag{3.1}$$

with

$$|\delta A| \leq c_1 m\varepsilon(|A| + |\bar{L}|E_1|\bar{U}|) + \mathcal{O}(\varepsilon^2). \tag{3.2}$$

We now use the previous properties to analyze the roundoff in the algorithm (2.3) for the matrix vector product. First of all, we have [20, Chapter 3]

$$fl(My) = (M+G)y \quad \text{with } |G| \leq 3\varepsilon|M| + \mathcal{O}(\varepsilon^2)$$

and

$$fl(\bar{L}^{-1}y) = (\bar{L}+\delta L_1)^{-1}y \quad \text{with } |\delta L_1| \leq 3\varepsilon|\bar{L}|E_1E_1^T + \mathcal{O}(\varepsilon^2),$$

$$fl(\bar{L}^{-T}y) = (\bar{L}+\delta L_2)^{-T}y \quad \text{with } |\delta L_2| \leq 3\varepsilon|\bar{L}|E_1E_1^T + \mathcal{O}(\varepsilon^2).$$

Therefore, it follows that for $i, j = 1, 2$,

$$\begin{aligned} fl(\tilde{M}_{ij}y) &= E_i^T fl(\bar{L}^{-1} fl(M fl(\bar{L}^{-T}E_j y))) \\ &= E_i^T(\bar{L}+\delta L_1)^{-1}(M+G)(\bar{L}+\delta L_2)^{-T}E_j y. \end{aligned} \tag{3.3}$$

If we assume that

$$\varepsilon\|L\|_\infty\|L^{-1}\|_\infty \ll 1, \tag{3.4}$$

we can linearize the expressions and obtain

$$fl(\tilde{M}_{ij}y) = E_i^T\bar{L}^{-1}(M+\mathcal{G}^{ij})\bar{L}^{-T}E_j y, \tag{3.5}$$

where

$$\mathcal{G}^{ij} = G - \delta L_1\bar{L}^{-1}M - M\bar{L}^{-T}\delta {L_2}^T + \mathcal{O}(\varepsilon^2)$$

and

$$|\mathcal{G}^{ij}| \leq c_3\varepsilon(|M| + |\bar{L}||\widehat{M}||\bar{L}^T| + \mathcal{O}(\varepsilon^2)), \quad \widehat{M} = \bar{L}^{-1}M\bar{L}^{-T}. \tag{3.6}$$

In the following, $\widehat{M}_{ij} = E_i^T\widehat{M}E_j$ with $(i,j) = (1,2)$, and we will assume that the matrix $\widehat{M}_{22}$ is invertible. We can now analyze the following variants of the null space algorithm.

Variant A. If we can afford both the complexity and the storage necessary to the explicit computation of $\tilde{M}_{22}$, then we can invert the diagonal block in (2.2) by the Cholesky factorization algorithm, which computes a lower triangular matrix R such that $\tilde{M}_{22} = RR^T$. Thus, following the analysis given by Wilkinson [28] (see also [20, Chapter 10]), we prove that $\overline{R}$, the computed value of R, and $fl(\tilde{M}_{22})$, the computed value of $\tilde{M}_{22}$, satisfy the following relations:

$$\begin{aligned} fl(\tilde{M}_{22}) &= E_2^T\bar{L}^{-1}(M+\mathcal{G}^{22})\bar{L}^{-T}E_2, \\ |\mathcal{G}^{22}| &\leq c_3\varepsilon(|M| + |\bar{L}||\widehat{M}||\bar{L}^T| + \mathcal{O}(\varepsilon^2)), \end{aligned} \tag{3.7}$$

$$\begin{aligned} \overline{R}\overline{R}^T &= E_2^T\bar{L}^{-1}(M+\mathcal{G}^{22})\bar{L}^{-T}E_2 + E_2^T\bar{L}^{-1}\bar{L}E_2G_4E_2^T\bar{L}^T\bar{L}^{-T}E_2 \\ &= E_2^T\bar{L}^{-1}(M+\overline{\mathcal{G}^{22}})\bar{L}^{-T}E_2 \end{aligned} \tag{3.8}$$

with

$$|G_4| \leq (n-m)\varepsilon|\overline{R}||\overline{R}^T|, \tag{3.9}$$

$$|\overline{\mathcal{G}^{22}}| \leq c_4(n-m)\varepsilon(|M| + |\bar{L}||\widehat{M}||\bar{L}^T| + E_2|\overline{R}||\overline{R}^T|E_2^T) + \mathcal{O}(\varepsilon^2). \tag{3.10}$$

Variant B. Alternatively, we can solve the system $\tilde{M}_{22}x_2 = h_2 - \tilde{M}_{12}^T z_1$ using an iterative method.

Variant B1. If we incorporate within the iterative method a stopping criterion based on the a posteriori componentwise backward error theory [20, Chapter 7], then a stopping criterion such as

$$\text{IF} \quad |\widehat{M}_{22}\overline{x}_2^{(j)} - fl(h_2 - \widehat{M}_{12}^T z_1)| \le \eta |E_2^T M E_2||\overline{x}_2^{(j)}| \quad \text{THEN STOP}, \tag{3.11}$$

with $\eta < 1$ an a priori threshold fixed by the user, will guarantee that the computed solution $\bar{x}_2 = \overline{x}_2^{(j)}$ satisfies the perturbed linear system

$$(\widehat{M}_{22} + E_{22})\overline{x}_2 = fl(h_2 - \widehat{M}_{12}^T \bar{z}_1), \qquad |E_{22}| \le \eta |E_2^T M E_2|.$$

Moreover, we can project the error on the null space such that we have

$$(E_2^T \bar{L}^{-1}(M + \mathcal{E})\bar{L}^{-T} E_2)\overline{x}_2 = h_2 - fl(\widehat{M}_{12}^T \bar{z}_1) \tag{3.12}$$

with

$$|\mathcal{E}| \le \eta(1+\varepsilon)(E_2E_2^T|M|E_2E_2^T) + \varepsilon E_2E_2^T|\bar{L}^{-1}||M| + \mathcal{O}(\varepsilon^2). \tag{3.13}$$

The choice of η will depend on the properties of the problem that we want to solve, and, in the practical cases, η can be frequently much larger than ε.

Variant B2. If we use the conjugate gradient method, it is quite natural to have a stopping criterion which takes advantage of the minimization property of this method. At each step j the conjugate gradient algorithm minimizes the energy norm of the error $\delta x_2 = x_2 - \bar{x}_2^{(j)}$ on a Krylov space. The space $\mathbb{R}^{n-m}$ with the norm

$$\|y\|_{\tilde{M}_{22}} = (y^T \tilde{M}_{22} y)^{(1/2)}$$

induces the dual norm

$$\|f\|_{\tilde{M}_{22}^{-1}} = (f^T \tilde{M}_{22}^{-1} f)^{(1/2)}$$

for its dual space. Therefore, a stopping criterion such as

$$\text{IF} \quad \|\widehat{M}_{22}\overline{x}_2^{(j)} - \tilde{h}\|_{\widehat{M}_{22}^{-1}} \le \eta \|\tilde{h}\|_{\widehat{M}_{22}^{-1}} \quad \text{THEN STOP}, \tag{3.14}$$

where $\tilde{h} = fl(h_2 - \widehat{M}_{12}^T \bar{z}_1)$, will guarantee [2] that the computed solution $\bar{x}_2 = \overline{x}_2^{(j)}$, which satisfies it, solves the perturbed linear system

$$\begin{aligned} \widehat{M}_{22}\overline{x}_2 &= \tilde{h} + f, \\ \|f\|_{\widehat{M}_{22}^{-1}} &\le \eta \|\tilde{h}\|_{\widehat{M}_{22}^{-1}}. \end{aligned}$$

Moreover, because $Z^T E_2 = I$, we can project the error on the null space as follows:

$$(E_2^T \bar{L}^{-1} M \bar{L}^{-T} E_2)\overline{x}_2 = \tilde{h} + E_2^T \bar{L}^{-1} \delta q \tag{3.15}$$

with

$$\delta q = E_2 f, \quad \|f\|_{\widehat{M}_{22}^{-1}} \le \eta \|\tilde{h}\|_{\widehat{M}_{22}^{-1}} \le \eta \|\overline{x}_2\|_{\widehat{M}_{22}} + \mathcal{O}(\eta^2). \tag{3.16}$$

Using the previous results, we can now prove the following theorem.

THEOREM 3.1. *Let $\bar{x}$ and $\bar{u}$ be the values of x and u, solutions of (2), computed with the null space algorithm. If $\varepsilon\|\bar{L}^{-1}\|_\infty\|\bar{L}\|_\infty \ll 1$, then there exist the matrices $\delta M \in \mathbb{R}^{n\times n}$ and $\delta A_1,\ \delta A_2 \in \mathbb{R}^{n\times m}$ and the vector $\delta q \in \mathbb{R}^n$ such that*

$$\begin{bmatrix} M+\delta M & A+\delta A_1 \\ (A+\delta A_2)^T & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} -(q+\delta q) \\ b \end{bmatrix}.$$

Furthermore, we have

$$\begin{aligned} |\delta A_1| &\le c_2 m\varepsilon(|A| + |\bar{L}|E_1|\bar{U}|) + \mathcal{O}(\varepsilon^2), \\ |\delta A_2| &\le c_2 m\varepsilon(|A| + |\bar{L}|E_1|\bar{U}|) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

If we use an iterative solver with (3.11) and a threshold η (Variant B1),

$$\begin{aligned} |\delta M| &\le c_6\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T)\mathcal{H}^T + \eta E_2E_2^T|M|E_2E_2^T + \mathcal{O}(\varepsilon^2), \\ \delta q &= 0, \end{aligned}$$

where $\mathcal{H} = I + E_2E_2^T|\bar{L}^{-1}|E_1E_1^T$.

If we use an iterative solver with (3.14) and a threshold η (Variant B2),

$$|\delta M| \le c_6\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T)\mathcal{H}^T + \mathcal{O}(\varepsilon^2),$$

$$\|\delta q\|_{E_2\widehat{M}_{22}^{-1}E_2^T} \le \eta\|\overline{x}_2\|_{\widehat{M}_{22}} + \mathcal{O}(\eta^2).$$

Otherwise, if we build $\tilde{M}_{22}$ and then factorize and solve the linear system by Cholesky factorization (Variant A), we have $\delta q = 0$ and

$$|\delta M| \le c_4(n-m)\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T + E_2|\overline{R}||\overline{R}|^TE_2^T)\mathcal{H}^T + \mathcal{O}(\varepsilon^2).$$

Proof. From [20, Theorem 8.5, p. 154] it follows that the computed values $\bar{x}$, $\bar{z_1}$, $\bar{z_2}$, $\bar{z_3}$, and $\bar{u}$ satisfy the equations

$$(\bar{U} + \delta U_1)\bar{u} = \bar{z}_3, \quad (\bar{L} + \delta L_3)^T\bar{x} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}, \tag{3.17}$$

where

$$|\delta U_1| \le 3\varepsilon|\bar{U}| + \mathcal{O}(\varepsilon^2), \tag{3.18}$$

$$|\delta L_3| \le 3\varepsilon|\bar{L}|E_1E_1^T + \mathcal{O}(\varepsilon^2). \tag{3.19}$$

Moreover, the vector $\bar{z}$ satisfies the system

$$\begin{bmatrix} \widehat{M}_{11}+\delta\widehat{M}_{11} & \widehat{M}_{12}+\delta\widehat{M}_{12} & I_m \\ \widehat{M}_{12}^T+\delta\widehat{M}_{21} & \widehat{M}_{22}+\delta\widehat{M}_{22} & 0 \\ I_m & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \\ \bar{z}_3 \end{bmatrix} = \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{v} \end{bmatrix}, \tag{3.20}$$

where $\widehat{M} = \bar{L}^{-1}M\bar{L}^{-T}$,

$$\delta\widehat{M}_{ij} = E_i^T\bar{L}^{-1}\mathcal{G}^{ij}\bar{L}^{-T}E_j, \quad (i,j) \ne (2,2),$$

and

$$\delta\widehat{M}_{22} = E_2^T \bar{L}^{-1} \overline{\mathcal{G}^{22}} \bar{L}^{-T} E_2,$$

or

$$\delta\widehat{M}_{22} = E_2^T \bar{L}^{-1} \mathcal{E} \bar{L}^{-T} E_2.$$

Once more, from [20, Theorem 8.5, p. 154] it follows that the computed values $\bar{h}$ and $\bar{v}$ satisfy

$$(\bar{U} + \delta U_2)^T \bar{v} = b, \quad (\bar{L} + \delta L_4)\bar{h} = -q, \tag{3.21}$$

where

$$|\delta U_2| \le 3\varepsilon|\bar{U}| + \mathcal{O}(\varepsilon^2), \quad |\delta L_4| \le 3\varepsilon|\bar{L}|E_1E_1^T + \mathcal{O}(\varepsilon^2). \tag{3.22}$$

Finally, combining (3.21), (3.20), and (3.17) with (3.1) we have

$$\begin{bmatrix} M + \delta M & A + \delta A_1 \\ (A + \delta A_2)^T & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix},$$

where

$$\delta M = \sum_{i,j=1,2} \bar{L} E_i \delta\widehat{M}_{ij} E_j^T \bar{L}^T,$$

and

$$\delta A_1 = \delta A + \delta L_4 E_1 \bar{U} + \bar{L} E_1 \delta U_1 + \mathcal{O}(\varepsilon^2), \quad \delta A_2 = \delta A + \delta L_3 E_1 \bar{U} + \bar{L} E_1 \delta U_2 + \mathcal{O}(\varepsilon^2).$$

The matrix δM can be split into two contributions: $\delta M = \delta M_1 + \delta M_2$, where

$$\delta M_2 = E_2 \delta\widehat{M}_{22} E_2^T.$$

Moreover, defining

$$\mathcal{H} = \begin{bmatrix} I & 0 \\ E_2^T|\bar{L}^{-1}|E_1 & I \end{bmatrix},$$

we point out that

$$|\bar{L} E_i E_i^T \bar{L}^{-1}| \le \mathcal{H}, \quad i = 1, 2.$$

If we choose to invert the diagonal block $\tilde{M}_{22}$ by the Cholesky factorization, the $|\delta M_2|$ can be bounded using (3.10) as follows:

$$|\delta M_2| \le c_5(n-m)\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T + E_2|\overline{R}||\overline{R}^T|E_2^T)\mathcal{H}^T + \mathcal{O}(\varepsilon^2). \tag{3.23}$$

Alternatively, using an iterative solver with (3.11), we deduce from (3.12) and (3.13) that

$$|\delta M_2| \le \eta E_2 E_2^T |M| E_2 E_2^T + 2\varepsilon\mathcal{H}|M|\mathcal{H}^T + \mathcal{O}(\varepsilon^2). \tag{3.24}$$

We can now evaluate the perturbations by the use of (3.1), (3.6), (3.18), (3.19), (3.22) and, alternatively, either of (3.23) or of (3.24):

$$|\delta A_1| \le c_2 m\varepsilon(|A| + |\bar{L}|E_1|\bar{U}|) + \mathcal{O}(\varepsilon^2),$$
$$|\delta A_2| \le c_2 m\varepsilon(|A| + |\bar{L}|E_1|\bar{U}|) + \mathcal{O}(\varepsilon^2),$$

and either

$$|\delta M| \le c_4(n-m)\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T + E_2|\overline{R}||\overline{R}^T|E_2^T)\mathcal{H}^T + \mathcal{O}(\varepsilon^2)$$

or

$$|\delta M| \le c_6\varepsilon\mathcal{H}(|M| + |\bar{L}||\widehat{M}||\bar{L}|^T)\mathcal{H}^T + \eta(E_2E_2^T|M|E_2E_2^T + \mathcal{O}(\varepsilon^2)).$$

Finally, using (3.14), the result follows straightforwardly from (3.15) and (3.16). ∎

4. Forward error. If we denote by $A^- = U^{-1}E_1^TL^{-1}$ a generalized inverse of A (A^- is not a Moore–Penrose pseudoinverse) [5, Chapter 6], by $Z = L^{-T}E_2$ the nonorthogonal basis for $\mathrm{Ker}(A^T)$, and by $\mathcal{P} = I - MZ\tilde{M}_{22}^{-1}Z^T$ the oblique projection onto $span(A)$ along $span(MZ)$ [23, section 5.8], it is easy to verify by direct computation that

$$\mathfrak{A}^{-1} = \begin{bmatrix} Z\tilde{M}_{22}^{-1}Z^T & \mathcal{P}^T(A^-)^T \\ A^-\mathcal{P} & -A^-\mathcal{P}M(A^-)^T \end{bmatrix}. \tag{4.1}$$

Following the results of section 3, we are able to represent the errors $\delta x = \overline{x} - x$ and $\delta u = \overline{u} - u$ as follows:

$$\begin{bmatrix} \delta x \\ \delta u \end{bmatrix} = \mathfrak{A}^{-1}\begin{bmatrix} -\delta A_1\overline{u} - \delta M\overline{x} - \delta q \\ -\delta A_2^T\overline{x} \end{bmatrix}.$$

Using the block form of the inverse of $\mathfrak{A}$ (4.1) and the results of Theorem 3.1, we obtain componentwise bounds which, in general, have a limited practical use when large sparse matrices are involved.

We can see from these bounds that the solution x is less sensitive than the Lagrangian parameters u to the perturbations in the data, in agreement with the analogous results that we have for the least squares problems (see [8]). Nevertheless, when we use, for instance, Variants B1 and B2 of the algorithm, the perturbation δM is the sum of two contributions,

$$\delta M = \delta M_1 + E_2\delta\widehat{M}_{22}E_2^T,$$

where $\delta\widehat{M}_{22}$ depends on the threshold η and δM_1 depends on the roundoff unit ε.

As normally $\varepsilon \ll \eta$, it is appropriate to analyze the influence of the perturbation δM on the error δx neglecting the part depending on ε. More precisely, we will assume for Variants B1 and B2 that

$$\left\|\mathfrak{A}^{-1}\begin{bmatrix} -\delta A_1\overline{u} - \delta M_1\overline{x} \\ -\delta A_2^T\overline{x} \end{bmatrix}\right\|_\infty < \left\|\mathfrak{A}^{-1}\begin{bmatrix} -E_2\delta\widehat{M}_{22}E_2^T\overline{x} - \delta q \\ 0 \end{bmatrix}\right\|_\infty.$$

Thus, for Variant B1, we have

$$\begin{bmatrix} \delta x \\ \delta u \end{bmatrix} = \begin{bmatrix} Z\tilde{M}_{22}^{-1}Z^T\delta\widehat{M}_{22}\overline{x}_2 \\ A^-\mathcal{P}E_2\delta\widehat{M}_{22}\overline{x}_2 \end{bmatrix} + \mathcal{O}(\varepsilon).$$

Let us denote the following:

$$\begin{aligned} M_{22} &= E_2^T M E_2, \\ D &= \operatorname{diag}(|M_{22}||\overline{x}_2|), \\ \mathfrak{K}_A &= \|A^- \mathcal{P} E_2 D\|_\infty, \\ \mathfrak{C}_M &= \|\tilde{M}_{22}^{-1} D\|_\infty. \end{aligned}$$

Therefore, the errors can be bounded as follows:

$$\begin{bmatrix} |\delta x| \\ |\delta u| \end{bmatrix} \leq \eta \begin{bmatrix} |Z\tilde{M}_{22}^{-1}||M_{22}||\overline{x}_2| \\ |A^- \mathcal{P} E_2||M_{22}||\overline{x}_2| \end{bmatrix} + \mathcal{O}(\varepsilon).$$

If we split $\delta x = [\delta x_1^T; \delta x_2^T]^T$, the norms can be bounded as follows:

$$\begin{bmatrix} \|\delta x_1\|_\infty \\ \|\delta x_2\|_\infty \\ \|\delta u\|_\infty \end{bmatrix} \leq \eta \begin{bmatrix} \|Z\|_\infty \mathfrak{C}_M \\ \mathfrak{C}_M \\ \mathfrak{K}_A \end{bmatrix} + \mathcal{O}(\varepsilon).$$

Taking into account the expressions of A^- and $\mathcal{P}$, we have

$$\mathfrak{K}_A = \|U^{-1}\tilde{M}_{12}\tilde{M}_{22}^{-1} D\|_\infty.$$

Both $\mathfrak{C}_M$ and $\mathfrak{K}_A$ can be estimated using the LAPACK norm estimator [1], which estimates the $\|B\|_\infty$ of a matrix B given only the ability to form matrix-vector products By and $B^T y$.

The case of Variant B2, when the conjugate gradient algorithm is used in conjunction with (3.14), deserves more attention.

First of all, because we cannot explicitly have $\tilde{M}_{22}$ but only its approximation $\widehat{M}_{22}$, we can give estimates only for the norms $\|\cdot\|_{\tilde{M}_{22}}$ and $\|\cdot\|_{\tilde{M}_{22}^{-1}}$.

Furthermore, we need to add within the conjugate gradient algorithm some tool for estimating the value $e^{(j)}_{\widehat{M}_{22}} = f^{(j)T}\widehat{M}_{22}^{-1} f^{(j)}$, $f^{(j)} = \widehat{M}_{22}\overline{x}_2^{(j)} - \tilde{h}$ at each step (j). This can be achieved using a Gauss quadrature rule as proposed in [16]. In particular, this variant of the conjugate gradient algorithm produces a lower bound ξ_j for $e^{(j)}_{\widehat{M}_{22}}$. In [16] the authors also propose other quadrature rules which, when used within the conjugate gradient algorithm, give lower and upper bounds for $e^{(j)}_{\widehat{M}_{22}}$. Unfortunately, these techniques require guesses for the smallest and the biggest eigenvalues of $\widehat{M}_{22}$, which are not cheaply available for our matrix. Nevertheless, as suggested in [16] the Gauss-quadrature-based lower bound can be made reasonably close to the value of $e^{(j)}_{\widehat{M}_{22}}$ at the price of d additional steps of the conjugate gradient algorithm. Therefore, ξ_j will be the estimate of $e_A^{(j-d)}$. In [16], $d = 10$ is indicated as a successful compromise, and our numerical experiments support this conclusion.

Finally, we also need an estimation for $\tilde{h}^T\widehat{M}_{22}^{-1}\tilde{h}$. Taking into account that

$$| \, \|\tilde{h}\|_{\widehat{M}_{22}^{-1}} - \|\overline{x}_2^{(j)T}\|_{\widehat{M}_{22}} \, | \leq \sqrt{e^{(j)}_{\widehat{M}_{22}}}, \tag{4.2}$$

we can choose to replace $\tilde{h}^T\widehat{M}_{22}^{-1}\tilde{h}$ with the current evaluation of $\overline{x}_2^{(j)T}\widehat{M}_{22}\overline{x}_2^{(j)}$ at step (j) of the conjugate gradient algorithm if $e^{(j)}_{\widehat{M}_{22}} < \eta^2\tilde{h}^T\tilde{h}$. Therefore, we can use (3.14)

only after an additional check:

$$\begin{array}{l}\text{IF } \sqrt{\xi_j} \le \eta\,\|\tilde{h}\|_2 \text{ THEN}\\ \quad\text{IF } \sqrt{\xi_j} \le \eta\,\|\overline{x}_2^{(j)}\|_{\widehat{M}_{22}} \text{ THEN STOP}\\ \text{ENDIF}\end{array} \tag{4.3}$$

As before, we can give expressions for the errors δx and δu neglecting the part depending on ε:

$$\begin{bmatrix} \delta x \\ \delta u \end{bmatrix} = \begin{bmatrix} Z\tilde{M}_{22}^{-1}Z^T E_2 f \\ A^{-}\mathcal{P}E_2 f \end{bmatrix}.$$

As we bound the backward error f in energy norm, it is natural to bound some appropriate energy norms of δx and δu. From the expressions of problems (1.1) and (1.3) the natural choices are

$$\|\delta x\|_M = ((\delta x)^T M \delta x)^{1/2}, \tag{4.4}$$

$$\|\delta u\|_{A^T M^{-1} A} = ((\delta u)^T A^T M^{-1} A \delta u)^{1/2}. \tag{4.5}$$

By a direct substitution of the expression of δx in (4.4) we have

$$\|\delta x\|_M = \|f\|_{\widehat{M}_{22}^{-1}} + \mathcal{O}(\varepsilon) \le \eta \|\overline{x}_2\|_{\widehat{M}_{22}}.$$

Furthermore, observing that

$$A\delta u = \mathcal{P}E_2 f$$

and

$$\mathcal{P}^2 = \mathcal{P},$$

we have

$$\begin{aligned}(\delta u)^T A^T M^{-1} A \delta u &= f^T E_2^T \mathcal{P}^T M^{-1} \mathcal{P} E_2 f\\ &= f^T E_2^T M^{-1} \mathcal{P} E_2 f\\ &= f^T E_2^T M^{-1} E_2 f - f^T \tilde{M}_{22}^{-1} f\\ &\le f^T \widehat{M}_{22}^{-1/2} (\widehat{M}_{22}^{1/2} E_2^T M^{-1} E_2 \widehat{M}_{22}^{1/2}) \widehat{M}_{22}^{-1/2} f.\end{aligned}$$

Therefore, we can bound $\|\delta u\|_{A^T M^{-1} A}$ as follows:

$$\|\delta u\|_{A^T M^{-1} A} \le \eta \|\overline{x}_2\|_{\widehat{M}_{22}} (\rho(\widehat{M}_{22} E_2^T M^{-1} E_2))^{1/2}, \tag{4.6}$$

where $\rho(B)$ of a square matrix B is the spectral radius of B.

5. Numerical experiments. In our numerical experiments, we used augmented systems obtained from the modeling of electrical networks. For the sake of simplicity, we will focus our attention on Variant B2 of the null space algorithm to show the effectiveness of the analysis given in sections 3 and 4 for this problem. Hereafter, we give a short description of the problem; more details can be found in [26, Chapter 2].

5.1. Model problem. An electrical network can be described by a directed graph $\mathcal{EN}$ with n nodes n_j, $j = 1, \ldots, n$, and m edges e_{ij}, and Ohm's law and Kirchhoff's law, which give the relations between the electric currents I_{ij}, the voltage drops V_j, and the resistances of the edges R_{ij}. The edge-node incidence matrix A of the graph $\mathcal{EN}$ is a totally unimodular matrix (see [24] for more details) with entries $(-1, 0, 1)$. Denoting the vector of the potentials at the nodes by u and the vector of electric currents through the edges by x, we can describe Kirchhoff's law by the equation

$$A^T x = b,$$

where b is the vector of the current sources at the nodes. Finally, Ohm's law is described by the equation

$$Mx + Au = -q,$$

where $M = \text{diag}(R_{ij})$ and $-q$ is the vector of the voltage sources.

We assume in our test examples that the units of measure are chosen such that the values of the entries in q and b are between -1 and 1.

5.2. Algebraic problem. First of all, we want to call to mind some useful basic definitions relative to the direct graph $\mathcal{EN}$ with $m + 1$ nodes and n edges. A *path* in a graph from node m_1 to node m_k is a list of nodes $[m_1, m_2, \ldots, m_k]$ such that (m_i, m_{i+1}) is an edge in the graph $\mathcal{EN}$ for $i = 1, \ldots, k-1$. The path *contains* node m_i for $i \in [1, \ldots, k]$ and edge (m_i, m_{i+1}) for $i \in [1, \ldots, k]$ and *avoids* all other nodes and edges. Nodes m_1 and m_k are the *ends* of the path. The path is *simple* if all its nodes are distinct. The path is a *cycle* if $k > 1$ and $m_1 = m_k$ and is a *simple cycle* if all its nodes are distinct. A graph without cycles is *acyclic*. If there is a path from node w to node v, then v is *reachable* from w. A graph is *connected* if every node of its undirected version is reachable from every other node.

A *rooted tree* is an undirected graph that is connected and acyclic with a distinguished node r, called *root*. A rooted tree with k nodes contains $k - 1$ edges and has a unique simple path from any node to any other. When appropriate we shall regard the edges of a rooted tree as directed. A *rooted spanning tree* $\mathcal{T}_r$ in $\mathcal{EN}$ is a rooted tree which is a subgraph of $\mathcal{EN}$ with $m + 1$ nodes. We will call the edges in the tree *in-tree* and the others *out-of-tree*.

In an electric network $\mathcal{EN}$ a node r must be chosen as ground (with potential zero, $u_r = 0$) and then its corresponding column must be eliminated from the incidence matrix of the graph. In this way the resulting matrix A is full rank [24, 26]. Choosing r as root it is always possible to find a rooted spanning tree $\mathcal{T}_r$ in $\mathcal{EN}$. We refer to [24, 27] for surveys of different algorithms for computing spanning trees. If we renumber the in-tree edges first and the out-of-tree edges last we can permute the in-tree edges and the nodes such that [24] the permuted A has the form

$$PAQ = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix},$$

where $L_1 \in \mathbb{R}^{m \times m}$ is a lower triangular and nonsingular matrix.

As the matrix A is totally unimodular, the matrix L_1^{-1} is also a matrix with entries $-1, 0, 1$. Moreover, the matrix $L_2 L_1^{-1}$ has entries $-1, 0, 1$ and its rows correspond to the out-of-tree edges. The number of nonzeros in one of its rows will be the number of

TABLE 1
Dimension and number of nonzeros for A, q, and b of each problem.

Prob. n.	m	n	$nz(A)$	$nz(q)$	$nz(b)$
1	2977	7495	14982	3	1
2	2969	7494	14982	3	1
3	2979	7504	15000	3	1
4	2975	7498	14991	3	1
5	2980	7499	14989	3	1
6	99	246	489	2	1
7	99	246	487	2	1
8	99	240	474	2	1
9	99	237	467	2	1
10	99	245	481	2	1
11	98	242	480	2	1
12	99	245	482	2	1
13	99	243	480	2	1
14	98	242	479	2	1
15	99	238	471	2	1

edges in the cycle of minimal length which the corresponding out-of-tree edge forms with the in-tree edges.

As a consequence of these properties, the LU factorization of matrices describing Kirchhoff's laws and, in general, of totally unimodular matrices is obtainable without floating point operations:

$$PAQ = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} = LE_1. \tag{5.1}$$

In this case, the roundoff error analysis of section 3 can be marginally improved and, in Theorem 3.1, we have the following slightly tighter bounds:

$$|\delta A_1| \leq c_2 \varepsilon |A|\,, \qquad |\delta A_2| \leq c_2 \varepsilon |A|\,.$$

5.3. Test problems. We use two sets of test problems. For the first set we randomly generated 10 small networks with a number of nodes of order 100, and 5 bigger networks with a number of nodes of order 3000. Each node is connected on average with 5 others, and the graph is connected. We fixed the node 0 as the ground node and we eliminated it from the incidence matrix A such that A has full rank.

In Table 1 we report the dimensions and the number of nonzeros in each matrix A and in each vector q and b. The second set consists of six power networks obtained from the Harwell–Boeing collection of matrices [11, 12]. We fixed an external root in node 0 for each network, we connected it with node 1, and we randomly generated the q and b vectors using the Matlab function **SPRAND(size,dens)** with **dens** $= 5/n$ for q and **dens** $= 1/n$ for b. In Table 2 we report the dimensions and the number of nonzeros in each matrix A and in each vector q and b.

For both sets we used a breadth-first search strategy to find the spanning tree with root r, and we assumed that the matrix A and the vectors q and b had been permuted in agreement with (5.1).

The matrix M is chosen such that

$$M = \begin{bmatrix} \alpha I_m & 0 & 0 & 0 \\ 0 & \beta I_{\lceil (n-m)/2 \rceil} & 0 & 0 \\ 0 & 0 & \gamma I_{\lfloor (n-m)/2 \rfloor - 1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

TABLE 2
Dimension and number of nonzeros for A, q, and b of each problem.

Harwell–Boeing name	Prob. n.	m	n	$nz(A)$	$nz(q)$	$nz(b)$
bcspwr07	16	1612	2107	4213	4	1
bcspwr08	17	1624	2214	4427	7	1
bcspwr09	18	1723	2395	4789	5	1
bcspwr10	19	5300	8272	16543	7	1
eris1176	20	1176	8690	17378	33	1

For each problem we generated two matrices M: for the first one we chose $\alpha = 1$, $\beta = 10^{-3}$, and $\gamma = 10^{-3}$; for the second we chose $\alpha = 10^{-6}$, $\beta = 10^{-4}$, and $\gamma = 1$.

5.4. Numerical results. For all the test problems we assumed the exact solution to be that computed by a direct solver: our direct solver computes the normal equations, factors them by the Cholesky algorithm, and, from the computed Lagrangian parameters u, computes the solution x. This assumption satisfies our purposes because we use a conjugate gradient method with a threshold much higher than ε in the null space algorithm.

In our run, we used a variant of the stopping criterion suggested in [16] to evaluate the backward error in energy norm during the conjugate gradient algorithm. As already discussed in section 3, we are unable to supply any lower or upper bounds on the eigenvalues of the projected Hessian matrix $E_2^T \tilde{M} E_2$ without a big computational cost. Therefore, we were able to use only the estimator based on the Gauss quadrature formula, which gives only a lower bound estimate of the dual norm of the residual f. Moreover, we wanted to avoid the many additional matrix-vector products necessary for the evaluation of $\tilde{h}^T \widehat{M}_{22}^{-1} \tilde{h}$ at each iteration. Therefore, we decided to modify (4.3), introducing two thresholds. We checked at each step (j) whether the estimate ξ_j of the value $e^{(j)}_{\widehat{M}_{22}}$ was less than $10^{-6} \overline{x}_2^{(j)T} \widehat{M}_{22} \overline{x}_2^{(j)}$, but only after $\xi_j \leq 10^{-8} \tilde{h}^T \tilde{h}$. As we mentioned in section 3, at iteration k of the conjugate gradient algorithm, we can estimate the backward error relative to the iteration $k - d$. We chose $d = 10$ in our test examples as suggested in [16]. In Figures 1 and 2, we plotted the ratio between estimated value at iteration $k - 10$ over the energy M-norm of the error between the solution computed at iteration k and the exact solution.

Analyzing the two figures, we can see that when the conjugate gradient algorithm converges fast, our lower bound of the error at iteration $k - 10$ becomes an upper bound for the true error relative to iteration k. Nevertheless, in the cases when the convergence is not fast, the lower bound gives more than reasonable estimates.

In Tables 3 and 4, we compare the complexity of the proposed method and the number of iterations performed by the conjugate gradient algorithm with the complexity of the direct solver. For the larger test examples in set 1, even when the conjugate gradient algorithm converges slowly, our algorithm requires less floating point operations than the direct solver. For the test examples in set 2, our algorithm is not competitive even when the conjugate gradient algorithm performs few iterations. Without any doubt the choice of a diagonal M represents a serious drawback for our method. Nevertheless, when n is not much larger than m, it is still competitive.

In Figures 3 and 4, we plot the relative errors in M-norm between the computed $\overline{x}$ and the solution computed by the direct method x,

$$((\overline{x} - x)^T M (\overline{x} - x))^{1/2} / (x^T M x)^{1/2},$$

for both choices of M. The results are consistent with the thresholds we have chosen

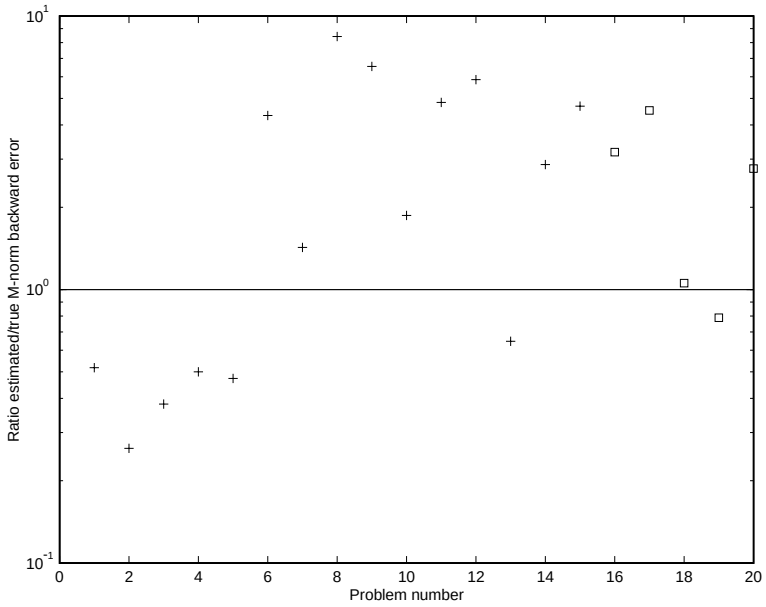


FIG. 1. *Ratio estimated backward error over true backward error ($\alpha = 1$, $\beta = 10^{-3}$, and $\gamma = 10^{-3}$).*

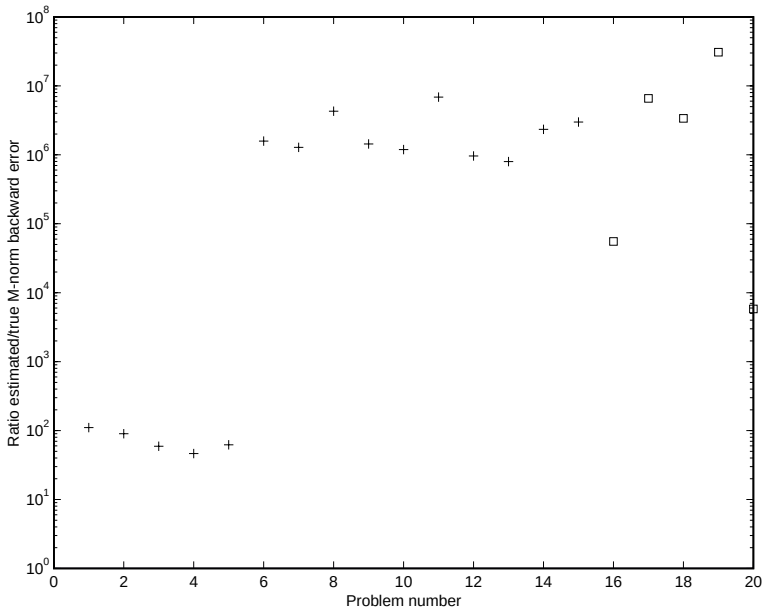


FIG. 2. *Ratio estimated backward error over true backward error ($\alpha = 10^{-6}$, $\beta = 10^{-4}$, and $\gamma = 1$).*

for the backward error. Moreover, when the conjugate gradient algorithm converges fast these errors are quite satisfactory. Finally, in Figures 5 and 6, we plot the errors on the Lagrangian parameters measured in the $A^T M^{-1} A$-norm. All the errors are

TABLE 3
Conjugate gradient floating point operations and number of iterations vs. direct solver floating point numbers ($\alpha = 1$, $\beta = 10^{-3}$, and $\gamma = 10^{-3}$).

Prob. num.	CG flops	iter.	dir. sol. flops
1	107314005	576	225244402
2	134831456	684	226935420
3	124145327	647	225892004
4	151407987	813	231680452
5	129431803	695	229054528
6	767530	124	24834
7	558864	90	24291
8	779981	131	22087
9	684317	117	22623
10	450914	73	24196
11	774722	128	24499
12	603367	98	21238
13	446926	73	21574
14	537206	88	23316
15	434045	73	21467
16	1285283	48	64771
17	1949670	64	68866
18	4033672	112	79141
19	44339011	303	332418
20	47080200	181	618637

TABLE 4
Conjugate gradient floating point operations and number of iterations vs. direct solver floating point numbers ($\alpha = 10^{-6}$, $\beta = 10^{-4}$, and $\gamma = 1$).

Prob. num.	CG flops	iter.	dir. sol. flops
1	8097857	42	225244402
2	7736113	40	226935412
3	8472034	44	225892016
4	7715806	40	231680456
5	8088062	42	229054528
6	144122	22	24834
7	144132	22	24295
8	139497	22	22083
9	129453	21	22623
10	142415	22	24196
11	152802	24	24499
12	142959	22	21238
13	153040	24	21574
14	153462	24	23316
15	137378	22	21467
16	387168	13	63971
17	442674	13	68266
18	530973	13	78730
19	2126403	13	329662
20	7553392	28	620004

consistent with the upper bound (4.6).

6. Conclusions. We have proved the backward stability of the null space algorithm for solving augmented systems. Moreover, we have given evidence of the good applicability of the method in the sparse case. We reserve for future work the analysis of the preconditioning techniques proposed by Nash and Sofer in [25]. We want to emphasize here that our test problems are significant examples of the more general class of problems, such as the mixed finite element discretization of elliptic problems

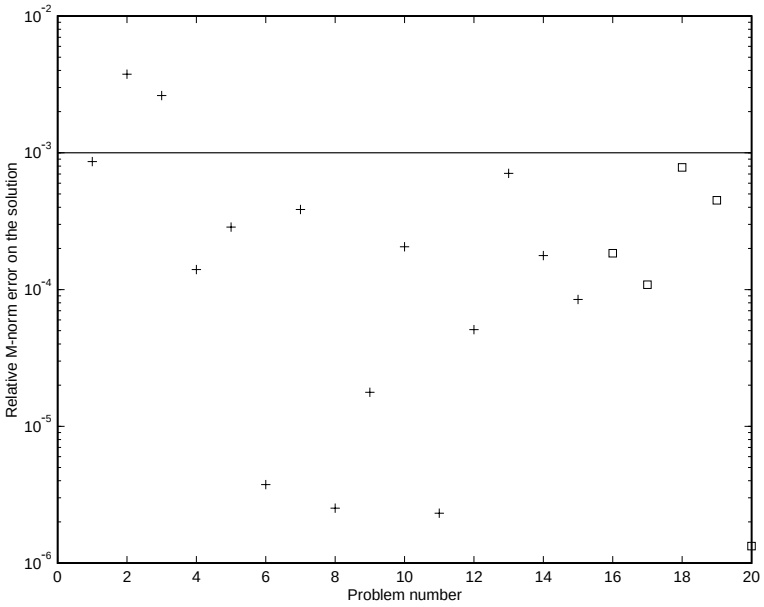


FIG. 3. *M-norm relative error on x* ($\alpha = 1$, $\beta = 10^{-3}$, *and* $\gamma = 10^{-3}$).

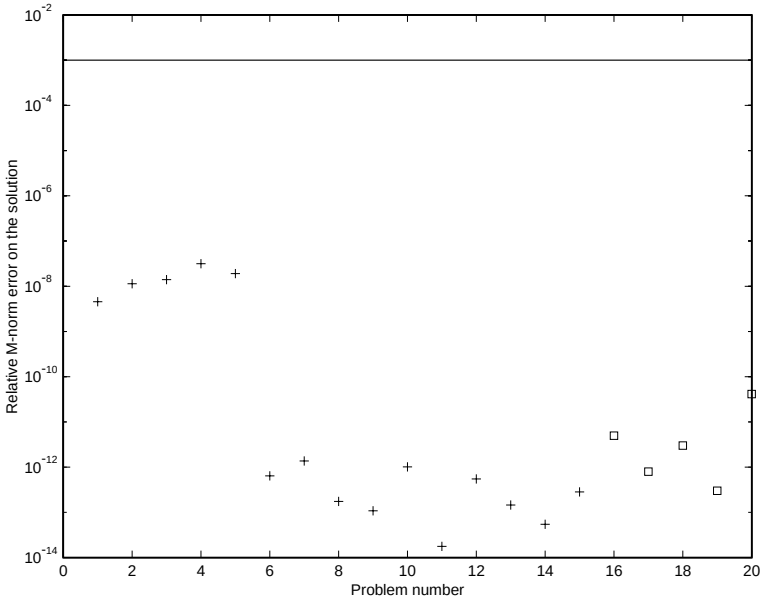


FIG. 4. *M-norm relative error on x* ($\alpha = 10^{-6}$, $\beta = 10^{-4}$, *and* $\gamma = 1$).

in saddle point formulation.

Finally, the method we propose could also be generalized to the solution of nonlinear problems with linear equality constraints, where it would be possible to build a specialized version of the conjugate gradient taking advantage of sparse LU factorizations.

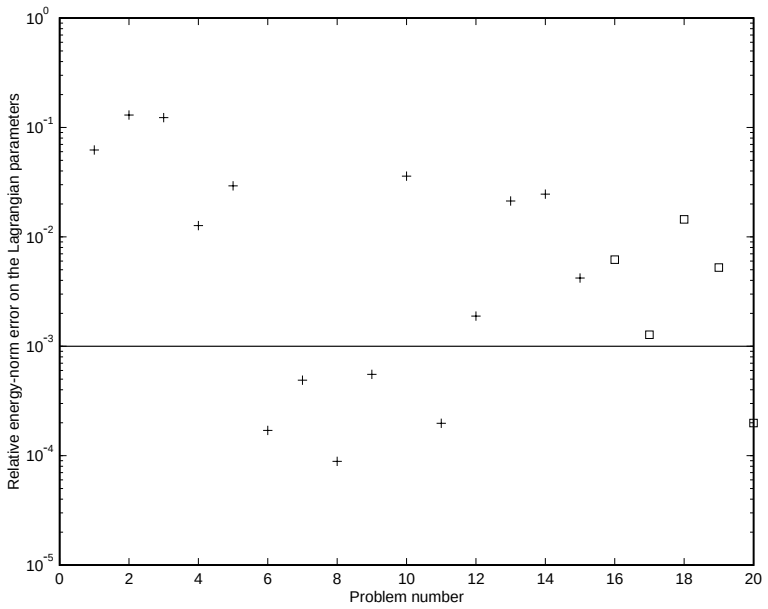


FIG. 5. *$A^T M^{-1} A$-norm relative error on u ($\alpha = 1$, $\beta = 10^{-3}$, and $\gamma = 10^{-3}$).*

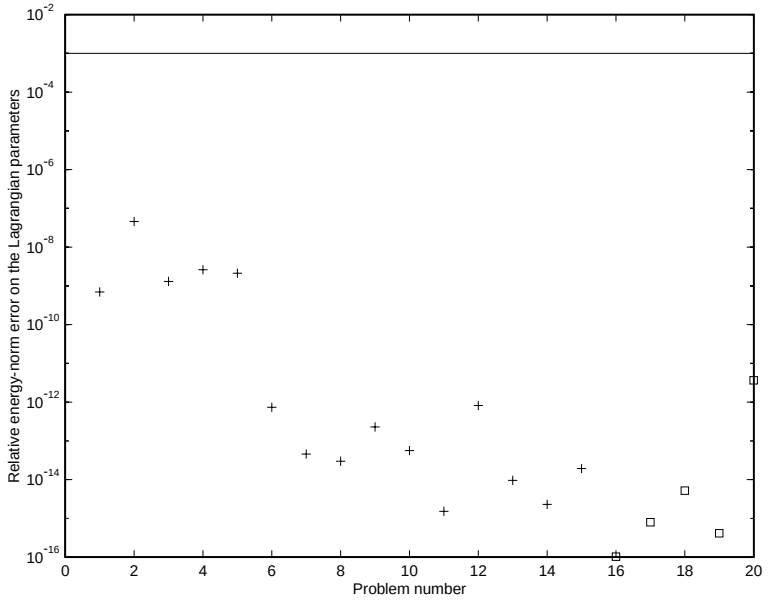


FIG. 6. *$A^T M^{-1} A$-norm relative error on u ($\alpha = 10^{-6}$, $\beta = 10^{-4}$, and $\gamma = 1$).*

Acknowledgments. The authors would like to thank the referees for their valuable comments. In particular, we are indebted to Dr. Miro Rozloznik for his careful reading, which helped us to amend the final version.

REFERENCES

[1] E. Anderson, Z. Bai, C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen, *LAPACK Users' Guide, Release* 2.0, 2nd ed., SIAM, Philadelphia, PA, 1995.
[2] M. Arioli, E. Noulard, and A. Russo, *Vector stopping criteria for iterative methods: Applications to PDE's*, Calcolo, 38 (2001), pp. 97–112.
[3] M. Arioli, *The use of QR factorization in sparse quadratic programming and backward error issues*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 825–839.
[4] J. L. Barlow, N. K. Nichols, and R. J. Plemmons, *Iterative methods for equality-constrained least squares problems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 892–906.
[5] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Pitman, San Francisco, 1979.
[6] T. F. Coleman, A. Edenbrant, and J. R. Gilbert, *Predicting fill for sparse orthogonal factorization*, J. ACM, 33 (1986), pp. 517–532.
[7] T. F. Coleman and A. Pothen, *The null space problem* I. *Complexity*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 527–537.
[8] A. J. Cox and N. J. Higham, *Accuracy and stability of the null space method for solving the equality constrained least squares problem*, BIT, 39 (1999), pp. 34–50.
[9] T. J. Dekker, *A floating point technique for extending the available precision*, Numer. Math., 18 (1971), pp. 224–242.
[10] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Monographs on Numerical Analysis, Oxford University Press, New York, 1989.
[11] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users' Guide for the Harwell–Boeing Sparse Matrix Collection (Release* I*)*, Tech. Rep. TR/PA/92/86, CERFACS, Toulouse, France, 1992.
[12] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[13] E. Galligani and L. Zanni, *Error analysis in equality constrained quadratic optimization problems*, Boll. Un. Mat. Ital. A (7), 11 (1997), pp. 595–611.
[14] P. H. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, UK, 1981.
[15] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, Comput. Surveys, 23 (1991), pp. 5–48.
[16] G. H. Golub and G. Meurant, *Matrices, moments and quadrature.* II. *How to compute the norm of the error in iterative methods*, BIT, 37 (1997), pp. 687–705.
[17] G. H. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[18] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, PA, 1997.
[19] M. Gulliksson, *Algorithms for Overdetermined System of Equations*, Thesis UMINF 93–11, Department of Computing Science, Institute of Information Processing, University of Umeå, Umeå, Sweden, 1993.
[20] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.
[21] D. James, *Order reducing conjugate gradients versus block AOR for constrained least-squares problems*, Linear Algebra Appl., 154/156 (1991), pp. 23–43.
[22] W. Kahan, *Further remarks on reducing truncation errors*, Comm. ACM, 8 (1965), pp. 1–40.
[23] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, New York, 1985.
[24] K. G. Murthy, *Network Programming*, Prentice–Hall, Englewood Cliffs, NJ, 1992.
[25] S. G. Nash and A. Sofer, *Preconditioning reduced matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 47–68.
[26] G. Strang, *Introduction to Applied Mathematics*, Wellesley, Cambridge, MA, 1986.
[27] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM, Philadelphia, PA, 1983.
[28] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 443–458
© 2001 Society for Industrial and Applied Mathematics

COMPUTING SYMMETRIC RANK-REVEALING DECOMPOSITIONS VIA TRIANGULAR FACTORIZATION*

PER CHRISTIAN HANSEN† AND PLAMEN Y. YALAMOV‡

Abstract. We present a family of algorithms for computing symmetric rank-revealing VSV decompositions based on triangular factorization of the matrix. The VSV decomposition consists of a middle symmetric matrix that reveals the numerical rank in having three blocks with small norm, plus an orthogonal matrix whose columns span approximations to the numerical range and null space. We show that for semidefinite matrices the VSV decomposition should be computed via the ULV decomposition, while for indefinite matrices it must be computed via a URV-like decomposition that involves hypernormal rotations.

Key words. rank-revealing decompositions, matrix approximation, symmetric matrices, hypernormal rotations

AMS subject classifications. 65F30, 65F35

PII. S0895479800370068

1. Introduction.

Rank-revealing decompositions of general dense matrices are widely used in signal processing and other applications where accurate and reliable computation of the numerical rank, as well as the numerical range and null space, are required. The singular value decomposition (SVD) is certainly a decomposition that reveals the numerical rank, but what we have in mind here are the RRQR (rank-revealing QR) and UTV (i.e., URV and ULV) decompositions which can be computed and, in particular, updated more efficiently than the SVD. See, e.g., [7, sections 2.7.5–2.7.7], [20, section 2.2], and [34, Chapter 5] for details and references to theory, algorithms, and applications.

The key to the efficiency of UTV algorithms is that they consist of an initial triangular factorization which can be tailored to the particular matrix, followed by an efficient rank-revealing postprocessing step. If the matrix is $m \times n$ with $m \geq n$ and with numerical rank k, then the initial triangular factorization requires $\mathcal{O}(mn^2)$ flops, while the rank-revealing step requires only $c\,(n-k)n^2$ flops if $k \approx n$, and $c\,kn^2$ flops if $k \ll n$, where c is an algorithm-dependent constant. The updating can always be done in $\mathcal{O}(n^2)$ flops, when implemented properly. The same is true for some RRQR algorithms, while others have a higher complexity. We refer to the original papers [9], [10], [16], [18], [19], [23], [32], [33] for details about the algorithms.

For structured matrices (e.g., Hankel and Toeplitz matrices), the initial triangular factorization in the RRQR and UTV algorithms has the same complexity as the rank-revealing step, namely, $\mathcal{O}(mn)$ flops; see [7, section 8.4.2] for signal processing aspects. However, accurate principal singular values and vectors can also be computed by means of Lanczos methods in the same complexity, $\mathcal{O}(mn)$ flops [13]. Hence the

*Received by the editors April 4, 2000; accepted for publication (in revised form) by I. Ipsen February 20, 2001; published electronically September 7, 2001.

http://www.siam.org/journals/simax/23-2/37006.html

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

‡Center of Applied Mathematics and Informatics, University of Rousse, 7017 Rousse, Bulgaria (yalamov@ami.ru.acad.bg). The work of this author was supported by a fellowship from the Danish Rectors' Conference and by grants MM-707/97 and I-702/97 from the National Scientific Research Fund of the Bulgarian Ministry of Education and Science.

advantage of a rank-revealing decomposition depends on the matrix structure and the numerical rank of the matrix.

Rank-revealing decompositions of general *sparse* matrices are also in use, for example, in optimization and geometric design [28]. For sparse matrices, the initial pivoted triangular factorization can exploit the sparsity of A. However, the UTV postprocessors may produce a severe amount of fill, while the fill in the RRQR postprocessor is restricted to lie in the columns that are permuted to the right of the triangular factor [7, Theorem 6.7.1]. An alternative sparse URL decomposition $A = U\,R\,L$, where U is orthogonal and R and L are upper and lower triangular, respectively, was proposed in [27]. This decomposition can be computed with less fill at the expense of working with only one orthogonal matrix.

Numerically rank-deficient *symmetric* matrices also arise in many applications, notably in signal processing and in optimization algorithms (cf. section 16.2 in [25]). In both areas, fast computation and efficient updating are key issues, and sparsity is also an issue in some optimization problems. Symmetric rank-revealing decompositions enable us to compute symmetric rank-deficient matrix approximations (obtained by neglecting blocks in the rank-revealing decomposition with small norm). This is important, for example, in rank-reduction algorithms in signal processing where one wants to compute rank-deficient symmetric semidefinite matrices. In addition, utilization of symmetry leads to faster algorithms, compared to algorithms for nonsymmetric matrices.

In spite of this, very little work has been done on symmetric rank-revealing decompositions. Luk and Qiao [24] introduced the term *VSV decomposition* and proposed an algorithm for symmetric indefinite Toeplitz matrices, while Baker and DeGroat [2] presented an algorithm for symmetric semidefinite matrices.

The purpose of this paper is to put the work in [2] and [24] into a broader perspective by surveying possible rank-revealing VSV decompositions and algorithms, including the underlying theory. Our emphasis is on algorithms which, in addition to revealing the numerical rank, provide accurate estimates of the numerical range and null space. We build our algorithms on existing methods for computing rank-revealing decompositions of triangular matrices, based on orthogonal transformations. Our symmetric decompositions and algorithms inherit the properties of these underlying algorithms which are well understood today.

We emphasize that the goal of this paper is not to present detailed implementations of our VSV algorithms, but rather to set the stage for such implementations. The papers [4] and [29] clearly demonstrate that careful implementations of efficient and robust mathematical software for numerically rank-deficient problems requires a major amount of research which is outside the scope of the present paper.

Our paper is organized as follows. After briefly surveying general rank-revealing decompositions in section 2, we define and analyze the rank-revealing VSV decomposition of a symmetric matrix in section 3. Numerical algorithms for computing VSV decompositions of symmetric semidefinite and indefinite matrices are presented in section 4, and we conclude with some numerical examples in section 5.

2. General rank-revealing decompositions. In this paper we restrict our attention to real square $n \times n$ matrices. The SVD of a square matrix is given by

$$A = U\,\Sigma\,V^T = \sum_{i=1}^{n} u_i\,\sigma_i\,v_i^T, \tag{2.1}$$

where u_i and v_i are the columns of the orthogonal matrices U and V, and $\Sigma = \mathrm{diag}(\sigma_i)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Then $\|A\|_2 = \sigma_1$, $\|A\|_F^2 = \sum_{i=1}^n \sigma_i^2$, and $\mathrm{cond}(A) = \sigma_1/\sigma_n$. The numerical rank k of A, with respect to the threshold τ, is the number of singular values greater than or equal to τ, i.e., $\sigma_k \geq \tau > \sigma_{k+1}$ [20, section 3.1].

The RRQR, URV, and ULV decompositions are given by

$$A = Q\,T\,\Pi^T = U_R\,R\,V_R^T = U_L\,L\,V_L^T.$$

Here, Q, U_R, U_L, V_R, and V_L are orthogonal matrices, Π is a permutation matrix, T and R are upper triangular matrices, and L is a lower triangular matrix. Moreover, if we partition the triangular matrices as

$$T = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix}, \qquad R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \qquad L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix},$$

then the numerical rank k of A is revealed in the triangular matrices in the sense that T_{11}, R_{11}, and L_{11} are $k \times k$ and

$$\mathrm{cond}(T_{11}) \simeq \sigma_1/\sigma_k, \qquad \|T_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2,$$

$$\mathrm{cond}(R_{11}) \simeq \sigma_1/\sigma_k, \qquad \|R_{12}\|_F^2 + \|R_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2,$$

$$\mathrm{cond}(L_{11}) \simeq \sigma_1/\sigma_k, \qquad \|L_{21}\|_F^2 + \|L_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2.$$

The first k columns of the left matrices Q, U_R, and U_L span approximations to the numerical range of A, defined as $\mathrm{span}\{u_1, \ldots, u_k\}$, and the last $n-k$ columns of the right matrices V_R and V_L span approximations to the numerical null space of A, defined as $\mathrm{span}\{v_{k+1}, \ldots, v_n\}$. See, e.g., [20, section 3.1] for details.

Precise definitions of RRQR decompositions and algorithms are given by Chandrasekaran and Ipsen [11], Gu and Eisenstat [19], and Hong and Pan [23], and associated large-scale implementations are available in Fortran [4]. Definitions of UTV decompositions and algorithms are given by Stewart [32], [33]. Matlab software for both RRQR and UTV decompositions is available in the UTV Tools package [17].

3. Symmetric rank-revealing decompositions. For a symmetric $n \times n$ matrix A, we need rank-revealing decompositions that inherit the symmetry of the original matrix. In particular this is true for the eigenvalue decomposition (EVD)

$$A = V\,\Lambda\,V^T = \sum_{i=1}^n v_i\,\lambda_i\,v_i^T, \tag{3.1}$$

where v_i are the right singular vectors, while $\sigma_i = |\lambda_i|$ and $u_i = \mathrm{sign}(\lambda_i)\,v_i$ for $i = 1, \ldots, n$.

Corresponding to the UTV decompositions, Luk and Qiao [24] defined the following VSV decomposition:

$$A = V_S\,S\,V_S^T, \tag{3.2}$$

where V_S is an orthogonal matrix, and S is a symmetric matrix with partitioning

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{pmatrix}, \tag{3.3}$$

in which S_{11} is $k \times k$. We say that the VSV decomposition is rank-revealing if

$$\mathrm{cond}(S_{11}) \simeq \sigma_1/\sigma_k, \qquad \|S_{12}\|_F^2 + \|S_{22}\|_F^2 \simeq \sigma_{k+1}^2 + \cdots + \sigma_n^2.$$

This definition is very similar to the definition used by Luk and Qiao, except that they use $\|\mathrm{triu}(S_{22})\|_F^2$ instead of $\|S_{22}\|_F^2$, where "triu" denotes the upper triangular part. Our choice is motivated by the fact that $\|S_{22}\|_F^2 \to \sigma_{k+1}^2 + \cdots + \sigma_n^2$ as $\|S_{12}\|_F \to 0$.

Given the VSV decomposition in (3.2), the first k columns of V_S and the last $n-k$ columns of V_S provide approximate basis vectors for the numerical range and null space, respectively. Moreover, given the ill-conditioned problem $A\,x = b$, we can compute a stabilized "truncated VSV solution" x_k by neglecting the three blocks in S with small norm, i.e., $x_k = V_{S,k} S_{11}^{-1} V_{S,k}^T b$, where $V_{S,k}$ consists of the first k columns of V_S. We return to the computation of x_k in section 4.4.

Instead of working directly with the matrix S, it is more convenient to work with a symmetric decomposition of S and, in particular, of S_{11}. The form of this decomposition depends on both the matrix A (semidefinite or indefinite) and the rank-revealing algorithm. Hence, we postpone a discussion of the particular form of S to the presentation of the algorithms. Instead, we summarize the approximation properties of the VSV decomposition.

THEOREM 3.1. *Let the VSV decompositions of A be given by* (3.2)*, and partition the matrix S as in* (3.3)*, where k is the numerical rank. Then the singular values $\bar{\sigma}_i$ of* $\mathrm{diag}(S_{11}\,,\, S_{22})$ *are related to those of A as*

$$|\bar{\sigma}_i - \sigma_i| \le \|S_{12}\|_2, \qquad i = 1, \ldots, n. \tag{3.4}$$

Moreover, the angle Θ between the subspaces spanned by the first k columns of V and V_S, defined by $\sin\Theta = \|V_k V_k^T - V_{S,k} V_{S,k}^T\|_2$*, is bounded as*

$$\frac{\|S_{12}\|_2}{\sigma_1 + \sigma_{k+1}} \le \sin\Theta \le \frac{\|S_{12}\|_2}{\sigma_k - \|S_{22}\|_2}. \tag{3.5}$$

Proof. The bound (3.4) follows from the standard perturbation bound for singular values:

$$|\bar{\sigma}_i - \sigma_i| \le \left\| \begin{pmatrix} 0 & S_{12} \\ S_{12}^T & 0 \end{pmatrix} \right\|_2 = \|S_{12}\|_2,$$

where we use that the singular values of the symmetric "perturbation matrix" appear in pairs. To prove the upper bound in (3.5), we partition $V = (V_k\,,\, V_0)$ and $V_S = (V_{S,k}\,,\, V_{S,0})$ such that V_k and $V_{S,k}$ have k columns. Moreover, we write $\Lambda = \mathrm{diag}(\Lambda_k\,,\, \Lambda_0)$, where Λ_k is $k \times k$. If we insert these partitionings as well as (3.1) and (3.2) into the product $A\,V_{S,0}$, then we obtain

$$\left(V_k \Lambda_k\, V_k^T + V_0\, \Lambda_0\, V_0^T\right) V_{S,0} = V_{S,k}\, S_{12} + V_{S,0}\, S_{22}.$$

Multiplying from the left with V_k^T we get

$$\Lambda_k\, V_k^T\, V_{S,0} = V_k^T V_{S,k}\, S_{12} + V_k^T V_{S,0}\, S_{22},$$

from which we obtain

$$V_k^T V_{S,0} = \Lambda_k^{-1} \left(V_k^T V_{S,k}\, S_{12} + V_k^T V_{S,0}\, S_{22}\right).$$

Taking norms in this expression and inserting $\sin\Theta = \|V_k^T V_{S,0}\|_2$ and $\|\Lambda_k^{-1}\|_2 = \sigma_k^{-1}$, we get

$$\sin\Theta \le \sigma_k^{-1}\|S_{12}\|_2 + \sigma_k^{-1}\|S_{22}\|_2 \sin\Theta,$$

which immediately leads to the upper bound in (3.5). To prove the lower bound, we use that

$$S_{12} = V_{S,k}^T A\, V_{S,0} = V_{S,k}^T V_k \Lambda_k V_k^T V_{S,0} + V_{S,k}^T V_0 \Lambda_0 V_0^T V_{S,0}.$$

Taking norms and using $\sin\Theta = \|V_k^T V_{S,0}\|_2 = \|V_{S,k}^T V_0\|_2$, $\|\Lambda_k\|_2 = \sigma_1$, and $\|\Lambda_0\|_2 = \sigma_{k+1}$, we obtain the left bound in (3.5). ∎

We conclude that if there is a well-defined gap between σ_k and $\|S_{22}\|_2$, and if the norm $\|S_{12}\|_2$ of the off-diagonal block is smaller than this gap, then the numerical rank k is indeed revealed in S, and the first k columns of V_S span an approximation to the singular subspace $\mathrm{span}\{v_1, \ldots, v_k\}$. The following theorem shows that a well-defined gap is also important for the perturbation bounds.

THEOREM 3.2. *Let $\widetilde{A} = A + \Delta A = \widetilde{V}_S\, \widetilde{S}\, \widetilde{V}_S^T$, and let Φ denote the angle between the subspaces spanned by the first k columns of V_S and $\widetilde{V}_S$; then*

$$\sin\Phi \le \frac{4\tau + \|\Delta A\|_2}{\sigma_k - \sigma_{k+1} - 4\tau - \|\Delta A\|_2}, \tag{3.6}$$

where $\tau = \max\{\|S_{12}\|_2, \|\widetilde{S}_{12}\|_2\}$.

Proof. The bound follows from Corollary 3.2 in [14]. ∎

We see that a small upper bound is guaranteed when $\|\Delta A\|_2$ as well as τ and σ_{k+1} are somewhat smaller than σ_k.

4. Algorithms for symmetric rank-revealing decompositions. Similar to general rank-revealing algorithms, the symmetric algorithms consist of an initial triangular factorization and a rank-revealing postprocessing step. The purpose of the latter step is to ensure that the largest k singular values are revealed in the leading submatrix S_{11} and that the corresponding singular subspace is approximated by the span of the first k columns of V_S.

For a semidefinite matrix A, our initial factorization is the symmetrically pivoted Cholesky factorization

$$P^T A\, P = C^T C, \tag{4.1}$$

where P is the permutation matrix, and C is the upper triangular (or trapezoidal) Cholesky factor. The numerical properties of this algorithm are discussed by Higham in [22]. If A is a symmetric semidefinite Toeplitz matrix, then there is good evidence (although no strict proof) that the Cholesky factor can be computed efficiently and reliably without the need for pivoting by means of the standard Schur algorithm [31].

When A is indefinite, then it would be convenient to work with an initial factorization of the form $P^T A\, P = C^T \Omega\, C$, where C is again triangular and $\Omega = \mathrm{diag}(\pm 1)$. Unfortunately such factorizations are not guaranteed to exist. Therefore our initial factorization is the symmetrically pivoted LDL^T factorization

$$P^T A\, P = L\, D\, L^T, \tag{4.2}$$

where P is the permutation matrix, L is a unit lower triangular matrix, and D is a block diagonal matrix with 1×1 and 2×2 blocks on the diagonal. The state of the art

in LDL^T algorithms is described in [1], where it is pointed out that special care must be taken in the implementation to avoid large entries in L when A is ill conditioned. Alternatively, one could use the factorization

$$P^T A\, P = G\, \Omega\, G^T, \qquad \Omega = \mathrm{diag}(\pm 1) \tag{4.3}$$

described in [30], where G is block triangular. If A is a symmetric indefinite Toeplitz matrix, then the currently most reliable approach to computing the LDL^T factorization seems to be via orthogonal transformation to a Cauchy matrix [21].

The reason why we need the postprocessing step is that the initial factorization may not reveal the numerical rank of A—there is no guarantee that small eigenvalues of A manifest themselves in small diagonal elements of C or in small eigenvalues of D. In particular, since $\|A^{-1}\|_2 = \sigma_n^{-1} \le \|L^{-1}\|_2^2\, \|D^{-1}\|_2 = \sigma_n(L)^{-2}\, \sigma_n(D)^{-1}$ and $\sigma_n \le \sigma_n(D)\, \|L\|_2^2$, we obtain

$$\sigma_n(L)^2 \le \frac{\sigma_n}{\sigma_n(D)} \le \|L\|_2^2,$$

showing that a small σ_n may not be revealed in D when L is ill conditioned.

4.1. Algorithms for semidefinite matrices. For symmetric semidefinite matrices there is a simple relationship between the SVDs of A and C.

THEOREM 4.1. *The right singular vectors of $P^T A\, P$ are also the right singular vectors of C, and*

$$\sigma_i(C) = \sigma_i^{1/2}, \qquad i = 1, \ldots, n. \tag{4.4}$$

Proof. The result follows from inserting the SVD of C into $P^T A\, P = C^T C$. ∎

Hence, once we have computed the initial pivoted Cholesky factorization (4.1), we can proceed by computing a rank-revealing decomposition of C, and this can be done in several ways. Let E denote the exchange matrix consisting of the columns of the identity matrix in reverse order, and write $P^T A\, P$ as

$$P^T A\, P = C^T C = E\, (ECE)^T (ECE)\, E.$$

Then we can compute a URV or RRQR decomposition of C, a ULV decomposition of ECE, or an RRQR decomposition of $(ECE)^T$, as shown in the left part of Table 4.1. The approach using the URV decomposition of C was suggested in [2]. Table 4.1 also shows the particular forms of the resulting symmetric matrix S, as derived from the following relations:

$$\begin{array}{rcll} P^T A\, P & = & V_R\, R^T R\, V_R^T & \text{(URV postprocessor)} \\ & = & \Pi\, T^T T\, \Pi^T & \text{(RRQR postprocessor)} \\ & = & (E\, V_L)\, L^T L\, (E\, V_L)^T & \text{(ULV postprocessor)} \\ & = & (E\, Q)\, T\, T^T (E\, Q)^T & \text{(RRQR postprocessor).} \end{array}$$

The first, third, and fourth approaches lead to a symmetric matrix S that reveals the numerical rank of A by having both an off-diagonal block S_{12} and a bottom right block S_{22} with small norm. The second approach does not produce blocks S_{12} and S_{22} with small norm; instead (since T_{11} is well conditioned) this algorithm provides a symmetric permutation $P\Pi$ that is guaranteed to produce a well-conditioned leading $k \times k$ submatrix in $(P\Pi)^T A\, (P\Pi)$.

TABLE 4.1
The four postprocessing rank-revealing steps for a symmetric semidefinite matrix.

Postproc.	Decomposition	Symmetric matrix
URV	$C = U_R\, R\, V_R^T$	$S = R^T R = \begin{pmatrix} R_{11}^T R_{11} & R_{11}^T R_{12} \\ R_{12}^T R_{11} & R_{12}^T R_{12} + R_{22}^T R_{22} \end{pmatrix}$
RRQR	$C = Q\, T\, \Pi^T$	$S = T^T T = \begin{pmatrix} T_{11}^T T_{11} & T_{11}^T T_{12} \\ T_{12}^T T_{11} & T_{12}^T T_{12} + T_{22}^T T_{22} \end{pmatrix}$
ULV	$ECE = U_L\, L\, V_L^T$	$S = L^T L = \begin{pmatrix} L_{11}^T L_{11} + L_{21}^T L_{21} & L_{21}^T L_{22} \\ L_{22}^T L_{21} & L_{22}^T L_{22} \end{pmatrix}$
RRQR	$(ECE)^T = Q\, T\, \Pi^T$	$S = T\, T^T = \begin{pmatrix} T_{11} T_{11}^T + T_{12}\, T_{12}^T & T_{12} T_{22}^T \\ T_{22} T_{12}^T & T_{22} T_{22}^T \end{pmatrix}$

The remaining three algorithms yield approximate bases for the range and null spaces of A, due to Theorem 3.1. It is well known that among the rank-revealing decompositions, the ULV decomposition can be expected to provide the most accurate bases for the right singular subspaces in the form of the columns of V_L; see, e.g., [33] and [15]. Therefore, the algorithm that computes the ULV decomposition of ECE is to be preferred. We remark that the matrix U_L in the ULV decomposition need not be computed.

In terms of the blocks S_{12} and S_{22}, the ULV-based algorithm is the only algorithm that guarantees small norms of both the off-diagonal block $S_{12} = L_{21}^T L_{22}$ and the bottom right block $S_{22} = L_{22}^T L_{22}$, because the norms of both L_{12} and L_{22} are guaranteed to be small. From Theorem 4.1 and the definition of the ULV decomposition we have $\|L_{21}\|_2 \simeq \|L_{22}\|_2 \simeq \sigma_{k+1}^{1/2}$, and therefore $\|S_{12}\|_2 \simeq \|S_{22}\|_2 \simeq \sigma_{k+1}$.

For a *sparse* matrix the situation is different, because the UTV postprocessors may produce severe fill, while the RRQR postprocessor produces only fill in the $n-k$ rightmost columns of T. For example, if A is the upper bidiagonal matrix

$$A = \begin{pmatrix} 10^{-5} B_{n-k} & e_k e_1^T \\ 0 & B_k \end{pmatrix},$$

in which B_p is an upper bidiagonal $p \times p$ matrix of all ones, and e_p is the pth column of the identity matrix, then URV with threshold $\tau = 10^{-4}$ produces a full $k \times k$ upper triangular R_{11}, while RRQR with the same threshold produces a $k \times k$ upper bidiagonal T_{11}. Hence, for sparsity reasons, the UTV approaches may not be suited for computing the VSV decomposition, depending on the sparsity pattern of A.

An alternative is to use the algorithm based on RRQR decomposition of the transposed and permuted Cholesky factor $(ECE)^T = E\, C^T E$, and we note that the permutation matrix Π is not needed. In terms of the matrix S, only the bottom right submatrix of S is guaranteed to have a norm of the order σ_{k+1} because of the relations $\|S_{12}\|_2 = \|T_{12} T_{22}^T\|_2 \simeq \sigma_1^{1/2} \sigma_{k+1}^{1/2}$ and $\|S_{22}\|_2 = \|T_{22} T_{22}^T\|_2 \simeq \sigma_{k+1}$.

In practice the situation can be better, because the RRQR-algorithm—when applied to the matrix $E\, C^T E$—may produce an off-diagonal block T_{12} whose norm is smaller than what is guaranteed (namely, of the order $\sigma_1^{1/2}$). The reason is that the initial Cholesky factor C often has a trailing $(n-k) \times (n-k)$ triangular block C_{22}

whose norm is close to $\sigma_{k+1}^{1/2}$, which may produce a norm $\|S_{12}\|_2$ close to σ_{k+1}. From the partitionings

$$C = \begin{pmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{pmatrix}, \qquad E\,C^T E = \begin{pmatrix} E_{n-k} C_{22}^T E_{n-k} & E_{n-k} C_{12}^T E_k \\ 0 & E_k C_{11}^T E_k \end{pmatrix}$$

and the fact that the RRQR postprocessor leaves column norms unchanged and may permute the leading $n-k$ columns of $E\,C^T E$ to the back, we see that the norm of the resulting off-diagonal block T_{12} in the RRQR decomposition can be bounded by $\|C_{22}\|_2$. Our numerical examples in section 5 illustrate this.

However, we stress that in the RRQR approach we can guarantee only that $\|S_{12}\|_2$ is of the order $\sigma_1^{1/2}\sigma_{k+1}^{1/2}$, and this point is illustrated by the matrix $A = K^T K$, where K is the "infamous" Kahan matrix [7, p. 105] that is left unchanged by QR factorization with ordinary column pivoting, yet its numerical rank is $k = n-1$. Cholesky factorization with symmetric pivoting computes the Cholesky factor $C = K$, and when we apply RRQR to $E\,C^T E$ we obtain an upper triangular matrix T in which only the (n,n)-element is small, while $\|T_{12}\|_2 = 1 \simeq \|T\|_2$ and $\|S_{12}\|_2 = \|T_{12}\,T_{22}^T\|_2 \simeq \|T_{22}\|_2 \simeq \sigma_n^{1/2}$.

4.2. Algorithms for indefinite matrices. No matter which factorization is used for an indefinite matrix, such as (4.2) or (4.3), there is no simple relationship between the singular values of A and the matrix factors. Hence the four "intrinsic" decompositions from Table 4.1 do not apply here, and the difficulty is to develop a new factorization from which the numerical rank can be determined.

All rank-revealing algorithms currently in use maintain the triangular form of the matrix in consideration, but when we apply the algorithms to the matrix L in the LDL^T factorization (4.2) we destroy the block diagonal form of D. We can avoid this difficulty by inserting an additional *interim stage* between the initial LDL^T factorization and the rank-revealing postprocessor, in which the middle block diagonal matrix D is replaced by the signature matrix $\Omega = \mathrm{diag}(\pm 1)$. At the same time, L is replaced by the product of an orthogonal matrix and a triangular matrix. The interim processor, which is summarized in Figure 4.1, thus computes the factorization

$$P^T A\,P = W\,C^T \Omega\,C\,W^T, \tag{4.5}$$

where W is orthogonal and C is upper triangular.

The interim processor is simple to implement and requires at most $\mathcal{O}(n^2)$ operations, because W and $\overline{W}$ are block diagonal matrices with the same block structure as D. For each 1×1 block d_{ii} in D, the corresponding 1×1 blocks in W, $|\Lambda|^{1/2}$, and $\overline{W}$ are equal to 1, $|d_{ii}|^{1/2}$, and 1, respectively. For each 2×2 block in D, we compute the eigenvalue decomposition

$$\begin{pmatrix} d_{ii} & d_{i,i+1} \\ d_{i,i+1} & d_{i+1,i+1} \end{pmatrix} = \overline{W}_{ii} \begin{pmatrix} \lambda_i & 0 \\ 0 & \lambda_{i+1} \end{pmatrix} \overline{W}_{ii}^T ;$$

1. Compute the eigenvalue decomposition $D = \overline{W}\,\Lambda\,\overline{W}^T$.
2. Write Λ as $\Lambda = |\Lambda|^{1/2}\,\Omega\,|\Lambda|^{1/2}$.
3. Compute an orthogonal W such that $C^T = W^T L\,\overline{W}\,|\Lambda|^{1/2}$ is lower triangular.

FIG. 4.1. *Interim processor for symmetric indefinite matrices.*

then the corresponding 2×2 block in $\overline{W}$ is $\overline{W}_{ii}$, and the associated 2×2 block in W is a Givens rotation chosen such that C stays triangular. If A is sparse, then some fill may be introduced in C by the interim processor, but since the Givens transformations are applied to nonoverlapping 2×2 blocks, fill introduced in the treatment of a particular block does not spread during the processing of the other blocks. The same type of interim processor can also be applied to the $G\,\Omega\,G^T$ factorization (4.3) in order to turn the block triangular matrix G into triangular form.

Future developments of rank-revealing algorithms for more general matrices than the triangular ones may render the interim processor superfluous. It may also be possible to compute the factorization (4.5) directly.

We shall now explore the possibilities for using triangular rank-revealing post-processors similar to the ones for semidefinite matrices, but modified such that they yield a decomposition of C in which the leftmost matrix U is *hypernormal* with respect to the signature matrices Ω and $\widehat{\Omega}$, i.e., we require that $U^T\Omega\,U = \widehat{\Omega}$ and that the inertia of Ω and $\widehat{\Omega}$ are the same. Hypernormal matrices and the corresponding transformations are introduced in [8] in connection with up- and downdating of symmetric indefinite matrices. Here we use them to maintain the triangular form of the matrix C.

The following theorem shows that a small singular value of A is guaranteed to be revealed in the triangular matrix C.

THEOREM 4.2. *If $\sigma_n(C)$ denotes the smallest singular value of C in the interim factorization* (4.5)*, then*

$$\sigma_n(C) \le \sigma_n^{1/2}. \tag{4.6}$$

Proof. We have $\sigma_n^{-1} = \|(C^T\Omega\,C)^{-1}\|_2 \le \|C^{-1}\|_2\,\|\Omega\|_2\,\|C^{-T}\|_2 = \|C^{-1}\|_2^2 = \sigma_n(C)^{-2}$, from which the result follows. ∎

Unfortunately, there is no guarantee that $\sigma_n(C)$ does not underestimate $\sigma_n^{1/2}$ dramatically, nor does it ensure that the size of σ_n is revealed in S. We illustrate this with a small 5×5 numerical example from [1] where A is given by $A = L\,D\,L^T$ with

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & -\frac{20}{13} & -\frac{8}{17} & 1 & 0 \\ 1 & \frac{6\cdot 10^6}{13} & -\frac{1}{17} & 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{10^{-19}}{3} & \frac{6\cdot 10^{-7}}{7} & 0 & 0 \\ 0 & \frac{6\cdot 10^{-7}}{7} & -\frac{3\cdot 10^{-6}}{13} & 0 & 0 \\ 0 & 0 & 0 & -\frac{4\cdot 10^{-5}}{17} & \frac{2}{7} \\ 0 & 0 & 0 & \frac{2}{7} & \frac{1}{300} \end{pmatrix},$$

and $\mathrm{cond}(L) = 3.01\cdot 10^{11}$. The singular values of A are

$$\sigma_1 = 5.13, \quad \sigma_2 = 0.270, \quad \sigma_3 = 0.142, \quad \sigma_4 = 2.66\cdot 10^{-7}, \quad \sigma_5 = 1.14\cdot 10^{-8}$$

such that A has full rank with respect to the threshold $\tau = 10^{-10}$. The corresponding matrix C has singular values

$$\sigma_1(C) = 104, \quad \sigma_2(C) = 2.02, \quad \sigma_3(C) = 0.459,$$
$$\sigma_4(C) = 3.10\cdot 10^{-4}, \quad \sigma_5(C) = 8.17\cdot 10^{-7}.$$

Thus, $\sigma_5(C)$ is not a good approximation of $\sigma_5^{1/2}$, and if we base the rank decision on $\sigma_5(C)$ and the threshold $\tau^{1/2} = 10^{-5}$, then we wrongly conclude that A is numerically rank deficient.

The conclusion is that for indefinite matrices, a well-conditioned C ensures that A is well conditioned, but we cannot rely solely on C for determination of the numerical rank of A. This rules out the use of RRQR factorization of C and ECE. The following theorem (which expands on results in [24]) shows how to proceed instead.

THEOREM 4.3. *Let w_n be an eigenvector of $C^T \Omega\, C$ corresponding to the eigenvalue λ_n that is smallest in absolute value, and let $\tilde{w}_n$ be an approximation to w_n. Moreover, choose the orthogonal matrix $\widehat{V}$ such that $\widehat{V}^T \tilde{w}_n = e_n$, the last column of the identity matrix, and partition the matrix*

$$\widehat{V}^T C^T \Omega\, C\, \widehat{V} = S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}$$

such that S_{11} is $(n-1) \times (n-1)$. Then

$$\|s_{12}\|_2 \le (\sigma_1 + \sigma_n)\, \|\tilde{w}_n - w_n\|_2 \tag{4.7}$$

and

$$|s_{22} - \lambda_n| \le (\sigma_1 + \sigma_n)\, \|\tilde{w}_n - w_n\|_2. \tag{4.8}$$

Proof. Let $\hat{A} = W^T P^T A\, P\, W = C^T \Omega\, C$ and consider first the quantity

$$\widehat{V}^T \hat{A}\, \tilde{w}_n = S\, \widehat{V}^T \tilde{w}_n = S\, e_n = \begin{pmatrix} s_{12} \\ s_{22} \end{pmatrix}.$$

Next, write $\tilde{w}_n = w_n + u$ to obtain

$$\begin{aligned} \widehat{V}^T \hat{A}\, \tilde{w}_n &= \widehat{V}^T \hat{A}\, (w_n + u) = \lambda_n\, \widehat{V}^T w_n + \widehat{V}^T \hat{A}\, u \\ &= \lambda_n\, \widehat{V}^T (\tilde{w}_n - u) + \widehat{V}^T \hat{A}\, u = \lambda_n\, e_n - \lambda_n\, \widehat{V}^T u + \widehat{V}^T \hat{A}\, u. \end{aligned}$$

Combining these two results we obtain

$$\begin{pmatrix} s_{12} \\ s_{22} - \lambda_n \end{pmatrix} = \widehat{V}^T (\hat{A} - \lambda_n\, I)\, u,$$

and taking norms we get

$$\|s_{12}\|_2^2 + (s_{22} - \lambda_n)^2 = \|(\hat{A} - \lambda_n\, I)\, u\|_2^2 \le \|\hat{A} - \lambda_n\, I\|_2^2\, \|u\|_2^2.$$

Both $\|s_{12}\|_2^2$ and $|s_{22} - \lambda_n|$ are lower bounds for the left-hand side. Combining this with the bound $\|\hat{A} - \lambda_n\, I\|_2 \le \sigma_1 + \sigma_n$, we obtain the two bounds in the theorem. ∎

We emphasize that small $\|s_{12}\|_2$ and $|s_{22}|$ do not imply that the elements in the last column of $C\, \widehat{V}$ are small (in contrast to the semidefinite case), due to the presence of the signature matrix Ω in $S = \widehat{V}^T C^T \Omega\, C\, \widehat{V}$.

The above theorem shows that in order for σ_n to reveal itself in S, we must compute an approximate null vector of $C^T \Omega\, C$, apply Givens rotations to this vector to transform it into e_n, and accumulate these rotations from the right into C. At the same time, we should apply hypernormal rotations from the left in order to keep C upper triangular. (Note that the hypernormal rotations should also be applied to the (1,2)-submatrix of C.) Theorem 4.3 ensures that if $\tilde{w}_n$ is close enough to w_n,

then $\|s_{12}\|_2$ is small and s_{22} approximates λ_n. We note that hypernormal transformations can be numerically unstable, and in our implementations we use the same stabilizations as in the stabilized hyperbolic rotations [7, section 3.3.4].

Once this step has been performed, we deflate the problem and apply the same technique to the $(n-1) \times (n-1)$ submatrix $S_{11} = C_{11}^T \widehat{\Omega}_{11} C_{11}$, where C_{11} and $\widehat{\Omega}_{11}$ are the leading submatrices of the *updated* factors. When the process stops (because all the small singular values of A are revealed) we have computed the URV-like decomposition $C = U_R R V_R^T$ such that $U_R^T \Omega U_R = \widehat{\Omega}$, and the middle rank-revealing matrix is given by

$$S = R^T \widehat{\Omega} R = \begin{pmatrix} R_{11}^T \widehat{\Omega}_1 R_{11} & R_{11}^T \widehat{\Omega}_1 R_{12} \\ R_{12}^T \widehat{\Omega}_1 R_{11} & R_{12}^T \widehat{\Omega}_1 R_{12} + R_{22}^T \widehat{\Omega}_2 R_{22} \end{pmatrix}, \tag{4.9}$$

where $\widehat{\Omega} = \operatorname{diag}(\widehat{\Omega}_1, \widehat{\Omega}_2)$ and $\widehat{\Omega}_1$ is $k \times k$. This is precisely the algorithm from [24]; it is summarized in Figure 4.2 (following the presentations from [17]), where τ is the rank-decision tolerance for A.

The condition estimator used in the URV-like postprocessor must be modified, compared to the standard URV algorithm, because we must now estimate the smallest singular value of the matrix $C^T \Omega C$. In our implementation we use one step of inverse iteration applied to $C^T \Omega C$, with starting vector from the condition estimator of the ordinary URV algorithm applied to C.

Next we consider a ULV-like approach applied to ECE. Again we must compute an approximate null vector of $C^T \Omega C$ and transform it into the form e_n by means of an orthogonal transformation. This transformation is applied from the right to ECE, and a hypernormal transformation from the left is then required to restore the lower triangular form of $L = ECE$.

To deflate this factorization, note that the leading $(n-1) \times (n-1)$ block of $S = L^T \widehat{\Omega} L$ is given by

$$S_{11} = L_{11}^T \widehat{\Omega}_1 L_{11} + \ell_{21} \omega_2 \ell_{21}^T , \qquad \text{where} \qquad L = \begin{pmatrix} L_{11} & \\ \ell_{21}^T & \ell_{22} \end{pmatrix}$$

and $\widehat{\Omega} = \operatorname{diag}(\widehat{\Omega}_1, \omega_2)$. This shows that we *cannot* merely work on the block L_{11}; also the $1 \times (n-1)$ block ℓ_{21}^T is needed because for indefinite matrices $\|\ell_{21}\|_2$ is not guaranteed to be small when $\|s_{12}\|_2$ is small (in contrast to the semidefinite case).

1. Let $k \leftarrow n$ and compute an initial factorization $P^T A P = L D L^T$.
2. Apply the interim processor to compute $P^T A P = W C^T \Omega C W^T$.
3. Condition estimation: let $\widetilde{\sigma}_k$ estimate $\sigma_k\big(C(1\colon k, 1\colon k)^T \Omega(1\colon k, 1\colon k)\, C(1\colon k, 1\colon k)\big)$ and let w_k estimate the corresponding right singular vector.
4. If $\widetilde{\sigma}_k > \tau^{1/2}$ then **exit**.
5. Revealment: determine an orthogonal Q_k such that $Q_k^T w_k = e_k$;
6. update $C(1\colon k, 1\colon k) \leftarrow C(1\colon k, 1\colon k)\, Q_k$;
7. update $C(1\colon k, 1\colon n) \leftarrow H_k^T C(1\colon k, 1\colon n)$, where the hypernormal matrix H_k is chosen such that the updated C is triangular;
8. Deflation: let $k \leftarrow k - 1$.
9. Go to step 3.

FIG. 4.2. *The URV-based VSV algorithm for symmetric indefinite matrices.*

TABLE 4.2
Summary of approaches for symmetric indefinite matrices. Note that the RRQR approaches do not reveal the numerical rank, and that the ULV-like approach is impractical.

Postproc.	Decomposition	Comments to decomposition
URV-like	$C = U_R R V_R^T$ $U_R^T \Omega U_R = \widehat{\Omega}$	$S = R^T \widehat{\Omega} R$ $S_{11} = R_{11}^T \widehat{\Omega} R_{11}$
RRQR	$C = Q R \Pi^T$ $Q^T \Omega Q = \widehat{\Omega}$	Cannot reveal numerical rank
ULV-like	$ECE = U_L L V_L^T$ $U_L^T E \Omega E U_L = \widehat{\Omega}$	$S = L^T \widehat{\Omega} L$ $S_{11} = L_{11}^T \widehat{\Omega}_1 L_{11} + L_{21}^T \widehat{\Omega}_2 L_{21}$
RRQR	$(ECE)^T = Q R \Pi^T$ $Q^T Q = I$	Cannot reveal numerical rank

Hence, after the deflation step we must work with trapezoidal matices instead of triangular matrices. This fact renders the ULV-like approach impractical.

We do not think that it is possible to derive an RRQR-like scheme that can produce the desired rank-revealing VSV decomposition. If we use $C = Q T \Pi^T$ with Π chosen from $\tilde{w}_n$, then we cannot prove that $s_{22} \approx \lambda$, and it is impossible to use $ECE = \Pi T^T Q^T$ because Q does not carry information about $\tilde{w}_n$.

To summarize, for symmetric indefinite matrices only the approach using the URV-like postprocessor leads to a practical algorithm for revealing the numerical rank of A. Moreover, a well-conditioned C signals a well-conditioned A, but C cannot reveal A's numerical rank. Our analysis is summarized in Table 4.2.

4.3. Updating the VSV decomposition. One of the advantages of the rank-revealing VSV decomposition over the SVD is that it can be updated efficiently when A is modified by a rank-one change $v\,v^T$. From the relation

$$A^{\mathrm{up}} = A + v\,v^T = V_S \left(S + (V_S^T v)(V_S^T v)^T \right) V_S^T$$

we see that the updating of A amounts to updating the rank-revealing matrix S by the rank-one matrix $w\,w^T$ with $w = V_S^T v$, i.e., $S^{\mathrm{up}} = S + w\,w^T$. This can be done in $\mathcal{O}(n^2)$ operations, while the EVD/SVD updating requires $\mathcal{O}(n^3)$ operations.

Consider first the semidefinite case, and let M denote one of the triangular matrices R, L, or T^T from the algorithms in Table 4.1. Then

$$S^{\mathrm{up}} = M^T M + w\,w^T = \begin{pmatrix} M \\ w^T \end{pmatrix}^T \begin{pmatrix} M \\ w^T \end{pmatrix}$$

and we see that the VSV updating is identical to standard updating of a triangular RRQR or UTV factor, which can be done stably and efficiently by means of Givens transformation as described in [5], [32], and [33].

Next we consider the indefinite case (4.9), where the updating takes the form

$$S^{\mathrm{up}} = R^T \widehat{\Omega} R + w\,w^T = \begin{pmatrix} R \\ w^T \end{pmatrix}^T \begin{pmatrix} \widehat{\Omega} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} R \\ w^T \end{pmatrix},$$

showing that the VSV updating now involves hypernormal rotations. Hence, the updating is computationally similar to UTV downdating, whose stable implementation is discussed in [3] and [26]. Downdating the VSV decomposition will, in both cases, also involve hypernormal rotations.

4.4. Computation of truncated VSV solutions. Here we briefly consider the computation of the truncated VSV solution, which we define as

$$V_{S,k}\, S_{11}^{-1} V_{S,k}^T b, \tag{4.10}$$

where $V_{S,k}$ consists of the first k columns of V_S. The decompositions based on the URV postprocessor (in the semidefinite case) and the URV-like postprocessor (in the indefinite case) are straightforward to use. For the ULV-based decomposition (in the semidefinite case only) we have $S_{11} = L_{11}^T L_{11} + L_{21}^T L_{21}$, and we can safely neglect the term $L_{21}^T L_{21}$ whose norm is at most of the order σ_{k+1}. Finally, for the RRQR-based decomposition we can use the following theorem.

THEOREM 4.4. *If $S = T\,T^T$ and $\widehat{T}$ is the triangular QR factor of $(T_{11}\,,\, T_{12})^T$, then*

$$S_{11}^{-1} = \widehat{T}^{-1}\big(\widehat{T}^{-1}\big)^T. \tag{4.11}$$

Alternatively, if the columns of the matrix

$$W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix}, \qquad W_1 \in \mathbb{R}^{(n-k)\times(n-k)},$$

form an orthonormal basis for the null space of $(T_{11}\,,\, T_{12})$, then

$$S_{11}^{-1} = \big(T_{11}^{-1}\big)^T \big(I - W_1\, W_1^T\big)\, T_{11}^{-1}. \tag{4.12}$$

Proof. If $(T_{11}\,,\, T_{12})^T = \widehat{Q}\,\widehat{T}$ is a QR factorization, then $S_{11} = \widehat{T}^T\widehat{T}$ and $S_{11}^{-1} = \widehat{T}^{-1}\big(\widehat{T}^{-1}\big)^T$ which is (4.11). The same relation leads to $S_{11}^{-1} = \widehat{T}^{-1}\widehat{Q}^T\widehat{Q}\,\big(\widehat{T}^{-1}\big)^T = \big((T_{11}\,,\, T_{12})^\dagger\big)^T\,(T_{11}\,,\, T_{12})^\dagger$, where † denotes the pseudoinverse. In [6] it is proved that

$$(T_{11}\,,\, T_{12})^\dagger = (I - W\,W^T)\,T^{-1}\begin{pmatrix} I_k \\ 0 \end{pmatrix},$$

which, combined with the relation $(I - W\,W^T)^2 = I - W\,W^T$, immediately leads to (4.12). ∎

The first relation (4.11) in Theorem 4.4 can be used when $k \ll n$, while the second relation (4.12) is more useful when $k \approx n$ because it avoids computing the large QR factorization. Note that W can be computed by orthonormalization of the columns of the matrix

$$Z = \begin{pmatrix} T_{11}^{-1}T_{12} \\ -I \end{pmatrix}.$$

This approach is particularly useful for sparse matrices because we introduce fill only when working with the "skinny" $n \times (n-k)$ matrix Z.

5. Numerical examples. The purpose of this section is to illustrate the theory derived in the previous sections by means of some test problems. Although robustness, efficiency, and flop counts are important practical issues, they are also tightly connected to the particular implementation of the rank-revealing postprocessor and not the subject of this paper.

All our experiments were done in Matlab, and we used the implementations of the ULV, URV, and RRQR algorithms from the UTV TOOLS package [17]. The condition estimation in all three implementations is the Cline–Conn–Van Loan (CCVL)

estimator [12]. The modified URV algorithm used for symmetric indefinite matrices is based on the URV algorithm from [17], augmented with stabilized hypernormal rotations when needed, and with a condition estimator consisting of the CCVL algorithm followed by one step of inverse iteration applied to the matrix $C^T \Omega\, C$.

Numerical results for all the rank-revealing algorithms are shown in Table 5.1, where we present mean and maximum values of the norms of various submatrices associated with the VSV decompositions. In particular, $X_{\rm off}$ denotes either R_{12}, L_{21}, or T_{12}, and X_{22} denotes either R_{22}, L_{22}, or T_{22}. The results are computed on the basis of randomly generated test matrices of sizes 64, 128, and 256 (100 matrices of each size), each with $n-4$ eigenvalues geometrically distributed between 1 and 10^{-4}, and the remaining four eigenvalues given by 10^{-7}, 10^{-8}, 10^{-9}, and 10^{-10}, such that the numerical rank with respect to the threshold $\tau = 10^{-5}$ is $k = n-4$.

The test matrices were produced by generating random orthogonal matrices and multiplying them to diagonal matrices with the desired eigenvalues. For the indefinite matrices the signs of the eigenvalues were chosen to alternate.

Table 5.1 illustrates the superiority of the ULV-based algorithm for semidefinite matrices, for which the norm $\|S_{12}\|_2$ of the off-diagonal block in S is always much smaller than the norm $\|S_{22}\|_2$ of the bottom right submatrix. This is due to the fact that the ULV algorithm produces a lower triangular matrix L whose off-diagonal block L_{21} has a very small norm (and we emphasize that the size of this norm depends on the condition estimator). The second best algorithm for semidefinite matrices is the one based on the RRQR algorithm, for which $\|S_{12}\|_2$ and $\|S_{22}\|_2$ are of the same size. Note that it is the latter algorithm which we recommend for sparse matrices. The URV-based algorithm for semidefinite matrices produces results that are consistently less satisfactory than the other two algorithms. All these results are consistent with our theory.

For the indefinite matrices, only the URV-like algorithm can be used, and the results in Table 5.1 show that this algorithm also behaves as expected from the theory. In order to judge the backward stability of this algorithm, which uses hypernormal rotations, we also computed the backward error $\|A - V_S S\, V_S^T\|_2$ for all three hundred test problems. The largest residual norm was $1.9 \cdot 10^{-11}$, and the average was $1.5 \cdot 10^{-12}$. We conclude that we lose a few digits of accuracy due to the use of the hypernormal rotations.

It is well known that the norm of the off-diagonal block in the triangular URV factor depends on the quality of the condition estimator—the better the singular vector estimate, the smaller the norm. Hence, it is interesting to see how much the

TABLE 5.1
Numerical results for the rank-revealing VSV algorithms.

Postprocessor		$\|X_{\rm off}\|_2$	$\|X_{22}\|_2$	$\|S_{12}\|_2$	$\|S_{22}\|_2$
URV	mean	$2.2 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	$6.5 \cdot 10^{-6}$	$2.5 \cdot 10^{-7}$
(semidef.)	max	$3.0 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$	$9.4 \cdot 10^{-6}$
ULV	mean	$2.7 \cdot 10^{-7}$	$3.2 \cdot 10^{-4}$	$8.6 \cdot 10^{-11}$	$1.0 \cdot 10^{-7}$
(semidef.)	max	$4.7 \cdot 10^{-7}$	$3.2 \cdot 10^{-4}$	$1.5 \cdot 10^{-10}$	$1.0 \cdot 10^{-7}$
RRQR	mean	$1.5 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$4.8 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$
(semidef.)	max	$2.9 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$9.2 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$
URV-like	mean	$1.4 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$4.6 \cdot 10^{-4}$	$2.3 \cdot 10^{-7}$
(indef.)	max	$2.1 \cdot 10^{-3}$	$3.2 \cdot 10^{-4}$	$9.4 \cdot 10^{-3}$	$4.3 \cdot 10^{-6}$

TABLE 5.2
Numerical results with improved singular vector estimates.

Postprocessor		$\|R_{12}\|_2$	$\|R_{22}\|_2$	$\|S_{12}\|_2$	$\|S_{22}\|_2$
URV	mean	$1.8 \cdot 10^{-7}$	$3.1 \cdot 10^{-4}$	$2.3 \cdot 10^{-9}$	$1.0 \cdot 10^{-7}$
(semidef.)	max	$5.1 \cdot 10^{-6}$	$3.2 \cdot 10^{-4}$	$6.1 \cdot 10^{-8}$	$1.0 \cdot 10^{-7}$
URV-like	mean	$8.6 \cdot 10^{-10}$	$3.1 \cdot 10^{-4}$	$4.4 \cdot 10^{-9}$	$1.0 \cdot 10^{-7}$
(indef.)	max	$2.3 \cdot 10^{-8}$	$3.2 \cdot 10^{-4}$	$1.5 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$

norms of the off-diagonal blocks in R and S decrease if we improve the singular vector estimates by means of one step of inverse iteration (at the expense of additional $2(n-k)n^2$ flops). In the semidefinite case we now apply an inverse iteration step to the CCVL estimate, and in the indefinite case we use two steps of inverse iteration applied to $C^T \Omega C$ instead of one. The results are shown in Table 5.2 for the same matrices as in Table 5.1. As expected, the norms of the off-diagonal blocks are now smaller, at the expense of more work. The average backward errors $\|A - V_S S V_S^T\|_2$ did not change in this experiment.

6. Conclusion. We have defined and analyzed a class of rank-revealing VSV decompositions for symmetric matrices and proposed algorithms for computing these decompositions. For semidefinite matrices, the ULV-based algorithm is the method of choice for dense matrices, while the RRQR-based algorithm is better suited for sparse matrices because it preserves sparsity better. For indefinite matrices, only the URV-based algorithm is guaranteed to work.

REFERENCES

[1] C. ASHCRAFT, R. G. GRIMES, AND J. G. LEWIS, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 513–561.

[2] E. S. BAKER AND R. D. DEGROAT, *A correlation-based subspace tracking algorithm*, IEEE Trans. Signal Process., 46 (1998), pp. 3112–3116.

[3] J. L. BARLOW, P. A. YOON, AND H. ZHA, *An algorithm and a stability theory for downdating the ULV decomposition*, BIT, 36 (1996), pp. 15–40.

[4] C. H. BISCHOF AND G. QUINTANA-ORTÍ, *Algorithm* 782: *Codes for rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24 (1998), pp. 254–257.

[5] C. H. BISCHOF AND G. M. SHROFF, *On updating signal subspaces*, IEEE Trans. Signal Process., 40 (1992), pp. 96–105.

[6] Å. BJÖRCK, *A bidiagonalization algorithm for solving ill-posed systems of linear equations*, BIT, 28 (1998), pp. 659–670.

[7] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[8] A. BOJANCZYK, S. QIAO, AND A. O. STEINHARDT, *Unifying unitary and hyperbolic transformations*, Linear Algebra Appl., 316 (2000), pp. 183–197.

[9] T. F. CHAN, *Rank-revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[10] T. F. CHAN AND P. C. HANSEN, *Low-rank revealing QR factorizations*, Numer. Linear Algebra Appl., 1 (1994), pp. 33–44.

[11] S. CHANDRASEKARAN AND I. C. F. IPSEN, *On rank-revealing factorisations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 592–622.

[12] A. K. CLINE, A. R. CONN, AND C. F. VAN LOAN, *Generalizing the LINPACK condition estimator*, in Numerical Analysis, Lecture Notes in Math. 909, J. P. Hennart, ed., Springer-Verlag, Berlin, 1992, pp. 78–83.

[13] L. ELDÉN AND E. SJÖSTRÖM, *Fast computation of the principal singular vectors of Toeplitz matrices arising in exponential data modelling*, Signal Process., 50 (1996), pp. 151–164.

[14] R. D. FIERRO, *Perturbation analysis for two-sided (or complete) orthogonal decompositions*, SIAM J. Matrix Anal. Appl, 17 (1996), pp. 383–400.

[15] R. D. FIERRO AND J. R. BUNCH, *Bounding the subspaces from rank revealing two-sided orthogonal decompositions*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 743–759.

[16] R. D. Fierro and P. C. Hansen, *Low-rank revealing two-sided orthogonal decompositions*, Numer. Algorithms, 15 (1997), pp. 37–55.
[17] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen, *UTV Tools: Matlab templates for rank-revealing UTV decompositions*, Numer. Algorithms, 20 (1999), pp. 165–194.
[18] L. V. Foster, *Rank and null space calculations using matrix decomposition without column interchanges*, Linear Algebra Appl., 74 (1986), pp. 47–71.
[19] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
[20] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1997.
[21] G. Heinig and A. Boyanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices* II. *Algorithms*, Linear Algebra Appl., 278 (1998), pp. 11–36.
[22] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix,* in Reliable Numerical Computing, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, London, 1990.
[23] Y. P. Hong and C.-T. Pan, *The rank revealing QR decomposition and SVD*, Math. Comp., 58 (1992), pp. 213–232.
[24] F. T. Luk and S. Qiao, *A symmetric rank-revealing Toeplitz matrix decomposition*, J. VLSI Signal Process., 14 (1996), pp. 19–28.
[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[26] H. Park and L. Eldén, *Downdating the rank-revealing URV decomposition*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 138–155.
[27] D. J. Pierce, *A Sparse URL Rather than a URV Factorization*, Report MEA-TR-203, Boeing Computer Services, Seattle, WA, 1992.
[28] D. J. Pierce and J. G. Lewis, *Sparse multifrontal rank revealing QR factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 159–180.
[29] G. Quintana-Ortí and E. S. Quintana-Ortí, *Parallel codes for computing the numerical rank*, Linear Algebra Appl., 275/276 (1998), pp. 451–470.
[30] I. Slapničar, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.
[31] M. Stewart, *Cholesky factorization of semi-definite Toeplitz matrices*, Linear Algebra Appl., 254 (1997), pp. 497–526.
[32] G. W. Stewart, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Process., 40 (1992), pp. 1535–1541.
[33] G. W. Stewart, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), 494–499.
[34] G. W. Stewart, *Matrix Algorithms Vol.* I: *Basic Decompositions*, SIAM, Philadelphia, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 459–471
© 2001 Society for Industrial and Applied Mathematics

THE POWER-COMPOSITIONS DETERMINANT AND ITS APPLICATION TO GLOBAL OPTIMIZATION*

JOSEP M. BRUNAT† AND ANTONIO MONTES†

Abstract. Let $C(n,p)$ be the set of p-compositions of an integer n, i.e., the set of p-tuples $\alpha=(\alpha_1,\dots,\alpha_p)$ of nonnegative integers such that $\alpha_1+\cdots+\alpha_p=n$. The main result of this paper is an explicit formula for the determinant of the matrix whose entries are $\alpha^\beta=\alpha_1^{\beta_1}\cdots\alpha_p^{\beta_p}$ where $\alpha,\beta\in C(n,p)$. The formula shows that the determinant is positive and has a nice factorization. As an application, it is shown that the polynomials $p_\alpha(x)=(\alpha_1x_1+\cdots+\alpha_px_p)^n$ with $\alpha\in C(n,p)$ form a basis of the vector space $H_n[x_1,\dots,x_p]$ of homogeneous polynomials of degree n in p variables. The result is of interest in the context of global optimization because it allows an explicit representation of polynomials as a difference of convex functions.

Key words. composition, homogeneous polynomial, Vandermonde's determinant, difference of convex functions

AMS subject classifications. 15A05, 15A36, 78M50

PII. S0895479800369141

1. Introduction.

Let n,p be positive integers. A *p-composition* of n is a p-tuple $\alpha=(\alpha_1,\dots,\alpha_p)$ of nonnegative integers α_i such that $\alpha_1+\cdots+\alpha_p=n$. The set of p-compositions of n is denoted by $C(n,p)$. It is well known that its cardinality is

$$|C(n,p)|=\binom{n+p-1}{p-1}.$$

Let $\alpha,\beta\in C(n,p)$. We denote $\alpha^\beta=\alpha_1^{\beta_1}\cdots\alpha_p^{\beta_p}$ where, to be consistent, it is assumed that $0^0=1$. Let $M(n,p)$ be the matrix whose entries are α^β. The power-compositions determinant $\Delta(n,p)$ is the determinant of the matrix $M(n,p)$. The rows and columns of $M(n,p)=(\alpha^\beta)$ are labeled by the compositions given in the same ordering. In this way, the determinant $\Delta(n,p)$ does not depend on the ordering in $C(n,p)$.

The main theorem is the following.

THEOREM 1.1. *The determinant of the matrix $M(n,p)=(\alpha^\beta)$, $\alpha,\beta\in C(n,p)$, is*

$$\Delta(n,p)=\prod_{k=1}^{\min\{n,p\}}\left(n^{\binom{n-1}{k}}\prod_{i=1}^{n-k+1}i^{(n-i+1)\binom{n-i-1}{k-2}}\right)^{\binom{p}{k}}. \tag{1.1}$$

The determinant $\Delta(n,p)$ appears in the following context. Let $H_n[x_1,\dots,x_p]$ be the vector space of homogeneous polynomials of degree n in p variables with coefficients in a field of characteristic zero. The monomials $x^\beta=x_1^{\beta_1}\cdots x_p^{\beta_p}$, for all $\beta\in C(n,p)$, form a basis of $H_n[x_1,\dots,x_p]$. The dimension of $H_n[x_1,\dots,x_p]$ is

*Received by the editors March 16, 2000; accepted for publication (in revised form) by M. Chu April 24, 2001; published electronically September 7, 2001. This work was partially supported by CUR Gen. Cat. under project 1999SGR00356, by Comisión Interministerial de Ciencia y Tecnología (CICYT) under project TIC2000–1017, and by MEC-DGES-SEUID under project PB98-0933.
http://www.siam.org/journals/simax/23-2/36914.html

†Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, C/Pau Gargallo, 5, E-02028, Barcelona, Spain (brunat@ma2.upc.es, montes@ma2.upc.es).

$$\dim H_n[x_1,\ldots,x_p] = |C(n,p)| = \binom{n+p-1}{p-1}.$$

Consider the polynomials

$$p_\alpha(x) = (\alpha_1 x_1 + \cdots + \alpha_p x_p)^n, \qquad \alpha = (\alpha_1,\ldots,\alpha_p) \in C(n,p).$$

The multinomial theorem gives

$$(\alpha_1 x_1 + \cdots + \alpha_p x_p)^n = \sum_{\beta\in C(n,p)} \binom{n}{\beta} \alpha^\beta x^\beta.$$

Since the x^β form a basis, the polynomials $\binom{n}{\beta}x^\beta$ form a basis, too. By Theorem 1.1, $\Delta(n,p) \neq 0$ for all $n,p \geq 1$, therefore we have the following corollary.

COROLLARY 1.2. *The set of polynomials*

$$p_\alpha(x) = (\alpha_1 x_1 + \cdots + \alpha_p x_p)^n, \quad \alpha = (\alpha_1,\ldots,\alpha_p) \in C(n,p)$$

is a basis of $H_n[x_1,\ldots,x_p]$.

In section 8 we will discuss why this result can be of interest in the context of global optimization.

It will be useful to abbreviate the numbers on the right-hand side of (1.1). Define

$$\begin{aligned} D^*(n,k) &= n^{\binom{n-1}{k}} \prod_{i=1}^{n-k+1} i^{(n-i+1)\binom{n-i-1}{k-2}}; \\ D(n,p) &= \prod_{k=1}^{\min\{n,p\}} (D^*(n,k))^{\binom{p}{k}}. \end{aligned} \tag{1.2}$$

With this notation, the theorem to be proved is $\Delta(n,p) = D(n,p)$.

The paper is basically devoted to the proof of Theorem 1.1 and is organized as follows.

In section 2 we define an order $\succ$ of the monomials for which the matrix $M(n,p)$ is block triangular, with zeros at the upper right part. Ordering the monomials x^β is equivalent to ordering the compositions in $C(n,p)$, which are the labels of rows and columns of the matrix $M(n,p)$. In the diagonal blocks we recognize submatrices $M^*(n,k)$ corresponding to proper k-compositions, which are k-compositions with all its entries different from zero. Thus, we can express the determinant $\Delta(n,p)$ in terms of a product of determinants $\Delta^*(n,k)$ of the matrices $M^*(n,k)$ corresponding to proper k-compositions with values of $k \in [\min\{n,p\}] = \{1,\ldots,\min\{n,p\}\}$. Then the proof of $\Delta(n,p) = D(n,p)$ is reduced to the proof of $\Delta^*(n,p) = D^*(n,p)$.

Given m elements $a_1,\ldots,a_m$ in a field, the Vandermonde matrix $V(a_1,\ldots,a_m)$ is the $m \times m$ matrix whose ith column is $1, a_i, \ldots, a_i^{m-1}$. Many generalizations of the Vandermonde matrix have been studied [3, 8, 10], including combinatorial ones [1, 7, 9]. Theorem 1.1 can be considered in this context in the sense that $\Delta^*(n,2)$ can be easily reduced to the Vandermonde's determinant $V(1,1/(n-1),1/(n-2),\ldots,1/2)$. In section 3 we study this determinant and some others special cases which are needed later.

In section 4 it is shown that the proposed ordering splits the matrix $M^*(n,p)$ into new block submatrices and that the diagonal blocks are matrices $M^*(n-j,p-1)$ except for constant factors, denoted $f_1(i,j)$. We proceed with a triangulation method which at each step reduces to zero the block matrices to the right of the diagonal block. This process modifies only the initial block matrices by constant factors $f_k(i,j)$. The method is recursive, but a compact form for the recurrence is not simple to obtain. Nevertheless, the recursion that modifies only the numerical factors $f_k(i,j)$ is independent of p. So comparing with the case $p=2$ that is solved in section 3, we are able to establish an explicit formula for $f_i(i,i)$, which are the only values of f appearing in the recursive formula for the determinant $\Delta^*(n,p)$. The result is a concrete recursion formula for $\Delta^*(n,p)$ in terms of $\Delta^*(i,p-1)$ for i running from $n-1$ till $p-1$.

In section 5 we introduce a class of numbers that we call multiplicative binomial numbers because they have multiplicative properties similar to those of the additive ones for the binomial numbers.

In sections 6 and 7 a recursion rule for $D^*(n,p)$ involving the multiplicative binomial numbers is obtained. It is shown that this recursion is the same and has the same initial values as the recursion satisfied by $\Delta^*(n,p)$. So the proof becomes complete.

Finally, section 8 is concerned with the applications to global optimization theory, which was the initial motivation of the problem. As it is known, representing a function as a difference of convex functions is an essential tool in this context. We show how Corollary 1.2 allows us to do it explicitly for any polynomial function.

Throughout the paper, some elementary properties of binomial numbers are used. They can be found, for instance, in [4].

2. Reduction to proper compositions. A *proper* p-composition of n is a p-composition $\alpha=(\alpha_1,\dots,\alpha_p)$ such that $\alpha_i\geq 1$ for all $i\in[p]$. We denote by $C^*(n,p)$ the set of proper p-compositions of n. Its cardinality is

$$|C^*(n,p)|=\binom{n-1}{p-1}.$$

Let $M^*(n,p)$ be the matrix (α^β) with $\alpha,\beta\in C^*(n,p)$ and let $\Delta^*(n,p)$ be the determinant of $M^*(n,p)$. We first reduce the problem to the calculation of the determinant $\Delta^*(n,p)$. To this end, we need to define an order in $C(n,p)$. Denote by

- $w(\alpha)$ the number of zeros of α;
- $z(\alpha)$ the vector of the positions in increasing order of the zeros in α if $w(\alpha)>0$, and the vector (0) if $w(\alpha)=0$;
- $\ell(v)$ the last nonzero entry of v if v is a vector with nonzero entries, and 0 otherwise.

The order $\succ$ is defined as follows: $\alpha\succ\beta$ if

(i) $w(\alpha)>w(\beta)$ or
(ii) $w(\alpha)=w(\beta)$ and $\ell(z(\alpha)-z(\beta))>0$ or
(iii) $w(\alpha)=w(\beta)$ and $\ell(z(\alpha)-z(\beta))=0$ and $\ell(\alpha-\beta)<0$.

For instance, the compositions of $C(4,3)$ written in decreasing order $\succ$ are shown in Table 2.1.

A composition α with $w(\alpha)=2$ can be identified in a natural way with a composition in $C^*(4,1)$; the compositions α with $w(\alpha)=1$ and $z(\alpha)=(i)$, $i\in[3]$, can be put into one-to-one correspondence with the compositions of $C^*(4,2)$; and the com-

TABLE 2.1
$(C(4,3), \succ)$.

$w(\alpha)$	$z(\alpha)$	Compositions
2	$(2,3)$	(4,0,0),
2	$(1,3)$	(0,4,0),
2	$(1,2)$	(0,0,4),
1	(3)	(3,1,0),(2,2,0),(1,3,0),
1	(2)	(3,0,1),(2,0,2),(1,0,3),
1	(1)	(0,3,1),(0,2,2),(0,1,3),
0	(0)	(2,1,1),(1,2,1),(1,1,2)

positions with $w(\alpha) = 0$ form the set $C^*(4,3)$ of proper 3-compositions of 4. With this ordering, the matrix $M(5,3)$ becomes

$$\left(\begin{array}{ccc|ccc|ccc|ccc|ccc}
256 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 256 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 256 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
81 & 1 & 0 & 27 & 9 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
16 & 16 & 0 & 16 & 16 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 81 & 0 & 3 & 9 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
81 & 0 & 1 & 0 & 0 & 0 & 27 & 9 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
16 & 0 & 16 & 0 & 0 & 0 & 16 & 16 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 81 & 0 & 0 & 0 & 3 & 9 & 27 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 81 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 9 & 3 & 0 & 0 & 0 \\
0 & 16 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 16 & 16 & 0 & 0 & 0 \\
0 & 1 & 81 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 9 & 27 & 0 & 0 & 0 \\
\hline
16 & 1 & 1 & 8 & 4 & 2 & 8 & 4 & 2 & 1 & 1 & 1 & 4 & 2 & 2 \\
1 & 16 & 1 & 2 & 4 & 8 & 1 & 1 & 1 & 8 & 4 & 2 & 2 & 4 & 2 \\
1 & 1 & 16 & 1 & 1 & 1 & 2 & 4 & 8 & 2 & 4 & 8 & 2 & 2 & 4
\end{array}\right).$$

A framed block corresponds to a set of entries such that the rows α of the block have constant values of $w(\alpha)$ and $z(\alpha)$, and the columns β of the block also have constant values of $w(\beta)$ and $z(\beta)$. Inside a block the ordering is given by condition (iii) of the definition, but at this point this is not important; nevertheless it will be significant later on.

From here onward, we assume that the matrix $M(n,p)$ is written with its rows and columns in decreasing $\succ$ order.

PROPOSITION 2.1.

$$\Delta(n,p) = \prod_{k=1}^{\min\{n,p\}} (\Delta^*(n,k))^{\binom{p}{k}}.$$

Proof. First assume $n \geq p$. For $p = 1$ the formulae are obvious. Let $p \geq 2$. Fix a $k \in [p-1]$ and a set of positions $1 \leq i_1 < i_2 < \cdots < i_k \leq p$. The set of rows of the matrix $M(n,p)$ labeled by compositions α with $w(\alpha) = k$ and $z(\alpha) = (i_1, \ldots, i_k)$ are consecutive by the ordering $\succ$, and they are in one-to-one correspondence with the compositions of $C^*(n, n-k)$. The diagonal submatrix of $M(n,p)$ formed by rows α and columns β such that $w(\alpha) = w(\beta)$ and $z(\alpha) = z(\beta)$ is the matrix $M^*(n, n-k)$ (recall the convention $0^0 = 1$).

If $w(\alpha) > w(\beta)$, then there exists a position i such that $\alpha_i = 0$ and $\beta_i \neq 0$. Hence $\alpha^\beta = 0$. If $w(\alpha) = w(\beta)$ and $\ell(z(\alpha) - z(\beta)) \neq 0$, then there also exists a position i such that $\alpha_i = 0$ and $\beta_i \neq 0$. Hence $\alpha^\beta = 0$. It follows that $M(n,p)$ has the following

diagonal form:

$$M(n,p)=\begin{pmatrix}
M^*(n,1) & \cdots & 0 & 0 & \cdots & 0 & \cdots & \cdots & 0\\
\vdots & \ddots & \vdots & \vdots & & \vdots & & & \vdots\\
0 & \cdots & M^*(n,1) & 0 & \cdots & 0 & \cdots & \cdots & 0\\
* & \cdots & * & M^*(n,2) & \cdots & 0 & \cdots & \cdots & 0\\
\vdots & & \vdots & \vdots & \ddots & \vdots & & & \vdots\\
* & \cdots & * & 0 & \cdots & M^*(n,2) & \cdots & \cdots & 0\\
\vdots & & \vdots & \vdots & & \vdots & \ddots & & \vdots\\
\vdots & & \vdots & \vdots & & \vdots & & \ddots & \vdots\\
* & \cdots & * & * & \cdots & * & \cdots & \cdots & M^*(n,p)
\end{pmatrix},$$

where the diagonal block $M^*(n,k)$ is repeated $\binom{p}{k}$ times, the $\binom{p}{k}$ possible positions of the k zeros. Since $\Delta(n,p)$ is the product of the determinants of the diagonal blocks, the formula of the proposition follows.

If $p > n$, the sets $C^*(n,k)$ are empty for $k > n$ and the last diagonal block in the matrix $M(n,p)$ is $M(n,n)$. Then the product is to be taken from $k=1$ to $k=n$. □

According to (1.1) and (1.2), Proposition 2.1 reduces the problem to prove that

$$\Delta^*(n,p) = D^*(n,p) \tag{2.1}$$

for all $n \geq p \geq 1$. In the next section it is shown that (2.1) holds for $p=1$, $p=2$, and $p=n$, and in sections 4 and 6 it will be shown that both $\Delta^*(n,p)$ and $D^*(n,p)$ satisfy the same recurrence.

3. Special cases. There are two extreme cases, namely, $p=1$ and $p=n$. In those cases we have

$$\begin{aligned}
&\Delta^*(n,1) = n^n;\\
&D^*(n,1) = n^{\binom{n-1}{1}}\, 1^0\, 2^0 \cdots (n-1)^0\, n^{\binom{-1}{-1}} = n^{n-1}\, n = n^n;\\
&\Delta^*(n,n) = 1;\\
&D^*(n,n) = n^{\binom{n-1}{n}}\, 1^{n\binom{n-2}{n-2}} = n^0 = 1;
\end{aligned}$$

and we notice the coincidence when binomials are extended in the usual form. We are interested in another special case: $p=2$. In this case the determinant $\Delta^*(n,2)$ can be easily reduced to the well-known Vandermonde's determinant. We will need later the value not only of $\Delta^*(n,2)$ but also of its principal minor of order r, which we denote by $\Delta_r^*(n,2)$. We have

$$\begin{aligned}
\Delta_r^*(n,2) &= \begin{vmatrix}
(n-1)^{n-1}\cdot 1 & (n-1)^{n-2}\cdot 1^2 & \cdots (n-1)^{n-r}\cdot 1^r\\
(n-2)^{n-1}\cdot 2 & (n-2)^{n-2}\cdot 2^2 & \cdots (n-2)^{n-r}\cdot 2^r\\
 & \cdots & \cdots\\
(n-r)^{n-1}\cdot r & (n-r)r^{n-2}\cdot r^2 & \cdots (n-r)^{n-r}\cdot r^r
\end{vmatrix}\\
&= \left(\prod_{i=1}^{r}(n-i)^{n-1}\cdot i\right)\begin{vmatrix}
1 & 1/(n-1) & \cdots & (1/(n-1))^{r-1}\\
 & \cdots & \cdots & \\
1 & r/(n-r) & \cdots & (r/(n-r))^{r-1}
\end{vmatrix}\\
&= \left(\prod_{i=1}^{r}(n-i)^{n-1}\cdot i\right)\prod_{1\leq i<j\leq r}\left(\frac{j}{n-j}-\frac{i}{n-i}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\prod_{i=1}^{r}(n-i)^{n-1}\cdot i\right)\prod_{1\le i<j\le r}\frac{n(j-i)}{(n-i)(n-j)}\\
&= \left(\prod_{i=1}^{r}(n-i)^{n-1}\cdot i\right)n^{\binom{r}{2}}\frac{\prod_{i=1}^{r-1}i^{r-i}}{\prod_{i=1}^{r}(n-i)^{r-1}}\\
&= n^{\binom{r}{2}}\left(\prod_{i=1}^{r}i^{r-i+1}\right)\left(\prod_{i=1}^{r}(n-i)\right)^{(n-r)}. \qquad (3.1)
\end{aligned}
$$

In particular, for $r=n$, we have

$$\Delta^*(n,2)=n^{\binom{n}{2}}\prod_{i=1}^{n}i^{n-i+1},$$

which is the expression given by formula (1.2) for $D^*(n,2)$. Therefore, $\Delta^*(n,p)=D^*(n,p)$ for $p=1$, $p=2$, and $p=n$.

4. A recurrence for $\Delta^*(n,p)$. Unfortunately, the method for $p=2$ cannot be easily generalized for $p\ge 3$. Nevertheless, a triangulation method and the knowledge of the values $\Delta_r^*(p,2)$ lead to a recurrence for $\Delta^*(n,p)$.

Let $\alpha=(\alpha_1,\dots,\alpha_p)\in C^*(n,p)$. We denote $\overline{\alpha}=(\alpha_1,\dots,\alpha_{p-1})$. In the definition of $\succ$, condition (iii) orders the compositions in $C^*(n,p)$ in such a way that compositions α with $\alpha_p=i$ are consecutive and immediately succeeded by those with $\alpha_p=i+1$. The submatrix of $M^*(n,p)$ corresponding to rows α with $\alpha_p=i$ and columns β with $\beta_p=j$ can be written in the form

$$S_{ij}f_1(i,j),\quad i,j\in[n-1],$$

where S_{ij} is the matrix $(\overline{\alpha}^{\overline{\beta}})$ and $f_1(i,j)=i^j$. Observe that $S_{ii}=M^*(n-i,p-1)$. Now we begin a triangulation method that will only modify the factors $f_1(i,j)$. Fix a column β with $\beta_p=j\ge 2$. For each column γ with $\gamma_p=1$ and $\gamma_k\ge\beta_k$ for $k\in[p-1]$, add to the column β the column γ multiplied by

$$-\frac{1}{(n-1)^{j-1}}\binom{j-1}{\overline{\gamma}-\overline{\beta}}\frac{f_1(1,j)}{f_1(1,1)}.$$

Since the differences $\overline{\gamma}-\overline{\beta}$ are all the compositions of $j-1$, the column β has been modified to the value

$$
\begin{aligned}
&\overline{\alpha}^{\overline{\beta}}f_1(i,j)-\sum_{\gamma}\frac{1}{(n-1)^{j-1}}\binom{j-1}{\overline{\gamma}-\overline{\beta}}\frac{f_1(1,j)}{f_1(1,1)}\overline{\alpha}^{\overline{\gamma}}f_1(i,1)\\
=\;&\overline{\alpha}^{\overline{\beta}}\left\{f_1(i,j)-\left(\frac{n-i}{n-1}\right)^{j-1}\frac{f_1(1,j)}{f_1(1,1)}f_1(i,1)\right\}\\
=\;&\overline{\alpha}^{\overline{\beta}}f_2(i,j).
\end{aligned}
$$

The transformed matrix, which has the same determinant as $M^*(n,p)$, is of the form

$$(S_{ij}f_2(i,j)),\quad i,j\in[n-1].$$

Observe that the blocks have been modified only by the numerical factor $f_2(i,j)$. Moreover, for $i=1$ and $j\ge 2$, we have $f_2(i,j)=0$ and the corresponding block is the zero matrix.

The procedure can be iterated. Assume that we have the $(r-1)$-iterated matrix in the form

$$\left(S_{ij} f_r(i,j)\right), \quad i,j \in [n-1],$$

with $f_k(i,j) = 0$ for $k < r$ and $j > i$. Select a column β with $\beta_p = j > r$. For each column γ with $\gamma_p = r$ and $\gamma_k \geq \beta_k$ for $k \in [p-1]$, add to the column β the column γ multiplied by

$$-\frac{1}{(n-r)^{j-r}} \binom{j-r}{\overline{\gamma}-\overline{\beta}} \frac{f_r(r,j)}{f_r(r,r)}.$$

The differences $\overline{\beta}-\overline{\gamma}$ are all the compositions of $j-r$. Then, the column β has been modified to the value

$$\begin{aligned} & \overline{\alpha}^{\overline{\beta}} f_r(i,j) - \sum_{\gamma} \frac{1}{(n-r)^{j-r}} \binom{j-r}{\overline{\gamma}-\overline{\beta}} \frac{f_r(r,j)}{f_1(r,r)} \overline{\alpha}^{\overline{\gamma}} f_r(i,r) \\ = \; & \overline{\alpha}^{\overline{\beta}} \left\{ f_r(i,j) - \left(\frac{n-i}{n-r}\right)^{j-r} \frac{f_r(r,j)}{f_r(r,r)} f_r(i,r) \right\} \\ = \; & \overline{\alpha}^{\overline{\beta}} f_{r+1}(i,j), \end{aligned}$$

and $f_{r+1}(i,j) = 0$ for $i < r+1$ and $j > i$. We have the recurrence

$$f_{r+1}(i,j) = f_r(i,j) - \left(\frac{n-i}{n-r}\right)^{j-r} \frac{f_r(r,j)}{f_r(r,r)} f_r(i,r), \tag{4.1}$$

and, after $n-p$ iterations, the matrix becomes

$$\left(S_{ij} f_{n-p+1}(i,j)\right), \quad i,j \in [n-1],$$

with $f_{n-p+1}(i,j) = 0$ for $j > i$. Therefore $\Delta^*(n,p)$ is the product of the determinants of the diagonal blocks $S_{ii} f_i(i,i) = M^*(n-i,p-1) f_i(i,i)$. Denoting $f_i = f_i(i,i)$, we have

$$\Delta^*(n,p) = \left(\prod_{i=1}^{n-p+1} f_i^{\binom{n-i-1}{p-2}} \right) \left(\prod_{i=1}^{n-p+1} \Delta^*(n-i,p-1) \right).$$

The recurrence (4.1) does not seem easy to solve. Nevertheless, note that it does not depend on p. In the product

$$F(n,p) = \prod_{i=1}^{n-p+1} f_i^{\binom{n-i-1}{p-2}} \tag{4.2}$$

the exponents and the number of factors depend on p, but the f_i's themselves do not. Then, consider the case $p = 2$ and apply the method for calculating the minors of order r and $r-1$. We have

$$\frac{\Delta_r^*(n,2)}{\Delta_{r-1}^*(n,2)} = \frac{\left(\prod_{i=1}^{r} f_i\right)\left(\prod_{i=1}^{r} (n-i)^{n-i}\right)}{\left(\prod_{i=1}^{r-1} f_i\right)\left(\prod_{i=1}^{r-1} (n-i)^{n-i}\right)} = (n-r)^{n-r} f_r. \tag{4.3}$$

On the other hand, the quotient of the minors can be evaluated directly by (3.1):

$$\begin{aligned}\frac{\Delta_r^*(n,2)}{\Delta_{r-1}^*(n,2)} &= \frac{n^{\binom{r}{2}}\displaystyle\prod_{i=1}^{r} i^{r-i+1}\prod_{i=1}^{r}(n-i)^{(n-r)}}{n^{\binom{r-1}{2}}\displaystyle\prod_{i=1}^{r-1} i^{r-i}\prod_{i=1}^{r-1}(n-i)^{(n-r+1)}} \\ &= \frac{n^{r-1}\left(\displaystyle\prod_{i=1}^{r} i\right)(n-r)^{n-r}}{\displaystyle\prod_{i=1}^{r-1}(n-i)} \\ &= \frac{n^r}{\binom{n}{r}}(n-r)^{n-r}. \qquad (4.4)\end{aligned}$$

Then, comparing (4.3) to (4.4), we obtain

$$f_r = \frac{n^r}{\binom{n}{r}} = r!\frac{n^{r-1}}{(n-1)\cdots(n-r+1)}, \qquad (4.5)$$

which gives an explicit formula for $F(n,p)$.

We have proved that the determinants $\Delta^*(n,p)$ satisfy the recurrence

$$\Delta^*(n,p) = F(n,p)\prod_{i=1}^{n-p+1}\Delta^*(n-i,p-1), \qquad (4.6)$$

with $F(n,p)$ given by (4.2) and (4.5). The last part of the proof consists of proving that the numbers $D^*(n,p)$ satisfy the same recurrence.

5. The multiplicative binomial numbers. To obtain a recurrence for $D^*(n,p)$ we introduce a class of numbers which have similar properties with respect to the product to those of the binomial numbers with respect to the sum.

Let $n \geq p \geq 1$ be integers. The *multiplicative binomial number* $a(n,p)$ is the number

$$a(n,p) = \prod_{i=1}^{n-p+1} i^{\binom{n-i}{p-1}} = 1^{\binom{n-1}{p-1}}\cdot 2^{\binom{n-2}{p-1}}\ \cdots\ (n-p+1)^{\binom{p-1}{p-1}}.$$

From the definition, we have

$$\begin{aligned} a(n,1) &= n!, \\ a(n,2) &= 1^{n-1}\,2^{n-2}\ldots(n-1), \\ a(n,n-1) &= 2, \\ a(n,n) &= 1. \end{aligned} \qquad (5.1)$$

A straightforward calculation and the binomial recurrence $\binom{n-k}{p-i} + \binom{n-k}{p-i-1} = \binom{n-k+1}{p-i}$ shows that

$$a(n,p) = a(n-1,p-1)\,a(n-1,p). \qquad (5.2)$$

TABLE 5.1

$n \setminus p$	0	1	2	3	4	5	
1	2	1					
2	3	2	1				
3	4	6	2	1			
4	5	24	12	2	1		
5	6	120	288	24	2	1	
6	7	720	34560	6912	48	2	1

Moreover,

$$\begin{aligned} a(n,p) &= a(n-1,p-1)\,a(n-1,p) \\ &= a(n-1,p-1)\,a(n-2,p-1)\,a(n-2,p) \\ &\quad \cdots \\ &= a(n-1,p-1)\,a(n-2,p-1)\cdots a(p-1,p-1). \end{aligned} \tag{5.3}$$

The properties (5.2) and (5.3) allow us to translate every additive property of the binomial numbers into a multiplicative one of the numbers $a(n,p)$.

If we want to define $a(n,p)$ also for $p=0$, then (5.1) and (5.2) give

$$(n+1)! = a(n+1,1) = a(n,0)\,a(n,1) = a(n,0)n!$$

and $a(n,0)$ must be defined as

$$a(n,0) = n+1.$$

Table 5.1 gives some values of $a(n,p)$. It is constructed as the Pascal triangle of binomial numbers but uses multiplication instead of addition.

6. A recurrence for $D^*(n,p)$. The number $D^*(n,p)$ has been defined in (1.2) by

$$D^*(n,p) = n^{\binom{n-1}{p}} \prod_{i=1}^{n-p+1} i^{(n-i+1)\binom{n-i-1}{p-2}}. \tag{6.1}$$

Then we have

$$\begin{aligned} &D^*(n-1,p-1)D^*(n-1,p) \\ &\quad = \left((n-1)^{\binom{n-2}{p-1}} \prod_{i=1}^{n-p+1} i^{(n-i)\binom{n-i-2}{p-3}}\right) \cdot \left((n-1)^{\binom{n-2}{p}} \prod_{i=1}^{n-p} i^{(n-i)\binom{n-i-2}{p-2}}\right) \\ &\quad = (n-1)^{\binom{n-1}{p}} \prod_{i=1}^{n-p+1} i^{(n-i)\binom{n-i-1}{p-2}}. \end{aligned}$$

Comparing this with (6.1) leads to

$$\begin{aligned} D^*(n,p) &= \left(\frac{n}{n-1}\right)^{\binom{n-1}{p}} \left(\prod_{i=1}^{n-p+1} i^{\binom{n-i-1}{p-2}}\right) D^*(n-1,p-1)D^*(n-1,p) \\ &= \left(\frac{n}{n-1}\right)^{\binom{n-1}{p}} a(n-1,p-1)D^*(n-1,p-1)D^*(n-1,p). \end{aligned}$$

Let us define $D^{**}(n,p)$ by

$$D^*(n,p) = n^{\binom{n-1}{p}} D^{**}(n,p). \tag{6.2}$$

We have

$$D^{**}(n,p) = a(n-1,p-1)D^{**}(n-1,p-1)D^{**}(n-1,p).$$

Iterating and taking into account (5.3) and that $D^*(p,p) = 1$ for any p, we obtain

$$D^{**}(n,p) = a(n,p)\prod_{i=1}^{n-p} D^{**}(n-i,p-1).$$

Substituting (6.2), we obtain

$$D^*(n,p) = b(n,p)a(n,p)\prod_{i=1}^{n-p} D^*(n-i,p-1), \tag{6.3}$$

where

$$b(n,p) = \frac{n^{\binom{n-1}{p}}}{\prod_{i=1}^{n-p}(n-i)^{\binom{n-i-1}{p-1}}}. \tag{6.4}$$

7. The last step.

LEMMA 7.1. $F(n,p) = b(n,p)a(n,p)$.

Proof. From the definition (4.2) of $F(n,p)$ and the values (4.5) of f_i, we have

$$\begin{aligned} F(n,p) &= \prod_{i=1}^{n-p+1} f_i^{\binom{n-i-1}{p-2}} \\ &= \prod_{i=1}^{n-p+1} \left(i! \frac{n^{i-1}}{(n-1)\cdots(n-i+1)} \right)^{\binom{n-i-1}{p-2}} \\ &= \left(\prod_{i=1}^{n-p+1} i!^{\binom{n-i-1}{p-2}} \right) \prod_{i=1}^{n-p+1} \left(\frac{n^{i-1}}{(n-1)\cdots(n-i+1)} \right)^{\binom{n-i-1}{p-2}}. \end{aligned}$$

In the product of powers of factorials, an $i \in [n-p+1]$ has the exponent

$$\binom{n-i-1}{p-2} + \binom{n-i-2}{p-2} + \cdots + \binom{p-2}{p-2} = \binom{n-i}{p-1}.$$

So the first product is

$$\prod_{i=1}^{n-p+1} i!^{\binom{n-i-1}{p-2}} = \prod_{i=1}^{n-p+1} i^{\binom{n-i}{p-1}} = a(n,p).$$

In the second product, the exponent of n is

$$\binom{n-3}{p-2} + 2\binom{n-4}{p-2} + \cdots + (n-p)\binom{p-2}{p-2} = \binom{n-1}{p}$$

and the exponent of $n-i$ is

$$-\binom{n-i-2}{p-2}-\binom{n-i-3,}{p-2}-\cdots-\binom{p-2}{p-2}=-\binom{n-i-1}{p-1}.$$

So we have

$$F(n,p)=\frac{a(n,p)\,n^{\binom{n-1}{p}}}{\prod_{i=1}^{n-p}(n-i)^{\binom{n-i-1}{p-1}}}=a(n,p)\,b(n,p). \qquad \square$$

THEOREM 7.2. *The determinant of the matrix $M^*(n,p)$ is*

$$\Delta^*(n,p)=D^*(n,p)=n^{\binom{n-1}{k}}\prod_{i=1}^{n-k+1} i^{(n-i+1)\binom{n-i-1}{k-2}}.$$

Proof. By induction on p. For $p=1,2$ the equality has been proved in section 2. Assume that $p\geq 3$ and that the theorem holds for $p-1$. Then (4.6) gives

$$\Delta^*(n,p)=F(n,p)\prod_{i=1}^{n-p+1}\Delta^*(n-i,p-1).$$

By Lemma 7.1 and the induction hypothesis, we have

$$\Delta^*(n,p)=b(n,p)a(n,p)\prod_{i=1}^{n-p+1}D^*(n-i,p-1),$$

and by (6.3)

$$\Delta^*(n,p)=D^*(n,p). \qquad \square$$

COROLLARY 7.3. *Let $n\geq p\geq 1$ be integers. The set of polynomials*

$$p_\alpha(x)=(\alpha_1x_1+\cdots+\alpha_px_p)^n, \qquad \alpha=(\alpha_1,\ldots,\alpha_p)\in C^*(n,p)$$

is a basis of the subspace $\langle x^\alpha:\alpha\in C^*(n,p)\rangle$ of $H_n[x_1,\ldots,x_p]$ generated by the proper monomials.

Corollaries 1.2 and 7.3 can be interpreted in the context of Gröbner bases and convex polytopes [11]. Corollary 1.2 states that the ideal I generated by the convex polynomials $p_\alpha(x)$ is the monomial ideal $\langle x^\beta:\beta\in C(n,p)\rangle$, for which the set $\{x^\beta:\beta\in C(n,p)\}$ is a universal Gröbner basis. The same is true for Corollary 7.3 relative to proper compositions. We note that the natural extension of the order $\succ$ defined in section 2 does not have the monotony property and, consequently, is not a monomial order.

8. Polynomials as difference of convex functions. In this section we show how our result can be of interest in the context of global optimization. We refer to [12] for details about convex sets and functions and difference of convex (d.c.) functions.

A subset $\Omega\subset\mathbb{R}^p$ is *convex* if $(1-\lambda)x+\lambda y\in\Omega$ whenever $x,y\in\Omega$, $\lambda\in\mathbb{R}$, $0\leq\lambda\leq 1$. Let $\Omega\subset\mathbb{R}^p$ be a convex set. A function $f\colon\Omega\to\mathbb{R}$ is *convex* if $f((1-\lambda)x+\lambda y)\leq(1-\lambda)f(x)+\lambda f(x)$ whenever $x,y\in\Omega$, $\lambda\in\mathbb{R}$, $0\leq\lambda\leq 1$. A function

$f\colon \Omega \to \mathbb{R}$ is a *difference of two convex functions* or a *d.c.* function if there exists two convex functions $f_1(x)$ and $f_2(x)$ on Ω such that $f(x) = f_1(x) - f_2(x)$ for all $x \in \Omega$. The class $DC(\Omega)$ of d.c. functions on Ω is the smallest vector space containing all convex functions on Ω. H. Tuy [12, p. ix] has pointed out that "... virtually every nonconvex optimization problem can be described in terms of functions representable as differences of convex functions...." A theorem by Hiriart-Urruty [6] proves that every function $f \in C^2(\mathbb{R}^p)$ is a d.c. function on any compact convex set $\Omega \subset \mathbb{R}^p$. In particular, any polynomial function is a d.c. function on any compact convex set. Nevertheless, how to represent effectively a polynomial as a difference of two convex polynomials has been considered a difficult problem, which arises, for instance, in the short-term hydrothermal coordination of electricity generation problem [2, 5].

Let $\mathbb{R}_+$ be the set of nonnegative real numbers. The linear functions $L_\alpha(x) = \alpha_1 x_1 + \cdots + \alpha_p x_p$, where $\alpha \in C(n,p)$, are convex and nonnegative in the convex set $\mathbb{R}^p_+$. Since the functions $t \mapsto t^n$ are increasing in $\mathbb{R}_+$, the functions $p_\alpha(x) = (L_\alpha(x))^n$ are convex in $\mathbb{R}^p_+$. Corollary 1.2 implies that, given a homogeneous polynomial $h(x) \in H_n[x_1, \ldots, x_p]$, the polynomial $h(x)$ can be expressed as

$$h(x) = \sum_{\alpha \in C(n,p)} \lambda_\alpha p_\alpha(x),$$

where the coefficients λ_α can be found from the coefficients of $h(x)$ and the inverse of $M(n,p)$. Let P and N be the set of $\alpha \in C(n,p)$ such that $\lambda_\alpha > 0$ and $\lambda_\alpha < 0$, respectively. Then

$$h(x) = \left(\sum_{\alpha \in P} \lambda_\alpha p_\alpha(x) \right) - \left(\sum_{\alpha \in N} (-\lambda_\alpha) p_\alpha(x) \right)$$

is an explicit representation of $h(x)$ as a difference of two convex functions on $\mathbb{R}^p_+$.

If $f(x) = f(x_1, \ldots, x_p)$ is a polynomial of total degree m, then $f(x) = h_0(x) + \cdots + h_m(x)$, where $h_j(x) \in H_j[x_1, \ldots, x_p]$. From the above representation of each $h_j(x)$ as a d.c. function, we obtain an expression of $f(x)$ as a difference of two convex functions on $\mathbb{R}^p_+$.

REFERENCES

[1] M. W. Buck, R. A. Coley, and D. P. Robbins, *A generalized Vandermonde determinant*, J. Algebraic Combin., 1 (1992), pp. 105–109.

[2] A. Ferrer, *Decomposition of a polynomial function as a difference of convex polynomials. Application*, in Generalized Convexity and Generalized Monotonicity, Proceedings of the 6th International Symposium on Generalized Coverxity/Monotonicity, N. Hadjisavvas, J. E. Martinez-Legaz, and J. P. Penot, eds., Springer-Verlag, New York, 2001, pp. 189–207.

[3] R. P. Flowe and G. A. Harris, *A note on generalized Vandermonde determinants*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1146–1151.

[4] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, Reading, MA, 1989.

[5] F. J. Heredia and N. Nabona, *Optimum short-term hydrothermal scheduling with spinning reverse through network flows*, IEEE Trans. Power Systems, 10 (1995), pp. 1642–1651.

[6] J.-B. Hiriart-Urruty, *Generalized differentiability, duality and optimization for problems dealing with differences of convex functions*, in Convexity and Duality in Optimization, Lecture Notes in Econom. Math. Systems 256, Springer-Verlag, Berlin, 1985, pp. 37–69.

[7] C. Krattenthaler, *Advanced determinant calculus*, Sém. Lothar. Combin., 42 (1999), Art. B42q, 67 pp. (electronic).

[8] T. Y. Lam, *A general theory of Vandermonde matrices*, Expo. Math., 4 (1986), pp. 193–215.

[9] M. Lassalle, *Quelques conjectures combinatoires relatives à la formule classique de Chu-Vandermonde*, Adv. Appl. Math., 21 (1998), pp. 457–472.
[10] D. R. Richman and Q. L. Wang, *On generalized Vandermonde determinants*, in Current Trends in Matrix Theory, North-Holland, New York, 1987, pp. 285–295.
[11] B. Sturmfels, *Gröbner Bases and Convex Polytopes*, AMS, Providence, RI, 1996.
[12] H. Tuy, *Convex Analysis and Global Optimization*, Kluwer Academic, Dordrecht, The Netherlands, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 472–493
© 2001 Society for Industrial and Applied Mathematics

ANALYSIS OF THE CHOLESKY METHOD WITH ITERATIVE REFINEMENT FOR SOLVING THE SYMMETRIC DEFINITE GENERALIZED EIGENPROBLEM*

PHILIP I. DAVIES†, NICHOLAS J. HIGHAM†, AND FRANÇOISE TISSEUR†

Abstract. A standard method for solving the symmetric definite generalized eigenvalue problem $Ax = \lambda Bx$, where A is symmetric and B is symmetric positive definite, is to compute a Cholesky factorization $B = LL^T$ (optionally with complete pivoting) and solve the equivalent standard symmetric eigenvalue problem $Cy = \lambda y$, where $C = L^{-1}AL^{-T}$. Provided that a stable eigensolver is used, standard error analysis says that the computed eigenvalues are exact for $A + \Delta A$ and $B + \Delta B$ with $\max(\|\Delta A\|_2/\|A\|_2, \|\Delta B\|_2/\|B\|_2)$ bounded by a multiple of $\kappa_2(B)u$, where u is the unit roundoff. We take the Jacobi method as the eigensolver and give a detailed error analysis that yields backward error bounds potentially much smaller than $\kappa_2(B)u$. To show the practical utility of our bounds we describe a vibration problem from structural engineering in which B is ill conditioned yet the error bounds are small. We show how, in cases of instability, iterative refinement based on Newton's method can be used to produce eigenpairs with small backward errors. Our analysis and experiments also give insight into the popular Cholesky–QR method, in which the QR method is used as the eigensolver. We argue that it is desirable to augment current implementations of this method with pivoting in the Cholesky factorization.

Key words. symmetric definite generalized eigenvalue problem, Cholesky method, Cholesky factorization with complete pivoting, Jacobi method, backward error analysis, rounding error analysis, iterative refinement, Newton's method, LAPACK, MATLAB

AMS subject classification. 65F15

PII. S0895479800373498

1. Introduction.

The symmetric definite generalized eigenvalue problem

$$Ax = \lambda Bx, \tag{1.1}$$

where $A, B \in \mathbb{R}^{n\times n}$ are symmetric and B is positive definite, arises in many applications in science and engineering [4, chapter 9], [16]. An important open problem is to derive a method of solution that takes advantage of the structure and is efficient and backward stable. Such a method should, for example, require half the storage of a method for the generalized nonsymmetric problem and produce real computed eigenvalues.

The QZ algorithm [18] can be used to solve (1.1). It computes orthogonal matrices Q and Z such that Q^TAZ is upper quasi-triangular and Q^TBZ is upper triangular. This method is numerically stable but it does not exploit the special structure of the problem and so does not necessarily produce real eigenpairs in floating point arithmetic.

*Received by the editors June 9, 2000; accepted for publication (in revised form) by M. Chu April 11, 2001; published electronically September 7, 2001.

http://www.siam.org/journals/simax/23-2/37349.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (ieuan@ma.man.ac.uk, http://www.ma.man.ac.uk/~ieuan/, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/, ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). The work of the first author was supported by an Engineering and Physical Sciences Research Council CASE Ph.D. Studentship with NAG Ltd. (Oxford) as the cooperating body. The work of the second author was supported by Engineering and Physical Sciences Research Council grant GR/L76532 and a Royal Society Leverhulme Trust Senior Research Fellowship. The work of the third author was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

A method that potentially has the desired properties has recently been proposed by Chandrasekaran [3], but the worst-case computational cost of this algorithm is not clear.

A standard method, apparently first suggested by Wilkinson [25, pp. 337–340], begins by computing the Cholesky factorization, optionally with complete pivoting [12, section 4.2.9], [14, section 10.3],

$$\Pi^T B \Pi = L D^2 L^T, \tag{1.2}$$

where Π is a permutation matrix, L is unit lower triangular, and $D^2 = \operatorname{diag}(d_i^2)$ is diagonal. The problem (1.1) is then reduced to the form

$$Cy \equiv D^{-1}L^{-1}\Pi^T A \Pi L^{-T} D^{-1} y = \lambda y, \qquad y = DL^T \Pi^T x. \tag{1.3}$$

Any method for solving the symmetric eigenvalue problem can now be applied to C [6], [19]. In LAPACK's `xSYGV` driver, (1.1) is solved by applying the QR algorithm to (1.3). MATLAB 6's `eig` function does likewise when it is given a symmetric definite generalized eigenproblem. As is well known, when B is ill conditioned numerical stability can be lost in the Cholesky-based method. However, it is also known that methods based on factorizing B and converting to a standard eigenvalue problem have some attractive features. In reference to the method that uses a spectral decomposition of B, Wilkinson [25, p. 344] states that

> In the ill-conditioned case the method of §68 has certain advantages in that "all the condition of B" is concentrated in the small elements of D. The matrix P of (68.5) [our C in (1.3)] has a certain number of rows and columns with large elements (corresponding to small d_{ii}) and eigenvalues of $(A - \lambda B)$ of normal size are more likely to be preserved.

In this work we aim to give new insight into the numerical behavior of the Cholesky method.

First, we make a simple but important observation about numerical stability. Assume that the Cholesky factorization is computed exactly and set $\Pi = I$ without loss of generality. We compute $\widehat{C} = C + \Delta C_1$ where, at best, ΔC_1 satisfies a bound of the form

$$|\Delta C_1| \le c_n u |D^{-1}||L^{-1}||A||L^{-T}||D^{-1}|,$$

where c_n is a constant and u is the unit roundoff (see section 3 for the floating point arithmetic model). Here, $|A| = (|a_{ij}|)$. Solution of the eigenproblem for $\widehat{C}$ can be assumed to yield the exact eigensystem of $\widehat{C} + \Delta C_2$ for some ΔC_2. Therefore the computed eigensystem is the exact eigensystem of

$$C + \Delta C_1 + \Delta C_2 = D^{-1}L^{-1}\big(A + \Delta A\big)L^{-T}D^{-1}, \qquad \Delta A = LD(\Delta C_1 + \Delta C_2)DL^T,$$

and

$$\begin{aligned} |\Delta A| &\le |L||D|\big(c_n u |D^{-1}||L^{-1}||A||L^{-T}||D^{-1}| + |\Delta C_2|\big)|D||L^T| \\ &\le c_n u |L||L^{-1}||A||L^{-T}||L^T| + |L||D||\Delta C_2||D||L^T|. \end{aligned} \tag{1.4}$$

If we are using complete pivoting in the Cholesky factorization then $|l_{ij}| \le 1$ for $i > j$ and

$$d_1^2 \ge \cdots \ge d_n^2 > 0. \tag{1.5}$$

Hence [14, Theorem 8.13]

$$\kappa_p(L) = \|L\|_p\|L^{-1}\|_p \le n2^{n-1}, \qquad p = 1, 2, \infty \tag{1.6}$$

(with approximate equality achieved for L^T the Kahan matrix [14, p. 161]), and so the first term in (1.4) is bounded independently of $\kappa(B)$. The second term will have the same property provided that ΔC_2 satisfies a bound of the form

$$|\Delta C_2| \le |D^{-1}|f(|A|, |L^{-1}|, u)|D^{-1}|,$$

where f is a matrix depending on $|A|$, $|L^{-1}|$, and u, but not $|D^{-1}|$.

If nothing more is known about ΔC_2 than that $\|\Delta C_2\| \le c_n u\|C\|$ (corresponding to using a normwise backward stable eigensolver for C), then the best bound we can obtain in terms of the original data is of the form

$$\|\Delta A\| \le g(n)u\kappa(B)\|A\|. \tag{1.7}$$

However, this analysis shows that there is hope for obtaining a bound without the factor $\kappa(B)$ if the eigensolver for C respects the scaling of C when D is ill conditioned. The QL variant of the QR algorithm has this property in many instances, since when D is ill conditioned the inequalities (1.5) imply that C is graded upward (that is, its elements generally increase from top left to bottom right) and the backward error matrix for the QL algorithm[1] then tends to be graded in the same way [19, chapter 8], [21, p. 337]. However, this is a heuristic and we know of no precise results.

In this work we show that if, instead of the QL and QR algorithms, the Jacobi method is applied to C, then we can derive rigorous backward error bounds that can be significantly smaller than bounds involving a factor $\kappa(B)$ when B is ill conditioned. We also give experimental evidence of the benefits of pivoting in the Cholesky–QR method.

Wilkinson [26] expressed the view that for most of the standard problems in numerical linear algebra iterative refinement is a valuable tool for which it is worth developing software. We investigate iterative refinement as a means for improving the backward errors of eigenpairs computed by the Cholesky–QR and Cholesky–Jacobi methods.

The organization of the paper is as follows. In section 2 we describe the Cholesky–Jacobi method and in section 3 we give a detailed rounding error analysis, making use of a diagonal scaling idea of Anjos, Hammarling, and Paige [2]. In section 4 we show how fixed precision iterative refinement can be used to improve the stability of selected eigenpairs. Section 5 contains a variety of numerical examples. In particular, we describe a vibration problem from structural engineering where B is ill conditioned yet our backward error bounds for the Cholesky–Jacobi method are found to be of order u, and we give examples where ill condition of B does cause instability of the method but iterative refinement cures the instability. Conclusions are given in section 6.

In our analysis $\|\cdot\|$ denotes any vector norm and the corresponding subordinate matrix norm, while $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the 2-norm and the Frobenius norm, respectively.

[1] For the original QR algorithm, we need C to be graded downward. However, the distinction is unimportant for our purposes since LAPACK's routines for the QR algorithm [1] include a strategy for switching between the QL and QR variants and thus automatically take advantage of either form of grading.

2. Method outline. The Cholesky–Jacobi method computes the Cholesky factorization with complete pivoting (1.2), forms

$$H_0 = D^{-1}L^{-1}\Pi^T A \Pi L^{-T} D^{-1} \tag{2.1}$$

in (1.3), and then applies Jacobi's method for the symmetric eigenproblem to H_0. Peters and Wilkinson [20] note that a variant of this method in which the Cholesky factorization of B is replaced by a spectral decomposition, computed also by the Jacobi method, was used by G. H. Golub on the Illiac at the University of Illinois in the 1950s.

Jacobi's method constructs a sequence of similar matrices starting with H_0. An orthogonal transformation is applied at each step,

$$H_{k+1} = Q_k^T H_k Q_k$$

in such a way that H_k tends to diagonal form $\Lambda = \operatorname{diag}(\lambda_i)$ as $k \to \infty$. Denoting by $Q = Q_0 Q_1 \ldots$ the product of the orthogonal transformations that diagonalizes H_0 and writing $X = \Pi L^{-T} D^{-1} Q$, we have, overall,

$$X^T A X = \Lambda, \quad X^T B X = I. \tag{2.2}$$

Thus X simultaneously diagonalizes A and B and is also easily seen to be a matrix of eigenvectors.

Now we describe the method in more detail. At the kth stage let Q_k be a Jacobi rotation in the (i, j) plane $(i \le j)$ such that $Q_k^T H_k Q_k$ has zeros in positions (i, j) and (j, i). Using MATLAB notation,

$$Q_k([i\ j], [i\ j]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{2.3}$$

where $c = \cos\theta$ and $s = \sin\theta$ are obtained from [12, section 8.4.2] (with $\operatorname{sign}(0) = 1$)

$$\tau = \frac{h_{jj} - h_{ii}}{2h_{ij}}, \tag{2.4}$$

$$t = \frac{\operatorname{sign}(\tau)}{|\tau| + \sqrt{1+\tau^2}}, \tag{2.5}$$

$$c = \frac{1}{\sqrt{1+t^2}}, \quad s = tc. \tag{2.6}$$

The corresponding rotation angle θ satisfies $|\theta| \le \pi/4$; choosing a small rotation angle is essential for the convergence theory [19, chapter 9]. We choose the index pairs (i, j) from a row cyclic ordering, in which a complete sweep has the form

$$(i, j) = (1,2), \ldots, (1,n), (2,3), \ldots, (2,n), \ldots, (n-1, n). \tag{2.7}$$

For this ordering and the choice of angle above, the Jacobi method converges quadratically [12, section 8.4.4], [19, section 9.4].

When forming $H_{k+1} = Q_k^T H_k Q_k = (\widetilde{h}_{ij})$ we explicitly set $\widetilde{h}_{ij} = 0$ and compute the new diagonal elements from [19, equation (9.9)]

$$\widetilde{h}_{ii} = h_{ii} - h_{ij}t, \tag{2.8}$$

$$\widetilde{h}_{jj} = h_{jj} + h_{ij}t, \tag{2.9}$$

where t is given in (2.5). The complete algorithm is summarized as follows.

ALGORITHM 2.1 (Cholesky–Jacobi method). Given $A, B \in \mathbb{R}^{n\times n}$ with A symmetric and B symmetric positive definite, this algorithm calculates the eigenvalues λ_i and corresponding eigenvectors x_i of the pair (A, B).

1. Compute the Cholesky factorization with complete pivoting $\Pi^T B \Pi = LD^2L^T$.
   Form $H = D^{-1}L^{-1}\Pi^T A \Pi L^{-T}D^{-1}$ by solving triangular systems.
   $X = \Pi L^{-T}D^{-1}$.
2. % Jacobi's method
   done_rot = true
   while done_rot = true
   done_rot = false
   for $i = 1\colon n$
   for $j = i+1\colon n$
   $(*)$ if $|h_{ij}| > u\sqrt{|h_{ii}h_{jj}|}$
   done_rot = true
   Form $Q_{ij} \equiv Q_k([i\ j],[i\ j])$ using (2.3)–(2.6).
   ind $= [1\colon i-1, i+1\colon j-1, j+1\colon n]$
   $H([i\ j], \text{ind}) = Q_{ij}^T H([i\ j], \text{ind})$
   $H(\text{ind}, [i\ j]) = H(\text{ind}, [i\ j])Q_{ij}$
   $H([i\ j],[i\ j]) = \begin{bmatrix} \widetilde{h}_{ii} & 0 \\ 0 & \widetilde{h}_{jj} \end{bmatrix}$ using (2.8), (2.9)
   $X(:, [i\ j]) = X(:, [i\ j])Q_{ij}$
   end
   end
   end
   end
   $\lambda_i = h_{ii}$, $x_i = X(:, i)$, $i = 1\colon n$

The test $(*)$ for whether to apply a rotation is adapted from the one used for Jacobi's method for a symmetric positive definite matrix [7]—we have added absolute values inside the square root since h_{ii} and h_{jj} can be negative. This test is too stringent in general and can cause the algorithm not to converge, but we have found it generally works well, and so we used it in our experiments in order to achieve the best possible numerical behavior.

3. Error analysis. Now we give an error analysis for Algorithm 2.1, with the aim of obtaining an error bound better than (1.7). We use the standard model for floating point arithmetic

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\delta_1) = \frac{x \operatorname{op} y}{1+\delta_2}, \quad |\delta_1|, |\delta_2| \le u, \quad \operatorname{op} = +, -, *, /,$$
$$fl(\sqrt{x}) = \sqrt{x}(1+\delta), \quad |\delta| \le u,$$

where u is the unit roundoff. We will make use of the following lemma [14].

LEMMA 3.1. *If $|\delta_i| \le u$ and $\rho_i = \pm 1$ for $i = 1\colon n$, and $nu < 1$, then*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n, \quad \textit{where} \quad |\theta_n| \le \frac{nu}{1-nu} =: \gamma_n.$$

We define

$$\tilde{\gamma}_k = \frac{pku}{1-pku},$$

where p denotes a small integer constant whose exact value is unimportant. We will also write $\tilde{\theta}_k$ to denote a quantity satisfying $|\tilde{\theta}_k| \leq \tilde{\gamma}_k$. Computed quantities are denoted with a hat.

We consider first the second part of Algorithm 2.1, beginning with the construction of the Jacobi rotation.

LEMMA 3.2. *Let a Jacobi rotation Q_k be constructed using (2.4)–(2.6) so that $Q_k^T H_k Q_k$ has zeros in the (i,j) and (j,i) positions. The computed $\widehat{c}$, $\widehat{s}$, and $\widehat{t}$ satisfy*

$$\widehat{c} = c(1+\tilde{\theta}_1), \quad \widehat{s} = s(1+\tilde{\theta}_1'), \quad \widehat{t} = t(1+\tilde{\theta}_1''),$$

where c, s, and t are the exact values for H_k.

Proof. The proof is straightforward. ∎

In most of the rest of our analysis we will assume that the computed $\widehat{c}$, $\widehat{s}$, and $\widehat{t}$ are exact. It is easily checked that, in view of Lemma 3.2, this simplification does not affect the bounds.

LEMMA 3.3. *If one step of Jacobi's method is performed in the (i,j) plane on the matrix H_m then the computed $\widehat{H}_{m+1}$ satisfies*

$$\widehat{H}_{m+1} = Q_m^T \left(H_m + \Delta H_m\right) Q_m,$$

where the elements of ΔH_m are bounded componentwise by

$$\left.\begin{array}{l} |\Delta h_{ik}| \leq \tilde{\gamma}_1 \left(|h_{ik}| + 2|sc||h_{jk}|\right) \\ |\Delta h_{jk}| \leq \tilde{\gamma}_1 \left(|h_{jk}| + 2|sc||h_{ik}|\right) \end{array}\right\} \quad k \neq i,j,$$

and

$$\begin{aligned} |\Delta h_{ii}| &\leq \tilde{\gamma}_1 \left(c^2|h_{ii}| + |s/c||h_{ij}| + s^2|h_{jj}|\right), \\ |\Delta h_{ij}|, |\Delta h_{ji}| &\leq \tilde{\gamma}_1 \left(|sc||h_{ii}| + 2s^2|h_{ij}| + |sc||h_{jj}|\right), \\ |\Delta h_{jj}| &\leq \tilde{\gamma}_1 \left(s^2|h_{ii}| + |s/c||h_{ij}| + c^2|h_{jj}|\right). \end{aligned}$$

Proof. For the duration of the proof let $Q_m := Q_m([i\ j],[i\ j])$. Writing $H_m = (h_{ij})$ and $\widehat{H}_{m+1} = (\widehat{h}_{ij})$ and using a standard result for matrix–vector multiplication [14, section 3.5], we have, for $k \neq i,j$,

$$\begin{aligned} \begin{bmatrix} \widehat{h}_{ik} \\ \widehat{h}_{jk} \end{bmatrix} &= fl\left(Q_m^T \begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix}\right), \\ &= \left(Q_m + \Delta Q_m\right)^T \begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix}, \quad |\Delta Q_m| \leq \tilde{\gamma}_1 |Q_m|, \\ &=: Q_m^T \left(\begin{bmatrix} h_{ik} \\ h_{jk} \end{bmatrix} + \begin{bmatrix} \Delta h_{ik} \\ \Delta h_{jk} \end{bmatrix}\right). \end{aligned}$$

Then

$$\begin{aligned} \begin{bmatrix} |\Delta h_{ik}| \\ |\Delta h_{jk}| \end{bmatrix} &\leq |Q_m||\Delta Q_m^T| \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix} \\ &\leq \tilde{\gamma}_1 |Q_m||Q_m^T| \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix} \\ &= \tilde{\gamma}_1 \begin{bmatrix} 1 & 2|sc| \\ 2|sc| & 1 \end{bmatrix} \begin{bmatrix} |h_{ik}| \\ |h_{jk}| \end{bmatrix}, \end{aligned}$$

which gives the first two bounds. We calculate the elements at the intersection of rows and columns i and j using

$$\widehat{h}_{ii} = fl(h_{ii} - h_{ij}t) = (1+\tilde{\theta}_1)h_{ii} - (1+\tilde{\theta}_1)h_{ij}t,$$
$$\widehat{h}_{jj} = fl(h_{jj} + h_{ij}t) = (1+\tilde{\theta}_1)h_{jj} + (1+\tilde{\theta}_1)h_{ij}t,$$

and by setting $\widehat{h}_{ij}$ and $\widehat{h}_{ji}$ to zero. The backward perturbations Δh_{ii}, Δh_{ij}, and Δh_{jj} satisfy

$$Q_m^T \left(\begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix} + \begin{bmatrix} \Delta h_{ii} & \Delta h_{ij} \\ \Delta h_{ij} & \Delta h_{jj} \end{bmatrix} \right) Q_m = \begin{bmatrix} \widehat{h}_{ii} & 0 \\ 0 & \widehat{h}_{jj} \end{bmatrix},$$

which can be expressed as

$$\begin{aligned} \begin{bmatrix} \Delta h_{ii} & \Delta h_{ij} \\ \Delta h_{ij} & \Delta h_{jj} \end{bmatrix} &= Q_m \begin{bmatrix} \widehat{h}_{ii} & 0 \\ 0 & \widehat{h}_{jj} \end{bmatrix} Q_m^T - \begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix} \\ &= \begin{bmatrix} c^2\widehat{h}_{ii} + s^2\widehat{h}_{jj} & -sc\widehat{h}_{ii} + sc\widehat{h}_{jj} \\ -sc\widehat{h}_{ii} + sc\widehat{h}_{jj} & s^2\widehat{h}_{ii} + c^2\widehat{h}_{jj} \end{bmatrix} - \begin{bmatrix} h_{ii} & h_{ij} \\ h_{ij} & h_{jj} \end{bmatrix}. \end{aligned}$$

Substituting in for $\widehat{h}_{ii}$ and $\widehat{h}_{jj}$ and taking absolute values we obtain the second group of inequalities. (Note that $\Delta h_{ij} = \Delta h_{ji} = 0$ if c and s are exact, so by bounding Δh_{ij} and Δh_{ji} in this way we are allowing for inexact c and s.) ∎

In the next lemma we show that in the first rotation of Jacobi's method in Algorithm 2.1 a factor D^{-1} can be scaled out of the backward error, leaving a term that we can bound. We make use of the identity

$$sc = \frac{h_{ij}}{\sqrt{4h_{ij}^2 + (h_{ii} - h_{jj})^2}}, \tag{3.1}$$

which comes from manipulating the equations defining a Jacobi rotation and solving for $sc = \frac{1}{2}\sin 2\theta$ in terms of $\tan 2\theta$. In this result, $A_0 \equiv L^{-1}\Pi^T A \Pi L^{-T}$ in (2.1).

LEMMA 3.4. *Given a symmetric A_0 and a positive diagonal matrix $D_0 = \operatorname{diag}(d_i^2)$, suppose we perform one step of Jacobi's method in the (i,j) plane on $H_0 = D_0^{-1}A_0D_0^{-1}$, obtaining $H_1 = Q_0^T H_0 Q_0$. Then*

$$\widehat{H}_1 = fl(Q_0^T \widehat{H}_0 Q_0) = Q_0^T D_0^{-1}(A_0 + \Delta A_0)D_0^{-1}Q_0, \tag{3.2}$$

where

$$\|\Delta A_0\|_2 \le \tilde{\gamma}_n(1+2\omega_0)\|A_0\|_2, \tag{3.3}$$

with

$$\omega_0 = |sc|\max(\rho, 1/\rho), \qquad \rho = d_i/d_j.$$

Proof. We start by forming the matrix $H_0 = (h_{ij})$. Since we are given the squared diagonal elements d_i^2 we have

$$\begin{aligned} \widehat{h}_{ij} &= fl\left(a_{ij}/\sqrt{d_i^2 d_j^2}\right) \\ &= (1+\theta_3)a_{ij}/(d_i d_j) = (1+\theta_3)h_{ij} \\ &=: \widehat{a}_{ij}/(d_i d_j). \end{aligned}$$

Thus these initial errors can be thrown onto A_0: $\widehat{H}_0 = D_0^{-1}(A_0 + \Delta_1)D_0^{-1}$, where $|\Delta_1| \le \gamma_3|A_0|$. The errors in applying one step of Jacobi's method to $\widehat{H}_0$ can be expressed as a backward perturbation ΔH_0 to $\widehat{H}_0$ using Lemma 3.3. The corresponding perturbation of $\widehat{A}_0 = A_0 + \Delta_1$ is $\Delta_2 = D_0 \Delta H_0 D_0$, so we simply scale the componentwise perturbation bounds of Lemma 3.3. We find

$$\left.\begin{array}{l} |(\Delta_2)_{ik}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{ik}| + 2|sc||\widehat{a}_{jk}|\rho\right) \\ |(\Delta_2)_{jk}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{jk}| + 2|sc||\widehat{a}_{ik}|/\rho\right) \end{array}\right\} \quad k \ne i, j,$$

$$|(\Delta_2)_{ii}| \le \tilde{\gamma}_1 \left(c^2|\widehat{a}_{ii}| + |s/c||\widehat{a}_{ij}|\rho + s^2|\widehat{a}_{jj}|\rho^2\right), \tag{3.4}$$

$$|(\Delta_2)_{ij,ji}| \le \tilde{\gamma}_1 \left(|sc||\widehat{a}_{ii}|/\rho + 2s^2|\widehat{a}_{ij}| + |sc||\widehat{a}_{jj}|\rho\right),$$

$$|(\Delta_2)_{jj}| \le \tilde{\gamma}_1 \left(s^2|\widehat{a}_{ii}|/\rho^2 + |s/c||\widehat{a}_{ij}|/\rho + c^2|\widehat{a}_{jj}|\right). \tag{3.5}$$

We now work to remove the potentially large ρ^2 and $1/\rho^2$ terms. We can rewrite (3.1) as

$$sc = \frac{\frac{a_{ij}}{d_i d_j}}{\sqrt{4\frac{a_{ij}^2}{d_i^2 d_j^2} + \left(\frac{a_{ii}}{d_i^2} - \frac{a_{jj}}{d_j^2}\right)^2}} = \frac{\rho a_{ij}}{\sqrt{\left(a_{ii} - \rho^2 a_{jj}\right)^2 + 4\rho^2 a_{ij}^2}}. \tag{3.6}$$

Further manipulation yields

$$|a_{jj}|\rho^2 \le |a_{ii}| + \sqrt{\frac{a_{ij}^2\rho^2}{(sc)^2} - 4a_{ij}^2\rho^2} = |a_{ii}| + \rho|a_{ij}|\sqrt{\frac{1}{(sc)^2} - 4}.$$

Therefore

$$s^2|a_{jj}|\rho^2 \le s^2|a_{ii}| + \rho|a_{ij}|\sqrt{t^2 - 4s^4}. \tag{3.7}$$

A similar manipulation of (3.1) (or a symmetry argument) gives

$$s^2|a_{ii}|/\rho^2 \le s^2|a_{jj}| + \frac{|a_{ij}|}{\rho}\sqrt{t^2 - 4s^4}. \tag{3.8}$$

Since $\widehat{a}_{ij} = a_{ij}(1 + \theta_3)$ there is no harm in replacing a_{ij} by $\widehat{a}_{ij}$ in (3.7) and (3.8). Since $\theta \in [-\pi/4, \pi/4]$ we have

$$\sqrt{t^2 - 4s^4} + |s/c| = 2|sc|, \tag{3.9}$$

and hence (3.4) and (3.5) may be bounded by

$$\left.\begin{array}{l} |(\Delta_2)_{ii}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{ii}| + 2|sc||\widehat{a}_{ij}|\rho\right), \\ |(\Delta_2)_{jj}| \le \tilde{\gamma}_1 \left(|\widehat{a}_{jj}| + 2|sc||\widehat{a}_{ij}|/\rho\right). \end{array}\right.$$

Setting $\Delta A = \Delta_1 + \Delta_2$ and using these componentwise bounds we obtain the overall bound given in (3.3). □

Lemma 3.4 shows that the Jacobi rotation results in a small backward perturbation to A_0 provided that ω_0 is of order 1. We see from (3.6) that in normal circumstances sc is proportional to $\min(\rho, 1/\rho)$, which keeps ω_0 small. However, in special situations ω_0 can be large, for example, when $|a_{ii} - \rho^2 a_{jj}| \ll \rho|a_{ij}|$ with ρ large, which requires that $|a_{jj}|$ be much smaller than $|a_{ij}|$ and B be ill conditioned.

By combining Lemma 3.4 with subsequent applications of Lemma 3.3 we find that after m steps of Jacobi's method on $H_0 = D_0^{-1}A_0D_0^{-1}$ we have

$$\widehat{H}_m = Q_{m-1}^T \dots Q_0^T (H_0 + \Delta_0) Q_0 \dots Q_{m-1},$$

where

$$\begin{aligned}\Delta_0 &= D_0^{-1}\Delta A_0 D_0^{-1} + \sum_{k=1}^{m-1} Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T \\ &= D_0^{-1}\left(\Delta A_0 + \sum_{k=1}^{m-1} D_0 Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T D_0\right) D_0^{-1}.\end{aligned}$$

The ΔH_k are bounded as in Lemma 3.3. We would like to bound the term in parentheses by a multiple of $u\|A_0\|_2$, but simply taking norms leads to an unsatisfactory $\kappa(D_0^2)$ factor. To obtain a better bound we introduce, purely for theoretical purposes, a scaling to $\widehat{H}_k$ at each stage of the iteration. For an arbitrary nonsingular diagonal D_k we write

$$\begin{aligned}\|D_0Q_0 \dots Q_{k-1}\Delta H_k Q_{k-1}^T \dots Q_0^T D_0\|_2 &= \|D_0Q_0 \dots Q_{k-1}D_k^{-1} \cdot D_k\Delta H_k D_k \\ &\qquad \cdot D_k^{-1}Q_{k-1}^T \dots Q_0^T D_0\|_2 \\ &\le \min_{D_k \text{ diag}} \left(\|D_0Q_0 \dots Q_{k-1}D_k^{-1}\|_2^2 \|D_k\Delta H_k D_k\|_2\right) \\ &= \min_{D_k \text{ diag}} \left(\|N_k^{-T}\|_2^2 \|D_k\Delta H_k D_k\|_2\right),\end{aligned}$$

where

$$N_k = D_0^{-1}Q_0 \dots Q_{k-1}D_k. \tag{3.10}$$

Define

$$A_k := N_k^T A_0 N_k = D_k H_k D_k. \tag{3.11}$$

By applying Lemma 3.4 to a rotation on H_k, we can see that

$$\|D_k\Delta H_k D_k\|_2 \le \tilde{\gamma}_n(1 + 2\omega_k)\|A_k\|_2, \tag{3.12}$$

where

$$\omega_k = |s_k c_k| \max(\rho_k, 1/\rho_k), \qquad \rho_k = d_i^{(k)}/d_j^{(k)},$$

with a subscript k denoting quantities on the kth step and where $D_k = \operatorname{diag}(d_i^{(k)})$. One way to proceed is to choose D_k to minimize $\kappa_2(M_{k-1})$, where

$$M_{k-1} = D_{k-1}^{-1}Q_{k-1}D_k. \tag{3.13}$$

Notice that

$$N_k = M_0 \dots M_{k-1}. \tag{3.14}$$

This idea is based on an algorithm of Anjos, Hammarling, and Paige [2] that avoids explicitly inverting any of the D_k and uses transformation matrices of the form in (3.13)

to diagonalize A while retaining the diagonal form of D_0. The algorithm computes the congruence transformations

$$A_{k+1} = M_k^T A_k M_k, \quad D_{k+1}^2 = M_k^T D_k^2 M_k,$$

where D_k is diagonal for all k and A_k tends to diagonal form as $k \to \infty$. The difference between our approach and that in [2] is that we form $H_0 = D_0^{-1} A_0 D_0^{-1}$ and use D_k in the analysis to obtain stronger error bounds, whereas in [2], in an effort to apply only well-conditioned similarity transformations, H_0 is never formed but M_k is computed and applied in the algorithm (and no error analysis is given in [2]).

Now we discuss the choice of D_k, drawing on analysis from [2]. Since Q_{k-1} is a rotation in the (i, j) plane, we choose D_k to be identical to D_{k-1} in all but the ith and jth diagonal entries. Thus M_{k-1} is the identity matrix except in the (i, j) plane, in which

$$M_{ij} = M([i\ j], [i\ j]) = \begin{bmatrix} d_i^{-1} & 0 \\ 0 & d_j^{-1} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \widetilde{d}_i & 0 \\ 0 & \widetilde{d}_j \end{bmatrix},$$

where we are writing

$$D_{k-1} = \operatorname{diag}(d_i), \quad D_k = \operatorname{diag}(\widetilde{d}_i).$$

We now choose D_k to minimize the 2-norm condition number $\kappa_2(M_{ij})$. It can be shown that for any 2×2 matrix, G, say,

$$\kappa_2(G) = \sigma_1(G)/\sigma_2(G) = \left(\phi^2 + \sqrt{\phi^4 - 4\delta^2}\right)/2\delta,$$

where $\phi = \|G\|_F$, $\delta = |\det(G)|$ and $\sigma_1(G) \geq \sigma_2(G)$ are the singular values of G. Using $\kappa_F(G) = \phi^2/\delta$, we obtain

$$\kappa_2(G) = \left(\kappa_F(G) + \sqrt{\kappa_F(G)^2 - 4}\right)/2,$$

so clearly $\kappa_2(G)$ has its minimum when $\kappa_F(G)$ does. Therefore it is only necessary to analyze $\kappa_F(M_{ij})$ in order to find the minimum of $\kappa_2(M_{ij})$. For M_{ij} we have

$$\begin{aligned} \phi^2 &= s^2 \left((\widetilde{d}_i/d_j)^2 + (\widetilde{d}_j/d_i)^2\right) + c^2 \left((\widetilde{d}_j/d_j)^2 + (\widetilde{d}_i/d_i)^2\right), \\ \delta &= \det(D_{k-1}^{-1}) \det(D_k) = (\widetilde{d}_i \widetilde{d}_j)/(d_i d_j). \end{aligned}$$

Setting $\xi = \widetilde{d}_i/\widetilde{d}_j$ we have

$$\kappa_F(M_{ij}) = \phi^2/\delta = \left(c^2(\rho^2 + \xi^2) + s^2(\rho^2\xi^2 + 1)\right)/(\rho\xi).$$

This is an equation with only one unknown, ξ. The minimum of $\kappa_F(M_{ij})$ over ξ occurs at

$$\xi_{\text{opt}}^2 = \left(s^2 + \rho^2 c^2\right) / \left(c^2 + \rho^2 s^2\right),$$

which gives the values

$$\begin{aligned} \kappa_F(M_{ij})_{\min} &= 2\sqrt{1 + s^2 c^2 \left(\rho - \rho^{-1}\right)^2}, \\ \kappa_2(M_{ij})_{\min} &= |sc(\rho - \rho^{-1})| + \sqrt{1 + s^2 c^2 \left(\rho - \rho^{-1}\right)^2}. \end{aligned} \tag{3.15}$$

Knowing the ratio $\widetilde{d}_i/\widetilde{d}_j$ that minimizes $\kappa_2(M_0)$, we now have to choose $\widetilde{d}_j$ and then set $\widetilde{d}_i = \widetilde{d}_j \xi_{\mathrm{opt}}$. We set $\|D_k\|_F = \|D_{k-1}\|_F$, or more simply,

$$d_i^2 + d_j^2 = \widetilde{d}_i^2 + \widetilde{d}_j^2 = \left(\xi_{\mathrm{opt}}^2 + 1\right) \widetilde{d}_j^2. \tag{3.16}$$

This yields the values

$$\begin{aligned} \widetilde{d}_i^2 &= c^2 d_i^2 + s^2 d_j^2, \\ \widetilde{d}_j^2 &= c^2 d_j^2 + s^2 d_i^2 \end{aligned} \tag{3.17}$$

and the matrix

$$M_{ij} = \begin{bmatrix} c\sqrt{c^2 + s^2/\rho^2} & s\sqrt{s^2 + c^2/\rho^2} \\ -s\sqrt{s^2 + c^2\rho^2} & c\sqrt{c^2 + s^2\rho^2} \end{bmatrix}. \tag{3.18}$$

Clearly,

$$\min(d_i^2, d_j^2) \le \widetilde{d}_k^2 \le \max(d_i^2, d_j^2), \qquad k = i, j. \tag{3.19}$$

We note for later reference that a direct calculation reveals

$$\|M_{ij}^{-1}\|_F = \sqrt{2}. \tag{3.20}$$

It is also interesting to note that M_{ij} has columns of equal 2-norm. This is not surprising in view of a result of van der Sluis [24], which states that scaling the columns of an $n \times n$ matrix to have equal 2-norms produces a matrix with 2-norm condition number within a factor $\sqrt{n}$ of the minimum over all column scalings.

To complete our analysis we need to bound $\|A_k\|_2$ and $\|N_i^{-1}\|_2$.

3.1. Growth of A_m. We now bound $\|A_m\|_2$, which appears in the bound (3.12). We consider the growth over one step from $A_m = (a_{ij})$ to $A_{m+1} = (\widetilde{a}_{ij}) = M_m^T A_m M_m$, as measured by $\phi_m = \max_{i,j} |\widetilde{a}_{ij}| / \max_{i,j} |a_{ij}|$. By rewriting (2.8) and (2.9) in terms of A_k, and using (3.11) and (3.17), we can show that

$$|\widetilde{a}_{ii}| \le c^2|a_{ii}| + s^2|a_{ii}|/\rho^2 + |a_{ij}| \left(\frac{|s^3|}{c\rho} + |sc|\rho \right), \tag{3.21}$$

$$|\widetilde{a}_{jj}| \le c^2|a_{jj}| + s^2|a_{jj}|\rho^2 + |a_{ij}| \left(\frac{|sc|}{\rho} + \frac{|s^3|}{c}\rho \right). \tag{3.22}$$

We would like to bound these two elements linearly in terms of $\max(\rho, 1/\rho)$ (recall that ρ can be greater than or less than 1). The troublesome terms in the bounds are $s^2|a_{jj}|\rho^2$ and $s^2|a_{ii}|/\rho^2$. Upon substitution of (3.7) and (3.8) in (3.21) and (3.22) we obtain bounds linear in ρ and $1/\rho$:

$$\begin{aligned} |\widetilde{a}_{ii}| &\le c^2|a_{ii}| + s^2|a_{jj}| + |a_{ij}| \left(\left(\sqrt{t^2 - 4s^4} + \frac{|s^3|}{c} \right) \frac{1}{\rho} + |sc|\rho \right), \\ |\widetilde{a}_{jj}| &\le c^2|a_{jj}| + s^2|a_{ii}| + |a_{ij}| \left(\left(\sqrt{t^2 - 4s^4} + \frac{|s^3|}{c} \right) \rho + \frac{|sc|}{\rho} \right). \end{aligned}$$

Using (3.9) we find that $\sqrt{t^2 - 4s^4} + |s^3|/|c| = |sc|$, and so

$$|\widetilde{a}_{ii}| \le c^2|a_{ii}| + s^2|a_{jj}| + |a_{ij}||sc| \left(\rho + 1/\rho\right), \tag{3.23}$$

$$|\widetilde{a}_{jj}| \le c^2|a_{jj}| + s^2|a_{ii}| + |a_{ij}||sc| \left(\rho + 1/\rho\right). \tag{3.24}$$

For the other affected elements in rows and columns i and j we have, for $k \neq i, j$,

$$\widetilde{a}_{ik} = \widetilde{a}_{ki} = a_{ik}c\sqrt{c^2 + s^2/\rho^2} - a_{jk}s\sqrt{s^2 + c^2\rho^2},$$
$$\widetilde{a}_{jk} = \widetilde{a}_{kj} = a_{ik}s\sqrt{s^2 + c^2/\rho^2} + a_{jk}c\sqrt{c^2 + s^2\rho^2}.$$

These elements can be bounded by

$$|\widetilde{a}_{ik}| \leq |a_{ik}|\left(c^2 + |sc|/\rho\right) + |a_{jk}|\left(s^2 + |sc|\rho\right), \tag{3.25}$$
$$|\widetilde{a}_{jk}| \leq |a_{ik}|\left(s^2 + |sc|/\rho\right) + |a_{jk}|\left(c^2 + |sc|\rho\right). \tag{3.26}$$

The bounds (3.23)–(3.26) can all be written in the form

$$|\widetilde{a}_{pq}| \leq \max_{r,s} |a_{rs}|\big(1 + |sc|(\rho + 1/\rho)\big),$$

and so the growth of A_m over one step is bounded by

$$\phi_m \leq 1 + |sc|(\rho + 1/\rho) \leq 1 + 2|sc| \max(\rho, 1/\rho) = 1 + 2\omega_m.$$

The overall growth bound is

$$\pi_m := \frac{\|A_m\|_2}{\|A_0\|_2} \leq \sqrt{n} \prod_{i=0}^{m-1} \phi_i. \tag{3.27}$$

3.2. Bounding $\|N_i^{-1}\|_2$. Our final task is to bound

$$\mu_i := \|N_i^{-1}\|_2 = \|D_i^{-1} Q_{i-1}^T \dots Q_0^T D_0\|_2$$

(see (3.10)). We describe two different bounds. In view of (3.19),

$$\|D_{i+1}^{-1}\|_2 \leq \|D_i^{-1}\|_2 \leq \cdots \leq \|D_0^{-1}\|_2.$$

Thus, since $D_0 = D$, where B has the Cholesky factorization (1.2),

$$\mu_i^2 \leq \kappa_2(D)^2 \leq \kappa_2(L)\kappa_2(B).$$

However, the point of our analysis is to avoid a $\kappa_2(B)$ term in the bounds. As an alternative way of bounding μ_i we note that, from (3.14),

$$N_i^{-1} = M_{i-1}^{-1} \dots M_0^{-1}.$$

For the row cyclic ordering in (2.7) the congruence transformations can be reordered into $2n-3$ groups of up to $\lceil n/2 \rceil$ disjoint transformations $M_{j+1}, \dots, M_{j+p}$ such that, using (3.20),

$$\|M_{j+p}^{-1} \dots M_{j+1}^{-1}\|_2 \leq \sqrt{2}.$$

For example, a sweep of a 6×6 matrix can be divided into 9 groups of disjoint rotations:

$$\begin{bmatrix} - & 1 & 2 & 3 & 4 & 5 \\ & - & 3 & 4 & 5 & 6 \\ & & - & 5 & 6 & 7 \\ & & & - & 7 & 8 \\ & & & & - & 9 \\ & & & & & - \end{bmatrix}.$$

Here, an integer k in position (i,j) denotes that the (i,j) element is eliminated on the kth step by a rotation in the (i,j) plane, and all rotations on the kth step are disjoint. Hence we can bound μ_i by

$$\mu_i \le (\sqrt{2})^{2n-3} = 2^{n-3/2}.$$

Although exponential in n, this bound is independent of $\kappa_2(B)$.

3.3. Summary. Our backward error analysis shows that, upon convergence after m Jacobi rotations, Algorithm 2.1 has computed a diagonal Λ such that

$$X^T(A+\Delta A)X = \Lambda, \quad X^T(B+\Delta B)X = I \tag{3.28}$$

for some nonsingular X, where

$$\|\Delta A\|_2 \le \tilde{\gamma}_{n^2}\|A\|_2\left(\kappa_2(L)^2 + \sum_{k=0}^{m-1}\mu_k^2\big(1+2\omega_k\big)\pi_k\right), \tag{3.29a}$$

$$\|\Delta B\|_2 \le \tilde{\gamma}_{n^2}\|B\|_2. \tag{3.29b}$$

The term involving $\kappa_2(L)$ takes account of errors in the first stage of Algorithm 2.1 and follows from standard error analysis [14, chapter 10] of Cholesky factorization and the solution of triangular systems. Because of the complete pivoting, $\kappa(L)$ is bounded as in (1.6), and in practice it is usually small. Even when $\kappa(L)$ is large, its full effect tends not to be felt on the backward error, since triangular systems are typically solved to higher accuracy than the bounds suggest [14, chapter 8].

We do not have a bound better than exponential in n for the term μ_i^2, but this term has been less than 10 in virtually all our numerical tests. We showed in section 3.1 that the growth factor $\pi_k = \|A_k\|_2/\|A_0\|_2$ in (3.27) is certainly bounded by $\pi_k \le \sqrt{n}\prod_{i=0}^{k-1}(1+2\omega_i)$. The term

$$\omega_k = |s_k c_k|\max(\rho_k, 1/\rho_k) \le |s_k c_k|\kappa_2(D) \le |s_k c_k|\kappa_2(L)\kappa_2(B)^{1/2} \tag{3.30}$$

is the most important quantity in our analysis. A large value of ω_k, for some k, is the main indicator of instability in Algorithm 2.1.

We stress that our error bounds do not depend on the ordering (1.5), as should be expected since the Jacobi method is insensitive to the ordering of the diagonal of D. The purpose of pivoting in the Cholesky factorization is to keep L well conditioned and thereby concentrate any ill conditioning of B into D.

The conclusion from the error analysis is that Algorithm 2.1 has much better stability properties than the bound (1.7) suggests. When $\kappa_2(B)$ is large it is usually the case that small values of $|s_k c_k|$ cancel any large values of $\max(\rho_k, 1/\rho_k)$ (see the discussion following Lemma 3.4) and that π_k is also small, with a resulting small backward error bound.

For the particular version of the Cholesky–QR method in which the initial tridiagonalization of the QR algorithm is performed using Givens rotations, Davies [5] uses suitable modifications of the analysis presented here to derive analogues of (3.28) and (3.29) in which the terms $1+2\omega_k$ and π_k in (3.29) are squared (the definitions of w_k and π_k are unchanged, but of course the underlying rotations are different). Unfortunately, Householder transformations rather than Givens rotations are almost always used for the tridiagonalization and our error analysis is specific to rotations; therefore (1.7) remains the best error bound for the practically used Cholesky–QR method.

4. Iterative refinement. The relative normwise backward error of an approximate eigenpair $(\widetilde{x}, \widetilde{\lambda})$ of (1.1) is defined by

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \min\{\epsilon : (A + \Delta A)\widetilde{x} = \widetilde{\lambda}(B + \Delta B)\widetilde{x}, \quad \|\Delta A\| \le \epsilon\|A\|, \quad \|\Delta B\| \le \epsilon\|B\|\}. \tag{4.1}$$

To evaluate the backward error we can use the explicit expression [11], [13]

$$\eta(\widetilde{x}, \widetilde{\lambda}) = \frac{\|r\|}{(|\widetilde{\lambda}|\,\|B\| + \|A\|)\|\widetilde{x}\|}, \tag{4.2}$$

where $r = \widetilde{\lambda}B\widetilde{x} - A\widetilde{x}$ is the residual. For symmetric A and B, we denote by $\eta^S(\widetilde{x}, \widetilde{\lambda})$ the backward error (4.1) with the additional constraint that the perturbations ΔA and ΔB are symmetric. Clearly $\eta^S(\widetilde{x}, \widetilde{\lambda}) \ge \eta(\widetilde{x}, \widetilde{\lambda})$. However, Higham and Higham [13] show that when $\widetilde{\lambda}$ is real, $\eta^S(\widetilde{x}, \widetilde{\lambda}) = \eta(\widetilde{x}, \widetilde{\lambda})$ for the 2-norm. Hence, for the symmetric definite generalized eigenproblem it is appropriate to use the general definition (4.1) and the formula (4.2).

The idea of using iterative refinement to improve numerical stability has been investigated for linear systems by several authors; see [14, chapter 11] for a survey and [15] for the most recent results. Iterative refinement has previously been used with residuals computed in extended precision to improve the accuracy of approximate solutions to the standard eigenproblem [8], [9], [22]. Tisseur [23] shows how iterative refinement can be used in fixed or extended precision to improve the forward and backward errors of approximate solutions to the generalized eigenvalue problem (GEP). She writes the GEP as

$$Ax = \lambda Bx, \qquad e_s^T x = 1 \text{ (for some fixed } s)$$

and applies Newton's method to the equivalent nonlinear equation problem

$$F\left(\begin{bmatrix} x \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} (A - \lambda B)x \\ e_s^T x - 1 \end{bmatrix} : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}.$$

This requires solving linear systems whose coefficient matrices are the Jacobian

$$J\left(\begin{bmatrix} x \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} A - \lambda B & -Bx \\ e_s^T & 0 \end{bmatrix}.$$

We use this technique with residuals computed in *fixed precision* to improve the backward errors of eigenpairs computed by Algorithm 2.1. We very briefly summarize the convergence results and two implementations of iterative refinement; full details may be found in [23].

If J is not too ill conditioned, the linear system solver is not too unstable, and the starting vector is sufficiently close to an eigenpair (x_*, λ_*), then iterative refinement by Newton's method in floating point arithmetic with residuals computed in fixed precision yields a refined eigenpair $(\widehat{x}, \widehat{\lambda})$ with backward error in the ∞-norm bounded by [23, Corollary 3.5]

$$\eta_\infty(\widehat{x}, \widehat{\lambda}) \le \tilde{\gamma}_n + u(3 + |\lambda|)\max\left(\frac{\|A\|_\infty}{\|B\|_\infty}, \frac{\|B\|_\infty}{\|A\|_\infty}\right). \tag{4.3}$$

This backward error bound is small if λ is of order 1 and the problem is well balanced, that is, $\|A\|_\infty \approx \|B\|_\infty$. If the problem is not well balanced, we can change the GEP

to make it so. We can scale the GEP to $(\alpha A)x = (\alpha\lambda)Bx$, where $\alpha = \|B\|_\infty/\|A\|_\infty$ and the backward error now depends on the size of $\overline{\lambda} = \alpha\lambda$. If $|\overline{\lambda}| \leq 1$, a small backward error is ensured, while for $|\overline{\lambda}| \geq 1$ we can consider the problem $Bx = \overline{\mu}\overline{A}x$, for which $|\overline{\mu}| \leq 1$. Practical experience shows that it is not necessary to scale or to reverse the problem—a backward error of order u is obtained as long as the starting vector is good enough for Newton's method to converge.

The following algorithm can be derived after some manipulation of the Newton equations [23].

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
  $r = \lambda Bx - Ax$
  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. By taking advantage of the eigendecomposition computed by Algorithm 2.1, the cost per iteration can be reduced to $O(n^2)$ flops [23].

ALGORITHM 4.2. Given A, B, X, and Λ such that $X^TAX = \Lambda$ and $X^TBX = I$, 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
  $r = \lambda Bx - Ax$
  $D_\lambda = \Lambda - \lambda I$
  $d = -Bx - c_{\lambda s}$, where $c_{\lambda s}$ is the sth column of $A - \lambda B$
  $v = X^Td$; $f = X^Te_s$
  Compute Givens rotations J_k in the $(k, k+1)$ plane, such that
    $Q_1^Tv := J_1^T \ldots J_{n-1}^Tv = \|v\|_2 e_1$
  Compute orthogonal Q_2 such that
    $T = Q_2^TQ_1^T(D_\lambda + vf^T)$ is upper triangular
  $z = Q_2^TQ_1^TX^Tr$
  Solve $Tw = z$ for w
  $\delta = Xw$
  $\lambda = \lambda + \delta_s$; $\delta_s = 0$
  $x = x + \delta$
end

The computed $\widehat{X}$ from Algorithm 2.1 does not necessarily give a backward stable diagonalization of A and B. However, Tisseur [23] shows that instability in the solver does not affect the overall limiting accuracy and limiting backward error (4.3) when iterative refinement converges, although of course it may inhibit convergence. The price to be paid for the greater efficiency of Algorithm 4.2 over Algorithm 4.1 is less frequent and less rapid convergence.

5. Numerical results. In this section we give several examples to illustrate the behavior of Algorithm 2.1 and the sharpness of our backward error bounds, to show how the algorithm compares with the Cholesky–QR method, to show the need for pivoting in the Cholesky–QR method, and to show the benefits of iterative refinement.

TABLE 5.1
Terms from error analysis and backward error for Example 1.

ϵ	$\kappa_2(B)$	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
10^{-1}	10^{7}	7.98e-1	3.33e0	3.12e0	1.31e-16
10^{-2}	10^{14}	1.90e0	4.38e0	7.02e0	5.35e-17
10^{-3}	10^{21}	2.38e0	4.67e0	1.04e1	3.50e-17

All our experiments were carried out in MATLAB 6, in which matrix computations are based on LAPACK; the unit roundoff is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. (Our implementation of the Cholesky–QR method uses the MATLAB/LAPACK implementation of the QR algorithm and so employs Householder tridiagonalization.) In Algorithms 4.1 and 4.2 convergence was declared when $\eta_\infty(\widehat{x}, \widehat{\lambda}) \leq u$.

Example 1. Our first example illustrates how our backward error bounds can correctly predict perfect backward stability of Algorithm 2.1 despite large values of $\kappa_2(B)$. We take $A = H - I \in \mathbb{R}^{n \times n}$, where H is the Hilbert matrix, and $B = \mathrm{diag}(1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1})$. For $n = 8$ and $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$, Table 5.1 shows the values of the terms appearing in the error analysis along with the maximum backward error over all the computed eigenpairs. The Cholesky–QR method is also stable on this example.

In a variation of this example we took $A = H$ and $B = \mathrm{diag}(\epsilon^{n-1}, \ldots, \epsilon, 1)$, with $n = 8$ and $\epsilon = 10^{-2}$. The computed eigenvalues from the Cholesky–Jacobi method and the Cholesky–QR method with pivoting both range from 10^{-9} to 10^{14} and the maximum backward error over all the computed eigenpairs is of order u. However, the Cholesky–QR method without pivoting produces two negative eigenvalues of order 10^{-2}, even though the exact eigenvalues are clearly positive, and the maximum backward error is of order 10^{-3}.

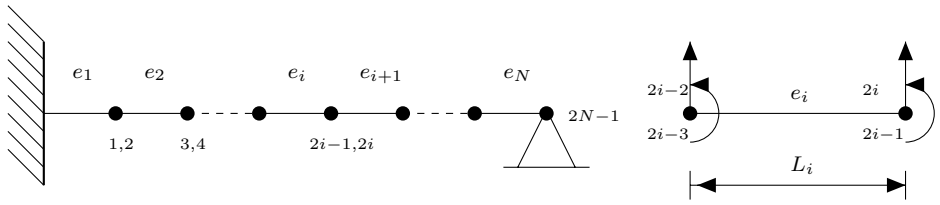
Example 2. This example is a structural engineering problem that again illustrates independence of our backward error bounds on $\kappa_2(B)$. We consider a cantilever beam as shown in Figure 5.1(a). We assume that the cantilever is rigid in its axial direction and that all the deformations are small. The boundary conditions are full-fixity at the base and zero translational displacement at the cantilever end. We also assume that the material properties and cross sections vary along the length of the beam. The equation of motion for the natural vibrations has the form

$$M\ddot{v} + Kv = 0,$$

where M denotes the symmetric positive definite mass inertia matrix and K the symmetric positive definite stiffness matrix. The finite element method leads to the generalized eigenvalue problem

$$K\phi = \lambda M\phi. \tag{5.1}$$

The cantilever is modeled with N finite elements. Each element has 4 degrees of freedom, namely, the two beam-end lateral displacements and the two beam-end rotations as shown in Figure 5.1(b). The length of the ith finite element e_i is taken to be L_i and its flexural characteristic to be $(EI)_i$, where E is the modulus of elasticity and I the moment of inertia. The global degrees of freedom are numbered as shown in Figure 5.1(a). If cubic Hermite interpolation polynomials are used to describe



(a) Geometry of supported cantilever beam. (b) Beam finite element.

FIG. 5.1. *Single span cantilever beam with supported end point.*

TABLE 5.2
Result for two instances of the cantilever beam problem.

$\kappa_2(M) = 3.9 \times 10^{10}$, $\kappa_2(L) = 1.8$

	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
Cholesky–Jacobi	4.58e0	8.3e0	1.63e0	5.18e-17
Cholesky–QR (no pivoting)				5.10e-17
Cholesky–QR (with pivoting)				7.48e-17

$\kappa_2(M) = 6.7 \times 10^{6}$, $\kappa_2(L) = 2.2$

	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$	$\max \eta_2(\widehat{x}, \widehat{\lambda})$
Cholesky–Jacobi	3.86e0	4.18e0	2.45e0	1.77e-16
Cholesky–QR (no pivoting)				1.23e-13
Cholesky–QR (with pivoting)				1.21e-16

displacement along the beam element, then the beam element stiffness matrix is [17]

$$K_i = \frac{2(EI)_i}{L_i^3} \begin{bmatrix} 6 & 3L_i & -6 & 3L_i \\ 3L_i & 2L_i^2 & -3L_i & L_i^2 \\ -6 & -3L_i & 6 & -3L_i \\ 3L_i & L_i^2 & -3L_i & 2L_i^2 \end{bmatrix}$$

and the beam element consistent mass matrix is

$$M_i = \frac{\overline{m}_i L_i}{420} \begin{bmatrix} 156 & 22L_i & 54 & -13L_i \\ 22L_i & 4L_i^2 & 13L_i & -3L_i^2 \\ 54 & 13L_i & 156 & -22L_i \\ -13L_i & -3L_i^2 & -22L_i & 4L_i^2 \end{bmatrix},$$

where $\overline{m}_i$ is the average mass per unit length for the ith beam. The global stiffness and mass inertia matrices are obtained by assembling the K_i and M_i, $i = 1\colon N$.

For our example, we chose $N = 5$ finite elements leading to 9 degrees of freedom and we varied the parameters e_i, L_i, $(EI)_i$, and $\overline{m}_i$, sometimes applying direct search to maximize the backward error over these variables. The backward errors for Algorithm 2.1 and the Cholesky–QR method with pivoting were always of order u, with our backward error bounds for Algorithm 2.1 also of order u. Table 5.2 shows results for two sets of parameters. The second set of results shows again that pivoting can be needed for stability of the Cholesky–QR method.

Example 3. This is an example where Algorithm 2.1 is unstable and there is only one large value of ω_k. With $n = 10$, we take $A \in \mathbb{R}^{n\times n}$ to be a random symmetric

TABLE 5.3
Iterative refinement of eigenpairs of Example 4. *For the entry marked* †, *convergence was not to the eigenvalue indicated in the leftmost column.*

	Before refinement		After refinement					
			Algorithm 4.1			Algorithm 4.2		
λ	$\eta_\infty(\widetilde{x},\widetilde{\lambda})$	$e(\widetilde{\lambda})$	$\eta_\infty(\widehat{x},\widehat{\lambda})$	$e(\widehat{\lambda})$	it	$\eta_\infty(\widehat{x},\widehat{\lambda})$	$e(\widehat{\lambda})$	it
$\epsilon = 2^{-6} \approx 1.6 \times 10^{-2}$								
1.4e0	4e-7	9e-6	5e-17	3e-16	2	7e-17	1e-15	2
−4.6e1	2e-8	6e-8	7e-18	2e-16	2	6e-18	2e-16	2
−8.4e3	2e-11	1e-9	5e-20	0	1	5e-20	0	2
$\epsilon = 2^{-8} \approx 3.9 \times 10^{-3}$								
1.4e0	2e-3	4e-2	4e-17	2e-16	3	4e-17	2e-16	12
−1.8e2	1e-5	3e-4	2e-17	2e-16	2	3e-17	8e-16	9
−1.4e4	4e-9	3e-6	5e-21	2e-16	2	2e-15	1e-12	∗
$\epsilon = 2^{-12} \approx 2.4 \times 10^{-4}$								
1.4e0	3e-3	1e0	4e-18	$0^\dagger$	5	1e-2	1e0	∗
−3.0e3	6e-4	8e-1	1e-22	0	5	3e-3	8e-1	∗
−3.5e7	4e-5	1e-1	2e-17	4e-16	3	2e-5	1e-1	∗

matrix and $B = I_n$ and replace the (n,n) entries of each matrix by 10^{-24}. Jacobi rotations not involving the nth plane have $\rho = 1$, and therefore ω_k is small. However, when we first apply a Jacobi rotation in the $(1,n)$ plane we see that $\rho = 10^{12}$ and

$$a_{11} - \rho^2 a_{nn} = a_{11} - 1 \ll \rho a_{1n} = 10^{12} a_{1n},$$

and therefore, from (3.6), $sc \approx 1/2$ and $\omega_k \approx 5 \times 10^{11}$. Note that this is an example where (3.30) is sharp. This is the only ill-conditioned M_k transformation as, using our scaling strategy, we set $\widetilde{d}_n^2 = c^2 d_n^2 + s^2 d_1^2 = O(1)$ in (3.17), and afterwards ρ is always approximately 1 for all subsequent rotations. The other key terms from the error bounds are $\max_k \pi_k = 8.4 \times 10^{11}$ and $\max_k \mu_k^2 = 2.0$. The computed eigenvalues consist of a group of 8 of order 1, all with backward errors of order 10^{-5} and two eigenvalues of order 10^{12}, with backward errors of order u. Applying Algorithm 4.1 to the eigenvalues with large backward errors we found that backward errors of order u were produced within 3–7 iterations; Algorithm 4.2 did not converge for any of the eigenvalues. The Cholesky–QR method was stable in this example.

Example 4. This example is one of a form suggested by G. W. Stewart that causes difficulties for Algorithm 2.1, and we use it to compare Algorithms 4.1 and 4.2. The matrices are

$$\operatorname{diag}(A) = d, \quad a_{ij} = \min(i,j) \text{ for } i \neq j, \quad B = \operatorname{diag}(d), \quad d = [1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1}]$$

with $0 < \epsilon < 1$. We take $n = 8$ with three choices of ϵ and concentrate on the three eigenvalues of smallest absolute value. We report in Table 5.3 the backward error $\eta_\infty(\widehat{x},\widehat{\lambda})$ of the computed eigenpair and the forward error

$$e(\widehat{\lambda}) = \frac{|\lambda - \widehat{\lambda}|}{|\lambda|}$$

of the computed eigenvalue, where the exact λ is obtained using MATLAB's Symbolic Math Toolbox; these statistics are given both before and after refinement, together

TABLE 5.4
Terms from error analysis for Example 4.

ϵ	$\kappa_2(B)$	$\max \omega_k$	$\max \mu_k^2$	$\max \pi_k$
2^{-6}	4e12	1.3e5	7.9	1.1e10
2^{-8}	7e16	1.7e7	8.0	8.8e13
2^{-12}	2e25	2.8e11	8.0	5.7e21

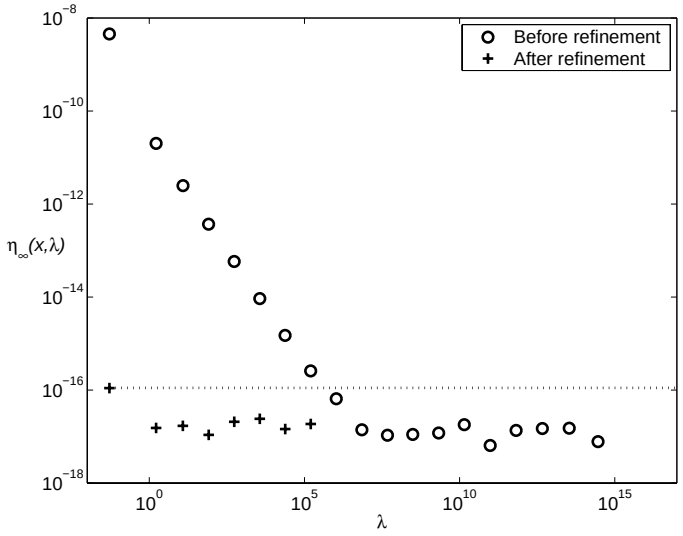


FIG. 5.2. *Backward errors for Cholesky–QR method before and after iterative refinement for Kahan matrix example (Example* 5*). Dotted line denotes unit roundoff level.*

with the number of iterations required by Algorithms 4.1 and 4.2, where "$*$" denotes no convergence after 50 iterations and in this case the quantities from the 50th iteration are shown. Table 5.4 shows the size of the terms appearing in the error bounds of section 3.3. The observed instability corresponds to large ω_k and π_k, but μ_k^2 is small, as is usually the case. We see that, as expected from the theory [23], refining with the unstable linear system solver produces the same limiting backward error as when the stable solver is used, but that it can produce slower convergence and is less likely to converge at all, as we saw also in Example 3. Iterative refinement also improves the forward error e. As one entry in the table shows, it is possible for iterative refinement to converge to a different eigenpair than expected when the original approximate eigenpair is sufficiently poor. The Cholesky–QR method performs stably on this example.

Example 5. The next example illustrates how ill condition of L can cause instability. Here, $n = 20$, $A = I$, and $B = R^T R$, where R is a Kahan matrix, and $\kappa_2(B) \approx 1/u$, $\kappa_2(L) \approx 3 \times 10^4$. Figure 5.2 plots the eigenvalues on the x-axis versus the ∞-norm backward errors of the eigenpairs on the y-axis, for eigenpairs both before and after refinement. At most one step of iterative refinement was required. The Cholesky–QR method was used, with Algorithm 4.2; Algorithms 2.1 and 4.1 give very similar results. The quantities in the error bounds for Algorithm 2.1 are $\max \omega_k = 0.6$, $\max \mu_k^2 = 315$, $\max \pi_k = 1.8$. As expected, it is the small eigenvalues that have large backward errors initially.

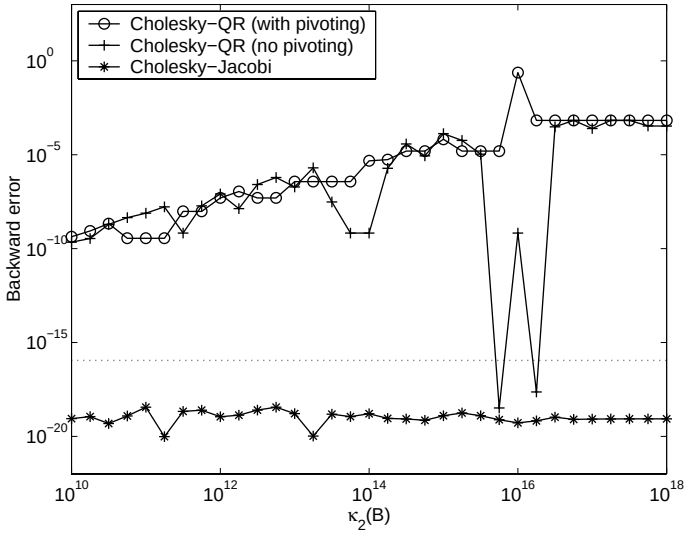


FIG. 5.3. *Behavior of the backward error for eigenvalue of smallest modulus of problem* (5.2) *with* $\alpha = 1$, $\delta = 10^{-3}$. *Dotted line denotes unit roundoff level.*

Example 6. Our penultimate example is adapted from a problem used by Fix and Heiberger [10] and shows that it is possible for the Cholesky–Jacobi method to be stable when the Cholesky–QR method both with and without pivoting is unstable. Let

$$A = \begin{bmatrix} 1 & \alpha & 0 & \delta \\ \alpha & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ \delta & 0 & 0 & \epsilon \end{bmatrix}, \quad B = \operatorname{diag}(\epsilon, 1, \epsilon, 1), \qquad \alpha, \delta > 0, \quad 0 < \epsilon < 1. \tag{5.2}$$

We solved the problem for $\alpha = 1$, $\delta = 10^{-3}$ and a range of ϵ from 10^{-10} to 10^{-18} by Algorithm 2.1 and the Cholesky–QR method with pivoting. Figure 5.3 plots the condition number of B against the backward error $\eta_2(\widehat{x}_{\min}, \widehat{\lambda}_{\min})$ for the eigenvalue $\widehat{\lambda}_{\min}$ of minimal modulus. The Cholesky–QR method performs unstably for most of the matrices B in the figure (strangely, producing generally better results without pivoting), while Algorithm 2.1 displays excellent stability. For Algorithm 2.1 we have $\max_k \omega_k = \max_k \pi_k = 1.0$ and $\max_k \mu_k^2 = 1.71$, so our error bounds predict the small backward errors.

Example 7. Our final example uses a class of random test problems suggested by Chandrasekaran [3]. They have the form

$$A = R + (10^{-8} n \lambda_n - \lambda_{[n/2]})I, \qquad B = S + (|\lambda_1| + 10^{-8} n \max(\lambda_1, \lambda_n))I,$$

where R and S are random matrices from the normal (0,1) distribution and $\lambda_1 \le \cdots \le \lambda_n$ are the eigenvalues of R (for A) or S (for B). With $5 \le n \le 100$ the backward errors of the eigenpairs produced by the Cholesky–Jacobi method and the Cholesky–QR method with and without pivoting were almost always less than nu, with a maximum value of 10^{-13} occurring for the Cholesky–QR method without pivoting for $n = 60$. Iterative refinement by Algorithms 4.1 and 4.2 reduced the backward error to u in at most three iterations, with only one iteration being required in over 95 percent of the cases. For Algorithm 2.1 we have $\max_k \omega_k = \max_k \mu_k^2 = 4$ and $\max_k \pi_k = 56$.

6. Conclusions. We have shown that the Cholesky–Jacobi method has better numerical stability properties than the standard backward error bound (1.7) suggests. For problems with an ill-conditioned B, the method can be, and often is, perfectly stable, and numerical experiments show that our bounds predict the stability well. The method is of practical use: it is easy to code, as Algorithm 2.1 shows, and the Jacobi method is particularly attractive in a parallel computing environment.

In practice, the Cholesky–QR method appears to perform as well as the Cholesky–Jacobi method, provided that complete pivoting is used in the Cholesky factorization. As we noted in section 1 this can, to some extent, be explained by the QR method's good performance on graded matrices. However, except for a rarely used variant employing Givens tridiagonalization, the best backward error bound for the Cholesky–QR method continues to contain a factor $\kappa_2(B)$. It is an important open problem to derive a sharper bound.

Instability of the Cholesky methods can be cured by iterative refinement, provided it is not too severe, as we have illustrated. Drawbacks are that refinement is expensive if applied to more than just a few eigenpairs, and practically verifiable conditions that guarantee convergence to the desired eigenpair are not available, though the method is surprisingly effective in practice.

The Cholesky–QR method (without pivoting) is the standard method for solving the symmetric definite generalized eigenproblem in LAPACK, MATLAB 6, and the NAG Library, all of which aim to provide exclusively backward stable algorithms. It is clearly desirable for these implementations to incorporate pivoting in the Cholesky factorization, in order to enhance the reliability, and to incorporate the option of iterative refinement of selected eigenpairs, to ameliorate those instances, which are rarer than we can explain, where the Cholesky–QR method behaves unstably.

Acknowledgment. We thank Sven Hammarling for many helpful discussions on this work.

REFERENCES

[1] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[2] M. F. Anjos, S. J. Hammarling, and C. C. Paige, *Solving the Generalized Symmetric Eigenvalue Problem*, manuscript, 1992.

[3] S. Chandrasekaran, *An efficient and stable algorithm for the symmetric-definite generalized eigenvalue problem*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1202–1228.

[4] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, Pacific Grove, CA, 1995.

[5] P. I. Davies, *Solving the Symmetric Definite Generalized Eigenvalue Problem*, Ph.D. thesis, University of Manchester, Manchester, England, 2000.

[6] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[7] J. W. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[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, C. B. Moler, and J. H. Wilkinson, *Improving the accuracy of computed eigenvalues and eigenvectors*, SIAM J. Numer. Anal., 20 (1983), pp. 23–45.

[10] G. Fix and R. Heiberger, *An algorithm for the ill-conditioned generalized eigenvalue problem*, SIAM J. Numer. Anal., 9 (1972), pp. 78–88.

[11] V. Frayssé and V. Toumazou, *A note on the normwise perturbation theory for the regular generalized eigenproblem*, Numer. Linear Algebra Appl., 5 (1998), pp. 1–10.

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

[13] 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.
[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[15] N. J. Higham, *Iterative refinement for linear systems and LAPACK*, IMA J. Numer. Anal., 17 (1997), pp. 495–509.
[16] W. Kerner, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85.
[17] L. Meirovitch, *Elements of Vibration Analysis*, 2nd ed., McGraw-Hill, New York, 1986.
[18] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.
[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1997.
[20] G. Peters and J. H. Wilkinson, *$Ax = \lambda Bx$ and the generalized eigenproblem*, SIAM J. Numer. Anal., 7 (1970), pp. 479–492.
[21] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[22] 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.
[23] F. Tisseur, *Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1038–1057.
[24] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.
[26] J. H. Wilkinson, *Error analysis revisited*, Bull. Inst. Math. Appl., 22 (1986), pp. 192–200.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 494–510
© 2001 Society for Industrial and Applied Mathematics

A STABILIZED SUPERFAST SOLVER FOR NONSYMMETRIC TOEPLITZ SYSTEMS*

MARC VAN BAREL†, GEORG HEINIG‡, AND PETER KRAVANJA†

Abstract. We present a stabilized superfast solver for nonsymmetric Toeplitz systems $Tx = b$. An explicit formula for T^{-1} is expressed in such a way that the matrix-vector product $T^{-1}b$ can be calculated via FFTs and Hadamard products. This inversion formula involves certain polynomials that can be computed by solving two linearized rational interpolation problems on the unit circle. The heart of our Toeplitz solver is a superfast algorithm to solve these interpolation problems. To stabilize the algorithm, i.e., to improve the accuracy, several techniques are used: pivoting, iterative improvement, downdating, and giving "difficult" interpolation points an adequate treatment. We have implemented our algorithm in Fortran 90. Numerical examples illustrate the effectiveness of our approach.

Key words. nonsymmetric Toeplitz systems, stabilized superfast algorithm, inversion formula, rational interpolation, pivoting, iterative improvement, downdating

AMS subject classifications. Primary, 65F05; Secondary, 47B35, 15A09

PII. S0895479899362302

1. Introduction.

The subject of this paper is linear systems of equations

$$T_n x = b \tag{1.1}$$

with a nonsingular, in general nonsymmetric, $n \times n$ Toeplitz coefficient matrix $T := T_n := [a_{k-l}]_{k,l=0}^{n-1}$ and a right-hand side vector $b \in \mathbb{C}^n$. The aim of the paper is to present a new solution algorithm that has (generically) complexity $\mathcal{O}(n \log^2 n)$ and that avoids unstable behavior by introducing several stabilizing techniques.

Since Toeplitz systems of equations appear in many applications, they are the subject of a large number of publications. During the last decades many algorithms have been developed that exploit the Toeplitz structure of the coefficient matrix. There are two types of direct fast solvers that require $\mathcal{O}(n^2)$ operations: Levinson-type and Schur-type algorithms. For more references and information concerning the history of these algorithms, we refer the reader to [36].

The flow of these classical fast methods is determined by the exact singularity of the leading principal submatrices of T. The fast methods compute the solutions corresponding to successive nonsingular leading principal submatrices (sections). However, in finite-precision arithmetic not only singular but also ill-conditioned sections should be avoided. In the case of a positive definite matrix, it can be guaranteed that the

*Received by the editors October 14, 1999; accepted for publication (in revised form) by L. Eldén March 21, 2001; published electronically October 11, 2001. This research was partially supported by the Fund for Scientific Research-Flanders (FWO-V), project "Orthogonal Systems and Their Applications," grant G.0278.97 from the K.U. Leuven (Bijzonder Onderzoeksfonds), project "SLAP: Structured Linear Algebra Package," grant OT/00/16 from the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, and Kuwait University Research Project SM-190. The scientific responsibility rests with the authors.

http://www.siam.org/journals/simax/23-2/36230.html

†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium (Marc.VanBarel@cs.kuleuven.ac.be, Peter.Kravanja@na-net.ornl.gov).

‡Department of Mathematics, Kuwait University, POB 5969, Safat 13060, Kuwait (georg@mcs.sci.kuniv.edu.kw).

principal submatrices are well-conditioned. Moreover, it was proved in [5] that the Schur (Bareiss) algorithm is weakly stable in this case. Weak stability of the Levinson algorithm for a certain class of positive definite Toeplitz matrices was already proven in [13]. In the case of an indefinite and nonsymmetric matrix, simple examples show that both the Levinson and the Schur algorithms can be highly unstable. One idea for overcoming these instabilities is to consider the normal equation $T_n^H T_n x = T_n^T b$ or some related augmented systems. The matrix $T_n^H T_n$ and its augmented relatives are positive definite, i.e., all principal submatrices are nonsingular, and, moreover, they have a displacement structure. Stable generalized Schur algorithms have been designed for these matrices by Sayed and Chandrasekaran (see [36, Chapter 3] and the references therein, such as [12]).

Another idea for overcoming unstable behavior of the classical algorithms is to "look ahead" from one well-conditioned section to the next one and jump over the ill-conditioned sections that lie in between. For Toeplitz systems several look-ahead algorithms were designed in [10, 11, 15, 16, 18, 23, 25, 26, 27, 28, 35]. For Hankel systems we refer to [6, 9, 17]. Several high-performance algorithms for Toeplitz matrices are described in [19], including two look-ahead Schur algorithms for symmetric indefinite block Toeplitz matrices. See also [51]. In [45] a look-ahead block Schur algorithm for Toeplitz-like matrices was presented. However, reliable look-ahead strategies are difficult to design and, moreover, the resulting algorithms may have complexity $O(n^3)$, i.e., they may not be fast.

A third approach for dealing with nonsymmetric Toeplitz systems, which is, in principle, also the approach used in this paper, was first proposed in [29] and further developed in [19, 21, 24, 30, 32, 33, 37, 38, 48] and other papers. A survey of the matter is given in [44]. The idea is to transform the Toeplitz (or Hankel or Toeplitz-plus-Hankel) matrix with the help of the DFT or real discrete trigonometric transforms into a generalized Cauchy or a generalized Vandermonde matrix, which can be done with $O(n \log n)$ complexity in a stable way. The advantage of these classes of matrices is that permutation of rows or columns does not destroy the structure. Therefore pivoting strategies can be applied to stabilize the algorithm. Slightly different is the idea, which was proposed in [30, 31], to transform the Toeplitz (or Hankel) system directly into a rational interpolation problem at roots of unity. One advantage of this approach is that one can guarantee that the length of the transformations is a power of 2.

Algorithms with complexity less than $O(n^2)$ are called *superfast*. Superfast solvers are based on divide and conquer strategies. The idea for a superfast Toeplitz solver was first announced in the Ph.D. thesis of Morf [41]. Superfast algorithms were designed by Sugiyama et al. [47], Bitmead and Anderson [4], Brent, Gustavson, and Yun [7], and Morf [42]. More recent algorithms can be found in [1, 2, 3, 8, 14, 20, 22, 39, 40, 43, 46].

The main disadvantage of these algorithms is that they cannot handle nearly singular leading principal submatrices and are therefore unstable in the nonsymmetric case. To overcome this problem, Gutknecht [26] and Gutknecht and Hochbruck [27, 28] developed an algorithm that combines the look-ahead idea with divide and conquer techniques. Because in most practical problems the look-ahead step will be small compared to the order of the system that is to be solved, the algorithm is *generically* superfast.

In [52] a divide and conquer approach was used to obtain a superfast algorithm for rational interpolation at roots of unity. The algorithm consists of two stages. The first part, for a small number of nodes, consists of the fast algorithm from [38] including pivoting. The second part is a doubling procedure. Instead of pivoting,

"difficult" points are sorted out and are treated at the end. Also, iterative refinement is applied to stabilize the algorithm. In this way a stabilized generically superfast solver for indefinite Hankel systems was obtained, i.e., superfast in case the number of difficult points is small, and fast otherwise.

In this paper we use a similar approach for Toeplitz systems. In contrast with [52], the Toeplitz is not first transformed into another matrix, but an explicit formula for the inverse of a Toeplitz matrix is used. This formula involves the values of the "fundamental system" at roots of unity, i.e., the DFT of the fundamental system. It will be presented in section 2. The fundamental system is a pair of polynomials containing all the information about the Toeplitz matrix. These polynomials are related to two linearized rational interpolation problems at roots of unity. This connection will be explained in section 3. To solve these interpolation problems, we extend the superfast algorithm presented in [52]. We will incorporate "downdating" into this algorithm as an additional stabilizing technique. In section 5 we will present numerical examples and compare our results with those obtained in [52]. Note that in [52] the size of the Hankel system is limited to a power of 2, whereas we can handle Toeplitz systems of arbitrary size.

2. DFT representation of Toeplitz matrix inverses. Let us introduce the following notation. To each column vector $u = [u_k]_{k=0}^{n} \in \mathbb{C}^{n+1}$ we associate the polynomial $u(z) := \sum_{k=0}^{n} u_k z^k \in \mathbb{C}[z]$. The column vector $\hat{u}$ is defined as $\hat{u} := [u_{n-k}]_{k=0}^{n}$. Thus, $\hat{u}(z) = z^n u(z^{-1})$.

Let $a_{-n} \in \mathbb{C}$ be arbitrary and let $\tilde{T} = \tilde{T}_n$ be given by the $n \times (n+1)$ matrix

$$\tilde{T} := [a_{k-l}]_{k,l=0}^{n-1,n} = \left[\begin{array}{cccc|c} a_0 & a_{-1} & \cdots & a_{-n+1} & a_{-n} \\ a_1 & a_0 & \ddots & & a_{-n+1} \\ \vdots & & \ddots & \ddots & \vdots \\ a_{n-1} & \cdots & \cdots & a_0 & a_{-1} \end{array}\right] = \left[\begin{array}{c|c} \multicolumn{1}{c}{} & a_{-n} \\ T & \vdots \\ \multicolumn{1}{c}{} & a_{-1} \end{array}\right].$$

The polynomials $u(z)$ and $v(z)$ are called the *canonical fundamental system* of T if

- $\tilde{T}u = e_1$ and $u_n = 0$, where $e_1 := [\ 1 \quad 0 \quad \cdots \quad 0\ ]^T$,
- $\tilde{T}v = 0$ and $v_n = 1$.

In other words, $u(z)$ is a polynomial of degree $n-1$ at most while $v(z)$ is a monic polynomial of degree n such that

$$\left[\begin{array}{cccc} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & & \ddots & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_0 \end{array}\right] \left[\begin{array}{c} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array}\right]$$

and

$$\left[\begin{array}{cccc} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & & \ddots & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_0 \end{array}\right] \left[\begin{array}{c} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{array}\right] = - \left[\begin{array}{c} a_{-n} \\ a_{-n+1} \\ \vdots \\ a_{-1} \end{array}\right].$$

As T is assumed to be nonsingular, these polynomials do exist. Moreover, $u(z)$ is unique whereas $v(z)$ is uniquely determined given a particular value of a_{-n}. For our

purposes the specific value of a_{-n} is immaterial and thus $v(z)$ is in fact unique up to a linear combination with $u(z)$. Note that by canceling the last (zero) component of u, one obtains the first column of T^{-1}.

Remark. Let $w = [\ w_0 \ \cdots \ w_{n-1}\ ]^T \in \mathbb{C}^n$ be the last column of T_n^{-1}. If $T_{n-1} := [a_{k-l}]_{k,l=0}^{n-2}$ is also nonsingular, then Cramer's rule implies that $w_{n-1} \neq 0$ and one may choose $v(z)$ as $v(z) = zw(z)/w_{n-1}$. This choice determines the value of a_{-n}. Also, if T_n is symmetric, then $w(z) = \hat{u}(z)$.

The *generating function* $M(t,s)$ of a matrix $M = [m_{k,l}]_{k,l=0}^{p,q}$ is defined as

$$M(t,s) := \sum_{k=0}^{p}\sum_{l=0}^{q} m_{k,l} t^k s^l.$$

THEOREM 2.1. *The generating function of T^{-1} is given by*

$$T^{-1}(t,s) = \frac{u(t)\hat{v}(s) - v(t)\hat{u}(s)}{1 - ts}. \tag{2.1}$$

Proof. For the proof, see [34, p. 32]. ∎

The matrix whose generating function is given by the right-hand side of (2.1) is called the *Toeplitz Bezoutian* of the polynomials $u(z)$ and $v(z)$.

Let $N \geq n$ be a power of 2. From the previous theorem we will now derive a formula for T^{-1} that will enable us to calculate the matrix-vector product $T^{-1}b$ in $\mathcal{O}(N \log N)$ floating point operations.

Define $\omega_0, \ldots, \omega_{2N-1}$ as the $2N$th roots of unity,

$$\omega_k := \exp\Big(\frac{2\pi i}{2N} k\Big), \qquad k = 0, 1, \ldots, 2N-1,$$

and let $\omega_k^+ := \omega_{2k}$ and $\omega_k^- := \omega_{2k+1}$ for $k = 0, 1, \ldots, N-1$. Note that $\omega_0^+, \ldots, \omega_{N-1}^+$ are the Nth roots of unity whereas $\omega_0^-, \ldots, \omega_{N-1}^-$ are the Nth roots of -1. Let $\eta := \exp(\pi i/N)$. Define the matrices $\mathcal{F}_+$ and $\mathcal{F}_-$ as

$$\mathcal{F}_+ := \big[\ (\omega_k^+)^l\ \big]_{k,l=0}^{N-1} \quad \text{and} \quad \mathcal{F}_- := \big[\ (\omega_k^-)^l\ \big]_{k,l=0}^{N-1}.$$

Then $\mathcal{F}_+/\sqrt{N}$ is unitary and $\mathcal{F}_- = \mathcal{F}_+ \operatorname{diag}(1, \eta, \ldots, \eta^{N-1})$. Matrix-vector products involving $\mathcal{F}_+$ or $\mathcal{F}_-$ can be evaluated with arithmetic complexity $\mathcal{O}(N \log N)$ via the celebrated FFT.

Let $[T^{-1}]_N$ denote the $N \times N$ matrix

$$[T^{-1}]_N := \begin{bmatrix} T^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{N \times N}.$$

Define $D := \operatorname{diag}\big((\omega_0^+)^{-n}, \ldots, (\omega_{N-1}^+)^{-n}\big)$, $D_\pm(u) := \operatorname{diag}\big(u(\omega_0^\pm), \ldots, u(\omega_{N-1}^\pm)\big)$ and similar for the matrices $D_\pm(v)$.

THEOREM 2.2. *The matrix $[T^{-1}]_N$ can be expressed as*

$$[T^{-1}]_N = \frac{1}{2}\mathcal{F}_-^{-1}\big[\, D_-(u)\mathcal{F}_-\mathcal{F}_+^{-1}D_+(v) - D_-(v)\mathcal{F}_-\mathcal{F}_+^{-1}D_+(u)\,\big]D\mathcal{F}_+. \tag{2.2}$$

Proof. As $\mathcal{F}_+^{-1} = \frac{1}{N}\mathcal{F}_+^H$, it follows that

$$\mathcal{F}_-[T^{-1}]_N\mathcal{F}_+^{-1} =: [c_{k,l}]_{k,l=0}^{N-1},$$

where

$$\begin{aligned} c_{k,l} &= \frac{1}{N} T^{-1}(\omega_k^-, 1/\omega_l^+) \\ &= \frac{1}{N} \frac{u(\omega_k^-)\hat{v}(1/\omega_l^+) - v(\omega_k^-)\hat{u}(1/\omega_l^+)}{1-\omega_k^-/\omega_l^+} \\ &= \frac{1}{N} \frac{u(\omega_k^-)v(\omega_l^+) - v(\omega_k^-)u(\omega_l^+)}{1-\omega_k^-/\omega_l^+} [\omega_l^+]^{-n}. \end{aligned}$$

One can easily verify that

$$\mathcal{F}_- \mathcal{F}_+^{-1} = \frac{2}{N} \left[\begin{array}{c} \\ \dfrac{1}{1-\omega_k^-/\omega_l^+} \\ \\ \end{array} \right]_{k,l=0}^{N-1}.$$

The expression for $[T^{-1}]_N$ then follows immediately. ∎

The formula in the previous theorem allows us to compute the product $x = T^{-1}b$ in $\mathcal{O}(N \log N)$ flops provided that the polynomials $u(z)$ and $v(z)$ are known. Indeed,

$$\left[\begin{array}{c} x \\ 0 \end{array} \right] = [T^{-1}]_N \left[\begin{array}{c} b \\ 0 \end{array} \right]$$

and the multiplication by $[T^{-1}]_N$ can be done via six N-point (inverse) FFTs and $\mathcal{O}(N)$ flops. For preprocessing one has to compute the values of $u(\omega_k)$ and $v(\omega_k)$ for $k = 0, 1, \ldots, 2N-1$, which amounts to two $2N$-point FFTs.

3. Interpolation interpretation. For a given Toeplitz matrix $T = [\, a_{k-l}]_{k,l=0}^{n-1}$, we introduce the function

$$a(z) := \sum_{k=1-n}^{n-1} a_k z^k,$$

where z is considered as a complex variable.

THEOREM 3.1. *Suppose $a_{-n} = 0$. Then the polynomials $u(z)$ and $v(z)$ are the canonical fundamental system of T if and only if there exist polynomials $\hat{r}_u(z)$ and $\hat{r}_v(z)$ satisfying the linearized rational interpolation conditions*

$$\omega_k^n \hat{r}_u(\omega_k) - a(\omega_k)u(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg \hat{r}_u(z) \le 2N-n$, $\hat{r}_{u,2N-n} = 1$ and $\deg u(z) \le n$, $u_n = 0$, and

$$\omega_k^n \hat{r}_v(\omega_k) - a(\omega_k)v(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg \hat{r}_v(z) \le 2N-n$, $\hat{r}_{v,2N-n} = 0$ and $\deg v(z) \le n$, $v_n = 1$.

Proof. Let $\delta \in \mathbb{C}$. Let $w(z)$ be a polynomial of degree $\le n$ such that $\tilde{T}w = \delta e_1$. The case $\delta = 1$ corresponds to $w(z) = u(z)$, whereas $\delta = 0$ corresponds to $w(z) = v(z)$. The condition $\tilde{T}w = \delta e_1$ is equivalent to the existence of polynomials $r_-(z)$ and $r_+(z)$ of degree $\le n-1$ with $r_{-,0} = \delta$ such that

$$a(z)w(z) = r_-(1/z) + z^n r_+(z).$$

It follows that

$$\begin{aligned} a(\omega_k)w(\omega_k) &= r_-(\omega_k^{-1}) + \omega_k^n r_+(\omega_k) \\ &= r_-(\omega_k^{-1}) + \omega_k^{n-2N} r_+(\omega_k) \end{aligned}$$

for $k = 0, 1, \ldots, 2N-1$. Define

$$r(z) := r_-(z) + z^{2N-n} r_+(1/z).$$

Then $r(z)$ is a polynomial of degree $\leq 2N-n$ and $r_0 = \delta$. Thus

$$\begin{aligned} a(\omega_k) w(\omega_k) &= r(\omega_k^{-1}) \\ &= \omega_k^n \left[\omega_k^{2N-n} r(\omega_k^{-1})\right] \\ &= \omega_k^n \hat{r}(\omega_k) \end{aligned}$$

for $k = 0, 1, \ldots, 2N-1$. In other words,

$$\omega_k^n \hat{r}(\omega_k) - a(\omega_k) w(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg \hat{r}(z) \leq 2N-n$, $\hat{r}_{2N-n} = \delta$, and $\deg w(z) \leq n$. ∎

These interpolation conditions can also be written as follows:

$$\begin{bmatrix} \omega_k^n & -a(\omega_k) \end{bmatrix} B^\star(\omega_k) = \begin{bmatrix} 0 & 0 \end{bmatrix}, \qquad k = 0, 1, \ldots, 2N-1,$$

where

$$B^\star(z) := \begin{bmatrix} \hat{r}_u(z) & \hat{r}_v(z) \\ u(z) & v(z) \end{bmatrix} \in \mathbb{C}[z]^{2\times 2}$$

is a 2×2 matrix polynomial. The degree of the first row of $B^\star(z)$ is equal to $2N-n$ whereas the degree of the second row of $B^\star(z)$ is equal to n. The second row of $B^\star(z)$ gives us the inversion parameters that are needed in formula (2.2).

4. A stabilized divide and conquer approach. Define

$$f_k := \begin{bmatrix} \omega_k^n \\ -a(\omega_k) \end{bmatrix} \in \mathbb{C}^{2\times 1}$$

for $k = 0, 1, \ldots, 2N-1$ and let $\mathcal{S}$ be the set of all the column vector polynomials $w(z) \in \mathbb{C}[z]^{2\times 1}$ that satisfy the interpolation conditions

$$f_k^T w(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1. \tag{4.1}$$

If $w(z) \in \mathbb{C}[z]^{2\times 1}$ is an arbitrary vector polynomial, then the left-hand side of (4.1) is called the *residual* with respect to $w(z)$ at the interpolation point ω_k.

The set $\mathcal{S}$ forms a submodule of the $\mathbb{C}[z]$-module $\mathbb{C}[z]^{2\times 1}$. A basis for $\mathcal{S}$ always consists of exactly two elements [49, Theorem 3.1]. Let $\{B_1(z), B_2(z)\}$ be a basis for $\mathcal{S}$. Then every element $w(z) \in \mathcal{S}$ can be written in a unique way as $w(z) = \alpha_1(z) B_1(z) + \alpha_2(z) B_2(z)$ with $\alpha_1(z), \alpha_2(z) \in \mathbb{C}[z]$. The matrix polynomial $B(z) := [\, B_1(z)\ B_2(z) \,] \in \mathbb{C}[z]^{2\times 2}$ is called a *basis matrix*. Basis matrices can be characterized as follows.

THEOREM 4.1. *A matrix polynomial $C(z) = [\, C_1(z)\ C_2(z) \,] \in \mathbb{C}[z]^{2\times 2}$ is a basis matrix if and only if $C_1(z), C_2(z) \in \mathcal{S}$ and $\deg \det C(z) = 2N$.*

Proof. The proof follows immediately from [49, Theorem 4.1]. ∎

Note that $B^\star(z)$ is a basis matrix.

Within the submodule $\mathcal{S}$ we want to be able to consider solutions $w(z)$ that satisfy additional conditions concerning their degree-structure. To describe the degree-structure of a vector polynomial, we use the concept of τ-degree [49]. Let $\tau \in \mathbb{Z}$.

The τ-*degree* of a vector polynomial $w(z) = [\, w_1(z)\ w_2(z)\,]^T \in \mathbb{C}[z]^{2\times 1}$ is defined as a generalization of the classical degree:

$$\tau\text{-deg}\, w(z) := \max\{\, \deg w_1(z), \deg w_2(z) - \tau\,\}$$

with τ-deg $0 := -\infty$. The τ-highest degree coefficient of a vector polynomial

$$[\, w_1(z)\ w_2(z)\,]^T$$

with τ-degree δ is defined as the vector $[\, \omega_1\ \omega_2\,]^T$ with ω_1 the coefficient of z^δ in $w_1(z)$ and ω_2 the coefficient of $z^{\delta+\tau}$ in $w_2(z)$. A set of vector polynomials in $\mathbb{C}[z]^{2\times 1}$ is called τ-*reduced* if the τ-highest degree coefficients are linearly independent. Every basis of $\mathcal{S}$ can be transformed into a τ-reduced one. For details, we refer to [49]. Once we have a basis in τ-reduced form, the elements of $\mathcal{S}$ can be parametrized as follows.

THEOREM 4.2. *Let* $\{\, B_1(z), B_2(z)\,\}$ *be a* τ*-reduced basis for* $\mathcal{S}$*. Define* $\delta_1 := \tau$-deg $B_1(z)$ *and* $\delta_2 := \tau$-deg $B_2(z)$*. Then every element* $w(z) \in \mathcal{S}$ *having* τ*-degree* $\le \delta$ *can be written in a unique way as*

$$w(z) = \alpha_1(z)B_1(z) + \alpha_2(z)B_2(z)$$

with $\alpha_1(z), \alpha_2(z) \in \mathbb{C}[z]$*,* $\deg \alpha_1(z) \le \delta - \delta_1$*, and* $\deg \alpha_2(z) \le \delta - \delta_2$*.*

Proof. For the proof, see Van Barel and Bultheel [49, Theorem 3.2]. ∎

We can now summarize our aim as follows: we want to design an algorithm for computing the 2×2 matrix polynomial $B^\star(z)$ as a τ-reduced basis matrix that corresponds to the interpolation data (ω_k, f_k), $k = 0, 1, \ldots, 2N-1$, where we set $\tau := 2(n - N)$.

The following theorem will enable us to devise an interpolation algorithm that is based on a *divide and conquer* approach. It shows how basis matrices can be coupled in case the degree-structure is important.

THEOREM 4.3. *Suppose* K *is a positive integer. Let* $\sigma_1, \ldots, \sigma_K \in \mathbb{C}$ *be mutually distinct and let* $\phi_1, \ldots, \phi_K \in \mathbb{C}^{2\times 1}$*. Suppose that* $\phi_k \neq [\ 0\ \ 0\ ]^T$ *for* $k = 1, \ldots, K$*. Let* $1 \le \kappa \le K$*. Let* $\tau_K \in \mathbb{Z}$*. Suppose that* $B_\kappa(z) \in \mathbb{C}[z]^{2\times 2}$ *is a* τ_K*-reduced basis matrix with basis vectors having* τ_K*-degree* δ_1 *and* δ_2*, respectively, corresponding to the interpolation data*

$$\{\, (\sigma_k, \phi_k)\ :\ k = 1, \ldots, \kappa\,\}.$$

Let $\tau_{\kappa\to K} := \delta_1 - \delta_2$*. Let* $B_{\kappa\to K}(z) \in \mathbb{C}[z]^{2\times 2}$ *be a* $\tau_{\kappa\to K}$*-reduced basis matrix corresponding to the interpolation data*

$$\{\, (\sigma_k, B_\kappa^T(\sigma_k)\phi_k)\ :\ k = \kappa+1, \ldots, K\,\}.$$

Then $B_K(z) := B_\kappa(z)B_{\kappa\to K}(z)$ *is a* τ_K*-reduced basis matrix corresponding to the interpolation data*

$$\{\, (\sigma_k, \phi_k)\ :\ k = 1, \ldots, K\,\}.$$

Proof. For the proof, see Van Barel and Bultheel [50, Theorem 3]. ∎

The following algorithm implements this theorem. We start with the $2N$th roots of unity as interpolation points. They are split into the Nth roots of unity s_1 and the rotated Nth roots of unity s_2. The fact that we are dealing with (rotated) roots of unity enables us to do all polynomial evaluations and multiplications via FFTs (and diagonal scalings). As N is a power of 2, this process can be repeated. At the lowest level the interpolation problems are solved by the fast solver RATINT developed by Kravanja and Van Barel [38].

recursive function $[\,B(z), s_{\mathrm{bad}}\,] \leftarrow \textsc{RecRatInt}(s, L_s, R_s, N_s, \tau)$
-- $\tau \in \mathbb{Z}$
-- $N_s = 2^{p+1}$ for some $p \in \mathbb{N}$: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(z) \in \mathbb{C}[z]^{2\times 2}$: a τ-reduced basis matrix corresponding to the given interpolation data
-- s_{bad}: a complex column vector containing the difficult interpolation points
if $N_s > 2^{\mathrm{limit}}$ **then**
  $[\,s_1, L_{s_1}, R_{s_1}, s_2, L_{s_2}, R_{s_2}\,] \leftarrow \textsc{Split}(s, L_s, R_s)$
  $[\,B_1(z), s_{\mathrm{bad},1}\,] \leftarrow \textsc{RecRatInt}(s_1, L_{s_1}, R_{s_1}, N_s/2, \tau)$
  for $k = 1(1)N_s/2$
    $[\,\tilde{L}_{s_2}(k)\;\; \tilde{R}_{s_2}(k)\,] \leftarrow [\,L_{s_2}(k)\;\; R_{s_2}(k)\,] \cdot B_1(s_2(k))$
  end for
  $\tilde{\tau} \leftarrow$ the difference between the left and right τ-degrees of $B_1(z)$
  $[\,\tilde{B}_2(z), \tilde{s}_{\mathrm{bad},2}\,] \leftarrow \textsc{RecRatInt}(s_2, \tilde{L}_{s_2}, \tilde{R}_{s_2}, N_s/2, \tilde{\tau})$
  $B(z) \leftarrow B_1(z) \cdot \tilde{B}_2(z)$
  $s_{\mathrm{bad}} \leftarrow s_{\mathrm{bad},1} \oplus \tilde{s}_{\mathrm{bad},2}$
else
  $[\,B(z), s_{\mathrm{bad}}\,] \leftarrow \textsc{RatInt}(s, L_s, R_s, N_s, \tau)$
end if
if $N_s = 2^{\mathrm{downdating}}$ **then**
  $s^+ \leftarrow s \ominus s_{\mathrm{bad}}$
  $[\,B(z), s_{\mathrm{bad},3}\,] \leftarrow \textsc{Downdating}(s^+, L_{s^+}, R_{s^+}, N_s)$
  $s_{\mathrm{bad}} \leftarrow s_{\mathrm{bad}} \oplus s_{\mathrm{bad},3}$
end if
if $N_s = 2^{\mathrm{reflimit}}$ **then**
  $s^+ \leftarrow s \ominus s_{\mathrm{bad}}$
  $[\,B(z)\,] \leftarrow \textsc{ItRef}(B(z), s^+, L_{s^+}, R_{s^+}, N_s, N_{\mathrm{ref}})$
end if
return

function $[\,B(z)\,] \leftarrow \textsc{RatIntAll}(s, L_s, R_s, N_s, \tau)$
-- $\tau \in \mathbb{Z}$
-- $N_s = 2^{p+1}$ for some $p \in \mathbb{N}$: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(z) \in \mathbb{C}[z]^{2\times 2}$: a τ-reduced basis matrix corresponding to the given interpolation data
$[\,B^+(z), s_{\mathrm{bad}}\,] \leftarrow \textsc{RecRatInt}(s, L_s, R_s, N_s, \tau)$
$N_{\mathrm{bad}} \leftarrow \textsc{Size}(s_{\mathrm{bad}})$
if $N_{\mathrm{bad}} > 0$ **then**
  calculate L_{bad} and R_{bad}
  $\tau^- \leftarrow$ the difference between the left and right τ-degrees of $B^+(z)$
  $[\,B^-(z)\,] \leftarrow \textsc{RatInt}(s_{\mathrm{bad}}, L_{\mathrm{bad}}, R_{\mathrm{bad}}, N_{\mathrm{bad}}, \tau^-)$
  $B(z) \leftarrow B^+(z) \cdot B^-(z)$
end if
return

Superfast Hankel (Toeplitz) solvers are notoriously unstable when applied to indefinite systems. Algorithm RecRatInt is stabilized in three ways.

Difficult points. During the execution of RatInt all the residuals at interpolation points that may be chosen as pivot elements can be smaller (in modulus) than a certain threshold. By processing these interpolation points the accuracy would decrease. These points are therefore marked as "difficult." They are handled only at the very end, after RecRatInt has finished, via the fast-only algorithm RatInt. If at this stage the corresponding transformed residuals are still small, this indicates that the problem is ill-conditioned. The overall complexity of our algorithm will be $\mathcal{O}(n \log^2 n)$ as long as the number of difficult points is not too large.

Iterative improvement. The approximations for the coefficients of the polynomials that appear in the basis matrix $B(z)$ can be updated iteratively by using an inversion formula for coupled Vandermonde matrices. For more details, we refer to Van Barel and Kravanja [52]. Iterative refinement can be applied at one or more intermediate levels of the divide and conquer process. In algorithm RecRatInt, it is used only if the number of interpolation conditions is equal to 2^{reflimit}.

Downdating. Finite precision arithmetic can lead to a situation where

$$f_k^T B(s_k) \approx \begin{bmatrix} 0 & 0 \end{bmatrix}$$

for one or more interpolation points s_k. As the matrix $B(s_k)$ is singular, there exists a vector $v \in \mathbb{C}^2$ such that

$$B(s_k)v = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Define

$$B(z) =: \begin{bmatrix} B_L(z) & B_R(z) \end{bmatrix}, \qquad v =: \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$

and let $\alpha_L := \tau\text{-deg}B_L(z)$ and $\alpha_R := \tau\text{-deg}B_R(z)$. If $\alpha_L \geq \alpha_R$ and $v_L \neq 0$, then we replace $B_L(z)$ by

$$B_L(z) \leftarrow B(z)v/(z - s_k).$$

If, on the other hand, $\alpha_L < \alpha_R$ and $v_R \neq 0$, then we replace $B_R(z)$ by

$$B_R(z) \leftarrow B(z)v/(z - s_k).$$

If $v_L = 0$, then $B_R(z)$ is divisible by $z - s_k$. Similarly, if $v_R = 0$, then $B_L(z)$ is divisible by $z - s_k$. These considerations lead to the following algorithm.

function $[\, B(z), s_{\text{bad}} \,] \leftarrow$ Downdating$(B(z), s, L_s, R_s, N_s)$
-- N_s: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(z) \in \mathbb{C}[z]^{2\times 2}$
-- on input: a basis matrix corresponding to the given interpolation data
-- on output: the corresponding downdated basis matrix
-- s_{bad}: a complex column vector containing the interpolation points that have been downdated
$s_{\text{bad}} \leftarrow \emptyset$

for $k = 1(1)N_s$
  if $\| \begin{bmatrix} L_s(k) & R_s(k) \end{bmatrix} \| > \eta$ **then**
    Choose $v \in \mathbb{C}^2$ such that $B(s(k))v = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and $\|v\| = 1$
    -- Let $B(z) =: \begin{bmatrix} B_L(z) & B_R(z) \end{bmatrix}$ and $v =: \begin{bmatrix} v_L & v_R \end{bmatrix}^T$
    $\alpha_L \leftarrow \tau\text{-deg}B_L(z)$
    $\alpha_R \leftarrow \tau\text{-deg}B_R(z)$
    if $\alpha_L \geq \alpha_R$ **then**
      if $v_L \neq 0$ **then**
        $B_L(z) \leftarrow B(z)v/(z - s(k))$
      else
        $B_R(z) \leftarrow B_R(z)/(z - s(k))$
      end if
    else
      if $v_R \neq 0$ **then**
        $B_R(z) \leftarrow B(z)v/(z - s(k))$
      else
        $B_L(z) \leftarrow B_L(z)/(z - s(k))$
      end if
    end if
    $s_{\text{bad}} \leftarrow s_{\text{bad}} \oplus s(k)$
  end if
end for
return

5. Numerical experiments. We consider double precision Toeplitz matrices T_n whose entries are real and random uniformly distributed in $[0, 1]$ with $n = 2^k$ for $k = 1, \ldots, 18$. Note that $2^{18} = 262144$. The right-hand sides $b_n \in \mathbb{R}^n$ are calculated such that $x_n := T_n^{-1} b_n = [1 \ \cdots \ 1]^T$. The calculations were done by an IBM SP2 with machine precision $\approx 0.22\, 10^{-15}$ in double precision. Our software is available at

http://www.cs.kuleuven.ac.be/~marc/software/

Figures 5.1 and 5.2 show the results obtained by our algorithm in which no iterative refinement is applied (the symbols "+") and in which up to 10 steps of iterative refinement are applied (the symbols "o") to enhance the accuracy of the computed solution to the Toeplitz system. Interpolation problems of size less than or equal to 2^8 are solved by our fast-only algorithm. For each value of k we consider five (random) Toeplitz matrices.

Our next figures represent timings. As on our computer system measurements of execution times are done in units of 0.01 seconds, we limit the k-axis to that part where the results are meaningful. This is why in the following figures k does not start at 1 but at a larger value.

Figure 5.3 shows the execution time (in seconds) for Gaussian elimination with partial pivoting (these results were calculated via the LAPACK routines ZGETRF and ZGETRS), our fast algorithm, and our superfast algorithm in which no iterative refinement is applied. The results are indicated with the symbols "+", "o", and "x", respectively.

Figure 5.4 presents the results shown in Figure 5.3 in a different way. It gives the magnification of the execution time. For each k, it tells us by which factor the execution time is to be multiplied if we go from $k - 1$ to k.

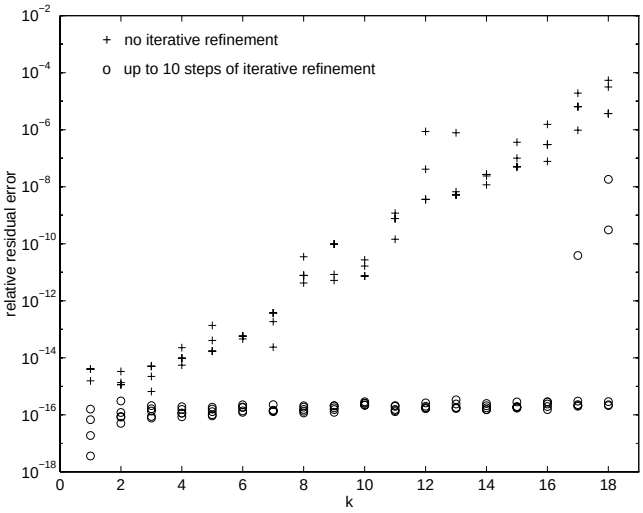


FIG. 5.1. $\frac{\|b_n - T_n \hat{x}_n\|_1}{\|b_n\|_1}$ *versus* $k = \log_2 n$ *for* $k = 1, \ldots, 18$.

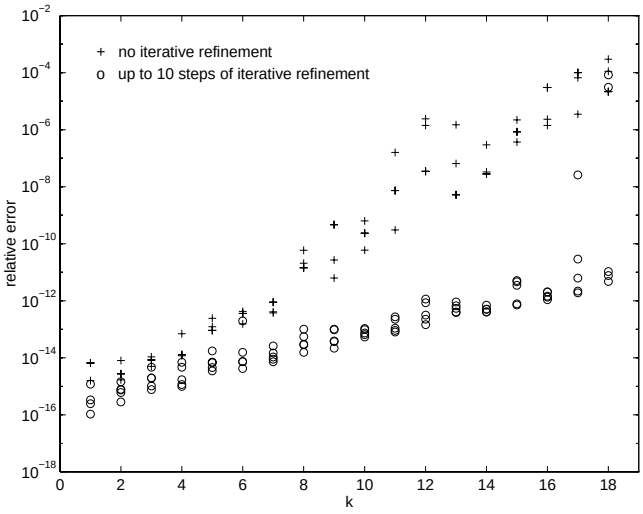


FIG. 5.2. $\frac{\|\hat{x}_n - x_n\|_1}{\|x_n\|_1}$ *versus* $k = \log_2 n$ *for* $k = 1, \ldots, 18$.

Figures 5.5 and 5.6 are related to our superfast solver. For each value of k we consider five (random) Toeplitz matrices. No iterative refinement is applied. Figure 5.5 shows the percentage of the execution time spent to compute the input data for the interpolation problem formulated in Theorem 3.1, i.e., the time needed to evaluate the $a(\omega_k)$'s. Figure 5.6 shows the percentage of the execution time spent to apply the inversion formula given in Theorem 2.2 once the interpolation problem has been solved.

We also consider matrices of size $n = 10000(5000)100000$. The entries are again real and random uniformly distributed in $[0, 1]$, and the right-hand sides are again

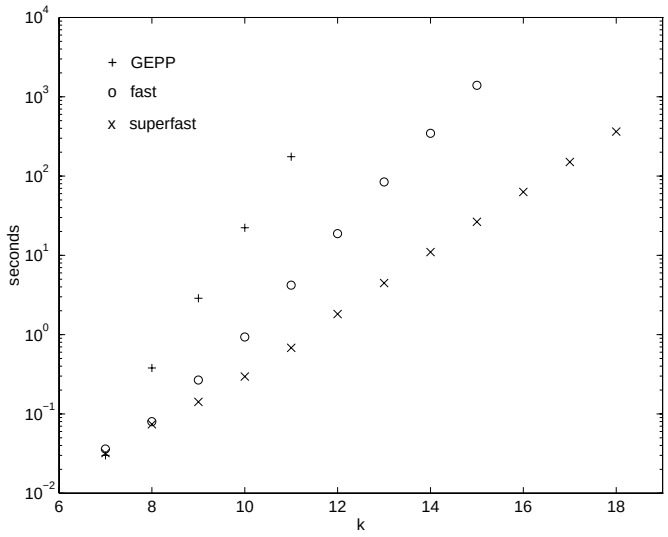


FIG. 5.3. *Execution time in seconds.*

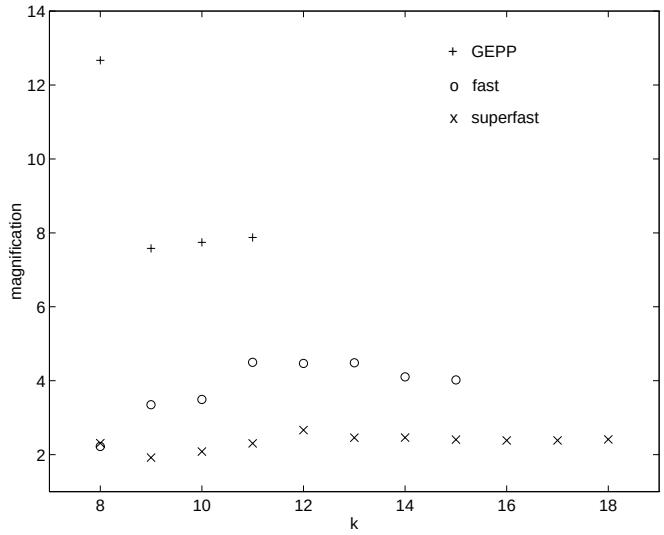


FIG. 5.4. *Magnification of the execution time.*

calculated such that all the entries of the solution vector are equal to 1. For each value of n we consider five matrices. Figure 5.7 shows the execution time (in seconds). The results are indicated with the symbol "x". The symbols "o" correspond to the case where n is a power of 2.

One expects that for n in the range $2^{k-1} < n \leq 2^k$ the execution time is more or less equal to that of the system of size 2^k. In practice, the execution time is less. This can be explained as follows. At the lowest level some of the first interpolation problems can be solved via polynomial interpolation, i.e., by applying FFT.

The computed solution can be refined iteratively. Figure 5.8 shows how much execution time is spent on iterative refinement as percentage of the execution time in

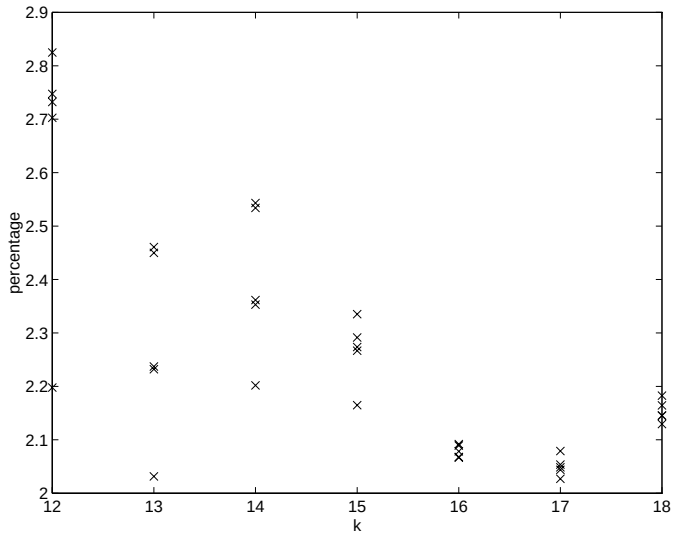


FIG. 5.5. *Percentage of the execution time spent to compute the input data for the interpolation problem.*

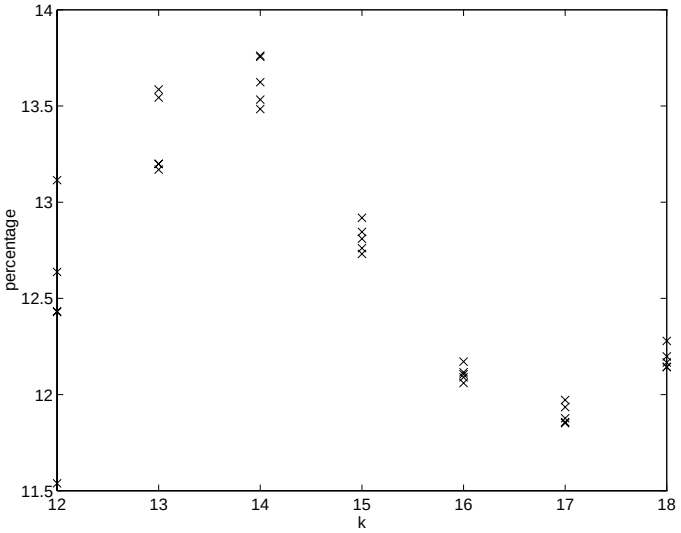


FIG. 5.6. *Percentage of the execution time spent to apply the inversion formula.*

which no iterative refinement is applied. We consider one, two, three, or four steps of iterative refinement. The results are represented by the symbols "x", "o", "+", and "$\star$", respectively. For each value of k and each number of iterative refinement steps, five Toeplitz matrices are considered.

So far we have considered only iterative refinement at the Toeplitz level, i.e., we have refined the computed solution to the Toeplitz system iteratively. Iterative refinement can also be applied at the interpolation level. In our next experiment we apply up to four steps of iterative refinement at the highest interpolation level. The timing results are shown in Figure 5.9. We compare the execution time spent

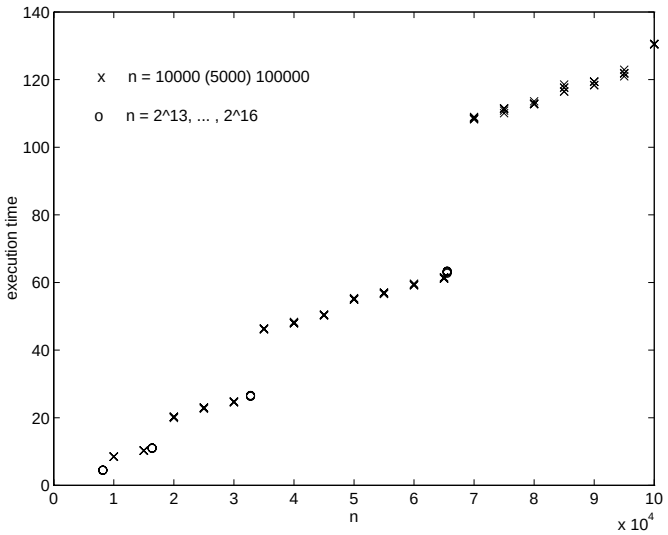


FIG. 5.7. *Execution time for* $n = 10000(5000)100000$ *and* $n = 2^{13}, \ldots, 2^{16}$.

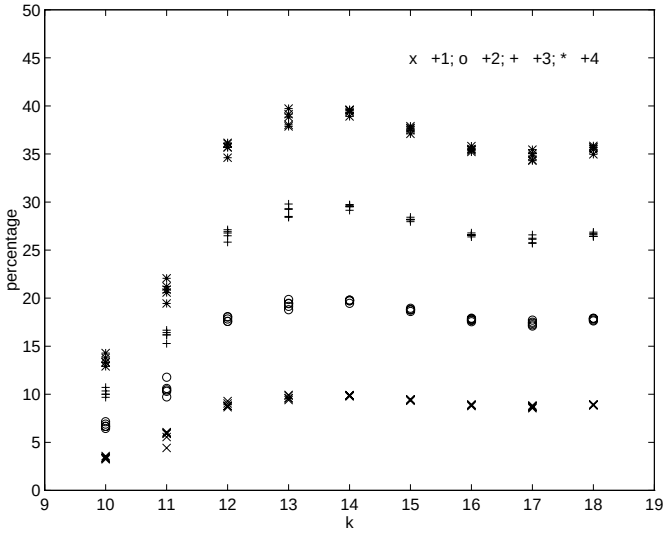


FIG. 5.8. *Time spent on iterative refinement as percentage of the execution time in which no iterative refinement is applied.*

on this kind of iterative refinement to the total execution time in which no iterative refinement whatsoever is applied. Observe that this kind of iterative refinement is much more expensive than iterative refinement applied at the Toeplitz level.

Iterative refinement at an interpolation level may be preceded by downdating. Numerical experiments indicate that the time needed to search the interpolation points that have to be downdated is approximately 45% of the time needed for one step of iterative refinement.

The following example illustrates how important it is to find the proper combination of the stabilizing techniques that we have developed. For a certain matrix

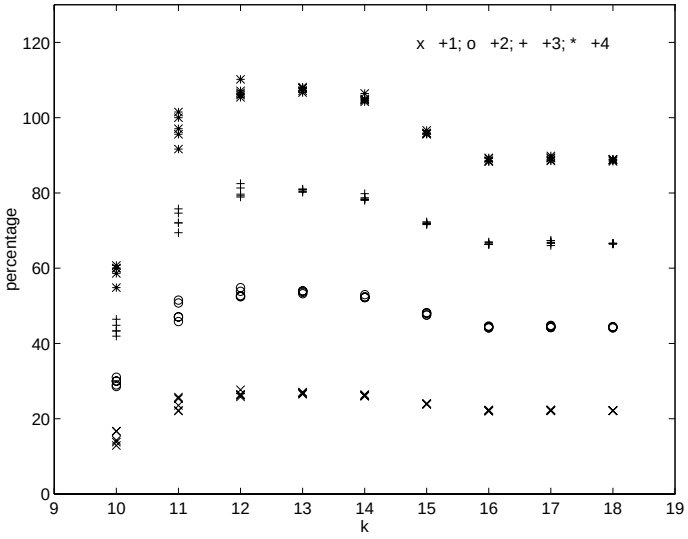


FIG. 5.9. *Up to four steps of iterative refinement at the highest interpolation level. We compare the corresponding execution time to the total execution time in which no iterative refinement at all is applied.*

of size 2^{18} whose entries are random uniformly distributed in the interval $[0, 1]$, we have observed the following. By applying at most 10 steps of iterative refinement on the interpolation problems of size 2^{18} (this is the one but highest interpolation level; remember that a matrix of size 2^{18} corresponds to 2^{19} interpolation conditions), by considering 85 difficult points, and by applying iterative refinement at the Toeplitz level, we obtain an approximation for the solution whose relative residual error is $\mathcal{O}(10^{-15})$. If we do not consider difficult points and do not use iterative refinement at the interpolation level, then the computed approximation is so bad that iterative refinement at the Toeplitz level fails. The same holds if we apply only iterative refinement on the interpolation problems of size 2^{19}. This clearly shows the importance of combining our stabilizing tools in the correct way. One can of course apply iterative refinement on each interpolation level, but this is very costly. One has to find the correct balance between accuracy and cost. This will be the subject of future research.

REFERENCES

[1] G. S. AMMAR AND W. B. GRAGG, *The generalized Schur algorithm for the superfast solution of Toeplitz systems*, in Rational Approximation and Its Applications in Mathematics and Physics, J. Gilewicz, M. Pindor, and W. Siemaszko, eds., Lecture Notes in Math. 1237, Springer, New York, 1987, pp. 315–330.

[2] G. S. AMMAR AND W. B. GRAGG, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[3] G. S. AMMAR AND W. B. GRAGG, *Numerical experience with a superfast real Toeplitz solver*, Linear Algebra Appl., 121 (1989), pp. 185–206.

[4] R. R. BITMEAD AND B. D. O. ANDERSON, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra Appl., 34 (1980), pp. 103–116.

[5] A. W. BOJANCZYK, R. P. BRENT, F. R. DE HOOG, AND D. R. SWEET, *On the stability of the Bareiss and related Toeplitz factorization algorithms*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 40–57.

[6] A. W. BOJANCZYK AND G. HEINIG, *A multi-step algorithm for Hankel matrices*, J. Complexity,

10 (1994), pp. 142–164.
[7] R. Brent, F. Gustavson, and D. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.
[8] S. Cabay and D. K. Choi, *Algebraic computations of scaled Padé fractions*, SIAM J. Comput., 15 (1986), pp. 243–270.
[9] 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.
[10] T. F. Chan and P. C. Hansen, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Process., 40 (1992), pp. 1079–1090.
[11] T. F. Chan and P. C. Hansen, *A look-ahead Levinson algorithm for indefinite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 490–506.
[12] S. Chandrasekaran and A. Sayed, *A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 107–139.
[13] G. Cybenko, *The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 303–319.
[14] F. de Hoog, *A new algorithm for solving Toeplitz systems of equations*, Linear Algebra Appl., 88/89 (1987), pp. 123–138.
[15] R. W. Freund, *A look-ahead Bareiss algorithm for general Toeplitz matrices*, Numer. Math., 68 (1994), pp. 35–69.
[16] R. W. Freund and H. Zha, *Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems*, Linear Algebra Appl., 188/189 (1993), pp. 255–303.
[17] R. W. Freund and H. Zha, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math., 64 (1993), pp. 295–321.
[18] K. A. Gallivan, S. Thirumalai, and P. Van Dooren, *A look-ahead Schur algorithm*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, UT, 1994, SIAM, Philadelphia, 1994, pp. 450–454.
[19] K. A. Gallivan, S. Thirumalai, P. V. Dooren, and V. Vermaut, *High performance algorithms for Toeplitz and block Toeplitz matrices*, Linear Algebra Appl., 241/242/243 (1996), pp. 343–388.
[20] L. Gemignani, *Schur complements of Bezoutians and the inversion of block Hankel and block Toeplitz matrices*, Linear Algebra Appl., 253 (1997), pp. 39–59.
[21] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.
[22] W. B. Gragg, F. D. Gustavson, D. D. Warner, and D. Y. Y. Yun, *On fast computation of superdiagonal Padé fractions*, Math. Programming Stud., 18 (1982), pp. 39–42.
[23] W. B. Gragg and M. H. Gutknecht, *Stable look-ahead versions of the Euclidean and Chebyshev algorithms*, in Approximation and Computation: A Festschrift in Honor of Walter Gautschi, R. V. M. Zahar, ed., Internet. Ser. Numer. Math. 119, Birkhäuser Boston, Boston, MA, 1994, pp. 231–260.
[24] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, SIAM J. Matrix Anal. Appl., 2 (1997), pp. 279–306.
[25] M. Gutknecht and M. Hochbruck, *Optimized look-ahead recurrences for adjacent rows in the Padé table*, BIT, 36 (1996), pp. 264–286.
[26] M. H. Gutknecht, *Stable row recurrences for the Padé table and a generically superfast look-ahead solver for non-Hermitian Toeplitz systems*, Linear Algebra Appl., 188/189 (1993), pp. 351–421.
[27] M. H. Gutknecht and M. Hochbruck, *Look-ahead Levinson- and Schur-type recurrences in the Padé table*, Electron. Trans. Numer. Anal., 2 (1994), pp. 104–129.
[28] M. H. Gutknecht and M. Hochbruck, *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems*, Numer. Math., 70 (1995), pp. 181–228.
[29] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in Linear Algebra in Signal Processing, IMA Vol. Math. Appl. 69, Springer, New York, 1994, pp. 95–114.
[30] G. Heinig, *Solving Toeplitz systems after extension and transformation*, Calcolo, 33 (1996), pp. 115–129.
[31] G. Heinig, *Transformation approaches for fast and stable solution of Toeplitz systems and polynomial equations*, in Proceedings of the International Workshop "Recent Advances in Applied Mathematics," Department of Mathematics and Computer Science, Kuwait University, State of Kuwait, 1996, pp. 223–238.
[32] G. Heinig and A. Bojanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices:* I. *Transformations*, Linear Algebra Appl., 254 (1997), pp. 193–226.

[33] G. Heinig and A. Bojanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices:* II. *Algorithms*, Linear Algebra Appl., 278 (1998), pp. 11–36.
[34] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Oper. Theory Adv. Appl. 13, Birkhäuser, Basel, 1984.
[35] T. Huckle, *A look-ahead algorithm for solving nonsymmetric linear Toeplitz equations*, in Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, Utah, 1994, SIAM, Philadelphia, PA, 1994, pp. 455–459.
[36] T. Kailath and A. H. Sayed, eds., *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.
[37] P. Kravanja and M. Van Barel, *A fast block Hankel solver based on an inversion formula for block Loewner matrices*, Calcolo, 33 (1996), pp. 147–164.
[38] P. Kravanja and M. Van Barel, *A fast Hankel solver based on an inversion formula for Loewner matrices*, Linear Algebra Appl., 282 (1998), pp. 275–295.
[39] R. Kumar, *A fast algorithm for solving a Toeplitz system of equations*, IEEE Trans. Acoust. Speech Signal Process., 33 (1985), pp. 254–267.
[40] G. Labahn and S. Cabay, *Matrix Padé fractions and their computation*, SIAM J. Comput., 18 (1989), pp. 639–657.
[41] M. Morf, *Fast Algorithms for Multivariable Systems*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, Stanford, CA, 1974.
[42] M. Morf, *Doubling algorithms for Toeplitz and related equations*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Denver, CO, 1980, pp. 954–959.
[43] B. R. Musicus, *Levinson and Fast Cholesky Algorithms for Toeplitz and Almost Toeplitz Matrices*, report, Res. Lab. of Electronics, M.I.T., Cambridge, MA, 1984.
[44] V. Olshevsky, *Pivoting for structured matrices with applications*, Linear Algebra Appl., to appear.
[45] A. H. Sayed and T. Kailath, *A look-ahead block Schur algorithm for Toeplitz-like matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 388–414.
[46] Y. Sugiyama, *An algorithm for solving discrete-time Wiener–Hopf equations based on Euclid's algorithm*, IEEE Trans. Inform. Theory, 32 (1986), pp. 394–409.
[47] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, *A new method for solving key equations for decoding Goppa codes*, Internat. J. Control, 27 (1975), pp. 87–99.
[48] D. R. Sweet and R. P. Brent, *Error analysis of a fast partial pivoting method for structured matrices*, in Advanced Signal Processing Algorithms, T. Luk, ed., Proc. SPIE 2563, Bellingham, WA, 1995, pp. 266–280.
[49] 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.
[50] M. Van Barel and A. Bultheel, *The "look-ahead" philosophy applied to matrix rational interpolation problems*, in Systems and Networks: Mathematical Theory and Applications, Vol. II: Invited and Contributed Papers, U. Helmke, R. Mennicken, and J. Saurer, eds., Mathematical Research 79, Akademie-Verlag, Berlin, 1994, pp. 891–894.
[51] 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.
[52] M. Van Barel and P. Kravanja, *A stabilized superfast solver for indefinite Hankel systems*, Linear Algebra Appl., 284 (1998), pp. 335–355.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 511–516
© 2001 Society for Industrial and Applied Mathematics

ON THE UNITARY ORBIT OF COMPLEX MATRICES*

FAN DING† AND XINGZHI ZHAN†‡

Abstract. Let $M_n(\mathbb{C})$ be the algebra of $n \times n$ complex matrices. Suppose $S \subset \mathbb{C}$ is a measure zero set. It is proved that if $A \in M_n(\mathbb{C})$ is not essentially Hermitian then for almost all unitary matrices $U \in M_n(\mathbb{C})$, each entry of U^*AU is not in S. Some other results of this type are obtained. The ideas here are mainly from differential topology.

Key words. complex matrices, unitary orbit, real analytic map

AMS subject classifications. 15A18, 15A42, 26E05, 32C05

PII. S0895479800381598

1. Introduction. Let $M_n(\mathbf{F})$ be the algebra of $n \times n$ matrices with entries in the field $\mathbf{F}$. To avoid the trivial case throughout the paper we suppose $n \geq 2$. Let I be the identity matrix. For $A \in M_n(\mathbf{F})$ if there exists some $\alpha \in \mathbf{F}$ such that $A = \alpha I$ then we call A a *scalar* matrix. Otherwise, A is *nonscalar*.

In 1968 Gaines [1, Thm. 5] proved that if $\mathbf{F}$ is an infinite field and $A \in M_n(\mathbf{F})$ is nonscalar then there exists a nonsingular $T \in M_n(\mathbf{F})$ so that all the entries of $T^{-1}AT$ are different from zero, while in 1991 Silva [4] showed that the same conclusion is true if the field $\mathbf{F}$ has at least seven elements.

In this paper we shall show that if $\mathbf{F}$ is the field $\mathbb{C}$ of complex numbers, then much stronger results hold. With very few exceptions, the above T can be replaced by almost all unitary matrices and the single point zero by any prescribed measure zero set in $\mathbb{C}$ (Theorem 10). We shall also consider the corresponding problem for real matrices (Theorem 13). The ideas we use to prove these results are mainly from differential topology. For basic concepts in differential topology, see [2].

2. Main results. The following proposition is the basis of our analysis. We will give a proof of it in the appendix.

PROPOSITION 1. *Let M and N be connected real analytic manifolds and $f : M \to N$ a real analytic map. Assume that f has a regular point. If $X \subset N$ has measure zero, so has $f^{-1}(X)$.*

Note that if f has no regular point, then by the Morse–Sard theorem [2], the image of f, $f(M)$, has measure zero.

Corollary 2 follows immediately from Proposition 1.

COROLLARY 2. *Let M be a connected real analytic manifold and $f : M \to \mathbb{R}$ a real analytic map. Assume that f is not a constant map. If $X \subset \mathbb{R}$ has measure zero, so has $f^{-1}(X)$.*

*Received by the editors November 27, 2000; accepted for publication (in revised form) by R. Bhatia March 23, 2001; published electronically October 11, 2001. This work was done while the first author was visiting the Mathematical Institute of Leiden University, part of which was supported by a postdoctoral fellowship from NWO (Netherlands Organization for Scientific Research), and while the second 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/23-2/38159.html

†Institute of Mathematics, Peking University, Beijing 100871, China.

‡Current address: Division of Mathematics, Graduate School of Information Sciences, Tohoku University, Aoba-ku, Sendai 980-8579, Japan (zhan@math.is.tohoku.ac.jp). This author was supported by the National Science Foundation of China under grant 19801004 and Grant-in-Aid for JSPS Fellows 99216.

Suppose that for each point x in a set X, there is given a proposition $P(x)$. We say that for almost all x in X, $P(x)$ is true if there exists a measure zero set E such that for x in $X\backslash E$, $P(x)$ is true. Let $\mathcal{U}(n) \equiv \{U : U^*U = I, U \in M_n(\mathbb{C})\}$ be the unitary group. Then the set $\{U^*AU : U \in \mathcal{U}(n)\}$ is the *unitary orbit* of A. Note that $\mathcal{U}(n)$ is a real analytic submanifold of $GL(n,\mathbb{C})$, the group of invertible $n \times n$ complex matrices [5]. For an $n \times n$ matrix A, we denote its (i,j) entry by $A(i,j)$.

THEOREM 3. *Let S be a measure zero set in $\mathbb{C}$. Suppose $A \in M_n(\mathbb{C})$ is nonscalar. Then for almost all U in $\mathcal{U}(n)$, $(U^*AU)(i,j) \notin S$ for $i,j = 1,\ldots,n, i \neq j$.*

Proof. The conclusion here is that $\cup_{i\neq j}\{U \in \mathcal{U}(n) : U^*AU(i,j) \in S\}$ is a measure zero set, which is equivalent to the statement that for each pair $i \neq j$, $1 \leq i,j \leq n$, $\{U \in \mathcal{U}(n) : U^*AU(i,j) \in S\}$ is a measure zero set. Later in the paper, we will use this fact several times but we will not repeat mentioning it.

Let $f_{ij} : \mathcal{U}(n) \to \mathbb{C}$ be the map given by $f_{ij}(U) = (U^*AU)(i,j)$, $U \in \mathcal{U}(n), i,j = 1,\ldots,n, i \neq j$. Then f_{ij} is real analytic. By Proposition 1, it suffices to show that f_{ij} has a regular point. We only consider the case where $i = 1, j = 2$. The proof for other $i \neq j$ is similar. Denote f_{12} by f.

Consider first the special case where $n = 2$. Denote by I_n the $n \times n$ identity matrix. The tangent space to $\mathcal{U}(2)$ at I_2 is a real vector space $u(2)$ spanned by

$$D_1 = \begin{pmatrix} \sqrt{-1} & 0 \\ 0 & 0 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{-1} \end{pmatrix},$$

$$D_3 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad D_4 = \begin{pmatrix} 0 & \sqrt{-1} \\ \sqrt{-1} & 0 \end{pmatrix}.$$

Consider the differential of f at I_2, $(df)_{I_2}$.

$$(df)_{I_2}(D) = (D^*A + AD)(1,2) = (AD - DA)(1,2) \quad \text{for} \quad D \in u(2).$$

For $U \in \mathcal{U}(2)$, the tangent space to $\mathcal{U}(2)$ at U is $Uu(2) = \{UD : \ D \in u(2)\}$. Let $(df)_U$ be the differential of f at U. For $D \in u(2)$,

$$\begin{aligned}(df)_U(UD) &= [(UD)^*AU + U^*A(UD)](1,2) \\ &= [D^*(U^*AU) + (U^*AU)D](1,2) \\ &= [(U^*AU)D - D(U^*AU)](1,2).\end{aligned}$$

Let $A = (a_{ij})$. Note that $(AD_1 - D_1A)(1,2) = -a_{12}\sqrt{-1}$, $(AD_2 - D_2A)(1,2) = a_{12}\sqrt{-1}$, $(AD_3 - D_3A)(1,2) = a_{11} - a_{22}$, and $(AD_4 - D_4A)(1,2) = (a_{11} - a_{22})\sqrt{-1}$.

Hence, if f has no regular point, then for any $U \in \mathcal{U}(2)$, $(U^*AU)(1,1) = (U^*AU)(2,2)$. If $U = I_2$, then we obtain $a_{11} = a_{22}$. If

$$U = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix},$$

then we obtain $a_{12} + a_{21} = 0$. If

$$U = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\sqrt{-1} \\ \frac{\sqrt{2}}{2}\sqrt{-1} & \frac{\sqrt{2}}{2} \end{pmatrix},$$

then we obtain $a_{12} - a_{21} = 0$. Thus $a_{12} = a_{21} = 0$ and A is scalar, a contradiction.

We now consider the general case. Assume that f has no regular point. Then by the Morse–Sard theorem [2], $f(\mathcal{U}(n))$ has measure zero. Denote

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

by A'. Note that for $U' \in \mathcal{U}(2)$,

$$U = \begin{pmatrix} U' & 0 \\ 0 & I_{n-2} \end{pmatrix} \in \mathcal{U}(n)$$

and $(U^*AU)(1,2) = (U'^*A'U')(1,2)$. Let $f_{A'} : \mathcal{U}(2) \to \mathbb{C}$ be the map given by $f_{A'}(U') = (U'^*A'U')(1,2)$, $U' \in \mathcal{U}(2)$. Then $f_{A'}(\mathcal{U}(2)) \subset f(\mathcal{U}(n))$. Thus $f_{A'}(\mathcal{U}(2))$ has measure zero. So $f_{A'}$ has no regular point. By the above argument, we have $a_{11} = a_{22}$ and $a_{12} = a_{21} = 0$. Similarly, we have $a_{kk} = a_{11}, k \geq 2$ and $a_{kl} = 0, k, l = 1, \ldots, n, k \neq l$. Thus A is scalar, a contradiction. This completes the proof. □

Note that for a measure zero set S in $\mathbb{R}$, $S \times \mathbb{R}$ and $\mathbb{R} \times S$ are measure zero sets in $\mathbb{C}$. Thus we have the following corollary.

COROLLARY 4. *Let S be a measure zero set in $\mathbb{R}$. Suppose A is nonscalar. Then for almost all U in $\mathcal{U}(n)$,* $\mathrm{Re}[(U^*AU)(i,j)] \notin S$, $\mathrm{Im}[(U^*AU)(i,j)] \notin S$ *for* $i, j = 1, \ldots, n, i \neq j$.

LEMMA 5. *Let S be a measure zero set in $\mathbb{R}$ and $A \in M_n(\mathbb{C})$. Suppose $A + A^*$ is nonscalar. Then for almost all U in $\mathcal{U}(n)$,* $\mathrm{Re}[(U^*AU)(i,i)] \notin S$ *for* $i = 1, \ldots, n$.

Proof. Let $f_i : \mathcal{U}(n) \to \mathbb{R}$ be the map given by $f_i(U) = \mathrm{Re}[(U^*AU)(i,i)]$, $U \in \mathcal{U}(n), i = 1, \ldots, n$. By Corollary 2, we need only to show that for each i, f_i is not a constant map. Since $f_i(\mathcal{U}(n)) = f_1(\mathcal{U}(n))$, $i \geq 2$, we need only consider the case $i = 1$.

Assume that for all U in $\mathcal{U}(n)$, $f_1(U) = \alpha$, where α is a fixed real number. Then for each $x \in \mathbb{C}^n$, the real part of $\langle (A - \alpha I_n)x, x\rangle$ is zero, where $\langle , \rangle$ is the standard inner product in $\mathbb{C}^n$. So $\langle [(A - \alpha I_n) + (A - \alpha I_n)^*]x, x\rangle = 0$ for $x \in \mathbb{C}^n$. Hence by the polarization identity for quadratic forms, $(A - \alpha I_n) + (A - \alpha I_n)^* = 0$, $A + A^* = 2\alpha I_n$, a contradiction. This completes the proof. □

Similarly, we have the following.

LEMMA 6. *Let S be a measure zero set in $\mathbb{R}$ and $A \in M_n(\mathbb{C})$. Suppose $A - A^*$ is nonscalar. Then for almost all U in $\mathcal{U}(n)$,* $\mathrm{Im}[(U^*AU)(i,i)] \notin S$ *for* $i = 1, \ldots, n$.

Remark. If $A + A^*$ is scalar, then for all U in $\mathcal{U}(n)$, $\mathrm{Re}[(U^*AU)(i,i)]$ is constant for $i = 1, \ldots, n$. Similarly, if $A - A^*$ is scalar, then for all U in $\mathcal{U}(n)$, $\mathrm{Im}[(U^*AU)(i,i)]$ is constant for $i = 1, \ldots, n$.

Using Corollary 4 and Lemmas 5 and 6, we have the following theorem.

THEOREM 7. *Let S be a measure zero set in $\mathbb{R}$ and $A \in M_n(\mathbb{C})$. Suppose that $A + A^*$ and $A - A^*$ are nonscalar. Then for almost all U in $\mathcal{U}(n)$,* $\mathrm{Re}[(U^*AU)(i,j)] \notin S$, $\mathrm{Im}[(U^*AU)(i,j)] \notin S$ *for* $i, j = 1, \ldots, n$.

It is clear that if A is nonscalar, then either $A + A^*$ is nonscalar or $A - A^*$ is nonscalar. The next theorem follows from Theorem 3 and Lemmas 5 and 6.

THEOREM 8. *Let $S \subset \mathbb{C}$ be a countable set. Suppose $A \in M_n(\mathbb{C})$ is nonscalar. Then for almost all U in $\mathcal{U}(n)$, each entry of U^*AU is not in S.*

An interesting special case of Theorem 8 is $S = S_r \equiv \{z \in \mathbb{C} : \text{both Re}z \text{ and Im}z \text{ are rational numbers}\}$. Note that S_r is dense in $\mathbb{C}$.

We say that $A \in M_n(\mathbb{C})$ is *essentially Hermitian* if A is normal and the eigenvalues of A are collinear in the complex plane.

The numerical range of $A \in M_n(\mathbb{C})$ is defined as

$$F(A) \equiv \{x^*Ax : ||x|| = 1,\ x \in \mathbb{C}^n\},$$

where $||\cdot||$ is the usual Euclidean norm. We remark that A is essentially Hermitian if and only if $F(A)$ is a line segment in $\mathbb{C}$. In fact, since the numerical range of a normal matrix is the convex hull of its spectrum [3, p. 11], if A is essentially Hermitian then obviously $F(A)$ is a line segment. Conversely, suppose $F(A)$ is a line segment. Note that all the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A are in $F(A)$. Let $-t$ be the angle between the line segment $F(A)$ and the real axis. Then the set $\{e^{it}(z - \lambda_1) : \ z \in F(A)\}$ is a subset of the real axis, or equivalently, $F(e^{it}(A - \lambda_1 I_n)) \in \mathbb{R}$. Thus $\phi(x) \equiv \langle e^{it}(A - \lambda_1 I_n)x, x\rangle \in \mathbb{R}$ for any $x \in \mathbb{C}^n$. By the polarization identity for quadratic forms, $e^{it}(A - \lambda_1 I_n)$ is Hermitian. Hence A is normal with all eigenvalues collinear, i.e., A is essentially Hermitian.

LEMMA 9. *Let S be a measure zero set in $\mathbb{C}$. Suppose $A \in M_n(\mathbb{C})$ is not essentially Hermitian. Then for almost all U in $\mathcal{U}(n)$, $(U^*AU)(i,i) \notin S$ for $i = 1, \ldots, n$.*

Remark. Let $f_i : \mathcal{U}(n) \to \mathbb{C}$ be the map given by $f_i(U) = (U^*AU)(i,i)$, $U \in \mathcal{U}(n), i = 1, \ldots, n$. Let $f : S^{2n-1} \to \mathbb{C}$ be the map given by $f(x) = x^*Ax,\ x \in S^{2n-1}$, where $S^{2n-1} = \{x \in \mathbb{C}^n : ||x|| = 1\}$. Then $f_i(\mathcal{U}(n)) = f(S^{2n-1}) = F(A)$, the numerical range of A.

Proof of Lemma 9. By Proposition 1, we need only to show that f_i has a regular point. We only consider the case $i = 1$. The proof for other cases is similar. Denote f_1 by g.

Assume that g has no regular point. Then by the Morse–Sard theorem [2], $g(\mathcal{U}(n))$ has measure zero. By the Toeplitz–Hausdorff theorem [3], $g(\mathcal{U}(n)) = F(A)$ is a convex set. So $g(\mathcal{U}(n))$ is a line segment in $\mathbb{C}$. But this is the case if and only if A is essentially Hermitian, which contradicts the hypothesis. This completes the proof. □

Using Theorem 3 and Lemma 9 we have the following theorem.

THEOREM 10. *Let S be a measure zero set in $\mathbb{C}$. If $A \in M_n(\mathbb{C})$ is not essentially Hermitian, then for almost all U in $\mathcal{U}(n)$, each entry of U^*AU is not in S.*

Next we consider real matrices. A^T denotes the transpose of A. Let $SO(n)$ denote the group of $n \times n$ real orthogonal matrices of determinant 1. Note that $SO(n)$ is a real analytic submanifold of $GL(n, \mathbb{R})$ [5].

LEMMA 11. *Let S be a measure zero set in $\mathbb{R}$ and $A \in M_n(\mathbb{R})$. If $n \geq 3$, then suppose A is nonscalar. If $n = 2$, then suppose $A + A^T$ is nonscalar. Then for almost all Q in $SO(n)$, $(Q^TAQ)(i,j) \notin S$ for $i, j = 1, \ldots, n, i \neq j$.*

Proof. Let $f_{ij} : SO(n) \to \mathbb{R}$ be the map given by $f_{ij}(Q) = (Q^TAQ)(i,j)$, $Q \in SO(n), i, j = 1, \ldots, n$. By Corollary 2, we need only to show that for $i \neq j$, f_{ij} is not a constant map. We only consider the case where $i = 2, j = 1$. The proof for other cases is similar.

First assume $n \geq 3$. Since A is nonscalar, there is a nonzero vector $y_1 \in \mathbb{R}^n$ which is not an eigenvector of A and $||y_1|| = 1$. Let $y_1, y_2, \ldots, y_n$ be an orthonormal basis of $\mathbb{R}^n$ such that the matrix $Q = (y_1\ y_2\ \ldots\ y_n)$ is in $SO(n)$. Assume that $Ay_1 = a_1y_1 + a_2y_2 + \cdots + a_ny_n$. Since y_1 is not an eigenvector, $a_i \neq 0$ for some $i > 1$. Without loss of generality, we assume that $a_2 \neq 0$. Then $(Q^TAQ)(2,1) = a_2 \neq 0$. Set $Q_1 = (y_1\ -y_2\ -y_3\ y_4\ \ldots\ y_n)$. Then $Q_1 \in SO(n)$ and $(Q_1^TAQ_1)(2,1) = -a_2$. So f_{21} is not a constant map.

Now assume that $n = 2$ and f_{21} is a constant map. For any $Q \in SO(2)$, $f_{21}(Q) = \alpha$, where α is a fixed real number. Denote $J = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. Let $x \in \mathbb{R}^2$ be a unit vector.

Set $Q = (x\ Jx)$. Then $Q \in SO(2)$ and $f_{21}(Q) = \langle x, A^T Jx\rangle = \alpha$ for any unit $x \in \mathbb{R}^2$, where $\langle , \rangle$ is the standard inner product in $\mathbb{R}^2$. From the above equality we get $\langle x, J^T Ax\rangle = \alpha$. Adding them we obtain $\langle x, (A^T J + J^T A - 2\alpha I_2)x\rangle = 0$ for any $x \in \mathbb{R}^2$, which implies $A^T J + J^T A - 2\alpha I_2 = 0$. Consequently $A + A^T$ is scalar, a contradiction. So f_{21} is not a constant map. This completes the proof. □

LEMMA 12. *Let S be a measure zero set in $\mathbb{R}$, $A \in M_n(\mathbb{R})$. Suppose $A + A^T$ is nonscalar. Then for almost all Q in $SO(n)$, $(Q^T AQ)(i, i) \notin S$ for $i = 1, \ldots, n$.*

The proof is similar to that of Lemma 5. We omit it.

Combining Lemmas 11 and 12, we have the following theorem.

THEOREM 13. *Let S be a measure zero set in $\mathbb{R}$, $A \in M_n(\mathbb{R})$. Suppose $A + A^T$ is nonscalar. Then for almost all Q in $SO(n)$, each entry of $Q^T AQ$ is not in S.*

COROLLARY 14. *Suppose $A \in M_n(\mathbb{R})$ is nonscalar and $A + A^T \neq 0$. Then for almost all Q in $SO(n)$, each entry of $Q^T AQ$ is not zero.*

Proof. If $A + A^T$ is nonscalar, then use Theorem 13. If $n \geq 3$, A is nonscalar and $A + A^T = kI_n$, where $k \neq 0$, then for any Q in $SO(n)$, we have $(Q^T AQ)(i, i) = \frac{k}{2} \neq 0, i = 1, \ldots, n$. Now use Lemma 11. If $n = 2$, A is nonscalar and $A + A^T = kI_2$, where $k \neq 0$, then for any $Q \in SO(2)$, $Q^T AQ = A$ and $a_{ij} \neq 0$ for $i, j = 1, 2$. This completes the proof. □

Appendix. Proof of Proposition 1. We begin with the following.

LEMMA 15. *Let $U \subset \mathbb{R}^m$ be a connected open set and $f : U \to \mathbb{R}$ a C^∞ map. Assume that f has an nth order partial derivative which is nowhere zero in U. Then $f^{-1}(0)$ has measure zero.*

Proof. We proceed by induction on n. If $n = 0$, then $f^{-1}(0)$ is empty. If $n = 1$ then the assertion follows from the implicit function theorem.

Assume now that the assertion is correct for $n - 1$. Denote $\frac{\partial f}{\partial x_i}$ by f_i, $i = 1, \ldots, m$. Then there exists an i such that an $(n - 1)$st order partial derivative of f_i is nowhere zero in U. Assume that $i = 1$. Note that the set of critical points of f, $\Sigma_f = f_1^{-1}(0) \cap f_2^{-1}(0) \cap \cdots \cap f_m^{-1}(0)$. By induction, $f_1^{-1}(0)$ has measure zero. So Σ_f has measure zero. Set $U' = U \backslash \Sigma_f$. Let g denote $f|U'$, the restriction of f to U'. Then by the implicit function theorem, $g^{-1}(0)$ has measure zero. Note that $f^{-1}(0) \subset g^{-1}(0) \cup \Sigma_f$. So $f^{-1}(0)$ has measure zero. This completes the proof. □

LEMMA 16. *Let $U \subset \mathbb{R}^m$ be a connected open set and $f : U \to \mathbb{R}$ a real analytic map. Suppose f is not the zero map. Then $f^{-1}(0) \subset U$ has measure zero.*

Proof. Let $x \in U$. Since f is real analytic and f is not the zero map, there exists an open connected neighborhood U_x of x such that f has a partial derivative which is nowhere zero in U_x. So by Lemma 15, $f^{-1}(0) \cap U_x$ has measure zero. Note that U satisfies the second axiom of countability. So $f^{-1}(0)$ has measure zero. This completes the proof. □

LEMMA 17. *Let M and N be connected real analytic manifolds and $f : M \to N$ a real analytic map. Assume that f has a regular point. Then the set of critical points of f, Σ_f, is a measure zero subset of M.*

Proof. First assume that M is an open subset of $\mathbb{R}^m$ and N is an open subset of $\mathbb{R}^n$.

Since f has a regular point, we can choose a linear map $g : \mathbb{R}^m \to \mathbb{R}^{m-n}$ such that the map $h : M \to \mathbb{R}^m$ given by $h(x) = (f(x), g(x)), x \in M$ has a regular point. Let h' be the determinant of the Jacobian of h. Then $x \in M$ is a regular point of h if and only if $h'(x) \neq 0$. So by Lemma 16, Σ_h has measure zero. Note that $\Sigma_f \subset \Sigma_h$. Thus Σ_f has measure zero.

Now let M and N be real analytic manifolds. We say that a connected open subset U of M is admissible if U is a coordinate patch of M and $f(U)$ is a subset of a coordinate patch of N. Assume that x_0 is a regular point of f. Then by the above argument, x_0 has an admissible neighborhood U_0 such that $\Sigma_f \cap U_0$ has measure zero. Note that if U, V are admissible, $U \cap V \neq \emptyset$, and f has a regular point in U, then $\Sigma_f \cap U$ has measure zero and f has a regular point in V. Let $x \in M$. Since M is connected, we have admissible subsets $U_0, U_1, U_2, \ldots, U_k$ of M such that $x \in U_k$ and for each $i = 0, \ldots, k-1$, $U_i \cap U_{i+1} \neq \emptyset$. So for each $i = 1, \ldots, k$, f has a regular point in U_i. Thus $\Sigma_f \cap U_k$ has measure zero. Note that M satisfies the second axiom of countability. Thus Σ_f has measure zero. This completes the proof of the lemma. □

Proof of Proposition 1. Let $\tilde{M} = M \backslash \Sigma_f$. Let g denote $f|\tilde{M}$, the restriction of f to $\tilde{M}$. Note that $\tilde{M}$ is open. We prove that $g^{-1}(X)$ has measure zero. Since $\tilde{M}$ satisfies the second axiom of countability, it suffices to prove this with $\tilde{M}$ an open subset of $\mathbb{R}^m$ and $N = \mathbb{R}^n$. By the implicit function theorem, every point $p \in \tilde{M}$ has an open neighborhood $U \subset \tilde{M}$ such that there are an open set $V \subset \mathbb{R}^m$ and a C^∞ diffeomorphism $\varphi : V \to U$ satisfying

$$(g\varphi)(x_1, \ldots, x_m) = (x_1, \ldots, x_n).$$

Thus $(g\varphi)^{-1}(X) \subset X \times \mathbb{R}^{m-n}$. Since $X \subset \mathbb{R}^n$ has measure zero, $X \times \mathbb{R}^{m-n} \subset \mathbb{R}^m$ has measure zero. Therefore, $(g\varphi)^{-1}(X)$ has measure zero. So $g^{-1}(X) \cap U$ has measure zero. Since $\tilde{M}$ satisfies the second axiom of countability, $g^{-1}(X)$ has measure zero.

Note that $f^{-1}(X) \subset g^{-1}(X) \cup \Sigma_f$. By Lemma 17, $f^{-1}(X)$ has measure zero. This completes the proof of Proposition 1. □

Acknowledgments. We thank Leiden University, NWO, and JSPS for their support.

REFERENCES

[1] F. Gaines, *Kato-Taussky-Wielandt commutator relations*, Linear Algebra Appl., 1 (1968), pp. 127–138.
[2] M. W. Hirsch, *Differential Topology*, Springer-Verlag, New York, Heidelberg, 1976.
[3] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
[4] F. C. Silva, *Matrices with nonzero entries*, Linear Algebra Appl., 146 (1991), pp. 111–119.
[5] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*, Springer-Verlag, New York, 1984.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 517–523
© 2001 Society for Industrial and Applied Mathematics

THE MULTIPLICATIVE INVERSE EIGENVALUE PROBLEM OVER AN ALGEBRAICALLY CLOSED FIELD*

JOACHIM ROSENTHAL† AND XIAOCHANG WANG‡

Abstract. Let M be an $n \times n$ square matrix and let $p(\lambda)$ be a monic polynomial of degree n. Let $\mathcal{Z}$ be a set of $n \times n$ matrices. The multiplicative inverse eigenvalue problem asks for the construction of a matrix $Z \in \mathcal{Z}$ such that the product matrix MZ has characteristic polynomial $p(\lambda)$.

In this paper we provide new necessary and sufficient conditions when $\mathcal{Z}$ is an affine variety over an algebraically closed field.

Key words. eigenvalue completion, inverse eigenvalue problems, dominant morphism theorem

AMS subject classifications. 15A29, 47A75, 14N05, 93B60

PII. S0895479800378192

1. Introduction.

Inverse eigenvalue problems involving partially specified matrices have drawn the attention of many researchers. The problems are of significance both from a theoretical point of view and from an applications point of view. For background material we refer to the monograph by Gohberg, Kaashoek, and van Schagen [8], the recent book by Xu [15], and the survey article by Chu [3].

The multiplicative eigenvalue problem asks for conditions which guarantee that the spectrum of a certain matrix M can be made arbitrarily through premultiplication by a matrix from a certain set. To be precise, let $\mathbb{F}$ be an arbitrary field. Let $Mat_{n\times n}$ be the space of all $n \times n$ matrices defined over the field $\mathbb{F}$. We will identify $Mat_{n\times n}$ with the vector space $\mathbb{F}^{n^2}$. Let $\mathcal{Z} \subset Mat_{n\times n}$ be an arbitrary subset and let $M \in Mat_{n\times n}$ be a fixed matrix. Then the (right) multiplicative inverse eigenvalue problem in its general form asks the following.

PROBLEM 1.1. *Given a monic polynomial $p(\lambda)$ of degree n, is there an $n \times n$ matrix $Z \in \mathcal{Z}$ such that MZ has characteristic polynomial*

$$\det(\lambda I - MZ) = p(\lambda)?$$

The formulation of the left multiplicative inverse eigenvalue problem is analogous, seeking a matrix $Z \in \mathcal{Z}$ such that ZM has characteristic polynomial $p(\lambda)$. The left and the right multiplicative inverse eigenvalue problems are equivalent to each other because of the identity

$$\det(\lambda I - ZA) = \det(\lambda I - A^t Z^t).$$

In its general form Problem 1.1 is an "open end problem" and until this point only very particular situations are well understood; e.g., we would like to mention the well-known result by Friedland [7], who considered the set $\mathcal{Z} = \mathcal{D}$ of diagonal matrices. Friedland did show in this case by topological methods that Problem 1.1

*Received by the editors September 15, 2000; accepted for publication (in revised form) by M. Chu March 21, 2001; published electronically October 11, 2001.

http://www.siam.org/journals/simax/23-2/37819.html

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-5683 (Rosenthal.1@nd.edu). This author was supported in part by NSF grants DMS-96-10389 and DMS-00-72383.

‡Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042 (mdxia@ttacs.ttu.edu).

has an affirmative answer if the base field $\mathbb{F}$ consists of the complex numbers $\mathbb{C}$. This diagonal perturbation result was later generalized by Dias da Silva [5] to situations where the base field can be any algebraically closed field.

The result which we are going to derive in this paper can be viewed as a large generalization of Friedland's result. Specifically we will deal with the situation where $\mathcal{Z} \subset Mat_{n\times n}$ represents an arbitrary affine variety over an arbitrary algebraically closed field $\mathbb{F}$. Under these assumptions we will derive necessary and sufficient conditions (Theorem 3.1) which will guarantee that Problem 1.1 has a positive answer for a "generic set" of matrices M and a "generic set" of monic polynomials $p(\lambda)$ of degree n.

The techniques which we use in this paper have been developed by the authors in the context of the additive inverse eigenvalue problem [2, 10, 13] and in the context of the pole placement problem [12].

The major tool from algebraic geometry which we will use is the "dominant morphism theorem" (see Theorem 2.1). This powerful theorem necessitates that the base field is algebraically closed. The situation over a nonalgebraically closed field seems to be much more complicated. Some new techniques applicable over the real numbers have been recently reported by Drew et al. [6].

2. Preliminaries. For the convenience of the reader we provide a summary of results which will be needed to establish the new results of this paper.

Denote by $\sigma_i(M)$ the ith elementary symmetric function in the eigenvalues of M, i.e., $\sigma_i(M)$ denotes up to sign the ith coefficient of the characteristic polynomial of M. Crucial for our purposes will be the *eigenvalue assignment map*

$$\psi : \mathcal{Z} \longrightarrow \mathbb{F}^n, \quad Z \longmapsto (-\sigma_1(MZ), \ldots, (-1)^n\sigma_n(MZ)). \tag{2.1}$$

ψ is a morphism in the sense of algebraic geometry. By identifying a monic polynomial $\lambda^n + b_1\lambda^{n-1} + \cdots + b_n$ with the point $(b_1, \ldots b_n) \in \mathbb{F}^n$ we can also write

$$\psi(Z) = \det(\lambda I - MZ). \tag{2.2}$$

Crucial for the proof of the main result (Theorem 3.1) will be the dominant morphism theorem. The following version can be immediately deduced from [1, Chapter AG, section 17, Theorem 17.3].

PROPOSITION 2.1. *Let $f : \mathcal{Z} \to \mathcal{Y}$ be a morphism of affine varieties over an algebraically closed field. Then the image of f contains a nonempty Zariski open set of $\mathcal{Y}$ if and only if the Jacobian $df_Z : T_{Z,\mathcal{Z}} \to T_{f(Z),\mathcal{Y}}$ is onto at some smooth point Z of $\mathcal{Z}$, where $T_{X,\mathcal{X}}$ is the tangent space of $\mathcal{X}$ at the point X.*

There are classical formulas, sometimes referred to as Newton formulas, which express the elementary symmetric functions $\sigma_i(M)$ uniquely as a polynomial in the power sum symmetric functions

$$p_i := \lambda_1^i + \cdots + \lambda_n^i = \operatorname{tr}(M)^i.$$

To be precise one has the formula (see, e.g., [11])

$$\sigma_i(M) = \frac{1}{n!}\begin{pmatrix} p_1 & 1 & 0 & \ldots & 0 \\ p_2 & p_1 & 2 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & p_1 & n-1 \\ p_n & \ldots & \ldots & p_2 & p_1 \end{pmatrix},$$

which induces an isomorphism $\mathbb{F}^n \to \mathbb{F}^n, (p_1, \ldots, p_n) \mapsto (\sigma_1, \ldots, \sigma_n)$. Based on this we can equally well study the map

$$\phi : \mathcal{Z} \longrightarrow \mathbb{F}^n, \quad M \longmapsto (\mathrm{tr}(MZ), \ldots, \mathrm{tr}((MZ)^n)) . \tag{2.3}$$

We will use the following result from [10].

PROPOSITION 2.2. *Let $\mathcal{L} \subset Mat_{n\times n}$ be a linear subspace of dimension $\geq n$, $\mathcal{L} \not\subset sl_n$ (i.e., $\mathcal{L}$ contains an element with nonzero trace). Define*

$$\pi(M) = (m_{11}, m_{22}, \ldots, m_{nn})$$

the projection onto the diagonal entries. Then there exists a $S \in Gl_n$ such that

$$\pi(S\mathcal{L}S^{-1}) = \mathbb{F}^n.$$

It is possible to "compactify" the problem. For this, consider the identity

$$\det(\lambda I - MZ) = \det \begin{bmatrix} I & Z \\ M & \lambda I \end{bmatrix}. \tag{2.4}$$

Denote by $\mathrm{Grass}\,(k, n)$ the Grassmann manifold consisting of all k-dimensional linear subspaces of $\mathbb{F}^n$. Algebraically, $\mathrm{Grass}\,(k, n)$ has the structure of a smooth projective variety. In what follows we will identify $\mathrm{rowsp}[I\ Z]$ with a point in the Grassmannian $\mathrm{Grass}\,(n, 2n)$. By identifying $\mathrm{rowsp}[I\ Z]$ with $Z \in Mat_{n\times n}$, we can say that $\mathcal{Z} \subset \mathrm{Grass}\,(n, 2n)$. Let $\bar{\mathcal{Z}}$ be the projective closure of $\mathcal{Z}$ in $\mathrm{Grass}\,(n, 2n)$. Every element in $\bar{\mathcal{Z}}$ can be represented simply by a subspace of the form $\mathrm{rowsp}[Z_1\ Z_2]$, where the $n \times n$ matrix Z_1 is not necessarily invertible. $\mathrm{rowsp}[Z_1\ Z_2]$ describes an element of $\mathcal{Z}$ if and only if Z_1 is invertible. For any element $\mathrm{rowsp}[Z_1\ Z_2] \in \bar{\mathcal{Z}}$, define $\bar{\psi} : \bar{\mathcal{Z}} \longrightarrow \mathbb{P}^n$

$$\bar{\psi}([Z_1\ Z_2]) = \det \begin{bmatrix} Z_1 & Z_2 \\ M & \lambda I \end{bmatrix}, \tag{2.5}$$

where a polynomial $b_0\lambda^n + b_1\lambda^{n-1} + \cdots + b_n$ is identified with the point $(b_0, b_1, \ldots, b_n) \in \mathbb{P}^n$. Recall that the Plücker coordinates of $\mathrm{rowsp}[Z_1\ Z_2] \in \mathrm{Grass}\,(n, 2n)$ are given by the full size minors $[Z_1\ Z_2]$, and by considering the Plücker coordinates as the homogeneous coordinates of points in $\mathbb{P}^N$, $N = \binom{2n}{n} - 1$, one has an embedding $\mathrm{Grass}\,(n, 2n) \subset \mathbb{P}^N$ which is called Plücker embedding. Under the Plücker coordinates, (2.5) becomes

$$\bar{\psi}([Z_1\ Z_2]) = \det \begin{bmatrix} Z_1 & Z_2 \\ M & \lambda I \end{bmatrix} = \sum_{i=0}^{N} z_i m_i(\lambda), \tag{2.6}$$

where $\{z_i\}$ are $n \times n$ minors of $[Z_1\ Z_2]$ and $m_i(\lambda)$ is the cofactor of the z_i in the determinate of (2.5). $\bar{\psi}$ is undefined on the elements where

$$\det \begin{bmatrix} Z_1 & Z_2 \\ M & \lambda I \end{bmatrix} = 0.$$

So $\bar{\psi}$ is a rational map.

3. New results. The next theorem constitutes the main result of this paper. As stated in the introduction we will identify the set $Mat_{n\times n}$ with the vector space $\mathbb{F}^{n^2}$ and we will identify the set of monic polynomials of degree n

$$\lambda^n + b_1\lambda^{n-1} + \cdots + b_n$$

with the vector space $\mathbb{F}^n$. If V is an arbitrary $\mathbb{F}$-vector space, one says that $U \subset V$ forms a generic set if U contains a nonempty Zariski open subset. Over the complex or real numbers a generic set is necessarily dense with respect to the natural topology. The dominant morphism theorem, Theorem 2.1, states that the image of an algebraic morphism forms a generic set as soon as the linearization around a smooth point is surjective and if the field is algebraically closed.

If Problem 1.1 has a positive answer for a generic set of matrices inside $Mat_{n\times n}$ and a generic set of monic polynomials, then we will say that Problem 1.1 is generically solvable. With this preliminary we have the main result of this paper.

THEOREM 3.1. *Let $\mathcal{Z} \subset Mat_{n\times n}$ be an affine variety over an algebraically closed field $\mathbb{F}$. Then Problem* 1.1 *is generically solvable if and only if* $\dim \mathcal{Z} \geq n$ *and* $\det(Z)$ *is not a constant function on $\mathcal{Z}$.*

Proof. The conditions are obviously necessary. So we only need to prove the sufficiency. Assume that $\dim \mathcal{Z} \geq n$ and $\det(Z)$ is not a constant on $\mathcal{Z}$. Then there exists a curve $Z(t) \subset \mathcal{Z}$ such that

$$\frac{d}{dt}\det Z(t)|_{t=0} \neq 0,$$

$Z(0) = Z_0$ is a smooth point of $\mathcal{Z}$, and $\det Z_0 \neq 0$.

Let $Z(t) = Z_0 + tL + O(t^2)$ where $L \in T_{Z_0,\mathcal{Z}}$. Then

$$\det Z(t) = \det Z_0 \det(I + tZ_0^{-1}L + O(t^2)) = \det Z_0(1 + t\mathrm{tr}Z_0^{-1}L + O(t^2))$$

and

$$\frac{d}{dt}\det Z(t)|_{t=0} = \det Z_0 \mathrm{tr}Z_0^{-1}L \neq 0,$$

i.e.,

$$Z_0^{-1}T_{Z_0,\mathcal{Z}} \not\subset sl_n.$$

By Proposition 2.2, there exists an $S \in Gl_n$ such that

$$\pi(SZ_0^{-1}T_{Z_0,\mathcal{Z}}S^{-1}) = \mathbb{F}^n.$$

Let

$$D := \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & n \end{bmatrix} \tag{3.1}$$

and

$$M := S^{-1}DSZ_0^{-1}.$$

Then for any curve through Z_0

$$Z(t) = Z_0 + tL + O(t^2) \subset \mathcal{Z}, \quad L \in T_{Z_0,\mathcal{Z}},$$

we have

$$\begin{aligned}\lim_{t\to 0} \frac{\operatorname{tr}(MZ(t))^i - \operatorname{tr}(MZ_0)^i}{t} &= \lim_{t\to 0} \frac{\operatorname{tr}(MZ_0 + tML + O(t^2))^i - \operatorname{tr}(MZ_0)^i}{t} \\ &= i \cdot \operatorname{tr}((MZ_0)^{i-1}ML) \\ &= i \cdot \operatorname{tr}(D^i S Z_0^{-1} L S^{-1}).\end{aligned}$$

Let

$$V = D \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & n & \cdots & n^{n-1} \end{bmatrix} D. \tag{3.2}$$

Then V is invertible and the Jacobian $d\phi_{Z_0} : T_{Z_0,\mathcal{Z}} \mapsto \mathbb{F}^n$

$$\begin{aligned}d\phi_{Z_0}(L) &= (\operatorname{tr}(DSZ_0^{-1}LS^{-1}), 2\operatorname{tr}(D^2SZ_0^{-1}LS^{-1}), \ldots, n\operatorname{tr}(D^nSZ_0^{-1}LS^{-1})) \\ &= \pi(SZ_0^{-1}LS^{-1})V\end{aligned}$$

is onto. By the dominant morphism theorem, Theorem 2.1, $\phi(\mathcal{Z})$ contains a nonempty Zariski open set of $\mathbb{F}^n$, so does $\psi(\mathcal{Z})$.

Since the set of M's such that ψ is almost onto is a Zariski open set, and we just showed that it is nonempty, ψ is almost onto for a generic set of matrices M. □

Theorem 3.1 says that if $\dim \mathcal{Z} \geq n$ and $\det(Z)$ is not a constant function on $\mathcal{Z}$, then there is a nonempty Zariski open set of $n \times n$ matrices such that for any M in this set, the multiplicative inverse eigenvalue problem is solvable for a Zariski open set of characteristic polynomials. From the proof of Theorem 3.1 we can get a description of such a Zariski open set of matrices.

COROLLARY 3.2. *Let $\mathcal{Z}$ be an affine variety of dimension at least n such that $\det(Z)$ is not a constant function on $\mathcal{Z}$. Pick a smooth point $Z_0 \in \mathcal{Z}$ such that $\det Z_0 \neq 0$, and let $\mathcal{E}$ be the nonempty Zariski open set of Gl_n defined by*

$$\mathcal{E} = \{R \in Gl_n \mid \pi(R^{-1}Z_0^{-1}T_{Z_0,\mathcal{Z}}S) = \mathbb{F}^n\}.$$

Then for every $M \in Gl_n$ such that MZ_0 has n distinct eigenvalues with the associated right eigenvectors $[\alpha_1, \ldots, \alpha_n] \in \mathcal{E}$, the multiplicative inverse eigenvalue problem is solvable for a nonempty Zariski open set of characteristic polynomials.

Next we consider the number of solutions of Problem 1.1 when $\dim \mathcal{Z} = n$. For this we introduce an important technical concept.

DEFINITION 3.3. *A matrix M is called $\mathcal{Z}$-nondegenerate for the right multiplicative inverse eigenvalue problem if*

$$\det \begin{bmatrix} Z_1 & Z_2 \\ M & \lambda I \end{bmatrix} \neq 0 \tag{3.3}$$

for any $\operatorname{rowsp}[Z_1, Z_2] \in \bar{\mathcal{Z}} \subset \operatorname{Grass}(n, 2n)$.

Thus if M is $\mathcal{Z}$-nondegenerate, then the map $\bar{\psi}$ defined by (2.5) becomes a morphism. In this situation we can say even quite a bit more.

THEOREM 3.4. *If M is $\mathcal{Z}$-nondegenerate and* $\dim \mathcal{Z} = n$, *then Problem* 1.1 *is solvable for any monic polynomial $p(\lambda)$ of degree n. Moreover, when counted with multiplicities, the number of matrices inside $\mathcal{Z}$ which results in a characteristic polynomial $p(\lambda)$ is exactly equal to the degree of the projective variety $\bar{\mathcal{Z}} \subset \mathrm{Grass}\,(n, 2n)$ when viewed under the Plücker embedding* $\mathrm{Grass}\,(n, 2n) \subset \mathbb{P}^N$.

Proof. We will repeatedly use the projective dimension theorem [9, Charter I, Theorem 7.2] which says that if X and Y are r-dimensional and s-codimensional projective varieties, respectively, then $\dim X \cap Y \geq r - s$. In particular, $X \cap Y$ is not empty if $r \geq s$.

Let

$$K = \left\{ (z_0, \ldots, z_{_N}) \in \mathbb{P}^N | \sum_{i=0}^{N} z_i m_i(\lambda) = 0 \right\}.$$

Then K must have codimension $n+1$ because of the condition $K \cap \mathcal{Z} = \emptyset$. Therefore the linear equation

$$\sum_{i=0}^{N} z_i m_i(\lambda) = p(\lambda) \tag{3.4}$$

has solutions in $\mathbb{P}^N$ for any $p(\lambda) \in \mathbb{P}^n$, and the set of all solutions for each $p(\lambda)$ is in the form of $z_p + K$ where z_p is a particular solution; i.e., the solution set is given by $K_p - K$, where K_p is the unique n-codimensional projective subspace through z_p and K. Since $K \cap \bar{\mathcal{Z}} = \emptyset$, we must have

$$\dim K_p \cap \bar{\mathcal{Z}} = 0,$$

and by Bézout's theorem [14], there are $\deg \bar{\mathcal{Z}}$ many points in $K_p \cap \bar{\mathcal{Z}}$ counted with multiplicities. If $p(\lambda)$ is a monic polynomial of degree n, then from (2.5) one can see that all the solutions are in $\mathcal{Z}$. □

An immediate application of Theorem 3.4 is a result of Friedland [7]: Let $\mathcal{Z}$ be the set of all diagonal matrices. Then closure $\bar{\mathcal{Z}}$ of $\mathcal{Z}$ inside the Grassmann variety $\mathrm{Grass}\,(n, 2n)$ is isomorphic to the product of n projective lines:

$$\mathbb{P}^1 \times \cdots \times \mathbb{P}^1.$$

As shown in [2] the degree of $\bar{\mathcal{Z}}$ is then equal to $n!$. Moreover all points of $\bar{\mathcal{Z}}$ are of the form $\mathrm{rowsp}[Z_1\ Z_2]$ where Z_1 and Z_2 are given by

$$Z_1 = \begin{bmatrix} z_{11} & 0 & \cdots & 0 \\ 0 & z_{12} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{1n} \end{bmatrix}, \qquad Z_2 = \begin{bmatrix} z_{21} & 0 & \cdots & 0 \\ 0 & z_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{2n} \end{bmatrix}.$$

In these matrices, (z_{1i}, z_{2i}) represent the homogeneous coordinates of the ith projective line $\mathbb{P}^1$.

In order to apply Theorem 3.4 we have to find the algebraic conditions which guarantee that a particular matrix M is $\mathcal{Z}$-nondegenerate, i.e., condition (3.3) has to be satisfied for every element $[Z_1\ Z_2] \in \bar{\mathcal{Z}}$. For this let I be a subset of $\{1, 2, \ldots, n\}$,

J be the complement of I, and $|J|$ be the number of elements in J. For any point $[Z_1\ Z_2] \in \bar{\mathcal{Z}}$, assume

$$\begin{aligned} z_{1i} &= 0 \quad \text{for } i \in I, \\ z_{1j} &\neq 0 \quad \text{for } j \in J. \end{aligned}$$

Without loss of generality we can take

$$\begin{aligned} z_{2i} &= 1 \quad \text{for } i \in I, \\ z_{1j} &= 1 \quad \text{for } j \in J, \end{aligned}$$

and (2.5) becomes

$$\bar{\psi}([Z_1\ Z_2]) = \pm M_{_I} \lambda^{|J|} + \text{lower power terms},$$

where $M_{_I}$ is the principal minor of M consisting of the ith rows and columns, $i \in I$. Furthermore if we take

$$z_{2j} = 0 \quad \text{for } j \in J,$$

then

$$\bar{\psi}([Z_1\ Z_2]) = \pm M_{_I} \lambda^{|J|}.$$

Therefore M is $\mathcal{Z}$-nondegenerate if and only if all the principal minors of M are nonzero. Thus we have Friedland's result [7, Theorem 2.3] formulated for an algebraically closed field: If all the principal minors of M are nonzero, then the multiplicative inverse eigenvalue problem with perturbation from the set of diagonal matrices is solvable for any monic polynomial $p(\lambda)$ of degree n, and there are $n!$ solutions, when counted with multiplicities.

REFERENCES

[1] A. Borel, *Linear Algebraic Groups*, 2nd enlarged ed., Grad. Texts in Math. 126, Springer-Verlag, New York, 1991.

[2] C. I. Byrnes and X. Wang, *The additive inverse eigenvalue problem for Lie perturbations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp 113–117.

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

[4] G. N. de Oliveira, *On the multiplicative inverse eigenvalue problem*, Canad. Math. Bull., 15 (1972), pp. 189–193.

[5] J. A. Dias da Silva, *On the multiplicative inverse eigenvalue problem*, Linear Algebra Appl., 78 (1986), pp. 133–145.

[6] J. H. Drew, C. R. Johnson, D. D. Olesky, and P. van den Driessche, *Spectrally arbitrary patterns*, Linear Algebra Appl., 308 (2000), pp. 121–137.

[7] S. Friedland, *Inverse eigenvalue problems*, Linear Algebra Appl., 17 (1977), pp. 15–51.

[8] I. Gohberg, M. A. Kaashoek, and F. van Schagen, *Partially Specified Matrices and Operators: Classification, Completion, Applications*, Birkhäuser, Boston, Basel, Berlin, 1995.

[9] R. Hartshorne, *Algebraic Geometry,* Springer-Verlag, New York, 1977.

[10] W. Helton, J. Rosenthal, and X. Wang, *Matrix extensions and eigenvalue completions, the generic case*, Trans. Amer. Math. Soc., 349 (1997), pp. 3401–3408.

[11] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, Oxford University Press, Oxford, 1979.

[12] J. Rosenthal and X. Wang, *Inverse eigenvalue problems for multivariable linear systems*, in Systems and Control in the Twenty-First Century, C. I. Byrnes, B. N. Datta, D. Gilliam, and C. F. Martin, eds., Birkäuser, Boston, Basel, Berlin, 1997, pp. 289–311.

[13] J. Rosenthal and X. Wang, *Eigenvalue completions by affine varieties*, Proc. Amer. Math. Soc., 128 (2000), pp. 643–646.

[14] W. Vogel, *Lectures on Results on Bézout's Theorem*, Tata Institute of Fundamental Research, Bombay, 1984.

[15] S.-F. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press, Beijing, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 524–533
© 2001 Society for Industrial and Applied Mathematics

EIGENVALUE PROBLEMS FOR ONE-DIMENSIONAL DISCRETE SCHRÖDINGER OPERATORS WITH SYMMETRIC BOUNDARY CONDITIONS*

JONQ JUANG†, WEN-WEI LIN‡, AND SHIH-FENG SHIEH†

Abstract. In this paper, we investigate the one-dimensional discrete Schrödinger equation with general, symmetric boundary conditions. Our results primarily concern the number of energy states lying in the wells.

Key words. eigenvalue, Schrödinger operator, boundary conditions

AMS subject classifications. Primary, 34L15, 34L40; Secondary, 81Q10, 34B05

PII. S0895479800371530

1. Introduction.

Allowing $a > 0$, we consider a discrete version of the eigenvalue problem of the Schrödinger operator

$$-\left(\frac{1}{m(x)}u'\right)' + V(x)u = \lambda u \tag{1.1a}$$

on a finite interval $I = [-a, a]$. Here $V(x)$ denotes the quantum well potential defined by

$$V(x) = \begin{cases} -v, & x \in [-b, b], \\ 0 & \text{otherwise}, \end{cases} \tag{1.1b}$$

where $a > b > 0$ and $v > 0$. Moreover, $m(x)$ is assumed to be a piecewise constant function. Specifically,

$$m(x) = \begin{cases} m', & x \in [-b, b], \\ m & \text{otherwise}. \end{cases} \tag{1.1c}$$

To discretize (1.1a), we divide $[-a, a]$ into equal parts of length h and assume a and b are integer multiples of h, i.e., $a = (M + N + 1)h$ and $b = Nh$. The node points of the discrete equation are

$$x_i = \left(\frac{a}{N + M + 1}\right) i, \qquad -M - N - 1 \le i \le M + N + 1.$$

*Received by the editors May 2, 2000; accepted for publication (in revised form) by P. Van Dooren June 14, 2001; published electronically November 13, 2001.
http://www.siam.org/journals/simax/23-2/37153.html

†Department of Applied Mathematics, National Chiao Tung University, Taiwan (jjuang@math.tamu.edu, ssf@math.nctu.edu.tw). The first and third authors completed their research while on sabbatical at Texas A&M University.

‡Department of Mathematics, National Tsing Hua University, Hsin-Chu 30050, Taiwan (wwlin@am.nthu.edu.tw).

Using a standard central-differencing technique, we then obtain the discrete version of (1.1) as follows:

$$u_{i+1} - 2(1 - m'(\lambda + v))u_i + u_{i-1} = 0, \qquad -N + 1 \le i \le N - 1, \tag{1.2a}$$

$$u_{i+1} - 2(1 - m\lambda)u_i + u_{i-1} = 0, \qquad N + 2 \le |i| \le N + M, \tag{1.2b}$$

$$\frac{u_{N+1}}{m + m'} + \frac{u_{N-1}}{2m'} - \frac{u_N}{m + m'} - \frac{u_N}{2m'} + (\lambda + v)u_N = 0, \tag{1.2c}$$

$$\frac{u_{N+2}}{2m} + \frac{u_N}{m + m'} - \frac{u_{N+1}}{2m} - \frac{u_{N+1}}{m + m'} + \lambda u_{N+1} = 0, \tag{1.2d}$$

$$\frac{u_{-N-1}}{m + m'} + \frac{u_{-N+1}}{2m'} - \frac{u_{-N}}{m + m'} - \frac{u_{-N}}{2m'} + (\lambda + v)u_{-N} = 0, \tag{1.2e}$$

and

$$\frac{u_{-N-2}}{2m} + \frac{u_{-N}}{m + m'} - \frac{u_{-N-1}}{2m} - \frac{u_{-N-1}}{m + m'} + \lambda u_{-N-1} = 0. \tag{1.2f}$$

We remark that the discrete formulation (1.2) of (1.1a) has been normalized in such a way that the step size h of the discretization is absorbed in λ and v. The following "symmetric" general boundary conditions are imposed:

$$u_{N+M+1} = \beta u_{-(N+M)} + \gamma u_{(N+M)} \tag{1.2g}$$

and

$$u_{-(N+M+1)} = \beta u_{(N+M)} + \gamma u_{-(N+M)}. \tag{1.2h}$$

In particular, $\beta = 0$ and $\gamma = 1$ (resp., $\gamma = 0$ and β=1) correspond to Neumann (resp., periodic) boundary conditions.

Eigenvalue ratios and gaps for the Schrödinger operators have been investigated by many authors (see, e.g., [1, 2, 5, 9, 10] and the works cited therein). On the other hand, the number $A(d)$ of eigenvalues less than a bound d is also of interest. Some partial results concerning the asymptotic behavior (i.e., as $d \to \infty$) of $A(d)$ are contained in [3]. We are led to investigate in this paper the number of energy states (eigenvalues) for a discrete Schrödinger problem (1.2a)–(1.2h) lying in the wells by the following work. In [6, 8], the spatial tunneling (from one well to the other) occurs in coupled quantum wells when the energy states in both wells are aligned. In the case of the hole tunneling in the coupled quantum wells, the tunneling mechanisms are significantly complicated due to band mixing effects. When the energy states are approximately aligned between heavy hole and light hole, the mixing tunneling occurs. Moreover, it was reported in [7] that the chaotic tunneling effect was generated when these two tunneling effects have a strong interaction between them. Our effort here is the first step toward understanding those phenomena.

Our main results are recorded in section 2. Specifically, we construct the characteristic equations of the problem (1.2a)–(1.2h). By analyzing the properties of such equations, we are able to compute the number of energy states in the well for some general, symmetric boundary conditions.

2. The main results. We begin with the following definition.

DEFINITION 2.1. *An eigenvector* $(u_i)_{i=-(N+M+1)}^{N+M+1}$ *of* (1.2a)–(1.2h) *is said to be symmetric (resp., antisymmetric) if* $u_i = u_{-i}$ *(resp.,* $u_i = -u_{-i}$*).*

PROPOSITION 2.1. *Each eigenvector of* (1.2a)–(1.2h) *is either symmetric or antisymmetric.*

Proof. It follows from the rank one modification of the symmetric matrices (see, e.g., [4]) that every eigenvalue of (1.2a)–(1.2h) is simple. Let $(u_i)_{i=-(N+M+1)}^{N+M+1}$ be an eigenvector corresponding to some eigenvalue λ. Note that $(u_{-i})_{i=-(N+M+1)}^{N+M+1}$ is also an eigenvector associated with λ. Thus, $u_i + u_{-i}$ and $u_i - u_{-i}$ are in the eigenspace corresponding to λ. We see that $u_i + u_{-i}$ is symmetric and $u_i - u_{-i}$ is antisymmetric. The assertion of the proposition now follows from the facts that every eigenspace is one-dimensional and that one of the vectors $u_i + u_{-i}$ and $u_i - u_{-i}$ is nonzero. ∎

DEFINITION 2.2. *Let λ be an eigenvalue of* (1.2a)–(1.2h) *whose corresponding eigenvector is symmetric (resp., antisymmetric); then λ is said to be symmetric (resp., antisymmetric).*

We remark that Definition 2.2 is well defined since every eigenspace is one-dimensional. We next derive some "characteristic" equations whose roots are eigenvalues of the system (1.2a)–(1.2h). To this end, we first assume λ is a symmetric eigenvalue; then

$$u_{i+1} - 2(1 - m'(v+\lambda))u_i + u_{i-1} = 0 \quad \text{for} \quad |i| \le N-1.$$

Hence,

$$u_i = A(s^i + s^{-i}) \quad \text{for} \quad |i| \le N, \tag{2.1}$$

where A is a constant to be determined and

$$s = 1 - m'(v+\lambda) + \sqrt{(1 - m'(v+\lambda))^2 - 1} \tag{2.2}$$

is a root of the characteristic polynomial $x^2 - 2(1-m'(v+\lambda))x + 1 = 0$. In particular, for

$$-v + \frac{2}{m'} \ge \lambda \ge -v, \tag{2.3}$$

we see that

$$u_i = 2A\cos i\theta. \tag{2.4}$$

Here $\theta = \cos^{-1}(1 - m'(v+\lambda))$. For $N+2 \le |i| \le N+M$,

$$u_{i+1} - 2(1 - m\lambda)u_i + u_{i-1} = 0,$$

and we have that

$$u_i = \begin{cases} Bt^{i-N} + Ct^{-(i-N)}, & N+1 \le i \le N+M+1, \quad \text{(2.5a)} \\ Bt^{-(i+N)} + Ct^{i+N}, & -(N+M+1) \le i \le -(N+1). \quad \text{(2.5b)} \end{cases}$$

Here B and C are constants to be determined and

$$t = (1 - m\lambda) + \sqrt{(1-m\lambda)^2 - 1}. \tag{2.6}$$

Using the boundary condition (1.2g) and the fact that u is symmetric, we see that

$$u_{N+M+1} = (\beta + \gamma)u_{N+M}. \tag{2.7}$$

Applying (2.5a) to (2.7), we get

$$C = (t^{2M+1})\left(\frac{\beta+\gamma-t}{1-(\beta+\gamma)t}\right)B \tag{2.8}$$
$$:= (t^{2M+1})(D_\beta)B.$$

Hence,

$$u_i = B(t^{i-N} + t^{2M+1}(D_\beta)t^{-(i-N)}), \qquad N+1 \le i \le N+M+1. \tag{2.9}$$

We next consider the connection at $i = N$ and $i = N+1$. Let

$$\alpha = \frac{2m}{m+m'} \quad \text{and} \quad \alpha' = \frac{2m'}{m+m'} \tag{2.10a, b}$$

at $i = N$; we thus write (1.2c) as

$$\alpha' u_{N+1} + (1-\alpha')u_N - 2(1-m'(\lambda+v))u_N + u_{N-1} = 0. \tag{2.11}$$

Using (2.11), (2.1), and (2.8), we see that (2.11) reduces to

$$\begin{aligned} &A(1-\alpha')(s^N + s^{-N}) + A(-2(1-m'(\lambda+v))(s^N+s^{-N})) + A(s^{N-1}+s^{-N+1}) \\ &= -\alpha' B(t + (D_\beta)t^{2M}). \end{aligned} \tag{2.12}$$

Noting that s^N and s^{-N} satisfy the recursive relation as given in (1.2a) with $i = N$, we see that (2.11) reduces to

$$A[(\alpha'-1)(s^N+s^{-N}) + (s^{N+1}+s^{-(N+1)})] = \alpha' B(t+(D_\beta)t^{2M}). \tag{2.13}$$

At $i = N+1$, a similar process yields

$$A\alpha(s^N + s^{-N}) = B[(\alpha-1)(t+(D_\beta)t^{2M}) + (1+(D_\beta)t^{2M+1})]. \tag{2.14}$$

Dividing (2.13) by (2.14), we conclude that every symmetric eigenvalue is a root of

$$\left[\frac{\alpha-1}{\alpha'} + \frac{1+(D_\beta)t^{2M+1}}{\alpha'(t+(D_\beta)t^{2M})}\right]^{-1} = \frac{\alpha'-1}{\alpha} + \frac{1}{\alpha}\frac{s^{N+1}+s^{-(N+1)}}{s^N+s^{-N}}, \tag{2.15}$$

where D_β is defined in (2.8). Similarly, we obtain that every antisymmetric eigenvalue is a root of

$$\left[\frac{\alpha-1}{\alpha'} + \frac{1+(D_{-\beta})t^{2M+1}}{\alpha'(t+(D_{-\beta})t^{2M})}\right]^{-1} = \frac{\alpha'-1}{\alpha} + \frac{1}{\alpha}\frac{s^{N+1}-s^{-(N+1)}}{s^N-s^{-N}}. \tag{2.16}$$

To investigate (2.15) and (2.16), we need to set up the following notations:

$$\xi_i^{\pm} := \frac{1+(D_{\pm\beta})t^{2i+1}}{t+(D_{\pm\beta})t^{2i}} = \frac{t^{-i}+(D_{\pm\beta})t^{i+1}}{t^{1-i}+(D_{\pm\beta})t^i}, \tag{2.17}$$

$$\eta_i^{\pm} := \frac{s^{i+1} \pm s^{-(i+1)}}{s^i \pm s^{-i}}, \tag{2.18}$$

$$f^{\pm}(\lambda) = \frac{\alpha'}{(\alpha-1)+\xi_M^{\pm}}, \tag{2.19}$$

and

$$g^{\pm}(\lambda) = \frac{\alpha' - 1}{\alpha} + \frac{\eta_N^{\pm}}{\alpha}. \tag{2.20}$$

Now (2.15) and (2.16) can be written as

$$f^{+}(\lambda) = g^{+}(\lambda) \tag{2.21}$$

and

$$f^{-}(\lambda) = g^{-}(\lambda). \tag{2.22}$$

DEFINITION 2.3. *Equations* (2.21) *and* (2.22) *are called the symmetric and antisymmetric characteristic equations of system* (1.2a)–(1.2h).

Obviously, the roots of symmetric (resp., antisymmetric) characteristic equations are the symmetric (resp., antisymmetric) eigenvalues of system (1.2a)–(1.2h). *The following useful recursive formulas can be verified directly:*

$$\xi_{i+1}^{\pm} = 2(1 - m\lambda) - \frac{1}{\xi_i^{\pm}}, \qquad 0 \le i \le M - 1, \tag{2.23a}$$

$$\xi_0^{\pm} = \frac{1}{\gamma \pm \beta}, \tag{2.23b}$$

and

$$\eta_{i+1}^{\pm} = 2(1 - m'(v + \lambda)) - \frac{1}{\eta_i^{\pm}}, \qquad 0 \le i \le N - 1, \tag{2.24a}$$

$$\eta_0^{+} = 1 - m'(v + \lambda), \quad \eta_0^{-} = \infty. \tag{2.24b}$$

In the following, we shall study the properties of functions $f^{\pm}$ and $g^{\pm}$.

PROPOSITION 2.2. *For whatever λ is defined, $\frac{d}{d\lambda} f^{\pm}(\lambda) > 0$ and $\frac{d}{d\lambda} g^{\pm}(\lambda) < 0$.*

Proof. We illustrate only the proof of $\frac{d}{d\lambda} g^{+}(\lambda) < 0$. The rest is similar. Using (2.24) we see that

$$\frac{d}{d\lambda}\eta_{i+1}^{+} = -2m' + \frac{1}{(\eta_i^{+})^2}\frac{d}{d\lambda}\eta_i^{+}$$

and

$$\frac{d}{d\lambda}\eta_0^{+} = -m' < 0.$$

By induction, we conclude that $\frac{d}{d\lambda}\eta_N^{+} < 0$ and, hence, that $\frac{d}{d\lambda} g^{+}(\lambda) < 0$. ∎

To further study $f^{\pm}$ and $g^{\pm}$, we need the following map:

$$F_{\lambda,m}(\xi) = 2(1 - m\lambda) - \frac{1}{\xi}. \tag{2.25}$$

PROPOSITION 2.3. *The following assertions hold true:*

(i) *$F_{\lambda,m}$ maps $[1, \infty] \cup [-\infty, 0)$ into $[1, \infty)$ for $\lambda \in (-\infty, 0]$.*
(ii) *$F_{\lambda,m}$ maps $[-\infty, -1] \cup (0, \infty]$ into $(-\infty, -1]$ for $\lambda \in \left[\frac{2}{m}, \infty\right)$.*
(iii) *$F_{\lambda,m}(\xi)$ is strictly increasing on $(-\infty, 0)$ and on $(0, \infty)$ for any λ.*

We skip the proof of Proposition 2.3.

PROPOSITION 2.4. *Let $2 > \alpha > 0$. If $\gamma + \beta \le 1$ (resp., $\gamma - \beta \le 1$), then $f^+(\lambda)$ (resp., $f^-(\lambda)$) is continuous on $(-\infty, 0]$. Moreover, $g^\pm(\lambda)$ are continuous on $(-\infty, -v] \cup \big[-v + \frac{2}{m'}, \infty\big)$. If $-1 \le \gamma+\beta$ (resp., $-1 \le \gamma-\beta$), then $f^+(\lambda)$ (resp., $f^-(\lambda)$) is continuous on $[\frac{2}{m}, \infty)$.*

Proof. Let $\lambda \in (-\infty, 0]$. If $\gamma \pm \beta \le 1$, then $\xi_0^\pm \in [1, \infty)$ or $(-\infty, 0)$. Suppose $\xi_0^\pm \in [1, \infty)$. Then it follows from Proposition 2.3(i) that

$$\xi_M^\pm(\lambda) = F_{\lambda,m}^M(\xi_0^\pm) \in [1, \infty). \tag{2.26}$$

Hence $-1 + \alpha + \xi_M^\pm(\lambda) \ne 0$, and so $f^\pm$ is continuous on $(-\infty, 0]$.

Suppose $\lambda \in (-\infty, -v] \cup \big[-v + \frac{2}{m'}, \infty\big)$. Then $\eta_0^-, \eta_0^+ = 1 - m'(v + \lambda) \in [1, \infty] \cup [-\infty, -1]$. It then follows from Proposition 2.3(i)–(ii) that

$$\eta_N^\pm(\lambda) = F_{\lambda+v,m'}^N(\eta_0^\pm) \in (-\infty, -1] \cup [1, \infty).$$

Hence, $g^\pm(\lambda)$ is continuous on $(-\infty, -v] \cup \big[-v + \frac{2}{m'}, \infty\big)$. If $-1 \le \gamma + \beta$ and λ is in $[\frac{2}{m}, \infty)$, then $F_{\lambda,m}(\xi_0^+) \le -2 - (\gamma + \beta) \le -1$. Hence, $\xi_M^+(\lambda) = F_{\lambda,m}^M(\xi_0^+) \in (-\infty, -1]$. The case for $-1 \le \gamma - \beta$ can be similarly obtained. We thus complete the proof of the proposition. □

It is clear from Proposition 2.4 that the singularities of $g^\pm(\lambda)$ occur in $\big(-v, -v + \frac{2}{m'}\big)$, and that, for $-1 \le \gamma + \beta \le 1$, the singularities of $f^\pm(\lambda)$ stay in $\big(0, \frac{2}{m}\big)$.

PROPOSITION 2.5. (i) *For $\lambda \in \big(-v, -v + \frac{2}{m'}\big)$, $g^+(\lambda)$, respectively, $g^-(\lambda)$, has singularities at*

$$-v + \frac{1}{m'}\left(1 - \cos\frac{2k-1}{2N}\pi\right) =: d_k, \qquad i = 1, 2, \ldots, N, \tag{2.27a}$$

respectively,

$$-v + \frac{1}{m'}\left(1 - \cos\frac{k}{N}\pi\right) =: e_k, \qquad k = 1, 2, \ldots, N-1. \tag{2.27b}$$

(ii) *The following ordering holds true:*

$$-v < d_1 < e_1 < d_2 < e_2 < \cdots < e_{n-1} < d_n < e_N := -v + \frac{2}{m'}. \tag{2.27c}$$

Proof. The proof of Proposition 2.5(i) follows from (2.18). The ordering in (2.27c) is obvious. □

Our main concern in this paper is the number of energy levels (eigenvalues) falling in the well, that is, the number of eigenvalues whose value is no greater than zero. Hence, the characterization of the singularities of $f^\pm(\lambda)$ on $\big(0, \frac{2}{m}\big)$ will not be pursued here.

PROPOSITION 2.6. (i) *Let $0 < \alpha < 2$. If $\gamma \pm \beta \le 1$, then*

$$\frac{\alpha'}{\alpha+1} < f^\pm(0) = \frac{\alpha'}{(\alpha-1) + \frac{1-(\gamma\pm\beta-1)M}{\gamma\pm\beta-(\gamma\pm\beta-1)M}} \le \frac{\alpha'}{\alpha}.$$

Consequently, $f^\pm(-v) < \frac{\alpha'}{\alpha}$. (ii) *$g^+(-v) = \frac{\alpha'}{\alpha}$, $\frac{\alpha+1}{\alpha} > g^-(-v) > \frac{\alpha'}{\alpha}$.* (iii) *If $v > \frac{2}{m'}$, then $g^\pm(0) < \frac{\alpha'-2}{\alpha}$.*

Proof. It follows from l'Hôspital's rule that

$$\lim_{\lambda\to 0}\xi_M^{\pm}=\lim_{t\to 1}\frac{1+\frac{(\gamma\pm\beta)-t}{1-(\gamma\pm\beta)t}t^{2M+1}}{t+\frac{(\gamma\pm\beta)-t}{1-(\gamma\pm\beta)t}t^{2M}}=\frac{1-(\gamma\pm\beta-1)M}{\gamma\pm\beta-(\gamma\pm\beta-1)M}.$$

To see the estimates for $\xi_M^{\pm}$ at $\lambda=0$, we consider the map $F_{0,m}$, as defined in (2.25). It is then clear that $1\le\xi_M^{\pm}<2$. Hence, $\frac{\alpha'}{\alpha+1}<f^{\pm}(0)\le\frac{\alpha'}{\alpha}$ as claimed. Since $f^{\pm}$ is increasing on $(-\infty,0]$, we obtain that $f^{\pm}(-v)<\frac{\alpha'}{\alpha}$. The assertions in (ii) are trivial. If $v>\frac{2}{m'}$, then, for $\lambda=0$, $\eta_0=1-m'v\le -1$. Using $F_{v,m'}$, we see, via Proposition 2.3(ii), that $\eta_N^{+}(0)\le -1$, and, hence, $g^{+}(0)\le\frac{\alpha'-2}{\alpha}$. Similarly, we obtain that $g^{-}(0)\le\frac{\alpha'-2}{\alpha}$. ∎

Notation 2.1. Set $R_1=\{(\gamma,\beta)\colon\ \beta\le 0,\gamma-\beta\le 1\}$ and $R_2=\{(\gamma,\beta)\colon\ \beta\ge 0,\ \gamma+\beta\le 1\}$.

PROPOSITION 2.7. *Let $2>\alpha>0$. If $(\gamma,\beta)\in R_1$ (resp., $(\gamma,\beta)\in R_2$), then $f^{-}(\lambda)\ge f^{+}(\lambda)$ (resp., $f^{-}(\lambda)\le f^{+}(\lambda)$) on $(-\infty,0]$. The equality holds only if $\beta=0$.*

Proof. We first note that $R_1\bigcup R_2=\{(\gamma,\beta)\colon\ \gamma+\beta\le 1 \text{ and } \gamma-\beta\le 1\}$. Let $(\gamma,\beta)\in R_1$; then one of the following three cases holds:

$$\xi_0^{+}\ge\xi_0^{-}\ge 1,\quad 0>\xi_0^{+}\ge\xi_0^{-},\quad\text{and}\quad \xi_0^{-}>1>0>\xi_0^{+}. \tag{2.28}$$

Furthermore, if the order of ξ_0^{+} and ξ_0^{-} satisfies one of the three cases in (2.28), then $F_{\lambda,m}(\xi_0^{+})\ge F_{\lambda,m}(\xi_0^{-})\ge 1$. It then follows from Proposition 2.5(ii)–(iii) that $\xi_M^{+}(\lambda)=F_{\lambda,m}^{M}(\xi_0^{+})\ge F_{\lambda,m}^{M}(\xi_0^{-})=\xi_M^{-}(\lambda)$. Consequently, $f^{+}(\lambda)\le f^{-}(\lambda)$. The other case can be similarly obtained. It is clear that $f^{-}(\lambda)=f^{+}(\lambda)$ only if $\xi_0^{+}=\xi_0^{-}$ or, equivalently, $\beta=0$. ∎

PROPOSITION 2.8. (i) *$g^{+}(\lambda)<g^{-}(\lambda)$ for $\lambda\in(-\infty,d_1)\cup\bigcup_{i=1}^{N-1}(e_i,d_{i+1})$.*

(ii) *$g^{+}(\lambda)>g^{-}(\lambda)$ for $\lambda\in\bigcup_{i=1}^{N}(d_i,e_i)\cup[e_N,\infty)$.*

Proof. For $\lambda\in(-\infty,-v]$, $\eta_0^{+}=1-m'(v+\lambda)\ge 1$. It follows from Proposition 2.3(i) that $F_{v+\lambda,m'}(\xi)\ge 1$ for $\xi=\eta_0^{+}$ or η_0^{-}. Hence, we see, via Proposition 2.3(iii), that $g^{+}(\lambda)<g^{-}(\lambda)$. It is clear that

$$\lim_{\lambda\to d_i^{-}}g^{+}(\lambda)=-\infty,\ \lim_{\lambda\to d_i^{+}}g^{+}(\lambda)=+\infty,\qquad i=1,2,\dots,N, \tag{2.29a}$$

and

$$\lim_{\lambda\to e_i^{-}}g^{-}(x)=-\infty,\ \lim_{\lambda\to e_i^{+}}g^{-}(x)=\infty,\qquad i=1,2,\dots,N-1. \tag{2.29b}$$

We next show that $g^{+}(\lambda)$ and $g^{-}(\lambda)$ do not intersect with each other. To this end, we set

$$e(s):=\frac{s^{2N+1}-s^{2N-1}}{s^{4N}-1}.$$

If $s^{2N+1}-s^{2N-1}=0$, then $s^2=1$. Now,

$$\lim_{s\to 1}e(s)=\lim_{s\to 1}\frac{(2N+1)s^{2N}-(2N-1)s^{2N}}{4Ns^{4N-1}}=\frac{1}{2N}.$$

Similarly,

$$\lim_{s\to 1^{-}}e(s)=-\frac{1}{2N}.$$

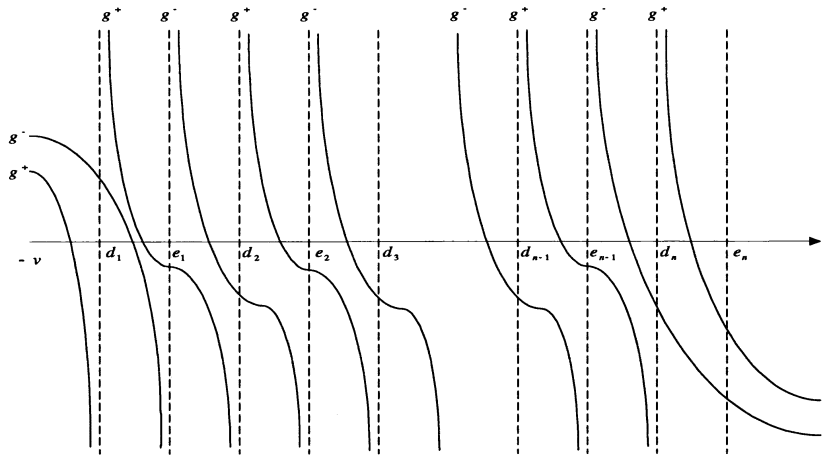


FIG. 1.

Thus, $e(s)$ is nonzero for all λ. It then follows from Proposition 2.2, equation (2.8), and the fact that $g^+(\lambda)$ does not intersect with $g^-(\lambda)$ for all λ that the assertions of the proposition hold true. □

Using the assertions of Proposition 2.8, we give a rough drawing of $g^+(\lambda)$ and $g^-(\lambda)$. Figure 1 reflects only the information of $g^{\pm}(\lambda)$ obtained in Proposition 2.8. It is by no means an accurate drawing of the graphs of $g^{\pm}(\lambda)$.

We are now ready to state our main results.

Notation 2.2. Set $S = \{-v < \lambda < 0\colon\ \lambda$ is the symmetric eigenvalue of system (1.2a)–(1.2h)$\}$, and $S_a = \{-v < \lambda < 0\colon\ \lambda$ is the antisymmetric eigenvalue of system (1.2a)–(1.2h)$\}$. The cardinalities of S and S_a are denoted by $\#_s$ and $\#_{sa}$, respectively. The sum of $\#_s$ and $\#_{sa}$ is denoted by $\#$.

THEOREM 2.1. *Let* $2 > \alpha > 0$. *Suppose* $(\gamma, \beta) \in R_1 \bigcup R_2$ *and* $v > \frac{2}{m'}$. *Then* $\#_s = N + 1$ *and* $\#_{as} = N$.

Proof. Using Proposition 2.6, we see that $f^{\pm}(-v) < g^{\pm}(-v)$. Thus, all eigenvalues of the system are greater than $-v$. It also follows from Proposition 2.6 that $f^{\pm}(0) \geq \frac{\alpha'}{\alpha+1} > \frac{\alpha'-2}{\alpha} = g^{\pm}(0)$. Hence, we conclude, via Figure 1, that $\# \geq 2N + 1$. However, the other eigenvalues of the system comes from the intersection of $f^{\pm}(\lambda)$ and $g^{\pm}(\lambda)$ on $\left(0, \frac{2}{m}\right)$. Upon using the fact that $f^{\pm}(0) \geq g^{\pm}(0)$, we conclude that $\# \leq 2N + 1$. Hence $\# = 2N + 1$. It is clear from Figure 1 that the assertions of the theorem hold. □

Remark 2.1.

1. Theorem 2.1 implies that if the depth of the well is "sufficiently large," then all the energy levels fall in the well.
2. Using Propositions 2.6 and 2.7, one also sees that for Neumann boundary conditions ($\gamma = 1,\ \beta = 0$), at least one energy level falls in the well regardless of the size of the well.

THEOREM 2.2. *Let* $(\gamma, \beta) \in R_1$, $\beta \neq 0$, *and* $v \leq \frac{2}{m'}$. *If* $e_k < 0 < d_{k+1}$ *for some* $1 \leq k \leq N$, *the following table holds true:*

	$f^+(0) > g^-(0)$	$f^+(0) > g^+(0)$ $f^-(0) < g^-(0)$	$g^+(0) > f^-(0)$
$\#_s$	$k+1$	$k+1$	k
$\#_{as}$	$k+1$	k	k

If $d_k < 0 < e_k$ for some $1 \le k \le N$, the following table holds true:

	$f^+(0) > g^+(0)$	$f^-(0) > g^-(0)$ $g^+(0) > f^+(0)$	$f^-(0) < g^-(0)$
$\#_s$	$k+1$	k	k
$\#_{as}$	k	k	$k-1$

If $e_k < 0 = d_{k+1}$ for some $1 \le k \le N$, the following table holds true:

	$f^-(0) > g^-(0)$	$f^-(0) < g^-(0)$
$\#_s$	$k+1$	$k+1$
$\#_{as}$	$k+1$	k

If $d_k < 0 = e_k$ for some $1 \le k \le N$, then the following table holds true:

	$f^+(0) > g^+(0)$	$f^+(0) < g^+(0)$
$\#_s$	$k+1$	k
$\#_{as}$	k	k

Proof. We first note that if $(\gamma, \beta) \in R_1$ and $\beta \neq 0$, then $f^+(\lambda) > f^-(\lambda)$ on $(-\infty, 0]$. The assertions of the theorem now follow from Propositions 2.2, 2.4, 2.8 and Figure 1. □

For $(\gamma, \beta) \in R_2$ or $\beta = 0$, similar tables as above can be obtained. In each of the tables above, the largest numbers of $\#_s$ and $\#_{as}$ occur when $f^{\pm}(0) > g^{\pm}(0)$. We next show that this is the case when the length v of the well is sufficiently close to $\frac{2}{m'}$.

THEOREM 2.3. *Let $m \approx m'$ and $(\gamma, \beta) \in R_1$. Suppose $v \le \frac{2}{m'}$ and v is sufficiently close to $\frac{2}{m'}$. Then $f^{\pm}(0) > g^{\pm}(0)$. Consequently, only the second column of each table in Theorem 2.2 holds.*

Proof. Direct calculation would yield that

$$g^+(e_k) = \frac{\alpha' - 1}{\alpha} + \frac{1}{\alpha} \cos \frac{k\pi}{N}$$

and

$$g^-(d_k) = \frac{\alpha' - 1}{\alpha} + \frac{1}{\alpha} \sin \frac{(2k-1)\pi}{2N}.$$

Suppose $v \left(\le \frac{2}{m'} \right)$ is sufficiently close to $\frac{2}{m'}$. Then $e_k, d_k \ge 0$ only if k is sufficiently large. If k is large, then $g^+(e_{k-1}) < 0$ and $g^-(d_{k-1}) \approx 0$, where we may assume that $d_{k-1}, e_{k-1} < 0$. Since $g^{\pm}$ are decreasing, $g^+(0) < g^+(e_{k-1}) < 0$ and $g^-(0) < g^-(d_{k-1}) \approx 0$. However, we see, via Proposition 2.6(i), that $f^{\pm}(0) = \frac{\alpha'}{\alpha+1} > 0$. We thus complete the proof of the theorem. □

We conclude the paper by mentioning some possible future related work. First, the study of discrete higher-dimensional Schrödinger problems is of considerable interest. Second, it would also be of interest to study a continuous version of the problem described in this paper. Finally, though the calculations would get more complicated, our approach here can be used to treat nonsymmetric boundary conditions.

REFERENCES

[1] M. S. ASHBAUGH AND R. D. BENGURIA, *Optimal bounds for ratios of eigenvalues of one-dimensional Schrödinger operators with Dirichlet boundary conditions and positive potentials*, Comm. Math. Phys., 124 (1989), pp. 403–415.

[2] M. S. Ashbaugh and R. D. Benguria, *Eigenvalue ratios for Sturm-Liouville operators*, J. Differential Equations, 103 (1993), pp. 205–219.
[3] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. 1, Wiley, New York, 1990.
[4] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
[5] Y. L. Huang and C. K. Law, *Eigenvalue ratios for the regular Sturm-Liouville system*, Proc. Amer. Math. Soc., 124 (1996), pp. 1427–1436.
[6] C. Juang, K. J. Kuhn, and R. B. Darling, *Stark shift and field-induced tunneling in $Al_xGa_{1-x}As/GaAs$ quantum-well structures*, Phys. Rev. B, 41 (1990), pp. 12047–12053.
[7] C. Juang, J. Y. Wang, and J. Juang, *Controlling chaotic behavior of heavy to light hole mixing tunneling by external electric fields*, IEEE J. Quantum Electron., 33 (1997), pp. 1345–1349.
[8] S. Luryi, *Polarization oscillations in coupled quantum wells-scheme for generation of submillimeter electromagnetic waves*, IEEE J. Quantum Electron., 27 (1991), pp. 54–60.
[9] B. N. Parlett and T. T. Lu, *Minimum eigenvalue separation*, Linear Algebra Appl., to appear.
[10] I. M. Singer, B. Wong, S.-T. Yau, and S. S.-T. Yau, *An estimate of the gap of the first two eigenvalues in the Schrödinger operator*, Ann. Scuola. Norm. Sup. Pisa Cl. Sci. (4), 12 (1985), pp. 319–333.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 534–550
© 2001 Society for Industrial and Applied Mathematics

RANK-ONE APPROXIMATION TO HIGH ORDER TENSORS*

TONG ZHANG† AND GENE H. GOLUB‡

Abstract. The singular value decomposition (SVD) has been extensively used in engineering and statistical applications. This method was originally discovered by Eckart and Young in [*Psychometrika*, 1 (1936), pp. 211–218], where they considered the problem of low-rank approximation to a matrix. A natural generalization of the SVD is the problem of low-rank approximation to high order tensors, which we call the multidimensional SVD. In this paper, we investigate certain properties of this decomposition as well as numerical algorithms.

Key words. singular value decomposition, low-rank approximation, tensor decomposition

AMS subject classifications. 15A69, 65F30

PII. S0895479899352045

1. Introduction of the problem. The problem of high order tensor decomposition has been studied by mathematicians who are interested in algebraic properties of tensors, by psychologists who need to analyze multiway data, as well as by engineers and statisticians who are interested in high order (tensor) statistics and independent component analysis (ICA). This decomposition is a generalization of the SVD that gives a low-rank approximation to a matrix (i.e., a second order tensor) [11]. However, a direct generalization of the SVD is nontrivial, since the definition of a rank that preserves all of the good properties of the SVD does not exist. At the current stage, little detail is known concerning general rank decompositions of a high order tensor, even though there have been a number of works in this direction [21, 24, 25, 27, 28]. As a consequence of the lack of a good tensor rank definition, there is no "best" way to define low-rank approximation for tensors of order higher than two, as pointed out in [27].

Computationally, the most popular method is based on alternating least squares minimization. However, the convergence behavior of this method has not been sufficiently analyzed. A rigorous analysis of the method is given in section 4. We also propose a new method to compute the optimal rank-one approximation. This algorithm is a generalization of the Rayleigh quotient iteration for eigenvalue problems. If we consider a matrix as a high order tensor, then an interesting application of this procedure leads to a novel method for computing a singular value/vector pair for the matrix.

An important application of multidimensional SVD is multiway analysis. Two models of decomposition have been frequently used: one is the Tucker3 model proposed in [30]; the other is the PARAFAC-CANDECOMP model proposed in [7, 16]. For third order tensors, the Tucker3 model is given by $\sum_{i,j,k} x_i \otimes y_j \otimes z_k \, g_{ijk}$, where g is an order 3 tensor called the *core array*. The PARAFAC-CANDECOMP model approximates a third order tensor by the sum of a few rank-one tensors—this is equivalent to the Tucker3 model with a diagonal core: $\sum_i x_i \otimes y_i \otimes z_i$. Both models can be

*Received by the editors February 17, 1999; accepted for publication (in revised form) by J. Varah April 2, 2001; published electronically November 13, 2001.
http://www.siam.org/journals/simax/23-2/35204.html
†IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (tzhang@watson.ibm.com).
‡Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305 (golub@sccm.stanford.edu).

easily extended to the higher order case. For more detailed descriptions of these models and existing computational algorithms, see [15, 17, 18, 22, 27] and the references therein.

Another application of multidimensional SVD is independent component analysis (or blind source separation). In this case, we attempt to find a matrix A from vector observations $x_1, \ldots, x_n$ that are taken from an unknown distribution D, such that the components of Ax are statistically independent when x is drawn from D. Many solutions have been proposed in the literature based on different formulations of this problem; see [3, 4, 5, 8, 9, 10, 26, 31] and the references therein. One solution, which is based on fourth order cumulants, solves the ICA problem by decomposing a symmetric fourth order tensor into the sum of symmetric orthogonal rank-one tensors [4, 8, 10] (see Definition 3.1). Note that a fourth order tensor $[a_{ijkl}]$ is symmetric if we have $a_{ijkl} = a_{i'j'k'l'}$ for any permutation $(i'j'k'l')$ of $(ijkl)$. From the tensor decomposition point of view, this approach to the ICA problem leads to an orthogonal PARAFAC-CANDECOMP model.

There is an interesting relationship between rank-one and rank-F approximation in the PARAFAC-CANDECOMP model. In the second order (matrix) case, from the optimal approximation property of the SVD, the optimal rank-F approximation of a tensor is equivalent to the following incremental rank-one approximation approach: we first fit the original tensor by a rank-one tensor, then subtract the rank-one approximation from the original tensor and fit the residue with another rank-one tensor. This procedure is repeated until F rank-one tensors are found. Therefore for second order tensors, the rank-F approximation problem can be reduced to the rank-one approximation problem. The simplicity of this incremental rank-one fitting procedure is very attractive; although for higher order tensors it is not necessarily equivalent to the PARAFAC-CANDECOMP approximation. However, we will show in section 3 that for the special case of *orthogonally decomposable tensors* defined later (this special case includes the fourth order cumulants approach to the ICA problem), the incremental rank-one approximation procedure yields the solution to the optimal rank-F approximation.

Therefore, computationally, we focus only on the following rank-one approximation problem: find vectors x, y, and z to minimize

$$\sum_{i,j,k} (x_i y_j z_k - a_{ijk})^2, \tag{1.1}$$

where $[a_{ijk}]$ denotes a third order tensor. For notational simplicity, we illustrate our results by using third order tensors whenever generalizations to higher order cases are straightforward. Subtle differences will be mentioned when they exist.

2. Equivalent rank-one formulations. Note that in (1.1), each vector x, y, or z is only determined up to a scaling factor. Therefore we can impose the constraints $\|x\|_2 = \|y\|_2 = \|z\|_2 = 1$ and write (1.1) as

$$\min \sum_{i,j,k} (\lambda x_i y_j z_k - a_{ijk})^2. \tag{2.1}$$

DEFINITION 2.1. *Given nonzero vectors x, y, and z, the generalized Rayleigh quotient (GRQ) is defined as*

$$\mathrm{GRQ}(x, y, z) = \frac{\sum_{i,j,k} a_{ijk} x_i y_j z_k}{\|x\|_2 \, \|y\|_2 \, \|z\|_2}.$$

Similar to the standard Rayleigh quotient, we define the generalized Rayleigh quotient $\mathrm{GRQ}(x, y, z)$ in a way that is invariant under a scaling of x, y, or z. It is easy to verify that if $\|x\|_2 = \|y\|_2 = \|z\|_2 = 1$, then $\lambda = \mathrm{GRQ}(x, y, z)$ minimizes (2.1), and the minimum value is

$$\sum_{i,j,k} a_{ijk}^2 - \mathrm{GRQ}(x, y, z)^2. \tag{2.2}$$

It thus follows that (1.1) is equivalent to the dual problem of maximizing GRQ

$$\max \sum_{i,j,k} a_{ijk} x_i y_j z_k, \tag{2.3}$$

under the constraints that

$$\sum_i x_i^2 = \sum_j y_j^2 = \sum_k z_k^2 = 1. \tag{2.4}$$

We can write down the Lagrangian for the dual problem as

$$\sum_{i,j,k} a_{ijk} x_i y_j z_k - \frac{\mu_1}{2} \sum_i (x_i^2 - 1) - \frac{\mu_2}{2} \sum_j (y_j^2 - 1) - \frac{\mu_3}{2} \sum_k (z_k^2 - 1). \tag{2.5}$$

By differentiating (2.5), we obtain the following system at a critical point for each component x_i of x, y_j of y, and z_k of z:

$$\begin{cases} \sum_{j,k} a_{ijk} y_j z_k = \mu_1 x_i, \\ \sum_{i,k} a_{ijk} x_i z_k = \mu_2 y_j, \\ \sum_{i,j} a_{ijk} x_i y_j = \mu_3 z_k. \end{cases} \tag{2.6}$$

We now multiply x_i, y_j, and z_k to the first, second, and third equations, and sum over i, j, and k, respectively. This gives $\mu_1 = \mu_2 = \mu_3 = \sum_{i,j,k} a_{ijk} x_i y_j z_k = \mathrm{GRQ}(x, y, z)$. Let $\lambda = \mathrm{GRQ}(x, y, z)$; then we can rewrite the above system as

$$\begin{cases} \sum_{j,k} a_{ijk} y_j z_k = \lambda x_i, \\ \sum_{i,k} a_{ijk} x_i z_k = \lambda y_j, \\ \sum_{i,j} a_{ijk} x_i y_j = \lambda z_k, \\ \sum_{i,j,k} a_{ijk} x_i y_j z_k = \lambda. \end{cases} \tag{2.7}$$

Note that a nonzero solution to (2.7) automatically guarantees that $\|x\|_2 = \|y\|_2 = \|z\|_2 = 1$.

3. A special tensor decomposition. In this section, we study the following orthogonal tensor decomposition.

DEFINITION 3.1. *We say that a tensor $[a_{ijk}]$ is orthogonally decomposable if it can be written as the sum of F rank-one tensors $x_p \otimes y_p \otimes z_p$ $(p = 1, \ldots, F)$ such that $x_p \perp x_q$, $y_p \perp y_q$, and $z_p \perp z_q$ for $p \neq q$.*

It is not difficult to extend this definition to include higher order tensors. In general, orthogonal decompositions do not necessarily exist. However, for the ICA problem, the fourth order cumulant tensors are orthogonally decomposable [4]. In the ICA literature, a Jacobi-type scheme for approximate diagonalization of multiple symmetric matrices has been proposed to compute this decomposition [4, 6]. Some

numerical aspects of simultaneous matrix diagonalization can be found in [2, 12, 13]. In the following, we show that if a tensor is of order higher than two and is orthogonally decomposable, then the decomposition can be correctly computed by the incremental rank-one approximation algorithm.[1] Therefore, the orthogonal tensor decomposition problem can be reduced to the rank-one approximation problem. A consequence of this result is the uniqueness of orthogonal decomposition for tensors of order higher than two. This uniqueness of decomposition (in the special case of third order tensors) has been previously discovered in [28] using a different analysis.

Now consider the orthogonal decomposition of tensor $[a_{ijk}]$:

$$a_{ijk} = \sum_{p=1}^{F} x_{ip} y_{jp} z_{kp},$$

where we assume that $\sum_i x_{ip} x_{iq} = 0$, $\sum_j y_{jp} y_{jq} = 0$, and $\sum_k z_{kp} z_{kq} = 0$ for $p \neq q$. Note that for convenience, we use the notation v_{ip} to denote the ith component of a vector v_p.

Consider the least squares rank-one approximation (1.1), for which we can always do an orthogonal transformation separately for each index i, j, and k without changing the least squares error. Therefore, without loss of generality, we can assume that $x_{ip} = \alpha_p \delta_{ip}$, $y_{ip} = \beta_p \delta_{ip}$, and $z_{ip} = \gamma_p \delta_{ip}$, where

$$\delta_{ij} = \begin{cases} 1 \text{ if } i = j, \\ 0 \text{ if } i \neq j \end{cases}$$

is the Kronecker delta symbol.

Let $\lambda_p = \alpha_p \beta_p \gamma_p$; then $a_{ijk} = \sum_p \lambda_p \delta_{ip} \delta_{jp} \delta_{kp}$. Let (x^*, y^*, z^*) be a nonzero solution of (1.1), and let $x' = x^*/\|x^*\|_2$, $y' = y^*/\|y^*\|_2$, and $z' = z^*/\|z^*\|_2$. Without loss of generality, we can assume that $|x'_1|$ achieves $\max(\|x'\|_\infty, \|y'\|_\infty, \|z'\|_\infty)$. By (2.7), we have $\lambda x'_1 = \lambda_1 y'_1 z'_1$, where $|\lambda|$ is given by (2.3).

Since $|\lambda|$ achieves the maximum in (2.3), $|\lambda| \geq |\lambda_1|$. Assume that $[a_{ijk}]$ is nonzero; then $\lambda \neq 0$ and $x'_1 \neq 0$. We thus obtain the inequality $|x'_1| \leq |y'_1 z'_1|$. Note that by assumption, $1 \geq |x'_1| \geq \max(|y'_1|, |z'_1|)$. Therefore the inequality can be achieved only at $|x'_1| = |y'_1| = |z'_1| = 1$, which shows that $|\lambda| = |\lambda_1| = \max_i |\lambda_i|$ and $x' = y' = z' = e_1$, where e_1 is the vector with 1 in the first element and 0 elsewhere. Therefore an optimal rank-one approximation is given by $x^* \otimes y^* \otimes z^* = \lambda_1 e_1 \otimes e_1 \otimes e_1$.

Since $[a_{ijk}] - x^* \otimes y^* \otimes z^*$ is still orthogonally decomposable with rank $F - 1$ by definition, it follows that by repeating the rank-one approximation algorithm F times, we obtain the decomposition $\sum_{p=1}^{F} x_{ip} y_{jp} z_{kp}$. Observe also that the uniqueness of the computational procedure implies that the orthogonal decomposition of $[a_{ijk}]$ is unique, and the same analysis is valid for tensors with order greater than 3. We can summarize the above results in the following theorem.

THEOREM 3.2. *If a tensor of order at least 3 is orthogonally decomposable, then this decomposition is unique, and the incremental rank-one approximation algorithm correctly computes it.*

System (2.7) is stable under a small perturbation if the Jacobi matrix J in (4.5) is not singular. Since our analysis is based on the equality $\lambda x'_1 = \lambda_1 y'_1 z'_1$ that comes from (2.7), the decomposition computed by the incremental rank-one approximation

[1] In [27], the authors introduced a more general concept of orthogonal PCA-k decomposition. They argued that the decomposition can be computed using the incremental rank-one approximation procedure. However, their proof was faulty. See [21] for a detailed study.

method is also stable under a small perturbation of $[a_{ijk}]$. This is not true for matrices since J becomes singular (see section 4). Consequently, the singular vectors can be nonunique and unstable under perturbation when some of the singular values are not distinct.

4. Algorithms.

4.1. Generalized Rayleigh–Newton iteration.

Newton's method can be applied to solve (2.7) for critical points. In order to derive the algorithm, we shall state an important property of the GRQ.

THEOREM 4.1. *Assume that (λ, x, y, z) is a nonzero solution to* (2.7)*; then for small perturbations δx, δy, and δz, we have*

$$\mathrm{GRQ}(x+\delta x, y+\delta y, z+\delta z) = \lambda + O(\|\delta x\|_2^2 + \|\delta y\|_2^2 + \|\delta z\|_2^2).$$

Proof. We can assume without loss of generality that $x^T\delta x = y^T\delta y = z^T\delta z = 0$. This is because we can write δx as the sum of a component orthogonal to x and a component parallel to x (similarly for δy and δz). The component parallel to x does not modify the GRQ; thus only the component orthogonal to x, which is at most as large as δx, contributes to the change of the GRQ.

Now we have

$$\sum_{i,j,k} a_{ijk}\delta x_i y_j z_k = \lambda \sum_i x_i \delta x_i = 0.$$

Similarly $\sum_{i,j,k} a_{ijk} x_i \delta y_j z_k = \sum_{i,j,k} a_{ijk} x_i y_j \delta z_k = 0$. Therefore

$$\begin{aligned}
&\sum_{i,j,k} a_{ijk}(x_i+\delta x_i)(y_j+\delta y_j)(z_k+\delta z_k) \\
&= \sum_{i,j,k} a_{ijk} x_i y_j z_k + \left(\sum_{i,j,k} a_{ijk}\delta x_i y_j z_k + \sum_{i,j,k} a_{ijk} x_i \delta y_j z_k + \sum_{i,j,k} a_{ijk} x_i y_j \delta z_k \right) \\
&\qquad + \sum_{i,j,k} (a_{ijk}\delta x_i \delta y_j z_k + a_{ijk}\delta x_i y_j \delta z_k + a_{ijk} x_i \delta y_j \delta z_k + a_{ijk}\delta x_i \delta y_j \delta z_k) \\
&= \lambda + 0 + O\left(\sum_{i,j} |\delta x_i \delta y_j| + \sum_{j,k} |\delta y_j \delta z_k| + \sum_{i,k} |\delta x_i \delta z_k| + \sum_{i,j,k} |\delta x_i \delta y_j \delta z_k| \right) \\
&= \lambda + O(\|\delta x\|_2 \|\delta y\|_2 + \|\delta x\|_2 \|\delta z\|_2 + \|\delta y\|_2 \|\delta z\|_2). \qquad (4.1)
\end{aligned}$$

The last equality follows from the Schwartz inequality. We also note that

$$\|x+\delta x\|_2 = \sqrt{\|x\|_2^2 + 2x^T\delta x + \|\delta x\|_2^2} = \sqrt{1+0+\|\delta x\|_2^2} = 1 + O(\|\delta x\|_2^2).$$

Similarly, $\|y+\delta y\|_2 = 1 + O(\|\delta y\|_2^2)$, and $\|z+\delta z\|_2 = 1 + O(\|\delta z\|_2^2)$. Therefore

$$\|x+\delta x\|_2 \, \|y+\delta y\|_2 \, \|z+\delta z\|_2 = 1 + O(\|\delta x\|_2^2 + \|\delta y\|_2^2 + \|\delta z\|_2^2). \qquad (4.2)$$

Observe that (4.1) and (4.2) are, respectively, the numerator and the denominator in

the definition of $\mathrm{GRQ}(x+\delta x, y+\delta y, z+\delta z)$; therefore

$$\begin{aligned}\mathrm{GRQ}(x+\delta x, y+\delta y, z+\delta z) &= \frac{\lambda + O(\|\delta x\|_2\|\delta y\|_2 + \|\delta x\|_2\|\delta z\|_2 + \|\delta y\|_2\|\delta z\|_2)}{1 + O(\|\delta x\|_2^2 + \|\delta y\|_2^2 + \|\delta z\|_2^2)} \\ &= \lambda + O(\|\delta x\|_2\|\delta y\|_2 + \|\delta x\|_2\|\delta z\|_2 + \|\delta y\|_2\|\delta z\|_2) \\ &\quad + O(\|\delta x\|_2^2 + \|\delta y\|_2^2 + \|\delta z\|_2^2) \\ &= \lambda + O(\|\delta x\|_2^2 + \|\delta y\|_2^2 + \|\delta z\|_2^2). \qquad \square\end{aligned}$$

This theorem is a generalization of the following well-known fact: for eigenvalue problems, the Rayleigh quotient is quadratically as accurate as the approximate eigenvector. Note that the theorem is also quite intuitive from the following nonrigorous argument: since a critical point of (2.3) optimizes $\mathrm{GRQ}(x,y,z)$, $\mathrm{GRQ}(x,y,z)$ has a zero gradient at a critical point.

Based on Theorem 4.1, a procedure similar to the Rayleigh quotient iteration (cf. [14]) can be obtained. By Taylor expansion, we know that given λ, a linearization of (2.7) at (x,y,z) gives

$$\begin{aligned}\sum_{i,j,k} a_{ijk}(y_j\delta z_k + \delta y_j z_k) - \lambda\delta x_i &= \lambda x_i - \sum_{i,j,k} a_{ijk} y_j z_k, \\ \sum_{i,j,k} a_{ijk}(x_i\delta z_k + \delta x_i z_k) - \lambda\delta y_j &= \lambda y_j - \sum_{i,j,k} a_{ijk} x_i z_k, \\ \sum_{i,j,k} a_{ijk}(x_i\delta y_j + \delta x_i y_j) - \lambda\delta z_k &= \lambda z_k - \sum_{i,j,k} a_{ijk} x_i y_j.\end{aligned} \tag{4.3}$$

Now, let the (approximate) true solution be $x^* = x+\delta x$, $y^* = y+\delta y$, and $z^* = z+\delta z$; we obtain the following linearizations:

$$\begin{aligned}-\lambda x_i^* + \sum_{j,k} a_{ijk}(y_j z_k^* + z_k y_j^*) &= \sum_{j,k} a_{ijk} y_j z_k, \\ -\lambda y_j^* + \sum_{i,k} a_{ijk}(x_i z_k^* + z_k x_i^*) &= \sum_{i,k} a_{ijk} x_i z_k, \\ -\lambda z_k^* + \sum_{i,j} a_{ijk}(x_i y_j^* + y_j x_i^*) &= \sum_{i,j} a_{ijk} x_i y_j.\end{aligned} \tag{4.4}$$

For rth order tensors, the right-hand side should be multiplied by $r-2$ (which is 1 in our case of $r=3$). The reason is that in the general case, 2-way product terms such as $y_j z_k$ in (2.7) are replaced by $(r-1)$-way product terms. The linearization of each of the $(r-1)$-way product terms contributes $r-1$ additive terms to the left-hand side of the equation, which needs to be compensated by a multiple of $r-2$ on the right-hand side for compatibility with (2.7). Note that for matrices (second order tensors), the right-hand side is zero, and this is related to the singularity of the left-hand side when $r=2$ (this point will be discussed later). In general, the above linear equation can be written in the matrix form as

$$J(\lambda, w) w^* = b(w), \tag{4.5}$$

where w denotes the vector concatenation of x, y, and z, and $b(w)$ is the right-hand side of (4.4). Here

$$J(\lambda, w) = \begin{bmatrix} -\lambda I_{d_1\times d_1} & A_3 & A_2 \\ A_3^T & -\lambda I_{d_2\times d_2} & A_1 \\ A_2^T & A_1^T & -\lambda I_{d_3\times d_3} \end{bmatrix},$$

where $I_{d_1\times d_1}$, $I_{d_2\times d_2}$, and $I_{d_3\times d_3}$ are identity matrices corresponding to the x, y, and z directions, respectively. The (i,j)th element of A_3 is $\sum_k a_{ijk}z_k$ $(i=1,\ldots,d_1;$ $j=1,\ldots,d_2)$; the (i,k)th element of A_2 is $\sum_j a_{ijk}y_j$ $(i=1,\ldots,d_1;$ $k=1,\ldots,d_3)$; and the (j,k)th element of A_1 is $\sum_i a_{ijk}x_i$ $(j=1,\ldots,d_2;$ $k=1,\ldots,d_3)$. To some extent, $J(\lambda,w)$ can be regarded as the Jacobian of (2.7) or the Hessian of (2.3).

Note that given x, y, and z, λ can be taken as the GRQ $\lambda=\mathrm{GRQ}(x,y,z)$; and given λ, w can be updated by (4.5). We can alternate between these two steps, and that leads to the following algorithm.

ALGORITHM 4.1 (GRQ-Newton iteration).
Given initial estimate $w^0=[x^0,y^0,z^0]^T$
for $p=0,\ldots,$
  normalize w^p so that $\|x^p\|_2=\|y^p\|_2=\|z^p\|_2=1$
  let $\lambda^p=\mathrm{GRQ}(x^p,y^p,z^p)$
  solve $J(\lambda^p,w^p)w^{p+1}=b(w^p)$ for w^{p+1}
endfor

Note that (4.5) is used in Algorithm 4.1 since this particular formulation is directly comparable to the standard RQI (Rayleigh quotient iteration). However, our later convergence analysis will mostly rely on (4.3), which can be written as

$$J(\lambda,w)\delta w=c(\lambda,w), \tag{4.6}$$

where $c(\lambda,w)=\lambda w-\frac{1}{r-2}b(w)$ and δw corresponds to the difference $w^{p+1}-w^p$ in Algorithm 4.1. Equation (4.6) can also be more suitable for iterative algorithms since problems introduced by the nondefiniteness of J are alleviated (this point is discussed shortly).

As we have mentioned, there is a factor $r-2$ in $b(w)$ for the order r tensor formulation. Consequently, an important observation is that Algorithm 4.1 fails at $r=2$ since $b(w)\equiv 0$. This case corresponds to the standard matrix SVD. A standard RQI replaces the definition of $b(w)$ by $b(w)=w$. The inconsistency of the algorithm at $r=2$ is due to the singularity of $J(\lambda^*,w^*)$ at the critical point (λ^*,w^*).

For the order r tensor formulation of (4.5), let W_r be the $r\times r$ matrix consisted of all 1's except for -1's on the diagonal. Let $(\mu,[\alpha_1,\ldots,\alpha_r]^T)$ be an eigenpair of W_r, and consider the vector $\tilde{w}^*=[\alpha_1 w_1^{*T},\ldots,\alpha_r w_r^{*T}]^T$, where (λ^*,w^*) is a solution to (2.7), and $w^*=[w_1^{*T},\ldots,w_r^{*T}]^T$. Since, for all i, $\sum_{j\neq i}\alpha_j=(\mu+1)\alpha_i$, by using the critical point equation (2.7) we obtain

$$J(\lambda^*,w^*)\tilde{w}^*=-\lambda^*\tilde{w}^*+(\mu+1)\lambda^*\tilde{w}^*.$$

This implies that $\tilde{w}^*$ is an eigenvector of $J(\lambda^*,w^*)$ with an eigenvalue $\lambda^*\mu$. Since

$$W_r=v_r v_r^T-2I_{r\times r},$$

where v_r is the column vector of dimension r that is composed of all 1's, W_r has one eigenvalue of $r-2$ and the rest are -2. It follows that when $r=2$, $J(\lambda^*,w^*)$ is always singular. However, when $r>2$, such a conclusion cannot be drawn from this analysis. In fact, $J(\lambda^*,w^*)$ becomes singular only in degenerate cases, which rarely happens in practice. This is the fundamental difference between the case $r=2$ and the case $r>2$. Therefore unlike the ill-conditioned standard RQI, matrix J is usually well-conditioned when $r>2$. In this case, Algorithm 4.1 is consistent, and it locally achieves quadratic convergence as shown in Theorem 4.2.

From the above discussion, we can see that vectors $[\alpha_1 w_1^{*T},\ldots,\alpha_r w_r^{*T}]^T$ span an invariant subspace of $J(\lambda^*,w^*)$. Although the matrix J is indefinite at (λ^*,w^*),

the only positive eigenvalue of $J(\lambda^*, w^*)$ is $(r-2)\lambda^*$ with the eigenvector w^*. Because $c(\lambda^*, w^*)$ in (4.6) is orthogonal to w^*, J behaves like a definite matrix in a neighborhood of w^* for (4.3) if iterative methods are employed. Computationally, we do not have to factorize the matrix at each iteration if the direct factorization of a size $\sum_j m_j$ matrix ($O((\sum_j m_j)^3)$ operations) is costly. A (preconditioned) Krylov subspace method [14] may be employed in practice. With a fixed number of inner iterations, the computation requires only $O((\sum_j m_j)^2)$ operations. However, this method reduces the quadratic convergence shown in the following theorem to linear convergence.

THEOREM 4.2. *Let (λ^*, w^*) be a nonzero solution to (2.7). If $J(\lambda^*, w^*)$ is nonsingular, then Algorithm 4.1 converges to (λ^*, w^*) quadratically in a neighborhood of (λ^*, w^*).*

Proof. Since $J(\lambda^*, w^*)w^* = b(w^*)$, it follows from the linearization formulations (4.3) and (4.5) that

$$\begin{aligned} 0 &= J(\lambda^*, w^*)w^* - b(w^*) \\ &= J(\lambda^*, w)w^* - b(w) + O(\|w^* - w\|_2^2). \end{aligned}$$

Since $\lambda^* = \mathrm{GRQ}(w^*)$ and, by Theorem 4.1, $\mathrm{GRQ}(w) = \lambda^* + O(\|w^* - w\|_2^2)$, we have $J(\lambda^*, w) = J(\mathrm{GRQ}(w), w) + O(\|w^* - w\|_2^2)$. Note that b does not depend on λ. Therefore

$$J(\mathrm{GRQ}(w), w)w^* - b(w) = O(\|w^* - w\|_2^2).$$

We have assumed that J is nonsingular at $J(\lambda^*, w^*)$; therefore

$$\|w^{p+1} - w^*\|_2 = \|J(\mathrm{GRQ}(w^p), w^p)^{-1} b(w^p) - w^*\|_2 = O(\|w^p - w^*\|_2^2).$$

Also note that if $w = [x^T, y^T, z^T]^T$ is in a neighborhood of $w^* = [x^{*T}, y^{*T}, z^{*T}]^T$, then

$$\left\| \left[\frac{x^T}{\|x\|_2}, \frac{y^T}{\|y\|_2}, \frac{z^T}{\|z\|_2} \right]^T - w^* \right\|_2 = O(\|w - w^*\|_2).$$

This implies that the normalization step in Algorithm 4.1 preserves the rate of convergence. ∎

Algorithm 4.1 can also be used to find a singular value and vector for a matrix (or an eigenvalue and vector for a symmetric matrix). Let B be a matrix of size $m_1 \times m_2$. We can regard it as a third order tensor of size $m_1 \times m_2 \times 1$. Since the third dimension is always unit 1 after normalization, the iteration depends only on vectors x and y corresponding to the first two dimensions of B. Matrix J can be written as

$$J(\lambda, [x, y]) = \begin{bmatrix} -\lambda I_{m_1 \times m_1} & B & By \\ B^T & -\lambda I_{m2 \times m_2} & B^T x \\ y^T B^T & x^T B & -\lambda \end{bmatrix}, \tag{4.7}$$

and b becomes

$$b([x, y]) = \begin{bmatrix} By \\ B^T x \\ x^T By \end{bmatrix}, \tag{4.8}$$

where both x and y have to be normalized: $\|x\|_2 = \|y\|_2 = 1$. If J is nonsingular, which is extremely likely, then Algorithm 4.1 is consistent with good convergence properties. This algorithm can be compared to the standard RQI, where the last row and the last column of J are omitted in the definition, and b is replaced by $w = [x^T, y^T]^T$. We note that in the RQI, two choices can be made in the normalization step. The first one is to normalize x and y separately after each iteration. This case corresponds to an application of Algorithm 4.1 in that the GRQ is equivalent to the traditional Rayleigh quotient. The second choice is only to normalize $w = [x^T, y^T]^T$ as a whole ($\|x\|_2$ and $\|y\|_2$ can be different). This choice is more standard since it is obtained from the direct application of RQI to the equivalent eigenvalue problem of a SVD problem. However, in this case, the GRQ is not equivalent to the traditional Rayleigh quotient. The difference between these two normalization procedures is very small, as indicated by Example 5.2. Finally, it should be noticed that we can regard B as even higher order tensors, and that leads to different procedures.

4.2. Alternating least squares method. In practice, the most commonly used method for solving (1.1) is the alternating least squares (ALS) algorithm, which was studied in [1, 20, 23]. An interesting property of this procedure is that it generalizes the power method for eigenvalue problems. Other generalizations are possible, such as the Jacobi procedure described in the next section. Although the convergence of this method was studied, the rate of convergence has not yet been analyzed in the literature. We show that by using the formulation developed in the previous sections, we can prove linear convergence of this method in a neighborhood of the optimal solution.

ALGORITHM 4.2 (ALS).
Given initial position $w^0 = [x^0, y^0, z^0]^T$
for $p = 0, \ldots,$
  for $i = 1, \ldots,$
    $x_i^{p+1} = \sum_{i,j,k} a_{ijk} y_j^p z_k^p$
  endfor
  for $j = 1, \ldots,$
    $y_j^{p+1} = \sum_{i,j,k} a_{ijk} x_i^{p+1} z_k^p$
  endfor
  for $k = 1, \ldots,$
    $z_k^{p+1} = \sum_{i,j,k} a_{ijk} x_i^{p+1} y_j^{p+1}$
  endfor
  normalize so that $\|x^{p+1}\|_2 = \|y^{p+1}\|_2 = \|z^{p+1}\|_2 = 1$
endfor

Algorithm 4.2 is derived by individually varying x (or y or z) while fixing the other two vectors in (1.1). It can be easily checked from (2.3) that the optimal x is proportional to $\sum_{i,j,k} a_{ijk} y_j z_k$. Due to the normalization step, λ does not need to appear in the algorithm.

Another way to look at this method is to regard it as a nonlinear version of the Gauss–Seidel iteration (cf. [14]) applied to (2.7). Locally, after a linearization of the original problem, this algorithm can be regarded as an approximation of the block Gauss–Seidel iteration for solving the linear system (4.5). Although the Jacobian matrix $J(\lambda, w)$ can be indefinite, as we have discussed in section 4.1, the right-hand side of (4.3) lies approximately in the subspace where J is definite. Moreover, at each Gauss–Seidel iteration, δx (δy or δz) is approximately orthogonal to x (y or z); therefore each direction generated by the Gauss–Seidel iteration lies in the subspace

where J is definite. This indicates that J can essentially be regarded as a definite operator. From this reasoning, we can obtain the following theorem.

THEOREM 4.3. *Assume that* $(\lambda^*, x^*, y^*, z^*)$ *maximizes* (2.3) *and* $J(\lambda^*, w^*)$ *is nonsingular, where* $w^* = [x^*, y^*, z^*]^T$*; then Algorithm* 4.2 *converges to* (λ^*, w^*) *linearly in a neighborhood of* (λ^*, w^*).

Proof. When w^p is close to w^*, consider w^{p+1} obtained by Algorithm 4.2, u^{p+1} obtained by Algorithm 4.1, and v^{p+1} obtained by approximately solving (4.5) with the block Gauss–Seidel iteration. That is, for each $k = 1, \ldots, 3$, solve the following equation for v_k^{p+1}:

$$\sum_{\ell \le k} J_{k,\ell}(\lambda^p, w^p) v_\ell^{p+1} + \sum_{\ell > k} J_{k,\ell}(\lambda^p, w^p) w_\ell^p = b_k(w^p), \tag{4.9}$$

where the subscript k (or ℓ) indicates one of the block components corresponding to x, y, or z. As mentioned earlier, w^p is obtained from the nonlinear block Gauss–Seidel version of (4.9). Now notice that (4.5) is a linearization of (2.7) and $\lambda^p = \mathrm{GRQ}(w^p)$ is second order in $w^p - w^*$ from λ^*. Therefore at each step k, $v_k^{p+1} - w_k^{p+1}$ is second order in $w^p - w^*$ and $w^p - w^{p+1}$. That is, $\|v^{p+1} - w^{p+1}\| = O(\|w^p - w^*\|^2 + \|w^p - w^{p+1}\|^2)$. Also by Theorem 4.2 we have $\|u^{p+1} - w^*\| = O(\|w^p - w^*\|^2)$. Therefore we only need to show that $\|v^{p+1} - u^{p+1}\| \le \alpha \|w^p - u^{p+1}\|$ for some $\alpha < 1$ where α is independent of w^p.

By (4.6), we know that $J(\lambda^p, w^p)(u^{p+1} - w^p) = c(\lambda^p, w^p)$. Equation (4.9) implies that $v^{p+1} - w^p$ can be regarded as the approximation of the solution $\delta w = u^{p+1} - w^p$ to (4.6) after one block Gauss–Seidel iteration. Let $J^* = J(\lambda^*, w^*)$; then (4.6) can be replaced by $J^* \delta w = c(\lambda^p, w^p)$ with second order accuracy both for the exact solution $u^{p+1} - w^p$ and for the Gauss–Seidel approximation $v^{p+1} - w^p$. We thus need to show only that the Gauss–Seidel iteration for solving $J^* \delta w = c(\lambda^p, w^p)$ converges linearly with the starting point $\delta w = 0$. Furthermore, each component $c_k(\lambda^p, w^p)$ of the right-hand side is orthogonal to w_k^p by the definition of λ^p. Therefore, if we let V denote the subspace spanned by $[x^{*T}, 0, 0]^T$, $[0, y^{*T}, 0]^T$, and $[0, 0, z^{*T}]^T$, then we can decompose $c(\lambda^p, w^p)$ as $\bar{c} + \Delta c$ such that $\bar{c} \in V^\perp$ and $\Delta c = O(\|\bar{c}\| \cdot \|w^p - w^*\|)$. Since Δc is a small perturbation which does not affect the linear convergence rate, we need to show only that the Gauss–Seidel iteration for solving $J^* \delta u = \bar{c}$ converges linearly with the starting point $\delta u = 0$.

It is easy to check that if $\bar{c} \in V^\perp$, then each new component δu_k generated from the block Gauss–Seidel iteration also lies in $V^\perp$. Therefore, the convergence relies only on the properties of J^* in the subspace $V^\perp$. Since (λ^*, w^*) maximizes (2.3) and J^* is nonsingular, J^* has to be negative definite on $V^\perp$. The theorem follows from the well-known fact that the Gauss–Seidel iteration converges linearly for definite matrices [14]. ∎

There are many possible variants of Algorithm 4.2. One is to replace the Gauss–Seidel iteration by an iterative algorithm with a better convergence behavior (such as the nonlinear version of successive overrelaxation or a Krylov subspace method [14]). Another variant is to vary two or more components (x or y or z) at the same time, instead of varying only one component. Note that the optimization of varying two components can be obtained from an SVD algorithm. Varying more than two components at the same time leads to a divide-and-conquer approach. However, all of these algorithms have linear convergence rates. Therefore locally the GRQ-Newton iteration is more efficient computationally.

4.3. Jacobi Gauss–Newton procedure. For problems that need to be solved on a parallel computer, it is often desirable to use the Jacobi version of Algorithm 4.2. There is also an interesting relationship between this Jacobi algorithm and the Gauss–Newton procedure for nonlinear least squares problems. The Gauss–Newton method is an approximation to Newton's method with the property that the resulted linear system is always semidefinite. As we show in section 5, this method (Algorithm 4.3) could be much slower than Newton's method. However, it still has many useful applications, such as for certain engineering problems where the enhanced stability of the Gauss–Newton procedure may be desirable. In addition, the method does not require the Hessian matrix, which may be expensive to compute.

ALGORITHM 4.3 (Jacobi Gauss–Newton iteration).
Given initial position $w^0 = [x^0, y^0, z^0]^T$
for $p = 0, \ldots,$
$x_i^{p+1} = \sum_{i,j,k} a_{ijk} y_j^p z_k^p,$
$y_j^{p+1} = \sum_{i,j,k} a_{ijk} x_i^p z_k^p,$
$z_k^{p+1} = \sum_{i,j,k} a_{ijk} x_i^p y_j^p$
normalize so that $\|x^{p+1}\|_2 = \|y^{p+1}\|_2 = \|z^{p+1}\|_2 = 1$
endfor

To derive Algorithm 4.3 as a Gauss–Newton procedure, we consider the formulation (2.1) where λ is still estimated as GRQ(w). Given this parameter λ, we linearize each term $\lambda(x_i + \delta x_i)(y_j + \delta y_j)(z_k + \delta z_k) - a_{ijk}$ as

$$\lambda x_i y_j z_k - a_{ijk} + \lambda(\delta x_i y_j z_k + x_i \delta y_j z_k + x_i y_j \delta z_k).$$

For the Gauss–Newton procedure, we work with the least squares formulation of this linearization as follows:

$$\min_{\delta x, \delta y, \delta z} \sum_{i,j,k} [\lambda x_i y_j z_k - a_{ijk} + \lambda(\delta x_i y_j z_k + x_i \delta y_j z_k + x_i y_j \delta z_k)]^2. \tag{4.10}$$

The above system is singular; therefore, we need to impose the following normalization constraints: $x^T \delta x = y^T \delta y = z^T \delta z = 0$. After some algebraic manipulations, we obtain the following solution to (4.10):

$$\delta x_i = \sum_{j,k} a_{ijk} y_j z_k / \lambda - x_i,$$
$$\delta y_j = \sum_{i,k} a_{ijk} x_i z_k / \lambda - y_j,$$
$$\delta z_k = \sum_{i,j} a_{ijk} x_i y_j / \lambda - z_k.$$

By normalizing $x_i + \delta x_i$, $y_j + \delta y_j$, and $z_k + \delta z_k$, we obtain an update rule that is equivalent to Algorithm 4.3. Similar to Algorithm 4.2, the Gauss–Newton iteration approximately solves (4.5), but it does not guarantee the convergence even locally. However, in our experiments the algorithm usually converges, although the observed rate of convergence is slightly slower than that of Algorithm 4.2. The main advantage of this method is its parallelizability.

4.4. Computational costs. In order to compare algorithms described in the previous sections, it is necessary to analyze their computational costs. To be more

general, we analyze these algorithms in the case of rth order tensor approximation. Assume that the dimension m_j for each side of the tensor satisfies $m_1 \geq m_2 \geq \cdots \geq m_r$. We also assume that $[a_{i_1,\ldots,i_r}]$ is not sparse and does not have any special structure that we can take advantage of. For simplicity, we keep only operation counts accurate up to the leading term in our analysis.

Given any vector w_j, we define the inner product of an rth order tensor $A = [a_{i_1,\ldots,i_r}]$ and a vector w_j as a tensor of order $r-1$: $\langle a, w_j\rangle = [\sum_{i_j} a_{i_1,\ldots,i_r} w_{i_j,j}]$. The computation of this inner product requires $2\prod_j m_j$ operations. Therefore, in order to evaluate GRQ, the best way is to compute the inner product of A and w_j in the order from $j = 1$ to $j = r$. This computation requires $2\sum_j \prod_{k\geq j} m_k \approx 2\prod_j m_j$ operations.

We can now consider Algorithm 4.2. In the inner loop, the equation for the first component requires us to compute the inner product of A and w_j^p $(j > 1)$, with a total of $2\prod_j m_j$ operations. Since the normalization of w_1^{p+1} requires $O(m_1)$ operations, the total cost should be $2\prod_j m_j$. For each component $k > 1$, we need to compute the inner product of A and w_j^{p+1} for $j = 1,\ldots,k-1$. Since the inner product of A and w_j^{p+1} for $j = 1,\ldots,k-2$ is available from the previous step, the overall computation at step k is to take the inner product of $\langle A, [w_j^{p+1}]_{1:k-2}\rangle$ with w_{k-1}^{p+1} in $2\prod_{j\geq k-1} m_j$ operations, and then the inner product of $\langle A, [w_j^{p+1}]_{1:k-1}\rangle$ with $[w_{k-1}^{p}]_{k+1:r}$ in $2\prod_{j\geq k} m_j$ operations. The total operation count at step $k > 1$ is therefore $2\prod_{j\geq k-1} m_j + 2\prod_{j\geq k} m_j$. Summing over k, we need $4\prod_j m_j$ operations for each outer iteration. Since the procedure described for computing Algorithm 4.2 is still valid for evaluating inner products $\langle A, [w_1,\ldots,\hat{w}_k,\ldots,w_r]\rangle$ for $k = 1,\ldots,n$ simultaneously ($\hat{w}_k$ indicates that w_k is omitted in the inner product computation), it follows that Algorithm 4.3 also requires $4\prod_j m_j$ operations.

We now show that $6\prod_j m_j$ operations are needed to compute J and b in Algorithm 4.1. Note that b can be obtained from J in $o(\prod_j m_j)$ operations; therefore we need to show only that J can be obtained in $6\prod_j m_j$ operations. This can be done in two steps. In the first step, we compute $J_{1,\ell}$ (and therefore $J_{\ell,1}$), which requires m_1 times $4\prod_{j>1} m_j$ operations from the previous analysis. In the second step, we first compute $\langle A, w_1\rangle$, which requires $2\prod_j m_j$ operations, and then recursively compute the inner products of $\langle A, w_1\rangle$ with different combinations of $r-3$ vectors from $[w_j]_{2:n}$, which takes only $O(\prod_{j>1} m_j)$ operations. Therefore a total of $6\prod_j m_j$ operations are needed to compute J and b. Since $O((\sum_j m_j)^3)$ operations are required to solve the linear system, each iteration in Algorithm 4.1 costs $6\prod_j m_j + O((\sum_j m_j)^3)$ operations. This will be comparable to the computational costs of the other two algorithms if $\prod_j m_j$ is at least of order m_1^3. Note that if an iterative method is employed, then $O((\sum_j m_j)^2)$ operations are required to solve the linear system approximately. In this case, however, we may obtain only a linear convergence rate.

The computational costs for each iteration of the algorithms are summarized in Table 4.1. Algorithm 4.1 is denoted by GRQI; Algorithm 4.2 is denoted by ALS; and Algorithm 4.3 is denoted by GN.

TABLE 4.1
Comparison of computational costs in flops.

Algorithm	GRQI	ALS	GN
Flops/iteration	$6\prod_{j=1}^r m_j + O((\sum_{j=1}^r m_j)^3)$	$4\prod_{j=1}^r m_j$	$4\prod_{j=1}^r m_j$

5. Experimental results. Since the computational cost for each method has already been discussed, we study only the convergence behavior for these methods. We give two examples: the first example compares the convergence of the algorithms with synthetic and real multiway datasets; the second example focuses on the matrix SVD problem. "Optimal solutions" in these examples are obtained numerically up to the machine precision (denoted by ϵ_{mach}). This has been done by first using ten iterations of ALS to find an approximate solution and then using GRQI until the error is within the machine precision. Strictly speaking, the computed optimal solution w^* may not be exact. The tensors are also not necessarily orthogonally decomposable in the examples, and we compute only rank-one approximations. However, rank-F approximations can also be obtained by using the incremental algorithm we have mentioned. Although the scheme may not lead to an optimal low-rank approximation, a reasonable approximation may still be obtained. In all of the following experiments, we report the average performance of ten runs of the algorithms with ten different randomly generated initial vectors. In each of the ten runs, the same initial vector is used for all algorithms.

5.1. Example 1. The optimal solution is denoted as w^*, and the residue $R(w)$ of w is defined by (1.1). We do not report convergence results for GRQ since its behavior is similar to $R(w)$ (both are of the order $\|w - w^*\|_2^2$).

In Table 5.1, we consider a random low-rank $40 \times 30 \times 40$ tensor generated as the sum of 20 rank-one tensors—each rank-one tensor $x \otimes y \otimes z$ is generated with components of x, y, and z uniformly distributed in $(0, 1)$. The 2-norm of this tensor is 569. For each of the ten runs, we start with a randomly generated initial vector, having an average residue of 448. The residue of w^* is 74. As we can see, a rank-one approximation reduces the residue by a factor of 6. The condition number of the Jacobian at w^* for GRQI is about 2, which explains why, in this case, both ALS and GN converge relatively quickly. Another interesting observation is that all three algorithms converge to the optimal solution from a randomly generated starting approximation. We believe this is related to the dominance of the optimal rank-one decomposition in this example.

TABLE 5.1
Positive low-rank random $40 \times 30 \times 40$ *tensor.*

$\|w_k - w^*\|_2$				
p	1	2	3	4
GRQI	7.4×10^{-3}	1.0×10^{-7}	5.4×10^{-16}	ϵ_{mach}
ALS	1.2×10^{-2}	1.6×10^{-4}	4.1×10^{-7}	3.2×10^{-9}
GN	1.8×10^{-2}	6.8×10^{-4}	3.0×10^{-5}	1.4×10^{-6}
$R(w_k) - R(w^*)$				
p	1	2	3	4
GRQI	1.2×10^{-1}	2.8×10^{-11}	ϵ_{mach}	ϵ_{mach}
ALS	3.5×10^{-1}	6.2×10^{-5}	3.9×10^{-10}	5.0×10^{-13}
GN	7.4×10^{-1}	1.0×10^{-3}	1.9×10^{-6}	4.6×10^{-9}

Table 5.2 shows the results with a random $10 \times 15 \times 20 \times 20$ tensor. Each entry of the tensor is an independently generated Gaussian variable with mean 0 and standard deviation 1. The norm of the tensor is 246.00 and the optimal approximation has residue 245.68. Therefore, the optimal rank-one approximation performs very poorly. Also in this case, we observe that GRQI and GN may converge to nonoptimal approximations if we start with random approximations. Therefore we use starting

approximations that are generated by randomly perturbing the computed optimal solution so that they are close to the optimal solution. The condition number of the Jacobian for GRQI at w^* is 58.4, which is relatively large. Thus both ALS and GN converge much more slowly than GRQI.

TABLE 5.2
Gaussian random $10 \times 15 \times 20 \times 20$ *tensor.*

$\|w_k - w^*\|_2$				
p	1	2	3	4
GRQI	1.3×10^{-3}	3.8×10^{-6}	6.0×10^{-11}	2.4×10^{-15}
ALS	1.9×10^{-2}	1.0×10^{-2}	7.6×10^{-3}	5.8×10^{-3}
GN	2.4×10^{-2}	1.8×10^{-2}	1.5×10^{-2}	1.3×10^{-2}
$R(w_k) - R(w^*)$				
p	1	2	3	4
GRQI	1.3×10^{-7}	8.1×10^{-13}	ϵ_{mach}	ϵ_{mach}
ALS	4.6×10^{-5}	8.3×10^{-6}	2.8×10^{-6}	1.2×10^{-6}
GN	1.6×10^{-4}	9.7×10^{-5}	6.8×10^{-5}	5.3×10^{-5}

Table 5.3 uses the growth curve data of French girls from [19]. The dataset is a $12 \times 30 \times 8$ tensor, indicating 12 ages, 30 French girls, and 8 additional variables. The rank-one approximation performs very well for this real data, partially because all variables are positive. While the original tensor norm is 70808, the optimal solution reduces the residue to 8123. The condition number of Jacobian is 2.3, and we observe similar convergence behaviors as shown in Table 5.1.

TABLE 5.3
Growth curve data of French girls.

$\|w_k - w^*\|_2$				
p	1	2	3	4
GRQI	1.4×10^{-1}	3.7×10^{-4}	1.8×10^{-9}	5.0×10^{-16}
ALS	4.2×10^{-2}	6.7×10^{-4}	9.7×10^{-6}	1.4×10^{-7}
GN	1.1×10^{-1}	1.2×10^{-2}	1.3×10^{-3}	1.5×10^{-4}
$R(w_k) - R(w^*)$				
p	1	2	3	4
GRQI	5.0×10^{3}	1.0×10^{-1}	9.0×10^{-13}	ϵ_{mach}
ALS	5.8×10^{2}	1.5×10^{-1}	3.2×10^{-5}	6.9×10^{-9}
GN	3.1×10^{3}	4.9×10^{1}	6.1×10^{-1}	7.4×10^{-3}

5.2. Example 2. In this example, we compare three algorithms for computing a singular value of a matrix. GRQI is the method of treating the matrix as a three-dimensional tensor and applying Algorithm 4.1, which is described in section 4.1. RQI denotes the standard Rayleigh quotient iteration applied to the equivalent eigenvalue problem. NRQI denotes the standard Rayleigh quotient with separate normalization of x and y after each iteration, as described in section 4.1. We generate a random 40×50 matrix with entries uniformly distributed in $(0, 1)$. Table 5.4 reports the convergence of singular values (denoted by σ) and singular vectors (denoted by w) obtained by the algorithms after each iteration. After the fourth iteration, the estimated condition numbers are 2.4 for GRQI, and of the order 10^{16} for both NRQI and RQI. For this problem, GRQI not only is much better conditioned, but it also seems

TABLE 5.4
40×50 *singular value problem.*

$\|w_k - w^*\|_2$				
p	1	2	3	4
GRQI	6.0×10^{-3}	1.5×10^{-8}	4.0×10^{-15}	ϵ_{mach}
NRQI	7.2×10^{-2}	2.3×10^{-4}	2.6×10^{-11}	ϵ_{mach}
RQI	7.2×10^{-2}	2.3×10^{-4}	2.6×10^{-11}	ϵ_{mach}
$\|\sigma_k - \sigma^*\|$				
p	1	2	3	4
GRQI	7.4×10^{-4}	2.1×10^{-14}	ϵ_{mach}	ϵ_{mach}
NRQI	1.1×10^{-1}	1.8×10^{-6}	9.7×10^{-15}	ϵ_{mach}
RQI	1.1×10^{-1}	1.8×10^{-6}	8.8×10^{-15}	ϵ_{mach}

to converge faster. From the table, we also see that all of the algorithms achieve the machine precision after four iterations.

We shall mention that it is possible to define the inverse iteration procedure for eigenvalue problems in a more well-conditioned way by introducing a constraint $x^T \delta x = 0$ (see, e.g., [29]). This leads to a more traditional Newton-type method, which is not equivalent to our formulation. Our formulation has the advantage that the eigenvalue structure of the J matrix is better understood (see section 4.1). However, the exact relationship between this method and the traditional Newton's method requires further investigation.

6. Conclusions. We have shown that if a tensor of order higher than 2 is orthogonally decomposable, then the decomposition is unique and can be computed by repeatedly applying a rank-one approximation algorithm. Furthermore, even if the tensor to be approximated is not orthogonally decomposable, incremental rank-one approximation can still be useful due to its simplicity.

Based on these observations, it is important to study numerical aspects of the rank-one tensor approximation problem. Specifically, we are able to prove a local linear convergence rate of the popular ALS algorithm. In addition, based on a formulation that generalizes the Rayleigh quotient variational method for symmetric matrix eigenvalue problems, we are able to derive a generalized Rayleigh quotient-Newton iteration (GRQI), which locally has a quadratic convergence rate. For dense high order tensors, the computation is likely to be dominated by tensor-vector products. Therefore, locally this method can be more efficient than ALS even after the cost of matrix factorization is taken into consideration. We have also pointed out that ALS can be viewed as a nonlinear Gauss–Seidel procedure for approximately solving the linear system in GRQI. This relationship implies that more sophisticated iterative algorithms can be applied.

Many open problems still remain though. For example, general properties of high order tensor decompositions are still not well understood. It might also be interesting to study numerical methods that directly compute a rank-F tensor approximation, instead of using the incremental rank-one approximation procedure suggested in this paper.

Acknowledgments. We are grateful to the referees for many constructive comments and suggestions that led to substantial improvement of this paper. One referee also brought reference [21] to our attention.

REFERENCES

[1] J. T. Berge, J. D. Leeuw, and P. Kroonenberg, *Some additional results on principal components analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 52 (1987), pp. 183–191.

[2] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *Numerical methods for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 927–949.

[3] J.-F. Cardoso, *Infomax and maximum likelihood for source separation*, IEEE Lett. Signal Process., 4 (1997), pp. 112–114.

[4] J.-F. Cardoso and P. Comon, *Independent component analysis, a survey of some algebraic methods*, in the IEEE International Symposium on Circuits and Systems, Vol. 2, IEEE, New York, 1996, pp. 93–96.

[5] J.-F. Cardoso and B. Laheld, *Equivariant adaptive source separation*, IEEE Trans. Signal Process., 44 (1996), pp. 3017–3030.

[6] J.-F. Cardoso and A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 161–164.

[7] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[8] P. Comon, *MA identification using fourth order cumulants*, Signal Process., 26 (1992), pp. 381–388.

[9] P. Comon, *Independent component analysis, a new concept?*, Signal Process., 36 (1994), pp. 287–314.

[10] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 93–107.

[11] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[12] B. N. Flury and W. Gautschi, *An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 169–184.

[13] B. Flury and B. Neuenschwander, *Simultaneous diagonalization algorithms with applications in multivariate statistics*, in Approximation and Computation, Internat. Ser. Numer. Math. 119, Birkhäuser Boston, Boston, MA, 1994, pp. 179–205.

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

[15] H. Harmann, *Modern Factor Analysis*, University of Chicago, Chicago, 1977.

[16] R. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an 'explanatory' multi-mode factor analysis*, in UCLA Working Papers in phonetics, Vol. 16, UCLA, Los Angeles, 1970, pp. 1–84.

[17] R. Harshman and M. Lundy, *PARAFAC: Parallel factor analysis*, Comput. Statist. Data Anal., 18 (1994), pp. 39–72.

[18] R. Henrion, *N-way principal component analysis. Theory, algorithms and applications*, Chem. Intell. Lab. Syst., 25 (1994), pp. 1–23.

[19] J. Janssen, F. Marcotorchino, and J. Proth, eds., *International Symposium on Data Analysis, the Ins and Outs of Solving Real Problems*, Plenum, New York, 1987.

[20] H. Kiers, *An alternating least squares algorithm for PARAFAC2 and three-way DEDICOM*, Comput. Statist. Data Anal., 16 (1993), pp. 103–118.

[21] T. G. Kolda, *Orthogonal Tensor Decompositions*, Tech. report SAND2000-8566, Sandia National Laboratories, Albuquerque, NM, and Livermore, CA, 2000.

[22] P. Kroonenberg, *Three-mode principal component analysis: Theory and applications*, DSWO Press, Leiden, The Netherlands, 1983.

[23] P. Kroonenberg and J. D. Leeuw, *Principal component analysis of three-mode data by means of alternating least square algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[24] J. Kruskal, *Three way arrays: Rank and uniqueness of trilinear decomposition with applications to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[25] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and n-way arrays*, in Multiway Data Analysis, North-Holland, Amsterdam, 1989, pp. 7–18.

[26] D. D. Lee, U. Rokni, and H. Sompolinsky, *Algorithms for independent components analysis and higher order statistics*, in Advances in Neural Information Processing Systems, Vol. 12, S. Solla, T. Leen, and K.-R. Müller, eds., MIT Press, Cambridge, MA, 2000, pp. 491–497.

[27] D. Leibovici and R. Sabatier, *A singular value decomposition of a k-way array for a principal component analysis of multiway data, pta-k*, Linear Algebra Appl., 269 (1998), pp. 307–329.

[28] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[29] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Rev., 21 (1979), pp. 339–360.
[30] L. Tucker, *Some mathematical notes on three mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[31] H. Yang and S. Amari, *Adaptive on-line learning algorithms for blind separation—maximum entropy and minimum mutual information*, Neural Comput., 9 (1997), pp. 1457–1482.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 551–562
© 2001 Society for Industrial and Applied Mathematics

IMPLICITLY RESTARTED ARNOLDI METHODS AND SUBSPACE ITERATION*

R. B. LEHOUCQ†

Abstract. This goal of this paper is to present an elegant relationship between an implicitly restarted Arnoldi method (IRAM) and nonstationary (subspace) simultaneous iteration. This relationship allows the geometric convergence theory developed for nonstationary simultaneous iteration due to Watkins and Elsner [*Linear Algebra Appl.,* 143 (1991), pp. 19–47] to be used for analyzing the rate of convergence of an IRAM. We also comment on the relationship with other restarting schemes. A set of experiments demonstrates that implicit restarted methods can converge at a much faster rate than simultaneous iteration when iterating on a subspace of equal dimension.

Key words. simultaneous iteration, Arnoldi reduction, Schur decomposition, restarting, eigenvalues

AMS subject classifications. 65F15, 65G05

PII. S0895479899358595

1. Introduction.

A classical method of solution for the large-scale eigenvalue problem is simultaneous iteration [6, 9, 25, 26, 28, 36, 38]. Simultaneous iteration was originally introduced by Bauer [7], who called the method *Treppeniteration* (staircase iteration). The iteration is a straightforward method for computing the eigenvalues of largest modulus of a matrix $\mathbf{A}$ and is a generalization of the power method in that a subspace of size larger than one is employed. Unfortunately, the rate of convergence of simultaneous iteration is often prohibitive for many large-scale eigenvalue problems.

The goal and contribution of this paper is the derivation of an elegant relationship between an implicitly restarted Arnoldi method (IRAM) and nonstationary subspace or simultaneous iteration. We show that an IRAM, including its block extension, is nonstationary simultaneous iteration in disguise. The relationship with nonstationary simultaneous iteration is demonstrated by exploiting a well-known connection [24, 37, 39] with the QR algorithm. By appealing to the results in [40], possible rates of convergence for an IRAM are established. Moreover, we show how an IRAM computes a nested sequence of approximations for the partial Schur decomposition associated with the dominant invariant subspace of a matrix. Numerical experiments show that an IRAM can converge significantly faster than simultaneous iteration when acting on subspaces of equal dimension.

We remark that the relationship between an IRAM and simultaneous iteration presented also applies to all other restarted Arnoldi methods that employ the same initial Krylov space and restart polynomial. The elegant relationship with simultaneous iteration provided by implicit restarting, however, also provides a robust and stable algorithm. This robustness and stability is a direct result of an exclusive reliance on

*Received by the editors July 12, 1999; accepted for publication (in revised form) by Z. Strakoš May 18, 2001; published electronically November 13, 2001. This work was supported by the Department of Energy (contracts DE-FG0f-91ER25103 and W-31-109-Eng-38). The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/simax/23-2/35859.html

†Sandia National Laboratories, P.O. Box 5800, MS 1110, Albuquerque, NM 87185-1110 (rblehou@sandia.gov). Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-AC04-94AL85000.

unitary transformations. Golub and Wilkinson [11] examine the many practical difficulties involved when computing invariant subspaces. They conclude that working with an orthonormal basis of approximate Schur vectors is a better-behaved numerical process. Within the context of simultaneous iteration, Stewart [38] also arrives at the same conclusion.

Section 2 discusses the eigenvalue problem and Schur decompositions. Section 3 introduces the Arnoldi reduction and its computation. The relationship between an IRAM and nonstationary simultaneous iteration is derived in section 4. Section 5 describes a geometric convergence theory of an IRAM. Section 6 discusses some practical issues related to the convergence of an IRAM. The results of some numerical experiments comparing simultaneous iteration with an IRAM are given in section 7.

We conclude this section with the basic notation to be used in this article. We employ Householder notational conventions. Capital and lowercase letters denote matrices and vectors, respectively, while lowercase Greek letters denote scalars.

The transpose of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^T$, and the complex conjugate of $\mathbf{x}^T$ is denoted by $\mathbf{x}^H$. The norms used are the Euclidean and Frobenius, denoted by $\|\cdot\|$ and $\|\cdot\|_F$, respectively. The range of a matrix $\mathbf{A}$ is denoted by $\mathcal{R}(\mathbf{A})$.

A matrix of lower bandwidth b will be called a banded upper Hessenberg matrix. We drop "upper" when the context is clear. Omission of the word *band* implies that the block size is one. We say that a banded Hessenberg matrix is unreduced if all the elements on the bth subdiagonal are nonzero.

2. The eigenvalue problem. Let $\mathbf{A}$ be a complex matrix of order n. We are interested in computing the $k \ll n$ dominant (those of largest magnitude) eigenvalues and associated invariant subspace of

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{2.1}$$

The eigenvalues and eigenvectors of $\mathbf{A}$ are denoted by λ_j and $\mathbf{x}_j$, respectively, for $j = 1, \ldots, n$. For the remainder of our article, we assume that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $|\lambda_k| > |\lambda_{k+1}|$.

The following decomposition proves central to the eigenvalue algorithms considered in this article. The decomposition's value is in providing us with a canonical form for which stable algorithms may be developed. For us, a stable algorithm computes the exact Schur decomposition of a nearby matrix.

THEOREM 2.1 (Schur decomposition). *If $\mathbf{A} \in \mathbf{C}^{n\times n}$, then there exists a unitary $\mathbf{Z} \in \mathbf{C}^{n\times n}$ such that*

$$\mathbf{Z}^H\mathbf{A}\mathbf{Z} = \mathbf{T}, \tag{2.2}$$

where $\mathbf{T}$ is an upper triangular matrix. The eigenvalues can appear in any order along the diagonal.

Proof. See [10, p. 313]. □

Let $\mathbf{D}$ be a diagonal unitary matrix. Then $(\mathbf{Z}\mathbf{D})^H\mathbf{A}\mathbf{Z}\mathbf{D} = \mathbf{D}^H\mathbf{T}\mathbf{D}$ has diagonal blocks equal to those of $\mathbf{T}$. Thus, apart from the eigenvalues of multiplicity larger than one, the decomposition is essentially unique, given some ordering of the eigenvalues. Denote the leading principal matrix of order k of $\mathbf{T}$ by $\widehat{\mathbf{T}}_k$. Let $\mathbf{Z}_k$ be the corresponding columns of $\mathbf{Z}$. Then $\mathbf{A}\mathbf{Z}_k = \mathbf{Z}_k\widehat{\mathbf{T}}_k$ is a *partial* Schur decomposition of $\mathbf{A}$ of order k. When $\mathbf{A}$ is Hermitian, $\mathbf{T}$ is a diagonal matrix, and hence the eigenvalues are real numbers.

The full decomposition is computed by the practical QR algorithm in the EISPACK [33] and LAPACK [1] software packages. Instead simultaneous iteration attempts to compute a partial Schur decomposition for $\mathbf{A}$ with the dominant eigenvalues located on the diagonal of $\widehat{\mathbf{T}}_k$.

3. Partial reduction to band Hessenberg form. The first step of the practical QR algorithm is to reduce $\mathbf{A}$ to upper Hessenberg form via Householder transformations. This is done because each step of the QR iteration performed on a Hessenberg matrix involves only order n^2 work. This is in contrast to the order n^3 work that would be required during each step of a QR iteration on $\mathbf{A}$ (assuming that $\mathbf{A}$ is a dense matrix).

Unfortunately, for large eigenvalue problems, Householder transformations cannot be used as they destroy any sparsity or structure in $\mathbf{A}$. We say an eigenvalue problem is large if the dense QR algorithm is prohibitive in storage and/or efficiency. Instead, the Arnoldi reduction [2] only requires knowledge of $\mathbf{A}$ through a matrix-vector product. Moreover, it allows us to sequentially reduce $\mathbf{A}$ to upper Hessenberg form, producing the leading portion of an upper Hessenberg matrix at every step. When the matrix $\mathbf{A}$ is Hermitian, the Lanczos reduction [16] is recovered.

Since our concern is in the solution of eigenvalue problems where $\mathbf{A}$ is not only large but also expensive to apply, block Arnoldi reductions [30, 31] are considered. In many instances, the cost of computing a few matrix-vector products is commensurate with that of one matrix-vector product. Moreover, in exact arithmetic, an unblocked Arnoldi reduction cannot detect the multiplicity of an eigenvalue. A blocksize enables the reduction to compute multiplicities less than or equal to the blocksize.

Let $b > 0$, an integer, be the blocksize and let $\mathbf{E}_j \equiv \begin{bmatrix} \mathbf{e}_{(j-1)b+1} & \cdots & \mathbf{e}_{jb} \end{bmatrix}$ where the jth canonical basis vector is denoted by $\mathbf{e}_j$. We say that

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_j\mathbf{H}_j + \mathbf{F}_j\mathbf{E}_j^T \tag{3.1}$$

is a block Arnoldi reduction of length j when $\mathbf{V}_j^H\mathbf{A}\mathbf{V}_j = \mathbf{H}_j$ is a banded upper Hessenberg matrix, $\mathbf{V}_j^H\mathbf{V}_j = \mathbf{I}_{j\cdot b}$, and $\mathbf{V}_j^H\mathbf{F}_j = \mathbf{0}$. The $j \cdot b$ columns of $\mathbf{V}_j$ are an orthonormal basis for the block Krylov subspace

$$\mathcal{K}_j^b(\mathbf{A}, \mathbf{U}_1) \equiv \text{Span}\{\mathbf{U}_1, \mathbf{A}\mathbf{U}_1, \ldots, \mathbf{A}^{j-1}\mathbf{U}_1\},$$

where $\mathbf{U}_1$ is a full rank matrix with b columns. Note that if $\mathbf{A} = \mathbf{A}^H$, then $\mathbf{H}_j$ is a block tridiagonal matrix.

In order to introduce notation that will be needed, we rewrite (3.1) as

$$\mathbf{A}\,\mathbf{V}_j = \begin{bmatrix} \mathbf{U}_1 & \cdots & \mathbf{U}_j \end{bmatrix} \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \cdots & \mathbf{G}_{1,j} \\ \mathbf{G}_{2,1} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{G}_{j,j-1} & \mathbf{G}_{j,j} \end{bmatrix} + \mathbf{U}_{j+1}\mathbf{G}_{j+1,j}\mathbf{E}_j^T,$$

where $\mathbf{U}_{j+1}\mathbf{G}_{j+1,j}$ is the QR factorization of $\mathbf{F}_j$. The reduction is advanced by one step (or its length incremented) by the following three operations:

1. $\mathbf{W} = \mathbf{A}\mathbf{U}_{j+1}$.
2. $\mathbf{G}_{i,j+1} = \mathbf{U}_i^H\mathbf{W} \;\; i = 1, \ldots, j+1$.
3. $\mathbf{F}_{j+1} = \mathbf{W} - \sum_{i=1}^{j+1} \mathbf{U}_i\mathbf{G}_{i,j+1}$.

A direct implementation of the second and third steps will not, in general, produce an orthogonal set of Arnoldi vectors. We refer the reader to [17] for details on an efficient and robust implementation. See [12] for references and information on a block Lanczos reduction implemented via a three-term block recurrence.

3.1. Computing an approximate partial Schur decomposition. Suppose that $\mathbf{H}_j\mathbf{Z} = \mathbf{Z}\mathbf{T}$ is a Schur decomposition ordered so that the eigenvalues of $\mathbf{T}$ are in descending order of magnitude along the diagonal. Postmultiplying (3.1) with $\mathbf{Z}$ gives

$$\|\mathbf{A}\mathbf{V}_j\mathbf{Z} - \mathbf{V}_j\mathbf{Z}\mathbf{T}\| = \|\mathbf{F}_j\mathbf{E}_j^T\mathbf{Z}\| = \|\mathbf{G}_{j+1,j}\mathbf{E}_j^T\mathbf{Z}\|. \tag{3.2}$$

Because $\mathbf{E}_j^T\mathbf{Z}$ is a matrix with $j \cdot b$ columns consisting of the last b rows of $\mathbf{Z}$, the quality of an approximate partial Schur decomposition is determined because

$$\|\mathbf{F}_j\mathbf{E}_j^T\mathbf{Z}\left[\begin{array}{cccc}\mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_k\end{array}\right]\| \equiv \|\mathbf{F}_j\mathbf{E}_j^T\mathbf{Z}_k\|.$$

If this quantity is small, then an approximate partial Schur decomposition of order k for the dominant invariant subspace is computed. We now establish this assertion. Suppose that $\mathbf{H}_j\mathbf{Z}_k = \mathbf{Z}_k\widehat{\mathbf{T}}_k$ is an order k partial Schur decomposition for $\mathbf{H}_j$; then

$$(\mathbf{A} + \mathbf{M})\mathbf{V}_j\mathbf{Z}_k = \mathbf{V}_j\mathbf{Z}_k\widehat{\mathbf{T}}_k,$$

where

$$\|\mathbf{M}\| = \| - \mathbf{F}_j\mathbf{E}_j^T\mathbf{Z}_k(\mathbf{V}_j\mathbf{Z}_k)^H\| \leq \|\mathbf{G}_{j+1,j}\mathbf{E}_j^T\mathbf{Z}_k\|,$$

implying that we have computed an exact partial Schur decomposition for a matrix near $\mathbf{A}$.

If approximate eigenvectors are of interest, they can be computed from the partial Schur decomposition $\mathbf{H}_j\mathbf{Z}_k = \mathbf{Z}_k\widehat{\mathbf{T}}_k$ because

$$\mathbf{A}\mathbf{V}_j\mathbf{Z}_k\mathbf{y} - \mathbf{V}_j\mathbf{Z}_k\mathbf{y}\theta = \mathbf{F}_j\mathbf{E}_j^T\mathbf{Z}_k\mathbf{y}, \tag{3.3}$$

where $\widehat{\mathbf{T}}_k\mathbf{y} = \mathbf{y}\theta$. We call $\mathbf{V}_j\mathbf{Z}_k\mathbf{y}$ a Ritz vector and θ a Ritz value. Note that the first Schur vector is always an eigenvector.

4. Connection with nonstationary simultaneous iteration. There is a well-known connection between nonstationary simultaneous iteration and the QR algorithm. This section will show a similar relationship between an IRAM and nonstationary simultaneous iteration. The following elementary but technical result is needed for this relationship. The result is a generalization of the special case $\psi(\lambda) = \lambda$ shown in [19]. A similar result was proved in Lemma 1 of [23] for the Lanczos reduction.

LEMMA 4.1. *Suppose that an integer p satisfies $1 \leq p < m$, and let $r = m - p$. Let $\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{F}_m\mathbf{E}_m^T$ be a length $r + p$ Arnoldi reduction, where $\mathbf{H}_m$ is an unreduced band upper Hessenberg matrix. If*

$$\psi_p(\lambda) = \prod_{i=1}^{p}(\lambda - \tau_i),$$

then

$$\psi_p(\mathbf{A})\mathbf{V}_m = \mathbf{V}_m\psi_p(\mathbf{H}_m) + \sum_{j=1}^{p}\psi_{j+1,p}(\mathbf{A})\mathbf{F}_m\mathbf{E}_m^T\psi_{j-1}(\mathbf{H}_m), \tag{4.1}$$

where $\psi_{j,p}(\lambda) = \prod_{i=j}^{p}(\lambda - \tau_i)$.

Moreover,

$$\psi_p(\mathbf{A})\mathbf{V}_r = \mathbf{V}_m\psi_p(\mathbf{H}_m)\begin{bmatrix} \mathbf{E}_1 & \cdots & \mathbf{E}_r \end{bmatrix}. \tag{4.2}$$

Proof. The proof is by mathematical induction. Define $m \equiv r+p$. The subscripts are suppressed on $\mathbf{V}_m$ and $\mathbf{H}_m$ for the proof. Since $\psi_1(\mathbf{A})\mathbf{V} = \mathbf{V}\psi_1(\mathbf{H}) + \mathbf{F}_m\mathbf{E}_m^T$, where $\psi_1(\lambda) = \lambda - \tau_1$, the base case for $p = 1$ is established. Assume the lemma's truth for polynomials $\psi_j(\lambda)$ of degree $j \le p$. Let $\psi_{p+1}(\lambda) = (\lambda - \tau_{p+1})\psi_p(\lambda)$. Using the induction hypothesis leads to

$$\begin{aligned}
\psi_{p+1}(\mathbf{A})\mathbf{V} &= (\mathbf{A} - \tau_{p+1}\mathbf{I})\psi_p(\mathbf{A})\mathbf{V} \\
&= (\mathbf{A} - \tau_{p+1}\mathbf{I})\left\{\mathbf{V}\psi_p(\mathbf{H}) + \sum_{j=1}^{p}\psi_{j+1,p}(\mathbf{A})\mathbf{F}_m\mathbf{E}_m^T\psi_{j-1}(\mathbf{H})\right\} \\
&= \mathbf{V}(\mathbf{H} - \tau_{p+1}\mathbf{I})\psi_p(\mathbf{H}) + \mathbf{F}_m\mathbf{E}_m^T\psi_p(\mathbf{H}) \\
&\quad + (\mathbf{A} - \tau_{p+1}\mathbf{I})\sum_{j=1}^{p}\psi_{j+1,p}(\mathbf{A})\mathbf{F}_m\mathbf{E}_m^T\psi_{j-1}(\mathbf{H}) \\
&= \mathbf{V}\psi_{p+1}(\mathbf{H}) + \sum_{j=1}^{p+1}\psi_{j+1,p+1}(\mathbf{A})\mathbf{F}_m\mathbf{E}_m^T\psi_{j-1}(\mathbf{H}),
\end{aligned}$$

which establishes (4.1).

Since $\mathbf{H}$ is unreduced, $\psi_{j-1}(\mathbf{H})$ is a band Hessenberg matrix of lower bandwidth $(j-1)\cdot b$. Hence $\mathbf{E}_i^T\psi_{j-1}(\mathbf{H})\mathbf{E}_l = \mathbf{0}$ for $l = 1, \ldots, m-j$, and the last matrix on the right-hand side of (4.1) is zero through its first $r \cdot b$ columns and (4.2) is established. ∎

In other words, (4.2) shows that $\psi_p(\mathbf{A})$ applied to the first $r \cdot b$ columns of $\mathbf{V}_m$ is equivalent to the basis representation given by the first $r \cdot b$ columns of $\mathbf{V}_m\psi_p(\mathbf{H}_m)$.

Suppose that p steps of the QR algorithm are performed on $\mathbf{H}_m$ with $p < m$ shifts $\tau_1, \ldots, \tau_p$ resulting in

$$\mathbf{H}_m\mathbf{W} = \mathbf{W}\mathbf{H}_m^+. \tag{4.3}$$

Note that $\mathbf{W}$ is a Hessenberg matrix of lower bandwidth $p \cdot b$ because it is a product of p unitary matrices each of lower bandwidth b (each computed during a step of the QR algorithm). If $\mathbf{W}_1$ denotes the initial $r \cdot b$ columns of $\mathbf{W}$, $\mathbf{W}_2$ denotes the next b columns, and $\mathbf{W}_3$ denotes the remaining columns, equating the first $r \cdot b$ columns of (4.3) results in

$$\mathbf{H}_m\mathbf{W}_1 \equiv \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \mathbf{W}_3 \end{bmatrix}\begin{bmatrix} \mathbf{H}_r^+ \\ \mathbf{G}_{r+1,r}^+\mathbf{E}_r^T \\ \mathbf{0} \end{bmatrix}. \tag{4.4}$$

Postmultiplying (3.1) with $\mathbf{W}_1$ and using (4.4), we obtain

$$\begin{aligned}
\mathbf{A}\mathbf{V}_m\mathbf{W}_1 &= \mathbf{V}_m\mathbf{H}_m\mathbf{W}_1 + \mathbf{F}_m\mathbf{E}_m^T\mathbf{W}_1 \\
&= \mathbf{V}_m\mathbf{W}_1\mathbf{H}_r^+ + \mathbf{V}_m\mathbf{W}_2\mathbf{G}_{r+1,r}^+\mathbf{E}_r^T + \mathbf{F}_m\mathbf{E}_m^T\mathbf{W}_1 \\
&\equiv \mathbf{V}_r^+\mathbf{H}_r^+ + \mathbf{F}_r^+\mathbf{E}_r^T,
\end{aligned} \tag{4.5}$$

where

$$\mathbf{V}_r^+ \equiv \mathbf{V}_m\mathbf{W}_1 \quad \text{and} \quad \mathbf{F}_r^+ \equiv \mathbf{V}_m\mathbf{W}_2\mathbf{G}_{r+1,r}^+ + \mathbf{F}_m\mathbf{E}_m^T\mathbf{W}_1\mathbf{E}_r. \tag{4.6}$$

Note the use of the identity $\mathbf{E}_m^T\mathbf{W}_1 = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{E}_m^T\mathbf{W}_1\mathbf{E}_r \end{bmatrix}$ in (4.6).

The following theorem establishes a direct relationship between nonstationary simultaneous iteration and the QR algorithm. The theorem is a partial or truncated version of Theorem 3.1 in [37, p. 353].

THEOREM 4.2. *Assume the hypothesis of Lemma* 4.1 *and the notation in* (4.4)–(4.6). *If the QR algorithm computed on* $\mathbf{H}_m$ *with the* p *shifts* $\tau_1, \ldots, \tau_p$ *results in* $\mathbf{H}_m\mathbf{W} = \mathbf{W}\mathbf{H}_m^+$, *then*

$$\psi_p(\mathbf{A})\mathbf{V}_r = \mathbf{V}_r^+\widehat{\mathbf{R}}_r, \tag{4.7}$$

where $\widehat{\mathbf{R}}_r$ *is an upper triangular matrix of order* $r \cdot b$.

Proof. From Lemma 4.1 it suffices to show that the QR factorization of

$$\psi_p(\mathbf{H}_m)\begin{bmatrix} \mathbf{E}_1 & \cdots & \mathbf{E}_r \end{bmatrix} = \mathbf{W}_1\widehat{\mathbf{R}}_r$$

for some upper triangular $\widehat{\mathbf{R}}_r$ of order $r \cdot b$. But this is a consequence of the link between the QR algorithm and nonstationary simultaneous iteration because

$$\psi_p(\mathbf{H}_m) = \mathbf{W}\mathbf{R}$$

is a QR factorization [37, p. 353]. The result follows from equating the initial $r \cdot b$ columns of this equality and letting $\widehat{\mathbf{R}}_r$ denote the leading submatrix of order $r \cdot b$ of $\mathbf{R}$. □

The theorem allow us to link simultaneous iteration with an IRAM. If $p = 1$, $\psi_1(\lambda) = \lambda$, $r = 1$, and $b > 1$, then (4.7) becomes $\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1^+\widehat{\mathbf{R}}_1$ and simultaneous iteration is recovered. From (4.5) it follows that for this specific choice of polynomial an IRAM is simultaneous iteration in disguise because $\mathbf{V}_1^+$ is computed. Two remarks are in order.

First, we note that (4.3) through (4.6) define a restart of the Arnoldi reduction via a QR algorithm. An IRAM is a sequence of implicit restarts that are terminated when the partial Schur decomposition of interest is sufficiently well approximated. The restart is implicit because as Theorem 4.2 demonstrates, polynomials in $\mathbf{A}$ can be implicitly applied to $\mathbf{V}_m$ via an application to $\mathbf{H}_m$.

Second, if $p = m$, implicitly restarting with $m - 1$ shifts produces

$$\mathbf{A}\mathbf{U}_1^+ = \mathbf{U}_1^+\mathbf{H}_1^+ + \mathbf{F}_1^+. \tag{4.8}$$

If $\mathbf{Q}\mathbf{R} = \mathbf{H}_1^+ - \tau_m\mathbf{I}$, then

$$(\mathbf{A} - \tau_m\mathbf{I})\mathbf{U}_1^+ = \mathbf{U}_1^+\mathbf{Q}\mathbf{R} + \mathbf{F}_1^+. \tag{4.9}$$

The right-hand side of (4.9) defines a new starting block of vectors (after orthogonalization) for a subsequent block Arnoldi reduction. Thus we can implicitly apply polynomials of degree m in $\mathbf{A}$. This bit of cleverness was first established in [3].

5. Convergence of an IRAM. Sorensen [34] gave some convergence results for an IRAM (blocksize of one). For nonsymmetric $\mathbf{A}$, a linear rate of convergence was given for a fixed $\psi_p(\cdot)$ per restart. He also showed that for symmetric $\mathbf{A}$, using the unwanted $m-k$ eigenvalues as shifts during each implicit restart resulted in convergence to k eigenvalues of $\mathbf{A}$.

Traditional convergence theory [13, 14, 27] for Arnoldi reductions investigates the quality of $\mathcal{K}_j^b(\mathbf{A}, \mathbf{U}_1)$ to approximate eigenvectors of $\mathbf{A}$ as j increases. However, with the connection between an IRAM and nonstationary simultaneous iteration in hand, a more elegant and powerful theory is possible. A comprehensive geometric convergence theory for the shifted QR algorithm is presented by Watkins and Elsner [40] within the more general framework of generic *GR algorithms*. A GR algorithm is a generalization of the QR algorithm where the QR factorization of $\mathbf{A}-\tau\mathbf{I}$ is replaced with a GR factorization, where $\mathbf{G}$ is a nonsingular matrix. The convergence theory is based on the idea that a GR algorithm is a nested sequence of nonstationary simultaneous iterations.

Theorem 5.1 in [40] discusses the convergence of nonstationary simultaneous iteration. Recall the notation established in sections 2 and 4. In addition, let $\phi_i(\cdot)$ denote $\psi_p(\cdot)$ (the polynomial in $\mathbf{A}$ implicitly applied) used at the ith restart of an IRAM.

THEOREM 5.1. *Let $\mathbf{A}$ be a simple matrix of order n. Let $r \cdot b \ll n$, where b and r are positive. Define the invariant subspaces $\mathcal{Z} = Span(\mathbf{x}_1, \ldots, \mathbf{x}_{r\cdot b})$ and $\mathcal{U} = Span(\mathbf{x}_{r\cdot b+1}, \ldots, \mathbf{x}_n)$. Let $\Phi_i = \phi_i \cdots \phi_1$ and suppose that $\Phi_i(\lambda_j) \neq 0$ for $j = 1, \ldots, r\cdot b$ and let*

$$\rho_i = \frac{\max_{j=r\cdot b+1,\ldots,n} |\Phi_i(\lambda_j)|}{\min_{j=1,\ldots,r\cdot b} |\Phi_i(\lambda_j)|}. \tag{5.1}$$

If $\mathcal{V}$ is a subspace of dimension $r \cdot b$ satisfying $\mathcal{V} \cap \mathcal{U} = \{0\}$, then there exists a constant C such that

$$dist(\Phi_i(\mathbf{A})\mathcal{V}, \mathcal{Z}) \leq C\rho_i$$

for all i.

Proof. See Theorem 5.1 established in [40]. □

We remark that the hypothesis on $\mathbf{A}$ is done only for notational and elaborative purposes. The paper [40] considers the more general case of an arbitrary matrix.

The distance between the subspaces [8, 10] $\mathcal{V}$ and $\mathcal{Z}$ may be shown to be equal to $\sqrt{1-\sigma_{\min}^2(\mathbf{V}^H\mathbf{Z})}$ where the columns of $\mathbf{V}$ and $\mathbf{Z}$ provide an orthonormal basis for $\mathcal{V}$ and $\mathcal{Z}$ and $\sigma_{\min}(\cdot)$ denotes the minimum singular value. For increasing values of i, the approximating subspace $\Phi_i(\mathbf{A})\mathcal{V}$ aligns itself with $\mathcal{Z}$, and hence the distance between the two subspaces goes to zero.

An expression for the constant C is given by Watkins and Elsner. They show that

$$C = \|\mathbf{X}\|\,\|\mathbf{X}^{-1}\|\frac{\sqrt{1-\sigma_{\min}^2(\mathbf{V}^H\mathbf{Z})}}{\sigma_{\min}(\mathbf{V}^H\mathbf{Z})},$$

where $\mathbf{X}$ is the matrix of eigenvectors for $\mathbf{A}$. The size of C is large whenever the eigenvectors form an ill-conditioned basis and/or the starting subspace $\mathcal{V}$ is nearly orthonormal to $\mathcal{Z}$.

Because of the relationship between an IRAM and nonstationary simultaneous iteration established in the last section, the $\text{dist}(\Phi_i(\mathbf{A})\mathcal{V}, \mathcal{Z}) \to 0$ and the eigenvalues

of the leading block of $\mathbf{H}_m$ of order $r \cdot b$ computed after i restarts are approximations to $\lambda_1, \ldots, \lambda_{r \cdot b}$. The rate of convergence given by Theorem 5.1 is determined by how well the max-min problem (5.1) is solved. This problem cannot, in general, be solved. Section 6 discusses and reviews practical issues related to convergence.

6. Some practical issues. The question of determining a shift strategy that leads to a provable rapid rate of convergence is a difficult problem that continues to be researched. However, it is clear from Theorem 5.1 that a polynomial Φ_i that minimizes ρ_i for the discrete max-min problem (5.1) is required. Unfortunately, this polynomial is impossible to compute because the eigenvalues are not known. A common alternative is to replace the max-min problem posed on the discrete set of eigenvalues with a problem posed on a convex region containing the unwanted eigenvalues. Before we consider shifting schemes based on this approach, we review two simple shifting schemes.

A special case of an iteration for which the discrete max-min problem (5.1) can be solved is when $\phi_i(\lambda) = \lambda^i$. In this special case, $\rho_i = |\lambda_k/\lambda_{k+1}|^{i \cdot p}$ and by hypothesis on the ordering of the eigenvalues, $|\lambda_{k-1}/\lambda_k| < 1$, and so convergence to the dominant invariant subspace is assured.

More general nontrivial stationary iterations where $\Phi_i(\cdot) = \phi_1(\cdot)^i$ and ϕ_1 is a degree p polynomial converge at the linear rate

$$\frac{\max_{j=r \cdot b+1,\ldots,n} |\Phi_i(\lambda_j)|}{\min_{j=1,\ldots,r \cdot b} |\Phi_i(\lambda_j)|}$$

if $|\phi_1(\lambda_i)| \geq |\phi_1(\lambda_k)|$ for $i = 1, \ldots, k$ and $|\phi_1(\lambda_k)| > |\phi_1(\lambda_i)|$ for $i = k+1, \ldots, n$. This is a slight generalization of Theorem 5.1 in [34] easily established by Theorem 5.1 in the paper.

We now replace the max-min problem posed on the discrete set of eigenvalues with a problem posed on a convex region that contains the unwanted eigenvalues. A first step in reducing ρ_i is to bound the distance from the k dominant eigenvalues of $\mathbf{A}$ to the set of unwanted eigenvalues. If we assume that the dominant eigenvalues of $\mathbf{A}$ are all to the right (respectively, to the left) of λ_k, the denominator of (5.1) is bounded if all the τ_i's can be enclosed in a convex region containing the unwanted eigenvalues not intersecting the region containing the dominant eigenvalues. If all the shifts lie in this latter region, then we can attempt to approximately solve

$$\rho_i = \frac{\max_{\mathcal{S}} |\Phi_i(\lambda)|}{\min_{\mathcal{W}} |\Phi_i(\lambda)|},$$

where $\mathcal{S}$ and $\mathcal{W}$ are, respectively, the unwanted and wanted regions.

When $\mathbf{A}$ is a real matrix, a standard approach is to set $\mathcal{S}$ to be an ellipse and then using the techniques described in [29, pp. 219–239] compute the roots of a Chebyshev polynomial. Saad shows that the per restart convergence to $\mathcal{V}$ is

$$\max_{i=1,\ldots,k} \frac{\alpha + \sqrt{\alpha^2 - \gamma^2}}{\alpha_i + \sqrt{\alpha_i^2 - \gamma_i^2}},$$

where $\mathcal{S}$ is enclosed by an ellipse with center δ, focal distance γ, major axis α; and γ_i, α_i are the focal distance and major axis of the ellipse including $\mathcal{S}$ and λ_i. The resulting rate of convergence can be far superior to the rate $|\lambda_{k-1}/\lambda_k|$ given by standard subspace iteration.

A second, more recent approach is the exact shift scheme proposed by Sorensen. This is the default scheme used by ARPACK [18]. An exact scheme uses a number of the unwanted eigenvalues of $\mathbf{H}_m$ for the $\psi_p(\cdot)$ implicitly applied per restart. The scheme is motivated by the observation that the scheme retains only the approximations to the wanted eigenvalues. As explained in [18, p. 71], the value of r is increased from the initial value of k as approximations to the eigenvalues of $\mathbf{A}$ satisfy the acceptance criterion. This has the effect of decreasing the numerator in (5.1) (see [35, 41] for similar conclusions). Unfortunately, there is no rigorous convergence theory giving a rate of convergence, but in practice the exact scheme has proven to be robust and gives slightly better results than using the zeros of Chebyshev polynomials.

Recent papers have elucidated the virtues of other restart polynomials. For non-Hermitian matrices $\mathbf{A}$, harmonic Ritz values [21, 20, 23, 32] and refined shifts [15] are interesting schemes that require further analysis. For symmetric eigenvalue problems, the roots of Leja polynomials [3, 5, 4] are successfully used for small m along with a deflation scheme that allows r to be increased from a value of one as satisfactory eigenvalue-eigenvector approximations emerge.

7. Numerical experiments. We present results for computing the 4 dominant eigenvalues and invariant subspace for the two-dimensional model convection-diffusion problem

$$-\Delta\mathbf{u}(x,y) + 4\mathbf{u}_x(x,y) + .5\mathbf{u}_y(x,y) = \lambda\mathbf{u}(x,y)$$

on the unit square $[0,1] \times [0,1]$ with zero boundary conditions. The problem is discretized by using centered finite differences where the mesh size $h = 1/(50+1)$ results in a matrix of order 2,500. Table 7.1 lists the results of some MATLAB experiments (IEEE double precision floating point arithmetic is used). "Subit" refers to simultaneous iteration along with a projection at each restart for unscrambling the Schur vector approximations. The purpose of these experiments is to demonstrate the dramatic difference in performance that can be achieved when iterating on subspaces of equal dimension.

Each computed dominant partial Schur decomposition of order 4 gave a residual no larger than 10^{-3}, and all experiments produced Ritz values that agreed to at least four significant digits. The first three lines of the table give results in terms of the number of restarts and matrix-vector products used when using $m \cdot b = 16$ vectors. The next five lines list the same output when $m \cdot b = 24$, and the final line indicates the total number of matrix-vector products needed when $b = 1$ and m is increased until the partial Schur decomposition is approximated. This final experiment gives an indication of the minimum number of matrix-vector products needed. When $b = 1$, IRAM uses the strategy for increasing r from k as the eigenvalues satisfy the acceptance criterion to a maximum of 8; for $b = 2$ and $b = 4$, the value of r stayed fixed at $m - 4$ and for $b = 3$ was fixed at $r = 5$.

The following are some other conclusions resulting from the experiments of Table 7.1:

- The simultaneous iteration results always performed an extra m matrix-vector products per restart needed for the projection step. This projection is not needed at every restart. Hence the minimum number of matrix-vector products needed is half the number listed.
- Decreasing the blocksize requires a larger value of acceptance tolerance to be used so that the four dominant eigenvalues are computed to four significant digits.

TABLE 7.1
Computing the partial Schur decomposition corresponding to the 4 *dominant eigenvalues of a convection-diffusion matrix of order* 2,500.

	k	m	Matvec	Restarts	b
Subit	4	1	17,664	552	16
IRAM	4	16	253	23	1
IRAM	4	8	288	35	2
Subit	4	1	16,800	350	24
IRAM	4	24	230	12	1
IRAM	4	12	264	16	2
IRAM	4	8	330	35	3
IRAM	4	6	376	45	4
IRAM	4	200	199	0	1

- If $\mathbf{A}$ is efficiently applied to the block of vectors, then the number of matrix-vector products should be divided by the blocksize. Hence simultaneous iteration may be feasible.
- Decreasing the acceptance tolerance and/or increasing k (number of eigenvalues to compute) hinders the efficiency of simultaneous iteration. This effect is less pronounced as b decreases. IRAM with a blocksize of one has an amazing ability to compute partial Schur decompositions with small backward error.

In summary, if storage considerations and/or the cost of orthogonalization prevent a large Arnoldi reduction from being computed, an IRAM does not require a substantial number more matrix-vector products. This was also noted by Morgan [22].

8. Conclusions. We showed that restarted Arnoldi methods are equivalent to nonstationary simultaneous iteration methods by demonstrating a relationship with implicitly restarted Arnoldi methods. Convergence theory and numerical experiments confirmed that $\Phi_i(A)\mathcal{K}_r^b(\mathbf{A}, \mathbf{U}_1)$ tends to the dominant invariant subspace of order $r \cdot b$ at a substantially faster rate than $\mathbf{A}^i\mathcal{K}_1^{r\cdot b}(\mathbf{A}, \mathbf{U}_1)$ when appropriate restarting polynomials are employed.

Acknowledgments. I thank Mark Embree, Steve Wright, and the referees for improvements to the initial manuscript. I also thank Chris Beattie for suggesting Lemma 4.1 to me during a workshop held in the spring of 1995.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. D. CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1995.

[2] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[3] J. BAGLAMA, D. CALVETTI, AND L. REICHEL, *Iterative methods for the computation of a few eigenvalues of a large symmetric matrix*, BIT, 36 (1996), pp. 400–440.

[4] J. BAGLAMA, D. CALVETTI, AND L. REICHEL, *Fast Leja points*, Electron. Trans. Numer. Anal., 7 (1998), pp. 124–140.

[5] J. BAGLAMA, D. CALVETTI, L. REICHEL, AND A. RUTTAN, *Computation of a few close eigenvalues of a large matrix with application to liquid crystal modeling*, J. Comput. Phys., 146 (1998), pp. 203–226.

[6] Z. Bai and G. W. Stewart, *SRRIT — A FORTRAN subroutine to calculate the dominant invariant subspace of a nonsymmetric matrix*, ACM Trans. Math. Software, 23 (1997), pp. 494–513.
[7] F. L. Bauer, *Das Verfahren der Treppeniteration und verwandte Verfahren zur Losung algebraischer Eigenwertproblem*, Z. Angew. Math. Phys., 8 (1957), pp. 214–235.
[8] F. Chatelin, *Eigenvalues of Matrices*, Wiley, Chichester, 1993.
[9] I. S. Duff and J. A. Scott, *Computing selected eigenvalues of large sparse unsymmetric matrices using subspace iteration*, ACM Trans. Math. Software, 19 (1993), pp. 137–159.
[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University, Baltimore, MD, 1996.
[11] G. H. Golub and J. H. Wilkinson, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Rev., 18 (1976), pp. 578–619.
[12] R. G. Grimes, J. G. Lewis, and H. D. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.
[13] Z. X. Jia, *The convergence of generalized Lanczos methods for large unsymmetric eigenproblems*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 843–862.
[14] Z. Jia, *Generalized block Lanczos methods for large unsymmetric eigenproblems*, Numer. Math., 80 (1998), pp. 239–266.
[15] Z. Jia, *Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm*, Linear Algebra Appl., 287 (1998), pp. 191–214.
[16] 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. 255–282.
[17] R. Lehoucq and K. Maschhoff, *Block Arnoldi method*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM, Philadelphia, 2000, pp. 185–189.
[18] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
[19] K. Meerbergen and A. Spence, *Implicitly restarted Arnoldi with purification for the shift–invert transformation*, Math. Comp., 218 (1997), pp. 667–689.
[20] R. Morgan and M. Zeng, *Harmonic projection methods for large non-symmetric eigenvalue problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 33–55.
[21] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154/156 (1991), pp. 289–309.
[22] R. B. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.
[23] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–134.
[24] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[25] H. Rutihauser, *Computational aspects of F.L. Bauer's simultaneous iteration method for symmetric matrices*, Numer. Math., 13 (1969), pp. 4–13.
[26] H. Rutihauser, *Simultaneous iteration method for symmetric matrices*, Numer. Math., 16 (1970), pp. 205–223.
[27] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
[28] Y. Saad, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.
[29] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted, New York, 1992.
[30] M. Sadkane, *A block Arnoldi–Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.
[31] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices*, ACM Trans. Math. Software, 21 (1995), pp. 432–475.
[32] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
[33] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines—EISPACK Guide*, 2nd ed., Lecture Notes in Comput. Sci. 6, Springer-Verlag, Berlin, 1976.
[34] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[35] 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.
[36] G. W. Stewart, *Accelerating the orthogonal iteration for the eigenvectors of a Hermitian matrix*, Numer. Math., 13 (1969), pp. 362–376.
[37] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, San Diego, CA, 1973.
[38] G. W. Stewart, *Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices*, Numer. Math., 25 (1976), pp. 123–136.
[39] D. S. Watkins, *Understanding the QR algorithm*, SIAM Rev., 24 (1982), pp. 427–440.
[40] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.
[41] G. Zhang, *Modified explicitly restarted Lanczos algorithm*, Comput. Phys. Comm., 109 (1998), pp. 27–33.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 563–580
© 2001 Society for Industrial and Applied Mathematics

ON THE SIMILARITY INVARIANT POLYNOMIALS OF MATRICES AND PRINCIPAL SUBMATRICES*

E. MARQUES DE SÁ† AND JOÃO DAVID VIEIRA‡

Abstract. We consider the problem of finding necessary and sufficient conditions for the existence of a real $m \times m$ matrix $\mathbb{B}$, over a field $\mathbb{F}$, having a principal $n \times n$ submatrix $\mathbb{A}$ such that each similarity invariant polynomial of $\mathbb{A}$ and each similarity invariant polynomial of $\mathbb{B}$ lie inside prescribed divisibility intervals (one interval for each similarity invariant polynomial). The main result gives a complete solution to this problem in case all prescribed polynomials (the extreme elements of the divisibility intervals) have all prime factors (over $\mathbb{F}$) of degree 1 or 2. In particular, this solves the problem for algebraically closed fields and for the field of real numbers.

Our arguments are of the algebraic and discrete kind. We use a localization technique to reduce the problem to a system of diophantine inequalities involving the degrees of the prime factors of the given polynomials and the degrees of the elementary divisors of the matrices under consideration. Feasibility conditions for this system are then determined for the particular case already described. In the last section we give some additional comments on proof techniques and state some consequences of the main result.

Key words. matrices, similarity, invariant polynomials, elementary divisors, localization

AMS subject classifications. 15A21, 15A39

PII. S0895479800367427

1. Introduction.

In what follows, $\mathbb{F}$ denotes an arbitrary field and $\mathbb{F}[\lambda]$ the ring of polynomials in the variable λ. A *polynomial chain (of length n)* is an n-tuple, $\gamma = (\gamma_1, \ldots, \gamma_n)$, where the γ_i's are monic polynomials such that $\gamma_i \,|\, \gamma_{i+1}$ for $1 \leqslant i < n$. The zero polynomial is also considered as monic. Very often it will be convenient to consider coordinates γ_i for $i > n$ and also for $i < 1$; our convention is

$$\begin{array}{ll} \gamma_i = 0 & \text{if } i > \text{the length of the chain;} \\ \gamma_i = 1 & \text{if } i < 1\,. \end{array} \tag{1}$$

Thus, any relation like $\gamma_i \,|\, \delta_i$ involving another chain δ of length m trivially holds for $i < 1$ and for $i > \max\{n, m\}$. Besides, if such a relation is true for *all* i, we have $\delta_i = 0$ for $i > n$.

The similarity invariant polynomials, $\alpha_1 \,|\, \alpha_2 \,|\, \ldots \,|\, \alpha_n$, of an $n \times n$ matrix $\mathbb{A}$ over $\mathbb{F}$ is the main example of a polynomial chain $(\alpha_1, \ldots, \alpha_n)$. Note that α_n is the minimal polynomial of $\mathbb{A}$ and $\alpha_1 \cdots \alpha_n$ is the characteristic polynomial of $\mathbb{A}$. According to the so-called *interlacing inequalities theorem for similarity invariants* [6, 7], given another $\mathbb{F}$-matrix $\mathbb{B}$ of dimension $m \times m$, $m \geqslant n$, with similarity invariant polynomials $\beta_1 \,|\, \ldots \,|\, \beta_m$, $\mathbb{B}$ is similar to a matrix having $\mathbb{A}$ as a principal submatrix if and only if the relations

$$\beta_i \,|\, \alpha_i \,|\, \beta_{i+h}\,, \tag{2}$$

*Received by the editors February 8, 2000; accepted for publication (in revised form) by P. Van Dooren May 25, 2001; published electronically November 13, 2001.

http://www.siam.org/journals/simax/23-2/36742.html

†Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal (emsa@mat.uc.pt). This author is a member of the research unit CMUC/FCT (Centro de Matemática da Universidade de Coimbra/Fundação para a Ciência e a Tecnologia) and was partially supported by Praxis XXI Mat/485/94 and Project 574/94 of Fundação Luso-Americana para o Desenvolvimento.

‡Departamento de Matemática, Universidade de Aveiro, 3800 Aveiro, Portugal (jdvieira@mat.ua.pt). This author is a member of the research unit UI&D Matemática e Aplicações da Universidade de Aveiro and was partially supported by CMUC/FCT.

where h denotes the integer $2m-2n$, hold for all i. These are known as the *interlacing relations for invariant factors*.

This result has been extended in several directions. One interesting source of related research arises in linear control theory and concerns the relationship between the invariant factors and controllability indices of pairs of matrices and the invariant factors and controllability indices of some of its submatrices. A generalized version of this kind of problem, also studied in detail in the literature, is the relationship between the invariant factors and Kronecker indices of rectangular matrix pencils and the invariant factors and Kronecker indices of some of its subpencils (see, e.g., [1, 2, 3, 8, 9, 10] and the references therein).

In [5] we generalized in a different direction the interlacing theorem for similarity invariant polynomials of square matrices $\mathbb{B}$ and those of its principal submatrices $\mathbb{A}$. Instead of prescribing *all* similarity invariants of $\mathbb{A}$ and $\mathbb{B}$, we may prescribe only *some* of them and let all others run freely; this gives rise to many interesting problems, which are very difficult to solve in general. For example, we may ask for conditions on $m+1$, given polynomials $\beta_1, \dots, \beta_m$ and α, that ensure the existence of a matrix $\mathbb{B}$ with similarity invariant polynomials $\beta_1, \dots, \beta_m$ and having an $n \times n$ principal submatrix with minimal polynomial α. (This and other problems of this kind are solved in [5] for algebraically closed fields.)

A generalization of this idea follows.

PROBLEM TO BE CONSIDERED. *Given $m+n$ sets $\mathbf{A}_1, \dots, \mathbf{A}_n$, $\mathbf{B}_1, \dots, \mathbf{B}_m$ of monic polynomials, find conditions equivalent to the existence of an $m \times m$ matrix $\mathbb{B}$ over the field $\mathbb{F}$, with similarity invariant factors $\beta_1, \dots, \beta_m$, having a principal $n \times n$ submatrix $\mathbb{A}$, with similarity invariant factors $\alpha_1, \dots, \alpha_n$, such that for all i and j,*

$$\alpha_i \in \mathbf{A}_i \quad \textit{and} \quad \beta_j \in \mathbf{B}_j\,.$$

As a matter of fact, we are interested in the case when the sets $\mathbf{A}_i$, $\mathbf{B}_j$ are *divisibility intervals*. To be precise, we are given two nonnegative integers m and n, such that $n \leqslant m$, and the four polynomial chains

$$\begin{aligned} &\underline{\alpha} = (\underline{\alpha}_1, \dots, \underline{\alpha}_n), \quad && \overline{\alpha} = (\overline{\alpha}_1, \dots, \overline{\alpha}_n), \\ &\underline{\beta} = (\underline{\beta}_1, \dots, \underline{\beta}_m), \quad && \overline{\beta} = (\overline{\beta}_1, \dots, \overline{\beta}_m). \end{aligned} \tag{3}$$

In [5] and in this paper we consider the problem above in the case that $\mathbf{A}_i$ (resp., $\mathbf{B}_j$) is the set of all monic polynomials φ such that $\underline{\alpha}_i \,|\, \varphi \,|\, \overline{\alpha}_i$ (resp., $\underline{\beta}_j \,|\, \varphi \,|\, \overline{\beta}_j$). In [5] we solved this problem under some restrictions on the prime factorization (over $\mathbb{F}$) of the polynomials in (3); in particular, [5] solves the problem when the polynomials in (3) split into linear factors over $\mathbb{F}$. (This is always the case when $\mathbb{F}$ is algebraically closed.)

In the main result of this paper we solve the problem in the case in which all given polynomials (3) have prime factors of degree 1 or 2; in particular, we solve the problem in the real field case, and we retrieve the result of [5] (when all prime factors are linear).

The methods in the present paper are quite different from those used in [5]. We now use a localization technique which is self-contained and has the advantage of revealing the nature of the difficulty of this kind of problem. In particular, it shows how hopeless it is to find a general solution to our problem in the form of a system of inequalities on the degrees of the prime factorizations of the given polynomials (3). The sources of trouble are the *degree conditions* expressed by (7). In section 3.1 we

reduce the problem to a very complex system of diophantine inequalities involving the degrees of the prime factors of the given polynomials (3) and the degrees of the elementary divisors of the matrices under consideration. Feasibility conditions for this system are then determined (in section 3.2) for the mentioned particular case of all prime factors with degree 1 or 2.

According to the *interlacing theorem for similarity invariant factors*, stated around (2), the problem amounts to finding conditions on the given entities (3) so that the system

$$\underline{\alpha}_i \mid \alpha_i \mid \overline{\alpha}_i\,, \tag{4}$$

$$\underline{\beta}_j \mid \beta_j \mid \overline{\beta}_j\,, \tag{5}$$

$$\beta_i \mid \alpha_i \mid \beta_{i+h}\,, \tag{6}$$

$$\deg(\alpha_1\alpha_2\cdots\alpha_n) = n\,, \qquad \deg(\beta_1\beta_2\cdots\beta_m) = m\,, \tag{7}$$

where h denotes $2m-2n$, is feasible with respect to the unknowns α_i, β_j. *From now on we shall assume that the subsystem* (4)–(6) *is satisfied* by at least one set of *nonzero* polynomials $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m$. Note that (4)–(6) is equivalent to

$$\alpha_i^\iota \mid \alpha_i \mid \alpha_i^\sigma\,,$$
$$\beta_j^\iota \mid \beta_j \mid \beta_j^\sigma\,,$$
$$\beta_i \mid \alpha_i \mid \beta_{i+h}\,,$$

where $\alpha_i^\iota := \operatorname{lcm}\{\underline{\alpha}_i, \underline{\beta}_i\}$, $\alpha_i^\sigma := \gcd\{\overline{\alpha}_i, \overline{\beta}_{i+h}\}$, $\beta_j^\iota := \operatorname{lcm}\{\underline{\beta}_j, \underline{\alpha}_{j-h}\}$, and $\beta_j^\sigma := \gcd\{\overline{\beta}_j, \overline{\alpha}_j\}$ (cf. [5]). This means that we may assume (as we shall) without loss of generality that the chains $\underline{\alpha}$, $\overline{\alpha}$, $\underline{\beta}$, $\overline{\beta}$ coincide with the chains α^ι, α^σ, β^ι, β^σ, respectively. This assumption, together with the assumed feasibility of (4)–(6), may be expressed as follows (cf. [5]):

$$\underline{\alpha}_n \underline{\beta}_m \neq 0\,, \qquad \underline{\beta}_i \mid \underline{\alpha}_i \mid \underline{\beta}_{i+h}\,, \qquad \underline{\alpha}_i \mid \overline{\alpha}_i \tag{8}$$

$$\overline{\beta}_i \mid \overline{\alpha}_i \mid \overline{\beta}_{i+h}\,, \qquad \underline{\beta}_j \mid \overline{\beta}_j\,. \tag{9}$$

Assuming (4)–(6) is feasible, we now go in search of a solution meeting with (7). This may be viewed as a sort of "packing problem" asking for an algorithmic solution. In section 4 we discuss this matter and give some sample corollaries to our main result.

2. Main result. The *roots* of an $\mathbb{F}$-polynomial are its roots over a fixed algebraic closure $\bar{\mathbb{F}}$ of $\mathbb{F}$. When counting roots, we count multiplicities, so the number of roots of a polynomial equals its degree. The roots that lie in $\mathbb{F}$ are called $\mathbb{F}$-*roots*. We adopt the convention that the number of roots ($\mathbb{F}$-roots) of the zero polynomial is $+\infty$. In the statement of our main theorem we use the following definitions:

a is the number of roots of $\underline{\alpha}_1\underline{\alpha}_2\cdots\underline{\alpha}_n$.

$a_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\underline{\alpha}_1\underline{\alpha}_2\cdots\underline{\alpha}_n$.

a' is the number of roots of $\prod_{i=1}^n \gcd\{\overline{\alpha}_i, \underline{\beta}_{i+h}\}$.

$a'_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\prod_{i=1}^n \gcd\{\overline{\alpha}_i, \underline{\beta}_{i+h}\}$.

A is the number of roots of $\overline{\alpha}_1\overline{\alpha}_2\cdots\overline{\alpha}_n$.

$A_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\overline{\alpha}_1\overline{\alpha}_2\cdots\overline{\alpha}_n$.

b is the number of roots of $\underline{\beta}_1\underline{\beta}_2\cdots\underline{\beta}_m$.
$b_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\underline{\beta}_1\underline{\beta}_2\cdots\underline{\beta}_m$.
b' is the number of roots of $\prod_{j=1}^{m}\gcd\{\overline{\beta}_j,\underline{\alpha}_j\}$.
$b'_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\prod_{j=1}^{m}\gcd\{\overline{\beta}_j,\underline{\alpha}_j\}$.
B is the number of roots of $\overline{\beta}_1\overline{\beta}_2\cdots\overline{\beta}_m$.
$B_{\mathbb{F}}$ is the number of $\mathbb{F}$-roots of $\overline{\beta}_1\overline{\beta}_2\cdots\overline{\beta}_m$.

As we are assuming that $\underline{\alpha}_n$ and $\underline{\beta}_m$ are nonzero, then $a, a_{\mathbb{F}}, b, b_{\mathbb{F}}$ are nonnegative integers. However, some (and eventually all) of the other entities may be $+\infty$. In what follows "$\equiv$" denotes *congruence modulo* 2; this symbol goes with the convention $x \equiv +\infty$ if and only if $x = +\infty$.

THEOREM 2.1. *We are given polynomial chains, $\underline{\alpha}$, $\overline{\alpha}$, $\underline{\beta}$, and $\overline{\beta}$ as in* (3), *satisfying* (8)–(9). *We further assume that the given polynomials contain only prime factors of degree* 1 *or* 2. *There exists an $m \times m$ matrix $\mathbb{B}$ over $\mathbb{F}$, with similarity invariant factors $\beta_1, \ldots, \beta_m$, having a principal $n \times n$ submatrix $\mathbb{A}$, with similarity invariant factors $\alpha_1, \ldots, \alpha_n$, such that $\underline{\alpha}_i \,|\, \alpha_i \,|\, \overline{\alpha}_i$ and $\underline{\beta}_j \,|\, \beta_j \,|\, \overline{\beta}_j$ (for all i and j) if and only if the entities $a, a_{\mathbb{F}}, \ldots, B, B_{\mathbb{F}}$ defined above satisfy the following conditions:*

$$a \leqslant n \leqslant A\,, \tag{10}$$

$$b \leqslant m \leqslant B\,, \tag{11}$$

$$n + b \leqslant m + a'\,, \tag{12}$$

$$m + a \leqslant n + b'\,, \tag{13}$$

$$[n = a \ \wedge\ b'_{\mathbb{F}} = b_{\mathbb{F}}] \ \Rightarrow\ m \equiv b\,, \tag{14}$$

$$[m = b \ \wedge\ a'_{\mathbb{F}} = a_{\mathbb{F}}] \ \Rightarrow\ n \equiv a\,, \tag{15}$$

$$[n = A \ \wedge\ A_{\mathbb{F}} + b_{\mathbb{F}} = B_{\mathbb{F}} + a'_{\mathbb{F}}] \ \Rightarrow\ m \equiv B_{\mathbb{F}}\,, \tag{16}$$

$$[m = B \ \wedge\ B_{\mathbb{F}} + a_{\mathbb{F}} = A_{\mathbb{F}} + b'_{\mathbb{F}}] \ \Rightarrow\ n \equiv A_{\mathbb{F}}\,, \tag{17}$$

$$[n + b = m + a' \ \wedge\ a'_{\mathbb{F}} = A_{\mathbb{F}}] \ \Rightarrow\ n \equiv A_{\mathbb{F}}\,, \tag{18}$$

$$[m + a = n + b' \ \wedge\ b'_{\mathbb{F}} = B_{\mathbb{F}}] \ \Rightarrow\ m \equiv B_{\mathbb{F}}\,, \tag{19}$$

$$a_{\mathbb{F}} = A_{\mathbb{F}} \ \Rightarrow\ n \equiv a\,, \tag{20}$$

$$b_{\mathbb{F}} = B_{\mathbb{F}} \ \Rightarrow\ m \equiv b\,, \tag{21}$$

$$[a'_{\mathbb{F}} = a_{\mathbb{F}} \ \wedge\ b'_{\mathbb{F}} = b_{\mathbb{F}}] \ \Rightarrow\ n - a \equiv m - b\,, \tag{22}$$

$$[A_{\mathbb{F}} = a'_{\mathbb{F}} = a_{\mathbb{F}} + 1 \ \wedge\ b_{\mathbb{F}} = b'_{\mathbb{F}} = B_{\mathbb{F}} - 1] \ \Rightarrow\ [\,n \equiv a \ \Rightarrow\ m \equiv b]\,, \tag{23}$$

$$[B_{\mathbb{F}} = b'_{\mathbb{F}} = b_{\mathbb{F}} + 1 \ \wedge\ a_{\mathbb{F}} = a'_{\mathbb{F}} = A_{\mathbb{F}} - 1] \ \Rightarrow\ [m \equiv b \ \Rightarrow\ n \equiv a]\,. \tag{24}$$

REMARK 2.2. *Note that if one of the parameters occurring in one of the conditions* (14)–(24) *equals $+\infty$, then that condition is satisfied because its antecedent is trivially false.* For example, if $A_{\mathbb{F}} = +\infty$, then the bracketed clause in (17) is false because $B_{\mathbb{F}} + a_{\mathbb{F}} = A_{\mathbb{F}} + b'_{\mathbb{F}}$ implies $B_{\mathbb{F}}$ and (a fortiori) B are infinite; on the other hand, $m = B$ implies B is finite.

If all polynomials in (3) *split into linear factors over* $\mathbb{F}$*, then the above result is a consequence of Theorem* 5.3 *of* [5]. To see this, note that the splitting condition implies $a = a_{\mathbb{F}}, a' = a'_{\mathbb{F}}, \dots, B = B_{\mathbb{F}}$. Therefore (14)–(24) are consequences of (10)–(13). For example, if the bracketed clause in (16) holds, then $n+b = B+a'$; together with (10)–(13), this implies $m = B$, and therefore (16) holds.

According to the last remark, we may view (10)–(13) as the *basic*, or *fundamental*, conditions to our problem: in fact they constitute the complete solution if the field is algebraically closed, while the remaining (14)–(24) refer to *exceptional situations*, which are relevant only when $\mathbb{F}$ is not algebraically closed.

3. Further results and proofs. In this section m, n, h denote any nonnegative integers such that $n \leqslant m \leqslant n+h$. As before, we are given four polynomial chains (3) satisfying the assumptions (8)–(9). These assumptions are equivalent to the existence of nonzero monic polynomials, $\alpha_1 \mid \dots \mid \alpha_n$, $\beta_1 \mid \dots \mid \beta_m$, satisfying (4)–(6).

3.1. Localization. Denote by $\mathcal{H}$ the set of pairs of integers $(\hat{n}, \hat{m})$ for which polynomial chains $(\alpha_1, \dots, \alpha_n)$ and $(\beta_1, \dots, \beta_m)$ exist, satisfying, for all i and j,

$$\underline{\alpha}_i \mid \alpha_i \mid \overline{\alpha}_i\,, \tag{25}$$

$$\underline{\beta}_j \mid \beta_j \mid \overline{\beta}_j\,, \tag{26}$$

$$\beta_i \mid \alpha_i \mid \beta_{i+h}\,, \tag{27}$$

$$\deg(\alpha_1\alpha_2\cdots\alpha_n) = \hat{n}\,, \qquad \deg(\beta_1\beta_2\cdots\beta_m) = \hat{m}\,. \tag{28}$$

We wish to characterize $\mathcal{H}$ by a set of integral inequalities. To do so, we first localize the problem. Let $\mathbb{P}$ $(\mathbb{P}_k)$ be the set of all monic prime polynomials (of degree k). Let f be a monic polynomial (possibly zero), and let $\pi \in \mathbb{P}$. The *exponent of* π *in* f, denoted by $\exp_\pi(f)$, is the least upper bound of all nonnegative k such that $\pi^k \mid f$; we let $\exp_\pi(0) := +\infty$.

It is clear that for any polynomials f and g, $f \mid g$ if and only if $\exp_\pi(f) \leqslant \exp_\pi(g)$ for every prime π. This we call *localization technique* to transform a polynomial system like (25)–(28) into a system of integral inequalities. To do so, we simplify our notation a bit by introducing

$$a_{\pi i} := \exp_\pi(\underline{\alpha}_i)\,, \qquad b_{\pi j} := \exp_\pi(\underline{\beta}_j)\,, \qquad x_{\pi i} := \exp_\pi(\alpha_i)\,, \tag{29}$$

$$A_{\pi i} := \exp_\pi(\overline{\alpha}_i)\,, \qquad B_{\pi j} := \exp_\pi(\overline{\beta}_j)\,, \qquad y_{\pi j} := \exp_\pi(\beta_j)\,. \tag{30}$$

Moreover, we let

$$a_\pi := \textstyle\sum_{i=1}^{n} a_{\pi i}\,, \qquad b_\pi := \sum_{j=1}^{m} b_{\pi j}\,,$$

$$A_\pi := \textstyle\sum_{i=1}^{n} A_{\pi i}\,, \qquad B_\pi := \sum_{j=1}^{m} B_{\pi j}\,,$$

$$a'_\pi := \textstyle\sum_{i=1}^{n} \min\{A_{\pi i}, b_{\pi,i+h}\}\,, \qquad b'_\pi := \sum_{j=1}^{m} \min\{B_{\pi j}, a_{\pi j}\}\,.$$

Note that $(a_{\pi 1}, \dots, a_{\pi n})$ and $(b_{\pi 1}, \dots, b_{\pi m})$ are *chains of nonnegative integers*, i.e., $0 \leqslant a_{\pi 1} \leqslant \cdots \leqslant a_{\pi n} < +\infty$ and $0 \leqslant b_{\pi 1} \leqslant \cdots \leqslant b_{\pi m} < +\infty$. Thus a_π and b_π are finite and nonnegative. Obviously $(A_{\pi 1}, \dots, A_{\pi n})$ (resp., $(B_{\pi 1}, \dots, B_{\pi m})$) is also a

chain, but some of its terms may be $+\infty$. So A_π, B_π, a'_π, or b'_π may be $+\infty$. According to (1), we have $a_{\pi i} = A_{\pi i} = x_{\pi i} = +\infty$ for $i > n$ and $b_{\pi j} = B_{\pi j} = y_{\pi j} = +\infty$ for $j > m$.

Our assumptions (8)–(9) give rise to the integral inequalities

$$a_{\pi n}\,,\ b_{\pi m} \neq +\infty\,, \qquad b_{\pi i} \leqslant a_{\pi i} \leqslant b_{\pi,i+h}\,, \qquad a_{\pi i} \leqslant A_{\pi i}\,,$$
$$B_{\pi i} \leqslant A_{\pi i} \leqslant B_{\pi,i+h}\,, \qquad b_{\pi j} \leqslant B_{\pi j}\,.$$

Let $\mathcal{H}_\pi$ be the set of the pairs of integers (x_π, y_π) for which there exist chains of integers $x_{\pi 1} \leqslant \cdots \leqslant x_{\pi n}$ and $y_{\pi 1} \leqslant \cdots \leqslant y_{\pi m}$ satisfying

$$a_{\pi i} \leqslant x_{\pi i} \leqslant A_{\pi i}\,, \tag{31}$$

$$b_{\pi j} \leqslant y_{\pi j} \leqslant B_{\pi j}\,, \tag{32}$$

$$y_{\pi i} \leqslant x_{\pi i} \leqslant y_{\pi,i+h}\,, \tag{33}$$

$$\sum_{i=1}^{n} x_{\pi i} = x_\pi\,, \qquad \sum_{j=1}^{m} y_{\pi j} = y_\pi\,. \tag{34}$$

Our assumptions (8)–(9) imply that all sets $\mathcal{H}$ and $\mathcal{H}_\pi$ are nonempty. All $\mathcal{H}_\pi$, except a finite number, contain the point $(0,0)$ because a_π (resp., b_π) is a positive integer if and only if π is a factor of $\underline{\alpha}_1\underline{\alpha}_2\cdots\underline{\alpha}_n$, (resp., $\underline{\beta}_1\underline{\beta}_2\cdots\underline{\beta}_m$), and there exist only a finite number of such prime factors. Expressions like $\sum_{\pi\in\mathbb{P}}\mathcal{H}_\pi$ (others of this kind will occur) denote, as is usual in algebra, the set of all sums $\sum_{\pi\in\mathbb{P}} h_\pi$, with all $h_\pi \in \mathcal{H}_\pi$, and $h_\pi = (0,0)$ except for a finite number of π's.

THEOREM 3.1. *We have* $\mathcal{H} = \sum_{\pi\in\mathbb{P}} \deg(\pi)\,\mathcal{H}_\pi$.

Proof. Let $(\hat{n}, \hat{m}) \in \mathcal{H}$. There exist chains α and β satisfying (25)–(28). For each $\pi \in \mathbb{P}$, define $x_{\pi i}, y_{\pi j}, x_\pi, y_\pi$ as in (29)–(30) and (34). Then clearly $(x_\pi, y_\pi) \in \mathcal{H}_\pi$. On the other hand, the degree of $\alpha_1\cdots\alpha_n$ (resp., $\beta_1\cdots\beta_m$) is obviously the sum of all integers $\deg(\pi)\,x_\pi$ (resp., $\deg(\pi)\,y_\pi$) for $\pi \in \mathbb{P}$. So $(\hat{n}, \hat{m})$ lies in the right-hand side set of the identity under consideration.

Conversely, if $(\hat{n}, \hat{m})$ is a sum of terms $\deg(\pi)(x_\pi, y_\pi)$, where $(x_\pi, y_\pi) \in \mathcal{H}_\pi$, there exist for each $\pi \in \mathbb{P}$ chains of integers $x_{\pi 1} \leqslant \cdots \leqslant x_{\pi n}$, $y_{\pi 1} \leqslant \cdots \leqslant y_{\pi m}$ satisfying (31)–(34). Now define α_i (resp., β_j) as the product of all powers $\pi^{x_{\pi i}}$ (resp., $\pi^{y_{\pi j}}$) for $\pi \in \mathbb{P}$. It is easily seen that (α_i) and (β_j) are chains satisfying (25)–(28). So $(\hat{n}, \hat{m}) \in \mathcal{H}$. □

THEOREM 3.2. *The pair of integers (x_π, y_π) belongs to $\mathcal{H}_\pi$ if and only if the following inequalities hold:*

$$a_\pi \leqslant x_\pi \leqslant A_\pi\,, \tag{35}$$

$$b_\pi \leqslant y_\pi \leqslant B_\pi\,, \tag{36}$$

$$x_\pi + b_\pi \leqslant y_\pi + a'_\pi\,, \tag{37}$$

$$y_\pi + a_\pi \leqslant x_\pi + b'_\pi\,. \tag{38}$$

Proof. Let $\mathcal{H}'_\pi$ be the set of pairs of integers (x_π, y_π) that satisfy (35)–(38). We must prove $\mathcal{H}_\pi = \mathcal{H}'_\pi$. To do so, let $(x_\pi, y_\pi) \in \mathcal{H}_\pi$. Equations (35)–(36) are obvious; the proof of (37)–(38) is not difficult, and we give only a sketchy proof of (37), where

we use the identity $z + w = \min\{z, w\} + \max\{z, w\}$, the convention $b_{\pi j} = +\infty$ for $j > m$, and the inequalities (31)–(33). Here is a proof of (37):

$$\begin{aligned} x_\pi + b_\pi &= \sum_{i=1}^{n} x_{\pi i} + \sum_{j=1}^{m} b_{\pi j} \;=\; \sum_{i=m-h+1}^{n} x_{\pi i} + \sum_{i\leqslant m-h} \bigl[x_{\pi i} + b_{\pi,i+h}\bigr] \\ &= \sum_{i=m-h+1}^{n} x_{\pi i} \;+ \sum_{i\leqslant m-h} \bigl[\min\{x_{\pi i}, b_{\pi,i+h}\} + \max\{x_{\pi i}, b_{\pi,i+h}\}\bigr] \\ &= \sum_{i=1}^{n} \min\{x_{\pi i}, b_{\pi,i+h}\} + \sum_{j=1}^{m} \max\{x_{\pi,j-h}, b_{\pi j}\} \\ &\leqslant \sum_{i=1}^{n} \min\{A_{\pi i}, b_{\pi,i+h}\} + \sum_{j=1}^{m} y_{\pi j} = a'_\pi + y_\pi \,. \end{aligned}$$

So we have (35)–(38) and, therefore, $\mathcal{H}_\pi \subseteq \mathcal{H}'_\pi$ holds. The dual inclusion is a bit harder to get.

Step 1, *in which we prove that* $b_\pi \leqslant y_\pi \leqslant b'_\pi$ *implies* $(a_\pi, y_\pi) \in \mathcal{H}_\pi$. Define a chain of integers $x_{\pi i} := a_{\pi i}$ for $i \in \{1, \dots, n\}$. To show the existence of integers $y_{\pi 1}, \dots, y_{\pi m}$ satisfying (31)–(34), let $\underline{y}_{\pi j} := b_{\pi j}$ and $\overline{y}_{\pi j} := \min\{B_{\pi j}, a_{\pi j}\}$. It is easy to check that $(\underline{y}_{\pi 1}, \dots, \underline{y}_{\pi m})$ $[(\overline{y}_{\pi 1}, \dots, \overline{y}_{\pi m})]$ is a chain satisfying (31)–(33) (for $x_{\pi i} := a_{\pi i}$); moreover, $y_{\pi 1}, \dots, y_{\pi m}$ satisfy (31)–(33) if and only if $\underline{y}_{\pi j} \leqslant y_{\pi j} \leqslant \overline{y}_{\pi j}$ for all j. On the other hand,

$$\sum_{j=1}^{m} \underline{y}_{\pi j} = b_\pi \quad \text{and} \quad \sum_{j=1}^{m} \overline{y}_{\pi j} = b'_\pi \,.$$

Therefore, if $b_\pi \leqslant y_\pi \leqslant b'_\pi$, there exists a chain of integers $Y_{\pi 1}, \dots, Y_{\pi m}$ such that $\underline{y}_{\pi j} \leqslant Y_{\pi j} \leqslant \overline{y}_{\pi j}$ and $\sum_{j=1}^{m} Y_j = y_\pi$. This completes Step 1.

Step 2, *in which we prove that* $a_\pi \leqslant x_\pi \leqslant a'_\pi$ *implies* $(x_\pi, b_\pi) \in \mathcal{H}_\pi$. This is the "dual" to Step 1, and the proof may follow the same strategy with obvious changes.

Step 3, *in which we prove that if* $(x_\pi - 1, y_\pi - 1) \in \mathcal{H}_\pi$ *and* $(x_\pi, y_\pi) \in \mathcal{H}'_\pi$, *then* $(x_\pi, y_\pi) \in \mathcal{H}_\pi$. Let $(\check{x}_{\pi i})$ and $(\check{y}_{\pi j})$ be chains of integers satisfying (31)–(33), $\sum_{i=1}^{n} \check{x}_{\pi i} = x_\pi - 1$, and $\sum_{j=1}^{m} \check{y}_{\pi j} = y_\pi - 1$. We consider the following two cases.

Case 1: *for all* j, $\check{y}_{\pi j} \geqslant A_{\pi,j-h}$. As $\sum_{j=1}^{m} \check{y}_{\pi j} < B_\pi$, there exists r such that $\check{y}_{\pi r} < B_{\pi r}$. Define $y_{\pi j} := \check{y}_{\pi j}$ for $j \neq r$, and $y_{\pi r} := \check{y}_{\pi r} + 1$. We obviously have (32). For the current $y_{\pi j}$'s, integers $x_{\pi 1}, \dots, x_{\pi n}$ satisfy (31) and (33) if and only if $\max\{a_{\pi i}, y_{\pi i}\} \leqslant x_{\pi i} \leqslant \min\{A_{\pi i}, y_{\pi,i+h}\}$ for all i. Therefore, integers $x_{\pi 1}, \dots, x_{\pi n}$ exist satisfying (31), (33), *and* (34) if and only if we have

$$\sum_{i=1}^{n} \max\{a_{\pi i}, y_{\pi i}\} \leqslant x_\pi \leqslant \sum_{i=1}^{n} \min\{A_{\pi i}, y_{\pi,i+h}\} \,. \tag{39}$$

These inequalities are easy to verify, so we are done with this case.

Case 2: *there exists* r *such that* $\check{y}_{\pi r} < A_{\pi,r-h}$. Define $y_{\pi j}$ exactly as in Case 1. The existence of integers $x_{\pi 1}, \dots, x_{\pi n}$ satisfying (31), (33), and (34) is proved as above; only the details to verify (39) are slightly different.

To finish the proof, note that the set of pairs of *real* numbers (x_π, y_π) satisfying (35)–(38) is a convex "hexagon" like the gray one in Figure 1, and $\mathcal{H}'_\pi$ is the set of its

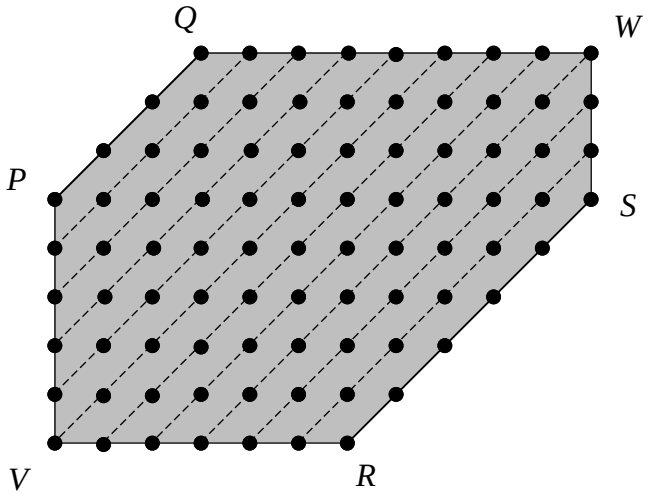


FIG. 1. *A "hexagon" and its integral points.*

integral points (the bold dots in Figure 1). In the figure, $V = (a_\pi, b_\pi)$, $W = (A_\pi, B_\pi)$, $R = (a'_\pi, b_\pi)$, $S = (A_\pi, b_\pi + A_\pi - a'_\pi)$, etc. It may not be a hexagon at all; e.g., if A_π (or B_π) is not finite, the vertex W does not occur and the set is unbounded; if $a'_\pi = a_\pi$, then $V = R$, etc. So a "hexagon" may eventually be a pentagon, a triangle, a line segment, a half-ray, a singleton, etc.; it can be as "degenerate," as we show in the five examples of Figure 2.

The sides $\overline{PQ}$ and $\overline{RS}$, when they exist, are slanted straight line segments, or half-rays, of slope 1. By definition, the *NE-successor* of an integral point (p, q) is the integral point $(p+1, q+1)$. In Step 3 we showed that if a given integral point lies in $\mathcal{H}_\pi$, then its NE-successor lies in $\mathcal{H}_\pi$ as well, if it lies inside the "hexagon" at all. In Step 1 (resp., Step 2) we proved that all integral points of $\overline{VP}$ (resp., $\overline{VR}$) lie in $\mathcal{H}_\pi$. By induction, all integral points of the "hexagon" lying on any fixed straight line of slope 1 (the dashed lines in Figure 1) belong to $\mathcal{H}_\pi$. So $\mathcal{H}'_\pi \subseteq \mathcal{H}_\pi$, and the theorem is proved. □

3.2. "Hexagons." In our problem, hexagon-like configurations as the one in Figure 1 occur in a decisive manner, and we need some results on them. For each $t \in \{1, 2\}$ we are given 6 parameters, $a_t, a'_t, A_t, b_t, b'_t, B_t$ (in this order), where a_t, b_t are real numbers and $a'_t, A_t, b'_t, B_t \in \mathbb{R} \cup \{+\infty\}$, satisfying

$$a_t \leqslant a'_t \leqslant A_t\,, \tag{40}$$

$$b_t \leqslant b'_t \leqslant B_t\,, \tag{41}$$

$$A_t + b_t \leqslant B_t + a'_t\,, \tag{42}$$

$$B_t + a_t \leqslant A_t + b'_t\,. \tag{43}$$

For each $t = 1, 2$, let H_t be the set of all $(x_t, y_t) \in \mathbb{R}^2$ such that

$$a_t \leqslant x_t \leqslant A_t\,, \tag{44}$$

$$b_t \leqslant y_t \leqslant B_t\,, \tag{45}$$

$$x_t + b_t \leqslant y_t + a'_t\,, \tag{46}$$

$$y_t + a_t \leqslant x_t + b'_t\,. \tag{47}$$

Such a closed convex set is called a "*hexagon*" (quotation marks included, by reasons pointed out at the end of the last proof). The relevance of the inequalities (40)–(43) is that they, together with the set H_t, uniquely determine the 6 parameters $a_t, \dots, B_t$ (in the order given). More precisely, $H_1 = H_2$ if and only if $a_1 = a_2, \dots, B_1 = B_2$. Accordingly, $a_t, \dots, B_t$ are called *the parameters of* H_t. If H_t is bounded (i.e., A_t and B_t are finite), its vertices (that is to say, its *extreme points* [4]) are

$$(a_t, b_t),\ (a'_t, b_t),\ (a_t, b'_t),\ (A_t, B_t),\ (A_t, b_t + A_t - a'_t),\ (a_t + B_t - b'_t, B_t)\,. \tag{48}$$

These six points are not distinct in general, and some of these pairs are not in $\mathbb{R}^2$ if H_t is unbounded. However, the "generic" case is neat and easy to prove: *H_t is bounded and has precisely* 6 *vertices if and only if all* (40)–(43) *are strict inequalities*; in this case we say H_t is a 6-*vertex-hexagon.* Here is an interesting fact.

LEMMA 3.3. *Any linear combination $\lambda_1 H_1 + \lambda_2 H_2$, where the λ_t are real nonnegative, is again a "hexagon" whose parameters are the corresponding linear combinations $\lambda_1 a_1 + \lambda_2 a_2, \dots, \lambda_1 B_1 + \lambda_2 B_2$.*

Proof. The proof trivially reduces to the case $\lambda_1 = \lambda_2 = 1$. Accordingly, let Σ be the "hexagon" with parameters $a_1 + a_2, \dots, B_1 + B_2$. The only difficulty of the proof is to show that $\Sigma \subseteq H_1 + H_2$. The compact case (all finite parameters) is very easy, because Σ is the convex hull of the set of its vertices, and each vertex of Σ is the sum of a vertex of H_1 and a vertex of H_2, as you may see by checking the list (48). So $\Sigma = H_1 + H_2$ in the compact case.

In the noncompact case a similar argument is available, using [4, Theorem 18.5]; first, however, we need to determine the relation between the extreme directions of Σ and those of the H_t's; this is a bit messy, because a "hexagon" has one of seven possible recession cones (check [4, p. 60] for the definition). A more conceptual argument goes like this. First we express each H_t as a union, $H_t = \bigcup_{k=1}^{\infty} H_t^k$, where $H_t^1 \subseteq H_t^2 \subseteq H_t^3 \subseteq \cdots$ is a nested sequence of compact "hexagons" such that the *finite* parameters of H_t coincide with the corresponding parameters of each H_t^k, and all other parameters of H_t^k increase to $+\infty$. So, for each t, the parameters of H_t^k converge to the corresponding parameters of H_t. By the previous argument, $H_1^k + H_2^k$ is a "hexagon" with known finite parameters; and these parameters converge to the parameters of Σ. So the proof ends up because Σ is the union of all $H_1^k + H_2^k$, and addition commutes with nested unions. ∎

Integral points. We now consider the integral points of "hexagons" whose vertices are integral points. For any subset S of $(\mathbb{R} \cup \{+\infty\})^k$, $S^{\mathbb{Z}}$ denotes the set of integral points of S. Theorem 3.1 shows that we have in between hands the following difficult diophantine problem.

PROBLEM. *Assume we are given "hexagons" $H_1, \dots, H_w$, with integral parameters, and positive integers, $d_1, \dots, d_w$. Describe the set $d_1 H_1^{\mathbb{Z}} + \cdots + d_w H_w^{\mathbb{Z}}$ in terms of the integral points of the given "hexagons" $H_1, \dots, H_w$.*

In this paper, we solve this problem in the case in which each d_s equals 1 or 2 and leave all other cases open. To begin with, we go back to the case of just two "hexagons," H_1, H_2, whose parameters $a_t, \dots, B_t$ now lie in $\mathbb{Z} \cup \{+\infty\}$ and satisfy (40)–(43). The parameters of the "hexagon" $H := H_1 + 2H_2$ are the following:

$$\left.\begin{array}{lll} a := a_1 + 2a_2\,, & a' := a'_1 + 2a'_2\,, & A := A_1 + 2A_2\,, \\ b := b_1 + 2b_2\,, & b' := b'_1 + 2b'_2\,, & B := B_1 + 2B_2\,, \end{array}\right\} \tag{49}$$

and H is the set of all $(X,Y)\in\mathbb{R}^2$ such that

$$a\leqslant X\leqslant A\,, \tag{50}$$

$$b\leqslant Y\leqslant B\,, \tag{51}$$

$$X+b\leqslant Y+a'\,, \tag{52}$$

$$Y+a\leqslant X+b'\,. \tag{53}$$

We now give the relationship between the set $H^{\mathbb{Z}}$, i.e., $(H_1+2H_2)^{\mathbb{Z}}$, and its subset $H_1^{\mathbb{Z}}+2H_2^{\mathbb{Z}}$.

THEOREM 3.4. *Assume (X,Y) is an element of $H^{\mathbb{Z}}$. Then (X,Y) belongs to $H_1^{\mathbb{Z}}+2H_2^{\mathbb{Z}}$ if and only if the following conditions hold, where $\equiv$ denotes congruence modulo 2:*

$$[X=a\ \wedge\ b_1'=b_1]\ \Rightarrow\ Y\equiv b\,, \tag{54}$$

$$[Y=b\ \wedge\ a_1'=a_1]\ \Rightarrow\ X\equiv a\,, \tag{55}$$

$$[X=A\ \wedge\ A_1+b_1=B_1+a_1']\ \Rightarrow\ Y\equiv B_1\,, \tag{56}$$

$$[Y=B\ \wedge\ B_1+a_1=A_1+b_1']\ \Rightarrow\ X\equiv A_1\,, \tag{57}$$

$$[X+b=Y+a'\ \wedge\ a_1'=A_1]\ \Rightarrow\ X\equiv A_1\,, \tag{58}$$

$$[Y+a=X+b'\ \wedge\ b_1'=B_1]\ \Rightarrow\ Y\equiv B_1\,, \tag{59}$$

$$a_1=A_1\ \Rightarrow\ X\equiv a\,, \tag{60}$$

$$b_1=B_1\ \Rightarrow\ Y\equiv b\,, \tag{61}$$

$$[a_1'=a_1\ \wedge\ b_1'=b_1]\ \Rightarrow\ X-a\equiv Y-b\,, \tag{62}$$

$$[A_1=a_1'=a_1+1\ \wedge\ b_1=b_1'=B_1-1]\ \Rightarrow\ [X\equiv a\ \Rightarrow\ Y\equiv b]\,, \tag{63}$$

$$[B_1=b_1'=b_1+1\ \wedge\ a_1=a_1'=A_1-1]\ \Rightarrow\ [Y\equiv b\ \Rightarrow\ X\equiv a]\,. \tag{64}$$

Note that as in Remark 2.2, if one of the parameters occurring in one of the conditions (54)–(64) equals $+\infty$, then that condition is satisfied, because its antecedent is trivially false.

In the proof of Theorem 3.4, the following simple observation will be used very often. Given closed intervals $I_t=[\alpha_t,\beta_t]\subset\mathbb{R}\cup\{+\infty\}$ $(t=1,2)$, where $\alpha_t\in\mathbb{Z}$ and $\beta_t\in\mathbb{Z}\cup\{+\infty\}$, the description of the set $I_1^{\mathbb{Z}}+2I_2^{\mathbb{Z}}$ splits into two cases.

Case I. If I_1 has more than one element, then

$$I_1^{\mathbb{Z}}+2I_2^{\mathbb{Z}}=(I_1+2I_2)^{\mathbb{Z}}\,. \tag{65}$$

Case II. If I_1 has just one element, $I_1=\{\alpha_1\}$, then

$$I_1^{\mathbb{Z}}+2I_2^{\mathbb{Z}}=\{p\in[I_1+2I_2]^{\mathbb{Z}}:\,p\equiv\alpha_1\}\,. \tag{66}$$

Thus in Case II our set is not, in general, an interval of integers.

DEFINITION 3.5. We say H_1 *is degenerate* if it satisfies one of the following conditions:

(a) *There exists at most one (integral) point in H_1 with abscissa a_1+1;*

(b) *There exists at most one (integral) point in H_1 with ordinate b_1+1.*

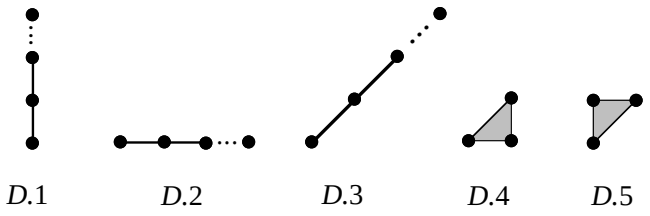


FIG. 2. *The five cases of degeneracy.*

The parenthetic "*integral*" means that if we replace, in either (a) or (b), "*point*" by "*integral point,*" an equivalent statement is obtained, which is easily proven.

LEMMA 3.6. *H_1 is degenerate if and only if one of the following conditions holds:*

(D.1) $a_1 = A_1$;
(D.2) $b_1 = B_1$;
(D.3) $a'_1 = a_1 \ \wedge \ b'_1 = b_1$;
(D.4) $A_1 = a'_1 = a_1 + 1 \ \wedge \ b_1 = b'_1 = B_1 - 1$;
(D.5) $B_1 = b'_1 = b_1 + 1 \ \wedge \ a_1 = a'_1 = A_1 - 1$.

Proof. We only sketch the argument. By the definition (44)–(47) of H_1, there is no point in H_1 of abscissa $a_1 + 1$ if and only if condition (D.1) holds. If $a_1 < A_1$, the set of all y_1 such that $(a_1 + 1, y_1)$ lies in H_1 is the closed interval with lower and upper bounds $\max\{b_1, b_1 + a_1 + 1 - a'_1\}$ and $\min\{B_1, 1 + b'_1\}$; the identity of these two elements is equivalent to $(D.2) \vee (D.3) \vee (D.5)$. So, item (a) of Definition 3.5 is equivalent to $(D.1) \vee (D.2) \vee (D.3) \vee (D.5)$. □

Note that the ($D.k$) are the antecedent conditions of (60)–(64). In Figure 2 we sketch the five cases of degeneracy. In the first 3 cases, H_1 may happen to be a singleton, or unbounded; in the last two cases, H_1 has exactly 3 integral points.

LEMMA 3.7. *Theorem* 3.4 *holds if H_1 is degenerate. More specifically, consider the following five conditions for a given $(X, Y) \in H^{\mathbb{Z}}$:* ($C$.1) $X \equiv a$; (C.2) $Y \equiv b$; (C.3) $X - a \equiv Y - b$; (C.4) $X \equiv a \Rightarrow Y \equiv b$; ($C$.5) $Y \equiv b \Rightarrow X \equiv a$.

(I) *If H_1 is a singleton, then (X, Y) belongs to $H_1^{\mathbb{Z}} + 2\,H_2^{\mathbb{Z}}$ if and only if the condition $(C.1) \wedge (C.2)$ holds.*

(II) *Assume H_1 is not a singleton and satisfies the condition ($D.k$) for a fixed $k \in \{1, 2, 3, 4, 5\}$. Then (X, Y) belongs to $H_1^{\mathbb{Z}} + 2\,H_2^{\mathbb{Z}}$ if and only if the condition ($C.k$) holds.*

(III) *If, for a fixed k, $(D.k) \wedge (C.k)$ holds, then* (54)–(64) *all hold.*

Proof. Clearly, Theorem 3.4 follows, in the degenerate case, from (I)–(III). The singleton case (I) is very easy to check. The proofs of (II) and (III) are also easy (intuitively, we have to scale-by-two a dotted "hexagon" like the one in Figure 1 and then add it up, dot by dot, with each one of the degenerate "hexagons" of Figure 2). The details are boring, so we only give them for the case $k = 4$.

(II) In case (D.4) $H_1^{\mathbb{Z}}$ reduces to the three points

$$\alpha = (a_1, b_1), \quad \beta = (a_1 + 1, b_1), \quad \text{and} \quad \gamma = (a_1 + 1, b_1 + 1)\,. \tag{67}$$

Assume that $(X, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$. Then this point is of one of three types: $(a_1 + 2x_2, b_1 + 2y_2)$, $(a_1 + 1 + 2x_2, b_1 + 2y_2)$, or $(a_1 + 1 + 2x_2, b_1 + 1 + 2y_2)$. If $X \equiv a$, then (X, Y) is of the first type; therefore $Y \equiv b$ and (C.4) holds.

Conversely, assume (C.4) and let $(X, Y) \in H^{\mathbb{Z}}$. It is easy to check that

if $X \not\equiv a_1$ and $Y \not\equiv b_1$, then $[(X, Y) - \gamma]/2 \in H_2$;

if $X \equiv a_1$ and $Y \equiv b_1$, then $[(X,Y) - \alpha]/2 \in H_2$;
if $X \not\equiv a_1$ and $Y \equiv b_1$, then $[(X,Y) - \beta]/2 \in H_2$.

So $(X,Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ in all cases compatible with $(C.4)$.

(III) Assume $(D.4) \wedge (C.4)$ holds. By (II), $(X,Y) = (x_1 + 2x_2, y_1 + 2y_2)$ for some $(x_1, y_1) \in H_1^{\mathbb{Z}}$ and $(x_2, y_2) \in H_2^{\mathbb{Z}}$.

Conditions (54) and (63) are clearly implied by $(C.4)$. By $(D.4)$, the antecedents of conditions (55)–(56), (59)–(62), and (64) are obviously false.

So we are left with (57) and (58). Assume the antecedent of (57) holds; then $Y = B_1 + 2B_2$, and so $Y \not\equiv b_1$; by (C.4) this implies $X \not\equiv a_1$; thus $X \equiv A_1$, and (57) holds. Assume the antecedent of (58) holds; then we have equality in (46) for both values of t; in particular, $x_1 = A_1 + y_1 - b_1$, and so $x_1 \geqslant A_1$; therefore $x_1 = A_1$, and we have $X \equiv A_1$; hence (58) holds. □

LEMMA 3.8. *Assume H_1 is nondegenerate. Any integral point in the interior of H belongs to $H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$.*

Proof. Claim 1: *an integral point lying in* $\operatorname{int} H$, *with abscissa* $a+1$ *(or with ordinate* $b+1$*) belongs to* $H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$. We shall consider only the integral points of abscissa $a+1$ (the case of ordinate $b+1$ is similar). It is easily seen that $(a+1, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ if and only if $Y \in J_1^{\mathbb{Z}} + 2[b_2, b_2']^{\mathbb{Z}}$, where

$$J_1 := \big[\max\{b_1, a_1 - a_1' + b_1 + 1\}, \min\{B_1, b_1' + 1\}\big] .$$

J_1 is the interval of all ordinates of points in H_1 with abscissa $a+1$ (with $+\infty$ included, in case $b_1' = +\infty$). As H_1 is nondegenerate, J_1 has at least two elements, and so $J_1^{\mathbb{Z}} + 2[b_2, b_2']^{\mathbb{Z}} = \big(J_1 + 2[b_2, b_2']\big)^{\mathbb{Z}}$. We clearly have $J_1 + 2[b_2, b_2'] \supseteq [b+1, b']$.

Now assume $(a+1, Y) \in \big(\operatorname{int} H\big)^{\mathbb{Z}}$. As the inequalities (50)–(53) are strict (with $X = a+1$), we have $Y \in [b+1, b']$. Therefore $(a+1, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$.

Claim 2: *in case* $(X,Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ *and* $(X+1, Y+1)$ *lies in* $\operatorname{int} H$, *we have* $(X+1, Y+1) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$.

Note that, by assumption, X and Y satisfy $X+1 < A$, $Y+1 < B$, $X + b < Y + a'$, and $Y + a < X + b'$, and there exist integers x_1, x_2, y_1, y_2 satisfying (44)–(47) such that $X = x_1 + 2x_2$ and $Y = y_1 + 2y_2$. Under these conditions we shall prove the existence of integers $\bar{x}_1, \bar{x}_2, \bar{y}_1, \bar{y}_2$ satisfying (44)–(47), and

$$X + 1 = \bar{x}_1 + 2\bar{x}_2 \quad \text{and} \quad Y + 1 = \bar{y}_1 + 2\bar{y}_2 . \tag{68}$$

Case 1: $x_1 < A_1$ *and* $y_1 < B_1$. Let $\bar{x}_1 := x_1 + 1$, $\bar{y}_1 := y_1 + 1$, $\bar{x}_2 := x_2$, and $\bar{y}_2 := y_2$. It is obvious that $\bar{x}_t, \bar{y}_t$ satisfy (44)–(47) and (68).

Case 2: $x_1 = A_1$ *or* $y_1 = B_1$. We examine only the assumption $x_1 = A_1$ ($y_1 = B_1$ is similarly treated). We must have $x_2 < A_2$, otherwise we get the contradiction $X + 1 > A$. We consider three subcases.

Subcase 2.1: $x_1 = A_1$ *and* $y_1 = b_1$. Clearly $a_1' = A_1$. Define $\bar{x}_1 := x_1 - 1$, $\bar{y}_1 := y_1 + 1$, $\bar{x}_2 := x_2 + 1$, and $\bar{y}_2 := y_2$. Then (68) and (44)–(47) hold, the inequalities $\bar{y}_1 + a_1 \leqslant \bar{x}_1 + b_1'$ and $\bar{x}_2 + b_2 \leqslant \bar{y}_2 + a_2'$ being the less obvious to check; for the current values of the variables, they are written as

$$b_1 + 1 + a_1 \leqslant A_1 - 1 + b_1' , \tag{69}$$

$$x_2 + 1 + b_2 \leqslant y_2 + a_2' . \tag{70}$$

If (69) is false, H_1 is degenerate. If (70) is false, we get $x_2 + b_2 = y_2 + a_2'$, and therefore $(X+1, Y+1) \notin \operatorname{int} H$. So (69) and (70) must hold.

Subcase 2.2: $x_1 = A_1$ *and* $y_2 = B_2$. Define the $\bar{x}_t, \bar{y}_t$ as in Subcase 2.1. As $Y + 1 < B$, we have $y_1 + 2 \leqslant B_1$ and therefore $\bar{y}_1 + a_1 \leqslant \bar{x}_1 + b'_1$; the inequality $\bar{x}_2 + b_2 \leqslant \bar{y}_2 + a'_2$ follows from $x_2 + 1 \leqslant A_2$ and (42). The rest is trivial.

Subcase 2.3: $x_1 = A_1$, $y_1 > b_1$, *and* $y_2 < B_2$. Define $\bar{x}_1 := x_1 - 1$, $\bar{y}_1 := y_1 - 1$, $\bar{x}_2 := x_2 + 1$, and $\bar{y}_2 := y_2 + 1$. These $\bar{x}_t, \bar{y}_t$ trivially satisfy (68) and (44)–(47).

This completes the proof of Claim 2. The lemma follows from Claims 1 and 2 by a straightforward induction using NE-successors as we did at the end of the proof of Theorem 3.2. □

LEMMA 3.9. *Let* (X, Y) *be an integral point of the boundary of* H *satisfying* (54)–(64). *Then* $(X, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$.

Proof. (X, Y) satisfies equality in one of the inequalities (50)–(53). So we have six cases: *Case* 1: $X = a$; *Case* 2: $Y = b$; *Case* 3: $X = A$; *Case* 4: $Y = B$; *Case* 5: $X + b = Y + a'$; *Case* 6: $Y + a = X + b'$. By symmetry reasons we only need to consider Cases 1, 3, and 5.

Case 1. The boundary points under consideration form the set

$$\mathcal{B}_a := \left\{ (a, y) : y \in \left([b_1, b'_1] + 2[b_2, b'_2] \right)^{\mathbb{Z}} \right\}.$$

So our assumption is $(X, Y) \in \mathcal{B}_a$. The points of $H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ with abscissa a form the set

$$\mathcal{B}_a^* := \left\{ (a, y) : y \in [b_1, b'_1]^{\mathbb{Z}} + 2[b_2, b'_2]^{\mathbb{Z}} \right\}.$$

According to (65)–(66), we have two subcases. (I) If $b_1 < b'_1$, then $\mathcal{B}_a = \mathcal{B}_a^*$, and so $(X, Y) \in \mathcal{B}_a^*$. (II) If $b_1 = b'_1$, (54) implies $Y \equiv b_1$; thus $(X, Y) \in \mathcal{B}_a^*$.

Case 3. Clearly A is finite. Our assumption is $(X, Y) \in \mathcal{B}_A$, where

$$\mathcal{B}_A := \left\{ (A, y) : y \in \left([b_1 + A_1 - a'_1, B_1] + 2[b_2 + A_2 - a'_2, B_2] \right)^{\mathbb{Z}} \right\}.$$

The points of $H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ with abscissa A form the set

$$\mathcal{B}_A^* := \left\{ (A, y) : y \in [b_1 + A_1 - a'_1, B_1]^{\mathbb{Z}} + 2[b_2 + A_2 - a'_2, B_2]^{\mathbb{Z}} \right\}.$$

If $b_1 + A_1 - a'_1 < B_1$, we have $\mathcal{B}_A = \mathcal{B}_A^*$, and so $(X, Y) \in \mathcal{B}_A^*$. If $b_1 + A_1 - a'_1 = B_1$, then (56) implies $Y \equiv B_1$; as $[b_1 + A_1 - a'_1, B_1] = \{B_1\}$, (66) implies $(X, Y) \in \mathcal{B}_A^*$.

Case 5. In this case a' is finite. (X, Y) belongs to the set

$$\begin{aligned} \Phi &:= \left\{ (x, x + b - a') : x \in \left([a'_1, A_1] + 2[a'_2, A_2] \right)^{\mathbb{Z}} \right\} \\ &= (0, b - a') + \left([a'_1, A_1] + 2[a'_2, A_2] \right)^{\mathbb{Z}} (1, 1). \end{aligned}$$

It is easily seen that

$$\begin{aligned} &\{(x, y) \in H : x + b = y + a'\} \\ &\quad = \{(x_1, y_1) \in H_1 : x_1 + b_1 = y_1 + a'_1\} + 2\{(x_2, y_2) \in H_2 : x_2 + b_2 = y_2 + a'_2\}. \end{aligned}$$

Therefore, the points of Φ that belong to $H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ form the set

$$\begin{aligned} \Phi^* &:= \left\{ (x_1, x_1 + b_1 - a'_1) : x_1 \in [a'_1, A_1]^{\mathbb{Z}} \right\} \\ &\quad + 2\left\{ (x_2, x_2 + b_2 - a'_2) : x_2 \in [a'_2, A_2]^{\mathbb{Z}} \right\} \\ &= (0, b - a') + \left([a'_1, A_1]^{\mathbb{Z}} + 2[a'_2, A_2]^{\mathbb{Z}} \right) (1, 1). \end{aligned}$$

According to (65)–(66), we consider two cases. If $a_1' < A_1$, we have $\Phi = \Phi^*$, and so $(X, Y) \in \Phi^*$. If $a_1' = A_1$, (58) implies $X \equiv A_1$; thus $(X, Y) \in \Phi^*$. □

Completion of the proof of Theorem 3.4. Lemmas 3.9, 3.8, and 3.7 prove the "if" part of the theorem and the necessity part of (60)–(64). We have to prove only that $(X, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ implies (54)–(59). By symmetry reasons, only (54), (56), and (58) have to be considered. However, a closer inspection to Cases 1, 3, and 5 of the proof of Lemma 3.9 shows they already prove that $(X, Y) \in H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$ implies (54), (56), and (58), respectively. □

The following result may be obtained from [5, Theorem 3.5]. We show it follows easily from Theorem 3.4.

COROLLARY 3.10. $H_1^{\mathbb{Z}} + H_2^{\mathbb{Z}} = (H_1 + H_2)^{\mathbb{Z}}$.

Proof. Apply Theorem 3.4 to the "hexagons" $K_1 := 2H_1$ and $K_2 := H_2$. The theorem characterizes the set $K_1^{\mathbb{Z}} + 2K_2^{\mathbb{Z}}$, with the symbols $a_1, a_1', A_1, b_1, b_1', B_1$ replaced by $2a_1, 2a_1', 2A_1, 2b_1, 2b_1', 2B_1$, respectively. If we consider only even values of X and Y, conditions (54)–(64) turn out to be redundant. In other words

$$(2x, 2y) \in K_1^{\mathbb{Z}} + 2K_2^{\mathbb{Z}} \;\Leftrightarrow\; (2x, 2y) \in (K_1 + 2K_2)^{\mathbb{Z}}\,.$$

As the points of $2H_2^{\mathbb{Z}}$ have even coordinates, and $2H_1^{\mathbb{Z}}$ is the set of points of $2H_1$ of even coordinates, we have

$$(2x, 2y) \in K_1^{\mathbb{Z}} + 2K_2^{\mathbb{Z}} \Leftrightarrow (x, y) \in H_1^{\mathbb{Z}} + H_2^{\mathbb{Z}}\,,$$
$$(2x, 2y) \in (K_1 + 2K_2)^{\mathbb{Z}} \Leftrightarrow (x, y) \in (H_1 + H_2)^{\mathbb{Z}}\,.$$

This proves the corollary. □

COROLLARY 3.11. *Let* $\{H_\tau : \tau \in T\}$ *be a family of "hexagons" with integer parameters such that* $(0, 0) \in H_\tau$, *except for a finite number of* τ*'s. Then* $\sum_{\tau\in T} H_\tau^{\mathbb{Z}} = \left(\sum_{\tau\in T} H_\tau\right)^{\mathbb{Z}}$.

Proof. If F denotes the (finite) set of all τ such that $(0, 0) \notin H_\tau$, the set $\sum_{\tau\in T} H_\tau$ is the union of all sets $\sum_{\tau\in X} H_\tau$, where X runs over the family of finite subsets of T that contain F. So the desired result follows by induction from the preceding corollary. □

COROLLARY 3.12. *If* H_1 *is a* 6*-vertex-hexagon (i.e.,* (40)–(43) *are strict inequalities for* $t = 1$*), then* $H^{\mathbb{Z}} = H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$.

Proof. If H_1 has six vertices, conditions (54)–(64) trivially hold. □

3.3. Proof of the main result. We go back to the notation of section 3.1; in particular, $\mathbb{P}_k$ represents the set of all prime, monic polynomials of degree k. In this section we assume that the polynomials in (3) contain only prime factors of degree 1 or 2. Then Theorem 3.1 asserts

$$\mathcal{H} = \left(\sum_{\pi\in\mathbb{P}_1} \mathcal{H}_\pi\right) + 2\left(\sum_{\pi\in\mathbb{P}_2} \mathcal{H}_\pi\right)\,.$$

Denote by H_π the set of the pairs of real numbers (x_π, y_π) satisfying (35)–(38), and let

$$H_1 := \sum_{\pi\in\mathbb{P}_1} H_\pi \quad \text{and} \quad H_2 := \sum_{\pi\in\mathbb{P}_2} H_\pi\,.$$

It is easily seen that for each π, the six parameters $a_\pi, A_\pi, a_\pi', b_\pi, B_\pi, b_\pi'$, defined just before Theorem 3.2, satisfy the inequalities (40)–(43) with the symbol t replaced by π;

hence we may apply the results of section 3.2 to the present situation. In particular, Corollary 3.11 implies

$$H_1^{\mathbb{Z}} = \textstyle\sum_{\pi\in\mathbb{P}_1} H_\pi^{\mathbb{Z}} \quad \text{and} \quad H_2^{\mathbb{Z}} = \textstyle\sum_{\pi\in\mathbb{P}_2} H_\pi^{\mathbb{Z}}\,.$$

By Theorem 3.2, $\mathcal{H}_\pi = H_\pi^{\mathbb{Z}}$. Therefore $\mathcal{H} = H_1^{\mathbb{Z}} + 2H_2^{\mathbb{Z}}$, and so Theorem 3.4 provides us with a characterization of the set $\mathcal{H}$, namely, (X, Y) *lies in* $\mathcal{H}$ *if and only if* X *and* Y *are integers that satisfy* (50)–(53) *and* (54)–(64). In this statement, a, a', A, b, b', B are given by (49), and the parameters $a_t, a'_t, A_t, b_t, b'_t, B_t$ $(t = 1, 2)$ have, in the present setting, the following values (recall $\exp_\pi(f)$ is the exponent of π in f):

$$a_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{i=1}^{n} \underline{\alpha}_i\right), \qquad b_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{j=1}^{m} \underline{\beta}_j\right),$$

$$a'_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{i=1}^{n} \gcd\{\overline{\alpha}_i, \underline{\beta}_{i+h}\}\right), \qquad b'_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{j=1}^{m} \gcd\{\overline{\beta}_j, \underline{\alpha}_j\}\right),$$

$$A_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{i=1}^{n} \overline{\alpha}_i\right), \qquad B_t := \sum_{\pi\in\mathbb{P}_t} \exp_\pi\left(\prod_{j=1}^{m} \overline{\beta}_j\right).$$

Now, if we put this together with the interlacing theorem for similarity invariant factors [6, 7], as stated and explained about (2) and (4)–(7), we immediately get Theorem 2.1.

4. Final comments and problems. The "if" part of the proof of Theorem 2.1 is clearly constructive in the sense that, given polynomials satisfying (8)–(24), we recursively built *one pair* of chains (α, β) that are the similarity invariant chains of matrices $\mathbb{A}, \mathbb{B}$ satisfying the required conditions. One of the referees suggested the much more difficult construction of an algorithm to give *all pairs* of chains that solve our problem. This is certainly possible, at least when all the prescribed polynomials (3) are nonzero, for in that case only a finite number of prime factors, say, $\pi_1, \ldots, \pi_w \in \mathbb{P}$, appear in the story, i.e., only a finite number of $\mathcal{H}_\pi$ of Theorem 3.1 are not equal to $\{(0, 0)\}$. (Note that if $\rho \in \mathbb{P}$ is not a factor of some *nonzero* polynomial in (3), then H_ρ is the recession cone [4, p. 60] of all "hexagons" H_π defined by (35)–(38). So, if one of (3) is zero, no H_π is a singleton; and thus if $\mathbb{F}$ is infinite, we have an infinite number of solutions to our problem.) In the finite case an algorithm to generate all the desired pairs of chains may be designed along the following rough lines. Note that it applies with no degree restrictions on the prime factors.

First we consider the sets $\deg(\pi_k)\mathcal{H}_{\pi_k}$, $k = 1, \ldots, w$, corresponding to the relevant prime factors; each one is a finite set of integral points that looks like the one in Figure 1, whose SW-corner V is, for the current value of k, given by $V_k = \deg(\pi_k)(a_{\pi_k}, b_{\pi_k})$; it is convenient to reduce each one of these sets to the origin, i.e., to consider the set $\mathcal{L}_k$, such that $\deg(\pi_k)\mathcal{H}_{\pi_k} = V_k + \mathcal{L}_k$. Next, n and m being given, we consider all w-tuples, $(P_1, \ldots, P_w)$, of integral pairs P_i such that

(i) $P_1 = V_1 + \cdots + V_w$;
(ii) $P_{k+1} \in P_k + \mathcal{L}_k$ for $1 \leqslant k < w$;
(iii) $(n, m) \in P_w$.

Finding such w-tuples may be visualized as a kind of brick game, where the sets $\mathcal{L}_k$ are translated in $\mathbb{Z}^2$ in such a way that the translated $\mathcal{L}_1, \ldots, \mathcal{L}_w$ form an overlapping

chain linking P_1 to (n,m). Obviously, such $(P_1,\dots,P_w)$ gives us one way of writing (n,m) as a member of the set $\mathcal{H}$ of Theorem 3.1. If x_{π_k}, y_{π_k} are the coordinates of $V_k+P_{k+1}-P_k$, we have w systems like (31)–(34), one system for each $\pi \in \{\pi_1,\dots,\pi_w\}$, that we have to solve with respect to the unknowns $x_{\pi_k i}, y_{\pi_k i}$ (all other symbols in (31)–(34) have fixed values, including the x_{π_k}, y_{π_k}!) to produce the required chains of similarity invariant factors.

So, to answer the referee's request, we still need two subroutines: the first one to play the "hexagon" overlapping game to produce all $(P_1,\dots,P_w)$ satisfying (i)–(iii), and a second routine, much easier to conceive, to generate all integral chain solutions of (31)–(34). Our main result gives conditions for the existence of at least one $(P_1,\dots,P_w)$ satisfying (i)–(iii) in the case in which $\deg(\pi_k)$ is 1 or 2.

As any other general result, Theorem 2.1 generates a corollary for each choice of polynomial chains $\underline{\alpha},\overline{\alpha},\underline{\beta},\overline{\beta}$. There are many interesting concrete problems that have not been solved so far and that can be settled by our methods. Let us give some sample problems, with hints on how a complete solution may be obtained. The examples mostly involve the minimal polynomials of our matrices, but other (groups of) similarity invariant factors may be considered instead. The first problem occurs when we prescribe the minimal polynomials of both $\mathbb{A}$ and $\mathbb{B}$.

MINIMAL POLYNOMIALS PROBLEM. *Given two nonzero monic polynomials φ and ψ, find necessary and sufficient conditions for the existence of square $\mathbb{F}$-matrices $\mathbb{A}$ and $\mathbb{B}$, of orders n and m, having minimal polynomials φ and ψ, respectively, such that $\mathbb{A}$ is a principal submatrix of $\mathbb{B}$.*

To apply the main theorem we have to assume that all prime factors of φ and ψ have degree 1 or 2. The search for a solution is organized in four steps.

Step 1. Find chains $\underline{\alpha},\overline{\alpha},\underline{\beta},\overline{\beta}$ to translate the problem in the form (4)–(7). In our example, the natural choice for our lower and upper chains is

$$\begin{aligned} \underline{\alpha} &:= (1,\dots,1,\varphi), \quad & \overline{\alpha} &:= (\varphi,\dots,\varphi,\varphi),\\ \underline{\beta} &:= (1,\dots,1,\psi), \quad & \overline{\beta} &:= (\psi,\dots,\psi,\psi). \end{aligned}$$

Step 2. These chains do not, in general, interlace in the sense of (8)–(9). So we redefine $\underline{\alpha}_i,\overline{\alpha}_i,\underline{\beta}_j,\overline{\beta}_j$ by replacing these by $\alpha_i^\iota,\alpha_i^\sigma,\beta_j^\iota,\beta_j^\sigma$ as given above (8)–(9). In the current example, the new values are

$$\left.\begin{aligned} \underline{\alpha} &:= (1,\dots,1,\varphi), \quad & \overline{\alpha} &:= (\underbrace{g,\dots,g}_{2n-m},\underbrace{\varphi,\dots,\varphi}_{m-n}),\\ \underline{\beta} &:= (1,\dots,1,\psi), \quad & \overline{\beta} &:= (\underbrace{g,\dots\ \dots\ \dots,g,g}_{n},\underbrace{\psi,\dots,\psi}_{m-n}), \end{aligned}\right\} \tag{71}$$

where g denotes $\gcd\{\varphi,\psi\}$. Clearly, if $m \geqslant 2n$, then $\overline{\alpha}_i = \varphi$ for $i = 1,\dots,n$.

Step 3. In this concrete problem, the new chains (71) satisfy (8)–(9). This means that *we already know* that there exist chains of monic polynomials α_i,β_j satisfying (4)–(6); now the problem is to force $\alpha_1\cdots\alpha_n$ to have degree n and $\beta_1\cdots\beta_m$ to have degree m. To cope with this degree requirement is, according to Theorem 2.1, the same as satisfying (10)–(24). So we replace in (10)–(24) the parameters $a, a_{\mathbb{F}}$, etc., by the values they currently have in terms of the degrees and the numbers of $\mathbb{F}$-roots of φ,ψ, and g. The 15 conditions we obtain in this way constitute a solution to our problem.

Step 4. For the sake of mathematical elegance, it is desirable to reduce this set of 15 conditions to a smaller set of "independent" conditions, i.e., the final result is

cleaner if it does not contain needless repetitions. For example, in the case in which $2n \leqslant m$, we may reduce the set of 15 conditions to a subset of only 6. Here is the result.

COROLLARY 4.1. *In the case in which $2n \leqslant m$ and all prime factors of φ and ψ have degree 1 or 2, the minimal polynomials problem has a solution if and only if the following conditions hold:*

$$(10^*) \qquad \deg \varphi \leqslant n \leqslant n \deg \varphi\,,$$

$$(11^*) \qquad \deg \psi \leqslant m \leqslant (m-n)\deg \psi + n \deg g\,,$$

$$(13^*) \qquad m + \deg \varphi \leqslant n + (m-n)\deg \psi + \deg g\,.$$

(19^*) *If equality holds in (13^*) and g has no $\mathbb{F}$-roots, then $m \equiv (m-n)\deg \psi$;*

(20^*) *If φ has no $\mathbb{F}$-roots, then n is even;*

(21^*) *If ψ has no $\mathbb{F}$-roots, then m is even.*

Proof. We use the notations a, b, c (resp., $a_{\mathbb{F}}, b_{\mathbb{F}}, c_{\mathbb{F}}$) for the degrees (resp., number of $\mathbb{F}$-roots) of the polynomials φ, ψ, g, respectively. These definitions of $a, b, a_{\mathbb{F}}, b_{\mathbb{F}}$ agree with the use made in (10)–(24); the other parameters in (10)–(24) have, in the current example, the following values:

$$a' = na\,, \qquad A = na\,, \qquad b' = (m-n)b + c\,, \qquad B = (m-n)b + nc\,,$$
$$a'_{\mathbb{F}}\,, = na_{\mathbb{F}}\,, \quad A_{\mathbb{F}}\,, = na_{\mathbb{F}}\,, \quad b'_{\mathbb{F}}\,, = (m-n)b_{\mathbb{F}} + c_{\mathbb{F}}\,, \quad B_{\mathbb{F}}\,, = (m-n)b_{\mathbb{F}} + nc_{\mathbb{F}}\,.$$

If we enter these values in (10)–(24), a corresponding system of 15 conditions is obtained; after elimination of redundant conditions, only 6 conditions survive, namely, those displayed as (10^*)–(21^*), where the starred condition (k^*) is originated by the corresponding (k), with k running from 10 to 24. We have thus eliminated 9 conditions from (10)–(24) and slightly modified the remaining 6. We give no more proof details. □

Another problem that can be solved by this method occurs when we start with a given matrix $\mathbb{A}$ and establish divisibility bounds on the minimal polynomial of $\mathbb{B}$. We give only the final result, and with no proof details.

COROLLARY 4.2. *We are given an n-square matrix $\mathbb{A}$ over $\mathbb{F}$, with similarity invariant polynomials $\alpha_1 \mid \ldots \mid \alpha_n$ and two nonzero monic polynomials ψ and Ψ such that $\psi \mid \Psi$. Assume that all prime factors of $\alpha_1 \ldots \alpha_n \psi \Psi$ have degree 1 or 2. For $m > n$, there exists an m-square matrix $\mathbb{B}$ over $\mathbb{F}$, whose minimal polynomial lies in the divisibility interval $[\psi, \Psi]$ and has $\mathbb{A}$ as a principal submatrix if and only if* (i) *$\alpha_{2n-m} \mid \Psi$;* (ii) *$b \leqslant m \leqslant B$, where*

$$b := \deg \operatorname{lcm}\{\alpha_{2n-m}, \psi\} + \sum_{i<2n-m} \deg \alpha_i \quad \text{and} \quad B := \sum_{j=1}^{m} \deg \gcd\{\alpha_j, \Psi\}\,;$$

and (iii) *$m - b$ is even whenever the polynomials $\gcd\{\alpha_i, \Psi\}/\alpha_{i-2m+2n}$, for $i < m$, and $\Psi/\operatorname{lcm}\{\alpha_{2n-m}, \psi\}$ have no roots in $\mathbb{F}$.*

Of course we might also consider and easily solve the "dual" problem where the chain of similarity invariant factors of $\mathbb{B}$ is fixed and the minimal polynomial of $\mathbb{A}$ is given a divisibility bound. The result and proof are left to the reader.

This kind of problem may as well include prescriptions on the spectrum (i.e., the *set* of eigenvalues) of a matrix because that is the set of $\bar{\mathbb{F}}$-roots of its minimal

polynomial. So the sentence *"the spectrum of* $\mathbb{A}$ *is contained in a finite set* Λ *and contains a set* Σ*,"* with both sets contained in $\overline{\mathbb{F}}$, may be expressed in the form $\underline{\alpha}_n \,|\, \alpha_n \,|\, \overline{\alpha}_n$ for appropriately chosen polynomials $\underline{\alpha}_n$ and $\overline{\alpha}_n$.

Acknowledgments. We are indebted to the referees for very carefully reading the paper, for giving us valuable suggestions to improve the presentation and contents of the paper, and for providing technical corrections in section 3.2.

REFERENCES

[1] I. Baragaña and I. Zaballa, *Column completion of a pair of matrices*, Linear and Multilinear Algebra, 27 (1990), pp. 243–273.
[2] I. Cabral and F. C. Silva, *Unified theorems on completions of matrix pencils*, Linear Algebra Appl., 159 (1991), pp. 43–54.
[3] I. Cabral and F. C. Silva, *Similarity invariants of completions of submatrices*, Linear Algebra Appl., 169 (1992), pp. 151–161.
[4] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[5] E. M. Sá and J. D. Vieira, *Matrices and submatrices with constrained invariant factors*, Portugal. Math., 48 (1991), pp. 113–142.
[6] E. M. Sá, *Imbedding conditions for λ-matrices*, Linear Algebra Appl., 24 (1979), pp. 33–50.
[7] R. C. Thompson, *Interlacing inequalities for invariant factors*, Linear Algebra Appl., 24 (1979), pp. 1–31.
[8] I. Zaballa, *Matrices with prescribed rows and invariant factors*, Linear Algebra Appl., 87 (1987), pp. 113–146.
[9] I. Zaballa, *Interlacing inequalities and control theory*, Linear Algebra Appl., 101 (1988), pp. 9–31.
[10] I. Zaballa, *Similarity and block similarity*, Linear Algebra Appl., 212/213 (1994), pp. 461–485.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 2, pp. 581–599
© 2001 Society for Industrial and Applied Mathematics

STOCHASTIC AUTOMATA NETWORKS AND NEAR COMPLETE DECOMPOSABILITY*

OLEG GUSAK†, TUĞRUL DAYAR†, AND JEAN-MICHEL FOURNEAU‡

Abstract. Stochastic automata networks (SANs) have been developed and used in the last fifteen years as a modeling formalism for large systems that can be decomposed into loosely connected components. In this work, we extend the near complete decomposability concept of Markov chains (MCs) to SANs so that the inherent difficulty associated with solving the underlying MC can be forecasted and solution techniques based on this concept can be investigated. A straightforward approach to finding a nearly completely decomposable (NCD) partitioning of the MC underlying a SAN requires the computation of the nonzero elements of its global generator. This is not feasible for very large systems even in sparse matrix representation due to memory and execution time constraints. We devise an efficient decompositional solution algorithm to this problem that is based on analyzing the NCD structure of each component of a given SAN. Numerical results show that the given algorithm performs much better than the straightforward approach.

Key words. Markov chains, stochastic automata networks, near complete decomposability, state classification

AMS subject classifications. 60J27, 60J10, 65F30, 65F10, 65F50

PII. S089547980036975X

1. Introduction.

Stochastic automata networks (SANs) [16, 18, 13, 17, 19, 21, 22, 9, 1, 6, 12, 24, 3] provide a methodology for modeling large systems with interacting components. The main idea is to decompose the system of interest into its components and to model each component separately. Once this is done, interactions and dependencies among components can be brought into the picture and the model finalized. With this decompositional approach, the global system ends up having as many states as the product of the number of states of the individual components. The benefit of the SAN approach is twofold. First, each component can be modeled much easier compared to the global system due to state space reduction. Second, space required to store the description of components is minimal compared to the case in which transitions from each global state are stored explicitly. However, all this happens at the expense of increased analysis time [13, 22, 1, 9, 6, 12, 24, 3].

An intimately related way of coping with the state space explosion problem is to consider hierarchical decompositions arising in queueing network and superposed stochastic Petri Net formalisms [4, 2, 5]. SANs which do not have dependencies among automata are, in fact, a special case of hierarchical Markovian models. Although somewhat distant from the problem domain compared to the SAN approach, there are recent results showing that hierarchical representations lend themselves naturally to distributed steady state analysis (see [5, p. 79]).

An important issue in choosing an efficient iterative solver for SANs is the conditioning [15] associated with the underlying Markov chain (MC). Recent numerical

*Received by the editors March 27, 2000; accepted for publication (in revised form) by D. O'Leary May 4, 2001; published electronically November 13, 2001. This work was supported by grant TÜBİTAK-CNRS.

http://www.siam.org/journals/simax/23-2/36975.html

†Department of Computer Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey (gusak@cs.bilkent.edu.tr, tugrul@cs.bilkent.edu.tr).

‡Lab. PRiSM, Université de Versailles, 45 Avenue des États-Unis, 78035 Versailles Cedex, France (jmf@prism.uvsq.fr).

experiments [11] show that two-level iterative solvers perform very well with nearly completely decomposable (NCD) partitionings [8] having balanced block sizes when the MC to be solved for its steady state vector is ill-conditioned. Block iterative methods based on classical splittings (block Jacobi, block Gauss–Seidel, block SOR) for SANs are introduced in [24]. Results with iterative aggregation-disaggregation type [23, 20, 10, 11] solvers for SANs appear in [1]. However, two-level iterative solvers considered so far do not exploit NCD partitionings. It should be emphasized that iterative aggregation-disaggregation based on NCD partitionings has certain rate of convergence guarantees [20] that may be useful for very large MCs.

In this paper, we extend the concept of near complete decomposability to SANs so that the inherent difficulty associated with solving the underlying MC can be forecasted and solution techniques based on this concept can be investigated. In doing this, we utilize the graph theoretic ideas for SANs given in [13]. In the next section, we review basic concepts of the SAN formalism and introduce NCD MCs. In section 3, we make assumptions regarding the description of a continuous-time SAN model and discuss how we proceed when we encounter an underlying MC with transient states and/or multiple essential subsets of states. In section 4, we present a three step algorithm that finds an NCD partitioning of the MC underlying a SAN based on a user specified decomposability parameter without computing the global generator matrix. In the first three subsections we discuss the three steps of the proposed algorithm, and in the last subsection we give a summary of its complexity analysis. Numerical results with the algorithm on a SAN model are presented in section 5. We conclude in section 6.

The extended version of this paper can be found in [14]. Therein, we discuss in more detail the approach presented in this paper and provide the algorithms for each of the three steps of the NCD partitioning algorithm introduced here, their detailed complexity analysis, and the results of experiments with two other SAN models.

2. Background. In the next two subsections, we discuss basic concepts related to the SAN formalism as a modeling paradigm and introduce NCD MCs.

2.1. SAN overview. In a SAN (see [21, Chapter 9]), each component of the global system is modeled by a stochastic automaton. When automata do not interact (i.e., when they are independent of each other), description of each automaton consists of local transitions only. In other words, local transitions are those that affect the state of one automaton. Local transitions can be constant (i.e., independent of the state of other automata) or they can be functional. In the latter case, the transition is a nonnegative real valued function that depends on the state of other automata. Interactions among components are captured by synchronizing transitions. Synchronization among automata happens when a state change in one automaton causes a state change in other automata. Similar to local transitions, synchronizing transitions can be constant or functional.

A continuous-time system of N components can be modeled by a single stochastic automaton for each component. Local transitions of automaton $i \in \{1, 2, \ldots, N\}$ (denoted by $\mathcal{A}^{(i)}$) are modeled by the local transition rate matrix $Q_l^{(i)}$. When there are E synchronizing events in the system, $\mathcal{A}^{(i)}$ has the corresponding synchronizing transition matrix $Q_{e_j}^{(i)}$ that represents its contribution to synchronization $j \in \{1, 2, \ldots, E\}$ and associated with it the diagonal corrector matrix $\bar{Q}_{e_j}^{(i)}$. The automaton that triggers a synchronizing event is called the master; the others that get affected by the event are called the slaves. Matrices associated with synchronizing events are either

transition rate matrices (corresponding to master automata) or transition probability matrices (corresponding to slave automata). If $\mathcal{A}^{(i)}$, $i \in \{1, 2, \ldots, N\}$, is not involved in event j, then $Q_{e_j}^{(i)} = \bar{Q}_{e_j}^{(i)} = I_{n_i}$, where n_i is the number of states in $\mathcal{A}^{(i)}$ and I_{n_i} is the identity matrix of order n_i.

The continuous-time Markov chain (CTMC) underlying the global system can be obtained from

$$Q = \bigoplus_{i=1}^{N} Q_l^{(i)} + \sum_{j=1}^{E} \bigotimes_{i=1}^{N} Q_{e_j}^{(i)} + \sum_{j=1}^{E} \bigotimes_{i=1}^{N} \bar{Q}_{e_j}^{(i)}, \tag{1}$$

where $\bigoplus$ is the tensor sum operator and $\bigotimes$ is the tensor product operator (see [7]). We refer to the tensor representation in (1) associated with the CTMC as the descriptor of the SAN. When there are functional elements, tensor products become generalized tensor products [19]. Assuming that the states of automata and the global states are numbered starting from 1, the global state s that corresponds to the state vector $(s_1, s_2, \ldots, s_N)$ is given by $s = 1 + \sum_{i=1}^{N}(s_i - 1)\prod_{k=i+1}^{N} n_k$, where $s_i \in \{1, 2, \ldots, n_i\}$ denotes the state of $\mathcal{A}^{(i)}$.

2.2. NCD MCs. NCD MCs [15] are irreducible stochastic matrices that can be symmetrically permuted [8] to the block form

$$P_{n\times n} = \begin{pmatrix} P_{11} & P_{12} & \ldots & P_{1K} \\ P_{21} & P_{22} & \ldots & P_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ P_{K1} & P_{K2} & \ldots & P_{KK} \end{pmatrix}$$

in which the nonzero elements of the off-diagonal blocks are small compared with those of the diagonal blocks [21, p. 286]. Hence, it is possible to represent an NCD MC as

$$P = \operatorname{diag}(P_{11}, P_{22}, \ldots, P_{KK}) + E,$$

where the diagonal blocks P_{ii} are square and possibly of different order. The quantity $\|E\|_\infty$ is referred to as the degree of coupling and is taken to be a measure of the decomposability of P. When the chain is NCD, it has eigenvalues close to 1, and the poor separation of the unit eigenvalue implies a slow rate of convergence for standard matrix iterative methods [10, p. 290]. Hence, NCD MCs are said to be ill-conditioned [15, p. 258]. We should remark that the definition of NCDness is given through a discrete-time Markov chain (DTMC). The underlying CTMC of a SAN can be transformed through uniformization [21, p. 24] to a DTMC for the purpose of computing its steady state vector as in

$$P = I + \frac{1}{\alpha}Q, \tag{2}$$

where $\alpha \geq \max_{1\leq i\leq n} |Q(i,i)|$. To preserve NCDness in this transformation, α must be chosen as $\max_{1\leq i\leq n} |Q(i,i)|$.

An NCD partitioning of P corresponding to a user specified decomposability parameter ϵ can be computed as follows (see [8] for details). First, construct an undirected graph whose vertices are the states of P by introducing an edge between vertices i and j if $P(i,j) \geq \epsilon$ or $P(j,i) \geq \epsilon$, and then identify its connected components[1] (CCs). Each CC forms a subset of the NCD partitioning. Notice that for

[1]Not to be confused with the word *component*, which we have been using so far to mean "subsystem."

a given value of ϵ, the maximum number of subsets in a computed partitioning is unique.

3. On continuous-time SAN descriptions and state classification. There is no standard specification for the description of a SAN model. In the next subsection, we state definitions and propositions that enable us to transform a continuous-time SAN description to one that is more convenient to work with when developing the NCD partitioning algorithm.

3.1. Description of a continuous-time SAN model. Without loss of generality, we restrict ourselves to the case in which row sums of synchronizing transition probability matrices are either 0 or 1.

DEFINITION 1. *A SAN description is said to be proper if and only if each synchronizing transition probability matrix has row sums of* 0 *or* 1.

The SAN descriptions of the three applications we consider in the numerical experiments are proper. However, in a given SAN description, row sums between 0 and 1 can very well be present in synchronizing transition rate matrices. Proposition 1 shows what should be done when such a case is encountered.

PROPOSITION 1. *A given SAN description can be transformed to a SAN description that is proper.*

Proof. Without loss of generality, consider a SAN description of N automata and one synchronizing event. There are two possible cases. In the first case, row sums of the synchronizing transition probability matrix $Q_{e_1}^{(k)}$ corresponding to slave automaton k are all equal to some constant β such that $0 < \beta < 1$. This is the trivial case; we can replace $Q_{e_1}^{(k)}$ with $\hat{Q}_{e_1}^{(k)} = \frac{1}{\beta} Q_{e_1}^{(k)}$ and $Q_{e_1}^{(m)}$ with $\hat{Q}_{e_1}^{(m)} = \beta Q_{e_1}^{(m)}$, where m is the index of the master automaton of the synchronizing event. Row sums of the transformed matrix $\hat{Q}_{e_1}^{(k)}$ are 1. In the second case, row sums of $Q_{e_1}^{(k)}$ are not equal, and some are between 0 and 1. This implies that transition rates of the master automaton m of the synchronizing event depend on the state of automaton k. Therefore, it is possible to replace $Q_{e_1}^{(k)}$ with a matrix that has row sums of 0 or 1 by introducing functional transitions to $Q_{e_1}^{(m)}$ as follows. Let β_l, $l = 1, 2, \ldots, n_k$, be the sum of row l in $Q_{e_1}^{(k)}$. We replace $Q_{e_1}^{(k)}$ with $\hat{Q}_{e_1}^{(k)}$ in which $\hat{Q}_{e_1}^{(k)}(i,j) = Q_{e_1}^{(k)}(i,j)/\beta_i$ if $0 < \beta_i < 1$, else $\hat{Q}_{e_1}^{(k)}(i,j) = Q_{e_1}^{(k)}(i,j)$, for $j = 1, 2, \ldots, n_k$. We also replace $Q_{e_1}^{(m)}$ with $\hat{Q}_{e_1}^{(m)}$ in which $\hat{Q}_{e_1}^{(m)}(i,j) = \beta_l Q_{e_1}^{(m)}(i,j)$ if $0 < \beta_l < 1$, else $\hat{Q}_{e_1}^{(m)}(i,j) = Q_{e_1}^{(m)}(i,j)$, for $i, j = 1, 2, \ldots, n_m$ when $\mathcal{A}^{(k)}$ is in state l. The transformed matrix $\hat{Q}_{e_1}^{(k)}$ has row sums of 0 or 1.

Given a synchronizing event, the above modifications must be made for each of its synchronizing transition probability matrices that has row sums between 0 and 1. After modifying the synchronizing event matrices, the corresponding diagonal corrector matrices must also be modified accordingly. The new SAN description has synchronizing transition probability matrices with row sums of 0 or 1, and therefore is proper.

The generalization to E (> 1) synchronizing events is straightforward. ∎

Observe that the transformation of a SAN description discussed in the proof of Proposition 1 may cause the number of functional elements in the synchronizing transition rate matrices of automata to increase. However, the number of synchronizing events as well as the nonzero structure of the synchronizing transition matrices of automata remain unchanged.

Now we introduce a definition related to the separability of synchronizing transition rates from local transition rates.

DEFINITION 2. *Synchronizations are separable from local transitions in a given SAN description if and only if for any synchronizing event t whose master is automaton m and $i, j = 1, 2, \ldots, n_m$, $Q_{e_t}^{(m)}(i,j) \neq 0$ implies $Q_l^{(m)}(i,j) = 0$.*

Definition 2 may seem to be specifying an artificial condition at first, yet the condition is satisfied by the three applications we consider. As we shall see in the next section, this property enables the preprocessing of local transition rate matrices separately from synchronizing transition matrices which significantly improves the complexity of the NCD partitioning algorithm we propose. Even though the three SAN descriptions we consider have separable synchronizations, one may very well encounter those that do not satisfy this property. Proposition 2 shows that a SAN description whose synchronizations are not separable can be handled in the framework discussed in this paper.

PROPOSITION 2. *A given SAN description can be transformed to a new SAN description whose synchronizations are separable from local transitions.*

Proof. Assume that the given SAN description does not satisfy the condition in Definition 2. Without loss of generality, let t be the event, m its master, and (i,j) the indices of the problematic element. Decompose $Q_l^{(m)}$ into three terms as

$$Q_l^{(m)} = R_l^{(m)} + Q_l^{(m)}(i,j) u_i u_j^T - Q_l^{(m)}(i,j) u_i u_i^T,$$

where u_i is the ith column of the identity matrix. Here $R_l^{(m)}$ is a transition rate matrix; the second term is a matrix with a single nonzero transition rate at element (i,j); and the third term is the diagonal corrector of the second term. Now, let $R_l^{(m)}$ be the local transition rate matrix of automaton m, and introduce the new synchronizing event v with master automaton m; $Q_{e_v}^{(m)} (= Q_l^{(m)}(i,j) u_i u_j^T)$ is the rate matrix associated with automaton m and synchronizing event v, and $\bar{Q}_{e_v}^{(m)}$ $(= -Q_l^{(m)}(i,j) u_i u_i^T)$ is its diagonal corrector. All other matrices corresponding to synchronizing event v are equal to identity. Now, recall the following identity from tensor algebra:

$$\begin{aligned} & A \bigoplus (Q_l^{(m)} + Q_{e_v}^{(m)} + \bar{Q}_{e_v}^{(m)}) \bigoplus B \\ & \quad = \left(A \bigoplus Q_l^{(m)} \bigoplus B \right) + \left(I \bigotimes Q_{e_v}^{(m)} \bigotimes I \right) + \left(I \bigotimes \bar{Q}_{e_v}^{(m)} \bigotimes I \right). \end{aligned}$$

Compare its right-hand side with (1). The new SAN description has separable synchronizations.

The generalization to the cases when event t has more than one problematic element and the SAN description has more than one synchronizing event that are not separable from local transitions is straightforward. □

The number of synchronizing events in the new SAN description obtained through the transformation discussed in the proof of Proposition 2 is larger than the number of synchronizing events in the original SAN. The difference in the number of synchronizing events corresponds to the number of the synchronizing events in the original SAN that are not separable. Nevertheless, assuming that identity matrices are not stored explicitly, the described transformation does not increase the number of nonzeros in the transformed SAN description.

Our next definition related to the SAN description involves the number of nonzero elements in synchronizing transition rate matrices. Without loss of generality, we restrict ourselves to the case where all synchronizing events in a SAN are simple.

DEFINITION 3. *Synchronizations in a given SAN description are simple if and only if for any synchronizing event t whose master is automaton m, $Q_{e_t}^{(m)}$ has only one nonzero element.*

In a SAN description whose synchronizations are simple, each synchronizing event can be characterized by a value that corresponds to the synchronizing transition rate of the event. In the next section, we show how to take advantage of this property. In most of the cases, we will not encounter SAN descriptions whose synchronizations are simple. The next proposition shows how SAN descriptions that do not satisfy the condition of Definition 3 can be handled in the framework of our approach.

PROPOSITION 3. *A given SAN description can be transformed to a new SAN description whose synchronizing events are all simple.*

Proof. Assume that the given SAN description does not satisfy the condition in Definition 3. Without loss of generality, let t be the event, m its master, and nz the number of nonzeros in $Q_{e_t}^{(m)}$. Decompose $Q_{e_t}^{(m)}$ into nz simple synchronizing transition rate matrices thereby creating nz new synchronizing events with master automaton m. The slave automata of the new synchronizing events are the slave automata of synchronizing event t. The transition probability matrices and their diagonal correctors associated with the new slave automata are, respectively, equal to the transition probability matrix and its diagonal corrector associated with the slave automata for synchronizing event t. All other matrices corresponding to the new synchronizing events are equal to identity. The new SAN description has simple synchronizations.

The generalization to E (> 1) synchronizing events that are not simple is straightforward. □

Application of the transformation described in the proof of Proposition 3 to a SAN description whose synchronizing events are not simple leads to an increase in the number of synchronizing events. The number of the simple synchronizing events in the new SAN description is equal to the number of nonzero elements in the synchronizing transition rate matrices of the original SAN. Note that the described transformation does not change the synchronizing transition probability matrices and their diagonal correctors. Hence, it is possible to keep the number of nonzero elements in the new SAN description the same as in the original SAN description.

In the next subsection, we discuss how we proceed when we encounter an underlying MC with transient states and/or multiple essential subsets of states.

3.2. State classification in SANs. As discussed in subsection 2.2, NCD MCs are irreducible by definition. However, the MC underlying a SAN may very well be reducible. When the MC underlying the given SAN has transient states and/or multiple essential subsets of states, NCD analysis can be carried out on the essential subsets of states one subset at a time. We name the states that do not belong to the essential subset of interest as uninteresting. We remark that uninteresting states should be omitted from further consideration when running the NCD partitioning algorithm.

We have implemented a state classification (SC) algorithm that classifies the states in the global state space of a SAN into essential and transient subsets following [21, pp. 25–26]. The detailed description of the SC algorithm is given in [14]. The input parameters of the SC algorithm are local transition rate matrices and synchronizing event matrices of the SAN. The output of the algorithm is an integer array of length n in which states corresponding to the essential subset of interest are marked.

4. NCD partitioning algorithm for SANs. The following is our proposed solution algorithm that computes NCD partitionings of the MC underlying a SAN without generating Q (or P).

ALGORITHM 1. NCD PARTITIONING OF MC UNDERLYING SAN FOR GIVEN ϵ.

Step 1. $Q \to P$ transformation.

Step 2. Preprocessing synchronizing events.

Step 3. Constructing NCD connected components.

Step 1 computes the scalar α in (2) that describes the transformation of the global generator Q to a DTMC P through uniformization. In the next subsection, we show how this can be achieved efficiently by inspecting the diagonal elements in local transition rate matrices and the nonzero elements in diagonal corrector matrices.

Step 2 considers the locations of off-diagonal nonzero elements in the global generator Q. Off-diagonal nonzero elements in local transition rate matrices cannot contribute to the same nonzero element in Q due to the fact that these matrices form a tensor sum. Hence, their analysis is straightforward. However, off-diagonal nonzero elements in synchronizing transition rate matrices may contribute to the same nonzero element in Q since these matrices form a sum of tensor products. Therefore, it is necessary to identify those synchronizing events that may influence the NCD partitioning of the MC underlying the SAN by contributing to the value of the same nonzero element in Q. In subsection 4.2, we explain how this is done.

Finally, Step 3 determines the NCD CCs by analyzing local transition rate matrices and matrices corresponding to synchronizing events identified in Step 2 using ϵ and the value of α computed in Step 1. This is discussed in subsection 4.3.

4.1. $Q \to P$ transformation. The CTMC Q can be transformed to a DTMC P using (2) after $\alpha = \max_{1 \le i \le n} |Q(i,i)|$ is computed. Since Q is a CTMC, we have $Q(i,i) = -\sum_{j \neq i} Q(i,j)$ for $i = 1, 2, \ldots, n$. Note also that only the off-diagonal elements in P contribute to NCDness. Regarding the off-diagonal elements in Q, which determine the off-diagonal elements in P, we make the following observations.

REMARK 1. *Each nonzero local transition rate in a SAN contributes to a different off-diagonal element in Q; two or more nonzero local transition rates cannot contribute to the same off-diagonal element in Q.*

This observation follows immediately from the term $\bigoplus_{i=1}^{N} Q_l^{(i)}$ in (1) and the definition of tensor sum.

REMARK 2. *A nonzero off-diagonal element in Q for a SAN with separable synchronizations is formed either of a nonzero local transition rate or of nonzero synchronizing transition rates but not of both.*

This observation follows from the definition of the SAN descriptor in (1) and Definition 2.

From Remarks 1 and 2 and from (1) and (2), P without its main diagonal follows as $P^* = \bigoplus_{i=1}^{N} (\frac{1}{\alpha} Q_l^{(i)}) + \sum_{j=1}^{E} \bigotimes_{i=1}^{N} \hat{Q}_{e_j}^{(i)}$, where $\hat{Q}_{e_j}^{(i)} = \frac{1}{\alpha} Q_{e_j}^{(i)}$ if $\mathcal{A}^{(i)}$ is the master of event j; otherwise, $\hat{Q}_{e_j}^{(i)} = Q_{e_j}^{(i)}$.

REMARK 3. *Dependencies among automata may arise either as explicit functions whose values depend on the states of automata other than the ones in which they are defined or implicitly by the existence of zero rows in synchronizing event matrices associated with slave automata. The latter case corresponds to the disabling of the synchronized transition when the slave automaton is in local state corresponding to the zero row.*

From now on, by dependencies we refer to both explicit and implicit dependencies

as discussed in Remark 3. A naive solution for a SAN having dependencies is to compute explicitly each diagonal element of Q and to find the element with maximum magnitude. However, this is expensive. To reduce the complexity, we propose to partition automata into dependency sets.

DEFINITION 4. *Let $G(\mathcal{V},\mathcal{E})$ be a digraph in which v_i corresponds to $\mathcal{A}^{(i)}$ and $(v_i,v_j)\in\mathcal{E}$ if transitions in $\mathcal{A}^{(i)}$ depend on the state of $\mathcal{A}^{(j)}$ either explicitly or implicitly as discussed in Remark* 3. *Then, the dependency sets of a SAN, denoted by $\mathcal{D}_k$, $k=1,2,\ldots,N_{\mathcal{D}}$, are the connected components of the dependency graph G.*

Assuming that the dependency sets of the SAN are known and referring to

$$\max\_\mathcal{D}_k = \max\left|\bigoplus_{i,\mathcal{A}^{(i)}\in\mathcal{D}_k}\operatorname{diag}(Q_l^{(i)}) + \sum_{j,e_j\in\mathcal{M}_{\mathcal{D}_k}}\ \bigotimes_{i,\mathcal{A}^{(i)}\in\mathcal{D}_k}\operatorname{diag}(\bar{Q}_{e_j}^{(i)})\right| \tag{3}$$

as the maximum of the dependency set $\mathcal{D}_k$, where diag returns a vector consisting of the diagonal elements of its matrix argument and $\mathcal{M}_{\mathcal{D}_k}$ is the set of synchronizing events whose masters are in $\mathcal{D}_k$, the diagonal element with maximum magnitude of the MC underlying a SAN can be obtained from

$$\alpha = \sum_{k=1}^{N_{\mathcal{D}}}\max\_\mathcal{D}_k. \tag{4}$$

The proof of this result is given in [14].

Observe that (4) is valid for irreducible MCs underlying SANs. When transient states and/or multiple essential subsets of states are present, the diagonal element with maximum magnitude given by (4) may not belong to the essential subset of interest (see subsection 3.2). In the presence of uninteresting states, we can compute α by finding the maximums of all $N_{\mathcal{D}}$ dependency sets (see (3) and (4)). For dependency set $\mathcal{D}_k$, this task amounts to the enumeration of $\prod_{i,\mathcal{A}^{(i)}\in\mathcal{D}_k} n_i$ states and an equal number of floating-point comparisons. Now, observe that to $\max\_\mathcal{D}_k$ of the dependency set $\mathcal{D}_k$ corresponds a state S_k. Hence, if the global state s that corresponds to $S_1,S_2,\ldots,S_{N_{\mathcal{D}}}$ maps into the essential subset of interest, then α given by (4) is taken as the diagonal element with maximum magnitude. However, if s is an uninteresting state, we omit from further consideration the element corresponding to $\max\_\mathcal{D}_k$ for $k=1,2,\ldots,N_{\mathcal{D}}$ and proceed as in the following paragraph.

In the first step, for $k=1,2,\ldots,N_{\mathcal{D}}$ we find the next largest value denoted by $\text{next\_max\_}\mathcal{D}_k$ from (3) and the corresponding state $\tilde{S}_k$. In order to find $\text{next\_max\_}\mathcal{D}_k$ rapidly, the vectors

$$\left|\bigoplus_{i,\mathcal{A}^{(i)}\in\mathcal{D}_k}\operatorname{diag}(Q_l^{(i)}) + \sum_{j,e_j\in\mathcal{M}_{\mathcal{D}_k}}\ \bigotimes_{i,\mathcal{A}^{(i)}\in\mathcal{D}_k}\operatorname{diag}(\bar{Q}_{e_j}^{(i)})\right|,\quad k=1,2,\ldots,N_{\mathcal{D}},$$

should be stored as sorted. In the second step, we find t such that $\text{next\_max\_}\mathcal{D}_t \geq \text{next\_max\_}\mathcal{D}_k$ for $k=1,2,\ldots,N_{\mathcal{D}}$. Finally, we replace $\max\_\mathcal{D}_t$ with $\text{next\_max\_}\mathcal{D}_t$, S_t with $\tilde{S}_t$, and omit the element corresponding to $\text{next\_max\_}\mathcal{D}_t$ from further consideration. If the updated global state s maps to a state in the essential subset of interest, then α given by (4) is taken as the diagonal element with maximum magnitude. Else we go back to the first step. Since finite MCs have at least one recurrent state in each essential subset, the algorithm is terminating.

Our final remark is about the special case of a SAN with a single dependency set; that is, $N_{\mathcal{D}} = 1$ and $\mathcal{D}_1 = \{\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \ldots, \mathcal{A}^{(N)}\}$. In this case, finding $\alpha = \max\_\mathcal{D}_1$ amounts to enumerating all diagonal elements of Q since we have the equality $\bigoplus_{i,\mathcal{A}^{(i)}\in\mathcal{D}_1} \operatorname{diag}(Q_l^{(i)}) + \sum_{j,e_j\in\mathcal{M}_{\mathcal{D}_k}} \bigotimes_{i,\mathcal{A}^{(i)}\in\mathcal{D}_1} \operatorname{diag}(\bar{Q}_{e_j}^{(i)}) = \operatorname{diag}(Q)$. Therefore, for a SAN with a single dependency set, there is no need to sort and store $\operatorname{diag}(Q)$ as suggested. When finding the maximum of $\operatorname{diag}(Q)$, we test an element of $\operatorname{diag}(Q)$ only if its index corresponds to a state in the essential subset of interest.

Example. This example shows the computation of the diagonal element with maximum magnitude of Q for the following SAN that has functional and synchronizing transitions. The parameters are $N = 3$, $E = 2$, $n_1 = 2$, $n_2 = 3$, $n_3 = 2$; $f = 3$ when $\mathcal{A}^{(1)}$ is in state 1, and $f = 5$ when $\mathcal{A}^{(1)}$ is in state 2. The master of synchronizing event 1 is $\mathcal{A}^{(3)}$, and the master of synchronizing event 2 is $\mathcal{A}^{(2)}$. The matrices are

$$Q_l^{(1)} = \begin{pmatrix} -2 & 2 \\ 1 & -1 \end{pmatrix}, \quad Q_l^{(2)} = \begin{pmatrix} -2 & 2 & 0 \\ 2 & -5 & 3 \\ 1 & 3 & -4 \end{pmatrix}, \quad Q_l^{(3)} = \begin{pmatrix} -f & f \\ 0 & 0 \end{pmatrix},$$

$$Q_{e_1}^{(1)} = \bar{Q}_{e_1}^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad Q_{e_2}^{(1)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \bar{Q}_{e_2}^{(1)} = I,$$

$$Q_{e_1}^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \bar{Q}_{e_1}^{(2)} = I, \quad Q_{e_2}^{(2)} = \begin{pmatrix} 0 & 0 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \bar{Q}_{e_2}^{(2)} = \begin{pmatrix} -5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$Q_{e_1}^{(3)} = \begin{pmatrix} 0 & 0 \\ 5 & 0 \end{pmatrix}, \quad \bar{Q}_{e_1}^{(3)} = \begin{pmatrix} 0 & 0 \\ 0 & -5 \end{pmatrix}, \quad Q_{e_2}^{(3)} = \bar{Q}_{e_2}^{(3)} = I.$$

The given SAN has two dependency sets: $\mathcal{D}_1 = \{\mathcal{A}^{(1)}, \mathcal{A}^{(3)}\}$ and $\mathcal{D}_2 = \{\mathcal{A}^{(2)}\}$. Note that $\mathcal{A}^{(3)}$ functionally depends on the state of $\mathcal{A}^{(1)}$ due to functional transition f as well as due to synchronizing event 1 (see $\bar{Q}_{e_1}^{(1)}$). Hence, the diagonal element with maximum magnitude of Q is comprised of two terms. The maximum of $\mathcal{D}_1$ is given by

$$\begin{aligned} \max\_\mathcal{D}_1 &= \max \left| \operatorname{diag}(Q_l^{(1)}) \bigoplus \operatorname{diag}(Q_l^{(3)}) + \operatorname{diag}(\bar{Q}_{e_1}^{(1)}) \bigotimes \operatorname{diag}(\bar{Q}_{e_1}^{(3)}) \right| \\ &= \max \left| \begin{pmatrix} -2-f \\ -2-0 \\ -1-f \\ -1-3 \end{pmatrix} + \begin{pmatrix} 0 \\ -5 \\ 0 \\ 0 \end{pmatrix} \right| = \max \left| \begin{pmatrix} -5 \\ -2 \\ -6 \\ -4 \end{pmatrix} + \begin{pmatrix} 0 \\ -5 \\ 0 \\ 0 \end{pmatrix} \right| = 7. \end{aligned}$$

On the other hand, $\mathcal{D}_2$ is a singleton, and therefore the maximum of $\mathcal{D}_2$ is given by

$$\max\_\mathcal{D}_2 = \max \left| \operatorname{diag}(Q_l^{(2)}) + \operatorname{diag}(\bar{Q}_{e_2}^{(2)}) \right| = \max \left| \begin{pmatrix} -2 \\ -5 \\ -4 \end{pmatrix} + \begin{pmatrix} -5 \\ 0 \\ 0 \end{pmatrix} \right| = 7.$$

Since the underlying MC is irreducible, $\alpha = \max\_\mathcal{D}_1 + \max\_\mathcal{D}_2 = 14$ as verified on

$$Q = \begin{pmatrix}
-12 & 3 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 5 & 0 \\
0 & -14 & 0 & 2 & 5 & 0 & 0 & 2 & 0 & 0 & 0 & 5 \\
2 & 0 & -10 & 3 & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
5 & 2 & 0 & -12 & 0 & 3 & 0 & 0 & 0 & 2 & 0 & 0 \\
1 & 0 & 3 & 0 & -9 & 3 & 0 & 0 & 0 & 0 & 2 & 0 \\
5 & 1 & 0 & 3 & 0 & -11 & 0 & 0 & 0 & 0 & 0 & 2 \\
1 & 0 & 0 & 0 & 5 & 0 & -13 & 5 & 2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 5 & 0 & -8 & 0 & 2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & -11 & 5 & 3 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & -6 & 0 & 3 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 3 & 0 & -10 & 5 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 3 & 0 & -5
\end{pmatrix}.$$

As pointed out at the beginning of this subsection, an NCD partitioning of P that corresponds to a user specified decomposability parameter ϵ is determined by the off-diagonal elements in P. In the next subsection we concentrate on those off-diagonal elements that originate from the synchronizing transition rates of the SAN.

4.2. Preprocessing synchronizing events. Transition rates from different synchronizing event matrices may sum up to form a nonzero in the generator matrix Q. Hence, in some cases it may not be possible to determine the value of an off-diagonal element in Q by inspecting each automaton separately. The aim of Step 2 in Algorithm 1 is to find sets of those synchronizing events that may influence the NCD partitioning of Q. We name these sets as potential sets of synchronizing events. The potential sets are disjoint, and their union is a subset of the set of synchronizing events. The input parameters of Step 2 are synchronizing event matrices, ϵ, and α computed in Step 1. The output of Step 2 is $N_{\mathcal{P}}$ potential sets denoted by $\mathcal{P}_r$, $r = 1, 2, \ldots, N_{\mathcal{P}}$.

There are two cases in which synchronizing events may influence the NCD partitioning of Q. First, a simple synchronizing event has the corresponding transition rate greater than or equal to $\alpha\epsilon$. Second, a set of synchronizing events contribute to the same element in Q, and the sum of the synchronizing transition rates of the events in the set is greater than or equal to $\alpha\epsilon$.

In the first case, each synchronizing event with transition rate greater than or equal to $\alpha\epsilon$ forms a potential set that is a singleton. When the transition rate of a synchronizing event is a function, its value can be evaluated only on the global state space. This can be done in Step 3 of Algorithm 1 when NCD CCs of the SAN are formed. Hence, if the synchronizing transition rate is a function and the maximum value of the function is not known in advance, then the corresponding synchronizing event also forms a potential set that is a singleton. Regarding the second case, we make the following observation. The position of a synchronizing transition rate in Q is uniquely determined by all synchronizing transition matrices that correspond to the synchronizing event. This can be seen from (1). Hence, we have the following proposition.

PROPOSITION 4. *In a SAN with simple synchronizations, the set $\mathcal{E}^*$ of synchronizing events contribute to the same nonzero element of Q if and only if there exists at least one nonzero element with the same indices in the matrices $Q_{e_j}^{(i)}$ for all $e_j \in \mathcal{E}^*$ and $i = 1, 2, \ldots, N$.*

Proof. The proof follows from (1), the definition of tensor product, and Definitions 2 and 3. □

Those synchronizing events that are not classified as potential sets of singletons must be tested for the condition in Proposition 4. The test of two events, t and u, for the condition requires the comparison of the indices of nonzero elements in $Q_{e_t}^{(i)}$ and $Q_{e_u}^{(i)}$ for $i = 1, 2, \ldots, N$; that is, we test N pairs of matrices. For k events, the number of matrix pairs that need to be tested is $Nk(k-1)/2$. Note that for three events, t, u, and v, the fact that the pairs $(Q_{e_t}^{(i)}, Q_{e_u}^{(i)})$ and $(Q_{e_u}^{(i)}, Q_{e_v}^{(i)})$ each have at least one nonzero element with the same indices for $i = 1, 2, \ldots, N$ does not imply that the events t and v also satisfy the condition. In other words, the condition is not transitive. This further complicates the test for the condition in Proposition 4.

In order to avoid excessive computation associated with the test, we consider the set of synchronizing events $\mathcal{P}$ as a potential set if for all $e_u \in \mathcal{P}$ there exists $e_v \in \mathcal{P}$ such that the condition in Proposition 4 is satisfied for synchronizing events u and v, and the sum of transition rates of synchronizing events in $\mathcal{P}$ is greater than or equal to $\alpha\epsilon$. According to this definition, we form potential sets as follows. Let $\mathcal{L}$ be the set of synchronizing events that are not classified as potential sets of singletons. We choose event $e_v \in \mathcal{L}$, remove it from $\mathcal{L}$, and test e_v with each event in $\mathcal{L}$ for the condition in Proposition 4. Let $\mathcal{K}$ be the set of events that satisfy this condition. Then, if the sum of the transition rates of synchronizing event v and those in $\mathcal{K}$ is greater than or equal to $\alpha\epsilon$, we remove the events that are in $\mathcal{K}$ from $\mathcal{L}$ and form the potential set $\mathcal{P} = \{e_v\} \cup \mathcal{K}$. We repeat this procedure for all events in $\mathcal{L}$ until $\mathcal{L} = \emptyset$.

Example (*continued*). Let $\epsilon = 0.3$, implying $\alpha\epsilon = 4.2$. The transition rate of the master automaton of simple synchronizing event 1 is 5 and greater than $\alpha\epsilon$ (see $Q_{e_1}^{(3)}(2,1)$). Hence, the first potential set, $\mathcal{P}_1$, consists of synchronizing event 1 only. The second synchronizing event of the SAN also forms a potential set. See $Q_{e_2}^{(2)}(1,3)$ for justification. Thus, $\mathcal{P}_1 = \{e_1\}$ and $\mathcal{P}_2 = \{e_2\}$. Now, consider the case in which $\epsilon = 0.4$, implying $\alpha\epsilon = 5.6$. Both transition rates of synchronizing events 1 and 2 are less than $\alpha\epsilon$. Hence, we have to test these two events for the condition in Proposition 4; that is, we check if each of the three pairs of matrices $(Q_{e_1}^{(1)}, Q_{e_2}^{(1)})$, $(Q_{e_1}^{(2)}, Q_{e_2}^{(2)})$, and $(Q_{e_1}^{(3)}, Q_{e_2}^{(3)})$ have at least one nonzero element with the same indices. However, the condition in Proposition 4 is not satisfied. Thus, the number of potential sets for the case of $\epsilon = 0.4$ is zero. This implies that neither of the synchronizing events influence the NCD partitioning of the underlying MC. Therefore, when $\epsilon = 0.4$, synchronizing events of the SAN are omitted from further consideration in Step 3 of Algorithm 1.

4.3. Constructing NCD connected components. As indicated in Remark 2, a nonzero element in the global generator of a SAN originates either from a local transition rate or from one or more synchronizing transition rates. Hence, NCD CCs of the underlying MC are determined by (i) constant local transition rates that are greater than or equal to $\alpha\epsilon$, (ii) functional local transition rates that can take values greater than or equal to $\alpha\epsilon$, or (iii) transition rates of synchronizing events that are in the potential sets $\mathcal{P}_r$, $r = 1, 2, \ldots, N_{\mathcal{P}}$. These three different possibilities are considered in Step 3 of Algorithm 1. The input parameters of Step 3 are local transition rate matrices and synchronizing event matrices, ϵ, α computed in Step 1, and potential sets formed in Step 2. The output of Step 3 is the set of NCD CCs of the underlying MC.

First, we consider possibility (i) in which local transition rates are constant, and assume that $Q = Q_l$ (see (1)). Using $\alpha\epsilon$, we can find the NCD CCs of $Q_l^{(i)}$, $i =$

$1, 2, \ldots, N$. Let $\mathcal{C}^{(i)}$ be the set of NCD CCs of $Q_l^{(i)}$, where a member of $\mathcal{C}^{(i)}$, denoted by $\mathtt{c}^{(i)}$, is a partition of states from $\mathcal{A}^{(i)}$. Let $\mathcal{B}$ and $\mathcal{H}$ be sets in which each member of either set is also a set. In other words, $\mathcal{B}$ as well as $\mathcal{H}$ is a set of sets. We define the binary operator $\odot$ between the two sets $\mathcal{B}$ and $\mathcal{H}$ as $\mathcal{B} \odot \mathcal{H} = \{\mathtt{b} \times \mathtt{h} \mid \mathtt{b} \in \mathcal{B}, \mathtt{h} \in \mathcal{H}\}$, where $\times$ is the ordinary Cartesian product operator. Then, based on the graph interpretation of the tensor sum operator discussed in [13], the set of NCD CCs is given by $\mathcal{C} = \mathcal{C}^{(1)} \odot \mathcal{C}^{(2)} \odot \cdots \odot \mathcal{C}^{(N)}$. Observe that if $\mathcal{C}^{(i)}$, $i = 1, 2, \ldots, N$, are singletons, then $\mathcal{C}$ is a singleton as well; that is, the underlying MC is not NCD for given ϵ. One can take advantage of the same property when there are only K ($< N$) $\mathcal{C}^{(i)}$ that are singletons. In this case, we renumber the automata so that these K sets assume indices from $(N - K + 1)$ to N. Then these K sets can be replaced with the set $\mathcal{C}^{[N-K+1]} = \{\{1, 2, \ldots, n_{N-K+1}\} \times \{1, 2, \ldots, n_{N-K}\} \times \cdots \times \{1, 2, \ldots, n_N\}\}$.

Now we bring into the picture functional local transition rates and consider possibility (ii). Let us assume that the automata of the given SAN can be reordered and renumbered so that transitions of automaton i depend (if at all) on the states of higher indexed automata, but they do not depend on the states of lower indexed automata (see [12] for details). Since Cartesian product is associative, $\odot$ is also associative, and one can rewrite the expression for $\mathcal{C}$ as

$$\mathcal{C} = \left(\mathcal{C}^{(1)} \odot \left(\mathcal{C}^{(2)} \odot \cdots \odot \left(\mathcal{C}^{(N-1)} \odot \mathcal{C}^{(N)}\right) \cdots\right)\right). \tag{5}$$

Given $\mathcal{C}^{[k]} = (\mathcal{C}^{(k)} \odot (\mathcal{C}^{(k+1)} \odot \cdots \odot (\mathcal{C}^{(N-1)} \odot \mathcal{C}^{(N)}) \cdots))$, the union of all members of $\mathcal{C}^{[k]}$ is a set that is equivalent to the product state space of $\mathcal{A}^{(k)}, \mathcal{A}^{(k+1)}, \ldots, \mathcal{A}^{(N)}$. Therefore, taking into account the assumed ordering of automata, functional transition rates of $\mathcal{A}^{(k)}$ can be evaluated and NCD CCs of $\mathcal{C}^{[k]}$ can be updated accordingly. More formally, let $Q_l^{(k)}(s_k, \tilde{s}_k)$ be a functional element, i.e., $Q_l^{(k)}(s_k, \tilde{s}_k) = f$. Then the NCD CCs $\mathtt{c}^{[k]}, \tilde{\mathtt{c}}^{[k]} \in \mathcal{C}^{[k]}$ must be joined if $(s_k, s_{k+1}, \ldots, s_N) \in \mathtt{c}^{[k]}$, $(\tilde{s}_k, s_{k+1}, \ldots, s_N) \in \tilde{\mathtt{c}}^{[k]}$, and $f(s_k, s_{k+1}, \ldots, s_N) \geq \alpha\epsilon$.

Example (*continued*). We illustrate possibilities (i) and (ii) on the SAN description by omitting synchronizing events 1 and 2. Synchronizing events are treated in possibility (iii). We set $\epsilon = 0.3$ implying $\alpha\epsilon = 4.2$ and assume that the automata are ordered as $\mathcal{A}^{(2)}$, $\mathcal{A}^{(3)}$, $\mathcal{A}^{(1)}$. First, we find the NCD CCs of all local transition rate matrices as in possibility (i) by treating functional transition rates as zero. Inspection of local transition rate matrices shows that local transition rates of all automata are less than $\alpha\epsilon$. Hence, we have $\mathcal{C}^{(1)} = \{\{1_1\}, \{2_1\}\}$, $\mathcal{C}^{(2)} = \{\{1_2\}, \{2_2\}, \{3_2\}\}$, and $\mathcal{C}^{(3)} = \{\{1_3\}, \{2_3\}\}$. The subscripts in the states enable us to distinguish between states with identical indices but that belong to different automata. According to (5), we form the NCD CCs of $Q_l^{(3)} \bigoplus Q_l^{(1)}$, i.e., $\mathcal{C}^{(3)} \odot \mathcal{C}^{(1)} = \{\{(1_3, 1_1)\}, \{(1_3, 2_1)\}, \{(2_3, 1_1)\}, \{(2_3, 2_1)\}\}$. Then we continue with possibility (ii). The value of the functional transition rate $Q_l^{(3)}(1, 2)$ ($= f$) depends on the state of $\mathcal{A}^{(1)}$ only. Hence, we can evaluate f when $\mathcal{C}^{(3)} \odot \mathcal{C}^{(1)}$ is formed. The functional transition rate f evaluates to 5, which is larger than $\alpha\epsilon$, when $\mathcal{A}^{(1)}$ is in state 2. Therefore, we join $\{(1_3, 2_1)\}$ and $\{(2_3, 2_1)\}$. Finally, the NCD CCs of Q_l are given by

$$\begin{aligned}\mathcal{C} &= \mathcal{C}^{(2)} \odot (\mathcal{C}^{(3)} \odot \mathcal{C}^{(1)}) = \{\{1_2\}, \{2_2\}, \{3_2\}\} \odot \{\{(1_3, 1_1)\}, \{(1_3, 2_1), (2_3, 2_1)\}, \{(2_3, 1_1)\}\} \\ &= \{\{(1_2, 1_3, 1_1)\}, \{(1_2, 2_3, 1_1)\}, \{(2_2, 1_3, 1_1)\}, \{(2_2, 2_3, 1_1)\}, \{(3_2, 1_3, 1_1)\}, \{(3_2, 2_3, 1_1)\}, \\ &\quad \{(1_2, 1_3, 2_1), (1_2, 2_3, 2_1)\}, \{(2_2, 1_3, 2_1), (2_2, 2_3, 2_1)\}, \{(3_2, 1_3, 2_1), (3_2, 2_3, 2_1)\}\}.\end{aligned}$$

Now we consider possibility (iii). When possibilities (i) and (ii) are handled, the union of all members in $\mathcal{C}$ is a set that corresponds to the global state space of

the SAN. The transition rate of synchronizing event t can be taken into account as follows. Let $(s_1, s_2, \ldots, s_N) \in \mathtt{c}$ and $(\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_N) \in \tilde{\mathtt{c}}$, where $\mathtt{c}, \tilde{\mathtt{c}} \in \mathcal{C}$. Then $\mathtt{c}$ and $\tilde{\mathtt{c}}$ must be joined if $\prod_{i=1}^{N} Q_{e_t}^{(i)}(s_i, \tilde{s}_i) \geq \alpha\epsilon$. Since the global state space of the SAN is usually very large, it may take a significant amount of time to find all pairs $\mathtt{c}$ and $\tilde{\mathtt{c}}$ that satisfy this condition. Fortunately, the situation can be improved. Let p, $1 < p \leq N$, be the smallest index among automata involved in event t, i.e., $Q_{e_t}^{(i)} = I_{n_i}$ for $i = 1, 2, \ldots, p-1$. We rewrite the first two terms of (1) as

$$\bigoplus_{i=1}^{N} Q_l^{(i)} + \sum_{j=1}^{E} \bigotimes_{i=1}^{N} Q_{e_j}^{(i)} = \left(\bigoplus_{i=1}^{p-1} Q_l^{(i)} \right) \bigoplus Q_l^{[p]} + \left(\bigotimes_{i=1}^{p-1} I_{n_i} \right) \bigotimes Q_{e_t}^{[p]} + \sum_{j=1, j \neq t}^{E} \bigotimes_{i=1}^{N} Q_{e_j}^{(i)}, \tag{6}$$

where $Q_l^{[p]} = \bigoplus_{i=p}^{N} Q_l^{(i)}$ and $Q_{e_t}^{[p]} = \bigotimes_{i=p}^{N} Q_{e_t}^{(i)}$. From the definition of tensor sum, the first two terms of expression (6) can be written as

$$\left(\bigoplus_{i=1}^{p-1} Q_l^{(i)} \right) \bigoplus Q_l^{[p]} + \left(\bigotimes_{i=1}^{p-1} I_{n_i} \right) \bigotimes Q_{e_t}^{[p]} = \left(\bigoplus_{i=1}^{p-1} Q_l^{(i)} \right) \bigoplus \left(Q_l^{[p]} + Q_{e_t}^{[p]} \right). \tag{7}$$

From (7), it can be seen that the transition rate of synchronizing event t can be taken into account on the smaller state space $\mathcal{C}^{(p)} \odot \mathcal{C}^{(p+1)} \odot \cdots \odot \mathcal{C}^{(N)}$. The same idea can be extended to the potential sets formed in Step 2. In other words, if for $\mathcal{P}_r$, there exists σ_r, $1 < \sigma_r \leq N$, such that $Q_{e_j}^{(i)} = I_{n_i}$ for $i = 1, 2, \ldots, \sigma_r - 1$ and all $e_j \in \mathcal{P}_r$, then transition rates of synchronizing events in $\mathcal{P}_r$ can be taken into account when the set $\mathcal{C}^{[\sigma_r]} = \mathcal{C}^{(\sigma_r)} \odot \mathcal{C}^{(\sigma_r+1)} \odot \cdots \odot \mathcal{C}^{(N)}$ is formed. We remark that for the assumed ordering of automata, all functional transitions that may be present in synchronizing transition matrices of events in $\mathcal{P}_r$ can be evaluated when $\mathcal{C}^{[\sigma_r]}$ is formed.

Example (*continued*). For $\epsilon = 0.3$, each of the two synchronizing events of the SAN is classified as a potential set. We assume the same ordering of automata, i.e., $\mathcal{A}^{(2)}$, $\mathcal{A}^{(3)}$, $\mathcal{A}^{(1)}$. After renumbering the automata, let the new indices of the automata be $\tilde{1}$, $\tilde{2}$, $\tilde{3}$, respectively. For the given ordering of automata, the smallest index among automata involved in event 1 as well as in event 2 is $\tilde{1}$. Hence, the transition rates of events 1 and 2 can be taken into account when $\mathcal{C}^{[\tilde{1}]} = \mathcal{C}$ is formed. Due to the transition rate of synchronizing event 1, we join the NCD CCs that have the members $(1_2, 2_3, 1_1)$ and $(3_2, 1_3, 1_1)$, $(2_2, 2_3, 1_1)$ and $(1_2, 1_3, 1_1)$, $(3_2, 2_3, 1_1)$ and $(1_2, 1_3, 1_1)$. Similarly, due to synchronizing event 2, we join the NCD CCs that have the members $(1_2, 1_3, 1_1)$ and $(3_2, 1_3, 2_1)$, $(1_2, 1_3, 2_1)$ and $(3_2, 1_3, 1_1)$, $(1_2, 2_3, 1_1)$ and $(3_2, 2_3, 2_1)$, $(1_2, 2_3, 2_1)$ and $(3_2, 2_3, 1_1)$. For justification, see $\mathcal{C}$ formed in the example following possibility (ii) and the SAN description.

When the automata of a SAN have cyclic dependencies, they cannot be ordered as discussed. Such cases can be handled as follows. Let $G(\mathcal{V}, \mathcal{E})$ be the digraph in which v_i corresponds to $\mathcal{A}^{(i)}$ and $(v_i, v_j) \in \mathcal{E}$ if transitions in $\mathcal{A}^{(i)}$ depend on the state of $\mathcal{A}^{(j)}$ (see Definition 4). Let G_{SCC} be the digraph obtained by collapsing each SCC of G to a single vertex. This graph is acyclic and the automata of the SAN can be ordered topologically with respect to G_{SCC}. Assuming that the automata are in this order, let p be the smallest index among cyclically dependent automata. Then we can evaluate all functions in the cyclically dependent automata when $\mathcal{C}^{[p]}$ is formed. The special case in which a cyclic dependency is created by transitions in the synchronizing transition matrices of a particular event can be handled in the same way as discussed in possibility (iii). There, the potential set $\mathcal{P}_r$, $r \in \{1, 2, \ldots, N_{\mathcal{P}}\}$, is taken into account when $\mathcal{C}^{[\sigma_r]}$ is formed. Assuming that the automata are ordered

topologically with respect to G_{SCC}, all functions in the matrices of synchronizing events that belong to $\mathcal{P}_r$ can be evaluated when $\mathcal{C}^{[\sigma_r]}$ is formed.

Our final remark is about a SAN with more that one essential subset of states and/or transient states. For $1 < i \leq N$, we do not have a one-to-one mapping between the global state space and the union of all members in $\mathcal{C}^{[i]}$. Hence, we cannot say whether a member of $\mathtt{c}^{[i]} \in \mathcal{C}^{[i]}$ maps to a state in the essential subset of interest or to an uninteresting state. Therefore, the decomposition of $\mathcal{C}$ as in (5) that allows us to handle functional local transition rates and synchronizing transition rates on a smaller state space cannot be used. This is because one or both of the members that belong to the joined NCD CCs may map to an uninteresting state. For a SAN with uninteresting states, possibilities (ii) and (iii) should be considered on the global state space. Hence, the NCD CCs $\mathtt{c}, \tilde{\mathtt{c}} \in \mathcal{C}$ should be joined only if the members under consideration from each of the two sets map into the essential subset of interest. When we compute $\mathcal{C} = \mathcal{C}^{(1)} \odot \mathcal{C}^{(2)} \odot \cdots \odot \mathcal{C}^{(N)}$, uninteresting states must also be omitted from consideration. From the definition of the binary operator $\odot$, if s_i and $\tilde{s}_i$ are in the same NCD CC of $\mathcal{C}^{(i)}$, then it must be that $(s_1, s_2, \ldots, s_{i-1}, s_i, s_{i+1} \ldots, s_N)$ and $(s_1, s_2, \ldots, s_{i-1}, \tilde{s}_i, s_{i+1} \ldots, s_N)$ are in the same NCD CC of $\mathcal{C}$. When uninteresting states are present, we exercise the additional constraint that $(s_1, s_2, \ldots, s_{i-1}, s_i, s_{i+1} \ldots, s_N)$ and $(s_1, s_2, \ldots, s_{i-1}, \tilde{s}_i, s_{i+1} \ldots, s_N)$ must belong to the essential subset of interest.

In the next subsection, we summarize for Algorithm 1 the detailed space and time complexity analysis that appears in [14] and apply the results to our example.

4.4. Complexity analysis of Algorithm 1. The core operation performed by an algorithm that finds the NCD CCs of a MC is floating-point comparison. Hence, we provide the number of floating-point comparisons performed in Algorithm 1. Regarding the algorithm's storage requirements, we remark that its three steps are executed sequentially. Hence, the maximum amount of memory required by Algorithm 1 is upper bounded by an integer array of length $O(n)$.

For the sake of simplicity, we assume that the MC underlying the SAN is irreducible. In Step 1, the number of floating-point comparisons is given by $\sum_{k=1}^{N_D} \prod_{i, \mathcal{A}^{(i)} \in \mathcal{D}_k} n_i$. For the best case in which each dependency set is a singleton, the number of floating-point comparisons reduces to $\sum_{i=1}^{N} n_i$. On the other hand, if all automata form a single dependency set, we have the upper bound $\prod_{i=1}^{N} n_i = n$. In Step 2, the lower bound on the number of floating-point comparisons is E, and it corresponds to the case in which the transition rate of each simple synchronizing event is greater than or equal to $\alpha\epsilon$. The upper bound is equal to $\frac{1}{2}E(E+1)$ floating-point comparisons. This number of floating-point comparisons is achieved when the transition rate of each simple synchronizing event is less than $\alpha\epsilon$ and the transition rates of synchronizing events do not sum up in Q. The number of floating-point comparisons in Step 3 depends strongly on the number of functional transitions and synchronizing events as well as the automata ordering. Assuming that in Step 2 of Algorithm 1 synchronizing event r is classified as the potential set $\mathcal{P}_r$, $r = 1, 2, \ldots, E$, and the automata are ordered as discussed in possibility (ii) in subsection 4.3, the number of floating-point comparisons in Step 3 is given by

$$\sum_{i=1}^{N} nz_l^{(i)} + \sum_{i=1}^{N-1} nf_i \prod_{j=i+1}^{N} n_j + \sum_{r=1}^{E} \prod_{j=\sigma_r, j \neq m_r}^{N} nz_{e_r}^{(j)},$$

where $nz_l^{(i)}$ is the number of nonzero off-diagonal elements in $Q_l^{(i)}$, nf_i is the number

of functional transitions in $Q_l^{(i)}$, $nz_{e_r}^{(j)}$ is the number of nonzeros in $Q_{e_r}^{(i)}$, and m_r is the index of the master automaton of event r. Finally, the number of floating-point comparisons performed in Algorithm 1 is given by

$$E + \sum_{i=1}^{N}(n_i + nz_l^{(i)}) + \sum_{i=1}^{N-1} nf_i \prod_{j=i+1}^{N} n_j + \sum_{r=1}^{E} \prod_{j=\sigma_r, j\neq m_r}^{N} nz_{e_r}^{(j)}$$

in the best case, and

$$n + \frac{1}{2}E(E+1) + \sum_{i=1}^{N} nz_l^{(i)} + \sum_{i=1}^{N-1} nf_i \prod_{j=i+1}^{N} n_j + \sum_{r=1}^{E} \prod_{j=\sigma_r, j\neq m_r}^{N} nz_{e_r}^{(j)}$$

in the worst case.

Step 3 of Algorithm 1 also incurs floating-point multiplications when synchronizing events are handled. Computation of a single nonzero transition originating from synchronizing event r requires $(N - \sigma_r)$ floating-point multiplications. For synchronizing event r, we compute $\prod_{j=\sigma_r, j\neq m_r}^{N} nz_{e_r}^{(j)}$ elements. Hence, the maximum number of floating-point multiplications in Step 3 is $\sum_{r=1}^{E}[(N - \sigma_r)\prod_{j=\sigma_r, j\neq m_r}^{N} nz_{e_r}^{(j)}]$. Observe that this expression is almost the same as the last term of the expression for the number of floating-point comparisons performed in Algorithm 1. Hence, assuming that the time it takes to perform floating-point multiplication and floating-point comparison are of the same order, the time complexity of Algorithm 1 is roughly the number of floating-point comparisons.

Example (*continued*). We calculate the number of floating-point comparisons performed by Algorithm 1 to find an NCD partitioning of the MC underlying the SAN. We use the same input parameters for Algorithm 1 as in subsection 4.3; that is, $\epsilon = 0.3$ and the automata are ordered as $\mathcal{A}^{(2)}$, $\mathcal{A}^{(3)}$, $\mathcal{A}^{(1)}$. Following the three steps of Algorithm 1 on our example, we see that Step 1 takes $n_1 n_3 + n_2 = 7$ floating-point comparisons to find the maximums of 2 dependency sets, and Step 2 takes 2 floating-point comparisons to form the 2 potential sets of singletons. Step 3 takes 7+2+3+4=16 floating-point comparisons, where 7 is the number of comparisons to find $\mathcal{C}^{(1)}$, $\mathcal{C}^{(2)}$, $\mathcal{C}^{(3)}$; 2 is the number of comparisons to handle the functional local transition of $\mathcal{A}^{(3)}$; and 3 and 4 are the numbers of comparisons to process transition rates of synchronizing events 1 and 2, respectively. Thus, the total number of floating comparisons performed in Algorithm 1 is 25. The number of floating-point multiplications performed to process synchronizing events 1 and 2 is $(N-1)(nz_{e_1}^{(1)} nz_{e_1}^{(2)} + nz_{e_2}^{(1)} nz_{e_2}^{(3)}) = 14$. When the global generator is stored in sparse format, the total number of floating-point comparisons performed by the straightforward algorithm that finds NCD CCs of Q is 57, which is almost two times as large as the corresponding value of Algorithm 1.

5. Numerical results. We implemented the SC algorithm and Algorithm 1 in C++ as part of the software package PEPS [18]. We ran all the experiments on a Sun UltraSparcstation 10 with 128 MBytes of RAM. To verify the NCD partitionings obtained for a given SAN, we compared our results with the straightforward approach of generating in core the submatrix of Q corresponding to the essential subset of states obtained using the SC algorithm and finding its NCD CCs. We remark that the same data structure for NCD CCs is used in Algorithm 1 and the straightforward approach.

The input parameters of Algorithm 1 are the user specified decomposability parameter ϵ, the vector output by the SC algorithm in which states corresponding to the

essential subset of interest are marked, and a file in PEPS format that contains the description of the SAN under consideration and the dependencies among automata. We remark that the only modification that we make on the SAN description is the transformation of each synchronizing event to the simple form (if the SAN is not already in that form). Note that this transformation is taken into account in the reported results.

As test problems, we use the three SAN models that appear in [24]. We name them resource sharing, three queues, and mass storage. Here, we present the results of experiments with the three queues problem. The results of experiments with the other two problems appear in [14]. The SAN model of the three queues problem consists of four automata $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \mathcal{A}^{(3_1)}, \mathcal{A}^{(3_2)}$ with, respectively, C_1, C_2, C_3, C_3 states and two synchronizing events. The state space size is given by $n = C_1 C_2 C_3^2$ and there is a single subset of $C_1 C_2 C_3 (C_3 + 1)/2$ essential states. Functional transition rates appear in local transition rate and synchronizing event matrices. There are two dependency sets $\mathcal{D}_1 = \{\mathcal{A}^{(1)}, \mathcal{A}^{(3_1)}, \mathcal{A}^{(3_2)}\}$ and $\mathcal{D}_2 = \{\mathcal{A}^{(2)}\}$. Detailed description of the three queues problem and its parameters can be found in [12]. In our experiments, we use the values of real parameters in [24].

Results of experiments for the three queues problem are presented in Table 1. All timing results are in seconds. In Table 1, n denotes the number of states in the global state space of the particular SAN under consideration, n_{ess} denotes the number of states in the essential subset, nz_{ess} denotes the number of nonzero elements in the submatrix of Q corresponding to the essential subset of states, and SC denotes the time for state classification. For each problem, we indicate in parentheses under n the values of the integer parameters used. The column ϵ denotes the value of the decomposability parameter used and $|CCs|$ denotes the number of NCD CCs corresponding to ϵ when transient states are removed. The column NCD_S contains timing results for Algorithm 1. The columns Gen. and NCD_N, respectively, contain timing results to generate in core the submatrix of Q corresponding to the essential subset of states and to naively compute its NCD partitioning for given ϵ after the SC algorithm is executed. We have varied the value of ϵ in each problem to see how the performance of Algorithm 1 changes for different number of NCD CCs.

We remark that the difference between the time required to generate in core the submatrix of Q corresponding to the essential subset of states for a given SAN and the time to find the corresponding NCD partitionings using Algorithm 1 is noticeable. Compare columns Gen. and NCD_S, and also compare the sum of columns Gen. and NCD_N with column NCD_S. Moreover, there are cases for which it is not possible to generate in core the submatrix of Q corresponding to the essential subset of states on the particular architecture. Hence, the straightforward approach of finding NCD partitionings is relatively more restricted with memory and is slower than using Algorithm 1.

The time spent for state classification does not involve any floating-point operations, whereas the time spent to generate in core the submatrix of Q corresponding to the essential subset of states primarily involves floating-point arithmetic operations. The overhead associated with evaluating functions slows down both tasks dramatically. Compare columns SC and Gen. with columns NCD_S and NCD_N. The time spent by the SC algorithm is larger than the time spent by Algorithm 1 in all experiments. This is not surprising since the former is based on finding SCCs while the latter is based on finding CCs. The difference is more pronounced when there are multiple dependency sets for which Algorithm 1 can bring in considerable savings.

TABLE 1
Results of the three queues problem (C_1, C_2, C_3).

n	n_{ess}	nz_{ess}	SC	ϵ	$\lvert CCs\rvert$	NCD_S	Gen.	NCD_N
68,850	36,720	207,279	0.82	0.10	1	0.10	0.39	0.05
(18,17,15)				0.22	544	0.22		0.09
				0.25	4,590	0.13		0.07
				0.35	36,720	0.11		0.06
202,400	106,260	608,474	2.63	0.10	1	0.30	1.24	0.17
(23,22,20)				0.22	924	0.68		0.28
				0.25	10,120	0.38		0.24
				0.35	106,260	0.33		0.23
756,000	390,600	2,264,460	9.83	0.10	1	1.03	4.58	0.62
(30,28,30)				0.22	1,652	2.46		1.04
				0.25	25,200	1.37		0.92
				0.35	390,600	1.12		0.90
1,414,875	727,650	4,239,795	19.04	0.10	1	1.88	8.37	1.16
(35,33,35)				0.22	2,277	4.60		1.94
				0.25	40,425	2.46		1.71
				0.35	727,650	2.02		1.56
6,875,000	3,506,250	20,632,250	96.37	0.10	1	8.63		
(50,55,50)				0.22	5,445	21.85		
				0.25	137,500	11.57		
				0.35	3,506,250	9.18		
9,150,625	4,658,500	27,445,825	131.34	0.10	1	11.25		
(55,55,55)				0.22	5,995	33.04		
				0.25	166,375	14.24		
				0.35	4,658,500	12.44		

The case of $|CCs| = 1$ corresponds to smaller ϵ and implies the largest number of nonzeros taken into account from automata matrices in Algorithm 1 and from the submatrix of Q corresponding to the essential subset of states in the naive NCD partitioning algorithm. The case of $|CCs| = n_{\text{ess}}$ corresponds to larger ϵ and implies larger temporary data structures being used by both algorithms when determining NCD CCs. Hence, for increasing ϵ, the results in columns NCD_S and NCD_N either increase then decrease.

6. Conclusion. In this work, we have considered the application of the near complete decomposability concept to SANs. The definitions, propositions, and remarks presented in sections 3 and 4 have enabled us to devise an efficient algorithm that computes NCD partitionings of the MC underlying a SAN. The approach is based on determining the NCD connected components of a SAN from the description of individual automata without generating the global transition rate matrix. We have also implemented a state classification algorithm for SANs that classifies each state in the global state space as essential or transient. The output of the state classification algorithm is used in the NCD partitioning algorithm for SANs. The time and space complexities of the NCD partitioning algorithm depend on the number of automata, the number of synchronizing events, the number of functions, the number of essential states of interest, the sparsity of automata matrices, the dependency sets, and the ordering of automata. Future work should focus on taking advantage of the partitionings computed by the devised algorithms in two-level iterative solvers.

Acknowledgments. We thank the anonymous referees for their constructive remarks, which led to an improved manuscript.

REFERENCES

[1] P. Buchholz, *An aggregation/disaggregation algorithm for stochastic automata networks*, Probab. Engrg. Inform. Sci., 11 (1997), pp. 229–253.

[2] P. Buchholz, *An adaptive aggregation/disaggregation algorithm for hierarchical Markovian models*, European J. Oper. Res., 116 (1999), pp. 545–564.

[3] P. Buchholz, *Projection methods for the analysis of stochastic automata networks*, in Proceedings of the 3rd International Workshop on the Numerical Solution of Markov Chains, B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 149–168.

[4] P. Buchholz, G. Ciardo, S. Donatelli, and P. Kemper, *Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models*, INFORMS J. Comput., 12 (2000), pp. 203–222.

[5] P. Buchholz, M. Fischer, and P. Kemper, *Distributed steady state analysis using Kronecker algebra*, in Proceedings of the 3rd International Workshop on the Numerical Solution of Markov Chains, B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 76–95.

[6] R. H. Chan and W. K. Ching, *Circulant preconditioners for stochastic automata networks*, Numer. Math., 87 (2000), pp. 35–57.

[7] M. Davio, *Kronecker products and shuffle algebra*, IEEE Trans. Comput., C-30 (1981), pp. 116–125.

[8] T. Dayar, *Permuting Markov Chains to Nearly Completely Decomposable Form*, Technical Report BU-CEIS-9808, Department of Computer Engineering and Information Science, Bilkent University, Ankara, Turkey, 1998; also available online from ftp://ftp.cs.bilkent.edu.tr/pub/tech-reports/1998/BU-CEIS-9808.ps.z.

[9] T. Dayar, O. I. Pentakalos, and A. B. Stephens, *Analytical Modeling of Robotic Tape Libraries Using Stochastic Automata*, Technical Report TR-97-189, CESDIS, NASA/GSFC, Greenbelt, MD, 1997.

[10] T. Dayar and W. J. Stewart, *On the effects of using the Grassmann–Taksar–Heyman method in iterative aggregation–disaggregation*, SIAM J. Sci. Comput., 17 (1996), pp. 287–303.

[11] 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.

[12] P. Fernandes, B. Plateau, and W. J. Stewart, *Efficient descriptor-vector multiplications in stochastic automata networks*, J. ACM, 45 (1998), pp. 381–414.

[13] J.-M. Fourneau and F. Quessette, *Graphs and stochastic automata networks*, in Computations with Markov Chains: Proceedings of the 2nd International Workshop on the Numerical Solution of Markov Chains, W. J. Stewart, ed., Kluwer, Boston, MA, 1995, pp. 217–235.

[14] O. Gusak, T. Dayar, and J.-M. Fourneau, *Stochastic Automata Networks and Near Complete Decomposability*, Technical Report BU-CE-0016, Department of Computer Engineering, Bilkent University, Ankara, Turkey, 2000; also available online from ftp://ftp.cs.bilkent.edu.tr/pub/tech-reports/2000/BU-CE-0016.ps.z.

[15] C. D. Meyer, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.

[16] B. Plateau, *On the stochastic structure of parallelism and synchronization models for distributed algorithms*, in Proceedings of the SIGMETRICS Conference on Measurement and Modelling of Computer Systems, Austin, TX, 1985, pp. 147–154.

[17] B. Plateau and K. Atif, *Stochastic automata network for modeling parallel systems*, IEEE Trans. Software Engrg., 17 (1991), pp. 1093–1108.

[18] B. Plateau, J.-M. Fourneau, and K.-H. Lee, *PEPS: A package for solving complex Markov models of parallel systems*, in Modeling Techniques and Tools for Computer Performance Evaluation, R. Puigjaner and D. Ptier, eds., Palma de Mallorca, Spain, 1988, pp. 291–305.

[19] B. Plateau and J.-M. Fourneau, *A methodology for solving Markov models of parallel systems*, J. Parallel Distrib. Comput., 12 (1991), pp. 370–387.

[20] G. W. Stewart, W. J. Stewart, and D. F. McAllister, *A two-stage iteration for solving nearly completely decomposable Markov chains*, in Recent Advances in Iterative Methods, IMA Vol. Math. Appl. 60, G. H. Golub, A. Greenbaum, and M. Luskin, eds., Springer-Verlag, New York, 1994, pp. 201–216.

[21] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

[22] W. J. Stewart, K. Atif, and B. Plateau, *The numerical solution of stochastic automata networks*, European J. Oper. Res., 86 (1995), pp. 503–525.

[23] W. J. Stewart and W. Wu, *Numerical experiments with iteration and aggregation for Markov*

chains, ORSA J. Comput., 4 (1992), pp. 336–350.
[24] E. Uysal and T. Dayar, *Iterative methods based on splittings for stochastic automata networks*, European J. Oper. Res., 110 (1998), pp. 166–186.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 601–614
© 2001 Society for Industrial and Applied Mathematics

A KRYLOV–SCHUR ALGORITHM FOR LARGE EIGENPROBLEMS*

G. W. STEWART†

Abstract. Sorensen's implicitly restarted Arnoldi algorithm is one of the most successful and flexible methods for finding a few eigenpairs of a large matrix. However, the need to preserve the structure of the Arnoldi decomposition on which the algorithm is based restricts the range of transformations that can be performed on the decomposition. In consequence, it is difficult to deflate converged Ritz vectors from the decomposition. Moreover, the potential forward instability of the implicit QR algorithm can cause unwanted Ritz vectors to persist in the computation. In this paper we introduce a general Krylov decomposition that solves both problems in a natural and efficient manner.

Key words. large eigenproblem, Krylov sequence, Arnoldi algorithm, Krylov decomposition, restarting, deflation

AMS subject classifications. 15A18, 65F15, 65F60

PII. S0895479800371529

1. Introduction and background.

In this paper we are going to describe an alternative to the Arnoldi method that resolves some difficulties with its implicitly restarted version. To understand the difficulties and their solution requires a detailed knowledge of the Arnoldi process. We therefore begin with a survey, which will also serve to set the notation for this paper.

Let A be a matrix of order n and let u_1 be a vector of 2-norm one. Let $u_1, u_2, u_3, \ldots$ be the result of sequentially orthogonalizing the Krylov sequence $u_1, Au_1, A^2u_1, \ldots$. In 1950, Lanczos [6] showed that if A is Hermitian then the vectors u_i satisfy a three-term recurrence of the form

$$\beta_k u_{k+1} = A_k u_k - \alpha_k u_k - \beta_{k-1} u_{k-1}, \tag{1.1}$$

a recursion that in principle allows the economical computation of the u_j.

There is an elegant representation of this recursion in matrix terms. Let

$$U_k = (u_1 \;\; u_2 \;\; \cdots \;\; u_k)$$

be the matrix formed from the Lanczos vectors u_j. Then there is a tridiagonal matrix T formed from the α's and β's in (1.1) such that

$$AU_k = U_k T_k + \beta_k u_{k+1} \mathbf{e}_k^{\mathrm{T}}, \tag{1.2}$$

where $\mathbf{e}_k$ is the vector whose last component is one and whose other components are zero. From the orthogonality of the u_j, it follows that T_k is the Rayleigh quotient

$$T_k = U_k^{\mathrm{H}} A U_k.$$

We will call (1.2) a Lanczos decomposition.

*Received by the editors May 2, 2000; accepted for publication (in revised form) by J. Varah June 8, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/simax/23-3/37152.html

†Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (stewart@cs.umd.edu). Work supported by the National Science Foundation under grant 970909-8562.

Lanczos appreciated the fact that even for comparatively small k the matrix T_k could contain accurate approximations to the eigenvalues of A. When this happens, the column space $\mathcal{U}_k$ of U_k will usually contain approximations to the corresponding eigenvectors. Such an approximation—call it z—can be calculated by computing a suitable eigenpair (μ, w) of T_k and setting $z = U_k w$. This process is called the Rayleigh–Ritz method; μ is called a Ritz value and z a Ritz vector.

In 1951, Arnoldi [1], building on Lanczos's work, showed that if A is non-Hermitian, then the Lanczos decomposition becomes

$$AU_k = U_k H_k + \beta_k u_{k+1}\mathbf{e}_k^{\mathrm{T}}, \tag{1.3}$$

where H_k is upper Hessenberg. We will call (1.3) an Arnoldi decomposition. Once again, H_k may contain accurate approximations to the eigenvalues of A, especially those on the periphery of the spectrum of A. Moreover, approximations to the eigenvectors may be obtained by the natural generalization of the Rayleigh–Ritz process.

Arnoldi decompositions are essentially unique. Specifically, if H_k is unreduced—that is, if its subdiagonal elements are nonzero—then up to scaling of the columns of U_{k+1} and the rows and columns of H_k, the decomposition is uniquely determined by the space spanned by U_{k+1}.[1] In particular, the Krylov subspace of an unreduced Arnoldi decomposition has a unique starting vector.

Since H_k is not tridiagonal, the Arnoldi vectors do not satisfy a three-term recurrence. To compute u_{k+1} all the columns of U_k must be readily available. If n is large, these vectors will soon consume all available storage, and the process must be restarted. The problem then becomes how to choose a new u_1 that does not discard the information about the eigenvectors contained in $\mathcal{U}_k$. There have been several proposals, whose drawbacks have been nicely surveyed by Morgan [11].

In 1992, Sorensen [14] suggested an elegant way to use the QR algorithm to restart the Arnoldi process. Specifically, suppose we have an Arnoldi decomposition

$$AU_m = U_m H_m + \beta_m u_{m+1}\mathbf{e}_m^{\mathrm{T}} \tag{1.4}$$

of order m that cannot be further expanded because of lack of storage. For some fixed k, choose $m-k$ shifts $\kappa_1, \ldots, \kappa_{m-k}$ and use them to perform $m-k$ steps of the implicitly shifted QR algorithm on the Rayleigh quotient H_m. The effect is to generate an orthogonal matrix Q such that $Q^{\mathrm{H}} H_m Q$ is upper Hessenberg. Then from (1.4)

$$A(U_m Q) = (U_m Q) Q^{\mathrm{H}} H_m Q + \beta_m u_{m+1}\mathbf{e}_m^{\mathrm{T}} Q$$

or

$$A\tilde{U}_m = \tilde{U}_m \tilde{H}_m + u_{m+1} c^{\mathrm{H}}.$$

Sorensen then observed that the structure of Q is such that the first $k-1$ components of c are zero. Consequently, if we let $\tilde{H}_k$ be the leading principal submatrix of $\tilde{H}_m$ of order k and set

$$\beta_k \tilde{u}_{k+1} = \bar{\gamma}_k u_{m+1} + \tilde{h}_{k+1,k} u_{k+1}, \tag{1.5}$$

then

$$A\tilde{U}_k = \tilde{U}_k \tilde{H}_k + \tilde{u}_{k+1} e_k^{\mathrm{T}}$$

[1] This fact is a direct consequence of the implicit Q theorem, which says that if $H = Q^{\mathrm{H}} A Q$ is an unreduced Hessenberg matrix then Q is determined by its first or last column. See [4, Theorem 7.4.2].

is an Arnoldi decomposition of order k. This process of truncating the decomposition is called implicit restarting.

A second key observation of Sorensen suggests a rationale for choosing the shifts. Specifically, if $p(t) = (t - \kappa_1 I) \cdots (t - \kappa_{m-k} I)$, then

$$\tilde{u}_1 = \frac{p(A)u_1}{\|p(A)u_1\|}.$$

It follows that if we choose the shifts to lie in the part of the spectrum that we are not interested in then the implicit restart process deemphasizes these very eigenvalues.

Each iteration of Sorensen's algorithm consists of two stages: an expansion stage, in which the decomposition is expanded until it is inconvenient to go further, and a contraction or purging stage, in which unwanted parts of the spectrum are suppressed. The contraction phase has two variants. In the exact variant, the shifts are taken to be unwanted eigenvalues of H_m. If, for example, we were concerned with stability, we might choose to retain only the eigenvalues with largest real parts. In the other, more general variant, the shifts are not necessarily eigenvalues of H_m. For example, they might be the zeros of a Chebyshev polynomial spanning an ellipse containing unwanted eigenvalues.

The implicitly restarted Arnoldi algorithm has been remarkably successful and has been implemented in the widely used ARPACK package [9]. However, the method has two important drawbacks.

First, for the exact restart procedure to be effective the unwanted Ritz values μ must be moved to the end of H_m, so that the Rayleigh quotient has the form illustrated below for $k = 3$ and $m = 6$:

$$\begin{pmatrix} h & h & h & h & h & h \\ h & h & h & h & h & h \\ 0 & h & h & h & h & h \\ 0 & 0 & 0 & \mu & h & h \\ 0 & 0 & 0 & 0 & \mu & h \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}. \tag{1.6}$$

If H_m is unreduced—that is, if the elements of its first subdiagonal are nonzero—then mathematically H_m must have the form (1.6). In the presence of rounding error, however, the process can fail (for a treatment of this phenomenon, see [17]). This has lead Lehoucq and Sorensen to propose an elaborate method for permanently ridding the decomposition of persistent unwanted Ritz values [8].

The second problem is to move converged Ritz values μ to the beginning of H_k, so that it assumes the form illustrated below:

$$\begin{pmatrix} \mu & h & h & h & h & h \\ 0 & \mu & h & h & h & h \\ 0 & 0 & h & h & h & h \\ 0 & 0 & h & h & h & h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \end{pmatrix}.$$

When the converged Ritz values are thus deflated (or locked), one does not have to update the Arnoldi u_1 and u_2 in the Arnoldi decomposition. Lehoucq and Sorensen have proposed a complicated deflation algorithm.

Most of the complications in the purging and deflating algorithms come from the need to preserve the structure of the Arnoldi decomposition (1.3)—in particular, to preserve the Hessenberg form of the Rayleigh quotient and the zero structure of the vector $\mathbf{e}_k$. The purpose of this paper is to show that if we relax the definition of an Arnoldi decomposition, we can solve the purging and deflating problems in a natural and efficient way. Since the method is centered about the Schur decomposition of the Rayleigh quotient, we will call the method the Krylov–Schur method.

The decompositions and algorithms proposed in this paper are not without precursors. Fokkema, Sleijpen, and van der Vorst [3] explicitly use Schur vectors to restart the Jacobi–Davidson algorithm. Stathopoulos, Saad, and Wu [15] point out that because the unprecondition Jacobi–Davidson algorithm is equivalent to the Arnoldi algorithm, one can also use Schur vectors to restart the latter. Lehoucq [7] has used Schur vectors in the deflation process in [8]. Closer to home, for symmetric matrices Wu and Simon [18] exhibit what might be called a Krylov-spectral decomposition, a special case of our Krylov–Schur decomposition to be introduced later. Finally, Morgan [12] has applied an orthogonal Krylov decomposition to the problem of restarting GMRES. What distinguishes our approach is the explicit introduction of general Krylov decompositions whose subspaces are invariant under certain formal operations—operations that can be used to derive and analyze new algorithms.

In the next section we introduce Krylov decompositions and, in particular, the Krylov–Schur decomposition, which lies at the heart of our method. Section 3 treats the Krylov–Schur method and its relation to the implicitly restarted Arnoldi method. In section 4 we treat the numerical stability of the combined steps. In section 5 we show how to deflate vectors and subspaces from a Krylov decomposition. In section 6 we compare the work done by the implicitly restarted Arnoldi and the Krylov–Schur methods. We end with some general comments. Throughout this paper $\|\cdot\|$ will denote the vector and matrix 2-norm, and $\|\cdot\|_{\mathrm{F}}$ will denote the Frobenius norm (see [16, section 1.4.1]).

2. Krylov decompositions. The structure of an Arnoldi decomposition restricts the operations we can perform on its Rayleigh quotient. The following definition introduces a less constraining decomposition.

DEFINITION 2.1. *A* Krylov decomposition of order k *is a relation of the form*

$$AU_k = U_k B_k + u_{k+1} b_{k+1}^{\mathrm{H}}, \tag{2.1}$$

where B_k is of order k and the columns of $(U_k \;\; u_{k+1})$ are independent. The columns of $(U_k \;\; u_{k+1})$ are called the basis *for the decomposition, and they span the* space *of the decomposition. If the basis is orthonormal, we say the decomposition is orthonormal. The matrix B_k is called the* Rayleigh quotient *of the decomposition.*

This definition removes practically all the restrictions imposed on an Arnoldi decomposition. The vectors of the decomposition are not required to be orthonormal and the vector b_{k+1} and the matrix B_k are allowed to be arbitrary. Nonetheless, we shall see that the relation (2.1) is sufficient to insure that $(U_k \;\; u_{k+1})$ is a basis for a Krylov subspace.

The name "Rayleigh quotient" is appropriate for the matrix B_k. For if $(V_k \;\; v_{k+1})^{\mathrm{H}}$ is a left inverse of $(U_k \;\; u_{k+1})$, then $B_k = V_k^{\mathrm{H}} A U_k$. In particular, if $(\mu, U_k w)$ is an eigenpair of A, then (μ, w) is an eigenpair of B_k. Thus the Rayleigh–Ritz procedure extends to Krylov decompositions.

The subspaces of Krylov decompositions are closed under two classes of transformations: translation and similarity. The first allows us to change the vector u_k. The

second allows us to change the pair (B_k, U_k) along with b_{k+1}^{H}. In what follows we will drop subscripts in k and write our Krylov decomposition in the form $AU = UB + ub^{\mathrm{H}}$.

To introduce the operation of translation, let

$$\gamma\tilde{u} = u - Ug,$$

where $\gamma \neq 0$. Then it is easily verified that

$$AU = U(B + gb^{\mathrm{H}}) + \tilde{u}b^{\mathrm{H}},$$

where $\tilde{b}^{\mathrm{H}} = \gamma b^{\mathrm{H}}$, is a Krylov decomposition with the same space as the original. This gives us considerable freedom to replace u by linear combinations of u and U, although the fact that $\gamma \neq 0$ implies that the vector $\tilde{u}$ always contains some component along u. In particular, we can choose $\tilde{u}$ so that $\|\tilde{u}\| = 1$ and $U^{\mathrm{H}}\tilde{u} = 0$.

To introduce similarity transformations, let W be nonsingular. Then

$$A(UW^{-1}) = (UW^{-1})(WBW^{-1}) + u(b^{\mathrm{H}}W^{-1}) \equiv A\tilde{U} = \tilde{U}\tilde{B} + u\tilde{b}^{\mathrm{H}}$$

is a Krylov decomposition whose space is the same as the original. Because the Rayleigh quotient of the new decomposition is similar to that of the old, we say that the two decompositions are similar.[2]

We will say that two Krylov decompositions related by a sequence of translations and similarities are *equivalent*. We are now going to show that any Krylov decomposition is equivalent to an Arnoldi decomposition. Since the space of an Arnoldi decomposition is a (possibly restarted) Krylov subspace, the result justifies the name Krylov decomposition.

THEOREM 2.2. *Let*

$$AU = UB + ub^{\mathrm{T}} \tag{2.2}$$

be a Krylov decomposition of order k. Then (2.2) *is equivalent to an Arnoldi decomposition. If the Hessenberg part of the Arnoldi decomposition is unreduced, the Arnoldi decomposition is essentially unique.*

Proof. The reduction, which is constructive, proceeds in four stages.

1. By a similarity transformation, orthogonalize the columns of U.
2. By a translation, transform u so that it is of norm one and is orthogonal to $\mathcal{R}(U)$.
3. By a unitary similarity transformation, reduce b to a multiple of $\mathbf{e}_k$.
4. Finally, by a unitary similarity reduce B to Hessenberg form. The reduction is performed rowwise by Householder transformations beginning with the last row, as illustrated in the following Wilkinson diagram:

$$\begin{pmatrix} b & b & b & b & b \\ b & b & b & b & b \\ b^3 & b & b & b & b \\ b^2 & b^2 & b & b & b \\ b^1 & b^1 & b^1 & b & b \end{pmatrix}.$$

[2] A referee has pointed out that these two types of transformations can be combined. Specifically, we say that the Arnoldi decompositions $AU = UB + ub^{\mathrm{T}}$ and $AV = VB + vb^{\mathrm{T}}$ are equivalent if there is a nonsingular matrix $\hat{W} = \left(\begin{smallmatrix} W & g \\ 0 & \gamma \end{smallmatrix}\right)$ such that $(V\ v) = (U\ u)\hat{W}$. With $W = I$ we obtain a translation; with $g = 0$ and $\gamma = 1$ we obtain a similarity.

The final reduction to Hessenberg form does not introduce nonzero elements into the first $k-1$ components of b, so that the result of this algorithm is an Arnoldi decomposition. The uniqueness in the unreduced case follows from the uniqueness of unreduced Arnoldi decompositions. ∎

The proof of Theorem 2.2 illustrates the power of translations and similarities to bring a Krylov decomposition into a useful form without losing the Krylov subspace property. In particular, any Krylov decomposition corresponds to an *orthonormal Krylov decomposition* in which the columns of the basis are orthonormal. (From here on, all our Krylov decompositions will be orthonormal.) Further, we can reduce the Rayleigh quotient to Schur form. The resulting *Krylov–Schur* decomposition is the basis of the main algorithm in this paper, to which we now turn.

3. The Krylov–Schur method. A step of the Krylov–Schur method begins and ends with a Krylov–Schur decomposition of the form

$$AU_k = U_k S_k + u_{k+1} b_{k+1}^{\mathrm{H}},$$

where the letter S (for Schur) stresses the triangularity of the Rayleigh quotient. It will be more convenient to work with the equivalent factored form

$$AU_k = U_{k+1} \hat{S}_k,$$

where

$$\hat{S}_k = \begin{pmatrix} S_k \\ b_{k+1}^{\mathrm{H}} \end{pmatrix}.$$

Like the implicitly restarted Arnoldi method the Krylov–Schur method consists of an expansion phase, in which the underlying Krylov sequence is extended, and a contraction phase, in which the unwanted Ritz values are purged from the decomposition. We will treat each in turn.

The expansion proceeds as in the usual Arnoldi algorithm: the vector Au_{k+1} is orthogonalized against U_{k+1} and normalized to give u_{k+2}, after which S_{k+1} is formed from S_k. The following pseudocode implements this procedure. We assume that U_{k+1} and $\hat{S}_k$ are contained in arrays U and S.

$$\begin{aligned} &1.\quad v = A * U[:, k+1], \\ &2.\quad w = U^{\mathrm{H}} * v, \\ &3.\quad v = v - U * w, \\ &4.\quad \nu = \|v\|_2, \\ &5.\quad U = (U \;\; v/\nu), \\ &6.\quad \hat{S} = \begin{pmatrix} \hat{S} & w \\ 0 & \nu \end{pmatrix}. \end{aligned} \tag{3.1}$$

Note that in a working implementation we would have to reorthogonalize to insure that the vector v is orthogonal to the column space of U to working accuracy (see [16, Algorithm 4.1.13]).

After this process the array $\hat{S}$ has the form illustrated below for $k = 3$:

$$\begin{pmatrix} s & s & s & h \\ 0 & s & s & h \\ 0 & 0 & s & h \\ b & b & b & h \\ 0 & 0 & 0 & h \end{pmatrix}.$$

Here the s's stand for the elements of the original S_k and the b's for the elements of b_{k+1}. The process may be repeated. After $m-k$ steps, the array S has the form illustrated below for $k = 3$ and $m = 6$:

$$\begin{pmatrix} s & s & s & h & h & h \\ 0 & s & s & h & h & h \\ 0 & 0 & s & h & h & h \\ b & b & b & h & h & h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \\ 0 & 0 & 0 & 0 & 0 & h \end{pmatrix}. \tag{3.2}$$

At this point the Rayleigh quotient, which resides in $S[1{:}m, 1{:}m]$, is reduced to Schur form to give the Arnoldi–Schur decomposition

$$AU_m = U_m S_m + u_{m+1} b_{m+1}^{\mathrm{H}}. \tag{3.3}$$

This reduction to Schur form begins with a reduction of the Rayleigh quotient to Hessenberg form, and some minor savings can be obtained at this stage by taking advantage of the structure illustrated in (3.2). Although (3.3) suggests that we are computing the entire decomposition, including U_m, in fact it will be more efficient to defer the computation of the columns of U_m until later. We will return to this point in section 6.

We now turn to the problem of purging the unwanted Ritz values from the Krylov–Schur decomposition (3.3)—the contraction phase of the method. The key is the observation that a Krylov–Schur decomposition can be truncated at any point. Specifically, if we partition a Krylov–Schur decomposition in the form

$$A(U_1 \ U_2) = (U_1 \ U_2) \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} + u(b_1^{\mathrm{H}} \ b_2^{\mathrm{H}}), \tag{3.4}$$

then

$$AU_1 = U_1 S_{11} + u b_1^{\mathrm{H}}$$

is also a Krylov–Schur decomposition. Thus the purging problem can be solved by moving the unwanted Ritz values into the southeast corner of the Rayleigh quotient and truncating the decomposition.

The process of using unitary similarities to move eigenvalues around in a Schur form has been well studied. The current front-running algorithm [2], which has been implemented in the LAPACK routine `xTREXC`, is quite reliable—far more so than implicit QR. Consequently, our deflation algorithm consists of little more than moving the unwanted Ritz values, which are visible on the diagonals of S_m, to the southeast corner of the Rayleigh quotient and truncating the decomposition.

The following theorem shows just what a combined expansion and contraction step produces.

THEOREM 3.1. *Let*

$$\mathbb{P} := AU = UH + \beta u \mathbf{e}_k^{\mathrm{T}}$$

be an unreduced Arnoldi decomposition and let

$$\mathbb{Q} := AV = VS + u b^{\mathrm{H}}$$

be an equivalent Krylov–Schur form. Suppose that an implicitly restarted Arnoldi cycle is performed on $\mathbb{P}$ and a Krylov–Schur cycle is performed on $\mathbb{Q}$. If the same Ritz values are discarded in both and those Ritz values are distinct from the other Ritz values, then the resulting decompositions are equivalent.

Proof. We must show that the subspaces associated with the final results are the same. First note that the expansion phase results in equivalent decompositions. In fact, since $\mathcal{R}(U) = \mathcal{R}(V)$ and in both cases we are orthogonalizing the same Krylov sequence, the vectors $u_{k+1} \dots u_{m+1}$ and $v_{k+1}, \dots, v_{m+1}$ are the same up to multiples of modulus one.

Now assume that both algorithms have gone through the expansion phase and have moved the unwanted Ritz values to the end of the decomposition. At this point denote the first decomposition by

$$\hat{\mathbb{P}} := A\hat{U} = \hat{U}\hat{H} + \hat{\beta}\hat{u}\mathbf{e}_m^{\mathrm{T}}$$

and the second by

$$\hat{\mathbb{Q}} := A\hat{V} = \hat{V}\hat{S} + \hat{u}\hat{b}^{\mathrm{H}}.$$

Note that for both methods, the final truncation leaves the vector $\hat{u}$ unaltered. Since $\hat{V} = \hat{U}W$ for some unitary W, we have

$$\hat{S} = \hat{V}^{\mathrm{H}}A\hat{V} = W^{\mathrm{H}}\hat{U}^{\mathrm{H}}A\hat{U}W = W^{\mathrm{H}}\hat{H}W.$$

Thus $\hat{H}$ and $\hat{S}$ are similar and have the same Ritz values. Thus it makes sense to say that both methods reject the same Ritz values.

Let P be the unitary transformation applied to the Rayleigh quotient in $\hat{\mathbb{P}}$, and let Q be the one applied to the Rayleigh quotient of $\hat{\mathbb{Q}}$. Then we must show that the subspaces spanned by $\hat{U}P[:, 1{:}k]$ and $\hat{V}Q[:, 1{:}k]$ are the same. For brevity, set $P_k = P[:, 1{:}k]$ and $Q_k = Q[:, 1{:}k]$.

By construction $\mathcal{R}(P_k)$ is the eigenspace $\mathcal{P}$ of Schur vectors of $\hat{H}$ corresponding to the retained Ritz values. Likewise, $\mathcal{R}(Q_k)$ is the eigenspace $\mathcal{Q}$ of Schur vectors of $\hat{S}$ corresponding to the retained Ritz values. By hypothesis these eigenspaces are simple and hence are the same. Since $W^{\mathrm{H}}\hat{S}W = \hat{H}$, the matrix $W^{\mathrm{H}}\hat{P}_k$ spans $\mathcal{Q}$. Hence there is a unitary matrix R such that $Q_k = W^{\mathrm{H}}P_kR$. We then have

$$\hat{V}Q_k = \hat{U}WW^{\mathrm{H}}P_kR = \hat{U}P_kR.$$

It follows that $\hat{V}Q_k$ and $\hat{U}P_k$ span the same subspace. ∎

The import of this theorem is that no matter how you perform the expansion and contraction, mathematically you end up with a decomposition that has been filtered through the polynomial $(t - \mu_1) \dots (t - \mu_{m-k})$. However, the procedure based on the Krylov–Schur form is numerically more reliable than the one based on implicit restarting.

4. Numerical stability. We now briefly consider the numerical stability of the algorithm. From standard techniques of rounding error analysis it can be shown that as the Krylov–Schur algorithm proceeds, the computed Krylov decompositions satisfy

$$AU = UB + ub^{\mathrm{H}} + R, \tag{4.1}$$

where $\|R\|/\|A\|$ is of the order of the rounding unit and grows slowly. If U is computed with reorthogonalization in the expansion phase, $U^{\mathrm{H}}U = I + F$, where $\|F\|$ is the order

of the rounding unit and also grows slowly. The following theorem shows that we can throw the residual error R back on the matrix A.

THEOREM 4.1. *Let* (4.1) *be satisfied and assume that U is of full rank. Let $E = -RU^\dagger$, where $U^\dagger = (U^{\mathrm{H}}U)^{-1}U^{\mathrm{H}}$ is the pseudoinverse of U. Then*

$$(A+E)U = UB + ub^{\mathrm{H}}, \tag{4.2}$$

and

$$\frac{\|R\|}{\|U\|} \le \|E\| \le \|R\|\|U^\dagger\|.$$

The lower bound holds for any matrix E satisfying (4.2).

Proof. The equation (4.2) is established by direct verification. The upper bound follows from taking norms in the definition of E. On the other hand, if E is any matrix satisfying (4.2), then $EU = -R$, and $\|R\| \le \|E\|\|U\|$, which establishes the lower bound. ∎

Since U is nearly orthonormal, $\|U\|$ and $\|U^\dagger\|$ are near one. Hence the theorem shows that the computed generalized Arnoldi decomposition is an exact decomposition of a matrix near A. In this sense the Krylov–Schur algorithm (as well as the implicitly restarted Arnoldi algorithm) is backward stable.

5. Deflation and convergence. We now turn to the problem of deflating converged vectors from an orthonormal Krylov decomposition. We shall see later that if the concern is with a single Ritz vector then the deflation is easy. However, we can also use Krylov decompositions to deflate approximate eigenvectors or eigenspaces that are not obtained by a Rayleigh–Ritz procedure. Moreover, dependencies among the vectors to be deflated can cause the deflation procedure to require smaller residuals in the individual vectors. Consequently, we give a general analysis that covers both of these points.

We say a Krylov decomposition has been deflated if it can be partitioned in the form

$$A(U_1\ U_2) = (U_1\ U_2)\begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} + u(0\ b_2^{\mathrm{H}}).$$

When this happens, we have $AU_1 = U_1B_{11}$, so that U_{11} spans an eigenspace of A.

There are two advantages to deflating a converged eigenspace. First, by freezing it at the beginning of the Krylov decomposition we insure that the remaining space of the decomposition remains orthogonal to it. In particular, this gives algorithms the opportunity to compute more than one independent eigenvector corresponding to a multiple eigenvalue.

The second advantage of the deflated decomposition is that we can save operations in the contraction phase of an Arnoldi or Krylov–Schur cycle. The expansion phase does not change, and we end up with a decomposition of the form

$$A(U_1\ U_2\ U_3) = (U_1\ U_2\ U_3)\begin{pmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & B_{23} & B_{33} \end{pmatrix} + \beta u\mathbf{e}_m^{\mathrm{T}}.$$

Now since B_{11} is uncoupled from the rest of the Rayleigh quotient, we can apply all subsequent transformations exclusively to the eastern part the Rayleigh quotient and

to $(U_2\ U_3)$. If the order of B_{11} is small, the savings will be marginal; but as its size increases during the course of the algorithm, the savings become significant.

Of course, we will never have an exact eigenspace in our decompositions. Instead we will have a basis, say UW, for an approximate eigenspace and an approximation representation M of A on that subspace. The following theorem relates the norm of the residual $A(UW) - (UW)M$ to the quantities in the decomposition we must set to zero in order to deflate.

THEOREM 5.1. *Let*

$$AU = UB + ub^{\mathrm{H}} \tag{5.1}$$

be an orthonormal Krylov decomposition, and let $(M, Z) = (M, UW)$ *be given with* U *and* W *orthonormal. Let* $(W\ W_\perp)$ *be unitary, and set*

$$\tilde{B} = \begin{pmatrix} W^{\mathrm{H}} \\ W_\perp^{\mathrm{H}} \end{pmatrix} B(W\ W_\perp) = \begin{pmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ \tilde{B}_{21} & \tilde{B}_{22} \end{pmatrix}$$

and

$$\tilde{b}^{\mathrm{H}} = b^{\mathrm{H}}(W\ W_\perp) = (\tilde{b}_1^{\mathrm{H}}\ \tilde{b}_2^{\mathrm{H}}).$$

Then

$$\|AZ - ZM\|_{\mathrm{F}}^2 = \|\tilde{B}_{21}\|_{\mathrm{F}}^2 + \|\tilde{b}_1\|_{\mathrm{F}}^2 + \|\tilde{B}_{11} - M\|_{\mathrm{F}}^2, \tag{5.2}$$

where $\|\cdot\|_{\mathrm{F}}$ *denotes the Frobenius norm.*

Proof. From (5.1) we have $AZ - ZM = UBW - UWM + ub^{\mathrm{H}}W$. If we set

$$(\tilde{U}_1\ \tilde{U}_2) = (Z\ \tilde{U}_2) = U(W\ W_\perp),$$

then

$$\begin{aligned} AZ - ZM &= U(W\ W_\perp)\left[\begin{pmatrix} W^{\mathrm{H}} \\ W_\perp^{\mathrm{H}} \end{pmatrix} B(W\ W_\perp)\begin{pmatrix} I \\ 0 \end{pmatrix} - \begin{pmatrix} I \\ 0 \end{pmatrix} M\right] \\ &\quad + ub^{\mathrm{H}}(W\ W_\perp)\begin{pmatrix} I \\ 0 \end{pmatrix} \\ &= (\tilde{U}_1\ \tilde{U}_2)\begin{pmatrix} \tilde{B}_{11} - M \\ \tilde{B}_{21} \end{pmatrix} + u\tilde{b}_1^{\mathrm{H}}. \end{aligned}$$

The theorem now follows on taking norms. ∎

To see the consequences of this theorem, suppose that $AZ - ZM$ is small, and, using $(W\ W_\perp)$, we transform the Krylov decomposition $AU - UB = ub^{\mathrm{H}}$ to the form

$$A(\tilde{U}_1\ \tilde{U}_2) = (\tilde{U}_1\ \tilde{U}_2\ u)\begin{pmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ \tilde{B}_{21} & \tilde{B}_{22} \\ \tilde{b}_1^{\mathrm{H}} & \tilde{b}_2^{\mathrm{H}} \end{pmatrix}. \tag{5.3}$$

Then by (5.2)

$$\left\|\begin{pmatrix} \tilde{B}_{21} \\ \tilde{b}_1^{\mathrm{H}} \end{pmatrix}\right\|_{\mathrm{F}} \leq \|AZ - ZM\|_{\mathrm{F}}, \tag{5.4}$$

with equality if and only if M is the Rayleigh quotient $W^{\mathrm{H}}BW$. Thus if the residual norm $\|AZ - ZM\|_{\mathrm{F}}$ is sufficiently small, we may set $\tilde{B}_{21}$ and $\tilde{b}_1$ to zero to get the deflated

$$A(\tilde{U}_1\ \tilde{U}_2) \cong (\tilde{U}_1\ \tilde{U}_2)\begin{pmatrix}\tilde{B}_{11} & \tilde{B}_{12}\\ 0 & \tilde{B}_{22}\end{pmatrix} + u(0\ b_2^{\mathrm{H}}). \tag{5.5}$$

The deflation procedure that leads to (5.5) is backwards stable. If we restore the quantities that were zeroed in forming (5.5), we get the following relation:

$$A(\tilde{U}_1\ \tilde{U}_2) = (\tilde{U}_1\ \tilde{U}_2)\begin{pmatrix}\tilde{B}_{11} & \tilde{B}_{12}\\ 0 & \tilde{B}_{22}\end{pmatrix} + u(0\ b_2^{\mathrm{H}}) + \tilde{U}_2\tilde{B}_{21} + u\tilde{b}_1^{\mathrm{H}}.$$

If we write this decomposition in the form

$$A\tilde{U} = \tilde{U}\breve{B} + u\breve{b}^{\mathrm{H}} + R, \quad \text{where} \quad R = \tilde{U}_2\tilde{B}_{21} + u\tilde{b}_1^{\mathrm{H}},$$

then

$$\|R\|_{\mathrm{F}} \le \|AZ - ZM\|_{\mathrm{F}}.$$

If we now set $E = R\tilde{U}^{\mathrm{H}}$, then $(A + E)\tilde{U} = \tilde{U}\breve{B} + u\breve{b}^{\mathrm{H}}$. We may summarize these results in the following theorem.

THEOREM 5.2. *Under the hypotheses of Theorem* 5.1, *write the deflated decomposition* (5.5) *in the form*

$$A\tilde{U} \cong \tilde{U}\breve{B} + u\breve{b}^{\mathrm{H}}.$$

Then there is an E *satisfying*

$$\|E\|_{\mathrm{F}} \le \|AZ - ZM\|_{\mathrm{F}} \tag{5.6}$$

such that

$$(A + E)\tilde{U} = \tilde{U}\breve{B} + u\breve{b}^{\mathrm{H}}.$$

Equality holds in (5.6) *if and only if* M *is the Rayleigh quotient* $Z^{\mathrm{H}}AZ = W^{\mathrm{H}}BW$.

Because backward stability is commonly used to determine convergence, Theorems 5.1 and 5.2 suggest how one might combine convergence testing and deflation. Given an approximate pair (M, UW), we transform to the tilde form as in Theorem 5.1 and compute the backward error that would result from deflation. If this is small enough compared with A, we deem the pair to have converged and deflate.[3]

In practice we will seldom encounter a converging subspace unless it is a 2-dimensional subspace corresponding to a complex eigenvalue in a real Schur decomposition. Instead we will be confronted with converged, normalized Ritz pairs $(\mu_i, \hat{z}_i)$ $(i = 1, \ldots, p)$ of one kind or another, and the vectors in these pairs cannot be guaranteed to be orthogonal. If we arrange the vectors in a matrix $\hat{Z}$ and set $\hat{M} = \operatorname{diag}(\mu_1, \ldots, \mu_p)$, the residual $\hat{R} = A\hat{Z} - \hat{Z}\hat{M}$ must be small because the individual residuals are small.

[3]If the concern is with eigenvalues that are small compared with $\|A\|_{\mathrm{F}}$, we may have to demand a smaller backward error to get accurate results. For more, see the discussion of convergence in [13].

The deflation procedure requires an orthonormal basis for the approximate eigenspace in question, which is given by the QR factorization

$$\hat{Z} = ZT \tag{5.7}$$

of $\hat{Z}$. Unfortunately, the residual for Z becomes

$$R = \hat{R}T^{-1} = A\hat{Z}T^{-1} - \hat{Z}\hat{M}T^{-1} = AZ - ZM,$$

where $M = T\hat{M}T^{-1}$. If the columns of $\hat{Z}$ are nearly dependent, $\|T^{-1}\|$ will be large, and the residual may be magnified—perhaps to the point where the deflation cannot be performed safely. The effects of dependency on a different deflation algorithm have also been noted in [8].

It may seem paradoxical that we could have, say, two vectors each of which we can deflate but which taken together cannot be deflated. The resolution of this paradox is to remember that we are not deflating two vectors but the subspace spanned by them. If the vectors are nearly dependent, they must be very accurate to determine their common subspace accurately.

As we have mentioned, the deflation procedure is not confined to eigenpairs calculated by a Rayleigh–Ritz procedure. For example, it can be used to deflate harmonic Ritz vectors [10] or refined Ritz vectors [5]. However, if Ritz vectors are the concern, there is an easy way to deflate them in the Krylov–Schur method. After a cycle of the algorithm, let the current decomposition have the form

$$A(U_1\ U_2) = (U_1\ U_1)\begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} + u(0\ b_2^{\mathrm{H}}).$$

Here U_1 represents a subspace that has already been deflated, and S_{22} is the Schur form that remains after the contraction phase.

In this decomposition, to deflate the Ritz pair corresponding to the $(1,1)$-element of S_{22} we must set the first component of b_2 to zero. Consequently, all we have to do to deflate is to verify that that component satisfies our deflation criterion. If some other diagonal element of S_{22} is the candidate for deflation, we can exchange it into the $(2,2)$-position and test as above.

6. Assessment. In comparing the Krylov–Schur algorithm with the implicitly restarted Arnoldi algorithm, we must distinguish the sources of work in the algorithms. The first is the multiplication of a vector by A. Since A will usually be sparse, the cost of this product is unpredictable in general, but it is reasonable to assume that it forms a significant part—perhaps the dominant part—of the computation.

The second source of work is the expansion of the decompositions from one of order k to one of order m. It is easily seen from (3.1) that the work is $2n(m^2 - k^2)$ floating-point adds and multiplies, assuming reorthogonalization is performed. This count is the same for both algorithms.

In the contraction step, both algorithms must transform the Rayleigh quotient and accumulate the transformations in U. For efficiency, we do not accumulate the transformations in U as they are generated but instead accumulate them in an $m \times m$ matrix Q and then compute the new U_k in the form

$$U_m Q[:, 1{:}k]. \tag{6.1}$$

If $n \gg m$, the last step will dominate the transformations applied to the Rayleigh quotient and their accumulation in Q.

For the Krylov–Schur method we must compute the Schur decomposition of the Rayleigh quotient and transform the triangular factor. This means that Q will be full, and the final accumulation step (6.1) will require nkm floating-point additions and multiplications.

For the implicitly restarted Arnoldi we must also compute the Schur decomposition of the Rayleigh quotient H_m. But it is used only to determine the shifts, which are applied directly to H_m. The structure of the transformations is such that $Q[:, 1{:}k]$ is zero below its $m-k$ subdiagonal. This means that the operation count for (6.1) is $nmk - \frac{1}{2}k^2$ additions and multiplication.

To put things together, if $m = 2k$ and reorthogonalization is performed during the expansion, the Krylov–Schur algorithm has an operation count of $7nk^2$ whereas implicitly restarted Arnoldi has an operation count of $6\frac{1}{2}nk^2$. Thus implicitly restarted Arnoldi is marginally superior to Arnoldi–Schur when it comes to accumulation of transformations. Against this must be set the fact that Krylov–Schur deflates in an inexpensive and natural manner and does not require a special routine for purging.

7. Concluding remarks. The Krylov–Schur method admits variations. An important one is based on the observation that we can truncate a Krylov decomposition at any point where the Rayleigh quotient is block triangular [see (3.4)]. This means that when A is real we can work with real Schur forms of the Rayleigh quotient and avoid the necessity of complex arithmetic. The algorithm for exchanging eigenvalues mentioned above will also move the 2×2 blocks of the real Schur form so that the contraction phase proceeds as usual. In deflation, the block in question is moved to the position just after the previously deflated eigenvalues and blocks, and two components of b are tested. An unusual feature of complex eigenvectors is that they may fail to deflate, not because they are dependent on other deflated vectors, but because the real and imaginary parts of their eigenvectors are not sufficiently independent.

When A is Hermitian, the Krylov–Schur method becomes a restarted Lanczos algorithm—in fact the algorithm of Wu and Simon [18]. The Rayleigh quotient is diagonal, so that reordering of the eigenvalues reduces to simple permutations. Moreover, because the eigenvectors of the Rayleigh quotient are orthogonal, a Ritz pair with a small residual norm ϵ will deflate with backward error of order ϵ.

Since the Krylov–Schur method works explicitly with the eigenvalues of the Rayleigh quotient, it is an exact-shift method. Nonetheless, it stands ready to help the general shift method to deflate Ritz pairs and to get rid of unwanted pairs. One simply computes a Krylov–Schur form of the current decomposition and performs the procedures described above. Theorem 2.2 assures us that we can then return to a pure Arnoldi decomposition.

In fact Theorem 2.2 is really the heart of the matter. It allows us to operate freely on the Rayleigh quotient with the knowledge that we are always attached to a Krylov sequence. It is hoped that this freedom will find other applications.

Acknowledgment. I would like to thank Rich Lehoucq and Dan Sorensen for their comments on preliminary versions of this paper. I am indebted to the Mathematical and Computational Sciences Division of the National Institute of Standards and Technology for the use of their research facilities.

REFERENCES

[1] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] Z. Bai and J. W. Demmel, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95.
[3] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1998), pp. 94–125.
[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.
[5] Z. Jia, *Refined iterative algorithm based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.
[6] 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. 255–282.
[7] R. B. Lehoucq, *private communication.*
[8] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.
[9] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
[10] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154–156 (1991), pp. 289–309.
[11] R. B. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.
[12] R. B. Morgan, *GMRES with Deflated Restarting*, Department of Mathematics, Baylor University, Waco, TX, 1999.
[13] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse, real, unsymmetric matrices*, ACM Trans. Math. Software, 21 (1995), pp. 432–475.
[14] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[15] 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.
[16] G. W. Stewart, *Matrix Algorithms* I: *Basic Decompositions*, SIAM, Philadelphia, 1998.
[17] D. S. Watkins, *Forward stability and transmission of shifts in the QR algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 469–487.
[18] K. Wu and H. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 615–631
© 2001 Society for Industrial and Applied Mathematics

THE NUMERICAL RANGE OF SELF-ADJOINT QUADRATIC MATRIX POLYNOMIALS*

PETER LANCASTER† AND PANAYIOTIS PSARRAKOS‡

Abstract. Matrix polynomials of the form $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ (where A_0 and A_1 are $n \times n$ Hermitian matrices and λ is a complex variable) arise in many applications. The numerical range of such a polynomial is

$$W(P) = \{\lambda \in \mathbb{C} : x^*P(\lambda)x = 0 \text{ for some nonzero } x \in \mathbb{C}^n\}$$

and it always contains the spectrum of $P(\lambda)$, i.e., the set of zeros of $\det P(\lambda)$. Properties of the numerical range are developed in detail, taking advantage of the close connection between $W(P)$ and the classical numerical range (field of values) of the (general) complex matrix $A := A_0 + iA_1$.

Eigenvalues and nondifferentiable points on the boundary are examined and a procedure for the numerical determination of $W(P)$ is presented and used for several illustrations. Some extensions of the theory to more general polynomials $P(\lambda)$ are also discussed, as well as special cases describing vibrating systems.

Key words. matrix polynomial, numerical range, eigenvalue, boundary

AMS subject classifications. 15A60, 15A63

PII. S0895479899364088

1. Introduction and fundamental concepts.

Consider the $n \times n$ *self-adjoint quadratic matrix polynomial*

$$P(\lambda) = A_2\lambda^2 + A_1\lambda + A_0, \tag{1}$$

where A_j $(j = 0, 1, 2)$ are $n \times n$ Hermitian matrices and λ is a complex variable. A complex number λ_0 is called an *eigenvalue* of $P(\lambda)$ if the equation $P(\lambda_0)x = 0$ has a nonzero solution in $\mathbb{C}^n$. The set of all eigenvalues of $P(\lambda)$ is known as the *spectrum* of $P(\lambda)$ and is written

$$\sigma(P) = \{\lambda \in \mathbb{C} : \det P(\lambda) = 0\}.$$

The *numerical range* of $P(\lambda)$ is defined as

$$W(P) = \{\lambda \in \mathbb{C} : x^*P(\lambda)x = 0 \text{ for some nonzero } x \in \mathbb{C}^n\}. \tag{2}$$

Evidently, $W(P)$ is always closed and contains $\sigma(P)$. For $P(\lambda) = I\lambda - A$, $W(P)$ coincides with the *classical numerical range (field of values)* of matrix A,

$$F(A) = \{x^*Ax \in \mathbb{C} : x \in \mathbb{C}^n \text{ with } x^*x = 1\}.$$

This paper presents a careful discussion of properties of $W(P)$ with emphasis on the connections with the spectrum of $P(\lambda)$, and also on monic polynomials. In the case of $F(A)$ a classical result (see [D], for example) states that nondifferentiable points

*Received by the editors November 18, 1999; accepted for publication (in revised form) by A.C.M. Ran July 13, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/simax/23-3/36408.html
†Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, T2N 1N4, Canada (lancaste@ucalgary.ca).
‡Department of Mathematics, National Technical University, Zografou Campus, Athens 15780, Greece (ppsarr@math.ntua.gr).

on the boundary $\partial F(A)$ are necessarily eigenvalues of A. Generalizations of this and related results to points on $\partial W(P)$ are considered in sections 4 and 5. The results are illustrated with several examples and, in sections 7, 8, and 9, with descriptions of the numerical ranges of damped systems and gyroscopic systems from the theory of vibrations.

A nonzero vector x_0 in $\mathrm{Ker}P(\lambda_0)$ is known as an *eigenvector* of $P(\lambda)$ corresponding to the eigenvalue λ_0, and vectors $x_1, x_2, \ldots, x_k$ are said to be *associated* with x_0 if

$$\sum_{j=1}^{p} \frac{1}{j!} P^{(j)}(\lambda_0) x_{p-j} = 0\,; \quad p = 1, 2, \ldots, k.$$

The system of vectors $x_0, x_1, x_2, \ldots, x_k$ is called a *Jordan chain* (of *length* $k+1$) of $P(\lambda)$ corresponding to the eigenvalue λ_0 and generates fundamental solutions of the differential equation

$$A_2 u''(t) + A_1 u'(t) + A_0 u(t) = 0.$$

The spectrum of $P(\lambda)$ either coincides with the complex plane $\mathbb{C}$ or contains no more than $2n$ points. The multiplicity of λ_0 as a root of the equation $\det P(\lambda) = 0$ is called the *algebraic multiplicity* of λ_0. The dimension of the kernel, $\mathrm{Ker}P(\lambda_0)$, is called the *geometric multiplicity* of λ_0 and is no greater than the algebraic multiplicity. If both multiplicities coincide, then the corresponding eigenvalue λ_0 is called *semisimple*. In this case, all the Jordan chains of the eigenvalue λ_0 have length equal to 1.

It is obvious that $W(P)$ (for (1)) is symmetric with respect to the real axis, and it is known that $W(P)$ is bounded if and only if $0 \notin F(A_2)$ (see [LR]). Moreover, if $W(P)$ is bounded, then it has either 1 or 2 connected components. If $W(P)$ is unbounded, then it may have as many as 4 connected components.

In our discussion, we will need the *joint numerical range* of the triple (A_0, A_1, A_2),

$$JNR(A_0, A_1, A_2) = \{(x^*A_0x, x^*A_1x, x^*A_2x) \in \mathbb{R}^3 : x \in \mathbb{C}^n,\ x^*x = 1\}. \tag{3}$$

The joint numerical range $JNR(A_0, A_1, A_2)$ is convex for $n \geq 3$ [AT]. If $n = 2$, then it is either convex or the surface of an ellipsoid.

Most of the results of this paper are also valid for quadratic polynomials on an infinite-dimensional Hilbert space H, whose coefficients are self-adjoint bounded linear operators, provided $W(P)$, $F(A)$, and $JNR(A_0, A_1, A_2)$ are replaced by their closures $\overline{W(P)}$, $\overline{F(A)}$, and $\overline{JNR(A_0, A_1, A_2)}$, respectively. This modification is needed to account for the fact that the numerical ranges of operators need not be compact sets. The basic exception is Theorem 6. This result is proved in the matrix case only, because we use the properties of normal eigenvalues of matrices. As a consequence, Theorem 7, Corollary 8, and Theorem 14 (for nonreal points) are also proved for the matrix case only. The rest of the results are valid in the infinite-dimensional case, since their proofs are based on the connection between the boundaries of numerical ranges (and it is not assumed that $\partial F(A) \subset F(A)$ or $\partial W(P) \subset W(P)$).

2. The nonmonic case. Let $P(\lambda) = A_2\lambda^2 + A_1\lambda + A_0$ ($A_2 \neq 0$) be an $n \times n$ self-adjoint quadratic matrix polynomial with numerical range $W(P)$ as in (2) and let the joint numerical range of its coefficients be $JNR(A_0, A_1, A_2)$ as in (3). Suppose that $(a_0, a_1, a_2) \in \mathbb{R}^3$ and the equation $a_2\lambda^2 + a_1\lambda + a_0 = 0$ has nonreal roots λ_0 and $\overline{\lambda}_0$. Then for every point (b_0, b_1, b_2) of the open halfline

$$\varepsilon = \{t(a_0, a_1, a_2) \in \mathbb{R}^3 : t \in (0, +\infty)\},$$

the equation $b_2\lambda^2 + b_1\lambda + b_0 = 0$ has the same roots. So, if we define the *supporting cone* of $JNR(A_0, A_1, A_2)$ as

$$\mathcal{K} = \bigcup_{t>0} tJNR(A_0, A_1, A_2), \tag{4}$$

then

$$\begin{aligned} W(P) &= \{\lambda \in \mathbb{C} : a_2\lambda^2 + a_1\lambda + a_0 = 0,\ (a_0, a_1, a_2) \in JNR(A_0, A_1, A_2)\} \\ &= \{\lambda \in \mathbb{C} : a_2\lambda^2 + a_1\lambda + a_0 = 0,\ (a_0, a_1, a_2) \in \mathcal{K}\} \\ &= \{\lambda \in \mathbb{C} : a_2\lambda^2 + a_1\lambda + a_0 = 0,\ (a_0, a_1, a_2) \in \partial JNR(A_0, A_1, A_2)\}. \end{aligned}$$

Consequently, two conjugate complex numbers λ_0 and $\overline{\lambda}_0$ belong to $W(P)$ if and only if the line

$$\varepsilon = \{t(|\lambda_0|^2, -2\mathrm{Re}\lambda_0, 1) \in \mathbb{R}^3 : t \in \mathbb{R}\}$$

intersects $JNR(A_0, A_1, A_2)$. Note also that $(0, 0, 0) \in JNR(A_0, A_1, A_2)$ if and only if $(0, 0, 0) \in \mathcal{K}$ and, in this case, $W(P) \equiv \mathbb{C}$.

THEOREM 1. *Let $P(\lambda) = A_2\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ self-adjoint matrix polynomial with $W(P) \neq \mathbb{C}$ and let $\lambda_0 \in W(P)\backslash\mathbb{R}$ and $(b_0, b_1, b_2) \in \mathcal{K}$ be such that $b_2\lambda_0^2 + b_1\lambda_0 + b_0 = 0$. Then $\lambda_0 \in \partial W(P)\backslash\mathbb{R}$ if and only if $(b_0, b_1, b_2) \in \partial\mathcal{K}$.*

Proof. Since $W(P) \neq \mathbb{C}$, then $(0, 0, 0) \notin JNR(A_0, A_1, A_2)$, and if $\lambda_0 \in \partial W(P)\backslash\mathbb{R}$, then there exists a sequence $\{\lambda_k\}_{k\in\mathbb{N}} \in \mathbb{C}\backslash(\mathbb{R}\cup W(P))$ converging to λ_0. The sequence of corresponding lines

$$\varepsilon_k = \{t(|\lambda_k|^2, -2\mathrm{Re}\lambda_k, 1) \in \mathbb{R}^3 : t \in \mathbb{R}\}\,; \quad k \in \mathbb{N},$$

converges to the line

$$\varepsilon_0 = \{t(|\lambda_0|^2, -2\mathrm{Re}\lambda_0, 1) \in \mathbb{R}^3 : t \in \mathbb{R}\}.$$

(Here, the absolute value of $\tan(\widehat{\varepsilon_1, \varepsilon_2})$ is considered as the distance between the lines ε_1 and ε_2.)

Since $\lambda_k \notin W(P)$ $(k \in \mathbb{N})$, the lines ε_k do not intersect $JNR(A_0, A_1, A_2)$. Thus, ε_0 is a supporting line of $JNR(A_0, A_1, A_2)$ and, consequently, (b_0, b_1, b_2) is a boundary point of $\mathcal{K}$.

For the converse assume that $(b_0, b_1, b_2) \in \partial\mathcal{K}$. Then $\varepsilon_0 = \{t(b_0, b_1, b_2) \in \mathbb{R}^3 : t \in \mathbb{R}\}$ is a supporting line of $JNR(A_0, A_1, A_2)$ and there exists a sequence of lines

$$\varepsilon_k = \{t(b_{0,k}, b_{1,k}, b_{2,k}) \in \mathbb{R}^3 : b_{1,k}^2 < 4b_{0,k}b_{2,k},\ t \in \mathbb{R}\}\,; \quad k \in \mathbb{N},$$

converging to ε_0 such that $\varepsilon_k \cap JNR(A_0, A_1, A_2) = \emptyset$ for every $k \in \mathbb{N}$. If λ_k and $\overline{\lambda}_k$ are the nonreal roots of equation

$$b_{2,k}\lambda^2 + b_{1,k}\lambda + b_{0,k} = 0\,; \quad k \in \mathbb{N},$$

then the sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ converges to λ_0 (because the roots of a polynomial depend continuously on the coefficients of the polynomial). Moreover, $\{\lambda_k\}_{k\in\mathbb{N}} \in \mathbb{C}\backslash(\mathbb{R} \cup W(P))$ and λ_0 is a boundary point of $W(P)\backslash\mathbb{R}$. ∎

COROLLARY 2. *Let $P(\lambda)$ be a matrix polynomial as in Theorem 1, let $\lambda_0 \in W(P)\backslash\mathbb{R}$, and let $(a_0, a_1, a_2) \in JNR(A_0, A_1, A_2)$ such that $a_2\lambda_0^2 + a_1\lambda_0 + a_0 = 0$. Then $\lambda_0 \in \partial W(P)$ if and only if $(a_0, a_1, a_2) \in \partial JNR(A_0, A_1, A_2) \cap \partial\mathcal{K}$.*

3. The monic case. If the matrix polynomial $P(\lambda)$ in (1) is *monic*, i.e., $A_2 = I$, then $JNR(A_0, A_1, A_2)$ is just the numerical range of the *associate matrix* $A = A_0 + iA_1$. Moreover,

$$W(P) = \{\lambda \in \mathbb{C} : \lambda^2 + a_1\lambda + a_0 = 0,\ a_0, a_1 \in \mathbb{R} \text{ and } a_0 + ia_1 \in F(A)\}.$$

THEOREM 3. *Let $P(\lambda)$ be an $n \times n$ monic self-adjoint matrix polynomial, let $\lambda_0 \in W(P)\backslash\mathbb{R}$, and let $a_0 + ia_1 \in F(A)$ such that $\lambda_0^2 + a_1\lambda_0 + a_0 = 0$. Then $\lambda_0 \in \partial W(P)\backslash\mathbb{R}$ if and only if $a_0 + ia_1 \in \partial F(A)$.*

Proof. Consider the cone $\mathcal{K}$ in (4). The joint numerical range $JNR(A_0, A_1, I)$ in (3) is a convex subset of the plane $\{(u, v, 1) \in \mathbb{R}^3 : (u, v) \in \mathbb{R}^2\} \subset \mathbb{R}^3$. Consequently, $\partial JNR(A_0, A_1, I) \cap \partial\mathcal{K} = \{(u, v, 1) \in \mathbb{R}^3 : u + iv \in \partial F(A)\}$, and the result follows immediately from Corollary 2. ∎

A method for the numerical determination of $\partial F(A)$ is presented in [HJ]. Using this method and Theorem 3, an algorithm to determine points on $\partial W(P)$ can be formulated and is used in subsequent examples. Unit vectors $x \in \mathbb{C}^n$ are found such that $x^*Ax \in \partial F(A)$, and then the nonreal roots of polynomials

$$x^*P(\lambda)x = \lambda^2 + (x^*A_1x)\lambda + x^*A_0x$$

are boundary points of $W(P)$. Thus, $\partial W(P)$ is a double image of the curve $\partial F(A)$. Since $\partial F(A)$ may include linear segments (which is certainly the case when A is normal), special provision is made for generating points on such segments (see Step 3 below). The algorithm has the following form:

Step 1. Choose a partition $0 = \theta_0, \theta_1, \ldots, \theta_s = 2\pi$ of the interval $[0, 2\pi]$ and set the number of points, R, to be interpolated on linear segments.

Step 2. For $k = 1, 2, \ldots, s$ compute the largest eigenvalue λ_k of the matrix

$$H_k = \cos\theta_k A_0 - \sin\theta_k A_1$$

and a corresponding eigenvector y_k. The point $a_0 + ia_1 = y_k^*A_0y_k + iy_k^*A_1y_k = y_k^*Ay_k$ is a point of $\partial F(A)$. Check for a linear segment joining the kth and $(k-1)$st boundary points. If there isn't one, go to Step 3; if there is, interpolate a set of $R - 1$ points in the linear segment and then go to Step 3.

Step 3. For all k and for all interpolated points on linear segments, compute the zeros of $\lambda^2 + a_1\lambda + a_0$. These are either nonreal points of $\partial W(P)$ or real points of $W(P)$.

In the following examples and elsewhere, observe the relative positions of $F(A)$ and the parabola $\mathcal{D} = \{u + iv \in \mathbb{C} : u, v \in \mathbb{R},\ v^2 = 4u\}$. Note also that the eigenvalues of $P(\lambda)$ are indicated along with $\partial W(P)$. These examples will be useful in what follows.

Example 1. Consider $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ with

$$A_1 = \begin{bmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 1.1 & 1 & 0 & 0 \\ 1 & 1.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 2.1 \end{bmatrix}.$$

The first part of Figure 1 shows the boundary of $F(A)$ and parabola $\mathcal{D}$. Since $F(A)$ is "inside" $\mathcal{D}$, $W(P)$ has no real points and consists of two connected components.

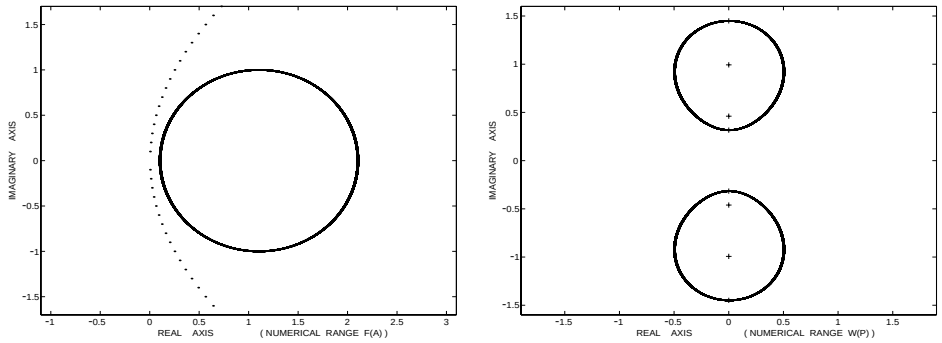


FIG. 1. *A numerical range with two components.*

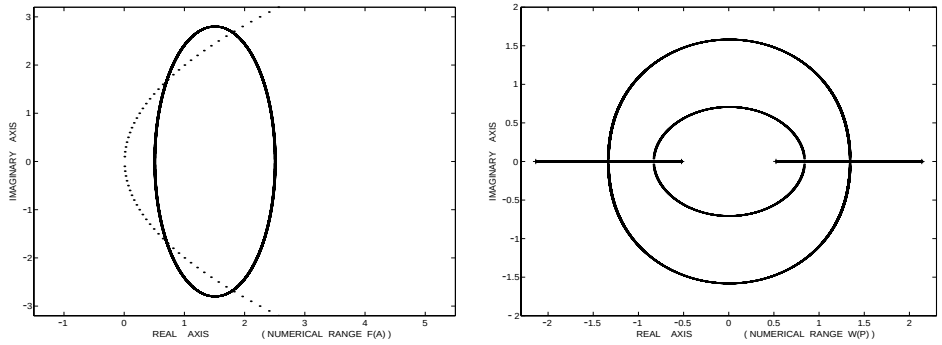


FIG. 2. *A numerical range with one component.*

Example 2. For the coefficients

$$A_1 = \begin{bmatrix} 0 & 2.8i \\ -2.8i & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 1.5 & 1 \\ 1 & 1.5 \end{bmatrix},$$

$\partial F(A)$ and $\partial W(P)$ are sketched in Figure 2. Now $F(A)$ is an elliptic disc which intersects with $\mathcal{D}$, so the part of $F(A)$ "inside" $\mathcal{D}$ is mapped onto the nonreal part of $W(P)$, etc.

Example 3. Take

$$A_1 = (-i)\begin{bmatrix} 0 & 0.2 & -0.2 \\ -0.2 & 0 & 0.2 \\ 0.2 & -0.2 & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.2 & 0.5 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{bmatrix},$$

and see Figure 3. Note the presence of linear segments on $\partial F(A)$.

It will be useful to introduce the closure of the nonreal points in $W(P)$, say $\mathcal{S} := \overline{W(P)\backslash\mathbb{R}}$. Then the real part of $W(P)$ can be written as a disjoint union:

$$\mathbb{R} \cap W(P) = (\mathbb{R} \cap \mathcal{S}) \cup (W(P)\backslash\mathcal{S}). \tag{5}$$

PROPOSITION 4. *Assume that* $\mathcal{S} \neq \emptyset$.

(i) *If* $a_0 + ia_1 \in \mathcal{D} \cap F(A)$, *then* $-a_1/2 \in \mathbb{R} \cap \mathcal{S}$.

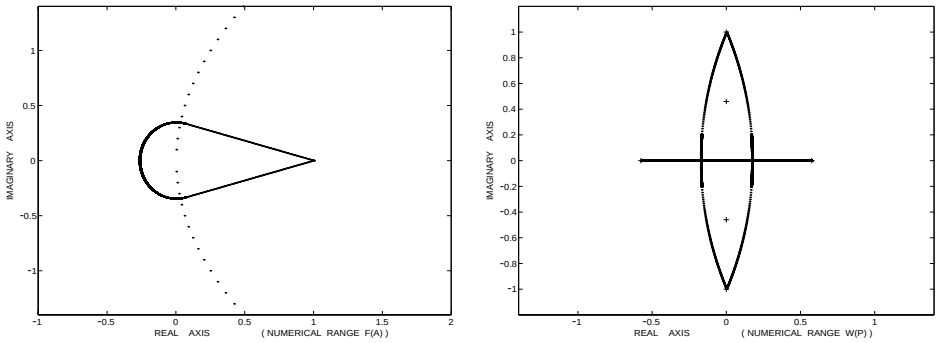


FIG. 3. *Linear segments on* $\partial F(A)$.

(ii) *Conversely, if* $\lambda_0 \in \mathbb{R} \cap \mathcal{S}$, *then there exists a nonzero* $x \in \mathbb{C}^n$ *such that*

$$a_0 + ia_1 := x^*A_0x + i(x^*A_1x) \in \mathcal{D} \cap F(A).$$

Proof. (i) $a_0 + ia_1 \in F(A)$ implies that there is an $x \neq 0$ such that $a_0 = x^*A_0x$, $a_1 = x^*A_1x$. In addition, if $a_0 + ia_1 \in \mathcal{D}$, then there is an $x \neq 0$ such that $a_1^2 = 4a_0$. Thus, $\lambda^2 + a_1\lambda + a_0 = (\lambda + a_1/2)^2$ and $-a_1/2 \in \mathbb{R} \cap W(P)$.

If $\mathbb{R} \cap W(P) = \{-a_1/2\}$, then $W(P)$ has only one component and it follows that $-a_1/2 \in \mathbb{R} \cap \mathcal{S}$.

If $\mathbb{R} \cap W(P)$ is not a singleton and $-a_1/2 \notin \mathcal{S}$, then there is no sequence of nonreal points in $W(P)$ converging to $-a_1/2$. It follows that $F(A)$ must lie "outside" $\mathcal{D}$ and, since $F(A)$ is convex, $\mathcal{D} \cap F(A) = \{a_0 + ia_1\}$. But this would imply that $W(P) \subset \mathbb{R}$ and $\mathcal{S} = \emptyset$, and a contradiction is obtained. Hence $-a_1/2 \in \mathbb{R} \cap \mathcal{S}$.

(ii) If a real $\lambda_0 \in \mathcal{S}$, then, since $W(P)$ is closed, $\lambda_0 \in W(P)$. Furthermore, there exists a sequence $\{\lambda_k\}$ of nonreal points in $W(P)$ converging to λ_0 *and* there is an $x \neq 0$ such that $\lambda^2 + a_1\lambda + a_0 = 0$. Then there is a corresponding sequence $\{a_{0k} + ia_{1k}\} \subset F(A)$, and also in $\mathcal{D}$, such that $a_{0k} + ia_{1k} \to a_0 + ia_1 \in \mathcal{D}$. Since $F(A)$ is closed, $a_0 + ia_1 \in F(A)$ as well. ∎

4. Eigenvalues on the boundary. Next we investigate the eigenvalues of $P(\lambda)$ on $\partial W(P)$ for monic $P(\lambda)$. In particular, the eigenvalues of $P(\lambda)$ on $\partial W(P)$ are directly connected with the eigenvalues of A on $\partial F(A)$. This is illustrated in Example 1, where A has eigenvalues 0.1 and 2.1 on $\partial F(A)$, and there are corresponding eigenvalues $\pm i\sqrt{0.1}$ and $\pm i\sqrt{2.1}$ of $P(\lambda)$ on $\partial W(P)$.

The next lemma is easily proved (see [MP2]).

LEMMA 5. *Let* $y_0 \in \mathbb{C}^n$ *be a common eigenvector of* A_0 *and* A_1 *corresponding to eigenvalues* μ_0 *and* μ_1, *respectively. Then the zeros of* $\lambda^2 + \mu_1\lambda + \mu_0$ *are eigenvalues of* $P(\lambda)$ *and* y_0 *is the corresponding eigenvector of* $P(\lambda)$.

Now consider the nonreal part of $\partial W(P)$.

THEOREM 6. *Let* $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ *be an* $n \times n$ *monic self-adjoint quadratic matrix polynomial, let* $\lambda_0 \in W(P)\backslash\mathbb{R}$, *and let* $a_0 + ia_1 \in F(A)$ *such that* $\lambda_0^2 + a_1\lambda_0 + a_0 = 0$. *Then* λ_0 *is an eigenvalue of* $P(\lambda)$ *on* $\partial W(P)$ *if and only if* $a_0 + ia_1$ *is an eigenvalue of matrix* $A = A_0 + iA_1$ *on* $\partial F(A)$.

Proof. Assume that λ_0 is an eigenvalue of $P(\lambda)$ on $\partial W(P)$. By Theorem 1.1 in [MP1], 0 is an eigenvalue of matrix $P(\lambda_0)$ on the boundary of $F(P(\lambda_0))$. Consequently, by Theorem 1.6.6 in [HJ], there exists a unitary matrix V such that

$$V^*P(\lambda_0)V = I\lambda_0^2 + (V^*A_1V)\lambda_0 + V^*A_0V = 0_k \oplus T_0, \tag{6}$$

where $1 \le k \le n$, 0_k is the $k \times k$ zero matrix, and T_0 is an $(n-k)\times(n-k)$ nonsingular upper triangular matrix (depending on λ_0). Thus,

$$V^*(\mathrm{Re}\lambda_0 A_1 + A_0)V + iV^*(\mathrm{Im}\lambda_0 A_1)V = 0_k \oplus T_0 - I\lambda_0^2 = 0_k \oplus T_0 + I(a_1\lambda_0 + a_0).$$

After comparing Hermitian and skew-Hermitian parts, we obtain

$$V^* A_1 V = -2\mathrm{Re}\lambda_0 I_k \oplus H_1 = a_1 I_k \oplus H_1$$

and

$$V^* A_0 V = |\lambda_0|^2 I_k \oplus H_0 = a_0 I_k \oplus H_0,$$

where H_0 and H_1 are $(n-k)\times(n-k)$ Hermitian matrices and I_k is the $k \times k$ identity matrix. So, $V^*(A_0 + iA_1)V = (a_0 + ia_1)I_k \oplus (H_0 + iH_1)$ and $a_0 + ia_1$ is an eigenvalue of matrix $A = A_0 + iA_1$. Moreover, by Theorem 3, $a_0 + ia_1$ is a boundary point of $F(A)$.

For the converse, assume that $a_0 + ia_1 \in \partial F(A)$ is an eigenvalue of A corresponding to the unit eigenvector y_0 of A. By Theorem 3, λ_0 is a boundary point of $W(P)$ and by Theorem 1.6.6 in [HJ], $a_0 + ia_1$ is a normal eigenvalue of matrix A. So there exists a unitary matrix U, with y_0 as its first column (Schur's theorem) such that

$$U^* A U = (a_0 + ia_1) \oplus T,$$

where T is an $(n-1)\times(n-1)$ upper triangular matrix. We write $H(T) = (T + T^*)/2$ and $S(T) = (T - T^*)/(2i)$, the Hermitian and the skew-Hermitian part of matrix T, respectively. Then

$$U^* A_0 U + iU^* A_1 U = a_0 \oplus H(T) + i\,(a_1 \oplus S(T))$$

and, consequently,

$$U^* A_0 U = a_0 \oplus H(T) \qquad \text{and} \qquad U^* A_1 U = a_1 \oplus S(T).$$

Hence, $A_0 y_0 = a_0 y_0$ and $A_1 y_0 = a_1 y_0$ and, using Lemma 5, λ_0 is an eigenvalue of $P(\lambda)$. □

If $a_0 + ia_1 \in \sigma(A) \cap \partial F(A)$ and the equation $\lambda^2 + a_1\lambda + a_0 = 0$ has a *real* root λ_0, then by the proof of Theorem 6, λ_0 is an eigenvalue of $P(\lambda)$, but it may belong to the interior of $W(P)$.

Note that in the previous theorem, the eigenvalues $\lambda_0 \in \sigma(P) \cap \partial W(P)$ and $a_0 + ia_1 \in \sigma(A) \cap \partial F(A)$ have the same eigenvectors, i.e.,

$$\mathrm{Ker}P(\lambda_0) \equiv \mathrm{Ker}[(a_0 + ia_1)I - A].$$

THEOREM 7. *Let λ_0 be a nonreal eigenvalue of $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ on the boundary of $W(P)$. Then*

(i) *λ_0 is a semisimple eigenvalue of $P(\lambda)$,*
(ii) *λ_0 and $\overline{\lambda}_0$ have the same eigenspace, and*
(iii) *if λ_1 is an eigenvalue of $P(\lambda)$, $\lambda_1 \neq \lambda_0, \overline{\lambda}_0$, then the eigenspaces of λ_1 and λ_0 are orthogonal, i.e.,* $\mathrm{Ker}P(\lambda_1) \subseteq \mathrm{Ker}P(\lambda_0)^{\perp}$.

Proof. (i), (ii) As in the proof of Theorem 6, there exists a unitary matrix V such that

$$V^*P(\lambda)V = I\lambda^2 + (a_1I_k \oplus H_1)\lambda + a_0I_k \oplus H_0 \ \ (1 \le k \le n).$$

It is claimed that λ_0 does not belong to the spectrum of the $(n-k)\times(n-k)$ self-adjoint matrix polynomial

$$Q(\lambda) := I\lambda^2 + H_1\lambda + H_0.$$

To see this assume, on the contrary, that $\lambda_0 \in \sigma(Q)$. Then λ_0 is a boundary point of the numerical range $W(Q) \subseteq W(P)$. Applying Theorem 6, it follows that $a_0 + ia_1 \in \sigma(H_0 + iH_1) \cap \partial F(H_0 + iH_1)$, and so $a_0 + ia_1$ must be a normal eigenvalue of matrix $H_0 + iH_1$. Thus, there exists an $(n-k)\times(n-k)$ unitary matrix W such that

$$W^*(H_0 + iH_1)W = (a_0 + ia_1) \oplus (S_0 + iS_1),$$

where S_0 and S_1 are $(n-k-1)\times(n-k-1)$ Hermitian matrices. Hence, with V as above,

$$(I_k \oplus W)^*V^*(A_0 + iA_1)V(I_k \oplus W) = (a_0 + ia_1)I_{k+1} \oplus (S_0 + iS_1)$$

and, consequently,

$$(I_k \oplus W)^*V^*A_jV(I_k \oplus W) = a_jI_{k+1} \oplus S_j\,; \quad j = 0, 1.$$

So,

$$(I_k \oplus W)^*V^*P(\lambda_0)V(I_k \oplus W) = 0_{k+1} \oplus T_1$$

for an $(n-k-1)\times(n-k-1)$ matrix T_1, which is a contradiction because T_0 in (6) is nonsingular. Thus, $a_0 + ia_1 \notin \sigma(H_0 + iH_1)$ and $\lambda_0 \notin \sigma(Q)$.

Part (i) now follows because $\lambda_0 \notin \sigma(Q)$ and

$$V^*P(\lambda)V = (\lambda^2 + a_1\lambda + a_0)I_k \oplus Q(\lambda). \tag{7}$$

Thus, λ_0 is a nonreal zero of $\lambda^2 + a_1\lambda + a_0$. Part (ii) also follows from (7) because, in this representation, λ_0 and $\overline{\lambda}_0$ have the same eigenvectors.

(iii) Let λ_1 be an eigenvalue of $P(\lambda)$, for which $\lambda_1 \neq \lambda_0, \overline{\lambda}_0$, with a corresponding eigenvector $y \in \mathbb{C}^n$. Then $\lambda_1 \in \sigma(Q)$ in (7) and

$$P(\lambda_1)y = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} (\lambda_1^2 + a_1\lambda_1 + a_0)I_k & 0 \\ 0 & Q(\lambda_1) \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} y = 0,$$

where $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ and the image of V_1 is the eigenspace of λ_0 (which is necessarily semisimple). Since $\lambda_1^2 + a_1\lambda_1 + a_0 \neq 0$, $V_1^*y = 0$, i.e., the eigenspace of λ_1 is orthogonal to that of λ_0. ∎

Corollary 8. *Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ be k distinct nonreal eigenvalues of $P(\lambda)$ (with positive imaginary parts) on $\partial W(P)$, and let $m_1, m_2, \ldots, m_k$ be their respective algebraic multiplicities. Then there exists a unitary matrix V such that*

$$V^*P(\lambda)V = D(\lambda) \oplus Q(\lambda),$$

where $D(\lambda)$ is a matrix polynomial of size $m_1 + m_2 + \cdots + m_k$ of the form

$$D(\lambda) = (\lambda - \lambda_1)(\lambda - \overline{\lambda}_1)I_{m_1} \oplus \cdots \oplus (\lambda - \lambda_k)(\lambda - \overline{\lambda}_k)I_{m_k},$$

and $Q(\lambda)$ is a monic self-adjoint matrix polynomial such that

$$\sigma(Q) = \sigma(P)\backslash\{\lambda_1, \overline{\lambda}_1, \lambda_2, \overline{\lambda}_2, \ldots, \lambda_k, \overline{\lambda}_k\}.$$

5. Nondifferentiable boundary points. Consider a closed set $\Omega \subset \mathbb{C}$. A point $\xi \in \partial\Omega$ is called a *sharp point* (or *corner*) of Ω if there exist a real number $r > 0$ and angles φ_1, φ_2, and ψ_0 with $0 \le \varphi_2 - \varphi_1 \le \psi_0 < \pi$ such that

$$\varphi_1 \le \operatorname{Arg}(z - \xi) \le \varphi_2$$

for every $z \in \Omega \cap S(\xi, r)$, where $S(\xi, r)$ is the disc center ξ and radius r. The angles φ_1, φ_2 are called the *supporting angles* for the sharp point ξ. (See Langer, Markus, and Tretter [LMT]. This definition is equivalent to that appearing in [HJ].)

Let $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ monic self-adjoint matrix polynomial and $A = A_0 + iA_1$. Sharp points of $F(A)$ are eigenvalues of matrix A on $\partial F(A)$ (see [D] or [HJ]), and sharp points of $W(P)$ are eigenvalues of $P(\lambda)$ on $\partial W(P)$ (see [MP1]). Now consider, more generally, the nondifferentiable points of $\partial W(P)$. Note that since $F(A)$ is convex, the sharp points of $F(A)$ are the only nondifferentiable points of $\partial F(A)$.

Consider a point $a_0 + ia_1 \in \partial F(A)$ $(a_0, a_1 \in \mathbb{R})$ such that the equation $\lambda^2 + a_1\lambda + a_0 = 0$ has two nonreal roots λ_0 and $\overline{\lambda}_0$, where

$$\lambda_0 = \frac{-a_1 + i\sqrt{4a_0 - a_1^2}}{2}.$$

Assume also that $\mathcal{C} \subset \partial F(A)$ is a continuous and rectifiable curve with $a_0 + ia_1$ as an endpoint. For every point $\mu = a_0 + ia_1 + re^{i\varphi}$ ($r \in (0, +\infty)$ and $\varphi \in [0, 2\pi]$) on $\mathcal{C}$, close enough to $a_0 + ia_1$, the equation

$$\lambda^2 + (a_1 + r\sin\varphi)\lambda + a_0 + r\cos\varphi = 0$$

has two nonreal roots λ_μ and $\overline{\lambda}_\mu$, say.

If $\mu \in \mathcal{C}$ converges to $a_0 + ia_1$ along $\mathcal{C}$, then r converges to 0 and φ converges to an angle $\varphi_0 \in [0, 2\pi)$. Assume that $\varphi_0 \ne 0, \pi$. Then a calculation gives

$$\lim_{\mu \to a_0 + ia_1} \tan \operatorname{Arg}(\lambda_\mu - \lambda_0) = \frac{-2\cot\varphi_0 + a_1}{\sqrt{4a_0 - a_1^2}}. \tag{8}$$

First consider nonreal points on $\partial W(P)$.

THEOREM 9. *Let $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ monic self-adjoint quadratic matrix polynomial, let $\lambda_0 \in \partial W(P) \backslash \mathbb{R}$, and let $a_0 + ia_1 \in \partial F(A)$ such that $\lambda_0^2 + a_1\lambda_0 + a_0 = 0$. Then λ_0 is a nondifferentiable point of $\partial W(P)$ if and only if $a_0 + ia_1$ is a sharp point of $F(A)$.*

This theorem is illustrated in Example 3, where $\pm i$ are sharp points of $W(P)$ with corresponding sharp point at the point 1 on $F(A)$.

Proof. If we assume that $a_0 + ia_1$ is a sharp point of $F(A)$ of zero angle (i.e., $\varphi_1 = \varphi_2$), then $F(A)$ is a linear segment and $a_0 + ia_1$ is an endpoint of $F(A)$. Moreover, $F(A)$ has no interior points and, by Theorem 3, $W(P)$ also has no interior points. Since $a_0 + ia_1$ is an endpoint of $F(A)$ (by the continuous dependence of the roots of polynomials on their coefficients), the nonreal roots λ_0 and $\overline{\lambda}_0$ are endpoints of $W(P) \backslash \mathbb{R}$, and hence sharp points of $\partial W(P)$.

Now suppose that $a_0 + ia_1 \in \partial F(A)$ and is not a sharp point of zero angle. Then there exists a real number $r > 0$ such that for every $b_0 + ib_1 \in S(a_0 + ia_1, r)$ $(b_0, b_1, a_0, a_1 \in \mathbb{R})$, the equation $\lambda^2 + b_1\lambda + b_0 = 0$ has nonreal roots. Furthermore, the curve $\partial F(A) \cap S(a_0 + ia_1, r)$ is the union of two curves $\mathcal{C}_1$ and $\mathcal{C}_2$ such

that $\mathcal{C}_1 \cap \mathcal{C}_2 = \{a_0 + ia_1\}$. The point $a_0 + ia_1$ is either a sharp point or a differentiable point of $\mathcal{C}_1 \cup \mathcal{C}_2$.

Case (i). Let $a_0 + ia_1$ be a sharp point of $\mathcal{C}_1 \cup \mathcal{C}_2$, with supporting angles φ_1 and φ_2 such that $0 < \varphi_2 - \varphi_1 \leq \psi_0 < \pi$. Without lost of generality, we can assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed linear segments. It follows from (8) that if $\varphi_1, \varphi_2 \neq 0$ or π, then

$$\lim_{\mu_1 \to a_0 + ia_1} \tan \operatorname{Arg}(\lambda_1 - \lambda_0) = -\frac{2\cot\varphi_1 + a_1}{\sqrt{4a_0 - a_1^2}}$$

and

$$\lim_{\mu_2 \to a_0 + ia_1} \tan \operatorname{Arg}(\lambda_2 - \lambda_0) = -\frac{2\cot\varphi_2 + a_1}{\sqrt{4a_0 - a_1^2}},$$

where μ_1 and μ_2 are constrained to lie on $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and $\lambda_1, \lambda_2 \in \partial W(P)$ are the nonreal roots of the corresponding quadratic equations. Since $0 < \varphi_2 - \varphi_1 < \psi_0 \leq \pi$, we have $\cot\varphi_1 \neq \cot\varphi_2$ and hence

$$\left| \lim_{\mu_1 \to a_0 + ia_1} \operatorname{Arg}(\lambda_1 - \lambda_0) - \lim_{\mu_2 \to a_0 + ia_1} \operatorname{Arg}(\lambda_2 - \lambda_0) \right| \neq \pi.$$

Thus, λ_0 is a nondifferentiable point of $\partial W[P(\lambda)]$.

If one of the angles φ_1 and φ_2 is equal to 0 or π, then obviously $\cot\varphi_1 \neq \cot\varphi_2$, and we have the same conclusion.

Case (ii). Let $a_0 + ia_1$ be a differentiable point of $\partial F(A)$ and φ_0 be the angle made by the tangent to $\mathcal{C}_1 \cup \mathcal{C}_2$ at $a_0 + ia_1$ with the positive direction along the real axis.

If $\varphi_0 \neq 0, \pi$, then $\cot\varphi_0 = \cot(\varphi_0 + \pi)$ and, consequently, taking limits along $\mathcal{C}_1$ and $\mathcal{C}_2$,

$$\lim_{\mu_1 \to a_0 + ia_1} \tan \operatorname{Arg}(\lambda_1 - \lambda_0) = \lim_{\mu_2 \to a_0 + ia_1} \tan \operatorname{Arg}(\lambda_2 - \lambda_0).$$

Since $\varphi_0 \neq 0$ or π,

$$(\operatorname{Im}\mu_1 - a_1)(\operatorname{Im}\mu_2 - a_1) < 0 \Rightarrow \operatorname{Re}(\lambda_1 - \lambda_0)\operatorname{Re}(\lambda_2 - \lambda_0) < 0$$

$$\Rightarrow \lim_{\mu_1 \to a_0 + ia_1} \operatorname{Arg}(\lambda_1 - \lambda_0) \neq \lim_{\mu_2 \to a_0 + ia_1} \operatorname{Arg}(\lambda_2 - \lambda_0).$$

So

$$\left| \lim_{\mu_1 \to a_0 + ia_1} \operatorname{Arg}(\lambda_1 - \lambda_0) \; - \lim_{\mu_2 \to a_0 + ia_1} \operatorname{Arg}(\lambda_2 - \lambda_0) \right| = \pi,$$

and λ_0 is a differentiable point of $\partial W(P)$.

If $\varphi_0 = 0$ or π, then

$$\lim_{\mu_j \to a_0 + ia_1} \operatorname{Arg}(\lambda_j - \lambda_0) = \frac{\pi}{2} \text{ or } \frac{3\pi}{2}; \quad j = 1, 2.$$

Moreover, $(\operatorname{Re}\mu_1 - a_0)(\operatorname{Re}\mu_2 - a_0) < 0$ and $(\operatorname{Im}\mu_1 - a_1)(\operatorname{Im}\mu_2 - a_1) > 0$. Consequently, $\operatorname{Im}(\lambda_1 - \lambda_0)\operatorname{Im}(\lambda_2 - \lambda_0) < 0$. Thus,

$$\lim_{\mu_1 \to a_0 + ia_1} \operatorname{Arg}(\lambda_1 - \lambda_0) \; \neq \lim_{\mu_2 \to a_0 + ia_1} \operatorname{Arg}(\lambda_2 - \lambda_0)$$

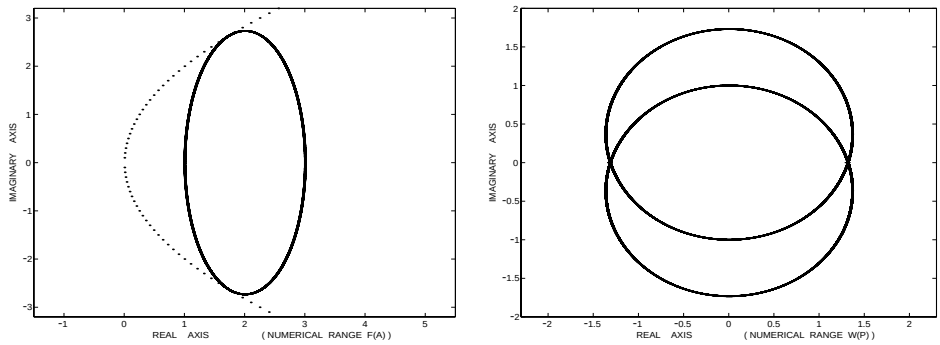


FIG. 4. *Nonorthogonal intersection of $\partial W(P)$ and the real axis.*

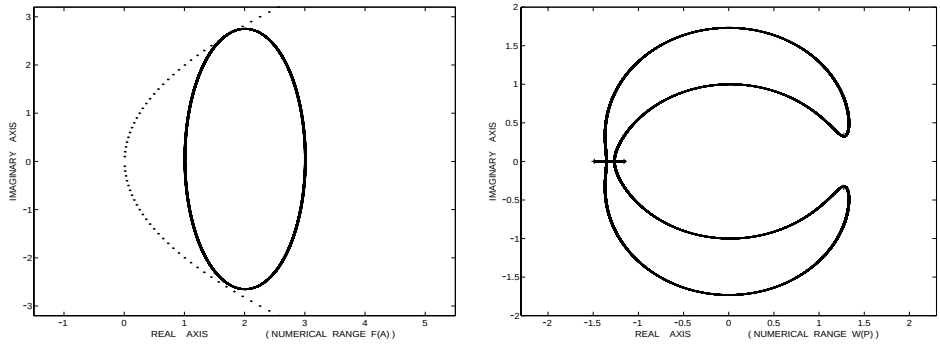


FIG. 5. *Orthogonal intersection of $\partial W(P)$ and the real axis.*

and λ_0 is a differentiable point of $\partial W(P)$. The proof is complete. □

Points of particular interest are the points where $\overline{\partial W(P)\backslash\mathbb{R}}$ intersects the real axis. The next two examples will be helpful.

Example 4. Take

$$A_1 = i\begin{bmatrix} 0 & \sqrt{4+\sqrt{12}} \\ -\sqrt{4+\sqrt{12}} & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and see Figure 4. Note that $F(A)$ just touches the parabola $\mathcal{D}$ and, consequently, $W(P) \cap \mathbb{R}$ consists of single points.

Example 5. This is a perturbation of Example 4. Take

$$A_1 = \begin{bmatrix} 0.05 & 2.7i \\ -2.7i & 0.05 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and see Figure 5.

The graphs of Figures 2 and 5 illustrate Theorem 10, and Figure 4 illustrates Theorem 11.

THEOREM 10. *Let $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ monic self-adjoint matrix polynomial, let $\lambda_0 \in \mathbb{R} \cap (\overline{\partial W(P)\backslash\mathbb{R}})$, and let $a_0 + ia_1 \in F(A)$ $(a_0, a_1 \in \mathbb{R}$ and $A = A_0 + iA_1)$ such that $\lambda_0^2 + a_1\lambda_0 + a_1 = 0$. Assume that the next two conditions hold:*

(i) *$a_0 + ia_1$ is not a differentiable point of $\partial F(A)$ such that its supporting line has an angle φ_0 with the real axis for which $\cot\varphi_0 = a_1/2$;*

(ii) *$F(A)$ is not a line segment at an angle φ_0 with the real axis for which* $\cot\varphi_0 = a_1/2$.

Then the curve $\overline{\partial W(P)\backslash\mathbb{R}}$ intersects the real axis orthogonally at λ_0.

Proof. Since $\lambda_0 \in (\overline{\partial W(P)\backslash\mathbb{R}}) \cap \mathbb{R}$, there exists a sequence

$$\{\lambda_k\}_{k\in\mathbb{N}} \in \partial W(P)\backslash\mathbb{R}$$

converging to λ_0 such that $\mathrm{Im}\lambda_k > 0$ ($k \in \mathbb{N}$). The sequence

$$\{|\lambda_k|^2 - i2\mathrm{Re}\lambda_k\}_{k\in\mathbb{N}}$$

converges to $a_0 + ia_1$ and, by Theorem 3, $|\lambda_k|^2 - i2\mathrm{Re}\lambda_k \in \partial F(A)$ for every $k \in \mathbb{N}$. So $a_0 + ia_1$ is a boundary point of $F(A)$. Moreover, λ_0 is a double root of equation $\lambda^2 + a_1\lambda + a_0 = 0$, i.e., $\lambda_0 = -a_1/2$.

For every $k \in \mathbb{N}$, we can write

$$|\lambda_k|^2 - i2\mathrm{Re}\lambda_k = |\lambda_{k+1}|^2 - i2\mathrm{Re}\lambda_{k+1} + r_k e^{i\varphi_k},$$

where $r_k > 0$ and $\varphi_k \in [0, 2\pi)$. Then, after some computations, it is found that

$$\begin{aligned}&\lim_{k\to+\infty} \tan \mathrm{Arg}(\lambda_k - \lambda_{k+1})\\ &= \lim_{k\to+\infty} \frac{-4\cos\varphi_k - 4\sin\varphi_k \mathrm{Re}\lambda_{k+1} + r_k \sin^2\varphi_k}{2\sin\varphi_k\left(\sqrt{|\lambda_k|^2 - (\mathrm{Re}\lambda_k)^2} + \sqrt{|\lambda_{k+1}|^2 - (\mathrm{Re}\lambda_{k+1})^2}\right)}.\end{aligned}$$

If the sequence $\{\varphi_k\}_{k\in\mathbb{N}} \in [0, 2\pi)$ converges to an angle $\varphi_0 \in (0, 2\pi)\backslash\pi$ such that $\cot\varphi_0 \neq -\lambda_0 = a_1/2$, then

$$\lim_{k\to+\infty} \tan \mathrm{Arg}(\lambda_k - \lambda_{k+1}) = \pm\infty. \tag{9}$$

Hence, the curve $\overline{\partial W(P)\backslash\mathbb{R}}$ intersects the real axis orthogonally at λ_0.

If the sequence $\{\varphi_k\}_{k\in\mathbb{N}} \in [0, 2\pi)$ converges to 0 or π, then

$$\lim_{k\to+\infty} \sin\varphi_k = 0, \qquad \lim_{k\to+\infty} \cos\varphi_k = \pm 1,$$

and (9) holds.

Note that if $a_0 + ia_1$ is a sharp point of $F(A)$ with supporting angles $\varphi_1 \neq \varphi_2$, then $\cot\varphi_1 \neq a_1/2$ or $\cot\varphi_2 \neq a_1/2$, and the proof is complete. ∎

If a point $\lambda_0 \in \mathbb{R} \cap (\overline{\partial W(P)\backslash\mathbb{R}})$ is an isolated point of $W(P) \cap \mathbb{R}$, then it follows from Theorem 10.15 of [GLR] that λ_0 is an eigenvalue of $P(\lambda)$. The next result holds for special points of this kind and casts some light on the exceptional cases of the preceding theorem.

THEOREM 11. *Let $P(\lambda)$ be a matrix polynomial as in Theorem* 10*, let $\lambda_0 \in \mathbb{R} \cap (\overline{\partial W(P)\backslash\mathbb{R}})$, and let $a_0 + ia_1 \in \partial F(A)$ ($a_0, a_1 \in \mathbb{R}$ and $A = A_0 + iA_1$) such that $\lambda_0^2 + a_1\lambda_0 + a_0 = 0$. If λ_0 is an isolated point of $W(P) \cap \mathbb{R}$ and $a_0 + ia_1$ is a differentiable point of $\partial F(A)$ such that its supporting line has an angle φ_0 with the real axis, then* $\cot\varphi_0 = a_1/2$.

Proof. Consider the parabola $\mathcal{D} = \{u + iv \in \mathbb{C} : u, v \in \mathbb{R},\ v^2 = 4u\}$. Then $v^2 - 4u < 0$ if and only if $u + iv$ lies "inside" the parabola $\mathcal{D}$. Observe that our hypotheses imply that $(\overline{\partial W(P)\backslash\mathbb{R}})$ is not empty. Consequently, there must be nonreal points in $W(P)$.

Obviously, $\lambda_0 \in \partial W(P) \cap \mathcal{D}$. Since λ_0 is an isolated point of $W(P) \cap \mathbb{R}$ and there are nonreal points in $W(P)$, there exists a real $r > 0$ such that

$$[\partial F(A) \cap S(a_0 + ia_1, r)] \backslash \{a_0 + ia_1\}$$

lies "inside" parabola $\mathcal{D}$. Moreover, $a_0 + ia_1$ is a differentiable point of $\partial F(A)$ and, consequently, the curves $\mathcal{D}$ and $\partial F(A)$ have a common supporting line at the point $a_0 + ia_1$. If a_0 and a_1 are nonzero, then $\sin\varphi_0$ and $\cos\varphi_0$ are also nonzero and

$$\tan\varphi_0 \cot\varphi_0 = 1 \Rightarrow \cot\varphi_0 = \pm a_1/2.$$

Furthermore, $\cot\varphi_0 > 0$ if and only if $a_1 > 0$. Thus, $\cot\varphi_0 = a_1/2$.

Finally, if $a_0 = a_1 = 0$, then $\varphi_0 = \pi/2$ and $\cot\varphi_0 = a_1/2 = 0$. □

6. Eigenvalue types. Let $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ monic self-adjoint matrix polynomial and let its numerical range be $W(P)$ as in (2). A real eigenvalue $\lambda_0 \in \sigma(P)$ is said to have *positive (negative) type* if $x^*P'(\lambda_0)x > 0$ ($x^*P'(\lambda_0)x < 0$) for all nonzero $x \in \mathrm{Ker}P(\lambda_0)$. Eigenvalues of either positive or negative type are said to have *definite type* and are necessarily semisimple. Real eigenvalues of $P(\lambda)$ which are not of definite type are said to be of *mixed type*. It is easy to verify that if $\lambda_0 \in \sigma(P) \cap \mathbb{R}$ is of mixed type, then there exists a nonzero vector $x_0 \in \mathrm{Ker}P(\lambda_0)$ such that $x_0^*P'(\lambda_0)x_0 = 0$.

With this notation, $P(\lambda)$ is said to be *hyperbolic* if, for all nonzero x, $x^*P(\lambda)x = 0$ has two distinct real roots. In particular, $W(P) \subset \mathbb{R}$. This class is well understood (see the monograph of Markus [M], for example) and all the eigenvalues are known to have definite type. However, the class of polynomials $P(\lambda)$ with all eigenvalues real and of definite type is wider, and, in general, we do not have the inclusion $W(P) \subset \mathbb{R}$. Such polynomials are said to be *quasi-hyperbolic*. For example, if all eigenvalues are real and distinct, then they are necessarily definite and the system is quasi-hyperbolic. This notion was introduced in [L1] (see also [LMM]), and a sufficient condition for $P(\lambda)$ to be quasi-hyperbolic can be formulated in terms of $W(P)$.

First recall the decomposition of (5) and assume that $\mathcal{S} \neq \emptyset$.

PROPOSITION 12. *If λ_0 is an eigenvalue of $P(\lambda)$ in $W(P)\backslash\mathcal{S}$, then λ_0 has definite type.*

Proof. If λ_0 is not of definite type, there exists a nonzero $x \in \mathbb{C}^n$ such that $x^*P'(\lambda_0)x = 0$. This implies that $x^*P(\lambda_0)x = 0$ has a double root and, hence, that $a_1 + ia_0 \in \mathcal{D} \cap F(A)$. It follows from Proposition 4 that $\lambda_0 = -a_1/2 \in \mathbb{R} \cap \mathcal{S}$, and this contradicts the disjointness of the union in (5). □

COROLLARY 13. *If all eigenvalues of $P(\lambda)$ are real and those in $\mathcal{S}$ have definite type, then $P(\lambda)$ is quasi-hyperbolic.*

Finally, we estimate the length of Jordan chains corresponding to eigenvalues of $P(\lambda)$ on the boundary of $W(P)$.

THEOREM 14. *Let $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ be an $n \times n$ monic self-adjoint matrix polynomial and let λ_0 be an eigenvalue of $P(\lambda)$ on $\partial W(P)$.*

(i) *If λ_0 is not real or $\lambda_0 \in W(P)\backslash\mathcal{S}$, then λ_0 is a semisimple eigenvalue of $P(\lambda)$.*

(ii) *If $\lambda_0 \in \mathbb{R} \cap \overline{\partial W(P)\backslash\mathbb{R}}$ and it satisfies the conditions of Theorem 10, then the length of the corresponding Jordan chains is not greater than 2.*

Proof. (i) See Theorem 7 and Proposition 12.

(ii) Obviously, there exists a cone

$$\mathcal{L} = \{z \in \mathbb{C} : \varphi_1 \leq \mathrm{Arg}(z - \lambda_0) \leq \varphi_2,\ 0 < \varphi_2 - \varphi_1 \leq \psi_0 < \pi\}$$

and a real number $r > 0$ such that

$$\mathcal{L} \cap S(\lambda_0, r) \cap W(P) = \{\lambda_0\}.$$

Thus, the result follows from Theorem 2 in [MM]. □

7. Damped vibrating systems. This section consists of some remarks putting this discussion in the context of known results for damped vibrating systems (from Chapter 13 of [GLR], for example).

For our purposes, a "damped vibrating system" means a matrix polynomial of the form (1) with $A_2 > 0$, $A_1 \geq 0$, and $A_0 \geq 0$. Without loss of generality, $P(\lambda)$ may be assumed to be monic. Thus, in this section $P(\lambda) = I\lambda^2 + A_1\lambda + A_0$ with $A_1 \geq 0$ and $A_0 \geq 0$. It is easily seen that, for matrix polynomials of this kind, the spectrum is always in the closed left half of the complex plane. Similarly, the numerical range, $W(P)$ is also in the closed left half plane.

A damped vibrating system is said to be *weakly damped* if, for all nonzero $x \in \mathbb{C}^n$,

$$(x^*A_1x)^2 < 4(x^*x)(x^*A_0x). \tag{10}$$

Clearly, this is just the case in which $F(A)$ lies "inside" the parabola $\mathcal{D}$ of section 3. Furthermore, $W(P)$ has two components, one in the upper half plane, and the other its reflection in the lower half plane. Of course, there are no real eigenvalues. Recall that there is a factorization

$$P(\lambda) = (I\lambda - Z^*)(I\lambda - Z),$$

with the spectrum of Z (of Z^*) inside one of the components of $W(P)$. It follows that for $\lambda \in \mathbb{R}$, $P(\lambda) > 0$.

Note also that for a weakly damped system $P(\lambda)$, all *principal subsystems* (i.e., determined by a principal submatrix of $P(\lambda)$) are also weakly damped. This is simply because, for a principal subsystem, (10) is required to hold on a subset of nonzero vectors of $\mathbb{C}^n$. Thus the numerical range of a principal subsystem is a subset of $W(P)$.

A damped vibrating system is said to be *overdamped* if the inequality above is reversed. Then $F(A)$ lies "outside" the parabola $\mathcal{D}$ and $P(\lambda)$ is hyperbolic. Hence, $W(P) \subset \mathbb{R}$ and all eigenvalues are real. In this case, a factorization

$$P(\lambda) = (I\lambda - Y)(I\lambda - Z)$$

is possible with all eigenvalues of Z strictly greater than all those of Y. (More generally, a self-adjoint polynomial of the form (1) can always be written as a product of linear factors, but the separation of eigenvalues of $P(\lambda)$ between the two factors may be complicated (see Chapter 11 of [GLR]).) Note also that overdamped systems are *strongly stable* in the sense that all neighboring systems, with the same symmetries, are also overdamped. Finally, observe that principal subsystems of overdamped systems are also overdamped so that their numerical ranges are also nested.

Frequently, systems are neither weakly damped nor overdamped. Example 6 is of this kind.

Example 6. Take

$$A_1 = \begin{bmatrix} 6 & 1 & 1 & 0 & 0 & 0 \\ 1 & 5 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix}, \qquad A_0 = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

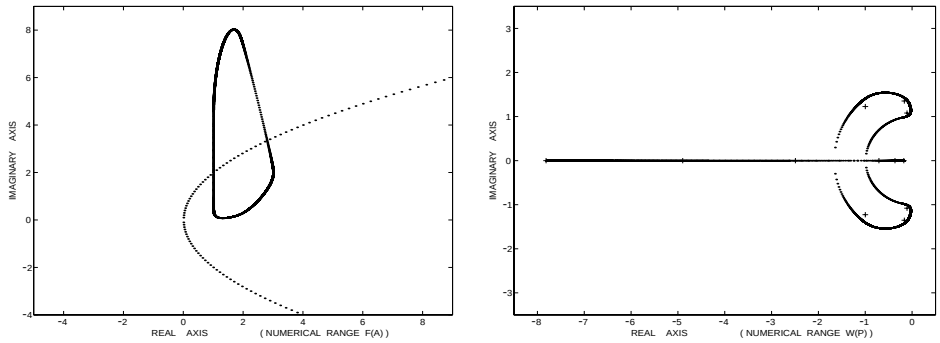


FIG. 6. *A damped vibrating system.*

It can be shown that $A_1 > 0$. $F(A)$ and $W(P)$ are sketched in Figure 6.

A convenient assumption often made in applications is known as *proportional damping*. After reduction to the case $A_2 = I$, this simply means that A_1 is a real linear combination of I and A_0. In this case, $F(A)$ is just a linear segment and $W(P)$ is made up of segments of the real axis and/or nonreal arcs.

More generally, if it is assumed that A_1 and A_0 can be simultaneously diagonalized with a unitary similarity, then $A = A_1 + iA_0$ is a normal matrix and $F(A)$ is a (possibly degenerate) polygon. The hypotheses are now equivalent to assuming that $P(\lambda) = A_2\lambda^2 + A_1\lambda + A_0$ can be transformed to a diagonal matrix polynomial by congruence.

Finally, if the condition $A_0 \geq 0$ is relaxed to admit any Hermitian A_0, then it is easy to see that any eigenvalue of $P(\lambda)$ in the open right half plane is necessarily real and of definite type. First, it is clear that any nonreal point of $W(P)$ must be in the closed left half plane, and it only remains to apply Proposition 12. It is also true that the number of eigenvalues in the open right half plane is just the number of negative eigenvalues of A_0. (Results of this kind are known in more general settings; see [AP] and references there, for example.)

8. Systems with variable damping. In the paper [GLR2] a study has been made of parameter-dependent systems of the form

$$P_s(\lambda) = I\lambda^2 + sA_1\lambda + A_0, \quad s > 0.$$

In particular, when $A_1 > 0$, the transition from a weakly to a strongly damped system as s increases was considered. The s-dependent numerical range provides an interesting view of this process. There is a family of associate matrices $A(s) = A_0 + isA_1$ and, clearly, $s_1 > s_2$ implies that $F(A(s_1))$ lies "higher" than $F(A(s_2))$. Now $F(A(s))$ is always contained in a vertical strip with boundaries defined by the extreme eigenvalues of A_0. As s increases $F(A(s))$ moves "upward" (and also changes shape).

It was shown by example in [GLR2] that the number of real eigenvalues of $P_s(\lambda)$ does not necessarily increase with s. Thus, the curves $\lambda(s)$ in the complex λ-plane have preimages in the plane of $F(A(s))$ which may cross the curve $\mathcal{D}$ more than once.

In contrast, it is shown in [FGK] that, as s increases, once the first real eigenvalue occurs, say at $s = s_0$ (so that $F(A(s_0)) \cap \mathcal{D} \neq \emptyset$), then there is at least one real eigenvalue for *all* $s > s_0$.

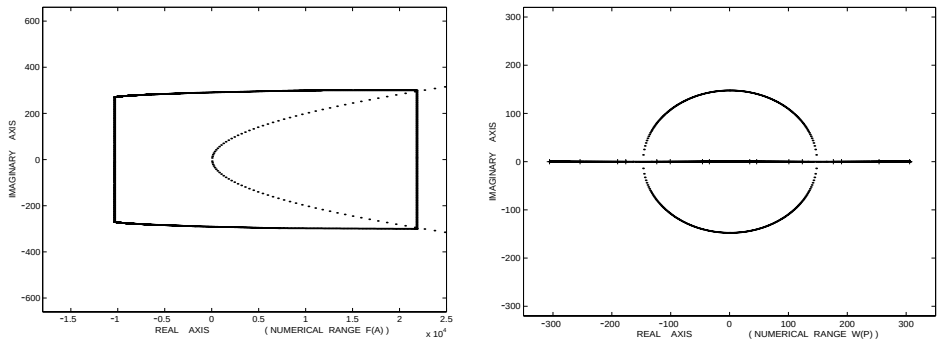


FIG. 7. *A gyroscopic system.*

9. Gyroscopic systems. As with damped systems, a gyroscopic system is, for us, a quadratic matrix function. Physical models first lead to consideration of a function of the form

$$Q(\mu) = M\mu^2 + G\mu + K, \tag{11}$$

where M, G, and K are *real* $n \times n$ matrices and

$$M > 0, \quad G^T = -G, \quad K^T = K.$$

As usual, the reduction to the case $M = I$ is straightforward. To include these systems in our notations, introduce a new eigenvalue parameter $\lambda = i\mu$ and the matrix polynomial

$$P(\lambda) = I\lambda^2 + iG\lambda - K =: I\lambda^2 + A_1\lambda + A_0.$$

Because $(iG)^* = iG$, $P(\lambda)$ is a self-adjoint matrix polynomial, and our ideas can be applied to this formulation. It is well known that $\sigma(Q)$, and hence $\sigma(P)$, has Hamiltonian symmetry, i.e., eigenvalues are distributed in the complex plane symmetrically with respect to *both* the real and imaginary axes. It is easily seen that $W(P)$ enjoys this same symmetry. Notice that Examples 1–4 are all gyroscopic systems.

Under the conditions of Corollary 13 (in particular, when $\sigma(P) \subset W(P)\backslash\mathcal{S}$) all the eigenvalues of $P(\lambda)$ are definite and the system (11) is strongly stable (see [L2], for example). In Example 4, all the eigenvalues are real, but the system is not strongly stable. As in Example 5, there are perturbations which produce nonreal eigenvalues.

Example 7. An eight-degrees-of-freedom modal approximation to a mechanical system (see [Mr]) yields the coefficient matrices

$$M = \begin{bmatrix} M_1 & 0 \\ 0 & M_1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & -G_1 \\ G_1 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} K_1 & 0 \\ 0 & K_1 \end{bmatrix},$$

where $M_1 = \text{diag}[0.2 \;\; 0.8 \;\; 0.2 \;\; 1/9]$, $G_1 = 150\text{diag}[0.4 \;\; 1.6 \;\; 0.4 \;\; 7/36]$, and

$$K_1 = \begin{bmatrix} -2800 & -1200 & 0 & -1200 \\ -1200 & -15600 & -1200 & 0 \\ 0 & -1200 & -2800 & 1200 \\ -1200 & 0 & 1200 & 561.48 \end{bmatrix}.$$

Since the eigenvalues are real and distinct (see Figure 7), the system is quasi-hyperbolic (see also Corollary 13), although this cannot be deduced directly from properties of $W(P)$ itself.

REFERENCES

[AP] V. Adamyan and V. Pivovarchik, *On the spectra of some classes of quadratic operator pencils*, in Contributions to Operator Theory in Spaces with an Indefinite Metric, Oper. Theory Adv. Appl. 106, Birkhäuser, Basel, 1998, pp. 23–36.

[AT] Y.-H. Au-Yeung and N.-K. Tsing, *An extension of the Hausdorff-Toeplitz Theorem on the numerical range*, Proc. Amer. Math. Soc., 89 (1983), pp. 215–218.

[D] W.F. Donoghue, *On the numerical range of a bounded operator*, Michigan Math. J., 4 (1957), pp. 261–263.

[FGK] P. Freitas, M. Grinfeld, and P. A. Knight, *Stability for finite-dimensional systems with indefinite damping*, Adv. Math. Sci. Appl., 7 (1997), pp. 435–446.

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

[GLR2] I. Gohberg, P. Lancaster, and L. Rodman, *Quadratic matrix polynomials with a parameter*, Adv. in Appl. Math., 7 (1986), pp. 253–281.

[HJ] R. Horn and C. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[L1] P. Lancaster, *Quadratic eigenvalue problems*, Linear Algebra Appl., 150 (1991), pp. 499–506.

[L2] P. Lancaster, *Strongly stable gyroscopic systems*, Electronic J. Linear Algebra, 5 (1999), pp. 53–66.

[LMM] P. Lancaster, A. S. Markus, and V. I. Matsaev, *Perturbations of G-self-adjoint operators and operator polynomials with real spectrum*, in Recent Developments in Operator Theory and Its Applications, Oper. Theory Adv. Appl. 87, Birkhäuser, Basel, 1996, pp. 207–221.

[LMT] H. Langer, A. S. Markus, and C. Tretter, *Corners of numerical ranges*, in Recent Advances in Operator Theory, Oper. Theory. Adv. Appl. 124, Birkhäuser, Basel, 2001, pp. 385–400.

[LR] C.-K. Li and L. Rodman, *Numerical range of matrix polynomials*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1256–1265.

[M] A.S. Markus, *Introduction to the Spectral Theory of Polynomial Operator Pencils*, Transl. Math. Monogr. 71, AMS, Providence, RI, 1988.

[Mr] P.C. Müller, *Stabilität und Matrizen*, Springer-Verlag, Berlin, 1977.

[MM] A.S. Markus and V.I. Matsaev, *Some estimates for the resolvent and for the lengths of Jordan chains of an analytic operator function*, in Recent Advances in Operator Theory, Oper. Theory Adv. Appl. 124, Birkhäuser, Basel, 2001, pp. 473–479.

[MP1] J. Maroulas and P. Psarrakos, *The boundary of numerical range of matrix polynomials*, Linear Algebra Appl., 267 (1997), pp. 101–111.

[MP2] J. Maroulas and P. Psarrakos, *A connection between numerical ranges of selfadjoint matrix polynomials*, Linear and Multilinear Algebra, 44 (1998), pp. 327–340.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 632–645
© 2001 Society for Industrial and Applied Mathematics

ON AN ITERATION METHOD FOR SOLVING A CLASS OF NONLINEAR MATRIX EQUATIONS*

SALAH M. EL-SAYED† AND ANDRÉ C. M. RAN‡

Abstract. This paper treats a set of equations of the form $X + A^\star \mathcal{F}(X)A = Q$, where $\mathcal{F}$ maps positive definite matrices either into positive definite matrices or into negative definite matrices, and satisfies some monotonicity property. Here A is arbitrary and Q is a positive definite matrix. It is shown that under some conditions an iteration method converges to a positive definite solution. An estimate for the rate of convergence is given under additional conditions, and some numerical results are given. Special cases are considered, which cover also particular cases of the discrete algebraic Riccati equation.

Key words. matrix equation, iteration methods, operator monotone functions, hermitian positive definite matrices

AMS subject classifications. 15A24, 47A62, 58F08

PII. S0895479899345571

1. Introduction.

Let $\mathcal{P}(n)$ denote the set of $n \times n$ positive semidefinite matrices. We consider the following class of nonlinear matrix equations

$$X + A^\star \mathcal{F}(X)A = Q, \tag{1.1}$$

where $\mathcal{F}(\cdot) : \mathcal{P}(n) \to \mathcal{P}(n)$ is either monotone (meaning that $0 \le X \le Y$ implies that $\mathcal{F}(X) \le \mathcal{F}(Y)$) or antimonotone (meaning that $0 \le X \le Y$ implies that $\mathcal{F}(X) \ge \mathcal{F}(Y)$). In particular, we shall be interested in the case where $\mathcal{F}(X)$ is generated by a function from $[0, \infty)$ to $[0, \infty)$ which is either operator monotone or operator antimonotone. For example, $\mathcal{F}(x) = x^r$ is operator monotone for $0 < r \le 1$, while $\mathcal{F}(x) = x^{-1}$ is operator antimonotone (see, e.g., [2], where a thorough study of operator monotone functions is presented). Also, in (1.1) A is an arbitrary $n \times n$ matrix, and Q and X are in $\mathcal{P}(n)$. We also shall consider the case where $\mathcal{F}(\cdot) : \mathcal{P}(n) \to -\mathcal{P}(n)$ and is antimonotone.

In the case where $\mathcal{F}$ is monotone we shall often assume that $\mathcal{F}$, A, and Q satisfy the additional requirement that

$$A^\star \mathcal{F}(Q)A < Q. \tag{1.2}$$

This type of nonlinear matrix equation often arises in the analysis of ladder networks, dynamic programming, control theory, stochastic filtering, statistics and in many applications [1].

Several authors [1, 3, 4, 5, 12, 13, 17, 16] have considered such a nonlinear matrix equation problem. Compare also [15], where a different type of nonlinear matrix equation was studied. Anderson, Morley, and Trapp [1] discussed the existence of

*Received by the editors March 3, 1999; accepted for publication (in revised form) by V. Mehrmann May 30, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/simax/23-3/34557.html

†Department of Mathematics, Faculty of Science, Benha University, Benha 13518, Egypt (mselsayed@frcu.eun.eg). Current address: Department of Mathematics, Scientific Departments, Education College for Girls, Al-Montazah, Buradah, Al-Qassim, Kingdom of Saudi Arabia.

‡Divisie Wiskunde en Informatica, Faculteit Exacte Wetenschappen, Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands (ran@cs.vu.nl). The work of this author was partially supported by the NATO grant CRG 960700.

the positive solution to the matrix equation (1.1) when $\mathcal{F}(X) = X^{-1}$ and with the right-hand side being an arbitrary matrix, while Engwerda, Ran, and Rijkeboer [3] established and proved theorems for the necessary and sufficient conditions of existence of a positive definite solution of the matrix equation as in [1]. They discussed both the real and complex cases and established recursive algorithms to compute the largest and smallest solutions of the equation. Also Engwerda [4] proved the existence of the positive definite solution of the real matrix equation (1.1) when the right-hand side is the identity matrix, and he also found an algorithm to calculate the solution. In [12] the first author obtained necessary and sufficient conditions for existence of a positive definite solution of the matrix equation (1.1) with several forms of $\mathcal{F}(\cdot)$, without any conditions on the equation. In [8] some properties of a positive definite solution of the equation for $\mathcal{F}(X) = X^{-2}$ and with A normal were investigated. In [17, 16] several numerical algorithms for finding solutions for the case $\mathcal{F}(X) = X^{-1}$ were proposed.

The goal of this paper is to discuss the matrix equation (1.1), with general function $\mathcal{F}(X)$ which is either monotone or antimonotone. Closely connected to this equation is the map $\mathcal{G}(X) = Q - A^\star \mathcal{F}(X)A$. We shall also be interested in the dynamics of the map $\mathcal{G}$. We use iterative methods to obtain numerically a solution of the nonlinear matrix equation (1.1) under some additional conditions. In the case where $\mathcal{F} : [0, \infty) \to [0, \infty)$ is operator monotone Banach's fixed point theorem is a basic theorem to establish the existence of a positive definite solution and to obtain the rate of convergence for the sequence which is generated by an iteration method. Some numerical examples are given. For antimonotone functions a different method for proving necessary and sufficient conditions for existence of a positive definite solution is given.

The paper is organized as follows. In section 2, we discuss the monotone case. Some properties of $\mathcal{G}$ are studied. Under some conditions on $\mathcal{F}$ we obtain the rate of convergence of the iterative sequence of approximate solutions and a stopping criterion. Section 3 discusses the antimonotone case. Section 4 illustrates the performance of the method with some numerical examples and contains some remarks on the matrix equation (1.1) and on the results in the preceding sections. Section 5 discusses the equation (1.1) for maps $\mathcal{F}$ that map positive definite matrices into negative definite matrices and are antimonotone. An application to the discrete algebraic Riccati equation is given.

The following notations are used throughout the rest of the paper. The notation $A \geq 0$ ($A > 0$) means that A is positive semidefinite (positive definite), $A^\star$ denotes the complex conjugate transpose of A, and I is the identity matrix. Moreover, $A \geq B$ ($A > B$) is used as a different notation for $A - B \geq 0$ ($A - B > 0$). This induces a partial ordering on the hermitian matrices. When we say that a hermitian matrix is the smallest (largest) in some set, then this is always meant with respect to the partial ordering induced in this way. We denote by ρ the largest eigenvalue of $A^\star A$. The norm used in this paper is the spectral norm of the matrix A, i.e., $\| A \| = \sqrt{\rho(AA^\star)}$, unless otherwise noted.

Associated to (1.1) is the map $\mathcal{G}$ defined by $\mathcal{G}(X) = Q - A^\star \mathcal{F}(X)A$, which will play an important role in our analysis. Observe that a solution of (1.1) is a fixed point of $\mathcal{G}$. By $\mathcal{G}^2(X)$ we denote $\mathcal{G}(\mathcal{G}(X))$, and by $\mathcal{G}^j(X)$ the jth iterate of $\mathcal{G}$ on X.

2. The monotone case. We start with some preliminary results. We will establish and prove some theorems concerning the dynamics of $\mathcal{G}$ first. Then we shall apply Banach's fixed point theorem to obtain a positive definite solution of (1.1) under some

restrictions on the map $\mathcal{F}$. Also we obtain some relations between the eigenvalues of the solution of (1.1) and the eigenvalues of the matrix A.

LEMMA 2.1. *Let $\mathcal{F}$ be monotone on $\mathcal{P}(n)$. Assume* (1.2) *holds. If X is a positive semidefinite solution of* (1.1), *then*

$$Q \geq X \geq Q - A^\star \mathcal{F}(Q)A = \mathcal{G}(Q). \tag{2.1}$$

In particular, X is positive definite.

Proof. From the matrix equation (1.1), we get immediately $0 \leq X \leq Q$ and $A^\star \mathcal{F}(X)A \leq Q$. Since X is a positive semidefinite solution of (1.1), by the monotonicity of $\mathcal{F}$ we have that $\mathcal{F}(X) \leq \mathcal{F}(Q)$. Therefore, $0 < Q - A^\star \mathcal{F}(Q)A \leq Q - A^\star \mathcal{F}(X)A = X$. □

First we show that condition (1.2) implies the existence of a fixed point of $\mathcal{G}^2$, and so implies either a periodic orbit of period 2 of the map $\mathcal{G}$ or a fixed point of $\mathcal{G}$, and gives information concerning the location of periodic orbits and, in particular, of fixed points of $\mathcal{G}$.

THEOREM 2.2. *If $\mathcal{F}$ is monotone on $\mathcal{P}(n)$ and* (1.2) *holds, then the following hold true.*

(i) *For any positive definite matrix X for which $\mathcal{G}(X)$ is positive definite we have $\mathcal{G}(Q) \leq \mathcal{G}^2(X) \leq Q$, and the set $\{X = X^\star \mid \mathcal{G}(Q) \leq X \leq Q\}$ is mapped into itself by $\mathcal{G}$.*

(ii) *There always exists either a periodic orbit of period 2 of the map $\mathcal{G}$ or a fixed point of $\mathcal{G}$. The sequence of matrices $\{\mathcal{G}^{(2j)}(Q)\}_{j=0}^{\infty}$ is a decreasing sequence of positive definite matrices converging to a positive definite matrix X_∞, and the sequence of matrices $\{\mathcal{G}^{(2j+1)}(Q)\}_{j=0}^{\infty}$ is an increasing sequence of positive definite matrices converging to a positive definite matrix $X_{-\infty}$, and the matrices $X_\infty, X_{-\infty}$ form either a periodic orbit of $\mathcal{G}$ of period 2, or $X_\infty = X_{-\infty}$, in which case it is a fixed point of $\mathcal{G}$, and hence a solution of* (1.1).

(iii) *Moreover, $\mathcal{G}$ maps the set $\{X = X^\star \mid X_{-\infty} \leq X \leq X_\infty\}$ into itself, and any periodic orbit of $\mathcal{G}$ is contained in this set. In particular, any solution of* (1.1) *is in between $X_{-\infty}$ and X_∞, and if $X_{-\infty} = X_\infty$, then there is a unique positive definite solution.*

(iv) *In the case where $X_{-\infty} = X_\infty$ this matrix is the global attractor for the map $\mathcal{G}$ in the following sense: for any positive definite X for which $\mathcal{G}(X)$ is positive definite as well, we have $\lim_{j\to\infty} \mathcal{G}^j(X) = X_\infty$.*

(v) *In the case where $X_{-\infty} \neq X_\infty$ the following holds: if $X \leq X_{-\infty}$, then the orbit of X under $\mathcal{G}$ converges to the periodic orbit $X_{-\infty}, X_\infty$ in the sense that $\lim_{j\to\infty} \mathcal{G}^{2j-1}(X) = X_\infty$, and $\lim_{j\to\infty} \mathcal{G}^{2j}(X) = X_{-\infty}$. If $X \geq X_\infty$ and $\mathcal{G}(X)$ is positive definite, then the orbit of X under $\mathcal{G}$ converges to the periodic orbit $X_{-\infty}, X_\infty$ in the sense that $\lim_{j\to\infty} \mathcal{G}^{2j-1}(X) = X_{-\infty}$, and $\lim_{j\to\infty} \mathcal{G}^{2j}(X) = X_\infty$.*

Proof. Observe that the set $[Q - A^\star \mathcal{F}(Q)A,\ Q] := \{X = X^\star | Q - A^\star \mathcal{F}(Q)A \leq X \leq Q\}$ is a compact and hence complete metric space. Put $\mathcal{G}(X) = Q - A^\star \mathcal{F}(X)A$. We observe that $\mathcal{G}$ maps the set $[Q - A^\star \mathcal{F}(Q)A,\ Q] = [\mathcal{G}(Q), Q]$ into itself.

Indeed, let Y be a hermitian matrix in $[Q - A^\star \mathcal{F}(Q)A,\ Q]$, i.e.,

$$Q - A^\star \mathcal{F}(Q)A \leq Y \leq Q;$$

then $\mathcal{F}(Y) \leq \mathcal{F}(Q)$ by the monotonicity, so

$$\mathcal{G}(Y) = Q - A^\star \mathcal{F}(Y)A \geq Q - A^\star \mathcal{F}(Q)A,$$

and as $\mathcal{F}(Y) \geq 0$, we have

$$\mathcal{G}(Y) = Q - A^\star \mathcal{F}(Y) A \leq Q.$$

That is,

$$Q - A^\star \mathcal{F}(Q) A \leq \mathcal{G}(Y) \leq Q.$$

Next we show that $\mathcal{G}$ is antimonotone on the set $[\mathcal{G}(Q), Q]$, so on this set $\mathcal{G}^2$ is monotone. Indeed, it is easily seen that $X \leq Y$ implies that

$$\mathcal{G}(X) - \mathcal{G}(Y) = A^*(\mathcal{F}(Y) - \mathcal{F}(X))A \geq 0,$$

so $\mathcal{G}$ is antimonotone. It follows that $\mathcal{G}^2$ is monotone. Now let X be any positive definite matrix. Then clearly $\mathcal{G}(X) \leq Q$. As $\mathcal{G}$ is antimonotone, this implies that $\mathcal{G}(Q) \leq \mathcal{G}^2(X) \leq Q$.

Also, for all j we have that $\mathcal{G}(Q) \leq \mathcal{G}^j(Q) \leq Q$. Taking $j = 2$ first, we see that $\mathcal{G}^2(Q) \leq Q$. Then applying $\mathcal{G}^2$ repeatedly, we see that the monotonicity of $\mathcal{G}^2$ on this set implies that the sequence $\{\mathcal{G}^{(2j)}(Q)\}_{j=0}^{\infty}$ is a decreasing sequence of positive definite matrices that is bounded below by the positive definite matrix $\mathcal{G}(Q)$. Hence it converges to a positive definite matrix X_∞, which is a fixed point of $\mathcal{G}^2$. Hence $X_\infty, \mathcal{G}(X_\infty)$ is a periodic orbit of $\mathcal{G}$ of period 2 or a fixed point.

Next take $j = 3$; then we see that $\mathcal{G}(Q) \leq \mathcal{G}^3(Q)$. Again, applying $\mathcal{G}^2$ repeatedly, the monotonicity of $\mathcal{G}^2$ on $[\mathcal{G}(Q), Q]$ implies that the sequence of matrices $\{\mathcal{G}^{(2j+1)}(Q)\}_{j=0}^{\infty}$ is an increasing sequence of positive definite matrices which is bounded above by Q. Hence this sequence has a limit $X_{-\infty}$, which is a fixed point of $\mathcal{G}^2$. Hence $X_{-\infty}, \mathcal{G}(X_{-\infty})$ is a periodic orbit of $\mathcal{G}$ of period 2 or a fixed point.

Now we shall show that $\mathcal{G}$ maps the set $[X_{-\infty}, X_\infty]$ into itself. First observe that $\mathcal{G}(Q) \leq X_\infty$, and thus, applying $\mathcal{G}^2$ repeatedly, we see that $\mathcal{G}^{(2j+1)}(Q) \leq X_\infty$ for all j. It follows that $X_{-\infty} \leq X_\infty$. Let $X_{-\infty} \leq X \leq X_\infty$. Then for all j we have $\mathcal{G}^{(2j+1)}(Q) \leq X \leq \mathcal{G}^{(2j)}(Q)$. Applying $\mathcal{G}$ and using the fact that $\mathcal{G}$ is antimonotone, we see that $\mathcal{G}^{(2j+1)}(Q) \leq \mathcal{G}(X) \leq \mathcal{G}^{(2j+2)}(Q)$. Letting $j \to \infty$ we see that $X_{-\infty} \leq \mathcal{G}(X) \leq X_\infty$.

To show that $\mathcal{G}(X_\infty) = X_{-\infty}$, observe that $X_{-\infty} \leq \mathcal{G}(X_{\pm\infty}) \leq X_\infty$. Now apply $\mathcal{G}$ to this to get $\mathcal{G}(X_\infty) \leq \mathcal{G}^2(X_{-\infty}) = X_{-\infty}$. So, $\mathcal{G}(X_\infty) = X_{-\infty}$ and $\mathcal{G}(X_{-\infty}) = X_\infty$.

Next, let $\{X_j\}_{j=1}^{p}$ be a periodic orbit of $\mathcal{G}$ of period p. Thus X_j is a fixed point of $\mathcal{G}^p$. Obviously, by part (i) we have $\mathcal{G}(Q) \leq X_j \leq Q$. Observe that $\mathcal{G}^{2p}$ is monotonic. Applying $\mathcal{G}^{2p}$ repeatedly, we readily see that $\mathcal{G}^{2kp+1}(Q) \leq X_j \leq \mathcal{G}^{2kp}(Q)$ for all $k = 0, 1, \ldots$. Hence, letting $k \to \infty$, we see that $X_{-\infty} \leq X_j \leq X_\infty$.

Finally, we shall prove (iv) and (v). Take a positive matrix X such that $\mathcal{G}(X)$ is positive definite as well. Recall that $\mathcal{G}(Q) \leq \mathcal{G}^2(X) \leq Q$. From the antimonotonicity of $\mathcal{G}$ we get that $\mathcal{G}(Q) \leq \mathcal{G}^3(X) \leq \mathcal{G}^2(Q)$. As $\mathcal{G}^2$ is monotone we deduce from these inequalities that

$$\begin{aligned} \mathcal{G}^{(2j-1)}(Q) &\leq \mathcal{G}^{2j}(X) \leq \mathcal{G}^{(2j-2)}(Q), \\ \mathcal{G}^{(2j-1)}(Q) &\leq \mathcal{G}^{(2j+1)}(X) \leq \mathcal{G}^{2j}(Q). \end{aligned}$$

It follows that if $X_{-\infty} = X_\infty$, then $\mathcal{G}^j(X)$ converges to X_∞ as well.

In a similar way, if $X \leq X_{-\infty}$, then $\mathcal{G}(X) \geq \mathcal{G}(X_\infty) = X_\infty$. (In particular, for such X we have that $\mathcal{G}(X)$ is positive definite.) Then it follows that $\mathcal{G}(Q) \leq \mathcal{G}^2(X) \leq X_{-\infty}$. Now use the fact that $\mathcal{G}^2$ is monotone to see that $\mathcal{G}^{2j+1}(Q) \leq \mathcal{G}^{2j+2} \leq X_{-\infty}$.

As the left-hand side in these inequalities converges to $X_{-\infty}$ we see that $\mathcal{G}^{2j}(X)$ converges to $X_{-\infty}$, and hence $\mathcal{G}^{2j-1}(X)$ converges to $\mathcal{G}(X_{-\infty}) = X_\infty$.

Likewise, if $X \geq X_\infty$ and $\mathcal{G}(X)$ is positive definite, then one uses the monotonicity of $\mathcal{G}^2$ and the first part of the theorem to see that $X_\infty \leq \mathcal{G}^2(X) \leq Q$. Then, again using the monotonicity of $\mathcal{G}^2$ we get that $X_\infty \leq \mathcal{G}^{2j+2}(X) \leq \mathcal{G}^{2j}(Q)$. As the right-hand side converges to X_∞ this proves that $\mathcal{G}^{2j}(X)$ converges to X_∞, and hence also $\mathcal{G}^{2j+1}(X)$ converges to $X_{-\infty}$. □

Example 2.1. As an example consider the case $\mathcal{F}(X) = X$, that is, consider the equation

$$X + A^\star X A = Q. \tag{2.2}$$

The condition $\mathcal{G}(Q) > 0$ now gives $Q - A^\star Q A > 0$. From [9, Theorem 13.2.1], we see that Q being positive definite implies that A is stable with respect to the unit circle. Now consider the periodic points of $\mathcal{G}$ with period 2. They are fixed points of the equation $\mathcal{G}^2(X) = X$, which becomes

$$X = Q - A^\star(Q - A^\star X A)A = \mathcal{G}(Q) + A^{2\star} X A^2.$$

This is a standard Stein equation, and by the same theorem in [9], this has a unique solution. It follows that in the notation of the previous theorem, $X_{-\infty} = X_\infty$ is a fixed point of $\mathcal{G}$, and it is the unique positive definite solution to (2.2).

That the condition $\mathcal{G}(Q) > 0$ is necessary here can be seen by considering $A = \left(\begin{smallmatrix} 1 & 1 \\ 0 & -1 \end{smallmatrix}\right)$. In that case $A^2 = I_2$. Taking $Q = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$, we get $\mathcal{G}(Q) = 0$. This implies that *every* positive definite matrix is a solution of $\mathcal{G}^2(X) = X$; however, one easily computes that a positive definite X solves (2.2) for this case if and only if $X = \left(\begin{smallmatrix} 1 & x \\ \bar{x} & \frac{1}{2} + \mathrm{Re}\ x \end{smallmatrix}\right)$, with $|x - \frac{1}{2}| < \frac{1}{2}\sqrt{3}$.

Example 2.2. A simple example shows that the conditions of Theorem 2.2 are not sufficient to guarantee existence of a solution of (1.1). Indeed, take $n = 1$, so that we are in the scalar case, and take $Q = 4, A = 1$ and $\mathcal{F}(x) = 2$ for $x \leq 1\frac{1}{2}$ and $\mathcal{F}(x) = 3$ for $x > 1\frac{1}{2}$. Clearly there is no solution to (1.1), and a periodic orbit of $\mathcal{G}$ is given by $1, 2$.

Also, in the scalar case it is easily seen that there can be no periodic orbits of period larger than 2. To see this, one uses the fact that the real numbers are totally ordered. Indeed, let $x_1 \leq x_2 \leq \cdots \leq x_p$ be the numbers in a periodic orbit of $\mathcal{G}$, arranged in increasing order. Since $\mathcal{G}$ is antimonotone we get $\mathcal{G}(x_p) \leq \cdots \leq \mathcal{G}(x_1)$. But since this is a periodic orbit, these must be the same numbers as $x_1, \ldots, x_p$. Thus $\mathcal{G}(x_1) = x_p$ and $\mathcal{G}(x_p) = x_1$. So, x_1, x_p form a periodic orbit of period 2.

In order for a fixed point to exist, we need an additional assumption of $\mathcal{F}$, and it is natural to assume that $\mathcal{F}$ is continuous. It turns out that in this case existence of a fixed point is guaranteed.

THEOREM 2.3. *If $\mathcal{F}$ is monotone and continuous on $\mathcal{P}(n)$ and if* (1.2) *holds, then there exists a solution to* (1.1).

Proof. As we have seen, the set $[\mathcal{G}(Q), Q]$ is compact in the set of all $n \times n$ matrices. It is easily seen to be convex as well; that is, if X and Y are in between $\mathcal{G}(Q)$ and Q, then so is $tX + (1-t)Y$ for $0 \leq t \leq 1$. Indeed, $tX - t\mathcal{G}(Q)$ and $(1-t)Y - (1-t)\mathcal{G}(Q)$ are positive semidefinite, and as the set of positive semidefinite matrices is a cone, their sum is positive semidefinite as well. So $tX + (1-t)Y - \mathcal{G}(Q) \geq 0$. Likewise one shows that $Q - tX - (1-t)Y \geq 0$.

Under the condition (1.2) $\mathcal{G}$ maps this compact convex subset of the Banach space of $n \times n$ matrices into itself. Since $\mathcal{F}$ is continuous, so is $\mathcal{G}$, and hence we can apply

the Schauder fixed point theorem (see, e.g., [7], section 106), to see that a fixed point of $\mathcal{G}$ must exist. ∎

In the scalar case if $\mathcal{F}$ is continuous, then it is easily seen that there is a unique fixed point of $\mathcal{G}$, but it is not necessarily obtained as the limit of the sequence $\mathcal{G}^j(Q)$. In order for this to hold we need something additional.

In the next theorem, existence and uniqueness of a solution of (1.1) is proven, and the rate of convergence of the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^{\infty}$ is studied under an additional condition. To do this we will use Banach's fixed point theorem. Recall that a function $\mathcal{F} : [0,\infty) \to [0,\infty)$ is called *operator monotone* if for any n and any pair of hermitian $n \times n$ matrices A and B with $A \le B$ we have $\mathcal{F}(A) \le \mathcal{F}(B)$. See [2] for a detailed study and a complete characterization of such functions. Observe in particular that an operator monotone map is differentiable [2, Theorem V.3.6].

THEOREM 2.4. *Let $\mathcal{F} : [0,\infty) \to [0,\infty)$ be operator monotone. Let α be the smallest eigenvalue of $Q - A^\star\mathcal{F}(Q)A$ and assume that the condition* (1.2) *holds. If $q := \|A\|^2\mathcal{F}'(\alpha) < 1$, then* (1.1) *has a unique positive solution X_∞ and the iteration $X_{n+1} = Q - A^\star\ \mathcal{F}(X_n)\ A$, started at $X_0 = Q$, converges to X_∞ with*

$$\|X_{j+1} - X_j\| \ \le\ q\ \|X_j - X_{j-1}\|. \tag{2.3}$$

Moreover, there are no periodic orbits of $\mathcal{G}$, and the iteration process converges to X_∞ from any positive definite X_0 for which $Q - A^\star\mathcal{F}(X_0)A$ is positive definite.

Proof. Observe that the set $[Q - A^\star\mathcal{F}(Q)A,\ Q] = [\mathcal{G}(Q), Q]$ is a compact and hence complete metric space and that $\mathcal{G}$ maps this set into itself. We shall prove that the operator $\mathcal{G}$ is a strict contraction on the set $[Q - A^\star\mathcal{F}(Q)A,\ Q]$. For this purpose, let X and Y be in $[Q - A^\star\mathcal{F}(Q)A,\ Q]$. Then

$$\|\mathcal{G}(X) - \mathcal{G}(Y)\| = \|A^\star\ (\mathcal{F}(X) - \mathcal{F}(Y))\ A\| \le \|A\|^2\|\mathcal{F}(X) - \mathcal{F}(Y)\|. \tag{2.4}$$

Since X and Y both are greater than or equal to $Q - A^\star\mathcal{F}(Q)A > 0$, we have

$$X \ge \alpha I \qquad \text{and} \qquad Y \ge \alpha I\ (\alpha > 0).$$

(Recall that α is the smallest eigenvalue of $Q - A^\star\mathcal{F}(Q)A$, which is positive by assumption.) So we have by Theorem X.3.8 in [2] that

$$\|\mathcal{F}(X) - \mathcal{F}(Y)\| \le \mathcal{F}'(\alpha)\|X - Y\|. \tag{2.5}$$

By combining the two inequalities (2.4) and (2.5), we get

$$\|\mathcal{G}(X) - \mathcal{G}(Y)\| \le \|A\|^2\ \mathcal{F}'(\alpha)\|X - Y\| = q\ \|X - Y\|. \tag{2.6}$$

Since $q < 1$ by assumption, we can apply Banach's fixed point theorem; hence (1.1) has a unique positive solution X_∞ in $[Q - A^\star\mathcal{F}(Q)A,\ Q]$. By Lemma 5.5 it follows that X_∞ is the unique positive definite solution. Moreover, the sequence of successive approximations

$$X_{j+1} = Q - A^\star\ \mathcal{F}(X_j)\ A = \mathcal{G}(X_j), \qquad j = 0, 1, 2, \ldots,$$

started at $X_0 = Q$, i.e., $X_j = \mathcal{G}^j(Q)$, converges to X_∞.

As $X_{-\infty} = X_\infty$ there can be no periodic orbits of $\mathcal{G}$, and the convergence of $\mathcal{G}^j(X_0)$ to X_∞ for any X_0 for which $\mathcal{G}(X_0)$ is positive definite follows from Theorem 2.2. This completes the proof of the theorem. ∎

COROLLARY 2.5. *If all assumptions in the above theorem are satisfied and* $X_0 = Q$, *then*

$$\|X_{j+1} - X_j\| \ \leq \ q^j \ \|X_1 - X_0\| = q^j \|A^\star \mathcal{F}(Q)A\|. \tag{2.7}$$

It follows from this that if $q < 1$ we have the following error bound:

$$\|X_\infty - X_j\| \leq q^j \|A^\star \mathcal{F}(Q)A\|.$$

Indeed, recall that X_∞ is always between two consecutive elements of the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$.

The next corollary describes the number of iterations to be taken to ensure that $\|X_\infty - X_j\| \leq \varepsilon$.

COROLLARY 2.6. *If* ε *is a convergence tolerance and* $X_0 = Q$, *then the number* n *of iterations to be taken is at most*

$$n = \left[\frac{\ln \varepsilon - \ln \|A^\star \mathcal{F}(Q) \ A\|}{\ln q}\right] + 1.$$

In the theory of Stein equations, i.e., equations of the form $X - A^\star X A = Q$, with $Q > 0$, there are well-known results relating the eigenvalues of A and X. The following theorem may be viewed as an analogue of these results for the type of equations under consideration in this section.

THEOREM 2.7. *Let* $\mathcal{F} : [0, \infty) \to [0, \infty)$ *be operator monotone. Let* X *be a positive definite solution of* (1.1), *and denote by* μ_+ *and* μ_- *the largest and smallest eigenvalue of* X, *respectively. Also, denote by* q_+ *and* q_- *the largest and smallest eigenvalue of* $Q - X$, *respectively. If* λ *is an eigenvalue of* A, *then*

$$\sqrt{\frac{q_-}{\mathcal{F}(\mu_+)}} \leq |\lambda| \leq \sqrt{\frac{q_+}{\mathcal{F}(\mu_-)}}.$$

In the particular case where $Q = I$, *then*

$$\sqrt{\frac{1 - \mu_+}{\mathcal{F}(\mu_+)}} \leq |\lambda| \leq \sqrt{\frac{1 - \mu_-}{\mathcal{F}(\mu_-)}}.$$

Proof. Let v be an eigenvector corresponding to an eigenvalue λ of the matrix A and $\|v\| = 1$. Then, with the usual scalar product denoted by $\langle \cdot, \cdot \rangle$, we have

$$\langle Xv, v\rangle + \langle A^\star \mathcal{F}(X)Av, v\rangle = \langle Qv, v\rangle.$$

So

$$|\lambda|^2 \langle \mathcal{F}(X)v, v\rangle = \langle (Q - X)v, v\rangle.$$

Now $Q - X$ is positive definite and $q_- I \leq Q - X \leq q_+ I$, and the largest eigenvalue of $\mathcal{F}(X)$ is $\mathcal{F}(\mu_+)$, so

$$|\lambda|^2 \mathcal{F}(\mu_+) \geq q_-.$$

Likewise, as the smallest eigenvalue of $\mathcal{F}(X)$ is $\mathcal{F}(\mu_-)$, we have that

$$|\lambda|^2 \mathcal{F}(\mu_-) \leq q_+.$$

In the case where $Q = I$, simply observe that $q_- = 1 - \mu_+$ and $q_+ = 1 - \mu_-$. ∎

3. The antimonotone case. In this section $\mathcal{F}$ is assumed to be antimonotone. First, we show that this implies that $\mathcal{G}$, defined by $\mathcal{G}(X) = Q - A^\star \mathcal{F}(X)A$, is monotone. Indeed, let $0 < X \le Y$. Then

$$\mathcal{G}(Y) - \mathcal{G}(X) = A^\star(\mathcal{F}(X) - \mathcal{F}(Y))A,$$

and as $\mathcal{F}$ is antimonotone, the latter is positive semidefinite. So $\mathcal{G}(X) \le \mathcal{G}(Y)$. Next we present necessary and sufficient conditions for the existence of a positive definite solution.

THEOREM 3.1. *Let $\mathcal{F}$ be antimonotone on $\mathcal{P}(n)$. There is a positive definite solution to* (1.1) *if and only if the sequence of matrices $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is positive definite for all j, and the sequence $\{(\mathcal{G}^j(Q))^{-1}\}_{j=0}^\infty$ is uniformly bounded.*

In this case the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is decreasing and converges to the largest positive definite solution of (1.1).

Proof. Suppose that X_0 is an arbitrary positive definite solution of (1.1). Clearly $X_0 \le Q$. As $\mathcal{G}$ is monotone it follows that $\mathcal{G}^j(Q) \ge X_0$ for all j. Thus $\mathcal{G}^j(Q)$ is positive definite for all j. As $Q - A^\star \mathcal{F}(Q)A = \mathcal{G}(Q) \le Q$ and $\mathcal{G}$ is monotone we see that the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is a decreasing sequence. As this sequence is bounded below by X_0 it converges to a positive definite solution of (1.1), which we denote by X_∞. Observe that it may be the case that $X_0 \ne X_\infty$, but certainly $X_\infty \ge X_0$. This also proves that X_∞ is the largest positive definite solution. Then $(\mathcal{G}^j(Q))^{-1}$ converges to X_∞^{-1} and hence is uniformly bounded.

Conversely, assume that $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is positive definite for all j, and the sequence $\{(\mathcal{G}^j(Q))^{-1}\}_{j=0}^\infty$ is uniformly bounded. We have already seen that the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is decreasing. As each element in the sequence is positive definite, it is also bounded below (by the zero matrix). Thus there exists a limit, again denoted by X_∞, which is positive semidefinite. We only have to show that X_∞ is invertible, as it then will follow that X_∞ solves (1.1). Since $\{\mathcal{G}^j(Q)\}_{j=0}^\infty$ is a decreasing sequence of positive definite matrices it follows that $\{(\mathcal{G}^j(Q))^{-1}\}_{j=0}^\infty$ is an increasing sequence of positive definite matrices. As this sequence is uniformly bounded, it has a limit, say Y_∞, which is positive definite. Then clearly $Y_\infty^{-1} = X_\infty$, which therefore is also positive definite. ∎

Concerning the order of convergence we can be less explicit here than in the previous section. In fact, in order to determine the order of convergence we would like to have an a priori lower bound on the eigenvalues of X_∞, as can be seen from the following theorem.

THEOREM 3.2. *Let $\mathcal{F} : [0,\infty) \to [0,\infty)$ be operator antimonotone. Assume that* (1.1) *has a positive definite solution. Let β be less than or equal to the smallest eigenvalue of the largest positive definite solution X_∞. Put $q = |\mathcal{F}'(\beta)| \cdot \|A\|^2$. Also denote $X_j = \mathcal{G}^j(Q)$ for $j = 0, 1, \ldots$. Then for $j \ge 1$*

$$\|X_{j+1} - X_j\| \le q\|X_j - X_{j-1}\|.$$

Proof. The proof uses the same methods as the proof of Theorem 2.2. In fact, we know from the proof of the previous theorem that $X_\infty \le X_j \le Q$ for all j. Now if $X_\infty \le Y \le Q$, then $X_\infty \le \mathcal{G}(Y) \le X_1 \le Q$, and for X and Y in $[X_\infty, Q]$ we have

$$\|\mathcal{G}(X) - \mathcal{G}(Y)\| \le \|A\|^2 \|\mathcal{F}(X) - \mathcal{F}(Y)\|.$$

As X and Y both are greater than or equal to X_∞ we have that $X \ge \beta I$ and $Y \ge \beta I$. Then we again apply Theorem X.3.8 in [2] to finish the proof. ∎

In the case where A is invertible and $\mathcal{F}^{-1}$ exists and is antimonotone as well, we can find an a priori lower bound for X_∞ as follows. Suppose X is any positive definite solution; then $A^\star\mathcal{F}(X)A \leq Q$, from which it follows that $X \geq \mathcal{F}^{-1}(A^{-\star}QA^{-1})$. Thus in that case we can take β to be the smallest eigenvalue of $\mathcal{F}^{-1}(A^{-\star}QA^{-1})$. For instance, in the case where $\mathcal{F}(x) = x^{-1}$ and A is invertible, we can use such an estimate.

4. Remarks and numerical results. So far we considered general nonlinear matrix equations and achieved general conditions for the existence of a positive definite solution. Moreover, we discussed an iterative algorithm from which a solution can always be calculated numerically whenever the equation is solvable.

Let us see how this works in particular cases. As a first example, take $\mathcal{F}(x) = x^r$, with $0 < r < 1$, take $Q = I$, and take A a contraction. Then all conditions of Theorem 2.2 are satisfied. The condition $q < 1$ of Theorem 2.4 becomes $\rho r(1-\rho)^{r-1} < 1$, where ρ denotes $\|A\|^2$.

As a second example, take $\mathcal{F}(x) = \frac{1}{x}$. Then we can apply the results of section 3. Assuming that A is invertible, we can take for β the minimal eigenvalue of $AQ^{-1}A^\star$. In the case in which we take $Q = I$, this is the minimal eigenvalue of $AA^\star$. We get that $\mathcal{F}'(\beta) = -\frac{1}{\beta^2}$, so $q = \frac{\rho}{\beta^2}$. If we would like $q < 1$, then that amounts to $\rho < \beta^2$. Observe that for this choice of β, however, we always have that $\rho > \beta$.

Finally, we note that the results obtained so far on the iterative procedure for finding positive definite solutions are more general than those obtained in [3, 4, 12, 16, 17] in the sense that we deal with a larger class of matrix equations. It should be emphasized that the methods proposed in [6, 16, 17], while performing probably better for the special case under consideration there ($\mathcal{F}(X) = X^{-1}$), may not be so readily applied to the very general case we have under consideration here. The reader is referred to [6, 16, 17], where the numerical procedures are discussed and calculated in greater detail.

In the remainder of this section, we report some numerical results. These numerical results describe the performance of the algorithm. The numerical experiments were carried out on an IBM-PC Pentium 233 MHz computer with double precision. The machine precision is approximately 1.11×10^{-16}. We use the FORTRAN language with FORTRAN PowerStation (visual workbench version 1.00) to calculate the appended results. Table 1 indicates the convergence pattern of the iterative sequence of approximate solutions. In the example we take $Q = I$. In Table 1 n denotes the order of the matrix, k denotes the number of iterations, ε_k denotes $\|X_k + A^\star\mathcal{F}(X_k)A - I\|_\infty$, and R_k denotes the relative errors $R_k = \frac{\|X_\infty - X_k\|_\infty}{\|X_\infty\|_\infty}$, where X_∞ is taken to be the final iterate after $\varepsilon_k < 10^{-8}$ is satisfied.

The algorithm has been tested for one form of $\mathcal{F}(X)$. We take $\mathcal{F} = \sqrt{X}$ (the operator monotone case). Observe that here we only comment on the iterative procedure described in the present paper. Although this works fine for many cases, there is no claim that it is the best available procedure.

Example 4.1.

$$A = \frac{0.5\ B}{\|B\|_\infty}, \quad \text{where } B = (b_{ij}) = (i + j + 1). \tag{4.1}$$

In this example $\rho = 0.25$ and $q = 0.144338 < 1$.

In Table 1, the number of iterations can be expected if we use Corollary 2.6. In the example the number of iterations is at least 8 ($\|X_\infty - X_j\| < 10^{-8}$). The results show that with this method the efficiency and the accuracy achieved are acceptable

TABLE 1
Error analysis for (4.1).

n	k	ε_k	R_k
4	13	1.122320E-08	1.745058E-09
6	11	7.450581E-09	1.042610E-08
8	10	1.303852E-08	1.253618E-08
10	10	3.725290E-09	4.896479E-09
12	10	3.725290E-09	4.818559E-09
14	9	6.519258E-09	8.335872E-09
16	9	2.793968E-09	3.541677E-09
18	9	1.862645E-09	2.345221E-09
20	9	1.396984E-09	1.166283E-09
22	8	9.778887E-09	1.045015E-08
24	8	8.676356E-09	6.984919E-09

in the sense that we get a numerically reliable solution (in single precision) within a relatively small number of iterations.

5. Antimonotone $\mathcal{F}$ mapping positive definite to negative definite matrices. In this section we consider (1.1) under the assumption that $\mathcal{F} : \mathcal{P}(n) \to -\mathcal{P}(n)$ is antimonotone. Obviously, this implies that $\mathcal{G}$ is a monotone map mapping positive definite matrices into matrices that are greater than or equal to Q.

First we state a general theorem concerning this class of equations.

THEOREM 5.1. *Let $\mathcal{F} : \mathcal{P}(n) \to -\mathcal{P}(n)$ be antimonotone. Assume there is a positive definite matrix $\tilde{X}_0$ such that $\tilde{X}_0 \geq \mathcal{G}(\tilde{X}_0)$. Then the following hold true.*

(i) *The sequence $\mathcal{G}^j(Q)$ is an increasing sequence that is bounded above. Its limit, which we denote by X_-, is a solution to* (1.1).

(ii) *The sequence $\mathcal{G}^j(\tilde{X}_0)$ is a decreasing sequence that is bounded below. Its limit is a solution to* (1.1).

(iii) *X_- is the smallest positive definite solution to* (1.1). *Moreover, if $X_j, j = 1, \ldots, p$ is a periodic orbit of $\mathcal{G}$ consisting of positive definite matrices, then $X_j \geq X_-$ for all j.*

Proof. Observe that for any positive definite matrix X we have $\mathcal{G}(X) \geq Q$ as $\mathcal{F}$ maps positive definite matrices into negative definite matrices. Now $\tilde{X}_0 \geq \mathcal{G}(\tilde{X}_0) \geq Q$. Since $\mathcal{G}$ is monotonic, repeated application of $\mathcal{G}$ gives

$$\tilde{X}_0 \geq \mathcal{G}^j(\tilde{X}_0) \geq \mathcal{G}^{j+1}(\tilde{X}_0) \geq \mathcal{G}^j(Q) \geq Q.$$

This proves (i) and (ii).

To prove (iii), suppose $X_j, j = 1, \ldots, p$, is a periodic orbit of $\mathcal{G}$ of period p. Then X_j is a fixed point of $\mathcal{G}^p$. In particular $X_j \geq Q$, and then by monotonicity of $\mathcal{G}^p$ we get that $X_j \geq \mathcal{G}^{kp}(Q)$ for all positive integers k. Thus $X_- \leq X_j$. ∎

Part of the theorem can be restated as follows.

THEOREM 5.2. *Let $\mathcal{F} : \mathcal{P}(n) \to -\mathcal{P}(n)$ be antimonotone. Assume that there is a solution $\tilde{X}_0$ to the inequality*

$$X + A^\star \mathcal{F}(X) A \geq Q.$$

Then there is a solution to the equation

$$X + A^\star \mathcal{F}(X) A = Q,$$

and the sequence $\{\mathcal{G}^j(Q)\}_{j=0}^{\infty}$ increases to the smallest hermitian solution of this equation.

For an important example of this type of map consider the discrete algebraic Riccati equation

$$X = A^\star XA + Q - (A^\star XB + S)(R + B^\star XB)^{-1}(B^\star XA + S^\star),$$

where we assume that $Q = Q^\star$ and $R = R^\star$ is invertible. In linear quadratic optimal control problems usually $R > 0$ and $Q - SR^{-1}S^\star \geq 0$. It is also well known that if in addition $(A, Q - SR^{-1}S^\star)$ is observable, the solution of interest is positive definite; see [10]. By Proposition 12.1.1 in [10] we can restrict our attention to the case $S = 0$, i.e., to the equation

$$X = A^\star XA + Q - A^\star XB(R + B^\star XB)^{-1}B^\star XA, \tag{5.1}$$

where we assume that $Q \geq 0$ and R is positive definite. A good source of information concerning the discrete algebraic Riccati equation is [10], where iteration methods for solving it are also discussed. See also [14].

Here we shall restrict our attention to the particular case where Q is positive definite. It should be noted that this is a serious restriction, as most practical applications in linear quadratic optimal control theory would have a Q which is not invertible. However, here we are interested to see how the methods developed before can be applied to (5.1). In the meantime we develop results for several wider classes of equations, as will be seen in what follows.

For this equation we first reduce to the case where $R = I$ by replacing B by $BR^{-\frac{1}{2}}$. Then it is of the form (1.1) with

$$\mathcal{F}(X) = -X + XB(I + B^\star XB)^{-1}B^\star X.$$

We shall show that $\mathcal{F}$ maps positive definite matrices to negative definite matrices and is antimonotone. Indeed, first observe that

$$B(I + B^\star XB)^{-1} = (I + BB^\star X)^{-1}B,$$

so that

$$\begin{aligned}\mathcal{F}(X) &= -X + X(I + BB^\star X)^{-1}BB^\star X \\ &= -X + X(I + BB^\star X)^{-1}\{(I + BB^\star X) - I\} \\ &= -X(I + BB^\star X)^{-1}.\end{aligned}$$

For X invertible we get

$$\mathcal{F}(X) = -(X^{-1} + BB^\star)^{-1}.$$

Clearly it follows that $\mathcal{F}$ maps $\mathcal{P}(n)$ into $-\mathcal{P}(n)$. Furthermore, if $X \geq Y > 0$ then $0 < X^{-1} + BB^\star \leq Y^{-1} + BB^\star$, and hence $(X^{-1} + BB^\star)^{-1} \geq (Y^{-1} + BB^\star)^{-1}$, so that $\mathcal{F}$ is antimonotone.

Compare Theorem 5.2 to [11, Theorem 3.1]. There, a similar statement was proved concerning the discrete algebraic Riccati equation. It should be noted, however, that the result in [11] does not follow from the above theorem.

Let us see how we can apply Theorem 5.1 to the case of the discrete algebraic Riccati equation (5.1). We shall only consider a special case, namely the case where, as before, $R = I$ and Q is positive definite, but in addition we require A to be stable. First we observe that in that case $\mathcal{F}(X) = -X + \mathcal{H}(X)$, where $\mathcal{H} : \mathcal{P}(n) \to \mathcal{P}(n)$. That situation is treated in the following theorem.

THEOREM 5.3. *Let $\mathcal{F} : \mathcal{P}(n) \to -\mathcal{P}(n)$ be antimonotone. Assume $\mathcal{F}(X) = -X + \mathcal{H}(X)$, where $\mathcal{H} : \mathcal{P}(n) \to \mathcal{P}(n)$, and that A is stable. Denote by $\tilde{X}_0$ the unique solution to the Stein equation*

$$X - A^\star X A = Q. \tag{5.2}$$

Then the sequence $\{\mathcal{G}^j(\tilde{X}_0)\}$ is a decreasing sequence of positive definite matrices having a positive definite limit X_+, and X_+ is the largest positive definite solution of

$$X + A^\star \mathcal{F}(X) A = Q.$$

Proof. Observe that $\mathcal{G}(\tilde{X}_0) = Q - A^\star \mathcal{F}(\tilde{X}_0)A = Q + A^\star \tilde{X}_0 A - A^\star \mathcal{H}(\tilde{X}_0)A = \tilde{X}_0 - A^\star \mathcal{H}(\tilde{X}_0)A \le \tilde{X}_0$. So we can apply Theorem 5.1 to see that the sequence $\{\mathcal{G}^j(\tilde{X}_0)\}$ is a decreasing sequence having a positive definite limit X_+, which is a solution to (1.1). So we only have to show that it is the largest positive definite solution.

Let X be any positive definite solution to (1.1). Then

$$\begin{aligned} \tilde{X}_0 - X &= Q + A^\star \tilde{X}_0 A - (Q - A^\star \mathcal{F}(X) A) \\ &= A^\star (\tilde{X}_0 + \mathcal{F}(X)) A \\ &= A^\star (\tilde{X}_0 - X) A + A^\star \mathcal{H}(X) A. \end{aligned}$$

So $\tilde{X}_0 - X$ solves the Stein equation

$$\tilde{X}_0 - X - A^\star (\tilde{X}_0 - X) A = A^\star \mathcal{H}(X) A.$$

As $\mathcal{H}(X)$ is positive definite and A is stable we see that $\tilde{X}_0 - X$ is positive semidefinite. So $X \le \tilde{X}_0$. As $\mathcal{G}$ is monotone, this implies that $X = \mathcal{G}^j(X) \le \mathcal{G}^j(\tilde{X}_0)$ for all positive integers j, so that $X \le X_+$. ∎

Observe that this result can be applied directly to the discrete algebraic Riccati equation under the assumptions that Q is positive definite and A is stable. (Recall that we may assume that $R = I$ without loss of generality.) That yields the following result.

COROLLARY 5.4. *Let A be stable and let Q be positive definite. Let $\tilde{X}_0$ be the unique solution of the Stein equation* (5.2)*. Define the sequence of matrices $\{X_j\}$ by*

$$X_{j+1} = Q + A^\star X_j A - A^\star X_j B (R + B^\star X_j B)^{-1} B^\star X_j A,$$

with $X_0 = \tilde{X}_0$. Then this sequence of matrices decreases to the largest positive definite solution of (5.1)*.*

Define the sequence of matrices $\{Q_j\}$ by

$$Q_{j+1} = Q + A^\star Q_j A - A^\star Q_j B (R + B^\star Q_j B)^{-1} B^\star Q_j A,$$

with $Q_0 = Q$. Then this sequence of matrices increases to the smallest positive definite solution of (5.1)*.*

This result is well known—as a matter of fact it can be proven that under the present conditions there is a unique positive definite solution (see, e.g., [10, Theorem 13.5.3]). To obtain this result we need to use much more of the structure of the map $\mathcal{F}$. We start with a lemma.

LEMMA 5.5. *Assume that $\mathcal{F} : \mathcal{P}(n) \to -\mathcal{P}(n)$ is antimonotone and is of the form $\mathcal{F}(X) = -X + X\mathcal{H}(X)X$, where $\mathcal{H}$ satisfies $\mathcal{H}(X)X\mathcal{H}(X) \le \mathcal{H}(X)$. Then for*

every positive definite solution X of $X = \mathcal{G}(X)$ the matrix A_X defined by $A_X = A - \mathcal{H}(X)XA$ is stable.

Proof. Let X be a positive definite solution of (1.1) and compute

$$\begin{aligned} &X - A_X^\star X A_X \\ &= X - A^\star XA + 2A^\star X\mathcal{H}(X)XA - A^\star X\mathcal{H}(X)X\mathcal{H}(X)XA \\ &= Q + A^\star X\mathcal{H}(X)XA - A^\star X\mathcal{H}(X)X\mathcal{H}(X)XA. \end{aligned}$$

From the assumption on $\mathcal{H}$ it follows that $X - A_X^\star X A_X$ is positive definite. As X is positive definite we get that A_X is stable (see, e.g., [9, section 13.2]). ∎

The next theorem describes conditions which are satisfied in the case of the discrete algebraic Riccati equation and which allow us to deduce that there is a unique positive definite solution.

THEOREM 5.6. *Let $\mathcal{F}(X) = -X + X\mathcal{H}(X)X$ map positive definite matrices into negative definite matrices, and let it be antimonotone. Assume that $\mathcal{H}$ satisfies the following two properties:*

$$\mathcal{H}(X)X\mathcal{H}(X) \le \mathcal{H}(X), \tag{5.3}$$

$$\mathcal{H}(Y) - \mathcal{H}(X) = \mathcal{H}(X)(X - Y)\mathcal{H}(Y). \tag{5.4}$$

Then there is a unique positive definite solution to the equation

$$X - A^\star XA + A^\star X\mathcal{H}(X)XA = Q. \tag{5.5}$$

Proof. From the results obtained so far it follows that we only have to show that $X_+ = X_-$. To do this, put $A_\pm = A - \mathcal{H}(X_\pm)X_\pm A$. As (5.3) holds, we can apply the previous lemma to see that both these matrices are stable. Now compute

$$\begin{aligned} &X_+ - X_- - A_+^\star(X_+ - X_-)A_- \\ &= X_+ - X_- - (A^\star - A^\star X_+\mathcal{H}(X_+))X_+(A - \mathcal{H}(X_-)X_-A) \\ &\quad + (A^\star - A^\star X_+\mathcal{H}(X_+))X_-(A - \mathcal{H}(X_-)X_-A) \\ &= X_+ - A^\star X_+A + A^\star X_+\mathcal{H}(X_+)X_+A \\ &\quad + A^\star X_+\mathcal{H}(X_-)X_-A - A^\star X_+\mathcal{H}(X_+)X_+\mathcal{H}(X_-)X_-A \\ &\quad - X_- + A^\star X_-A - A^\star X_-\mathcal{H}(X_-)X_-A \\ &\quad - A^\star X_+\mathcal{H}(X_+)X_-A + A^\star X_+\mathcal{H}(X_+)X_-\mathcal{H}(X_-)X_-A. \end{aligned}$$

Using the fact that X_+ and X_- are both solutions to (5.5) we see that

$$\begin{aligned} &X_+ - X_- - A_+^\star(X_+ - X_-)A_- \\ &= A^\star X_+(\mathcal{H}(X_-) - \mathcal{H}(X_+) + \mathcal{H}(X_+)X_-\mathcal{H}(X_-) - \mathcal{H}(X_+)X_+\mathcal{H}(X_-))X_-A = 0, \end{aligned}$$

where the last equality follows from (5.4). As A_+ and A_- are both stable, the equation $Y - A_+^\star Y A_- = 0$ has a unique solution, being the zero matrix. It follows that $X_+ = X_-$, as desired. ∎

It is easily seen in the case of the discrete algebraic Riccati equation that both conditions (5.3) and (5.4) are satisfied. Indeed, in that case $\mathcal{H}(X) = B^\star(R + B^\star XB)^{-1}B$, where we may assume without loss of generality that $R = I$, as before. Then

$$\begin{aligned}
&\mathcal{H}(X)X\mathcal{H}(X)\\
&= B(I + B^\star XB)^{-1}B^\star XB(I + B^\star XB)^{-1}B^\star\\
&= B(I + B^\star XB)^{-1}B^\star - B(I + B^\star XB)^{-2}B^\star\\
&\le B(I + B^\star XB)^{-1}B^\star = \mathcal{H}(X).
\end{aligned}$$

Also,

$$\begin{aligned}
&\mathcal{H}(Y) - \mathcal{H}(X)\\
&= B\{(I + B^\star YB)^{-1} - (I + B^\star XB)^{-1}\}B^\star\\
&= B(I + B^\star XB)^{-1}\{(I + B^\star XB) - (I + B^\star YB)\}(I + B^\star YB)^{-1}B^\star\\
&= B(I + B^\star XB)^{-1}B^\star(X - Y)B(I + B^\star YB)^{-1}B^\star\\
&= \mathcal{H}(X)(X - Y)\mathcal{H}(Y).
\end{aligned}$$

Thus the theorem above can be applied directly to the discrete algebraic Riccati equation.

COROLLARY 5.7. *Assume that Q is positive definite and that A is a stable matrix. Then the algebraic Riccati equation* (5.1) *has a unique positive definite solution, which is the stabilizing solution.*

Acknowledgment. The authors would like to thank Prof. Salah E. El-Gendi for his careful reading of one of the early versions of the manuscript.

REFERENCES

[1] W. N. ANDERSON, JR., T. D. MORLEY, AND G. E. TRAPP, *Positive solution to $X = A - BX^{-1}B^\star$*, Linear Algebra Appl., 134 (1990), pp. 53–62.
[2] R. BHATIA, *Matrix Analysis*, Grad. Texts in Math. 169, Springer-Verlag, New York, 1997.
[3] J. C. ENGWERDA, A. C. M. RAN, AND A. L. RIJKEBOER, *Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^\star X^{-1}A = Q$*, Linear Algebra Appl., 186 (1993), pp. 255–275.
[4] J. C. ENGWERDA, *On the existence of the positive definite solution of the matrix equation $X + A^T X^{-1}A = I$*, Linear Algebra Appl., 194 (1993), pp. 91–108.
[5] A. FERRANTE AND B. C. LEVY, *Hermitian solutions of the equation $X = Q + NX^{-1}N^\star$*, Linear Algebra Appl., 247 (1996), pp. 359–373.
[6] C.-H. GUO AND P. LANCASTER, *Iterative solution of two matrix equations*, Math. Comp., 68 (1999), pp. 1589–1603.
[7] H. HEUSER, *Funktionalanalysis*, Teubner, Stuttgart, 1975.
[8] I. G. IVANOV AND S. M. EL-SAYED, *Properties of positive definite solutions of the equation $X + A^\star X^{-2}A = I$*, Linear Algebra Appl., 279 (1998), pp. 303–316.
[9] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, Orlando, FL, 1985.
[10] P. LANCASTER AND L. RODMAN, *Algebraic Riccati Equations*, Oxford Science, New York, 1995.
[11] A. C. M. RAN AND R. VREUGDENHIL, *Existence and comparison theorems for algebraic Riccati equations for continuous- and discrete-time systems*, Linear Algebra Appl., 99 (1988), pp. 63–83.
[12] S. M. EL-SAYED, *The Study on Special Matrices and Numerical Methods for Special Matrix Equations*, Ph.D. thesis, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria, 1996.
[13] S. M. EL-SAYED, *Theorems for the existence and computing of positive definite solutions for two nonlinear matrix equations*, in Proceedings of 25th Spring Conference of the Union of Bulgarian Mathematicians, Bulgarian Academy of Sciences, Kazanlak, 1996.
[14] V. SIMA, *Algorithms for Linear-Quadratic Optimization*, Pure Appl. Math. 200, Marcel Dekker, New York, 1996.
[15] W.-Y. YAN, J. B. MOORE, AND U. HELMKE, *Recursive algorithms for solving a class of nonlinear matrix equations with applications to certain sensitivity optimization problems*, SIAM J. Control Optim., 32 (1994), pp. 1559–1576.
[16] X. ZHAN, *Computing the extremal positive definite solutions of a matrix equation*, SIAM J. Sci. Comput., 17 (1996), pp. 1167–1174.
[17] X. ZHAN AND J. XIE, *On the matrix equation $X + A^T X^{-1}A = I$*, Linear Algebra Appl., 247, (1996), pp. 337–345.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 646–655
© 2001 Society for Industrial and Applied Mathematics

SPECTRAL FACTORIZATIONS AND SUMS OF SQUARES REPRESENTATIONS VIA SEMIDEFINITE PROGRAMMING*

J. W. MCLEAN† AND H. J. WOERDEMAN†

Abstract. In this paper we find a characterization for when a multivariable trigonometric polynomial can be written as a sum of squares. In addition, the truncated moment problem is addressed. A numerical algorithm for finding a sum of squares representation is presented as well. In the one-variable case, the algorithm finds a spectral factorization. The latter may also be used to find inner-outer factorizations.

Key words. spectral factorization, inner-outer factorization, sums of squares, multivariable trigonometric polynomial, truncated moment problem, semidefinite programming

AMS subject classifications. 15A48, 42B05, 93B36

PII. S0895479800371177

1. Introduction.

The classical Riesz–Fejer factorization theorem states that a trigonometric polynomial

$$q(z) = \sum_{i=-m}^{m} q_i z^i, \qquad |z| = 1,$$

on the unit circle that solely takes on nonnegative values can be written as

$$q(z) = |p(z)|^2, \qquad |z| = 1, \tag{1.1}$$

where $p(z)$ is a polynomial that has no zeros in the disk $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$. A proof of this result based on the fundamental theorem of algebra may be found in [36, pp. 117–118]. For matrix-valued and operator-valued functions a similar result holds true; [34] and [24] proved the matrix-valued case, [22] the compact operator-valued case, and [35] the general operator-valued case. Based on an observation that was made in [17] we propose in this paper a simple algorithm that computes the spectral factorization (1.1). This algorithm is a straightforward application of semidefinite programming. Existing techniques for finding spectral factorizations use realization theory and come down to finding positive semidefinite solutions to Riccati equations (see, e.g., [28], [21], [16], [11], [31], [41], and references therein). Most literature applies to the continuous-time case, but the discrete-time case (which is relevant to this paper) may be converted to the continuous-time case (see, e.g., [23]). In addition, we use the spectral factorization algorithm to find inner-outer factorizations.

The multivariable analogue of the observation made in [17] allows for a characterization when a trigonometric polynomial of two or more variables may be written as a sum of squares. Based on this observation we present a simple algorithm for determining whether a matrix-valued trigonometric polynomial on the d-torus may be written as a sum of squares. It has been known since [12] and [33] that nonnegativity

*Received by the editors April 27, 2000; accepted for publication (in revised form) by A. C. M. Ran May 3, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/simax/23-3/37117.html

†Department of Mathematics, P.O. Box 8795, College of William and Mary, Williamsburg, VA 23187-8795 (jwmcle@wm.edu, hugo@math.wm.edu). Both authors were supported by NSF grant DMS 9800704. The first author performed the research as part of a Research Experience for Undergraduates (REU) program.

of the trigonometric polynomial does not ensure the existence of a representation as a sum of squares, and thus the above algorithm provides at least a numerical solution to this problem. In addition, we discuss the representation problem for trigonometric polynomials that are positive on an arc of the unit circle.

As was shown in [33] the question of sum of squares representations and the positive truncated moment problem are related by duality. Using this duality we shall characterize when the truncated moment problem with finite data has a solution.

The factorization, sum of squares, and moment problem have numerous applications. We mention here Darlington synthesis in electrical network theory ([15]; see also [4], [5], [6], and [27]), stability of control systems (see, e.g., [32], [7], [8], [10], [14], [21], [16], [31], [41]), and prediction theory for stationary stochastic processes (see [29], [39], [40], [25], [26]).

The paper is organized as follows. In section 2 we give necessary and sufficient conditions for a multivariable trigonometric polynomial to be written as a sum of squares. Moreover, we remind the reader how in the one-variable result the spectral factorization may be singled out. Subsequently, algorithms for finding spectral factorizations and sums of squares representations are outlined and numerical results are presented. In section 3 we give a characterization for the existence of a solution to the positive truncated moment problem.

2. Factorizations and sum of squares representations. Let $\mathcal{H}, \mathcal{K}$ be Hilbert spaces and let $\mathcal{B}(\mathcal{H}, \mathcal{K})$ be the Banach space of bounded linear operators acting $\mathcal{H} \to \mathcal{K}$. We denote $\mathcal{B}(\mathcal{H}, \mathcal{H})$ by $\mathcal{B}(\mathcal{H})$. We also let $S \subseteq \mathbb{Z}^d$ be a *halfspace*. That is, $S \cap (-S) = (0), S \cup (-S) = \mathbb{Z}^d$, and $S + S \subseteq S$. The standard halfspace in $\mathbb{Z}$ is $E_1 = \{0, 1, 2, \ldots\}$, and in $\mathbb{Z}^d$ the standard halfspace is given by

$$E_d = (\{0\} \times E_{d-1}) \cup (\{1, 2, \ldots\} \times \mathbb{Z}^{d-1}), \qquad d = 2, 3, \ldots.$$

Let Λ_+ be a finite subset of S that contains $0 \in \mathbb{Z}^d$, and consider an operator-valued trigonometric polynomial on the d-torus $\mathbb{T}^d$

$$Q(z) = \sum_{k \in \Lambda_+ - \Lambda_+} Q_k z^k, \qquad z \in \mathbb{T}^d,$$

where $z^k = (z_1, \ldots, z_d)^{(k_1, \ldots, k_d)} := z_1^{k_1} z_2^{k_2} \cdots z_d^{k_d}$. Here $Q_k, k \in \Lambda_+ - \Lambda_+$, are operators on $\mathcal{H}$, and for subsets A and B of $\mathbb{Z}^d$, we define $A - B = \{a - b : a \in A, b \in B\}$. The following theorem gives a characterization when $Q(z)$ may be written as $P(z)^* P(z)$ where P has Fourier support in Λ_+. Let δ_{jk} denote the Kronecker delta on $\mathbb{Z}^d$, i.e.,

$$\delta_{jk} = \begin{cases} 0 & \text{otherwise,} \\ 1, & j = k. \end{cases}$$

For any set A we denote its cardinality by $|A|$.

THEOREM 2.1. *Let $\mathcal{H}$ be a Hilbert space, S a halfspace of $\mathbb{Z}^d$, $0 \in \Lambda_+ \subseteq S$ a finite set, and $Q(z) = \sum_{k \in \Lambda_+ - \Lambda_+} Q_k z^k$ a $B(\mathcal{H})$-valued trigonometric polynomial. Denote by $\mathcal{F}$ the affine subspace of $\mathcal{B}(\mathcal{H}^{|\Lambda_+|})$ given by*

$$\mathcal{F} = \left\{ (F_{jk})_{j,k \in \Lambda_+} : \sum_{j,k \in \Lambda_+} F_{jk} \delta_{l,j-k} = Q_l, l \in \Lambda_+ - \Lambda_+ \right\}.$$

There exists a pseudopolynomial $P(z) = \sum_{k\in\Lambda_+} P_k z^k$ *where* $P_k \in \mathcal{B}(\mathcal{H},\mathcal{K})$ *for some Hilbert space* $\mathcal{K}$ *so that*

$$Q(z) = P(z)^* P(z), \qquad z \in \mathbb{T}^d, \tag{2.1}$$

if and only if $\mathcal{F}$ *contains a positive semidefinite operator. In that case,* $\mathcal{K}$ *may be chosen to be a subspace of* $\mathcal{H}^{|\Lambda_+|}$.

Proof. "if." Let $P(z) = \sum_{k\in\Lambda_+} P_k z^k$ satisfy (2.1). Put $G = \mathrm{row}(P_k)_{k\in\Lambda_+}$ and $F = G^*G \in \mathcal{B}(H^{|\Lambda_+|})$. Clearly $F \geq 0$, and one easily checks that (2.1) is equivalent to $F \in \mathcal{F}$.

"only if." Let $F \in \mathcal{F}$ be positive semidefinite. Factorize

$$F = G^*G, \qquad G \in \mathcal{B}(\mathcal{K}, \mathcal{H}^{|\Lambda_+|}), \tag{2.2}$$

by, for instance, taking $G = F^{1/2}$ and $\mathcal{K} = \text{closure Ran}F^{1/2} \subseteq \mathcal{B}(\mathcal{H}^{|\Lambda_+|})$. Write $\Lambda_+ = \{0 = k_0, k_1, \ldots, k_{|\Lambda_+|-1}\}$,

$$G = [P_{k_0} P_{k_1} \cdots P_{k_{|\Lambda_+|-1}}] : \bigoplus_{j=0}^{|\Lambda_+|-1} \mathcal{H} \to \mathcal{K},$$

and put $P(z) = \sum_{j=0}^{|\Lambda_+|-1} P_{k_j} z^{k_j}$. One easily checks that (2.2) yields (2.1). ∎

For the case when $\Lambda_+ = \{0, 1, \ldots, n\} \subseteq \mathbb{Z}$ the above observation was made in [17]. Note that when $\dim\mathcal{H} = n < \infty$, that is, when $Q(z)$ is scalar or matrix valued, Theorem 2.1 yields that $Q(z)$ may be written as $Q(z) = \sum_{k=1}^{l} P_k(z)^* P_k(z)$ with $P_k(z) \in \mathbb{C}^{n\times 1}$ and $l = \mathrm{rank}F$. Thus we obtain the following corollary. For a pseudopolynomial $p(z) = \sum p_k z^k$ we denote by $\mathrm{supp}(\hat{p})$ its Fourier support $\{k \in \mathbb{Z}^d : p_k \neq 0\}$.

COROLLARY 2.2. *Let* $Q(z)$ *be an* $n \times n$ *matrix-valued trigonometric polynomial on* $\mathbb{T}^d$ *with Fourier support in* $\Lambda_+ - \Lambda_+$. *Then there exists a representation*

$$Q(z) = \sum_{k=1}^{l} P_k(z)^* P_k(z), \qquad z \in \mathbb{T}^d, \tag{2.3}$$

with $P_k(z) \in \mathbb{C}^{n\times 1}$ *and* $\mathrm{supp}(\hat{P}_k) \subseteq \Lambda_+, k \in \{1, \ldots, l\}$, *if and only if* $\mathcal{F}$ *as in Theorem* 2.1 *contains a positive semidefinite element. In that case the lowest possible number of terms in* (2.3) *equals*

$$\min_{F\in\mathcal{F}, F\geq 0} \mathrm{rank}F. \tag{2.4}$$

In the case that $Q(z) = \sum_{k\in\Lambda_+-\Lambda_+} Q_k z^k$ *has coefficients* Q_k *that are real matrices, then* $P_k, k = 1, \ldots, l$, *may also be chosen to have real vector coefficients.*

Proof. The proof follows directly from the proof of Theorem 2.1. ∎

In the classical case, $d = 1$ and $\Lambda_+ = \{0, 1, 2, \ldots, m\}$, the quantity in (2.4) is bounded above by $\mathrm{rank}(Q_0)$. Though this may be proven using the matrix-valued Riesz–Fejer factorization theorem we will next prove it using the basic notion of the Schur complement.

PROPOSITION 2.3. *Let* $Q_k, k \in \{-m, \ldots, m\}$, *be given* $n \times n$ *complex matrices, and consider the compact set*

$$\mathcal{F}_+ = \left\{ (F_{ij})_{i,j=0}^{m} \geq 0 : \sum_{p=k}^{m} F_{p,p-k} = Q_k \right\}.$$

If $\mathcal{F}_+$ is nonempty, then it contains an element of rank less than or equal to $\mathrm{rank}Q_0$. *One may find such a member of $\mathcal{F}_+$ by maximizing* $\mathrm{trace}F_{00}$ *(or by maximizing* $\mathrm{trace}F_{mm}$*).*

Proof. Assume $\mathcal{F} \neq \emptyset$. Since $f(F) = \mathrm{trace}F_{00}$ is a continuous function on the compact set $\mathcal{F}_+$, a maximizer $F = (F_{ij})_{i,j=0}^m$ for f exists. We claim that $\mathrm{rank}F_{00} = \mathrm{rank}F$. Suppose this is not the case, i.e., $\mathrm{rank}F > \mathrm{rank}F_{00}$. Then the Schur complement S of F_{00} in F (for a definition see, e.g., [3]) is not zero. Write

$$S = (S_{ij})_{i,j=1}^m.$$

Let $q \in \{1, \ldots, m\}$ be so that $S_{11} = \cdots = S_{q-1,q-1} = 0$ but $S_{qq} \neq 0$. (Such q exists, since $0 \neq S \geq 0$.) Define now $\tilde{F} = (\tilde{F_{ij}})_{i,j=0}^m$ by

$$\tilde{F} = F - \begin{bmatrix} 0 & 0 \\ 0 & (S_{ij})_{i,j=1}^m \end{bmatrix} + \begin{bmatrix} (S_{ij})_{i,j=q}^m & 0 \\ 0 & 0 \end{bmatrix},$$

where the diagonal zero block matrices are of block size 1×1 and $q \times q$, respectively. Since, by definition of a Schur complement,

$$F - \begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} \geq 0,$$

we get that $\tilde{F} \geq 0$. In addition, notice that $\sum_{p=k}^m \tilde{F}_{p,p-k} = Q_k$, where we used the fact that $S \geq 0$ and $S_{ii} = 0$ implies that $S_{ij} = 0$ for all j. Thus it follows that $\tilde{F} \in \mathcal{F}_+$. Since $\mathrm{trace}S_{qq} > 0$, we get that $f(\tilde{F}) > f(F)$ gives a contradiction. Thus $\mathrm{rank}F = \mathrm{rank}F_{00}$. Moreover, since $0 \leq F_{00} \leq \sum_{p=0}^m F_{pp} = Q_0$ we get that $\mathrm{rank}F_{00} \leq \mathrm{rank}Q_0$. When maximizing $\mathrm{trace}F_{mm}$ one may reason analogously. ∎

As was observed in [17] (based on results in [37]) the element in $\mathcal{F}_+$ that has a maximal F_{00} in the Loewner ordering leads to a spectral factorization. Since $F_{00} \geq \tilde{F_{00}}$ and $\mathrm{trace}F_{00} = \mathrm{trace}\tilde{F_{00}}$ imply $F_{00} = \tilde{F_{00}}$, we in fact get the spectral factorization by maximizing trace F_{00}. Consequently, we may propose the following algorithm for finding spectral factorizations. As an aside, we mention that maximizing $\mathrm{trace}F_{mm}$ leads to a co-outer factorization (see [17]).

ALGORITHM 1. *Finding spectral factorizations of matrix-valued trigonometric polynomials.*

Let $Q_k, k \in \{-m, \ldots, m\}$, be given $n \times n$ complex matrices, and define $\mathcal{F}_+$ as in Proposition 2.3.

Step 1. Use one of the existing semidefinite programming packages ([38], [1], possibly with interface [18]) to see whether $\mathcal{F}_+ \neq \emptyset$. If so, go to Step 2; If not, no factorization exists.

Step 2. Use semidefinite programming to find $F_* := \mathrm{argmax}_{F \in \mathcal{F}_+} \mathrm{trace}F_{00}$.

Step 3. Write F_* as

$$F_* = \begin{pmatrix} P_0^* \\ \vdots \\ P_n^* \end{pmatrix} \begin{pmatrix} P_0 & \cdots & P_n \end{pmatrix},$$

which can be done by performing a Cholesky factorization. The spectral factor is now given by $P(z) = \sum_{i=0}^n P_i z^i$.

We implemented this using [1], which was very simple. To cite one experiment, let

$$Q_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad Q_1 = \begin{pmatrix} 0.2000 & 0.4000 \\ 0.3000 & 0.1000 \end{pmatrix}, \qquad Q_2 = \begin{pmatrix} 0.0200 & 0.0400 \\ 0.0100 & 0.0400 \end{pmatrix},$$

and let $Q_{-j} = Q_j^*$, $j = 1, 2$. Then we find the spectral factorization $Q(z) = P^*(z)P(z)$ with

$$P_0 = \begin{pmatrix} 0.9066 & -0.1601 \\ 0 & 0.8815 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0.2093 & 0.4095 \\ 0.3654 & 0.1578 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0.0221 & 0.0441 \\ 0.0154 & 0.0534 \end{pmatrix}.$$

Clearly, the algorithm is as good, simple, and accessible as the semidefinite package that one uses. We do observe, though, that in at least some cases simpler algorithms are available; e.g., in the case when $n = 1$ and one has a strictly positive trigonometric polynomial, the spectral factorization problem is equivalent to finding a positive definite solution X to the matrix equation $X + A^*X^{-1}A = Q$, where $Q > 0$ (see, e.g., [20]). The simple Schur complement-based algorithm discussed in [2], [20], and [19] to solve this problem is very elementary and seems to work very well. In such cases, using a semidefinite programming package is clearly overkill.

As is well known (see, e.g., [21]), as soon as one can determine spectral factorizations one may find inner-outer factorizations. Recall that for a rational matrix function $A(z)$ on the unit circle, the factorization $A(z) = F_i(z)F_o(z)$ is called *inner-outer* if $F_i(\frac{1}{\bar{z}})^*F_i(z) = I$, $z \in \mathbb{T}$, and F_o has a right inverse which is analytic in $\mathbb{D}$ (i.e., F_i is inner and F_o is outer). An inner-outer factorization exists if $\operatorname{rank} A(z)$ is constant for $|z| = 1$. In order to obtain an inner-outer factorization of $A(z)$ when $A(z)$ has full rank on $\mathbb{T}$, one may proceed as follows. Find a spectral factorization of $A(\frac{1}{\bar{z}})^*A(z)$. Call the spectral factor $F_o(z)$, and let $F_i(z) = A(z)F_o(z)^{-1}$. Then $A(z) = F_i(z)F_o(z)$ is a spectral factorization. When we let

$$A(z) = \begin{pmatrix} z^2 + 1 & z \\ 2 + z - z^2 & 3 - z^2 \end{pmatrix},$$

we find

$$A\left(\frac{1}{\bar{z}}\right)^* A(z) = \begin{pmatrix} -\frac{1}{z^2} + \frac{1}{z} + 8 + z - z^2 & -\frac{3}{z^2} + \frac{4}{z} + 7 - 2z^2 \\ -\frac{2}{z^2} + 7 + 4z - 3z^2 & -\frac{3}{z^2} + 11 - 3z^2 \end{pmatrix}.$$

Performing Algorithm 1 on this rational matrix function yields

$$F_0(z) = \begin{pmatrix} 2.2361 + 0.8944z - 0.4472z^2 & 2.6833 + 0.4472z - 0.8944z^2 \\ 0.4472z - 1.3416z^2 & 1.3416 - 0.8944z - 0.4472z^2 \end{pmatrix},$$

and by putting $F_i(z) = A(z)F_o(z)^{-1}$, we find an inner-outer factorization.

We now address the multivariable case.

ALGORITHM 2. *Finding a sums of squares representation.*

Let $\Lambda_+ \subset S \subset \mathbb{Z}^d$ and $Q_k \in \mathbb{C}^{n\times n}, k \in \Lambda_+ - \Lambda_+$, be given, and define $\mathcal{F}$ as in Theorem 2.1. Let $\mathcal{P}$ denote the cone of positive semidefinite matrices (of size $n|\Lambda_+|$).

Step 1. Use one of the existing semidefinite programming packages ([38], [1], possibly with interface [18]) to see whether $\mathcal{F} \cap \mathcal{P} \neq \emptyset$. If so, go to Step 2; if not, no sum of squares representation exists.

Step 2. Choose $F \in \mathcal{F} \cap \mathcal{P}$, and write F as $F = \mathrm{col}(P_k^*)_{k\in\Lambda_+}\mathrm{row}(P_k)_{k\in\Lambda_+}$, so that P_k has rank$F$ rows. (One may use a Cholesky factorization for this and discard zero rows and columns.) Then

$$Q(z) = \sum_{\ell=1}^{\mathrm{rank}F} P^{(\ell)}(z)^* P^{(\ell)}(z), \qquad \mathrm{col}(P^{(\ell)}(z))_{\ell=1}^{\mathrm{rank}F} := \sum_{k\in\Lambda_+} P_k z^k.$$

For example, for $\Lambda_+ = \{(0,0,0),(1,0,0),(1,1,0),(1,1,1)\}$ and $q_{000} = 2, q_{100} = .5, q_{110} = .2, q_{111} = .2, q_{010} = .04, q_{001} = .06, q_{011} = .08$ (and setting $q_{-k} = \overline{q_k}$ for the appropriate k), we find

$$\begin{aligned} q(z_1,z_2,z_3) = {} & |1.2839 + 0.3894z_1 + 0.1558z_1z_2 + 0.1558z_1z_2z_3|^2 \\ & + |0.2000z_1 - 0.1033z_1z_2 + 0.0967z_1z_2z_3|^2 \\ & + |0.2138z_1z_2 + 0.2138z_1z_2z_3|^2, \qquad (z_1,z_2,z_3) \in \mathbb{T}^3. \end{aligned}$$

In finding this solution we have computed $F_* = \mathrm{argmax}_{F\in\mathcal{F}\cap\mathcal{P}}\mathrm{trace}F_{00}$ and used it for the sum of squares representation. It is an open question, however, how to characterize in the several-variable case the sum of squares representation that leads to this maximal traceF_{00}. To provide one more example, we let $f(z,w) = |1+z|^2 + |1+w|^2$, $|z| = |w| = 1$, and $\Lambda_+ = \{0,1\} \times \{0,1\}$. Maximizing trace$F_{00}$ the algorithm finds the sum of squares representation

$$f(z,w) = |1.4151 + 0.7067z + 0.7067w|^2 + |0.7067z - 0.7067w|^2.$$

There are also other representations that one may find using semidefinite programming. For instance, it is shown in [30] that a trigonometric polynomial $q(z)$ of degree m that is nonnegative on the arc $\{e^{it} \ : \ t \in [-\gamma,\gamma]\}$ $(0 < \gamma \le \pi)$ may be written in the form

$$q(z) = |p(z)|^2 + (z + 1/z - 2\cos(\gamma))|r(z)|^2, \qquad z \in \mathbb{T}, \tag{2.5}$$

where $p(z)$ and $r(z)$ are polynomials of degree m and $m-1$, respectively. In fact, in [30] the more general case of a union of several arcs is treated. Using the theory developed in this section we may now propose an algorithm for finding a representation (2.5). In [30] a basic construction using the fundamental theorem of algebra and trigonometric identities is described.

Algorithm 3. *Finding a representation* (2.5).

Let $q_0,\dots,q_m$ be given complex numbers, and put $q_{-j} = \overline{q_j}$, $j = 1,\dots,m$. Let

$$\mathcal{F} = \left\{ (f_{ij})_{i,j=0}^m \ : \ \sum_{p=k}^m f_{p,p-k} = q_k, k = -m,\dots,m \right\}.$$

Step 1. Use one of the existing semidefinite programming packages ([38], [1], possibly with interface [18]) to see whether there exist positive semidefinite matrices $A \in \mathbb{C}^{(m+1)\times(m+1)}$ and $B \in \mathbb{C}^{m\times m}$ so that

$$A + \begin{pmatrix} 0 & 0 \\ B & 0 \end{pmatrix} + \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix} - 2\cos(\gamma)\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \in \mathcal{F}. \tag{2.6}$$

(Notice that the 2×2 block decompositions are all of different sizes, as the zeros are of size $1\times m$, 1×1, or $m\times 1$.) If so, pick such A and B, and continue to Step 2. If not, no representation (2.5) exists.

Step 2. Use semidefinite programming to find rank ≤ 1 positive semidefinite $\tilde{A}$ and $\tilde{B}$ so that the diagonal sums of A and $\tilde{A}$ coincide and the diagonal sums of B and $\tilde{B}$ coincide (use Proposition 2.3).

Step 3. Write

$$\tilde{A} = \mathrm{col}(\overline{p_i})_{i=0}^{m}\mathrm{row}(p_i)_{i=0}^{m}, \qquad \tilde{B} = \mathrm{col}(\overline{r_i})_{i=0}^{m-1}\mathrm{row}(r_i)_{i=0}^{m-1}.$$

Putting $p(z) = \sum_{i=0}^{m} p_i z^i$ and $r(z) = \sum_{i=0}^{m-1} r_i z^i$ one finds the desired representation (2.5).

Performing this algorithm on the data $q_0 = 3$, $q_1 = 4 - 2\sqrt{2}$, $q_2 = 2 - \sqrt{2}$, $q_3 = 1$, and $\gamma = \frac{3\pi}{4}$, we obtain the representation

$$\begin{aligned} q(z) = {} & |0.8672 - 0.5394z + 0.5394z^2 - 0.8672z^3|^2 \\ & + \left(z + \frac{1}{z} + \sqrt{2}\right)|1.3237 - 1.0681z + 1.3237z^2|^2. \end{aligned}$$

This algorithm is based on the following proposition.

PROPOSITION 2.4. *There exists a representation* (2.5) *if and only if there exists positive semidefinite matrices* $A \in \mathbb{C}^{(m+1)\times(m+1)}$ *and* $B \in \mathbb{C}^{m\times m}$ *so that* (2.6) *holds. In that case* A *and* B *may be chosen to be of rank* 1.

Proof. Suppose that (2.5) holds with $p(z) = p_0 + \cdots + p_m z^m$ and $r(z) = r_0 + \cdots + r_{m-1}z^{m-1}$. Put

$$A = \mathrm{col}(\overline{p_i})_{i=0}^{m}\mathrm{row}(p_i)_{i=0}^{m}, \qquad B = \mathrm{col}(\overline{r_i})_{i=0}^{m-1}\mathrm{row}(r_i)_{i=0}^{m-1}.$$

Clearly $A \geq 0$, $B \geq 0$, and one easily checks that equality (2.5) implies (2.6).

Conversely, suppose positive semidefinite matrices $A = (a_{ij})_{i,j=0}^{m} \in \mathbb{C}^{(m+1)\times(m+1)}$ and $B = (b_{ij})_{i,j=0}^{m-1} \in \mathbb{C}^{m\times m}$ satisfy (2.6). By Proposition 2.3 we may find positive semidefinite matrices $\tilde{A} = (\tilde{a}_{ij})_{i,j=0}^{m} \in \mathbb{C}^{(m+1)\times(m+1)}$ and $\tilde{B} = (\tilde{b}_{ij})_{i,j=0}^{m-1} \in \mathbb{C}^{m\times m}$ both of rank ≤ 1 so that A and $\tilde{A}$ have the same sums along diagonals (i.e., $\sum_{i-j=k} a_{ij} = \sum_{i-j=k} \tilde{a}_{ij}$ for all appropriate k), and so that B and $\tilde{B}$ have the same sums along diagonals. Notice that since (2.6) holds we also have that

$$\tilde{A} + \begin{pmatrix} 0 & 0 \\ \tilde{B} & 0 \end{pmatrix} + \begin{pmatrix} 0 & \tilde{B} \\ 0 & 0 \end{pmatrix} - 2\cos(\gamma)\begin{pmatrix} \tilde{B} & 0 \\ 0 & 0 \end{pmatrix} \in \mathcal{F} \tag{2.7}$$

holds. Now write

$$\tilde{A} = \mathrm{col}(\overline{p_i})_{i=0}^{m}\mathrm{row}(p_i)_{i=0}^{m}, \qquad \tilde{B} = \mathrm{col}(\overline{r_i})_{i=0}^{m-1}\mathrm{row}(r_i)_{i=0}^{m-1},$$

where $p_i, r_i \in \mathbb{C}$, and put $p(z) = p_0 + \cdots + p_m z^m$ and $r(z) = r_0 + \cdots + r_{m-1}z^{m-1}$. One now easily checks that (2.7) implies (2.5). ∎

It should be observed that in our implementation we considered only the real case, as the semidefinite programming package we used only deals with real symmetric matrices. It is not hard, though, to convert the complex case to the real case when one uses the observation that $A + Bi$, with $A, B \in \mathbb{R}^{n\times n}$, is positive semidefinite if and only if $\left(\begin{smallmatrix} A & -B \\ B & A \end{smallmatrix}\right)$ is.

3. The positive truncated moment problem. Dual to the factorization problem is the truncated moment problem, as is well explained in [33]. In this section, we present the following observation on the moment problem. Let Λ be a subset of

$\mathbb{Z}^d$ and let $\mathcal{H}$ be a Hilbert space. A sequence $\{c_k\}_{k\in\Lambda-\Lambda}$ of bounded linear operators on $\mathcal{H}$ is called *positive semidefinite* with respect to Λ if for every sequence of $\{h_k\}_{k\in\Lambda}$ of elements in H with finite support we have that

$$\sum_{k,l\in\Lambda} \langle c_{k-l}h_k, h_l\rangle \geq 0.$$

For a positive $\mathcal{B}(\mathcal{H})$-valued Borel measure μ on $\mathbb{T}^d$ the *moments* of μ are defined by

$$c_k(\mu) = \int_{\mathbb{T}^d} z^k d\mu(z),$$

where $k = (k_1, \ldots, k_d) \in \mathbb{Z}^d$. A positive Borel measure μ on $\mathbb{T}^d$ is called a *positive extension* of $\{c_k\}_{k\in\Lambda-\Lambda}$ if $c_k = c_k(\mu)$ for every $k \in \Lambda - \Lambda$. We have the following characterization for the existence of a positive extension.

THEOREM 3.1. *Let $\Lambda \subseteq \mathbb{Z}^d$ and $\{c_k\}_{k\in\Lambda-\Lambda}$ be given operators on the Hilbert space $\mathcal{H}$. Then $\{c_k\}_{k\in\Lambda-\Lambda}$ has a positive extension if and only if the operator matrix*

$$(c_{k-l})_{k,l\in\Lambda}$$

lies in the closed cone generated by the operator matrices

$$[\mathrm{col}(z^k)_{k\in\Lambda}][\mathrm{col}(z^k)_{k\in\Lambda}]^* \otimes A, z \in \mathbb{T}^d, \qquad A \in \mathcal{B}_+(\mathcal{H}), \tag{3.1}$$

where $\mathcal{B}_+(\mathcal{H})$ is the cone of positive semidefinite operators on $\mathcal{H}$.

In [13] it was proven that every $(n+1)\times(n+1)$ singular positive semidefinite complex Toeplitz matrix lies in the cone generated by

$$[\mathrm{col}(z^k)_{k=0}^n][\mathrm{col}(z^k)_{k=0}^n]^*, \qquad |z| = 1. \tag{3.2}$$

Though not explicitly mentioned, it should be observed that Carathéodory uses the Riesz–Fejer factorization theorem in his proof. Since

$$I_{n+1} = \frac{1}{n+1}\sum [\mathrm{col}(e^{\frac{2\pi ki}{n+1}})_{k=0}^n][\mathrm{col}(e^{\frac{2\pi ki}{n+1}})_{k=0}^n]^*,$$

we get that every positive semidefinite $(n+1)\times(n+1)$ Toeplitz matrix lies in the cone generated by (3.2). This combined with Theorem 3.1 yields the classical result [13] that for $d = 1$, $\Lambda = \{0, \ldots, m\}$, and $\mathcal{H} = \mathbb{C}$ the positive truncated moment has a solution for $\{c_k\}_{k=-m}^m$ if and only if

$$(c_{k-l})_{k,l=0}^m \geq 0.$$

The negative result by [12] and [33] and the later ones by [9] imply that for $\Lambda \subseteq \mathbb{Z}^2, \mathcal{H} = \mathbb{C}$, unless

$$\Lambda = \{0,1,\ldots,m\}\times\{0,1\}, \qquad \Lambda = \{0,1,\ldots,m\}\times\{0,1\}\setminus\{(m,1)\},$$

$$\Lambda = \{0,1,\ldots,m\}\times\{0\},$$

or trivial variations of it (see [9]), the corresponding cone of positive semidefinite Toeplitz matrices $(c_{k-l})_{k,l\in\Lambda}$ is strictly larger than the cone generated by the matrices in (3.1). It is an interesting question what the extreme points of the positive semidefinite Toeplitz matrices are.

Proof of Theorem 3.1. Without loss of generality we may assume that $0 \in \Lambda \subseteq S$ for some halfspace S of $\mathbb{Z}^d$. Indeed, pick any halfspace and let $\lambda_0 = \min \Lambda$ where the minimum is taken with respect to the linear order induced by S. Now put $\tilde{\Lambda} = \Lambda - \lambda_0$. Then $\tilde{\Lambda} - \tilde{\Lambda} = \Lambda - \Lambda$ and $0 \in \tilde{\Lambda} \subseteq S$. By [33, Lemma 1.2] we have that $\{c_k\}_{k\in\Lambda-\Lambda}$ has a positive extension if and only if

$$\operatorname{trace}[(c_{k-l})_{k,l\in\Lambda} F] \geq 0 \tag{3.3}$$

for all trace class $F = (F_{jk})_{j,k\in\Lambda}$ which have the property that by putting $Q_l = \sum_{j,k\in\Lambda} F_{jk}\delta_{l,j-k}$, $l \in \Lambda-\Lambda$, we obtain a trigonometric polynomial $Q(z) = \sum_{l\in\Lambda-\Lambda} Q_l z^l$ that takes on nonnegative values only. Next, observe that checking that $Q(z) \geq 0$ for all $z \in \mathbb{T}^d$ is equivalent to checking that

$$\operatorname{trace}((\operatorname{col}(z^k)_{k\in\Lambda}[\operatorname{col}(z^k)_{k\in\Lambda}]^* \otimes A)F) = \operatorname{trace}(Q(z)A) \geq 0$$

for all $z \in \mathbb{T}^d$ and all $A \in \mathcal{B}_+(\mathcal{H})$. In other words, if we denote by $\mathcal{C}$ the closed cone generated by (3.1), then $Q(z) \geq 0$ if and only if F belongs to the cone $\mathcal{C}^*$ that is the dual of $\mathcal{C}$. Condition (3.3) tells us now that $\{c_k\}_{k\in\Lambda-\Lambda}$ has a positive extension if and only if $(c_{k-l})_{k,l\in\Lambda} \in (\mathcal{C}^*)^* = \mathcal{C}$. ∎

We observe that the positive measure that corresponds to the moments $\{z^k A\}_{k\in\mathbb{Z}^d}$, $z \in \mathbb{T}$, is the Dirac measure that has value A at z, and 0 elsewhere on the d-torus.

Acknowledgments. We are thankful to Professor Ilya Spitkovsky for useful discussions, and we also thank Professor André Ran for providing us with manuscript [19].

REFERENCES

[1] F. Alizadeh, J.-P. A. Haeberly, M. V. Nayakkankuppam, and M. L. Overton, *SDPpack Version* 0.8 *Beta for Matlab* 4.2: *Semidefinite Programs*, Courant Institute, New York, 1997; available online from http://www.cs.nyu.edu/overton/sdppack/sdppack.html.

[2] W. N. Anderson, Jr., T. D. Morley, and G. E. Trapp, *Positive solutions to* $X = A - BX^{-1}B^*$, Linear Algebra Appl., 134 (1990), pp. 53–62.

[3] T. Ando, *Generalized Schur complements*, Linear Algebra Appl., 27 (1979), pp. 173–186.

[4] D. Z. Arov, *Darlington realization of matrix-valued functions*, Izv. Akad. Nauk SSSR Ser. Mat., 37 (1973), pp. 1299–1331.

[5] D. Z. Arov, *On unitary couplings with losses (scattering theory with losses)*, Funktsional. Anal. i Prilozhen., 8 (1974), pp. 5–22.

[6] D. Z. Arov, *The realization of a canonical system with dissipative boundary conditions at one end of a segment in terms of the coefficient of dynamic flexibility*, Sibirsk. Mat. Zh., 16 (1975), pp. 440–463.

[7] J. A. Ball and J. W. Helton, *Factorization results related to shifts in an indefinite metric*, Integral Equations Operator Theory, 5 (1982), pp. 632–658.

[8] J. A. Ball and J. W. Helton, *Lie groups over the field of rational functions, signed spectral factorization, signed interpolation, and amplifier design*, J. Operator Theory, 8 (1982), pp. 19–64.

[9] M. Bakonyi and G. Naevdal, *On the matrix completion method for multidimensional moment problems*, Acta Sci. Math. (Szeged), 64 (1998), pp. 547–558.

[10] H. Bart, I. Gohberg, and M. A. Kaashoek, *Minimal Factorization of Matrix and Operator Functions*, Oper. Theory Adv. Appl. 1, Birkhäuser Verlag, Basel, Boston, MA, 1979.

[11] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.

[12] A. Calderon and R. Pepinsky, *On the phases of Fourier coefficients for positive real periodic functions*, in Computing Methods and the Phase Problem in X-Ray Crystal Analysis, R. Pepinsky, ed., Pergamon Press, New York, 1950, pp. 339–346.

[13] C. Carathéodory, *Über den Variabilitatsbereich der Fourierschen Konstanten von Positiven Harmonischen Funktionen*, Rend. Circ. Mat. Palermo, 32 (1911), pp. 193–217.

[14] K. Clancey and I. Gohberg, *Factorization of Matrix Functions and Singular Integral Operators*, Oper. Theory Adv. Appl. 3, Birkhäuser Verlag, Basel, 1981.

[15] R. G. Douglas and J. W. Helton, *Inner dilations of analytic matrix functions and Darlington synthesis,* Acta Sci. Math. (Szeged), 34 (1973), pp. 61–67.
[16] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, Macmillan, New York, 1992.
[17] M. A. Dritschel, S. McCullough, and H. J. Woerdeman, *Model theory for ρ-contractions, $\rho \leq 2$*, J. Operator Theory, 41 (1999), pp. 321–350.
[18] L. El Ghaoui, F. Delebecque, and R. Nikoukhah, *LMITool:* -2.0 *Package: An Interface to Solve LMI Problems,* ENSTA, Paris, 1994; available online from http://www.ensta.fr/uer/uma/gropco/lmi/lmitool.html.
[19] S. M. El-Sayed and A. C. M. Ran, *On an iteration method for solving a class of nonlinear matrix equations*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 632–645.
[20] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer, *Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^*X^{-1}A = Q$*, Linear Algebra Appl., 186 (1993), pp. 255–275.
[21] B. A. Francis, *A Course in H_∞ Control Theory,* Lecture Notes in Control and Inform. Sci. 88, Springer-Verlag, Berlin, New York, 1987.
[22] I. Gohberg, *The factorization problem for operator functions*, Izv. Akad. Nauk SSSR Ser. Mat., 28 (1964), pp. 1055–1082.
[23] D.-W. Gu, M.-C. Tsai, S. D. O'Young, and I. Postlethwaite, *I. State-space formulae for discrete-time H^∞ optimization*, Internat. J. Control, 49 (1989), pp. 1683–1723.
[24] H. Helson, *Lectures on Invariant Subspaces*, Academic Press, New York, 1964.
[25] H. Helson and D. Lowdenslager, *Prediction theory and Fourier series in several variables.* I, Acta Math., 99 (1958), pp. 165–202.
[26] H. Helson and D. Lowdenslager, *Prediction theory and Fourier series in several variables.* II, Acta Math., 106 (1961), pp. 175–213.
[27] J. W. Helton, *Orbit structure of the Möbius transformation semigroup acting in H^∞ (broadband matching),* in Topics in Functional Analysis, Adv. in Math. Suppl. Stud. 3, Academic Press, New York, 1978, pp. 129–157.
[28] T. Kailath, *A view of three decades of linear filtering theory*, IEEE Trans. Inform. Theory, IT-20 (1974), pp. 146–181.
[29] A. N. Kolmogorov, *Stationary sequences in Hilbert's space,* Bull. Mosk. Gos. Univ. Mat., 2 (1941), pp. 1–40.
[30] M. G. Krein and A. A. Nudelman, *The Markov Moment Problem and Extremal Problems*, Transl. Math. Monogr. 50, AMS, Providence, RI, 1977.
[31] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford Science Publications, Clarendon Press, Oxford University Press, New York, 1995.
[32] V. M. Popov, *Hyperstability of Control Systems*, Springer-Verlag, New York, 1973.
[33] W. Rudin, *The extension problem of positive definite functions,* Illinois J. Math., 7 (1963), pp. 532–539.
[34] M. Rosenblatt, *A multi-dimensional prediction problem,* Ark. Mat., 3 (1958), pp. 407–424.
[35] M. Rosenblum, *Vectorial Toeplitz operators and the Fejer-Riesz theorem,* J. Math. Anal. Appl., 23 (1968), pp. 139–147.
[36] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Frederick Ungar, New York, 1955.
[37] M. Rosenblum and J. Rovnyak, *Hardy Classes and Operator Theory*, Oxford University Press, Oxford, 1985.
[38] L. Vandenberghe and S. Boyd, *SP, Software for Semidefinite Programming, User's Guide*, 1994, available online from http://www.stanford.edu/~boyd/SP.html.
[39] N. Wiener and P. R. Masani, *The prediction theory of multivariate stochastic processes.* I, Acta Math., 98 (1957), pp. 111–150.
[40] N. Wiener and P. R. Masani, *The prediction theory of multivariate stochastic processes.* II, Acta Math., 99 (1958), pp. 93–137.
[41] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 656–672
© 2001 Society for Industrial and Applied Mathematics

PRODUCTS OF REAL MATRICES WITH PRESCRIBED CHARACTERISTIC POLYNOMIALS*

SUSANA FURTADO†, LAURA IGLÉSIAS‡, AND FERNANDO C. SILVA§

Abstract. Let A be a matrix with entries in the field of real numbers. In this paper we give necessary and sufficient conditions for the existence of real matrices B and C, with prescribed characteristic polynomials, such that $A = BC$.

Key words. eigenvalues, factorization of matrices, real matrices

AMS subject classifications. 15A23, 15A18

PII. S0895479800381690

1. Introduction.

Some papers have been published studying the possibility of writing a square matrix as the product of two matrices with prescribed spectra. Wu [15] proved that any complex singular matrix A can be written as the product of two nilpotent matrices, except when A is a 2×2 nilpotent matrix of rank 1. Laffey [6] and Sourour [12] proved that Wu's theorem is valid in any field F. Sourour [11] proved that if A is a nonscalar nonsingular $n \times n$ matrix over a field F, $b_1, \ldots, b_n$, $c_1, \ldots, c_n \in F$ and $\det A = b_1 \cdots b_n c_1 \cdots c_n$, then there exist matrices $B, C \in F^{n \times n}$, with eigenvalues $b_1, \ldots, b_n$ and $c_1, \ldots, c_n$, respectively, such that $A = BC$. Horn and Johnson [5, Theorem 4.5.4] extended Sourour's theorem to the case where exactly $n - \operatorname{rank} A$ of the elements $b_1, \ldots, b_n, c_1, \ldots, c_n$ are equal to zero. Assuming that A is an $n \times n$ singular matrix over a field F, Sourour and Tang [13] gave a necessary and sufficient condition for the existence of matrices B and C, with prescribed spectra, such that $A = BC$.

In all these papers, the prescribed eigenvalues belong to the field F. An obvious problem is to consider the case where some of those eigenvalues are allowed to be in $\bar{F} \setminus F$, as follows.

PROBLEM 1. *Given $A \in F^{n \times n}$ and monic polynomials $f, g \in F[x]$ of degree n, find necessary and sufficient conditions for the existence of matrices $B, C \in F^{n \times n}$, with characteristic polynomials f and g, respectively, such that $A = BC$.*

This problem seems to be much harder, and the authors were not able to solve it in general fields.

The purpose of this paper is to give a solution when F is the field $\mathbb{R}$ of real numbers.

2. Partial results in arbitrary fields.

Although our purpose is to solve Problem 1 over the field of real numbers, some partial and related results are known to be

*Received by the editors November 28, 2000; accepted for publication (in revised form) by R. Brualdi June 18, 2001; published electronically December 14, 2001. This research was done within the activities of the Centro de Estruturas Lineares e Combinatórias and was partially supported by Fundação para a Ciência e a Tecnologia and Programa Ciência, Tecnologia e Inovação do Quadro Comunitário de Apoio.

http://www.siam.org/journals/simax/23-3/38169.html

†Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, 4200 Porto, Portugal (sbf@fep.up.pt).

‡Departamento de Engenharia Mecânica, Instituto Superior de Engenharia de Lisboa, 1949-014 Lisboa, Portugal (laura@hermite.cii.fc.ul.pt).

§Departamento de Matemática, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal (fcsilva@fc.ul.pt).

valid in arbitrary fields. These results are presented in this section and will be used to prove our main theorem. Let F denote a field.

Suppose that a matrix $A \in F^{n\times n}$ can be written as a product of matrices $B, C \in F^{n\times n}$, with characteristic polynomials f, g, respectively. Then it is easy to see that every matrix $A' \in F^{n\times n}$, similar to A, can be written as a product of matrices $B', C' \in F^{n\times n}$, with characteristic polynomials f, g, respectively. As a square matrix is similar to its transpose, it follows that $A \in F^{n\times n}$ can also be written as a product of matrices $C'', B'' \in F^{n\times n}$, with characteristic polynomials g, f, respectively.

LEMMA 2. *Let $f \in F[x]$ be a monic irreducible polynomial of degree 2. Let $A \in F^{2\times 2}$ be a nilpotent matrix of rank 1. Then there exist no matrices $B, C \in F^{2\times 2}$ such that one of them is nilpotent, the other has characteristic polynomial f, and $A = BC$.*

Proof. In order to get a contradiction, suppose that there exist matrices $B, C \in F^{2\times 2}$ such that one of them is nilpotent, the other has characteristic polynomial f, and $A = BC$. Without loss of generality, suppose that C is a nilpotent matrix of rank 1. Then there exists a nonsingular matrix $X \in F^{2\times 2}$ such that

$$X^{-1}CX = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Suppose that $X^{-1}BX = [b_{i,j}]$. Clearly, $X^{-1}AX$ has eigenvalues $0, b_{2,1}$. Therefore $b_{2,1} = 0$ and B has two eigenvalues in F, which is impossible. □

Wu [15] proved the following lemma when F is the field of complex numbers. Proofs valid in arbitrary fields can be found in Laffey [6] and Sourour [12].

LEMMA 3. *Let $A \in F^{n\times n}$ be a singular matrix. Then there exist nilpotent matrices $B, C \in F^{n\times n}$ such that $A = BC$ if and only if A is not a 2×2 nilpotent matrix of rank 1.*

LEMMA 4. *Let $c_1, \ldots, c_{n-1} \in F$, with $n \geq 3$. Then there exist $e_1, \ldots, e_n \in F^{(n-1)\times 1}$ such that all the invariant factors of*

$$\begin{bmatrix} xI_{n-1} & 0 \end{bmatrix} - \begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix}$$

are constant polynomials and $\begin{bmatrix} e_2 & \cdots & e_n \end{bmatrix}$ has eigenvalues $c_1, \ldots, c_{n-1}$.

Proof. Let $\eta = \#\{i \in \{1, \ldots, n-1\} : c_i = 0\}$. If $\eta > 0$, then suppose, without loss of generality, that $c_{n-\eta} = \cdots = c_{n-1} = 0$.

If $\eta = 0$, let

$$\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} 0 & \operatorname{diag}(c_1, \ldots, c_{n-1}) \end{bmatrix}.$$

If $1 \leq \eta \leq 2$, let

$$\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} 0 & \operatorname{diag}(c_1, \ldots, c_{n-3}) & 0 & 0 \\ 0 & 0 & c_{n-2} & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

If $\eta = 3$, let

$$\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} 0 & \operatorname{diag}(c_1, \ldots, c_{n-\eta-1}) & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

If $\eta \geq 4$ and η is even, let

$$\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} 0 & \operatorname{diag}(c_1, \ldots, c_{n-\eta-1}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{\eta-2}^t & 0 & 0 \end{bmatrix}.$$

If $\eta \geq 5$ and η is odd, let

$$\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} 0 & \operatorname{diag}(c_1, \ldots, c_{n-\eta-1}) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{\eta-4}^t & 0 & 0 \end{bmatrix}. \qquad \square$$

Throughout this paper, we shall make use of some results on matrix completions that we recall in the following lemmas. The first one is the well-known Sá–Thompson theorem [8, 10].

LEMMA 5. *Let $D \in F^{p\times p}$, $E \in F^{n\times n}$, with $p < n$. Let $\delta_1 \mid \cdots \mid \delta_p \in F[x]$ and $\epsilon_1 \mid \cdots \mid \epsilon_n \in F[x]$ be the invariant factors of $xI_p - D$ and $xI_n - E$, respectively. Then there exists a matrix $E' \in F^{n\times n}$ similar to E containing D as a principal submatrix if and only if $\epsilon_i \mid \delta_i$ for every $i \in \{1, \ldots, p\}$, and $\delta_i \mid \epsilon_{i+2n-2p}$ for every $i \in \{1, \ldots, \min\{p, 2p-n\}\}$.*

Zaballa [16] has given a complete description of the possible similarity classes of a square matrix when some rows (or columns) are fixed and the others vary. The next lemma is a particular case of [16, Corollary III]. Lemma 7 is a trivial consequence of [16, Theorem 5.1]. Recall that a matrix $A \in F^{n\times n}$ is nonderogatory if and only if its minimum and characteristic polynomials coincide if and only if $xI_n - A$ has exactly one nonconstant invariant factor.

LEMMA 6. *Let p, q be positive integers, $A_1 \in F^{p\times p}$, $A_2 \in F^{p\times q}$, $n = p + q$. Suppose all the invariant factors of*

$$\begin{bmatrix} xI_p - A_1 & | & -A_2 \end{bmatrix} \tag{2.1}$$

are constant polynomials. Let $D \in F^{n\times n}$ be a nonderogatory matrix. Then there exists a matrix $E \in F^{n\times n}$ similar to D such that $xI_n - E$ contains (2.1) *as a submatrix.*

Given an arbitrary monic polynomial $f \in F[x]$ of degree n, there exists a nonderogatory matrix with characteristic polynomial f, e.g., the companion matrix of f. It follows from Lemma 6 that if all the invariant factors of (2.1) are constant polynomials, then there exists a nonderogatory matrix $E \in F^{n\times n}$ with characteristic polynomial f such that $xI_n - E$ contains (2.1) as a submatrix. (See also [14].)

LEMMA 7. *Let $A_1 \in F^{(n-1)\times(n-1)}$, $A_2 \in F^{(n-1)\times 1}$. Suppose that* (2.1), *with $p = n-1$, has at most one nonconstant invariant factor. Then there exists a nonderogatory matrix $E \in F^{n\times n}$ such that $xI_n - E$ contains $\begin{bmatrix} xI_{n-1} - A_1 & | & -A_2 \end{bmatrix}$ as a submatrix.*

The following lemma is a simple consequence of [17, Theorem 3.1].

LEMMA 8. *Let $E \in F^{n\times n}$ be a nonderogatory matrix. Then there exists $e \in F^{n\times 1}$ such that all the invariant factors of $\begin{bmatrix} xI_n - E & | & -e \end{bmatrix}$ are constant polynomials.*

LEMMA 9. *Let $A \in F^{n\times n}$ be a nonsingular matrix and $c_1, \ldots, c_n \in F$. Then there exist $A', C \in F^{n\times n}$ such that A' is similar to A, C has eigenvalues $c_1, \ldots, c_n$, and $A'C$ is nonderogatory.*

Proof. By induction on n. If $n = 1$, the result is trivial. Suppose that $n \geq 2$. Let $\alpha_1 \mid \cdots \mid \alpha_n$ be the invariant factors of $xI_n - A$. Let $\delta_i = \alpha_i$, $i \in \{1 \ldots, n-2\}$, $\delta_{n-1} = \alpha_{n-1}(x-1)^k$, where k is a nonnegative integer chosen so that $\deg(\delta_1 \cdots \delta_{n-1}) = n-1$. It follows from Lemma 5 that A is similar to a matrix of the form

$$\begin{bmatrix} A_0 & a \\ * & * \end{bmatrix} \in F^{n\times n},$$

where $A_0 \in F^{(n-1)\times(n-1)}$ and $xI_{n-1} - A_0$ has invariant factors $\delta_1 \mid \cdots \mid \delta_{n-1}$. Since 0 is not a root of $\delta_1 \cdots \delta_{n-1}$, A_0 is nonsingular. According to the induction assumption, there exist $X_0, C_0 \in F^{n\times n}$ such that X_0 is nonsingular, C_0 has eigenvalues $c_1, \ldots, c_{n-1}$, and $X_0 A_0 X_0^{-1} C_0$ is nonderogatory. According to Lemma 8, there exists a vector $e \in F^{(n-1)\times 1}$ such that all the invariant factors of

$$\left[\begin{array}{c|c} xI_{n-1} - X_0 A_0 X_0^{-1} C_0 & -e \end{array}\right] \tag{2.2}$$

are constant polynomials. Then

$$C' = \begin{bmatrix} C_0 & X_0 A_0^{-1} X_0^{-1}(e - ac_n X_0) \\ 0 & c_n \end{bmatrix}$$

has eigenvalues $c_1, \ldots, c_n$. On the other hand, A is similar to a matrix of the form

$$A' = \begin{bmatrix} X_0 A_0 X_0^{-1} & X_0 a \\ * & * \end{bmatrix} \in F^{n\times n}.$$

As the invariant factors of (2.2) and $xI_n - A'C'$ interlace [8, 10], it follows that $A'C'$ is nonderogatory. □

LEMMA 10. *Let $A \in F^{n\times n}$ and $c_1, \ldots, c_n \in F$. Suppose that*

$$\#\{i \in \{1, \ldots, n\} : c_i \neq 0\} \leq \operatorname{rank} A.$$

Then there exist $A', B \in F^{n\times n}$ such that A' is similar to A, B is nonderogatory, and $A'B$ has eigenvalues $c_1, \ldots, c_n$.

Proof. The proof is obtained by induction on n. If $n = 1$, the result is trivial. Suppose that $n \geq 2$.

Suppose that A is nonsingular. According to Lemma 9, there exist $D, C \in F^{n\times n}$ such that D is similar to A^{-1}, C has eigenvalues $c_1, \ldots, c_n$, and $B = DC$ is nonderogatory. Then D^{-1} is similar to A and $D^{-1}B$ has eigenvalues $c_1, \ldots, c_n$.

Now suppose that A is singular. Then A is similar to a matrix of the form

$$\begin{bmatrix} A_0 & a \\ 0 & 0 \end{bmatrix} \in F^{n\times n},$$

where $A_0 \in F^{(n-1)\times(n-1)}$. Without loss of generality, suppose that $c_n = 0$. Choose $b \in F^{1\times(n-1)}$ such that $\operatorname{rank}(A_0 + ab) = \operatorname{rank} A$. According to the induction assumption, there exist $X_0, B_0 \in F^{(n-1)\times(n-1)}$ such that X_0 is nonsingular, B_0 is nonderogatory, and $X_0^{-1}(A_0 + ab)X_0 B_0$ has eigenvalues $c_1, \ldots, c_{n-1}$. It follows from the interlacing relations for the invariant factors [8, 10] that $\left[\begin{array}{c|c} xI_{n-1} - B_0^t & -bB_0^t X_0^t \end{array}\right]$ has at most one nonconstant invariant factor. It follows from Lemma 7 that there exists a nonderogatory matrix of the form

$$B = \begin{bmatrix} B_0 & * \\ bX_0 B_0 & * \end{bmatrix} \in F^{n\times n},$$

while A is similar to

$$A' = \begin{bmatrix} X_0^{-1}A_0X_0 & X_0^{-1}a \\ 0 & 0 \end{bmatrix}.$$

Clearly $A'B$ has eigenvalues $c_1, \ldots, c_n$. □

For every positive integer n, let

$$J_n = \begin{bmatrix} 0 & I_{n-1} \\ 0 & 0 \end{bmatrix} \in F^{n\times n}.$$

In particular, $J_1 = 0 \in F$.

The problem considered in this paper is closely related with the problem of describing the possible eigenvalues of AB, where A and B run over prescribed similarity orbits. This question has not been solved, although partial results are known. In the next lemma, we give another partial result that will be used to prove our main theorem. For results analogous to the next lemma, see [7, Theorem 3] and [9, Lemma 3].

LEMMA 11. *Let $A, B \in F^{n\times n}$, $c_1, \ldots, c_n \in F$. Suppose that one of the matrices A, B is singular, the other is nonderogatory, and*

$$\#\{i \in \{1, \ldots, n\} : c_i \neq 0\} \leq \min\{\operatorname{rank} A, \operatorname{rank} B\}.$$

Then there exist matrices $A', B' \in F^{n\times n}$, similar to A, B, respectively, such that $A'B'$ has eigenvalues $c_1, \ldots, c_n$, except if, simultaneously, $n = 2$, $c_1 = c_2 = 0$, one of the matrices A, B is nilpotent of rank 1, and the other does not have eigenvalues in F.

Proof. The exception is a consequence of Lemma 2. Let $c_1, \ldots, c_n \in F$. Note that if A' and B' are similar to A and B, respectively, and $A'B'$ has eigenvalues $c_1, \ldots, c_n$, then A'^t and B'^t are similar to A and B, respectively, and $B'^tA'^t$ has eigenvalues $c_1, \ldots, c_n$. From now on, without loss of generality, suppose that A is singular, B is nonderogatory, and $c_n = 0$.

The proof follows by induction on n. As the case $A = 0$ is trivial, suppose that $A \neq 0$. Suppose that $n = 2$.

- If A has two eigenvalues equal to zero and $c_1 \neq 0$, then there exist $d_1, d_2 \in F$ such that A and B are, respectively, similar to

$$A' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad B' = \begin{bmatrix} 0 & d_1 \\ c_1 & d_2 \end{bmatrix}.$$

- If A has two eigenvalues equal to zero and $c_1 = 0$, then B has two eigenvalues $b_1, b_2 \in F$, and A and B are, respectively, similar to

$$A' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad B' = \begin{bmatrix} b_1 & 1 \\ 0 & b_2 \end{bmatrix}.$$

- If A has eigenvalues $a_1, 0$, with $a_1 \neq 0$, then there exist $d_1, d_2 \in F$ such that A and B are, respectively, similar to

$$A' = \begin{bmatrix} a_1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad B' = \begin{bmatrix} a_1^{-1}c_1 & 1 \\ d_1 & d_2 \end{bmatrix}.$$

In any case, $A'B'$ has eigenvalues $c_1, 0$.

Suppose that $n \geq 3$. As A is singular, $xI_n - A$ has at least one elementary divisor that is a power of x.

Case 1. Suppose that all the elementary divisors of $xI_n - A$ that are powers of x have degree 2.

Subcase 1.1. Suppose that $xI_n - A$ has exactly one elementary divisor equal to x^2. Then A is similar to a matrix of the form $A_0 \oplus J_2$, where A_0 is nonsingular. According to Lemma 10, there exist $A_0', B_0 \in F^{(n-2)\times(n-2)}$ such that A_0' is similar to A_0, B_0 is nonderogatory, and $A_0'B_0$ has eigenvalues $c_1, \ldots, c_{n-2}$. According to Lemma 8, there exists $e \in F^{1\times(n-2)}$ such that all the invariant factors of

$$\begin{bmatrix} xI_{n-2} - B_0 \\ -e \end{bmatrix}$$

are constant polynomials. Then all the invariant factors of

$$\begin{bmatrix} xI_{n-2} - B_0 & 0 \\ -e & x-u \\ 0 & -1 \end{bmatrix},$$

where $u = -1 + \operatorname{trace} B - \operatorname{trace} B_0$, are constant polynomials. According to Lemma 6, B is similar to a matrix of the form

$$B' = \begin{bmatrix} B_0 & 0 & * \\ e & u & v \\ 0 & 1 & 1 \end{bmatrix}$$

for some $v \in F$. Since $e \neq 0$, there exists $V \in F^{(n-2)\times 1}$ such that $v - eV = c_{n-1}$. Let

$$X = \begin{bmatrix} I_{n-2} & 0 & V \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then B' is similar to

$$B'' = XB'X^{-1} = \begin{bmatrix} B_0 & * & h \\ e & u & c_{n-1} \\ 0 & 1 & 1 \end{bmatrix},$$

where $h \in F^{(n-2)\times 1}$. Let

$$Y = \left[\begin{array}{c|cc} I_{n-2} & A_0'^{-1}h & h \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right].$$

Then A is similar to $A' = Y(A_0' \oplus J_2^t)Y^{-1}$ and $A'B''$ has eigenvalues $c_1, \ldots, c_n$.

Subcase 1.2. Suppose that $xI_n - A$ has at least two elementary divisors equal to x^2. Then A is similar to a matrix of the form

$$A' = A_0 \oplus \begin{bmatrix} 0_r & 0_r \\ I_r & 0_r \end{bmatrix},$$

where, if $n = 2r$, the block A_0 vanishes and, if $n > 2r$, then $A_0 \in F^{(n-2r)\times(n-2r)}$ is nonsingular. Without loss of generality, suppose that $c_{n-r+1} = \cdots = c_n = 0$; also suppose that, if $2r < n$,

$$\#\{i \in \{1, \ldots, n-2r\} : c_i \neq 0\} \leq \operatorname{rank} A_0.$$

If there exists $i \in \{n-2r+1, \ldots, n-r\}$ such that $c_i \neq 0$, suppose, without loss of generality, that $c_{n-r} \neq 0$ and let

$$B_{2,3} = \operatorname{diag}(c_{n-2r+1}, \ldots, c_{n-r}) \in F^{r\times r};$$

otherwise, let

$$B_{2,3} = 0_{r-2} \oplus \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \in F^{r\times r}.$$

Let $B_{2,2} = J_r$. Note that $B_{2,3}$ has eigenvalues $c_{n-2r+1}, \ldots, c_{n-r}$ and all the invariant factors of $\begin{bmatrix} xI_r - B_{2,2} & -B_{2,3} \end{bmatrix}$ are constant polynomials.

Subcase 1.2.1. Suppose that $n = 2r$. According to Lemma 6, B is similar to a matrix of the form

$$B' = \begin{bmatrix} B_{2,2} & B_{2,3} \\ * & * \end{bmatrix} \in F^{n\times n}.$$

Then $A'B'$ has eigenvalues $c_1, \ldots, c_n$.

Subcase 1.2.2. Suppose that $n > 2r$. According to Lemma 10, there exist $A'_0, B_{1,1} \in F^{(n-2r)\times(n-2r)}$ such that A'_0 is similar to A_0, $B_{1,1}$ is nonderogatory, and $A'_0B_{1,1}$ has eigenvalues $c_1, \ldots, c_{n-2r}$. According to Lemma 8, there exists $e \in F^{(n-2r)\times 1}$ such that all the invariant factors of

$$\begin{bmatrix} xI_{n-2r} - B_{1,1} & -e \end{bmatrix}$$

are constant polynomials. Let $B_{1,2} = \begin{bmatrix} e & 0 \end{bmatrix} \in F^{(n-2r)\times r}$. Then all the invariant factors of

$$\begin{bmatrix} xI_{n-2r} - B_{1,1} & -B_{1,2} & 0 \\ 0 & xI_r - B_{2,2} & -B_{2,3} \end{bmatrix}$$

are constant polynomials. According to Lemma 6, B is similar to a matrix of the form

$$B' = \begin{bmatrix} B_{1,1} & B_{1,2} & 0 \\ 0 & B_{2,2} & B_{2,3} \\ * & * & * \end{bmatrix} \in F^{n\times n}.$$

On the other hand, A is similar to

$$A'' = A'_0 \oplus \begin{bmatrix} 0_r & 0_r \\ I_r & 0_r \end{bmatrix}$$

and $A''B'$ has eigenvalues $c_1, \ldots, c_n$.

Case 2. Suppose that $xI_n - A$ has at least one elementary divisor of the form x^p, with $p \neq 2$. Then A is similar to a matrix of the form $A_0 \oplus J_p$, where $A_0 \in F^{(n-p)\times(n-p)}$. (When $p = n$, A_0 does not exist.) Without loss of generality, suppose that

$$\begin{aligned} \#\{i \in \{1, \ldots, n-p\} : c_i \neq 0\} &\leq \operatorname{rank} A_0 = \operatorname{rank} A - p + 1, \\ \#\{i \in \{n-p+1, \ldots, n\} : c_i \neq 0\} &\leq p - 1. \end{aligned}$$

Subcase 2.1. Suppose that $p = 1$. According to Lemma 10, there exist $A'_0, B_0 \in F^{(n-1)\times(n-1)}$ such that A'_0 is similar to A_0, B_0 is nonderogatory, and A'_0B_0 has

eigenvalues $c_1, \dots, c_{n-1}$. Then A is similar to $A' = A'_0 \oplus [0]$. According to Lemma 5, B is similar to a matrix of the form

$$B' = \begin{bmatrix} B_0 & * \\ * & * \end{bmatrix} \in F^{n\times n}.$$

Clearly, $A'B'$ has eigenvalues $c_1, \dots, c_n$.

Subcase 2.2. Suppose that $p \geq 3$. According to Lemma 4, there exist $e_1, \dots, e_p \in F^{(p-1)\times 1}$ such that all the invariant factors of $\begin{bmatrix} xI_{p-1} & 0 \end{bmatrix} - \begin{bmatrix} e_1 & \cdots & e_p \end{bmatrix}$ are constant and $\begin{bmatrix} e_2 & \cdots & e_p \end{bmatrix}$ has eigenvalues $c_{n-p+1}, \dots, c_{n-1}$.

Subcase 2.2.1. Suppose that $p = n$. Then A is similar to $A' = J_p^t$. According to Lemma 6, B is similar to a matrix of the form

$$B' = \left[\begin{array}{ccc} e_1 & \cdots & e_n \\ \hline & * & \end{array}\right] \in F^{n\times n}.$$

Then $A'B'$ has eigenvalues $c_1, \dots, c_n$.

Subcase 2.2.2. Suppose that $p < n$. According to Lemma 10, there exist $A'_0, B_0 \in F^{(n-p)\times(n-p)}$ such that A'_0 is similar to A_0, B_0 is nonderogatory, and A'_0B_0 has eigenvalues $c_1, \dots, c_{n-p}$. According to Lemma 8, there exists $e \in F^{(n-p)\times 1}$ such that all the invariant factors of $\left[\begin{array}{c|c} xI_{n-p} - B_0 & -e \end{array}\right]$ are constant polynomials. Let $B_1 = \begin{bmatrix} e & 0 \end{bmatrix} \in F^{(n-p)\times p}$. Bearing in mind the Kronecker canonical form (cf. [3]), we see that $\begin{bmatrix} xI_{p-1} & 0 \end{bmatrix} - \begin{bmatrix} e_1 & \cdots & e_p \end{bmatrix}$ is strictly equivalent to $\begin{bmatrix} xI_{p-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & I_{p-1} \end{bmatrix}$. It follows easily that there exists a nonsingular matrix

$$X = \begin{bmatrix} X_0 & 0 \\ * & * \end{bmatrix} \in F^{p\times p},$$

where $X_0 \in F^{(p-1)\times(p-1)}$, such that

$$X_0^{-1} \begin{bmatrix} e_1 & \cdots & e_p \end{bmatrix} X = \begin{bmatrix} 0 & I_{p-1} \end{bmatrix}.$$

Then all the invariant factors of

$$\begin{bmatrix} xI_{n-1} & 0 \end{bmatrix} - \left[\begin{array}{c|c} B_0 & B_1X^{-1} \\ \hline 0 & \begin{matrix} e_1 & \cdots & e_p \end{matrix} \end{array}\right]$$

are constant polynomials. According to Lemma 6, B is similar to a matrix of the form

$$B' = \left[\begin{array}{c|c} B_0 & B_1X^{-1} \\ \hline 0 & \begin{matrix} e_1 & \cdots & e_p \end{matrix} \\ \hline * & * \end{array}\right] \in F^{n\times n}.$$

Then A is similar to $A' = A'_0 \oplus J_p^t$ and $A'B'$ has eigenvalues $c_1, \dots, c_n$. □

LEMMA 12. *Let $A \in F^{n\times n}$ be a singular matrix and $f \in F[x]$ be a monic polynomial of degree n such that $f(0) \neq 0$. Let $c_1, \dots, c_n \in F$. Suppose that*

$$\#\{i \in \{1, \dots, n\} : c_i \neq 0\} \leq \operatorname{rank} A.$$

Then there exist $B, C \in F^{n\times n}$ such that B has characteristic polynomial f, C has eigenvalues $c_1, \dots, c_n$, and $A = BC$, except if, simultaneously, A is a 2×2 nilpotent matrix of rank 1, f is irreducible, and $c_1 = c_2 = 0$.

Proof. The exception is a consequence of Lemma 2. If $A = 0$, the result is trivial. Suppose that $A \neq 0$ and it is false that, simultaneously, A is a 2×2 nilpotent matrix of rank 1, f is irreducible, and $c_1 = c_2 = 0$. Let $D \in F^{n\times n}$ be a nonderogatory matrix with characteristic polynomial f. Then D^{-1} is also a nonderogatory matrix. Moreover the characteristic polynomial of D^{-1} is reducible whenever f is reducible. According to Lemma 11, there exists a nonsingular matrix $X \in F^{n\times n}$ and a matrix $E \in F^{n\times n}$ similar to D^{-1} such that $C = EX^{-1}AX$ has eigenvalues $c_1, \ldots, c_n$. Then $A = (XE^{-1}X^{-1})(XCX^{-1})$. ∎

LEMMA 13. *Let $A \in F^{n\times n}$, $n \geq 2$, be a singular matrix and $f \in F[x]$ be a monic polynomial of degree n such that $f \neq x^n$ and $f(0) = 0$. Then there exist $B, C \in F^{n\times n}$ such that B has characteristic polynomial f, C is nilpotent, and $A = BC$.*

Proof. By induction on n. If $A = 0$, the result is trivial. Suppose that $A \neq 0$. Suppose that $n = 2$. Since A is singular, A is similar to

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

for some $a \in F$. Suppose that $f = x(x - b)$, with $b \in F \setminus \{0\}$. Then

$$A' = \begin{bmatrix} b & 0 \\ ab & 0 \end{bmatrix} \begin{bmatrix} 0 & b^{-1} \\ 0 & 0 \end{bmatrix}.$$

Suppose that $n > 2$. Let s be the number of eigenvalues of A equal to 0 and let r be the number of roots of f equal to 0. Suppose that $f = x^r h$.

Case 1. Suppose that A is similar to J_3. Suppose that $f = x(x^2 - bx - a)$. Let

$$B = \begin{bmatrix} b & a & -ab+1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & b & -b^2 \\ 0 & 1 & -b \end{bmatrix}.$$

Clearly, B has characteristic polynomial f, C is nilpotent, and A is similar to BC.

Case 2. Suppose that $r \geq 2$ and $s \geq 2$. Then A is similar to a matrix of the form

$$A' = \begin{bmatrix} 0 & 0 & a_1 \\ 0 & 0 & 0 \\ 0 & a_2 & A_0 \end{bmatrix},$$

where $a_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in F^{1\times(n-2)}$ and $a_2 \in F^{(n-2)\times 1}$. (See [2, Lemma 3].) Let $B_{1,1} = C_{1,1} = J_2$. Choose $l \in F^{(n-2)\times 1}$ such that $A_0 - \begin{bmatrix} l & 0 \end{bmatrix}$ is singular and is not a nilpotent matrix of rank 1. Let $B_{2,1} = \begin{bmatrix} a_2 & l \end{bmatrix}$ and

$$C_{1,2} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in F^{2\times(n-2)}.$$

According to either the induction assumption or Lemma 12, there exist $B_{2,2}$, $C_{2,2} \in F^{(n-2)\times(n-2)}$ such that $B_{2,2}$ has characteristic polynomial $x^{r-2}h$, $C_{2,2}$ is nilpotent, and $A_0 - B_{2,1}C_{1,2} = B_{2,2}C_{2,2}$. Then $A' = BC$, where

$$B = \begin{bmatrix} B_{1,1} & 0 \\ B_{2,1} & B_{2,2} \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ 0 & C_{2,2} \end{bmatrix}. \tag{2.3}$$

Case 3. Suppose that $s > r = 1$ and Case 1 is not satisfied. In this case, A is similar to a matrix of the form

$$A' = \begin{bmatrix} 0 & 0 \\ A_{2,1} & A_{2,2} \end{bmatrix},$$

where $A_{2,2} \in F^{(n-1)\times(n-1)}$ is singular and is not similar to J_2. According to Lemma 12, there exist $B_{2,2} \in F^{(n-1)\times(n-1)}$ with characteristic polynomial h and a nilpotent matrix $C_{2,2} \in F^{(n-1)\times(n-1)}$ such that $A_{2,2} = B_{2,2}C_{2,2}$. Let $C_{2,1} = B_{2,2}^{-1}A_{2,1}$. Then $A' = BC$, where

$$B = \begin{bmatrix} 0 & 0 \\ 0 & B_{2,2} \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} 0 & 0 \\ C_{2,1} & C_{2,2} \end{bmatrix}.$$

Case 4. Suppose that $r \geq s = 1$. In this case, there exists a nonsingular matrix $A_{2,2} \in F^{(n-1)\times(n-1)}$ such that, for every $A_{1,2} \in F^{1\times(n-1)}$, A is similar to

$$\begin{bmatrix} 0 & A_{1,2} \\ 0 & A_{2,2} \end{bmatrix}.$$

Choose $B_{2,1} \in F^{(n-1)\times 1}$ and $C_{1,2} \in F^{1\times(n-1)}$ so that $A_{2,2} - B_{2,1}C_{1,2}$ is singular and not similar to J_2. According to either the induction assumption or Lemma 12, there exist $B_{2,2}, C_{2,2} \in F^{(n-1)\times(n-1)}$ such that $B_{2,2}$ has characteristic polynomial $x^{r-1}h$, $C_{2,2}$ is nilpotent, and $A_{2,2} - B_{2,1}C_{1,2} = B_{2,2}C_{2,2}$. Then A is similar to BC, where B and C have the forms (2.3), with $B_{1,1} = C_{1,1} = 0$. □

LEMMA 14. *Let $A \in F^{n\times n}$ be a nonzero singular matrix, let $g \in F[x]$ be a monic polynomial of degree n, and let $b \in F \setminus \{0\}$. Then there exist $B, C \in F^{n\times n}$ such that B has characteristic polynomial $f = (x-b)x^{n-1}$, C has characteristic polynomial g, and $A = BC$.*

Proof. If $g = x^n$, the result has been proved in Lemma 13. If $g(0) \neq 0$, the result follows from Lemma 12. Now suppose that $g \neq x^n$ and $g(0) = 0$. The proof continues by induction on n.

Suppose that $n = 2$. Then A is similar to

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

for some $a \in F$, $B = \operatorname{diag}(b, 0)$ has characteristic polynomial f, there exist $u, v \in F$ such that

$$C = \begin{bmatrix} b^{-1}a & b^{-1} \\ u & v \end{bmatrix}$$

has characteristic polynomial g, and $A' = BC$.

Now suppose that $n > 2$. Suppose that $g = xh$. As A is singular, A is similar to a matrix of the form

$$A' = \begin{bmatrix} A_0 & 0 \\ a & 0 \end{bmatrix},$$

where $A_0 \in F^{(n-1)\times(n-1)}$ and $a \in F^{1\times(n-1)}$.

Case 1. Suppose that $a = 0$. Choose $k \in F^{(n-1)\times 1}$ and $l \in F^{1\times(n-1)}$ such that $A_0 - kl$ is a nonzero singular matrix. According to the induction assumption, there

exist $B_0, C_0 \in F^{(n-1)\times(n-1)}$ such that B_0 has characteristic polynomial $(x-b)x^{n-2}$, C_0 has characteristic polynomial h, and $A_0 - kl = B_0C_0$. Then A is similar to BC, where B and C have the forms

$$B = \begin{bmatrix} B_0 & k \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} C_0 & 0 \\ l & 0 \end{bmatrix}$$

and have characteristic polynomials f and g, respectively.

Case 2. Suppose that $a \neq 0$. Choose $l \in F^{1\times(n-1)}$ such that $A_0 - lb^{-1}a$ is singular and is not a nilpotent matrix of rank 1. According to either Lemma 12 or Lemma 13, there exist $B_0, C_0 \in F^{(n-1)\times(n-1)}$ such that B_0 is nilpotent, C_0 has characteristic polynomial h, and $A_0 - lb^{-1}a = B_0C_0$. Then $A' = BC$, where B and C have the forms

$$B = \begin{bmatrix} B_0 & l \\ 0 & b \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} C_0 & 0 \\ b^{-1}a & 0 \end{bmatrix}$$

and have characteristic polynomials f and g, respectively. □

LEMMA 15. *Let $A \in F^{n\times n}$ and let $f, g \in F[x]$ be monic polynomials of degree n. If there exist $B, C \in F^{n\times n}$ such that B has characteristic polynomial f, C has characteristic polynomial g, and $A = BC$, then $x^{n-\operatorname{rank} A}$ divides fg.*

Proof. As $A = BC$, we have $\operatorname{rank} A \geq \operatorname{rank} B + \operatorname{rank} C - n$. Therefore $\eta \leq \nu + \mu$, where $\eta = n - \operatorname{rank} A$, $\nu = n - \operatorname{rank} B$, $\mu = n - \operatorname{rank} C$ are the numbers of elementary divisors of $xI_n - A, xI_n - B, xI_n - C$, respectively, that are powers of x. Hence $x^\eta \mid x^{\nu+\mu} = x^\nu x^\mu \mid fg$. □

LEMMA 16. *Let $A \in F^{n\times n}$ and let $f, g \in F[x]$ be monic polynomials of degree n. Suppose that $n > 2$ and $\operatorname{rank} A \leq 1$. If $x^{n-\operatorname{rank} A}$ divides fg, then there exist $B, C \in F^{n\times n}$ such that B has characteristic polynomial f, C has characteristic polynomial g, and $A = BC$.*

Proof. First, suppose that $A = 0$. In this case, x^n divides fg. Suppose that $f = x^k f_1$ and $g = x^{n-k}g_1$. Let $B_0 \in F^{(n-k)\times(n-k)}$ and $C_0 \in F^{k\times k}$ be matrices with characteristic polynomials f_1 and g_1, respectively. Then $A = (0_k \oplus B_0)(C_0 \oplus 0_{n-k})$.

Now suppose that $\operatorname{rank} A = 1$. Let $x^{n-1}(x-a)$, where $a \in F$, be the characteristic polynomial of A. As x^{n-1} divides fg, we have $f = x^k f_1$ and $g = x^l g_1$ for some nonnegative integers k, l such that $k+l = n-1$. We assume, without loss of generality, that $k \leq l$.

Case 1. Suppose that $k = 0$. Then $l = n-1$ and $g = x^{n-1}(x-c)$ for some $c \in F$. It follows from Lemma 6 that there exists a matrix of the form

$$B = \begin{bmatrix} 0 & a & * & * \\ 1 & 0 & * & * \\ 0 & 1 & * & * \\ 0 & 0 & * & * \end{bmatrix} \in F^{n\times n}$$

with characteristic polynomial f. On the other hand,

$$C = \begin{bmatrix} c & 0 \\ 1 & 0 \end{bmatrix} \oplus 0_{n-2}$$

has characteristic polynomial g. Then A is similar to BC.

Case 2. Suppose that $k > 0$. It follows from Lemma 6 that there exist matrices of the forms

$$B = \begin{bmatrix} 0_k & 0 & 0 & 0 \\ 0 & 1 & * & * \\ 0 & 1 & * & * \\ 0 & 0 & * & * \end{bmatrix}, \quad C = \begin{bmatrix} * & * & * & 0 \\ * & * & * & 0 \\ 0 & 1 & a & 0 \\ 0 & 0 & 0 & 0_l \end{bmatrix} \in F^{n\times n}$$

with characteristic polynomials f and g, respectively. Then A is similar to

$$BC = 0_{k-1} \oplus \begin{bmatrix} 0 & 0 & 0 \\ 1 & a & 0 \\ 1 & a & 0 \end{bmatrix} \oplus 0_{n-k-2}. \qquad \square$$

LEMMA 17. *Let $A \in F^{2\times 2}$ be a singular matrix and let $f, g \in F[x]$ be monic polynomials of degree* 2. *Then there exist $B, C \in F^{2\times 2}$ such that B has characteristic polynomial f, C has characteristic polynomial g, and $A = BC$ if and only if one of the following conditions is satisfied:*

(a_{17}) *$A = 0$ and x^2 divides fg;*

(b_{17}) *A is nilpotent of rank* 1, *$(fg)(0) = 0$, and at least one of the polynomials f, g is reducible and different from x^2;*

(c_{17}) *A is not nilpotent and $(fg)(0) = 0$.*

Proof. Necessity. If $A = 0$, the necessity of (a_{17}) has already been proved in Lemma 15. As A is singular, the necessity of the condition $(fg)(0) = 0$ is trivial. If A is nilpotent of rank 1, the necessity of (b_{17}) follows from Lemmas 2 and 3.

Sufficiency. Suppose that (a_{17}) is satisfied. If x^2 divides f, let $B = 0$ and let C be any matrix with characteristic polynomial g. If x^2 divides g, let B be any matrix with characteristic polynomial f and let $C = 0$. If $f = x(x - b)$ and $g = x(x - c)$, with $b, c \in F$, then $A = 0 = \operatorname{diag}(b, 0) \operatorname{diag}(0, c)$.

Now suppose that $A \neq 0$. If one of the polynomials f, g has the form $x(x-b)$, with $b \in F \setminus \{0\}$, then the proof follows from Lemma 14. Otherwise, one of the polynomials f, g is equal to x^2. Without loss of generality, suppose that $f = x^2$. If A is nilpotent of rank 1, then $g = (x - c_1)(x - c_2)$, with $c_1 \in F$, $c_2 \in F \setminus \{0\}$, the matrices $B = J_2$ and $C = \operatorname{diag}(c_1, c_2)$ have characteristic polynomials f and g, respectively, and BC is similar to A. If A is not nilpotent, then A has characteristic polynomial $x(x - a)$, with $a \in F \setminus \{0\}$, $B = J_2$ has characteristic polynomial f, there exist $c_1, c_2 \in F$ such that

$$C = \begin{bmatrix} 0 & c_1 \\ a & c_2 \end{bmatrix}$$

has characteristic polynomial g, and BC is similar to A. $\square$

LEMMA 18. *Let $A \in F^{2\times 2}$ be a nonscalar nonsingular matrix and let $f, g \in F[x]$ be monic polynomials of degree* 2. *Suppose that at least one of the polynomials f, g is reducible. Then there exist $B, C \in F^{2\times 2}$ such that B has characteristic polynomial f, C has characteristic polynomial g, and $A = BC$ if and only if* $\det A = (fg)(0)$.

Proof. The necessity of the condition $\det A = (fg)(0)$ is trivial. In order to prove the sufficiency, suppose, without loss of generality, that f is reducible. Let $C \in F^{2\times 2}$ be a nonderogatory matrix with characteristic polynomial g. According to [9], there exist matrices $A', D \in F^{2\times 2}$, similar to A, C^{-1}, respectively, such that $B = A'D$ has characteristic polynomial f. Then $A' = BD^{-1}$, where B and D^{-1} have characteristic polynomials f and g, respectively. $\square$

3. Main result. In this section, we solve Problem 1 when F is the field $\mathbb{R}$ of real numbers. The case $A = 0$ has already been considered in Lemmas 15, 16, and 17 and the nonsingular scalar case is trivial. From now on, we shall assume that A is nonscalar. We shall consider separately the case $n = 2$ since it has a more irregular solution.

LEMMA 19. *Suppose that $f = x^2 + bx + 1$ and $g = x^2 + cx + 1$ are real irreducible polynomials. Let $A \in \mathbb{R}^{2\times 2}$ be a nonscalar matrix. Then there exist matrices $B, C \in \mathbb{R}^{2\times 2}$ such that B and C have characteristic polynomials f and g, respectively, and $A = BC$ if and only if* $\det A = 1$, $\operatorname{trace} A$ *belongs to*

$$\left(-\infty, \frac{1}{2}\left(cb - \sqrt{4-b^2}\sqrt{4-c^2}\right)\right] \cup \left[\frac{1}{2}\left(cb + \sqrt{4-b^2}\sqrt{4-c^2}\right), +\infty\right), \tag{3.1}$$

and

- $b \neq c$ *or* $\operatorname{trace} A \neq 2$,
- $b \neq -c$ *or* $\operatorname{trace} A \neq -2$.

Proof. As f and g are irreducible, it follows that $\max\{|b|, |c|\} < 2$. Note that

$$-2 \leq \frac{1}{2}\left(cb - \sqrt{4-b^2}\sqrt{4-c^2}\right) < \frac{1}{2}\left(cb + \sqrt{4-b^2}\sqrt{4-c^2}\right) \leq 2.$$

Necessary condition. Let $A', B', C' \in \mathbb{R}^{2\times 2}$ be matrices simultaneously similar to A, B, C, respectively, where C' is the companion matrix of g. Then

$$B' = \begin{bmatrix} -b-r & s \\ (-br-r^2-1)/s & r \end{bmatrix}, \qquad C' = \begin{bmatrix} 0 & 1 \\ -1 & -c \end{bmatrix} \tag{3.2}$$

for some $r, s \in \mathbb{R}$, $s \neq 0$, $A' = B'C'$, and $\operatorname{trace} A = \operatorname{trace} A' = -s - rc + (-br - r^2 - 1)/s$. The real map $t(r, s) = -s - rc + (-br - r^2 - 1)/s$, with domain $\mathbb{R} \times (\mathbb{R} \setminus \{0\})$, has range given by (3.1).

Suppose that $b = c$ and $\operatorname{trace} A = 2$. Then the equation $t(r, s) = 2$ has a unique real solution: $r = 0$ and $s = -1$. By replacing these values in (3.2), we deduce that $A' = B'C'$ is scalar, which is a contradiction. Analogously, it is impossible that $b = -c$ and $\operatorname{trace} A = -2$.

Sufficient condition. With the previous notation, choose $(r, s) \in \mathbb{R} \times (\mathbb{R} \setminus \{0\})$ such that $t(r, s) = \operatorname{trace} A$. Let B', C' be the matrices (3.2) and let $A' = B'C'$. Since A and A' have the same trace and the same determinant and are nonscalar, A and A' are similar. Because B', C' have characteristic polynomials f and g, respectively, the proof is complete. ∎

The next lemma is easy to prove and, jointly with Lemmas 17, 18, and 19, completes the study of the case $n = 2$.

LEMMA 20. *Let $f = x^2 + b_1x + b_0$ and $g = x^2 + c_1x + c_0$ be real irreducible polynomials and let $A \in \mathbb{R}^{2\times 2}$ be a nonscalar matrix. Then there exist matrices $B, C \in \mathbb{R}^{2\times 2}$ such that B and C have characteristic polynomials f and g, respectively, and $A = BC$ if and only if there exist matrices $B', C' \in \mathbb{R}^{2\times 2}$ such that B' and C' have characteristic polynomials $f' = x^2 + (b_1/\sqrt{b_0})x + 1$ and $g' = x^2 + (c_1/\sqrt{c_0})x + 1$, respectively, and $(1/\sqrt{b_0c_0})A = B'C'$.*

COROLLARY 21. *Let $f = x^2 + b_1x + b_0$ and $g = x^2 + c_1x + c_0$ be real irreducible polynomials and let $A \in \mathbb{R}^{2\times 2}$ be a nonscalar matrix with real eigenvalues. Then there exist matrices $B, C \in \mathbb{R}^{2\times 2}$ such that B and C have characteristic polynomials f and g, respectively, and $A = BC$ if and only if* $\det A = b_0c_0$ *and*

- $b_1\sqrt{c_0} \neq c_1\sqrt{b_0}$ *or* $\operatorname{trace} A \neq 2\sqrt{b_0c_0}$,
- $b_1\sqrt{c_0} \neq -c_1\sqrt{b_0}$ *or* $\operatorname{trace} A \neq -2\sqrt{b_0c_0}$.

Proof. First suppose that $b_0 = c_0 = 1$. The necessity has been stated in Lemma 19.

Let us prove the sufficiency. Since A has real eigenvalues and $\det A = 1$, it follows that $|\operatorname{trace} A| \geq 2$. According to a remark in the proof of Lemma 19, $\operatorname{trace} A$ belongs to (3.1). Hence the proof is a consequence of Lemma 19.

The general case reduces to the case $b_0 = c_0 = 1$ by applying Lemma 20. ∎

LEMMA 22. *Let $A \in \mathbb{R}^{n\times n}$ be a nonscalar matrix, let T be a finite subset of $\mathbb{R}$ such that $0 \notin T$, and let $d \in \mathbb{R} \setminus \{0\}$. Suppose that $n \geq 3$ and $xI_n - A$ has at most $n-2$ nonconstant invariant factors. Then A is similar to a matrix of the form*

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $A_{1,1} \in \mathbb{R}^{2\times 2}$ is nonscalar and has real eigenvalues, $\det A_{1,1} = d$, $\operatorname{trace} A_{1,1} \notin T$,

- $A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$ *has real eigenvalues and* $\operatorname{trace}(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}) \notin T$ *if* $n = 4$,
- $A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$ *is not a nonsingular scalar matrix if* $n \geq 4$.

Proof. Suppose that $n = 3$. Choose $t \in \mathbb{R}\setminus T$ such that the polynomial $x^2 - tx + d$ has real roots. As $xI_3 - A$ has exactly one nonconstant invariant factor, A is similar to a companion matrix and, therefore, is similar to a matrix of the form

$$\begin{bmatrix} 1 & 0 & -d \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & * \\ 1 & 0 & * \\ 0 & 1 & * \end{bmatrix} \begin{bmatrix} 1 & 0 & d \\ 0 & 1 & -t \\ 0 & 0 & 1 \end{bmatrix},$$

which has the prescribed form.

Suppose that $n \geq 4$. Due to [16, Corollary I] and the fact that $xI_n - A$ has at most $n-2$ nonconstant invariant factors, A is similar to a matrix of the form

$$D = \begin{bmatrix} 0 & D_{1,2} & D_{1,3} \\ I_2 & D_{2,2} & D_{2,3} \\ 0 & D_{3,2} & D_{3,3} \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $D_{2,2}$ and $D_{3,3}$ are square blocks.

Case 1. Suppose that $n = 4$.

- If $D_{1,2}$ is nonscalar, choose $\lambda_1, \lambda_2 \in \mathbb{R}$ such that $\lambda_1 \neq \lambda_2$, $\lambda_1\lambda_2 = d$, and $\lambda_1 + \lambda_2 \notin T$. Choose $\mu_1, \mu_2 \in \mathbb{R}$ such that $\mu_1 \neq \mu_2$, $-\det D_{1,2} = \lambda_1\lambda_2\mu_1\mu_2$ and $\mu_1 + \mu_2 \notin T$. According to either Sourour's theorem [11] or Lemma 12, there exist $L, M \in \mathbb{R}^{2\times 2}$ with eigenvalues λ_1, λ_2 and μ_1, μ_2, respectively, such that $-D_{1,2} = LM$.
- If $D_{1,2} = aI_2$, with $a \in \mathbb{R}$, choose $\lambda_1, \lambda_2 \in \mathbb{R}$ such that $\lambda_1 \neq \lambda_2$, $\lambda_1\lambda_2 = d$, $\lambda_1 + \lambda_2 \notin T$, and $-a\lambda_1^{-1} - a\lambda_2^{-1} \notin T$. Let $L = \operatorname{diag}(\lambda_1, \lambda_2)$ and $M = \operatorname{diag}(-a\lambda_1^{-1}, -a\lambda_2^{-1})$. Then $-D_{1,2} = LM$.

In any case, A is similar to

$$\begin{bmatrix} I_2 & L \\ 0 & I_2 \end{bmatrix} D \begin{bmatrix} I_2 & -L \\ 0 & I_2 \end{bmatrix},$$

which has the prescribed form.

Case 2. Suppose that $n \geq 5$. Let

$$X = \begin{bmatrix} I_2 & A_{1,1} \\ 0 & I_2 \end{bmatrix} \oplus I_{n-4},$$

where $A_{1,1} \in \mathbb{R}^{2\times 2}$ is a nonscalar matrix with real eigenvalues such that $\det A_{1,1} = d$ and $\operatorname{trace} A_{1,1} \notin T$. If XDX^{-1} has already the prescribed form, the proof is complete. Otherwise, it follows that $D_{3,3} = aI_{n-4}$ for some $a \in \mathbb{R} \setminus \{0\}$. Choose $W \in \mathbb{R}^{2\times(n-4)} \setminus \{0\}$. Let

$$Y = \begin{bmatrix} I_2 & 0 & W \\ 0 & I_2 & 0 \\ 0 & 0 & I_{n-4} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Then $YXDX^{-1}Y^{-1}$ has the prescribed form. □

The next theorem is our main result.

THEOREM 23. *Let $A \in \mathbb{R}^{n\times n}$ be a nonscalar matrix. Let $f, g \in \mathbb{R}[x]$ be monic polynomials of degree n, where $n \geq 3$.*

Then there exist $B, C \in \mathbb{R}^{n\times n}$ such that B has characteristic polynomial f, C has characteristic polynomial g, and $A = BC$ if and only if $(fg)(0) = \det A$ and $x^{n-\operatorname{rank} A}$ divides fg.

Proof. The necessity of the first condition is trivial and the necessity of the second condition has been proved in Lemma 15.

From now on, suppose that $(fg)(0) = \det A$ and $x^{n-\operatorname{rank} A}$ divides fg in order to prove the sufficiency. The proof continues by induction on n. If x^{n-1} divides one of the polynomials f, g, then the proof follows from Lemmas 3, 12, 13, and 14. The case $\operatorname{rank} A \leq 1$ has already been considered in Lemma 16. From now on, suppose that $\operatorname{rank} A > 1$ and x^{n-1} divides neither f nor g.

Case 1. Suppose that $xI_n - A$ has exactly $n-1$ nonconstant invariant factors. Then A is similar to

$$A' = [a] \oplus \begin{bmatrix} a & 1 \\ 0 & b \end{bmatrix} \oplus aI_{n-3}$$

for some $a, b \in \mathbb{R}$, with $a \neq 0$. It follows from Lemma 6 that there exist matrices

$$B = \begin{bmatrix} * & aI_{n-1} \\ * & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & * \\ I_{n-1} & * \end{bmatrix} \in \mathbb{R}^{n\times n},$$

with characteristic polynomials f and g, respectively. As $\det(BC) = \det A = a^{n-1}b$,

$$BC = \begin{bmatrix} aI_{n-1} & * \\ 0 & b \end{bmatrix}.$$

If BC is nonscalar, then BC is similar to A and the proof is complete. Now suppose that $BC = aI_n$. Let $X \in \mathbb{R}^{n\times n}$ be a nonsingular matrix such that

$$X^{-1}BX = \begin{bmatrix} B_1 & & * \\ & \ddots & \\ 0 & & B_p \end{bmatrix},$$

where $p = n/2$ and the blocks $B_1, \ldots, B_p$ are of size 2×2 if n is even, and $p = (n+1)/2$ and the blocks $B_1, \ldots, B_{p-1}$ are of size 2×2 if n is odd. Note that $B_1 \oplus \cdots \oplus B_p$

and $aB_1^{-1} \oplus \cdots \oplus aB_p^{-1}$ have characteristic polynomials f and g, respectively. Let $e \in \mathbb{R}^{2\times k}$ be a matrix of rank 1, where $k = 2$ if n is even, and $k = 1$ if n is odd. Then

$$B' = B_1 \oplus \cdots \oplus B_{p-2} \oplus \begin{bmatrix} B_{p-1} & e \\ 0 & B_p \end{bmatrix} \quad \text{and} \quad C' = aB_1^{-1} \oplus \cdots \oplus aB_p^{-1}$$

have characteristic polynomials f and g, respectively, and $B'C'$ is similar to A.

Case 2. Suppose that $xI_n - A$ has at most $n-2$ nonconstant invariant factors. Suppose that $f = f_1 f_2$ and $g = g_1 g_2$, where $f_1 = x^2 - a_1 x - a_0$ and $g_1 = x^2 - b_1 x - b_0$, with $a_0 b_0 \neq 0$. Then A is similar to a matrix of the form described in Lemma 22, with $d = a_0 b_0$. Note that

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} I_2 & -A_{1,1}^{-1}A_{1,2} \\ 0 & I_{n-2} \end{bmatrix} = \begin{bmatrix} A_{1,1} & 0 \\ A_{2,1} & A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2} \end{bmatrix}.$$

Therefore

$$\det A = (\det A_{1,1})(\det(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}))$$

and

$$\operatorname{rank}(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}) = \operatorname{rank} A - 2.$$

According to either Lemma 18 or Corollary 21, there exist $B_{1,1}, C_{1,1} \in \mathbb{R}^{2\times 2}$ such that $B_{1,1}$ has characteristic polynomial f_1, $C_{1,1}$ has characteristic polynomial g_1, and $A_{1,1} = B_{1,1}C_{1,1}$. Let $C_{1,2} = B_{1,1}^{-1}A_{1,2}$ and $B_{2,1} = A_{2,1}C_{1,1}^{-1}$. On the other hand, there exist $B_{2,2}, C_{2,2} \in \mathbb{R}^{(n-2)\times(n-2)}$ such that $B_{2,2}$ has characteristic polynomial f_2, $C_{2,2}$ has characteristic polynomial g_2, and $A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2} = B_{2,2}C_{2,2}$. This last conclusion follows from the induction assumption when $n \geq 5$, follows from either Lemma 18 or Corollary 21 when $n = 4$, and is trivial when $n = 3$. Then A is similar to BC, where B and C have the forms (2.3). □

Recently, Freitas [1, Proposition 5.2] has shown that Sourour's theorem can be used to describe possible eigenvalues of the quadratic polynomial $x^2 I_n + xB + C$, where $C \in F^{n\times n}$ is a fixed square matrix and B varies on $F^{n\times n}$. Freitas's argument runs as follows.

Suppose that Sourour's theorem (or any of its generalizations) allows us to write C as the product of two matrices M and L with characteristic polynomials f and g, respectively. Then

$$\begin{bmatrix} 0 & I_n \\ -C & M+L \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ L & I_n \end{bmatrix} \begin{bmatrix} L & I_n \\ 0 & M \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -L & I_n \end{bmatrix}$$

has characteristic polynomial fg. It follows from [4, Theorem 1.1] that the eigenvalues of $x^2 I_n - x(M+L) + C$ are the roots of fg.

REFERENCES

[1] P. Freitas, *Spectral sequences for quadratic pencils and the inverse spectral problem for the damped wave equation*, J. Math. Pures Appl., 78 (1999), pp. 965–980.

[2] S. Furtado, L. Iglésias, and F. C. Silva, *Products of Matrices with Prescribed Spectra and Ranks*, preprint.

[3] F. R. Gantmacher, *The Theory of Matrices*, Vol. 2, Chelsea Publishing, New York, 1960.

[4] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.
[5] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[6] T. J. Laffey, *Products of matrices*, in Generators and Relations in Groups and Geometries, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci. 333, Kluwer Academic, Dordrecht, the Netherlands, 1991, pp. 95–123.
[7] G. N. Oliveira, E. M. Sá, and J. A. Dias da Silva, *On the eigenvalues of the matrix* $A + XBX^{-1}$, Linear and Multilinear Algebra, 5 (1977), pp. 119–128.
[8] E. M. Sá, *Imbedding conditions for* λ*-matrices*, Linear Algebra Appl., 24 (1979), pp. 33–50.
[9] F. C. Silva, *The eigenvalues of the product of matrices with prescribed similarity classes*, Linear and Multilinear Algebra, 34 (1993), pp. 269–277.
[10] R. C. Thompson, *Interlacing inequalities for invariant factors*, Linear Algebra Appl., 24 (1979), pp. 1–31.
[11] A. Sourour, *A factorization theorem for matrices*, Linear and Multilinear Algebra, 19 (1986), pp. 141–147.
[12] A. Sourour, *Nilpotent factorization of matrices*, Linear and Multilinear Algebra, 31 (1992), pp. 303–308.
[13] A. Sourour and K. Tang, *Factorization of singular matrices*, Proc. Amer. Math. Soc., 116 (1992), pp. 629–634.
[14] H. K. Wimmer, *Existenzsätze in der theorie der matrizen und lineare kontrolltheorie*, Monatsh. Math., 78 (1974), pp. 256–263.
[15] P. Y. Wu, *Products of nilpotent matrices*, Linear Algebra Appl., 96 (1987), pp. 227–232.
[16] I. Zaballa, *Matrices with prescribed rows and invariant factors*, Linear Algebra Appl., 87 (1987), pp. 113–146.
[17] I. Zaballa, *Interlacing inequalities and control theory*, Linear Algebra Appl., 101 (1988), pp. 9–31.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 673–691
© 2001 Society for Industrial and Applied Mathematics

A SHIFTED CYCLIC REDUCTION ALGORITHM FOR QUASI-BIRTH-DEATH PROBLEMS*

C. HE†, B. MEINI‡, AND N. H. RHEE§

Abstract. The problem of the computation of the stochastic matrix G associated with discrete-time quasi-birth-death (QBD) Markov chains is analyzed. We present a shifted cyclic reduction algorithm and show that the speed of convergence of the latter modified algorithm is always faster than that of the original cyclic reduction.

Key words. cyclic reduction, quasi-birth-death (QBD) Markov chains, quadratic matrix equation

AMS subject classifications. 15A51, 15A24, 60J10, 60K25, 65H05

PII. S0895479800371955

1. Introduction.

The block tridiagonal and block Toeplitz structure of the probability transition matrix P associated with quasi-birth-death (QBD) problems, i.e.,

$$P = \begin{bmatrix} B_0 & B_1 & & 0 \\ A_0 & A_1 & A_2 & \\ & A_0 & A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix},$$

allowed the design of fast and reliable methods for computation of the stochastic matrix G [16, 11, 10, 2, 3, 4, 12, 5]. Here, A_0, A_1, A_2, B_0, B_1 are $k \times k$ nonnegative matrices such that $A_0 + A_1 + A_2$ is irreducible, and P is irreducible, stochastic, and positive recurrent. Moreover, we assume without loss of generality [7] that G has only one eigenvalue of modulus one.

These fast methods rely on the property that the matrix G, which solves the nonlinear matrix equation

$$G = A_0 + A_1 G + A_2 G^2, \tag{1.1}$$

can be computed by solving the infinite block tridiagonal, block Toeplitz system

$$\begin{bmatrix} I - A_1 & -A_2 & & 0 \\ -A_0 & I - A_1 & -A_2 & \\ & -A_0 & I - A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^2 \\ G^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}. \tag{1.2}$$

*Received by the editors May 9, 2000; accepted for publication (in revised form) by D. Calvetti June 8, 2001; published electronically December 14, 2001.
http://www.siam.org/journals/simax/23-3/37195.html

†Sprint Corporation, Network Planning and Design, 7171 W. 95th Street Overland Park, KS 66212 (charlie.he@mail.sprint.com).

‡Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, 56127 Pisa, Italy (meini@dm.unipi.it).

§Department of Mathematics and Statistics, University of Missouri at Kansas City, Kansas City, MO 64110 (rhee@cctr.umkc.edu).

Recently in [2, 3, 4] Bini and Meini devised a new quadratically convergent and numerically stable algorithm for the computation of G based on a functional representation of cyclic reduction, which applies to general M/G/1 type Markov chains [16] and which extends the method of Latouche and Ramaswami [10].

The aim of this paper is to introduce a shifted cyclic reduction algorithm for QBDs and to show that the speed of convergence of a shifted cyclic reduction algorithm is always faster than that of the original cyclic reduction algorithm.

More precisely, since G has a known eigenvalue (equal to 1) and a known corresponding right eigenvector, we may apply a shifting technique, which consists of moving the eigenvalue 1 to 0 and maintaining the remaining ones. This trick leads to a new quadratic matrix equation having as its solution a singular matrix H such that $G = H + \boldsymbol{e}\boldsymbol{u}^T$, where $\boldsymbol{e}$ is the vector having all the entries equal to 1, and $\boldsymbol{u}$ is a known arbitrary vector with positive entries such that $\boldsymbol{u}^T\boldsymbol{e} = 1$. Thus the problem of the computation of G is reduced to the problem of the computation of H, which shares with G all the eigenvalues except for the eigenvalue 1, which is moved to 0.

In order to compute H we apply the cyclic reduction algorithm and show that the shifting leads to a faster convergence. In the standard cyclic reduction algorithm, the error of the approximation to G is $O((1/\sigma)^{2^j})$ as $j \to \infty$, where $\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}$ and $\phi(\lambda) = \det(-A_0 + (I - A_1)\lambda - A_2\lambda^2)$. With the shifting technique the error is $O((\theta/\sigma)^{2^j})$, where θ is any real number such that $\max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\} < \theta < 1$. Thus the convergence speed can be greatly increased when $\sigma \approx 1$ or $\theta \ll 1$.

Finally, we introduce a means to measure the conditioning of the quadratic matrix equations, and we prove that the shifted equation is better conditioned than the original one. This is important because even though the shifting technique leads to a better rate of convergence, it destroys the nonnegativity and the M-matrix properties of the blocks generated at each step of standard cyclic reduction, and in principle this fact could lead to a loss of accuracy of the results. We have performed several numerical experiments which show that the shifted cyclic reduction algorithm is fast and numerically accurate.

The paper is organized as follows. In section 2 we recall the cyclic reduction algorithm for QBDs. In section 3 we apply the shifting technique and show how the solutions of the shifted matrix equation are related to the solution of the original one. In section 4 we analyze the convergence properties of the cyclic reduction algorithm applied to the shifted matrix equation. In section 5 we study the conditioning of the two matrix equations. In section 6 we present some numerical results.

2. The cyclic reduction algorithm. In this section we recall the cyclic reduction algorithm for QBDs described in [3].

Let us consider the system (1.2). By recursively applying block cyclic reduction, i.e., an odd-even permutation of block rows and block columns, followed by one step of Gaussian elimination, the following sequence of infinite block tridiagonal systems is generated:

$$\begin{bmatrix} I - \widehat{A}_1^{(j)} & -A_2^{(j)} & & 0 \\ -A_0^{(j)} & I - A_1^{(j)} & -A_2^{(j)} & \\ & -A_0^{(j)} & I - A_1^{(j)} & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^{2^j+1} \\ G^{2\cdot 2^j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad j \geq 0, \tag{2.1}$$

where $A_0^{(0)} = A_0$, $\widehat{A}_1^{(0)} = A_1^{(0)} = A_1$, $A_2^{(0)} = A_2$, and the blocks $\widehat{A}_1^{(j)}$, $A_i^{(j)}$, $i = 0, 1, 2$, $j \geq 1$, are defined by the recurrences

$$\begin{aligned} A_0^{(j+1)} &= A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)}, \\ A_1^{(j+1)} &= A_1^{(j)} + A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)}, \\ A_2^{(j+1)} &= A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)}, \\ \widehat{A}_1^{(j+1)} &= \widehat{A}_1^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)}. \end{aligned} \tag{2.2}$$

The sequences of matrices generated by the above relations allow the fast computation of the matrix G. Indeed, from (2.1) it follows that

$$(I - \widehat{A}_1^{(j)})G - A_2^{(j)} G^{2^j+1} = A_0. \tag{2.3}$$

On the other hand, if we denote by R the minimal nonnegative solution of the matrix equation $R = A_2 + RA_1 + R^2 A_0$, then the following equation holds (see [3, 10]):

$$-A_2^{(j)} + R^{2^j}(I - A_1^{(j)}) - R^{2\cdot 2^j} A_0^{(j)} = 0.$$

Moreover, the maximum modulus eigenvalue of R is real, simple, and unique, and it is equal to $1/\sigma$ (see [7]), where $\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}$, and $\phi(\lambda) = \det(-A_0 + (I - A_1)\lambda - A_2\lambda^2)$. Thus, there exists an operator norm $||\cdot||$ such that $||R|| = \rho(R)$, where $\rho(R)$ denotes the spectral radius of R; hence, we obtain

$$||A_2^{(j)}|| \leq \rho(R)^{2^j}(||I - A_1^{(j)}|| + \rho(R)^{2^j}||A_0^{(j)}||).$$

Since the matrices $A_1^{(j)}$ and $A_0^{(j)}$ are bounded in norm [3], it follows that $||A_2^{(j)}|| = O(\rho(R)^{2^j})$ and from (2.3), since G^{2^j+1} is bounded, that

$$(I - \widehat{A}_1^{(j)})G - A_0 = O\left(\left(\frac{1}{\sigma}\right)^{2^j}\right). \tag{2.4}$$

Hence an approximation of the matrix G is given by $(I - \widehat{A}_1^{(j)})^{-1} A_0$ for j sufficiently large.

Due to the double exponential convergence to zero of the sequence $A_2^{(j)}$, just a small number of steps can be sufficient to reach a good approximation of G.

The rate of convergence is given by $1/\sigma$, and it is therefore related to the closeness to the unit circle of the smallest zero of $\phi(\lambda)$ of modulus larger than 1.

In the next section we present a trick to improve the rate of convergence; indeed, we show that by applying a deflating technique, which consists of removing the zero $\lambda = 1$ of $\phi(\lambda)$, the rate of convergence can be reduced to θ/σ, where θ is any real number such that $\max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\} < \theta < 1$.

3. A shifted matrix equation. Since G has a known eigenvalue 1 and a known eigenvector $\boldsymbol{e}$ where $\boldsymbol{e}$ is the k-dimensional vector having all the entries equal to 1, we can modify G so that the eigenvalue 1 is shifted to 0. This shifting technique improves the convergence speed of the cyclic reduction algorithm.

Set $H = G - \boldsymbol{e}\boldsymbol{u}^T$, where $\boldsymbol{u}$ is any vector whose elements are positive and such that $\boldsymbol{u}^T\boldsymbol{e} = 1$. Then the eigenvalues of H are those of G except that in H the eigenvalue

1 of G is replaced by 0. Moreover, $\boldsymbol{e}$ is an eigenvector of H corresponding to the eigenvalue 0, and hence $H\boldsymbol{e} = 0$. So we have

$$G = H + \boldsymbol{e}\boldsymbol{u}^T \quad \text{and} \quad G^2 = (H + \boldsymbol{e}\boldsymbol{u}^T)^2 = H^2 + \boldsymbol{e}\boldsymbol{u}^T H + \boldsymbol{e}\boldsymbol{u}^T.$$

By replacing G by H in (1.1) we obtain that H solves the following shifted equation:

$$B_0 + B_1 H + B_2 H^2 = H, \tag{3.1}$$

where

$$\begin{aligned} B_0 &= A_0 + (A_1 + A_2 - I)\boldsymbol{e}\boldsymbol{u}^T = A_0(I - \boldsymbol{e}\boldsymbol{u}^T), \\ B_1 &= A_1 + A_2\boldsymbol{e}\boldsymbol{u}^T, \\ B_2 &= A_2. \end{aligned} \tag{3.2}$$

So the computation of the matrix G can be reduced to the computation of the matrix $H = G - \boldsymbol{e}\boldsymbol{u}^T$, which solves the nonlinear matrix equation (3.1).

3.1. Spectral properties of the shifted matrix polynomial. The rank 1 corrections of the matrices A_0 and A_1 have the effect of keeping unchanged all the zeros of the polynomial $\phi(\lambda) = \det A(\lambda)$, $A(\lambda) = -A_0 + (I - A_1)\lambda - A_2\lambda^2$, except for the zero $\lambda = 1$, which is moved to $\lambda = 0$. Indeed, the following result holds.

THEOREM 3.1. *The zeros of the polynomial $\psi(\lambda) = \det B(\lambda)$, $B(\lambda) = -B_0 + (I - B_1)\lambda - B_2\lambda^2$, are*

$$\{\lambda \ : \ \det A(\lambda) = 0, \ \lambda \neq 1\} \cup \{0\}.$$

To prove the above theorem we need to introduce some notations and some results on pairs of matrices (see [8]).

DEFINITION 3.2. *Let A and B be $n \times n$ complex matrices. We define $\Lambda(A, B)$ by*

$$\Lambda(A, B) = \{\lambda \in \mathbf{C} : \det(A - \lambda B) = 0\}.$$

If $\lambda \in \Lambda(A, B)$, λ is called an eigenvalue of the pair (A, B). If λ is an eigenvalue of the pair (A, B), then there exists a nonzero vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \lambda B\boldsymbol{x}$, and such an $\boldsymbol{x}$ is called an eigenvector *of the pair (A, B) corresponding to the eigenvalue λ.*

THEOREM 3.3. *If A and B are $n \times n$ complex matrices, then there exist unitary matrices U and V such that $U^H AV = R$ and $U^H BV = \tilde{R}$ where R and $\tilde{R}$ are upper triangular. If for some i, r_{ii} and $\tilde{r}_{ii}$ are both zero, then $\Lambda(A, B) = \mathbf{C}$. Otherwise*

$$\Lambda(A, B) = \{r_{ii}/\tilde{r}_{ii} : \tilde{r}_{ii} \neq 0, \ i = 1, \ldots, n\}.$$

Note that $\boldsymbol{v}_1$, the first column of V, is an eigenvector of the pair (A, B) corresponding to the eigenvalue $r_{11}/\tilde{r}_{11}$.

DEFINITION 3.4. *Two pairs of matrices (A, B) and (C, D) are said to be* equivalent *if there exist nonsingular matrices L and M such that $C = LAM$ and $D = LBM$. It is easy to verify that if (A, B) and (C, D) are equivalent pairs, then $\Lambda(A, B) = \Lambda(C, D)$.*

Proof of Theorem 3.1. Let

$$\Lambda(A) = \{\lambda \in \mathbf{C} : \det A(\lambda) = 0\}, \qquad \Lambda(B) = \{\lambda \in \mathbf{C} : \det B(\lambda) = 0\}.$$

If we set

$$S = \begin{bmatrix} I - A_1 & -A_0 \\ I & 0 \end{bmatrix} \qquad \text{and} \qquad T = \begin{bmatrix} A_2 & 0 \\ 0 & I \end{bmatrix},$$

it is easy to verify that $\Lambda(A) = \Lambda(S,T)$. Similarly, if we set

$$\tilde{S} = \begin{bmatrix} I - B_1 & -B_0 \\ I & 0 \end{bmatrix},$$

then $\Lambda(B) = \Lambda(\tilde{S},T)$.

Now if we let

$$L = \begin{bmatrix} I & -A_2\boldsymbol{e}\boldsymbol{u}^T \\ 0 & I \end{bmatrix} \qquad \text{and} \qquad M = \begin{bmatrix} I & \boldsymbol{e}\boldsymbol{u}^T \\ 0 & I \end{bmatrix},$$

then by direct computation one can verify that

$$\tilde{S} = L\left(S - T\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}[\boldsymbol{0}, \boldsymbol{u}^T]\right)M \qquad \text{and} \qquad T = LTM.$$

So

$$\Lambda(B) = \Lambda(\tilde{S},T) = \Lambda\left(S - T\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}[\boldsymbol{0}, \boldsymbol{u}^T], T\right).$$

Note that the vector $[\boldsymbol{e}^T, \boldsymbol{e}^T]^T$ is an eigenvector of the pair (S,T) corresponding to the eigenvalue 1. So there exist unitary matrices U and V such that

$$U^H S V = R \qquad \text{and} \qquad U^H T V = \tilde{R}$$

with

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{2k}}\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}.$$

Since $\boldsymbol{v}_1$ is an eigenvector of the pair (S,T) corresponding to the eigenvalue 1, we have $r_{11}/\tilde{r}_{11} = 1$. By Theorem 3.3 we know that

$$\begin{aligned} \Lambda(A) &= \Lambda(S,T) \\ &= \{\tfrac{r_{11}}{\tilde{r}_{11}}, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\} \\ &= \{1, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\}. \end{aligned} \tag{3.3}$$

Here we assumed that $\tilde{r}_{i,i} = 0$ for $i = l+1, \ldots, 2k$.

Since $U^H T[\boldsymbol{e}^T, \boldsymbol{e}^T]^T = \tilde{R}V^H(\sqrt{2k}\boldsymbol{v}_1) = \sqrt{2k}\tilde{r}_{11}\boldsymbol{e}_1$, where $\boldsymbol{e}_1$ is the first standard unit vector of length $2k$, and the first component of the row vector $[\boldsymbol{0}, \boldsymbol{u}^T]V$ is $1/\sqrt{2k}$, it follows that the matrix

$$U^H\left\{S - T\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}[\boldsymbol{0}, \boldsymbol{u}^T]\right\}V = R - U^H T\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}[\boldsymbol{0}, \boldsymbol{u}^T]V$$

is upper triangular and has $r_{11} - \tilde{r}_{11}, r_{22}, \ldots, r_{2k,2k}$ as its diagonal elements. Hence by Theorem 3.3 we have

$$\begin{aligned} \Lambda(B) &= \Lambda(\tilde{S},T) \\ &= \Lambda\left(S - T\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}[\boldsymbol{0}, \boldsymbol{u}^T], T\right) \\ &= \{\tfrac{r_{11}-\tilde{r}_{11}}{\tilde{r}_{11}}, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\} \\ &= \{0, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\}. \end{aligned} \tag{3.4}$$

From (3.3) and (3.4) we see that $\Lambda(B)$ is the same as $\Lambda(A)$ except that in $\Lambda(B)$ $1 \in \Lambda(A)$ is replaced by 0. ∎

3.2. Solutions of the shifted matrix equation. The blocks B_0, B_1, B_2 are such that $H = G - \boldsymbol{e}\boldsymbol{u}^T$ solves the nonlinear matrix equation (3.1). The following theorem shows how the minimal nonnegative solutions of the nonlinear matrix equations (3.5) are transformed by the shifting.

THEOREM 3.5. *Let $\boldsymbol{u}$ be positive, and let G, R, S, and F be the minimal nonnegative solutions of the matrix equations*

$$\begin{aligned} G &= A_0 + A_1 G + A_2 G^2, \\ S &= A_0 S^2 + A_1 S + A_2, \\ R &= R^2 A_0 + R A_1 + A_2, \\ F &= A_0 + F A_1 + F^2 A_2, \end{aligned} \tag{3.5}$$

respectively. Then $H = G - \boldsymbol{e}\boldsymbol{u}^T$, $K = (I - \boldsymbol{e}\boldsymbol{u}^T S)S(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1}$, $T = R$, and $V = F - \boldsymbol{w}\boldsymbol{u}^T(I - A_1 - A_0 S)^{-1}$, where $\boldsymbol{w} = (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1} A_0(I - S)\boldsymbol{e}$, solve the matrix equations

$$\begin{aligned} H &= B_0 + B_1 H + B_2 H^2, \\ K &= B_0 K^2 + B_1 K + B_2, \\ T &= T^2 B_0 + T B_1 + B_2, \\ V &= B_0 + V B_1 + V^2 B_2. \end{aligned} \tag{3.6}$$

Proof. First observe that if $\boldsymbol{u}$ is positive, then $\boldsymbol{u}^T S\boldsymbol{e} < 1$, since $S\boldsymbol{e} \le \boldsymbol{e}$ and S is substochastic [9, 14]; hence $I - \boldsymbol{e}\boldsymbol{u}^T S$ is nonsingular.

The matrix H solves the first equation of (3.6) by construction. For the matrix K we have

$$\begin{aligned} &B_0 K^2 + B_1 K + B_2 \\ &\quad = B_0(I - \boldsymbol{e}\boldsymbol{u}^T S)S^2(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} + B_1(I - \boldsymbol{e}\boldsymbol{u}^T S)S(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} + B_2 \\ &\quad = \left(A_0(I - \boldsymbol{e}\boldsymbol{u}^T)S^2 + \left(A_1 + (A_0\boldsymbol{e} - \boldsymbol{e})\boldsymbol{u}^T S + A_2\boldsymbol{e}\boldsymbol{u}^T\right) S \right. \\ &\qquad \left. + A_2(I - \boldsymbol{e}\boldsymbol{u}^T S)\right)(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} \\ &\quad = (S - \boldsymbol{e}\boldsymbol{u}^T S^2)(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} = K. \end{aligned}$$

For the matrix $T = R$, since [9] $R = A_2(I - A_1 - A_2 G)^{-1}$ and

$$\begin{aligned} A_2\boldsymbol{e} &= R(I - A_1 - R A_0)\boldsymbol{e} = R\left(I - A_1 - A_2(I - A_1 - A_2 G)^{-1} A_0\right)\boldsymbol{e} \\ &= R(I - A_1 - A_2 G)\boldsymbol{e} = R A_0 \boldsymbol{e}, \end{aligned}$$

we obtain

$$\begin{aligned} B_2 + R B_1 + R^2 B_0 &= A_2 + R(A_1 + A_2\boldsymbol{e}\boldsymbol{u}^T) + R^2 A_0(I - \boldsymbol{e}\boldsymbol{u}^T) \\ &= A_2 + R A_1 + R^2 A_0 + R A_2\boldsymbol{e}\boldsymbol{u}^T - R^2 A_0\boldsymbol{e}\boldsymbol{u}^T \\ &= R + R(A_2\boldsymbol{e} - R A_0\boldsymbol{e})\boldsymbol{u}^T = R. \end{aligned}$$

Concerning the matrix V, observe that $I - B_1 - B_0 K$ is nonsingular, since $I - A_1 - A_0 S - B_2 G$ is nonsingular [9, 14], and $I - A_1 - A_0 S - B_2 G = (I - B_1 - B_0 K - B_2 H)(I - \boldsymbol{e}\boldsymbol{u}^T S) = (I - B_1 - B_0 K)(I - KH)(I - \boldsymbol{e}\boldsymbol{u}^T S)$. It can be shown by direct substitution that $V = B_0(I - B_1 - B_0 K)^{-1}$ solves the last matrix equation in (3.6). By rewriting the matrix V in terms of the blocks A_0, A_1, A_2 we obtain that

$$V = A_0(I - \boldsymbol{e}\boldsymbol{u}^T)(I - A_1 - A_0 S - A_2\boldsymbol{e}\boldsymbol{u}^T)^{-1}.$$

On the other hand, by applying the Sherman–Woodbury–Morrison formula [8], we have

$$\begin{aligned}
&(I - A_1 - A_0 S - A_2 \boldsymbol{e}\boldsymbol{u}^T)^{-1} = (I - A_1 - A_0 S)^{-1} \\
&+ (1 - \boldsymbol{u}^T (I - A_1 - A_0 S)^{-1} A_2 \boldsymbol{e})^{-1} (I - A_1 - A_0 S)^{-1} A_2 \boldsymbol{e}\boldsymbol{u}^T (I - A_1 - A_0 S)^{-1} \\
&= (I - A_1 - A_0 S)^{-1} + (1 - \boldsymbol{u}^T S \boldsymbol{e})^{-1} S \boldsymbol{e}\boldsymbol{u}^T (I - A_1 - A_0 S)^{-1} \\
&= \left(I - (1 - \boldsymbol{u}^T S \boldsymbol{e})^{-1} S \boldsymbol{e}\boldsymbol{u}^T\right) (I - A_1 - A_0 S)^{-1}.
\end{aligned}$$

From the latter equation and from the relation $F = A_0 (I - A_1 - A_0 S)^{-1}$ [9], we obtain

$$\begin{aligned}
V &= F + (1 - \boldsymbol{u}^T S \boldsymbol{e})^{-1} A_0 S \boldsymbol{e}\boldsymbol{u}^T (I - A_1 - A_0 S)^{-1} \\
&\quad - \left(A_0 \boldsymbol{e} + (\boldsymbol{u}^T S \boldsymbol{e})(1 - \boldsymbol{u}^T S \boldsymbol{e})^{-1} A_0 \boldsymbol{e}\right) \boldsymbol{u}^T (I - A_1 - A_0 S)^{-1} \\
&= F - \boldsymbol{w}\boldsymbol{u}^T (I - A_1 - A_0 S)^{-1},
\end{aligned}$$

where $\boldsymbol{w} = (1 - \boldsymbol{u}^T S \boldsymbol{e})^{-1} A_0 (I - S) \boldsymbol{e}$. ∎

The spectral properties of the solutions of the shifted matrix equations allow us to prove that the rate of convergence is improved with respect to standard cyclic reduction. Let us define

$$\begin{aligned}
&\bar{\theta} = \max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\}, \\
&\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}.
\end{aligned} \tag{3.7}$$

The convergence properties which will be proved in the next section rely on the fact that $\bar{\theta} < 1$, $\sigma = 1/\rho(R) > 1$, S and R have the same eigenvalues (see [14]), and thus also K and R, and $\rho(K) = 1/\sigma < 1$.

4. A shifted cyclic reduction algorithm. In this section we apply the cyclic reduction algorithm to solve the shifted matrix equation (3.1), and we show that the rate of convergence is improved with respect to the same algorithm applied to the original matrix equation (1.1).

By following the approach described in section 2, we generate by means of cyclic reduction the sequence of infinite block tridiagonal systems:

$$\begin{bmatrix}
I - \widehat{B}_1^{(j)} & -B_2^{(j)} & & 0 \\
-B_0^{(j)} & I - B_1^{(j)} & -B_2^{(j)} & \\
& -B_0^{(j)} & I - B_1^{(j)} & \ddots \\
0 & & \ddots & \ddots
\end{bmatrix}
\begin{bmatrix} H \\ H^{2^j+1} \\ H^{2 \cdot 2^j+1} \\ \vdots \end{bmatrix}
= \begin{bmatrix} B_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad j \geq 0, \tag{4.1}$$

where $B_0^{(0)} = B_0$, $\widehat{B}_1^{(0)} = B_1^{(0)} = B_1$, $B_2^{(0)} = B_2$, and the blocks $\widehat{B}_1^{(j)}$, $B_i^{(j)}$, $i = 0, 1, 2$, $j \geq 1$, are defined by the recurrences

$$\begin{aligned}
B_0^{(j+1)} &= B_0^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)}, \\
B_1^{(j+1)} &= B_1^{(j)} + B_0^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_2^{(j)} + B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)}, \\
B_2^{(j+1)} &= B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_2^{(j)}, \\
\widehat{B}_1^{(j+1)} &= \widehat{B}_1^{(j)} + B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)}.
\end{aligned} \tag{4.2}$$

In the shifted case the problem of the nonsingularity of the blocks $I - B_1^{(j)}$ must be considered. In fact, in the original case the matrices $I - A_1^{(j)}$ are nonsingular M-matrices for any j, since $I - A_1^{(j)}$ can be viewed as a Schur complement of the block

$(2^{j+1}-1)\times(2^{j+1}-1)$ matrix obtained by truncating the infinite matrix (1.2) at the block size $2^{j+1}-1$ (see [1, 6]), and this finite matrix is a nonsingular M-matrix. Analogously, the matrix $I-B_1^{(j)}$ can be viewed as a Schur complement of the matrix $Q_{2^{j+1}-2}$, where Q_n is the $(n+1)\times(n+1)$ block matrix

$$Q_n=\begin{bmatrix} I-B_1 & -B_2 & & 0\\ -B_0 & \ddots & \ddots & \\ & \ddots & \ddots & -B_2\\ 0 & & -B_0 & I-B_1 \end{bmatrix}. \tag{4.3}$$

Thus, the $(j+1)$st step of cyclic reduction can be performed, i.e., $I-B_1^{(j)}$ is nonsingular, if and only if $Q_{2^{j+1}-2}$ is nonsingular [1]. Based on this property, we prove the following result.

THEOREM 4.1. *Let $\boldsymbol{u}$ be any positive vector such that $\boldsymbol{u}^T\boldsymbol{e}=1$ and let $H=G-\boldsymbol{u}\boldsymbol{e}^T$, $K=(I-\boldsymbol{e}\boldsymbol{u}^TS)S(I-\boldsymbol{e}\boldsymbol{u}^TS)^{-1}$, where G and S are the minimal nonnegative solutions of the matrix equations $G=A_0+A_1G+A_2G^2$, $S=A_0S^2+A_1S+A_2$. Let $B_0^{(j)}$, $B_1^{(j)}$, $B_2^{(j)}$, $\widehat{B}_1^{(j)}$ be the blocks generated at the jth step of cyclic reduction, $j\geq 1$. Then the following conditions are equivalent:*

1. *$I-(HK^{2^{j+1}-1})(KH^{2^{j+1}-1})$ is nonsingular;*
2. *$I-(KH^{2^{j+1}-1})(HK^{2^{j+1}-1})$ is nonsingular;*
3. *$I-B_1^{(j)}$ is nonsingular.*

Proof. The matrix $I-B_1^{(j)}$ is nonsingular if and only if $Q_{2^{j+1}-2}$ is nonsingular, where Q_n is defined in (4.3). For simplicity of notation, set $n=2^{j+1}-2$. Observe that the matrix $W=I-B_1-B_0K-B_2H$ is nonsingular, since $W=(I-A_1-A_0S-A_2G)(I-\boldsymbol{e}\boldsymbol{u}^TS)$ and $I-A_1-A_0S-B_2G$ is nonsingular (see [9, 14]). Let $\boldsymbol{x}=(\boldsymbol{x}_i)_{i=0,\dots,n}$ such that $Q_n\boldsymbol{x}=0$. Therefore, since $I-B_1-B_0K-B_2H$ is nonsingular, then

$$\boldsymbol{x}_i=H^i\boldsymbol{r}+K^{n-i}\boldsymbol{s},\quad i=0,\dots,n,$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are suitable vectors such that the boundary conditions

$$\begin{aligned}&(I-B_1)\boldsymbol{x}_0-B_2\boldsymbol{x}_1=\boldsymbol{0},\\ &-B_0\boldsymbol{x}_{n-1}+(I-B_1)\boldsymbol{x}_n=\boldsymbol{0}\end{aligned} \tag{4.4}$$

are satisfied (see [15]). By imposing the above equalities, we obtain that $\boldsymbol{r}$ and $\boldsymbol{s}$ must solve the following homogeneous linear system:

$$\begin{bmatrix} I-B_1-B_2H & ((I-B_1)K-B_2)K^{n-1}\\ ((I-B_1)H-B_0)H^{n-1} & I-B_1-B_0K \end{bmatrix}\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{s}\end{bmatrix}=0,$$

which can be written as

$$\begin{bmatrix} I-B_1-B_2H & B_0K^{n+1}\\ B_2H^{n+1} & I-B_1-B_0K \end{bmatrix}\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{s}\end{bmatrix}=0, \tag{4.5}$$

since H and K solve the matrix equations (1.1), (3.1). Hence, $\boldsymbol{x}=\boldsymbol{0}$ if and only if $\boldsymbol{r}=\boldsymbol{s}=\boldsymbol{0}$. Now, the block diagonal entries in the matrix of (4.5) are nonsingular since $I-B_1-B_0K-B_2H=(I-B_1-B_2H)(I-HK)=(I-B_1-B_0K)(I-KH)$,

and $I - B_1 - B_0K - B_2H$ is nonsingular. By computing the Schur complement of $I - B_1 - B_2H$ in $I - B_1 - B_0K$ we obtain the matrix

$$\begin{aligned} S_1 &= I - B_1 - B_0K - B_2H^{n+1}(I - B_1 - B_2H)^{-1}B_0K^{n+1} \\ &= I - B_1 - B_0K - B_2H^{n+2}K^{n+1}. \end{aligned}$$

Thus, S_1 (and hence Q_n) is nonsingular if and only if $(I - B_1 - B_0K)^{-1}S_1 = I - KH^{n+2}K^{n+1}$ is nonsingular. By taking the Schur complement of $I - B_1 - B_0K$ in $I - B_1 - B_2H$ we complete the proof of the theorem. ∎

The above theorem gives necessary and sufficient conditions for the applicability of cyclic reduction. Since $\rho(H) < 1$ and $\rho(K) < 1$, the matrix $Z_j = I - (HK^{2^{j+1}-1})(KH^{2^{j+1}-1})$ is nonsingular for sufficiently large values of j. From the numerical experiments that we have performed, and from the fact that the matrix $I - (GS^{2^{j+1}-1})(SG^{2^{j+1}-1})$ is nonsingular for any value of j, we conjecture that also Z_j is nonsingular for any value of j. However, if it were not, the cyclic reduction algorithm can still be applied by performing a different permutation of block rows and columns, which allows one to skip the steps that cannot be performed for the singularity of the blocks $I - B_1^{(j)}$. (We refer to the paper [1] for details of this subject.)

Henceforth we assume that the matrices $I - B_1^{(j)}$ are nonsingular for any j. The convergence properties stated by the next theorem enable us to efficiently compute the matrix H.

THEOREM 4.2. *Let $B_0^{(j)}$, $B_1^{(j)}$, $B_2^{(j)}$ be the blocks generated at the jth step of cyclic reduction. Then $B_1^{(j)}$, $(I - B_1^{(j)})^{-1}$ are bounded, and for any operator norm it holds that $||B_2^{(j)}|| = O((1/\sigma)^{2^j})$ and $||B_0^{(j)}|| = O(\theta^{2^j})$ for any $\bar{\theta} < \theta < 1$, where $\bar{\theta}$ and σ are defined in* (3.7).

Proof. From (4.1), at each step j the following equations hold:

$$H^{2^j} = B_0^{(j)} + B_1^{(j)}H^{2^j} + B_2^{(j)}H^{2\cdot 2^j}, \tag{4.6}$$

$$K^{2^j} = B_0^{(j)}K^{2\cdot 2^j} + B_1^{(j)}K^{2^j} + B_2^{(j)}, \tag{4.7}$$

and therefore

$$(I - B_1^{(j)})^{-1}B_0^{(j)} + H^{2^j} + (I - B_1^{(j)})^{-1}B_2^{(j)}H^{2\cdot 2^j} = 0, \tag{4.8}$$

$$(I - B_1^{(j)})^{-1}B_2^{(j)} + K^{2^j} + (I - B_1^{(j)})^{-1}B_0^{(j)}K^{2\cdot 2^j} = 0. \tag{4.9}$$

We first prove that the matrices $C_0^{(j)} = (I - B_1^{(j)})^{-1}B_0^{(j)}$ and $C_2^{(j)} = (I - B_1^{(j)})^{-1}B_2^{(j)}$ are bounded in norm.

Since the maximum modulus eigenvalue of R, and thus of K, is real, simple, and unique [7], then there exists an operator norm $||\cdot||_K$ such that $||K||_K = \rho(K)$. Moreover, let $\epsilon > 0$ such that $\rho(H) + \epsilon < 1$, and let $||\cdot||_{H,\epsilon}$ be an operator norm such that $||H||_{H,\epsilon} \le \rho(H) + \epsilon$ (see [8]).

Let $\alpha_j = ||C_0^{(j)}||_{H,\epsilon}$. If α_j is not bounded, then there exists a subsequence α_{j_h} such that α_{j_h} diverges to infinity. From (4.8) we have

$$\begin{aligned} \alpha_{j_h} &\le ||C_2^{(j_h)}||_{H,\epsilon}||H^{2\cdot 2^{j_h}}||_{H,\epsilon} + ||H^{2^{j_h}}||_{H,\epsilon} \\ &\le (\rho(H) + \epsilon)^{2^{j_h}} \left((\rho(H) + \epsilon)^{2^{j_h}} ||C_2^{(j_h)}||_{H,\epsilon} + 1 \right). \end{aligned}$$

Thus,

$$||C_2^{(j_h)}||_{H,\epsilon} \ge \left(\frac{\alpha_{j_h}}{(\rho(H) + \epsilon)^{2^{j_h}}} - 1 \right) \frac{1}{(\rho(H) + \epsilon)^{2^{j_h}}},$$

and, since α_{j_h} diverges to infinity, for the equivalence of matrix norms, there exists a constant $c' > 0$ such that

$$||C_2^{(j_h)}||_K \geq c' \frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}}. \tag{4.10}$$

Moreover, there exist a constant $c'' > 0$ such that

$$||C_0^{(j)}||_K \leq c''||C_0^{(j)}||_{H,\epsilon} = c''\alpha_j. \tag{4.11}$$

Thus, from (4.9) and (4.11), we have

$$||C_2^{(j_h)}||_K \leq c''\alpha_{j_h}\rho(K)^{2\cdot 2^{j_h}} + \rho(K)^{2^{j_h}}.$$

Hence, from (4.10), we obtain

$$c' \frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}} \leq c''\alpha_{j_h}\rho(K)^{2\cdot 2^{j_h}} + \rho(K)^{2^{j_h}},$$

which contradicts the assumption that α_{j_h} goes to infinity. By using a similar argument we can prove that $C_2^{(j)}$ is also bounded in norm. Thus, from (4.8), for any operator norm $||\cdot||$ there exists a constant γ such that

$$||C_0^{(j)}|| \leq \gamma(\rho(H)+\epsilon)^{2^j}. \tag{4.12}$$

Similarly, from (4.9) for any operator norm $||\cdot||$ there exists a constant γ' such that

$$||C_2^{(j)}|| \leq \gamma'\rho(K)^{2^j}. \tag{4.13}$$

From (4.12) and (4.13), since

$$I - B_1^{(j+1)} = (I - B_1^{(j)})(I - C_0^{(j)}C_2^{(j)} + C_2^{(j)}C_0^{(j)}),$$

we have

$$||I - B_1^{(j+1)}|| \leq ||I - B_1^{(j)}||(1+\sigma_j),$$

where $\sigma_j = O((\rho(H)+\epsilon)^{2^j}\rho(K)^{2^j})$. Thus the matrices $I - B_1^{(j)}$, and hence the matrices $B_1^{(j)}$, are bounded in norm. Similarly, it holds that

$$||(I - B_1^{(j+1)})^{-1}|| \leq \frac{1}{1-\sigma_j}||(I - B_1^{(j)})^{-1}||,$$

and thus $||(I - B_1^{(j)})^{-1}||$ is bounded.

Now, from the boundedness of $||I - B_1^{(j)}||$, and from (4.6) and (4.7), by applying the same argument used to derive (4.12) and (4.13), we can show that $||B_2^{(j)}||$ and $||B_0^{(j)}||$ are bounded, and thus that $||B_0^{(j)}|| = O((\rho(H)+\epsilon)^{2^j})$, $||B_2^{(j)}|| = O(\rho(K)^{2^j})$. ∎

The matrix G can be directly recovered, without computing the matrix H, according to the following result.

THEOREM 4.3. *For any operator norm $||\cdot||$ and for any $\bar{\theta} < \theta < 1$, it holds that*

$$A_0 - (I - \widehat{B}_1^{(j)})G = O\left(\left(\frac{\theta}{\sigma}\right)^{2^j}\right). \tag{4.14}$$

Proof. From the relation

$$(I - \widehat{B}_1^{(j)})H - B_2^{(j)}H^{2^j+1} - B_0 = 0$$

and from Theorem 4.2, it follows that for any operator norm $||\cdot||$ and for any $\bar{\theta} < \theta < 1$, it holds that

$$B_0 - (I - \widehat{B}_1^{(j)})H = O\left(\left(\frac{\theta}{\sigma}\right)^{2^j}\right).$$

On the other hand, it can be easily proved by induction that $(I - \widehat{B}_1^{(j)})\boldsymbol{e} = A_0\boldsymbol{e}$ for any $j \geq 0$. Thus, by replacing H with $G - \boldsymbol{e}\boldsymbol{u}^T$ and B_0 with $A_0 - A_0\boldsymbol{e}\boldsymbol{u}^T$, we arrive at (4.14). ∎

From the above result, compared with (2.4), it follows that the shifted cyclic reduction can be much faster than the original one. Indeed, if the second largest modulus eigenvalue $\bar{\theta}$ of G is far from the unit circle, then the rate of convergence is much improved.

Thus, the deflating technique leads to a better rate of convergence but destroys the nonnegativity and M-matrix properties of the blocks generated at each step. Indeed, generally neither $I - B_1^{(j)}$ is an M-matrix, nor are $B_0^{(j)}$, $B_2^{(j)}$ nonnegative matrices. (From the numerical experiments, it seems that $I - \widehat{B}_1^{(j)}$ is an M-matrix for $j = 0, 1, \ldots$.) In principle this fact could lead to a loss of accuracy of the results obtained with the shifting technique. In practice, we have not observed any differences, in terms of accuracy, between the results obtained with the two algorithms. Furthermore, as we will prove in the next section, the shifted equation is better conditioned than the original one.

5. Conditioning of the shifted matrix equation. In this section we introduce a measure of the conditioning of the matrix equation, and we show that the shifted equation is better conditioned than the original one.

Consider the matrix equation (1.1) and a perturbed equation

$$G + \triangle G = (A_0 + \triangle A_0) + (A_1 + \triangle A_1)(G + \triangle G) + (A_2 + \triangle A_2)(G + \triangle G)^2. \tag{5.1}$$

Using (1.1) this perturbed equation simplifies up to the first order in $\triangle G$,

$$(I - A_1 - A_2G)\triangle G - A_2(\triangle G)G = \triangle A, \tag{5.2}$$

where

$$\triangle A = \triangle A_0 + (\triangle A_1)G + (\triangle A_2)G^2.$$

Note that (5.2) can be written

$$W\mathbf{vec}(\triangle G) = \mathbf{vec}(\triangle A), \tag{5.3}$$

where

$$W = I \otimes (I - A_1 - A_2G) + G^T \otimes (-A_2)$$

and $\mathbf{vec}(A)$ is the k^2-dimensional vector obtained by arranging columnwise the entries of the matrix A.

Let $\Lambda(M)$ denote the set of all eigenvalues of any square matrix M.

THEOREM 5.1. *The $k^2 \times k^2$ matrix W is nonsingular. Furthermore*

$$\min\{|\lambda| : \lambda \in \Lambda(W)\} = 1 - \rho(A_1 + A_2G + A_2).$$

Proof. Let S be the Schur canonical form of G^T. Then the matrix W is similar to

$$\tilde{W} = I \otimes (I - A_1 - A_2G) + S \otimes (-A_2).$$

Due to the upper triangular structure of S and the fact that $\Lambda(S) = \Lambda(G)$, it is easy to see that

$$\Lambda(W) = \Lambda(\tilde{W}) = \bigcup_{\lambda \in \Lambda(G)} \Lambda(I - A_1 - A_2G - \lambda A_2).$$

Note that whenever $\lambda \in \Lambda(G)$,

$$\begin{aligned}
\rho(A_1 + A_2G + \lambda A_2) &\le \rho(|A_1 + A_2G + \lambda A_2|) \\
&\le \rho(A_1 + A_2G + |\lambda| A_2) \\
&\le \rho(A_1 + A_2G + A_2) \\
&< 1.
\end{aligned}$$

In the last step we used the fact that since $A_0 + A_1 + A_2$ is irreducible and positive recurrent, the matrix $A_1 + A_2G + A_2$ has spectral radius less than 1. It follows that 0 does not belong to $\Lambda(W)$, and hence W is nonsingular. Clearly

$$\min\{|\lambda| : \lambda \in \Lambda(W)\} = 1 - \rho(A_1 + A_2G + A_2). \qquad \square$$

Since W is nonsingular,

$$\mathbf{vec}(\triangle G) = W^{-1}\mathbf{vec}(\triangle A).$$

Hence

$$\begin{aligned}
\|\triangle G\|_F &= \|\mathbf{vec}(\triangle G)\|_2 \\
&\le \|W^{-1}\|_2 \|\mathbf{vec}(\triangle A)\|_2 \\
&= \frac{1}{\sigma_{min}(W)} \|\triangle A_0 + (\triangle A_1)G + (\triangle A_2)G^2\|_F \\
&\le \frac{\sqrt{k}}{\sigma_{min}(W)} (\|\triangle A_0\|_F + \|\triangle A_1\|_F + \|\triangle A_2\|_F).
\end{aligned}$$

Here $\sigma_{min}(W)$ is the minimum singular value of the matrix W. Therefore we may view $1/\sigma_{min}(W)$ as a condition number of (1.1). Even though $1/\min\{|\lambda| : \lambda \in \Lambda(W)\} \le 1/\sigma_{min}(W)$, we may consider

$$\frac{1}{\min\{|\lambda| : \lambda \in \Lambda(W)\}} = \frac{1}{1 - \rho(A_1 + A_2G + A_2)}$$

as a number which reflects the conditioning of (1.1).

Consider now the shifted matrix equation (3.1) and the perturbed equation

$$H + \triangle H = (B_0 + \triangle B_0) + (B_1 + \triangle B_1)(H + \triangle H) + (B_2 + \triangle B_2)(H + \triangle H)^2.$$

As before, this perturbed equation simplifies up to the first order in $\triangle H$,

$$(I - B_1 - B_2 H)\triangle H - B_2(\triangle H)H = \triangle B, \tag{5.4}$$

where

$$\triangle B = \triangle B_0 + (\triangle B_1)H + (\triangle B_2)H^2.$$

Note that (5.4) can be written

$$Q\mathbf{vec}(\triangle H) = \mathbf{vec}(\triangle B),$$

where

$$Q = I \otimes (I - B_1 - B_2 H) + H^T \otimes (-B_2).$$

Note that

$$B_1 + B_2 H = A_1 + A_2 \boldsymbol{e}\boldsymbol{u}^T + A_2(G - \boldsymbol{e}\boldsymbol{u}^T) = A_1 + A_2 G.$$

Hence

$$Q = I \otimes (I - A_1 - A_2 G) + H^T \otimes (-A_2).$$

Note that if we denote $\Lambda(G)$ by

$$\Lambda(G) = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$$

with

$$1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_k|,$$

then

$$\Lambda(H) = \{\lambda_2, \lambda_3, \ldots, \lambda_k, 0\}$$

and $|\lambda_2| = \bar{\theta}$.

THEOREM 5.2. *The $k^2 \times k^2$ matrix Q is nonsingular. Furthermore*

$$\min\{|\lambda| : \lambda \in \Lambda(Q)\} \geq 1 - \rho(A_1 + A_2 G + \bar{\theta} A_2),$$

and the equality holds if λ_2 is a positive real number.

Proof. As in the proof of Theorem 5.1 it is easy to see that

$$\Lambda(Q) = \Lambda(\tilde{Q}) = \bigcup_{\lambda \in \Lambda(H)} \Lambda(I - A_1 - A_2 G - \lambda A_2).$$

Note that since $\rho(H) = \bar{\theta} = |\lambda_2|$ whenever $\lambda \in \Lambda(H)$,

$$\begin{aligned}\rho(A_1 + A_2G + \lambda A_2) &\le \rho(|A_1 + A_2G + \lambda A_2|)\\ &\le \rho(A_1 + A_2G + |\lambda|A_2)\\ &\le \rho(A_1 + A_2G + |\lambda_2|A_2)\\ &< 1.\end{aligned}$$

In the last step we used the fact that

$$\rho(A_1 + A_2G + |\lambda_2|A_2) < \rho(A_1 + A_2G + A_2) < 1.$$

It follows that 0 does not belong to $\Lambda(Q)$, and hence Q is nonsingular. Clearly

$$\min\{|\lambda| : \lambda \in \Lambda(Q)\} \ge 1 - \rho(A_1 + A_2G + |\lambda_2|A_2),$$

and the equality holds if λ_2 is a positive real number. □

Since Q is nonsingular,

$$\mathbf{vec}(\triangle H) = Q^{-1}\mathbf{vec}(\triangle B).$$

Hence

$$\begin{aligned}\|\triangle H\|_F &= \|\mathbf{vec}(\triangle H)\|_2\\ &\le \|Q^{-1}\|_2\|\mathbf{vec}(\triangle B)\|_2\\ &= \frac{1}{\sigma_{min}(Q)}\|\triangle B_0 + (\triangle B_1)H + (\triangle B_2)H^2\|_F\\ &\le \frac{2\sqrt{k}}{\sigma_{min}(Q)}(\|\triangle B_0\|_F + \|\triangle B_1\|_F + \|\triangle B_2\|_F).\end{aligned}$$

Here $\sigma_{min}(Q)$ is the minimum singular value of the matrix Q. Comparing the bound of $\|\triangle G\|_F$ here we have an extra factor 2 because $H = G - \boldsymbol{e}\boldsymbol{u}^T$ is the difference of two stochastic matrices.

So we may view $1/\sigma_{min}(Q)$ as a condition number of the shifted equation (3.1). Even though $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\} \le 1/\sigma_{min}(Q)$, we may consider $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\}$ as a number which reflects the conditioning of the shifted equation (3.1). Since $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\} \le 1/(1 - \rho(A_1 + A_2G + \bar{\theta}A_2))$,

$$\frac{1}{1 - \rho(A_1 + A_2G + \bar{\theta}A_2)}$$

is a number which reflects the conditioning of the shifted equation (3.1). The inequality

$$\frac{1}{1 - \rho(A_1 + A_2G + \bar{\theta}A_2)} < \frac{1}{1 - \rho(A_1 + A_2G + A_2)}$$

suggests that the shifted equation (3.1) has a better conditioning than the original equation (1.1).

6. Numerical results. We have tested the cyclic reduction algorithm and the shifted cyclic reduction algorithm on the examples in [13] using Matlab. In the case of the cyclic reduction algorithm, we stopped when

$$\|\widehat{A}_1^{(j)} - \widehat{A}_1^{(j-1)}\|_\infty \leq 10^{-12},$$

and we accepted

$$\tilde{G} = \left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$$

as an approximation of G. Similarly, in the case of a shifted cyclic reduction, we stopped when

$$\|\widehat{B}_1^{(j)} - \widehat{B}_1^{(j-1)}\|_\infty \leq 10^{-12},$$

and we accepted

$$\tilde{G} = \left(I - \widehat{B}_1^{(j)}\right)^{-1} A_0$$

as an approximation of G. For each example we have reported tables with the number of iterations, the residual error, Res., which is defined by $\|\tilde{G} - A_0 - A_1\tilde{G} - A_2\tilde{G}^2\|_\infty$, and the closeness to stochasticity, Stoc., which is defined by $\|\tilde{G}\boldsymbol{e} - \boldsymbol{e}\|_\infty$, for standard and shifted cyclic reduction. For particular examples we have also reported tables with the values of $\bar{\theta}$, of σ, $1/\sigma$, and $\bar{\theta}/\sigma$, to show the reduction of the rate of convergence, and tables with $1 - \rho(A_1 + A_2G + A_2)$, $1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$, $\sigma_{min}(W)$, $\sigma_{min}(Q)$ that show the conditioning of the original and shifted matrix equations. Even though we proved that $1-\rho(A_1+A_2G+A_2) < 1-\rho(A_1+A_2G+\bar{\theta}A_2)$, it does not mean that $\sigma_{min}(W) < \sigma_{min}(Q)$. See the first row of columns $\sigma_{min}(W)$ and $\sigma_{min}(Q)$ in Table 6.6.

Example 1. In this example we define 24×24 matrices A_0', A_1', and A_2' as follows:

$$(A_0') = \begin{cases} 192(1 - i/24), & i = j, \\ 0, & i \neq j, \end{cases} \qquad (A_2') = \begin{cases} 192\rho_d, & i = j, \\ 0, & i \neq j, \end{cases}$$

and

$$(A_1') = \begin{cases} ar(M - i)/M, & i - j = -1, \\ ir, & i - j = 1, \\ \alpha_i, & i - j = 0, \\ 0 & \text{elsewhere.} \end{cases}$$

Here i and j are integers between 0 and 23, a, r, M, and ρ_d are parameters, and α_i are such that $(A_0' + A_1' + A_2')\boldsymbol{e} = \boldsymbol{0}$. Now A_0, A_1, and A_2 are $A_0 = -(A_1')^{-1}A_0'$, $A_1 = 0$, $A_2 = -(A_1')^{-1}A_2'$. Tables 6.1, 6.2, and 6.3 report the results obtained by choosing different values of M while we fix $r = 1/300$, $a = 18.244$, and $\rho_d = 0.280$. Tables 6.4, 6.5, and 6.6 report the results obtained by choosing different values of ρ_d while we fix $r = 1/100$, $a = 18.244$, and $M = 512$.

TABLE 6.1
Example 1: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$.

	Cyclic reduction			Shifted cyclic reduction		
M	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
64	19	$1.6 \cdot 10^{-16}$	$3.2 \cdot 10^{-12}$	18	$5.1 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
128	20	$2.2 \cdot 10^{-16}$	$8.4 \cdot 10^{-13}$	19	$6.9 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
256	21	$3.0 \cdot 10^{-16}$	$5.9 \cdot 10^{-12}$	19	$4.6 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
512	22	$2.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-12}$	19	$4.7 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
1024	23	$2.4 \cdot 10^{-16}$	$2.1 \cdot 10^{-11}$	19	$5.3 \cdot 10^{-16}$	$5.6 \cdot 10^{-16}$
2048	24	$3.0 \cdot 10^{-16}$	$6.1 \cdot 10^{-11}$	19	$5.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
4096	25	$2.6 \cdot 10^{-16}$	$5.4 \cdot 10^{-11}$	19	$5.0 \cdot 10^{-17}$	$6.7 \cdot 10^{-16}$
8192	26	$3.1 \cdot 10^{-16}$	$3.9 \cdot 10^{-10}$	19	$4.8 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
16384	27	$2.2 \cdot 10^{-16}$	$2.0 \cdot 10^{-11}$	19	$5.9 \cdot 10^{-17}$	$6.7 \cdot 10^{-16}$
32768	29	$1.3 \cdot 10^{-16}$	$1.1 \cdot 10^{-9}$	19	$3.5 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
65536	34	$2.2 \cdot 10^{-16}$	$5.5 \cdot 10^{-8}$	19	$5.0 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$

TABLE 6.2
Example 1: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$.

M	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
64	0.9999	1.0002	0.9998	0.9997
128	0.9999	1.0001	0.9999	0.9998
256	0.9999	1.000039	0.999961	0.9998
512	0.9999	1.000019	0.999981	0.9998
1024	0.9999	1.000009	0.999991	0.9998
2048	0.9999	1.000004	0.999996	0.9998
4096	0.9999	1.000002	0.999998	0.9999
8192	0.9999	1.000001	0.999999	0.9999
16384	0.9999	1.0000004	0.9999996	0.9999
32768	0.9999	1.0000001	0.9999999	0.9999
65536	0.9999	1.000000002	0.999999998	0.9999

TABLE 6.3
Example 1: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$.

M	$1 - \rho(A_1 + A_2G + A_2)$	$1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
64	$1.4 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$2.7 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$
8192	$7.1 \cdot 10^{-7}$	$1.1 \cdot 10^{-4}$	$1.5 \cdot 10^{-7}$	$1.5 \cdot 10^{-5}$
65536	$1.6 \cdot 10^{-9}$	$1.0 \cdot 10^{-4}$	$3.5 \cdot 10^{-10}$	$1.5 \cdot 10^{-5}$

Example 2. In this example we construct a QBD problem defined by the $k \times k$ matrices $A_0 = R + \delta I$, $A_1 = A_2 = R$, where R is a matrix having null diagonal entries and constant off-diagonal entries, and $0 < \delta < 1$. As was observed in [13], the rate $\rho = p^T(A_1 + 2A_2)e$, where $p^T(A_0 + A_1 + A_2) = p^T$, $p^Te = 1$, is exactly $1 - \delta$. We have tested with eight different δ values. Tables 6.7, 6.8, and 6.9 report the results obtained with $k = 16$, and Tables 6.10 and 6.11 report the results obtained with sizes $k = 32$ and $k = 64$, respectively.

TABLE 6.4
Example 1: $r = 1/100$, $a = 18.244$, $M = 512$.

	Cyclic reduction			Shifted cyclic reduction		
ρ_d	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
0.01	6	$2.2 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$	6	$3.9 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
0.025	7	$3.3 \cdot 10^{-16}$	$6.6 \cdot 10^{-16}$	7	$5.7 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.05	10	$1.5 \cdot 10^{-16}$	$8.9 \cdot 10^{-16}$	10	$3.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.075	12	$2.2 \cdot 10^{-16}$	$9.0 \cdot 10^{-15}$	12	$6.1 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.1	14	$2.2 \cdot 10^{-16}$	$2.2 \cdot 10^{-15}$	14	$4.2 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.12	14	$2.2 \cdot 10^{-16}$	$2.8 \cdot 10^{-14}$	14	$3.9 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.14	15	$2.2 \cdot 10^{-16}$	$2.1 \cdot 10^{-13}$	15	$5.1 \cdot 10^{-17}$	$4.4 \cdot 10^{-16}$
0.16	16	$2.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-14}$	16	$5.3 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.18	16	$2.2 \cdot 10^{-16}$	$5.9 \cdot 10^{-14}$	16	$4.7 \cdot 10^{-17}$	$3.3 \cdot 10^{-16}$
0.2	17	$2.2 \cdot 10^{-16}$	$1.0 \cdot 10^{-14}$	16	$4.5 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
0.22	18	$2.2 \cdot 10^{-16}$	$4.3 \cdot 10^{-13}$	17	$6.0 \cdot 10^{-16}$	$3.3 \cdot 10^{-16}$
0.24	18	$2.3 \cdot 10^{-16}$	$7.8 \cdot 10^{-13}$	17	$3.7 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
0.26	19	$1.9 \cdot 10^{-16}$	$8.6 \cdot 10^{-13}$	17	$5.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.28	20	$2.3 \cdot 10^{-16}$	$3.5 \cdot 10^{-12}$	17	$5.1 \cdot 10^{-16}$	$3.3 \cdot 10^{-16}$
0.29	22	$2.6 \cdot 10^{-16}$	$1.8 \cdot 10^{-11}$	17	$3.7 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
0.29568	31	$2.6 \cdot 10^{-16}$	$1.9 \cdot 10^{-8}$	17	$6.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$

TABLE 6.5
Example 1: $r = 1/100$, $a = 18.244$, $M = 512$.

ρ_d	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
0.01	0.9998	4.3182	0.2316	0.2315
0.025	0.9998	1.7714	0.5645	0.5644
0.05	0.9998	1.0814	0.9248	0.9246
0.075	0.9998	1.0170	0.9833	0.9831
0.1	0.9998	1.0063	0.9938	0.9936
0.12	0.9998	1.0034	0.9966	0.9964
0.14	0.9997	1.0021	0.9979	0.9977
0.16	0.9997	1.0013	0.9987	0.9984
0.18	0.9997	1.0009	0.9991	0.9988
0.2	0.9997	1.0006	0.9994	0.9991
0.22	0.9997	1.0004	0.9996	0.9993
0.24	0.9996	1.0002	0.9998	0.9994
0.26	0.9996	1.0001	0.9999	0.9995
0.28	0.9996	1.0001	0.9999	0.9995
0.29	0.9996	1.000019	0.99998	0.9995
0.29568	0.9995	1.00000003	0.99999997	0.9995

TABLE 6.6
Example 1: $r = 1/100$, $a = 18.244$, $M = 512$.

p	$1 - \rho(A_1 + A_2G + A_2)$	$1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
0.01	0.6265	0.6265	0.6259	0.6175
0.28	$4.0 \cdot 10^{-5}$	$3.4 \cdot 10^{-4}$	$8.7 \cdot 10^{-6}$	$5.3 \cdot 10^{-5}$
0.29568	$1.9 \cdot 10^{-8}$	$3.2 \cdot 10^{-4}$	$3.6 \cdot 10^{-9}$	$4.4 \cdot 10^{-5}$

TABLE 6.7
Example 2: $k = 16$.

	Cyclic reduction			Shifted cyclic reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$5.7 \cdot 10^{-16}$	$1.3 \cdot 10^{-15}$	5	$2.8 \cdot 10^{-16}$	$3.3 \cdot 10^{-16}$
10^{-2}	11	$6.5 \cdot 10^{-16}$	$7.2 \cdot 10^{-15}$	4	$3.7 \cdot 10^{-16}$	0.0
10^{-3}	14	$6.7 \cdot 10^{-16}$	$9.7 \cdot 10^{-14}$	4	$1.9 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
10^{-4}	17	$6.4 \cdot 10^{-16}$	$8.8 \cdot 10^{-13}$	4	$4.5 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
10^{-5}	21	$1.1 \cdot 10^{-15}$	$1.2 \cdot 10^{-11}$	4	$3.5 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
10^{-6}	24	$6.5 \cdot 10^{-16}$	$5.8 \cdot 10^{-11}$	4	$2.5 \cdot 10^{-16}$	$8.9 \cdot 10^{-16}$
10^{-7}	27	$6.3 \cdot 10^{-16}$	$1.4 \cdot 10^{-9}$	4	$2.5 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
10^{-8}	29	$7.0 \cdot 10^{-16}$	$3.5 \cdot 10^{-9}$	4	$2.5 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$

TABLE 6.8
Example 2: $k = 16$.

δ	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
10^{-1}	0.0783	1.3333	0.75	0.0587
10^{-2}	0.0174	1.0303	0.9706	0.0140
10^{-3}	0.0207	1.0030	0.9970	0.0207
10^{-4}	0.0216	1.0003	0.9997	0.0216
10^{-5}	0.0217	1.00003	0.99997	0.0217

TABLE 6.9
Example 2: $k = 16$.

δ	$1 - \rho(A_1 + A_2G + A_2)$	$1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
10^{-1}	0.1	0.3765	0.1	0.3234
10^{-5}	$1.0 \cdot 10^{-5}$	0.3261	$1.0 \cdot 10^{-5}$	0.2310
10^{-8}	$1.0 \cdot 10^{-8}$	0.3261	$1.0 \cdot 10^{-8}$	0.2721

TABLE 6.10
Example 2: $k = 32$.

	Cyclic reduction			Shifted cyclic reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$1.3 \cdot 10^{-15}$	$1.1 \cdot 10^{-15}$	5	$1.2 \cdot 10^{-15}$	$8.9 \cdot 10^{-16}$
10^{-2}	11	$9.5 \cdot 10^{-16}$	$1.1 \cdot 10^{-14}$	4	$4.1 \cdot 10^{-16}$	$7.8 \cdot 10^{-16}$
10^{-3}	14	$1.4 \cdot 10^{-15}$	$8.0 \cdot 10^{-14}$	4	$5.4 \cdot 10^{-16}$	$1.1 \cdot 10^{-15}$
10^{-4}	17	$1.6 \cdot 10^{-15}$	$3.7 \cdot 10^{-12}$	4	$6.8 \cdot 10^{-16}$	$1.8 \cdot 10^{-15}$
10^{-5}	21	$1.1 \cdot 10^{-15}$	$1.0 \cdot 10^{-11}$	4	$5.1 \cdot 10^{-16}$	$6.6 \cdot 10^{-16}$
10^{-6}	24	$1.2 \cdot 10^{-15}$	$2.1 \cdot 10^{-10}$	4	$5.5 \cdot 10^{-16}$	$8.9 \cdot 10^{-16}$
10^{-7}	27	$1.1 \cdot 10^{-15}$	$7.9 \cdot 10^{-10}$	4	$6.5 \cdot 10^{-16}$	$1.0 \cdot 10^{-15}$
10^{-8}	29	$8.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-8}$	4	$7.1 \cdot 10^{-16}$	$7.8 \cdot 10^{-16}$

TABLE 6.11
Example 2: $k = 64$.

	Cyclic reduction			Shifted cyclic reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$2.8 \cdot 10^{-15}$	$4.9 \cdot 10^{-15}$	5	$2.0 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$
10^{-2}	11	$3.1 \cdot 10^{-15}$	$1.1 \cdot 10^{-14}$	4	$1.4 \cdot 10^{-15}$	$2.7 \cdot 10^{-15}$
10^{-3}	14	$3.1 \cdot 10^{-15}$	$1.8 \cdot 10^{-13}$	4	$1.3 \cdot 10^{-15}$	$1.1 \cdot 10^{-15}$
10^{-4}	17	$2.4 \cdot 10^{-15}$	$4.6 \cdot 10^{-12}$	4	$1.2 \cdot 10^{-15}$	$2.9 \cdot 10^{-15}$
10^{-5}	21	$2.3 \cdot 10^{-15}$	$2.9 \cdot 10^{-11}$	4	$1.6 \cdot 10^{-15}$	$4.4 \cdot 10^{-16}$
10^{-6}	24	$3.3 \cdot 10^{-15}$	$2.8 \cdot 10^{-10}$	4	$1.4 \cdot 10^{-15}$	$2.4 \cdot 10^{-15}$
10^{-7}	27	$2.4 \cdot 10^{-15}$	$1.9 \cdot 10^{-10}$	4	$1.3 \cdot 10^{-15}$	$3.8 \cdot 10^{-15}$
10^{-8}	29	$2.6 \cdot 10^{-15}$	$1.7 \cdot 10^{-8}$	4	$1.2 \cdot 10^{-15}$	$3.3 \cdot 10^{-15}$

REFERENCES

[1] D. A. Bini, L. Gemignani, and B. Meini, *Factorization of analytic functions by means of Koenig's theorem and Toeplitz computations*, Numer. Math., 89 (2001), pp. 49–82.

[2] D. A. Bini and B. Meini, *On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic, Dordrecht, The Netherlands, 1995, pp. 21–38.

[3] D. A. Bini and B. Meini, *On the solution of a nonlinear matrix equation arising in queueing problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 906–926.

[4] D. A. Bini and B. Meini, *Improved cyclic reduction for solving queueing problems*, Numer. Algorithms, 15 (1997), pp. 57–74.

[5] D. A. Bini and B. Meini, *Inverting block Toeplitz matrices in block Hessenberg form by means of displacement operators: Application to queueing problems*, Linear Algebra Appl., 272 (1998), pp. 1–16.

[6] D. A. Bini and B. Meini, *Effective methods for solving banded Toeplitz systems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 700–719.

[7] H. R. Gail, S. L. Hantler, and B. A. Taylor, *Use of characteristics roots for solving infinite state Markov chains*, in Computational Probability, W. K. Grassmann, ed., Kluwer Academic, Dordrecht, The Netherlands, 2000, pp. 205–255.

[8] G. Golub and C. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989.

[9] G. Latouche, *A note on two matrices occurring in the solution of quasi-birth-and-death processes*, Comm. Statist. Stochastic Models, 3 (1987), pp. 251–257.

[10] G. Latouche and V. Ramaswami, *A logarithmic reduction algorithm for quasi-birth-death processes*, J. Appl. Probab., 30 (1993), pp. 650–674.

[11] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM Series on Statistics and Applied Probability 5, SIAM, Philadelphia, PA, 1999.

[12] G. Latouche and G. Stewart, *Numerical methods for M/G/1 type queues*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic, Dordrecht, The Netherlands, 1995, pp. 571–581.

[13] B. Meini, *Solving QBD problems: The cyclic reduction algorithm versus the invariant subspace method*, Adv. Perf. Anal., 1 (1998), pp. 215–225.

[14] V. Naoumov, *Matrix-multiplicative approach to quasi-birth-and-death processes analysis*, in Matrix-Analytic Methods in Stochastic Models, A. S. Alfa and S. Chakravarthy, eds., Dekker, New York, 1997, pp. 85–106.

[15] V. Naoumov, U. R. Krieger, and D. Wagner, *Analysis of a multiserver delay-loss system with a general Markovian arrival process*, in Matrix-Analytic Methods in Stochastic Models, A. S. Alfa and S. Chakravarthy, eds., Dekker, New York, 1997, pp. 43–66.

[16] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Dekker, New York, 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 692–705
© 2002 Society for Industrial and Applied Mathematics

A FACTORED APPROXIMATE INVERSE PRECONDITIONER WITH PIVOTING*

MATTHIAS BOLLHÖFER† AND YOUSEF SAAD‡

Abstract. In this paper we develop new techniques for stabilizing factored approximate inverse preconditioners (AINV) using pivoting. This method yields stable preconditioners in many cases and can provide successful preconditioners in many situations when the underlying system is highly indefinite. Numerical examples illustrate the effectiveness of this approach.

Key words. sparse matrices, incomplete LU factorization, sparse approximate inverse, factored approximate inverse preconditioners, pivoting

AMS subject classifications. 65F05, 65F10, 65F50

PII. S0895479800372122

1. Introduction.

Many applications lead to solving large sparse linear systems of the form

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

with $A \in \mathbb{R}^{n,n}$ and $b \in \mathbb{R}^n$. In many cases, such systems are not only very large but also exceedingly difficult to solve by iterative techniques because A is ill-conditioned or highly indefinite or both. In some instances these equations arise from special applications, and solvers tailored to the underlying physical problem may give the best results. However, there are situations in which "general purpose" solvers are desirable. Such is the case when building general purpose software, or when the linear system has very little inherent structure. General purpose solvers have many other advantages, the most significant being that changes in the physics or model do not require the development of new methods. For these situations, preconditioned Krylov-subspace solvers (see, e.g., [16, 25, 13]) are often seen as promising alternatives to "black-box" direct solution methods. Among all preconditioning techniques, those based on incomplete LU factorizations (see, e.g., [20, 21, 22]) are known to give excellent results for many important classes of problems, such as those arising from the discretization of elliptic partial differential equations.

Motivated by the emergence of parallel computing platforms, researchers have developed a number of new techniques in recent years, which approximate directly the inverse of A. A few of these approaches are based on minimizing the norm $\|I - AM\|$ in some appropriate norm [19, 17, 15, 9], while others approximately solve the equation $W^\top AZ = D$, where the unknown matrices Z, W are unit upper triangular and D is a diagonal matrix; see, e.g., [24, 4, 5, 1, 18]. In particular, the algebraic behavior of the latter class of methods bears strong similarities to that of incomplete LU

*Received by the editors May 11, 2000; accepted for publication (in revised form) by E. Ng July 12, 2001; published electronically January 11, 2002.
http://www.siam.org/journals/simax/23-3/37212.html

†Institute of Mathematics, MA 4-5, Berlin University of Technology, D-10623 Berlin, Germany (bolle@math.tu-berlin.de, http://www.math.tu-berlin.de/∼bolle/). This author was supported by grants from the DFG BO 1680/1-1 and by the University of Minnesota. He performed his research while visiting the University of Minnesota at Minneapolis.

‡Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CSci Building, 200 Union Street SE, Minneapolis, MN 55455-0154 (saad@cs.umn.edu, http://www.cs.umn.edu/∼saad/). This author was supported by NSF and by the Minnesota Supercomputing Institute.

decompositions, e.g., they are stable for M- and H-matrices. See [6] for a detailed analysis of factored approximate inverses and incomplete LU decompositions. Without describing the details of these relations we briefly sketch both methods to describe these links.

For solving the linear system (1.1), incomplete LU techniques begin by approximately constructing a factorization

$$A \approx LDU,$$

where $L, U^\top$ are lower triangular matrices, with unit diagonal, and D is diagonal. One way to construct these decompositions is to partition A as

$$A = \begin{bmatrix} B & F \\ E & C \end{bmatrix} \in \mathbb{R}^{n,n}$$

with $B \in \mathbb{R}$ and the other blocks have corresponding size. Then A is factored as

$$\begin{bmatrix} B & F \\ E & C \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ L_E & I \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} D_B & 0 \\ 0 & S \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & U_F \\ 0 & I \end{bmatrix}}_{U}, \tag{1.2}$$

where

$$S = C - L_E D_B U_F \in \mathbb{R}^{n-k,n-k} \tag{1.3}$$

denotes the so-called Schur complement. The exact LU decomposition of A (if it exists) can be obtained by successively applying (1.2) to the Schur complement S. Even if there exists a decomposition (1.2) for A and for S, there is no need to compute L_E and U_F, S exactly when constructing a preconditioner. A common approach for reducing fill-in consists of discarding entries of L_E and U_F of small size and defining the approximate Schur complement only with these sparsified vectors $\tilde{L}_E$ and $\tilde{U}_F$. We will use

$$\tilde{S} = B - \tilde{L}_E D_B \tilde{U}_F \tag{1.4}$$

as one possible definition of an approximate Schur complement. The associated ILU algorithm is roughly given by Algorithm 1.

ALGORITHM 1 (incomplete LU factorization (ILU)).
Let $A = (a_{ij})_{ij} \in \mathbb{R}^{n,n}$ and let $\tau \in (0,1)$ be a drop tolerance. Compute $A \approx LDU$.
Set $L = U = I$, $S = A$.
for $i = 1, \ldots, n$
  $d_{ii} = s_{ii}$
  for $j = i+1, \ldots, n$
    $p_j = s_{ji}/d_{ii}, \quad q_j = s_{ij}/d_{ii}$
    drop entries $|p_j|, |q_j|$ if they are less than τ
    $l_{ji} = p_j, \quad u_{ij} = q_j$
    for $k = i+1, \ldots, n$
      $s_{kj} = s_{kj} - l_{ki} d_{ii} u_{ij}$
    end
  end
end

Practical versions of incomplete LU decompositions are typically implemented in a slightly different way. It is usually not advisable to update the whole matrix $\hat{S} = (s_{kl})_{k,l \geqslant i}$ by a rank-1 modification. Instead, the leading row of $\hat{S}$ is typically computed, and the transformations on the other rows are postponed. In essence this means that the so-called I, K, J version of Gaussian elimination is used. For details, see [24]. In addition to saving memory, this has the advantage that all updates and modifications are performed only once for each row, thus making it possible to use very simple sparse row storage schemes such as the compressed sparse row (CSR) format.

In [4, 5] algorithms have been presented that directly compute upper unit triangular matrices W and Z such that $W^\top AZ \approx D$ is approximately diagonal. Here we choose to outline a version (Algorithm 2) that has already been used for the symmetric positive definite case [1, 18]. In short, at any given step i the algorithm performs a Gram–Schmidt step to A-orthogonalize the columns $w_j, j = i+1, \ldots, n$, against z_i and then the columns $z_j, j = i+1, \ldots, n$, against w_i. Dropping is then applied to the resulting columns. Clearly, A-orthogonality is only achieved approximately.

ALGORITHM 2 (factored approximate inverse (AINV)).
Let $A = (a_{ij})_{ij} \in \mathbb{R}^{n,n}$ and let $\tau \in (0,1)$ be a drop tolerance. Compute $A^{-1} \approx ZD^{-1}W^\top$.
Set $p = q = (0, \ldots, 0) \in \mathbb{R}^n$, $Z = [z_1, \ldots, z_n] = I_n$, $W = [w_1, \ldots, w_n] = I_n$.
for $i = 1, \ldots, n$
  $d_{ii} = w_i^\top Az_i$
  for $j = i+1, \ldots, n$
    $p_j = \left(w_j^\top Az_i\right)/d_{ii}, \quad q_j = \left(w_i^\top Az_j\right)/d_{ii}$
    $w_j = w_j - w_i\, p_j, \quad z_j = z_j - z_i\, q_j$
    for all $l \leqslant i$: drop entries w_{lj}, z_{lj} if their absolute values are less than τ.
  end
end

Suppose that Algorithms 1 and 2 do not break down. In step i of Algorithm 1, entries in the ith row and ith column of the Schur complement $(s_{kl})_{k,l \geqslant i}$ are eliminated. Analogously Algorithm 2 eliminates the off-diagonal entries in row i and column i of $(w_k^\top Az_l)_{k,l \geqslant i}$. If no dropping is applied, then we would obtain

$$(s_{kl})_{k,l \geqslant i} = \left(w_k^\top Az_l\right)_{k,l \geqslant i}. \tag{1.5}$$

This can been seen, e.g., from (1.2) using $L^{-1} = W^\top$ and $U^{-1} = Z$. This means that p_j and q_j (resp., d_{ii}) play similar roles in both algorithms.

The main advantage of this observation is the possibility of exploiting these connections to adapt pivoting techniques used in (incomplete) Gaussian elimination and to carry them over to sparse approximate inverse techniques, with the goal of improving the performance of factored approximate inverses.

2. Approximate inverses with pivoting. One way to exploit the connection between approximate inverses and ILUs is to introduce pivoting to approximate inverses. This can be done by first adding pivoting to Algorithm 1 and then using relation (1.5) to transfer pivoting strategies to Algorithm 2. The main reason for introducing pivoting to Algorithms 1 and 2 is the fact that the algorithms might encounter a zero or small pivot during the computation. At some step i of either algorithm it may turn out that $d_{ii} \approx 0$, in which case the algorithms break down. It is possible to shift the zero pivots away by adding an artificial small perturbation

(e.g., 10^{-8}) to d_{ii}, but this will rarely solve the problem. Instead, we could ensure that zero pivots do not occur, and this is traditionally achieved by pivoting in direct Gaussian elimination. This technique was implemented in incomplete factorizations as well [22]. Although this makes the underlying data structure more complex, it often stabilizes the processes and even ensures that the growth in the element size of L, U will remain fairly moderate. However, unlike complete Gaussian elimination, ILU with pivoting might still break down.

We first discuss how column and row pivoting could be added to Algorithm 1. We will introduce column and row interchanges that keep the algorithm consistent when $\tau = 0$ is used. In other words, the algorithm without dropping will compute $\Pi^\top A\Sigma = LDU$, where Π and Σ are permutation matrices that will be determined throughout the process. If π and σ are permutation vectors associated with permutation matrices Π and Σ, then we will sometimes write $A(\pi, \sigma)$ for the permuted matrix $\Pi^\top A\Sigma$.

Suppose that a diagonal pivot s_{ii} is not satisfactory. The property to be satisfied by, say, a column pivot $k \geq i$ at step i, could be a criterion such as

$$|s_{ik}| \geq \alpha \, |s_{ij}| \quad \text{for} \quad j \geq i$$

for a prescribed constant $0 < \alpha \leqslant 1$. After the column interchange takes place, one could consider in addition the analogous row criterion

$$|s_{ii}| \geq \alpha \, |s_{ji}| \quad \text{for} \quad j \geq i.$$

If this inequality is no longer satisfied, a row interchange will be performed. This process can alternate between both criteria and usually takes a few steps to complete, in many cases requiring just one step. It allows a better selection by iterating on the choice of the pivots, if necessary, without entailing substantial additional cost in most cases. With pivoting, (1.2) locally changes to

$$\underbrace{\begin{bmatrix} I & \mathrm{O} \\ \mathrm{O} & \hat{\Pi} \end{bmatrix}^\top}_{\Pi^\top} \begin{bmatrix} B & F \\ E & C \end{bmatrix} \underbrace{\begin{bmatrix} I & \mathrm{O} \\ \mathrm{O} & \hat{\Sigma} \end{bmatrix}}_{\Sigma} \approx \begin{bmatrix} I & \mathrm{O} \\ \hat{\Pi}^\top L_E & I \end{bmatrix} \begin{bmatrix} D_B & \mathrm{O} \\ \mathrm{O} & \hat{\Pi}^\top S \hat{\Sigma} \end{bmatrix} \begin{bmatrix} I & U_F \hat{\Sigma} \\ \mathrm{O} & I \end{bmatrix}, \tag{2.1}$$

where equality holds, if no dropping is applied. The approximate identity (2.1) shows that the columns of $U - I$ also need to be interchanged with respect to the column pivoting step. The rows of $L - I$ need to be processed similarly if row pivoting is applied. The additional row pivoting step is not so common in practice.

This may be more expensive due to the additional overhead for computing not only the leading row of the Schur complement, but also its leading column. In particular, we note that this version of pivoting is hard to implement with the common I, K, J variant of Gaussian elimination, since columns of the Schur complement are not available and their corresponding data structure is expensive to obtain.

After showing how pivoting affects Algorithm 1 it is now easy to extend naturally the idea of pivoting to Algorithm 2. To do so we only have to keep in mind that $W = L^{-\top}$ and $Z = U^{-1}$ if $\tau = 0$ is used. Clearly the columns of $W - I$ and $Z - I$ have to be permuted analogously to the rows of $L - I$ and columns of $U - I$. If no dropping is applied, then (1.5) now reads

$$(s_{kl})_{k,l \geqslant i} = \left(w_k^\top A(\pi, \sigma) z_l\right)_{k,l \geqslant i}. \tag{2.2}$$

This restricts the application of π and σ to the initial matrix A if we reorder columns of $Z - I$ and $W - I$. This leads to Algorithm 3.

ALGORITHM 3 (factored approximate inverse with pivoting (AINVP)).
Let $A = (A_{ij})_{ij} \in \mathbb{R}^{n,n}$ *and let* τ *be drop tolerance. Compute* $A^{-1} \approx ZD^{-1}W^\top$.
Set $p = q = (0, \dots, 0) \in \mathbb{R}^n$, $Z = [z_1, \dots, z_n] = I_n$, $W = [w_1, \dots, w_n] = I_n$,
and $\pi = \sigma = (1, \dots, n)$.
for $i = 1, \dots, n$
  while *pivots not satisfactory*
    for all $j \geqslant i$: $p_j = w_j^\top A(\pi, \sigma) z_i$.
    Find a column pivot $k \geqslant i$.
    Interchange columns i, k *of* $Z - I$ *and components* i, k *of* p *and* σ.
    for all $j \geqslant i$: $q_j = w_i^\top A(\pi, \sigma) z_j$.
    Find a row pivot $l \geqslant i$.
    Interchange columns i, l *of* $W - I$ *and components* i, l *of* q *and* π.
  end
  $d_{ii} = p_i$.
  for $j = i + 1, \dots, n$
    $p_j = p_j / d_{ii}, \quad q_j = q_j / d_{ii}$.
    $w_j = w_j - w_i p_j, \quad z_j = z_j - z_i q_j$.
    for all $l \leqslant i$: *drop entries* w_{lj}, z_{lj}, *if their absolute values are less than* τ.
  end
end

Note that the while-loop in Algorithm 3 is optional and has been included for the purpose of greater generality. If no dropping is applied, we obtain $W^\top A(\pi, \sigma) Z = D$, by construction. Pivoting for a related direct projection method can already be found in [3]. Instead of using $W^\top AZ$ to compute p and q, only $W^\top A$ is used, which is equivalent in this case because no dropping is applied. In a sense Algorithm 3 generalizes the pivoting approach of [3] in that it is applied to an incomplete factorization and both row and column interchanges are performed.

As it is described, Algorithm 3 does not specify any rule on how to select the pivots k and l. A reasonable strategy could be to choose k such that $|p_k|$ is maximal, and this obviously requires p to be computed before pivoting is applied. We observe that in Algorithm 3 now the p and q columns are inside the while-loop which searches for adequate pivots. Indeed, p and q must be recomputed whenever q (resp., p) requires an interchange. In the situation when one pivoting step is applied for any i, there is no need to recompute p and q. However, when more than one pivoting step is required, one of p or q at least must be recomputed. In this case the algorithm incurs some additional overhead. Clearly, any pivoting strategy should try to keep this additional overhead small. For better stability in Algorithm 3 the pivot p_k should satisfy

$$|p_k| \geqslant \alpha \max_m |p_m|$$

for some constant $0 < \alpha \leqslant 1$, e.g., $\alpha = 0.1$. However, since Algorithm 3 is a biorthogonalization technique, i.e., the outcome is to compute Z *and* W, we should ensure that $|p_k| = |q_k| \geqslant \alpha \max_m |q_m|$ is also fulfilled to guarantee that the entries of both factors Z and W are sufficiently bounded. Algorithm 4 gives us a simple and relatively inexpensive strategy for controlling the growth of the entries in Z and W and for stabilizing Algorithm 3.

ALGORITHM 4 (controlled pivoting).
Prescribe a tolerance $\alpha \in (0, 1]$, *e.g.,* $\alpha = 0.1$
satisfied_p=false, satisfied_q=false
while *not satisfied_p*

for all $j \geqslant i$: $p_j = w_j^\top A(\pi,\sigma)z_i$
if $|p_i| < \alpha \max_m |p_m|$
satisfied_q=false, choose k *such that* $|p_k| = \max_m |p_m|$.
Interchange column i *and* k *of* $Z - I$ *and components* i *and* k *of* σ.
end
satisfied_p=true
if *not satisfied_q*
for all $j \geqslant i$: $q_j = w_i^\top A(\pi,\sigma)z_j$
end
if $|q_i| < \alpha \max_m |q_m|$.
satisfied_p=false, choose l *such that* $|q_l| = \max_m |q_m|$
Interchange column i *and* l *of* $W - I$ *and components* i *and* l *of* π.
end
satisfied_q=true
end

An additional improvement for preventing too many pivoting steps might be to prescale the rows and/or columns of A. As a rule, more than one pivoting step, say, column pivoting, is performed in any given step of Algorithm 3. However, in order for $|p_i| \geqslant \alpha \max_m |p_m|$ to imply that $|p_i| = |q_i| \geqslant \alpha \max_m |q_m|$ it is necessary that entries of q have magnitudes that are comparable to those of p. By (2.2), p and q are the first column/row of

$$\left(w_k^\top A(\pi,\sigma)z_l\right)_{k,l\geqslant i}$$

before a further step of pivoting is applied. If W and Z are moderately bounded, which is more or less achieved by pivoting, then $A(\pi,\sigma)$ having rows of comparable absolute row sums might be a good start to prevent too many pivoting steps.

One might of course think of other pivoting strategies, especially in the context of parallel computations. Pivoting is undoubtedly harder to implement in parallel. As is often done, however, it is possible to exploit relaxed pivoting to search for satisfactory pivots locally, i.e., in each processor. This means that k and l are restricted to a certain subset to maintain distributed storage schemes. Other strategies could be, for example, to restrict k and l in order to keep the lower right $(n-i)\times(n-i)$ part of $W^\top A(\pi,\sigma)Z$ as sparse as possible.

3. Numerical results. This section presents numerical experiments to validate the algorithms. Additional details and comments on the implementations of the algorithms will also be provided.

- All input matrices are assumed to be given in the CSR format [24].
- The matrices are initially scaled such that they have unit 1-norm for any row. As mentioned in section 2 this is done to reduce the number of necessary column/row interchanges.
- $W^\top$ and $Z^\top$ are stored in CSR format.
- Interchanges of columns of W (resp., Z) are performed by interchanging only the references (pointers) instead of the whole data array.
- The computation of p and q in Algorithm 3 requires a multiplication $A(\pi,\sigma)z_i$, $A(\pi,\sigma)^\top w_i$. To be efficient, this operation must be done in sparse-sparse mode. If A is given in CSR format, only $A(\pi,\sigma)^\top w_i$ is easy to access, while $A(\pi,\sigma)z_i$ requires $A^\top$ to be stored in CSR format. For this purpose we initially compute the pattern of $A^\top$ in CSR format, but we omit the numerical values. The nonzero components of the computed vectors $A(\pi,\sigma)z_i$,

$A(\pi,\sigma)^\top w_i$ are stored as a full vector but with an additional index list of the nonzeros. The index list is important to inherit the sparse nature of this vector when the computations are performed. Otherwise dense computations would slow down the algorithm. See, e.g., [24, p. 291] for this kind of technique. Finally to compute, for all $j \geqslant i$, $w_j^\top (A(\pi,\sigma)z_i)$ and $z_j^\top (A(\pi,\sigma)^\top w_i)$ we use a list which contains the nontrivial columns of W and Z, i.e., those columns which contain more than just the diagonal entry. The use of permutation vectors π and σ requires us to have the inverse permutations π^{-1}, σ^{-1} which are computed simultaneously.

- Two values were used for the parameter α which controls the pivoting process: $\alpha = 0.1$ and $\alpha = 1.0$.
- Two different values were used for the drop tolerance: $\tau = 0.1$ and $\tau = 0.01$.

For the numerical experiments several collections were chosen from the Harwell–Boeing collection [11], the SPARSKIT collection [23], and finally from the Davis collection [10]. Throughout the computations the matrices were initially reordered using the symmetric minimum degree ordering [14]. But clearly for specific problems other orderings can be more beneficial (cf. [2]). Note that for unsymmetric matrices other orderings might be used. See, e.g., [7] for a reordering (MIP) for approximate inverses.

The computations were performed on an IBM RS6000 (44P model 270) under AIX 4.3 with 4GB memory. The approximate inverse algorithms were implemented in C. Dynamic memory allocation in C is a flexible and convenient tool to use relative to a "manual" memory management that would be required under standard FORTRAN 77. However, the overhead is sometimes nonnegligible, since the memory manager cannot take advantage of the underlying structure of the problem, leading to nonoptimal layout of data in memory.

We used GMRES(30) and QMR as iterative solvers. The iteration was stopped after the residual norm was less than $\sqrt{\text{eps}}$ times the initial residual norm, where eps $\approx 2.2204 \cdot 10^{-16}$ denotes the machine precision. For some matrices a smaller tolerance was necessary, since the exact solution $(1,\ldots,1)^\top$ was not sufficiently well approximated. In this case eps was used. The iteration was stopped after 500 steps. Every iterative solution which broke down or did not converge within this number of steps was noted as a failure. The approximate inverse algorithms were compared with the SPARSKIT algorithms ILUT and ILUTP [24] using the same settings.

We briefly describe the results for several matrices and then give detailed numerical results for several selected examples. We focus on examples where we observed major differences for AINV with and without pivoting.

To give a rough idea on how the method performed on the selected collections, we summarize in Table 3.1 which method successfully solved how many problems with respect to the parameters τ and α. The tests were done on 94 matrices from the Harwell–Boeing collection, 26 matrices from the Davis collection, and 58 matrices from the SPARSKIT collection.

From Table 3.1 one gets the impression that Algorithm 3 (AINVP) behaves slightly better than ILUTP. This might be due to the following reasons.

1. AINVP uses column *and* row pivoting to ensure that W *and* Z are well-bounded. Pivoting applied only to the columns, for example, would locally bound only one factor. But this is essentially what ILUTP does. For reasons of efficiency, pivoting with respect to the rows is not done.
2. Dropping in AINVP seems to be less harmful than in ILUTP. Approximation

TABLE 3.1
Summary of results: Total of 178 *matrices. Number of successful computed problems for drop tolerance* τ *and pivot threshold* α.

Preconditioner	Accelerator	Parameters			
		$\tau = 0.1$		$\tau = 0.01$	
		$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$
Harwell–Boeing collection (94 test matrices)					
AINV	GMRES(30)	35		39	
AINV	QMR	38		38	
AINVP	GMRES(30)	57	63	78	86
AINVP	QMR	66	72	85	84
ILUT	GMRES(30)	44		43	
ILUT	QMR	41		44	
ILUTP	GMRES(30)	53	54	69	71
ILUTP	QMR	59	58	74	76
Davis collection (26 matrices)					
AINV	GMRES(30)	14		14	
AINV	QMR	14		14	
AINVP	GMRES(30)	12	16	17	19
AINVP	QMR	15	17	18	20
ILUT	GMRES(30)	14		14	
ILUT	QMR	14		14	
ILUTP	GMRES(30)	13	15	16	18
ILUTP	QMR	14	15	16	17
SPARSKIT collection (58 matrices)					
AINV	GMRES(30)	2		6	
AINV	QMR	4		9	
AINVP	GMRES(30)	3	16	14	32
AINVP	QMR	4	27	17	34
ILUT	GMRES(30)	7		18	
ILUT	QMR	9		20	
ILUTP	GMRES(30)	6	9	19	19
ILUTP	QMR	11	12	25	23

errors caused by dropping in AINVP behave somehow between linear and quadratic with respect to the values that are dropped when regarding the off-diagonal part of $W^\top AZ$. For ILUTP the analogous effect is rational, which means that small perturbation in L, U may cause huge approximation errors in the off-diagonal entries of $L^{-1}AU^{-1}$.

3. AINVP sometimes ends up with more fill-in. In the numerical examples dropping was performed only with respect to a fixed drop tolerance but not with respect to the number of nonzeros. The results show that sometimes AINVP needs significantly more fill-in than ILUTP (e.g., Table 3.6). The higher amount of fill-in then slows down AINVP. But the fill-in could be reduced using more suitable symbolic factorization techniques as well as a reformulation of the algorithm to exploit more zeros [8].

We now comment briefly on some matrix families from the three collections (Harwell–Boeing, SPARSKIT, Davis). We use tables that indicate on which matrix family pivoting showed improvements. The following symbols indicate how strongly the treatment of the matrix family changed when pivoting was used:

++	+	∘	−	−−
much improved	improved	no great changes	worse	much worse

TABLE 3.2

Harwell–Boeing collection. Changes in each matrix family when pivoting is used for drop tolerance τ and pivot threshold α.

Matrix family	$\tau = 0.1$		$\tau = 0.01$	
	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$
ASTROPH	○	○	○	○
CHEMIMP	++	++	++	++
CHEMWEST	+	++	+	++
CIRPHYS	○	○	○	○
ECONAUS	++	++	++	++
FACSIMILE	−	−	○/−	○/−
GEMAT	○	○	○	+
GRENOBLE	○	+	+	○/+
LNS	○	○/+	○/+	○/+
NNCENG	○	○	○	○
NUCL	○	○	○	○/+
OILGEN	○	○/−	○	○
PORES	+/−	○/+	○	○/+
PSMIGR	○/++	○/++	○/+	○/+
SAYLOR	○	○	○	○
SHERMAN	○/+	○/++	○/++	○/++
SMTAPE(BP*)	○	○	+	+/++
SMTAPE(SHL*,STR*)	++	++	++	++
STEAM	○	○	○	○
WATT	○	○	○	○

In general a + is used whenever the choice of parameters moderately improved either the fill-in or the time for the iterative solution (or both) for several matrices of a matrix family. If any of these two criteria were not improved, it had essentially to stay constant. In a similar way − is used. ++ is used if the fill-in or the iterative process were significantly improved for most of the matrices of a matrix family. This includes the case when the iterative process changed from no convergence to convergence within $\min\{n, 500\}$ steps. Again, any criteria had to stay at least constant.

For some matrix families the behavior was not uniform. In these cases we used two symbols. If, for example, for some matrices the behavior was better, but for others it was worse, then we used the symbol +/−.

Table 3.2 gives an overview of the matrix families of the Harwell–Boeing collection. As one can see from Table 3.2 the most significant improvements using pivoting were achieved for the CHEMWEST (chemical engineering), ECONAUS (economic models) families, and the *shl*,str** matrices (linear programming). The opposite behavior was observed when pivoting was applied to the FACSIMILE matrices (chemical kinetics). For representative examples, see Table 3.3 (chemical engineering) and Table 3.4 (thermal simulation, steam injection).

Next are some comments on matrices from the Davis collection. The improvements using pivoting are summarized in Table 3.5. Pivoting was especially successful for the strongly off-diagonal dominant ZITNEY matrices (chemical process separation). Similarly, pivoting resulted in some improvements on the SHYY (Navier–Stokes equations) and MALLYA (light hydrocarbon recovery) matrices which are also extremely strong off-diagonal dominant. For a representative example, see Table 3.6 (chemical process separation). Here only the algorithms with pivoting worked at all.

Finally, we comment on our experiments with sample matrices from the SPARSKIT collection. Results are summarized in Table 3.7. Pivoting improved the solution

TABLE 3.3
Matrix CHEMWEST/WEST2021. AINVP and ILUTP with different drop tolerances τ and pivot thresholds α.

Method	Pivot thresh. α	Drop tol. τ	Decomposition Fill-in/time[sec]	$GMRES(30)$ Steps/time[sec]	QMR Steps/time[sec]
AINVP	0.1	10^{-3}	9.9, $1.0{\cdot}10^{0}$	30, $1.2{\cdot}10^{-1}$	33, $2.2{\cdot}10^{-1}$
	1.0	10^{-1}	1.5, $2.0{\cdot}10^{-1}$	121, $3.0{\cdot}10^{-1}$	81, $4.0{\cdot}10^{-1}$
		10^{-2}	2.8, $3.2{\cdot}10^{-1}$	31, $9.0{\cdot}10^{-2}$	37, $1.9{\cdot}10^{-1}$
ILUTP	0.1	10^{-5}	3.1, $2.0{\cdot}10^{-2}$	14, $1.0{\cdot}10^{-2}$	23, $6.0{\cdot}10^{-2}$
	1.0	10^{-5}	3.0, $2.0{\cdot}10^{-2}$	14, $2.0{\cdot}10^{-2}$	27, $6.0{\cdot}10^{-2}$

TABLE 3.4
Matrix SHERMAN/SHERMAN2. AINVP and ILUTP with different drop tolerances τ and pivot thresholds α.

Method	Pivot thresh. α	Drop tol. τ	Decomposition Fill-in/time[sec]	$GMRES(30)$ Steps/time[sec]	QMR Steps/time[sec]
AINV		10^{-4}	3.1, $5.6{\cdot}10^{-1}$	371, $9.6{\cdot}10^{-1}$	104, $4.9{\cdot}10^{-1}$
		10^{-5}	4.7, $9.8{\cdot}10^{-1}$	14, $5.0{\cdot}10^{-2}$	18, $1.2{\cdot}10^{-1}$
AINVP	0.1	10^{-2}	1.0, $4.3{\cdot}10^{-1}$	24, $4.0{\cdot}10^{-2}$	25, $9.0{\cdot}10^{-2}$
	1.0	10^{-1}	0.3, $2.3{\cdot}10^{-1}$	151, $2.1{\cdot}10^{-1}$	100, $2.6{\cdot}10^{-1}$
		10^{-2}	0.6, $3.4{\cdot}10^{-1}$	22, $3.0{\cdot}10^{-2}$	24, $7.0{\cdot}10^{-2}$
ILUT		10^{-5}	1.5, $2.0{\cdot}10^{-2}$	82, $8.0{\cdot}10^{-2}$	111, $2.2{\cdot}10^{-1}$
		10^{-6}	1.9, $3.0{\cdot}10^{-2}$	10, $1.0{\cdot}10^{-2}$	17, $3.0{\cdot}10^{-2}$
ILUTP	0.1	10^{-5}	2.0, $5.0{\cdot}10^{-2}$	89, $1.0{\cdot}10^{-1}$	76, $1.7{\cdot}10^{-1}$
		10^{-6}	2.6, $8.0{\cdot}10^{-2}$	8, $1.0{\cdot}10^{-2}$	9, $3.0{\cdot}10^{-2}$
	1.0	10^{-5}	2.3, $6.0{\cdot}10^{-2}$	30, $4.0{\cdot}10^{-2}$	59, $1.5{\cdot}10^{-1}$

TABLE 3.5
Davis collection. Changes in each matrix family when pivoting is used for drop tolerance τ and pivot threshold α.

Matrix family	$\tau = 0.1$		$\tau = 0.01$	
	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$
HAMM	∘	∘	∘	∘
MALLYA	∘	∘	∘	∘/+
PORTFOLIO	∘	∘	∘	∘
SIMON	∘/−	∘	∘/−	∘
SHYY	∘	∘	∘	+
WANG	∘/−	∘/−	∘/−	∘/−
ZITNEY	∘/+	∘/++	∘/++	++

significantly for the DRIVCAV (CFD, driven cavity problems) and FIDAP (fully coupled Navier–Stokes equations) matrices. For the TOKAMAK matrices (nuclear physics, plasmas) the algorithms without pivoting were superior. See Table 3.8 for examples.

In most cases the codes for ILUT/ILUTP are much faster than those for the approximate inverses. In fact the implementation of the approximate inverse algorithm with or without pivoting is much more technical and the codes used for the experi-

TABLE 3.6
Matrix ZITNEY/RDIST1. AINVP and ILUTP with different drop tolerances τ and pivot thresholds α.

Method	Pivot thresh. α	Drop tol. τ	Decomposition Fill-in/time[sec]	$GMRES(30)$ Steps/time[sec]	QMR Steps/time[sec]
AINVP	0.1	10^{-2}	21.8, $9.8{\cdot}10^{1}$	—	248, $2.6{\cdot}10^{1}$
		10^{-3}	34.5, $1.8{\cdot}10^{2}$	11, $8.1{\cdot}10^{-1}$	11, $1.7{\cdot}10^{0}$
	1.0	10^{-2}	7.7, $3.5{\cdot}10^{1}$	50, $1.5{\cdot}10^{0}$	46, $2.6{\cdot}10^{0}$
ILUTP	0.1	10^{-2}	3.2, $5.3{\cdot}10^{-1}$	60, $6.0{\cdot}10^{-1}$	55, $9.4{\cdot}10^{-1}$
	1.0	10^{-2}	2.9, $4.2{\cdot}10^{-1}$	23, $2.2{\cdot}10^{-1}$	30, $5.0{\cdot}10^{-1}$

TABLE 3.7
SPARSKIT collection. Changes in each matrix family when pivoting is used for drop tolerance τ and pivot threshold α.

Matrix family	$\tau = 0.1$		$\tau = 0.01$	
	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$
DRIVCAV	∘	∘/+	∘/+	∘/++
FIDAP	∘	∘/+	∘	∘/+
TOKAMAK	∘/−	∘/−	−	−

TABLE 3.8
Matrix SPARSKIT/FIDAP31. AINVP and ILUT(P) with different drop tolerances τ and pivot thresholds α.

Method	Pivot thresh. α	Drop tol. τ	Decomposition Fill-in/time[sec]	$GMRES(30)$ Steps/time[sec]	QMR Steps/time[sec]
AINVP	0.1	10^{-2}	9.9, $6.1{\cdot}10^{1}$	—	155, $1.1{\cdot}10^{1}$
		10^{-3}	15.8, $1.0{\cdot}10^{2}$	13, $6.1{\cdot}10^{-1}$	13, $1.2{\cdot}10^{0}$
	1.0	10^{-1}	0.6, $1.8{\cdot}10^{0}$	—	194, $4.6{\cdot}10^{0}$
		10^{-2}	3.8, $1.0{\cdot}10^{1}$	48, $1.1{\cdot}10^{0}$	37, $1.6{\cdot}10^{0}$
ILUT		10^{-2}	1.2, $1.0{\cdot}10^{-1}$	78, $4.8{\cdot}10^{-1}$	72, $8.0{\cdot}10^{-1}$
		10^{-3}	1.5, $1.3{\cdot}10^{-1}$	23, $1.6{\cdot}10^{-1}$	26, $3.1{\cdot}10^{-1}$
ILUTP	0.1	10^{-1}	1.8, $3.4{\cdot}10^{-1}$	—	421, $5.7{\cdot}10^{0}$
		10^{-2}	3.7, $1.2{\cdot}10^{0}$	22, $2.7{\cdot}10^{-1}$	24, $5.5{\cdot}10^{-1}$
	1.0	10^{-2}	5.8, $3.6{\cdot}10^{0}$	27, $4.5{\cdot}10^{-1}$	29, $9.3{\cdot}10^{-1}$

ments are research codes which have not been profiled and optimized yet. A much improved implementation is still possible; see, e.g., the numerical results in [5, 8].

After illustrating the benefits of using pivoting in the approximate inverse preconditioner with several examples, we now examine the combination of pivoting with an a priori permutation and scaling suggested in [12, 2]. At first glance the use of pivoting and especially the use of strict pivoting seems to be a complementary approach to gain more stability. However, it is clear that combining two different approaches in an appropriate way can be a good compromise. We illustrate this on some matrices which have been reordered and scaled using the method from [12, 2] together with relaxed pivoting ($\alpha = 0.1$). We compare these results with strict pivoting ($\alpha = 1$) and no a priori permutation and with only a priori permutation but no pivoting. We applied

TABLE 3.9
AINV(P): Comparison of pivoting and preprocessing.

Number of successful computed problems for drop tolerance τ						
	$\tau = 10^{-1}$			$\tau = 10^{-2}$		
Matrix family (# matrices)	Only pivoting $\alpha = 1.0$ GMRES / QMR	Only preproc. GMRES / QMR	Prepr.+ pivoting $\alpha = 0.1$ GMRES / QMR	Only pivoting $\alpha = 1.0$ GMRES / QMR	Only preproc. GMRES / QMR	Prepr.+ pivoting $\alpha = 0.1$ GMRES / QMR
Harwell–Boeing collection (20 test matrices)						
CHEMW.(11)	10 / 11	9 / 10	11 / 11	11 / 11	11 / 11	11 / 11
LNS(6)	4 / 4	4 / 6	4 / 4	5 / 6	6 / 6	6 / 6
NNC(3)	0 / 1	0 / 0	1 / 2	0 / 1	0 / 0	2 / 3
Davis collection (11 test matrices)						
MALLYA(6)	0 / 0	1 / 2	1 / 2	0 / 1	2 / 2	2 / 3
ZITNEY(6)	3 / 3	0 / 0	1 / 2	4 / 5	3 / 4	6 / 6
SPARSKIT collection (64 test matrices)						
DRIVC.(22)	5 / 13	4 / 8	7 / 12	15 / 17	10 / 10	14 / 14
FIDAP(37)	12 / 14	8 / 8	11 / 12	20 / 19	9 / 9	23 / 25
TOKAM.(5)	0 / 1	3 / 4	0 / 1	1 / 2	2 / 5	1 / 3

TABLE 3.10
*Matrix BP/BP*1200*. AINV(P) with different versions of pivoting and preprocessing.*

Version of AINV(P)	Drop tol. τ	Fill-in/ time[sec]	$GMRES(30)$ steps/time[sec]	QMR steps/time[sec]
Only pivoting ($\alpha = 1.0$)	10^{-2}	8.3, $4.2 \cdot 10^{-1}$	—	56, $1.5 \cdot 10^{-1}$
	10^{-3}	13.4, $6.9 \cdot 10^{-1}$	15, $3.0 \cdot 10^{-2}$	15, $5.0 \cdot 10^{-2}$
Only preprocessing	10^{-1}	5.1, $9.0 \cdot 10^{-2}$	—	281, $5.5 \cdot 10^{-1}$
	10^{-2}	7.3, $1.1 \cdot 10^{-1}$	20, $3.0 \cdot 10^{-2}$	40, $8.0 \cdot 10^{-2}$
Preprocessing + pivoting ($\alpha = 0.1$)	10^{-1}	4.8, $1.4 \cdot 10^{-1}$	49, $5.0 \cdot 10^{-2}$	37, $7.0 \cdot 10^{-2}$
	10^{-2}	7.5, $1.6 \cdot 10^{-1}$	9, $1.0 \cdot 10^{-2}$	8, $2.0 \cdot 10^{-2}$

these algorithms to some strongly indefinite problems as well as some ill-conditioned problems. For a summary of the results, see Table 3.9. For some problems from partial differential equations (LNS, NNC, DRIVCAV, FIDAP) the factors of either AINV version were quite dense. These systems from partial differential equations have dense inverses even if entries of small magnitude are skipped. Therefore plain approximate inverse techniques (without any additional techniques like multigrid) might not be suitable to solve these specific systems. The TOKAMAK matrices (nuclear physics) were again easier to solve without dynamic pivoting, just as was the case when no preprocessing is applied. Conversely, several of the FIDAP matrices were solvable only with dynamic pivoting, even if preprocessing was used. For some examples, see Tables 3.10 (linear programming), 3.11 (chemical engineering), 3.12 (chemical process separation), and 3.13 (light hydrocarbon recovery). These tables show that for several problems the a priori permutation in combination with the factored approximate inverse with dynamic pivoting results in significantly better results compared with those cases where only one technique, either pivoting or a priori permutation, is used.

TABLE 3.11
Matrix WEST/WEST2021. AINV(P) with different versions of pivoting and preprocessing.

Version of AINV(P)	Drop tol. τ	Fill-in/ time[sec]	$GMRES(30)$ steps/time[sec]	QMR steps/time[sec]
Only pivoting ($\alpha = 1.0$)	10^{-1}	1.5, $2.0 \cdot 10^{-1}$	121, $3.0 \cdot 10^{-1}$	81, $4.0 \cdot 10^{-1}$
	10^{-2}	2.8, $3.2 \cdot 10^{-1}$	31, $9.0 \cdot 10^{-2}$	37, $1.9 \cdot 10^{-1}$
Only preprocessing	10^{-2}	8.1, $1.7 \cdot 10^{-1}$	13, $3.0 \cdot 10^{-2}$	13, $7.0 \cdot 10^{-2}$
Preprocessing + pivoting ($\alpha = 0.1$)	10^{-1}	3.5, $1.7 \cdot 10^{-1}$	24, $4.0 \cdot 10^{-2}$	25, $9.0 \cdot 10^{-2}$
	10^{-2}	7.8, $2.5 \cdot 10^{-1}$	9, $2.0 \cdot 10^{-2}$	8, $4.0 \cdot 10^{-2}$

TABLE 3.12
Matrix ZITNEY/EXTR1. AINV(P) with different versions of pivoting and preprocessing.

Version of AINV(P)	Drop tol. τ	Fill-in/ time[sec]	$GMRES(30)$ steps/time[sec]	QMR steps/time[sec]
Only pivoting ($\alpha = 1.0$)	10^{-2}	19.3, $4.7 \cdot 10^{0}$	—	159, $2.9 \cdot 10^{0}$
	10^{-3}	31.3, $9.8 \cdot 10^{0}$	26, $3.5 \cdot 10^{-1}$	29, $7.4 \cdot 10^{-1}$
Only preprocessing	10^{-2}	19.0, $5.9 \cdot 10^{-1}$	—	183, $3.0 \cdot 10^{0}$
	10^{-3}	24.0, $7.2 \cdot 10^{-1}$	21, $1.9 \cdot 10^{-1}$	29, $4.9 \cdot 10^{-1}$
Preprocessing + pivoting ($\alpha = 0.1$)	10^{-1}	11.5, $5.5 \cdot 10^{-1}$	—	255, $3.9 \cdot 10^{0}$
	10^{-2}	16.0, $7.9 \cdot 10^{-1}$	12, $1.2 \cdot 10^{-1}$	12, $2.7 \cdot 10^{-1}$

TABLE 3.13
Matrix MALLYA/LHR07C. AINV(P) with different versions of pivoting and preprocessing.

Version of AINV(P)	Drop tol. τ	Fill-in/ time[sec]	$GMRES(30)$ steps/time[sec]	QMR steps/time[sec]
Only pivoting ($\alpha = 1.0$)	10^{-3}	9.3, $1.2 \cdot 10^{2}$	—	281, $2.9 \cdot 10^{1}$
	10^{-4}	16.5, $2.9 \cdot 10^{2}$	27, $2.0 \cdot 10^{0}$	29, $4.3 \cdot 10^{0}$
Only preprocessing	10^{-4}	14.4, $2.2 \cdot 10^{1}$	—	257, $3.2 \cdot 10^{1}$
	10^{-5}	17.9, $2.9 \cdot 10^{1}$	13, $1.0 \cdot 10^{0}$	13, $2.2 \cdot 10^{0}$
Preprocessing + pivoting ($\alpha = 0.1$)	10^{-3}	7.3, $3.3 \cdot 10^{1}$	21, $9.3 \cdot 10^{-1}$	22, $2.0 \cdot 10^{0}$

4. Conclusions. We have presented a version of a factored approximate inverse with enhanced stability properties. The algorithm is obtained by carrying over pivoting strategies from *LU* decomposition techniques to approximate inverse, exploiting a strong connection between ILU-type methods and factored approximate inverse-type methods. A test with a fairly large collection of test matrices established clearly the advantages of using pivoting. Pivoting in AINV increases robustness in the harder cases and is unlikely to hamper performance too much in the easier cases. Combining approximate inverse with pivoting, row scaling, and a technique of nonsymmetric permutation developed elsewhere [2, 12] shows excellent improvements in robustness of AINV and opens the possibility of developing reliable preconditioners for very poorly structured matrices.

Acknowledgments. We wish to thank Michele Benzi for providing us with some of the sample matrices that have been preprocessed using the technique from [2]. This helped us obtain some of the results at the end of section 3. We are also thankful to the reviewers for their valuable comments.

REFERENCES

[1] M. Benzi, J. K. Cullum, and M. Tůma, *Robust approximate inverse preconditioning for the conjugate gradient method*, SIAM J. Sci. Comput., 22 (2000), pp. 1318–1332.
[2] M. Benzi, J. C. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.
[3] M. Benzi and C. D. Meyer, *A direct projection method for sparse linear systems*, SIAM J. Sci. Comput., 16 (1995), pp. 1159–1176.
[4] M. Benzi, C. D. Meyer, and M. Tůma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.
[5] M. Benzi and M. Tůma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 968–994.
[6] M. Bollhöfer and Y. Saad, *ILUs and factorized approximate inverses are strongly related. Part* I: *Overview of results*, SIAM J. Matrix. Anal. Appl., submitted.
[7] R. Bridson and W.-P. Tang, *Ordering, anisotropy and factored sparse approximate inverses*, SIAM J. Sci. Comput., 21 (1999), pp. 867–882.
[8] R. Bridson and W.-P. Tang, *Refining an approximate inverse*, J. Comput. Appl. Math., 123 (2000), pp. 293–306.
[9] E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.
[10] T. Davis, *Sparse matrix collection*, Numer. Anal. Digest, October 16, 1994.
[11] I. Duff, R. Grimes, and J. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[12] 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.
[13] R. Freund, G. Golub, and N. Nachtigal, *Iterative solution of linear systems*, Acta Numer., 1992, pp. 1–44.
[14] J. A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.
[15] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
[16] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[17] I. E. Kaporin, *New convergence results and preconditioning strategies for the conjugate gradient method*, Numer. Linear Algebra Appl., 1 (1994), pp. 179–210.
[18] S. Kharchenko, L. Kolotilina, A. Nikishin, and A. Yeremin, *A reliable AINV-type preconditioning method for constructing sparse approximate inverse preconditioners in factored form*, Numer. Linear Algebra Appl., 8 (2001), pp. 165–179.
[19] L. Yu. Kolotilina and A. Yu. Yeremin, *Factorized sparse approximate inverse preconditionings.* I. *Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
[20] J. Meijerink and H. A. V. 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.
[21] N. Munksgaard, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradient method*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.
[22] Y. Saad, *ILUT: A dual threshold incomplete ILU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
[23] Y. Saad, *SPARSKIT and sparse examples*, Numer. Anal. Digest, June 26, 1994.
[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 706–727
© 2002 Society for Industrial and Applied Mathematics

LOW-RANK APPROXIMATIONS WITH SPARSE FACTORS I: BASIC ALGORITHMS AND ERROR ANALYSIS*

ZHENYUE ZHANG†, HONGYUAN ZHA‡, AND HORST SIMON§

Abstract. We consider the problem of computing low-rank approximations of matrices. The novel aspects of our approach are that we require the low-rank approximations to be written in a factorized form with sparse factors, and the degree of sparsity of the factors can be traded off for reduced reconstruction error by certain user-determined parameters. We give a detailed error analysis of our proposed algorithms and compare the computed sparse low-rank approximations with those obtained from singular value decomposition. We present numerical examples arising from some application areas to illustrate the efficiency and accuracy of our algorithms.

Key words. low-rank matrix approximation, singular value decomposition, sparse factorization, perturbation analysis

AMS subject classifications. 15A18, 15A23, 65F15, 65F50

PII. S0895479899359631

1. Introduction.

We consider the problem of computing low-rank approximations of a given matrix $A \in \mathcal{R}^{m\times n}$ which arises in many applications areas; see [5, 14, 17] for a few examples. The theory of singular value decomposition (SVD) provides the following characterization of the best low-rank approximations of A in terms of Frobenius norm $\|\cdot\|_F$ [5, Theorem 2.5.3].

THEOREM 1.1. *Let the singular value decomposition of $A \in \mathcal{R}^{m\times n}$ be $A = U\Sigma V^T$,*

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min(m,n)}), \quad \sigma_1 \geq \cdots \geq \sigma_{\min(m,n)},$$

and let U and V be orthogonal. Then for $1 \leq k \leq \min(m,n)$,

$$\sum_{i=k+1}^{\min(m,n)} \sigma_i^2 = \min\{\, \|A - B\|_F^2 \mid \operatorname{rank}(B) \leq k\}.$$

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.

*Received by the editors July 30, 1999; accepted for publication (in revised form) by D.P. O'Leary July 27, 2001; published electronically January 11, 2002. Part of this work was done while the first two authors were visiting National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory.

http://www.siam.org/journals/simax/23-3/35963.html

†Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou, 310027, People's Republic of China (zyzhang@math.zju.edu.cn), and National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, One Cyclotron Road, M/S: 50F, Berkeley, CA 94720. The work of this author was supported in part by NSFC (project 19771073), the Special Funds for Major State Basic Research Projects of China (project G19990328), and Foundation for University Key Teacher by the Ministry of Education, China. The work also was supported in part by NSF grant CCR-9619452 and 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.

‡Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 (zha@cse.psu.edu). The work of this author was supported in part by NSF grants CCR-9619452 and CCR-9901986.

§National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, One Cyclotron Road, M/S: 50B, Berkeley, CA 94720 (HDSimon@lbl.gov). The work of this author 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.

For any low-rank approximation B of A, we call $\|A - B\|_F$ the *reconstruction error* of using B as an approximation of A. By Theorem 1.1, $\text{best}_k(A)$ has the smallest reconstruction error in Frobenius norm among all the rank-k approximations of A. In certain applications, it is desirable to impose further constraints on the low-rank approximation B in addition to requiring that it be of low rank. Consider the case where, for example, the matrix A is sparse; it is generally not true that $\text{best}_k(A) = U_k \Sigma_k V_k^T$ or even that its associated factors U_k and V_k also will be sparse. Therefore, the storage requirement of $\text{best}_k(A)$ in the factorized form $\text{best}_k(A) = U_k \Sigma_k V_k^T$ can be even greater than that of the original matrix A. To overcome this difficulty, we seek to find low-rank approximations that simultaneously also possess some sparsity properties. One possibility will be to impose sparsity requirements directly on the low-rank approximation B itself, i.e., we require that B be sparse. However, this approach is less flexible, and it is very difficult to achieve a reasonable reconstruction error (compared with that obtained from $\text{best}_k(A)$, for example) using a sparse B. Inspired by the work reported in [7, 15], we consider the approach of writing B in a factorized form as $B = XDY^T$ and imposing sparsity requirements on the factors X and Y instead while keeping D in positive diagonal form. Therefore, even though X and Y are sparse, B may be rather dense, and this actually gives the flexibility to achieve smaller reconstruction errors. One by-product of using the factorized form is that the low-rank constraint on B is trivially satisfied once B is in the factored form, i.e., $\text{rank}(B) \leq k$ if X has k columns. Although the focus of this paper is on imposing sparsity constraints, we should also mention that other constraints on the low-rank approximations may also be desirable: in latent class models for two-way contingency tables [4], probabilistic latent semantic indexing [6], and nonnegative matrix factorization [8], for example, elements of columns X and Y represent conditional probabilities and therefore are required to be nonnegative. As another example, in the so-called structured total least squares problems, the low-rank approximations need to have certain structures such as Toeplitz or Hankel [12]. We also mention that there has been research on solving linear systems and linear least squares problems with sparse solution vectors [3, 10].

The rest of the paper is organized as follows: In section 2, we cast the problem of computing sparse low-rank approximations in the framework of an optimization problem. We then propose algorithms and heuristics for finding approximate optimal solutions of this optimization problem. In section 3, we give a detailed error analysis of the proposed algorithms and heuristics. Specifically, we prove that the reconstruction errors of the computed sparse low-rank approximations are within a constant factor of those that are obtained by SVD. In section 4, we discuss several computational variations of the basic algorithms proposed in section 2, and in section 5, we conduct several numerical experiments to illustrate the various numerical and efficiency issues of our proposed algorithms. We also compare the low-rank approximations computed by our algorithms with those obtained by SVD and the approaches developed in [15]. In section 6, we summarize our contributions and point out future research directions.

Notation. We use $\sigma_k(A)$ to denote the kth singular value of a matrix A in nonincreasing order. We also replace $\sigma_k(A)$ by σ_k when the matrix in question is unambiguous. By $\|\cdot\|$ we denote the 2-norm of vectors or matrices.

2. Sparse low-rank approximations. We first review some previous work on computing low-rank approximations with sparse factors. O'Leary and Peleg proposed a method for computing low-rank approximations for image processing [11]. In [7] Kolda and O'Leary called this *semidiscrete decomposition* (SDD), where they write

a low-rank approximation as $B_k = X_k D_k Y_k^T$ with $X_k \in \mathcal{R}^{m\times k}$, $Y_k \in \mathcal{R}^{k\times n}$, and D_k nonnegative diagonal. Furthermore, they require that the entries of X_k and Y_k belong to the three-element set $\{-1, 0, 1\}$. The restriction on the elements of X_k and Y_k usually demands a much larger $k \gg K$ in order for B_k to achieve a reconstruction error comparable to that of $\text{best}_K(A)$, and therefore the low-rank property of B_k may not hold. Despite this the storage requirement of B_k in the factored form is usually much lower than that of A, and this is certainly the major strength of SDD, as is demonstrated in the application in latent semantic indexing. In [15], Stewart proposes to construct low-rank approximations of a *sparse* matrix A by selecting a subset of its columns and rows, i.e., he writes a low-rank approximation as $B_k = A_c M A_r^T$, where A_c and A_r^T are certain k columns and k rows of A, respectively, and M is chosen to minimize the error $\|A - A_c M A_r^T\|_F$ once the left and right factors A_c and A_r are chosen. The matrices A_c and A_r are determined by variations of QR algorithms with a certain pivoting strategy. In general, the matrix M will be dense. Due to the denseness of M, the storage requirement of B_k can become rather high as k increases, and also the low-rank approximation will not be sparse if A itself is not sparse. Numerical experiments showed that Stewart's approach is especially effective when A itself is close to high rank-deficiency. The approach we now propose builds on the strength of the above two approaches: we seek an approximation that is of low-rank and at the same time we also want to have greater control of the sparsity properties of the low-rank approximation. To this end, we consider the following general minimization problem:[1]

$$\begin{aligned} &\min \|A - X_k D_k Y_k^T\|_F \\ &\text{subject to } D_k \text{ positive diagonal, } X_k \in \mathcal{R}^{m\times k}, \text{ and } Y_k \in \mathcal{R}^{n\times k} \text{ sparse.} \end{aligned} \tag{2.1}$$

The above optimization problem in its present form is not completely specified because the minimum depends on the sparsity constraints: the number of nonzero elements of the left and right factors and the positions of those nonzero elements which constitute what we call their *sparse patterns*. So *ideally* the goal is to make the reconstruction error $\|A - X_k D_k Y_k^T\|_F$ as small as possible and keep in mind the following questions:

- How do we determine good sparsity patterns for the left and right factors?
- How do we find the best approximation $B_k = X_k D_k Y_k^T$ with the chosen sparsity patterns for X_k and Y_k?

In this paper we will not discuss how to impose the sparsity constraints on the factors X_k and Y_k in general, but rather we will first start with a heuristic. In this section, we propose the framework of our sparse low-rank approximation (SLRA) approach based on the idea of deflation. As can be seen, the heuristic *dynamically and implicitly* imposes sparsity constraints on X_k and Y_k. See Figure 1.

Algorithm SLRA consists of a sequence of k deflation steps [13] which allows us to build a low-rank approximation one rank at a time. This general approach is also adopted in [7], but the actual deflation step there is very different from ours. After k steps, $A_k = A - X_k D_k Y_k$ with $X_k = [x_1, \ldots, x_k]$, $Y_k = [y_1, \ldots, y_k]$ and $D_k = \text{diag}(d_1, \ldots, d_k)$. It is worthwhile to point out that the integer k, the rank of B_k in general, can be determined by the stopping criterion $\|A - X_k D_k Y_k^T\|_F \le \texttt{tol}$ because the error $\|A - X_k D_k Y_k^T\|_F = \|A_k\|_F$ can be easily calculated by a recurrence relation derived in section 4.

[1] The diagonal elements of D can certainly be constructed to be positive, as we will do in what follows.

ALGORITHM SLRA (sparse low-rank approximation). Given a matrix $A \in \mathcal{R}^{m\times n}$ and an integer $k \leq \min\{m,n\}$, this algorithm produces a positive diagonal matrix D_k and sparse matrices X_k and Y_k. At the conclusion of the algorithm, $B_k \equiv X_k D_k Y_k^T$ gives a low-rank approximation of A with sparse factors.

1. [Initialize] Set $A_0 = A$.

2. For $i = 1, 2, \ldots, k$.

2.1. [Rank-one approximation] Find a *sparse* rank-one approximation $x_i d_i y_i^T$ to A_{i-1} with sparse unit vectors x_i and y_i.

2.2. Set $A_i = A_{i-1} - x_i d_i y_i^T$.

FIG. 1. *Algorithm SLRA.*

The key step of Algorithm SLRA is Step 2.1, i.e., computing *sparse* rank-one approximations. By Theorem 1.1 the best rank-one approximation to A is given by $u\sigma v^T$ with $\{u,\sigma,v\}$ the largest singular triplet of A. The triplet $\{u,\sigma,v\}$ can also be used to produce a good sparse rank-one approximation. The basic idea is to sparsify u and v to get sparse vectors x and y and choose a scalar d such that

$$\|A - xdy^T\|_F^2 = \min_s \|A - xsy^T\|_F^2 = \|A\|_F^2 - d^2. \tag{2.2}$$

Since the left and right singular vectors u and v will undergo this sparsification process, it is not necessary to compute them to high accuracy (see the remark after Theorem 3.2). The details of Step 2.1 of SLRA are listed in Figure 2.

3. Error analysis. In this section we will compare the low-rank approximations computed by Algorithm SLRA with those obtained by SVD with respect to the reconstruction errors. One possible potential alternative is to make the comparison directly with the optimal solutions of (2.1) assuming we have made more specifications on the sparsity of X_k and Y_k. For example, we can impose constraints on the number of nonzeros of X_k and Y_k and leave the positions of those nonzeros open. This approach at the moment is rather difficult to pursue because we still do not have a good understanding of the structures of the optimal solutions (2.1). Fortunately, $\text{best}_k(A)$ obtained from SVD gives the optimal solutions for (2.1) when there are no sparsity constraints on X_k and Y_k, and the heuristic of Algorithm SLRA takes advantage of this connection. Therefore we choose to compare the low-rank approximation $B_k = U_k D_k V_k^T$ with $\text{best}_k(A)$ computed by SVD. To proceed, we first consider the rank-one case, assuming we have computed the largest singular triplet *exactly*. Throughout the rest of the paper, we assume that $A \in \mathcal{R}^{m\times n}$.

THEOREM 3.1. *Let $\{u,\sigma,v\}$ be the largest singular triplet of A. Use the same notation as in Step 2.1 of Algorithm SLRA, and assume that $\|u_-\|^2 + \|v_-\|^2 \leq 2\epsilon^2$ with $\epsilon \leq 1/\sqrt{3}$. Then*

$$\|A - xdy^T\|_F \leq \sqrt{1+\alpha\tau}\,\|A - u\sigma v^T\|_F, \tag{3.1}$$

where

$$\alpha = \frac{\sigma_1^2}{\sum_{j=2}^n \sigma_j^2}, \quad \tau = 4\epsilon^2\left(1 - \frac{\epsilon^4}{(1-\epsilon^2)^2}\right) < 4\epsilon^2.$$

STEP 2.1 OF ALGORITHM SLRA. Given a matrix A, this algorithm produces a rank-one matrix xdy^T with sparse vectors x and y.

1. Compute (approximations of) the largest left and right singular vectors u and v of A.

2. Sparsify u and v to get sparse vectors x and y with $\|x\| = \|y\| = 1$.

2.1. [Sort] Sort the entries of u and v in two sections:

$$P_1 u = \begin{bmatrix} u_+ \\ u_- \end{bmatrix}, \quad P_2 v = \begin{bmatrix} v_+ \\ v_- \end{bmatrix},$$

where P_1 and P_2 are the permutation matrices resulted from the sorting process.

2.2. [Sparsify] Discard the second sections u_- and v_- to get sparse vectors x and y:

$$x \leftarrow P_1^T \begin{bmatrix} u_+ \\ 0 \end{bmatrix} / \|u_+\|, \quad y \leftarrow P_2^T \begin{bmatrix} v_+ \\ 0 \end{bmatrix} / \|v_+\|.$$

3. Set $d \equiv x^T A y$ which minimizes

$$\{\|A - xsy^T\|_F \,|\, s \text{ scalar}\}.$$

FIG. 2. *Step* 2.1 *of SLRA.*

Proof. Notice that d is chosen such that $\|A - xdy^T\|_F^2 = \|A\|_F^2 - d^2$ as shown in (2.2); we need to derive a lower bound for $|d|$. To this end, partition

$$P_1 A P_2^T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

conformably with $P_1 u$ and $P_2 v$ (see Step 2.1 of Algorithm SLRA). It follows from the choice of d that

$$d = x^T A y = u_+^T A_{11} v_+ / (\|u_+\| \cdot \|v_+\|). \tag{3.2}$$

Recalling that $Au = \sigma v$ and $A^T v = \sigma u$, we obtain

$$u_+^T A_{11} v_+ + u_+^T A_{12} v_- = \sigma \|u_+\|^2, \quad u_-^T A_{21} v_+ + u_-^T A_{22} v_- = \sigma \|u_-\|^2,$$

and, similarly, we have

$$v_+^T A_{11}^T u_+ + v_+^T A_{21}^T u_- = \sigma \|v_+\|^2, \quad v_-^T A_{12}^T u_+ + v_-^T A_{22}^T u_- = \sigma \|v_-\|^2.$$

Subtracting the sum of the last two equations of the four equations above from the sum of the first two yields

$$u_+^T A_{11} v_+ = u_-^T A_{22} v_- + \sigma(1 - \|u_-\|^2 - \|v_-\|^2). \tag{3.3}$$

Then substituting (3.3) into (3.2) gives

$$
\begin{aligned}
|d| &= \frac{|u_-^T A_{22} v_- + \sigma(1 - \|u_-\|^2 - \|v_-\|^2)|}{\|u_+\| \cdot \|v_+\|} \\
&\geq \frac{\sigma(1 - \|u_-\|^2 - \|v_-\|^2) - \sigma\|u_-\| \cdot \|v_-\|}{\|u_+\| \cdot \|v_+\|} \\
&\geq \sigma \frac{1 - \frac{3}{2}(\|u_-\|^2 + \|v_-\|^2)}{1 - \frac{1}{2}(\|u_-\|^2 + \|v_-\|^2)} \\
&\geq \sigma \frac{1 - 3\epsilon^2}{1 - \epsilon^2} \\
&= \sigma \left(1 - \frac{2\epsilon^2}{1 - \epsilon^2}\right) \geq 0.
\end{aligned} \tag{3.4}
$$

Here we have used the fact that $\|A_{22}\| \leq \|A\| = \sigma$. It follows from $\|A - u\sigma v^T\|_F^2 = \|A\|_F^2 - \sigma^2$ that

$$
\|A - xdy^T\|_F^2 \leq \|A\|_F^2 - \sigma_1^2 \left(1 - \frac{2\epsilon^2}{1 - \epsilon^2}\right)^2 = (1 + \alpha\tau)\|A - u\sigma v^T\|_F^2,
$$

where

$$
\tau = 1 - \left(1 - \frac{2\epsilon^2}{1 - \epsilon^2}\right)^2 = 4\epsilon^2 \left(1 - \frac{\epsilon^4}{(1 - \epsilon^2)^2}\right),
$$

completing the proof. ∎

In practice, the exact largest singular triplet is not available, and as we mentioned before it may not even be desirable to have it computed to high accuracy since we will sparsify u and v by discarding some of their nonzero elements anyway during the sparsification process. Hence, we need to consider the case when we only have approximations of the left and right singular vectors.

THEOREM 3.2. *Let $\{u, v\}$ be the approximate largest left and right singular vectors of A and $\sigma = \sigma_1(A)$. Use the notation of Step* 2.1 *of Algorithm SLRA and that of Theorem* 3.1 *and assume that $\|u_-\|^2 + \|v_-\|^2 \leq 2\epsilon^2$. Then*

$$
\|A - xdy^T\|_F \leq \sqrt{1 + \alpha(\tau + \delta)}\|A - u\sigma v^T\|_F, \tag{3.5}
$$

where τ is the same as that defined in Theorem 3.1 *and*

$$
\delta = \frac{2 - 6\epsilon^2 - \eta}{1 - 2\epsilon^2}\eta, \quad \eta = \frac{\|Av - \sigma u\| + \|A^T u - \sigma v\|}{2\sigma}.
$$

Proof. Define $r_1 = P_1(Av - \sigma u)$ and $r_2 = P_2^T(A^T u - \sigma v)$. Similarly as in the proof of (3.3), we have

$$
u_+^T A_{11} v_+ = u_-^T A_{22} v_- + \sigma(1 - \|u_-\|^2 - \|v_-\|^2) + r,
$$

where $r = \big([u_+^T, u_-^T]r_1 + [v_+^T, v_-^T]r_2\big)/2$ with norm $\|r\| \leq \eta\sigma$. By (3.4) and the inequality $\|u_+\|\|u_+\| \geq \sqrt{1 - 2\epsilon^2}$ we obtain

$$
\begin{aligned}
|d| &\geq \sigma \left(1 - \frac{2\epsilon^2}{1 - \epsilon^2}\right) - \frac{\|r\|}{\|u_+\|\|u_+\|} \\
&\geq \sigma \left(1 - \frac{2\epsilon^2}{1 - \epsilon^2} - \frac{\eta}{\sqrt{1 - 2\epsilon^2}}\right).
\end{aligned}
$$

The result (3.5) follows immediately from (2.2) and the inequality

$$1-\left(1-\frac{2\epsilon^2}{1-\epsilon^2}-\frac{\eta}{\sqrt{1-2\epsilon^2}}\right)^2=\tau+\left(\frac{2-6\epsilon^2}{1-\epsilon^2}-\frac{\eta}{\sqrt{1-2\epsilon^2}}\right)\frac{\eta}{\sqrt{1-2\epsilon^2}}$$
$$\leq \tau+\frac{\eta(2-6\epsilon^2-\eta)}{1-2\epsilon^2}=\tau+\delta,$$

completing the proof. □

Remark. We notice that η defined in Theorem 3.1 measures the accuracy of the approximate left and right singular vectors in a certain relative sense. The results in Theorem 3.1 say that $\tau = O(\epsilon^2)$. Consequently, if ϵ is fixed, there is no point to compute u and v to higher accuracy than $O(\epsilon^2)$. On the other hand, given approximate u and v and the corresponding η, we should choose ϵ to match their accuracy, i.e., $\epsilon = O(\sqrt{\eta})$.

Now we proceed to estimate the reconstruction error of $\|A - X_k D_k Y_k^T\|$ for the general case with $k > 1$. The basic idea is to estimate $\{\sigma_j^2(A_k)\}$ in terms of $\{\sigma_j^2(A)\}$ (recall that $A_k = A_{k-1} - x_k d_k y_k^T$). The key of our proof is to derive a *tight* upper bound on $\sum_{j=1}^{k}\sigma_j^2(A - xdy^T)$ in terms of $\sum_{j=2}^{k+1}\sigma_j^2(A)$. Then we will apply the bounds to A_{k-1} and $x_k d_k y_k^T$ step by step to obtain an upper bound of $\|A_k\| = \|A - X_k D_k Y_k^T\|$ in terms of $\{\sigma_j^2(A_0)\}$ with $A_0 = A$. With the assumptions that the left and right singular vectors are only approximate, the proof becomes rather unwieldy, and the bounds obtained are less transparent. Therefore, in the following we will assume that the left and right singular vectors u and v are computed exactly for each rank-one SVD approximation in the deflation process.

Notice that if $\{x, d, y\}$ is the *exact* largest singular triplet of A, $\sigma_i(A - xdy^T) = \sigma_{i+1}(A)$ for $i = 1, \ldots, \min\{m, n\} - 1$, and $\sigma_i(A - xdy^T) = 0$ for $i \geq \min\{m, n\}$, i.e., the second largest singular value of A becomes the largest singular value of $A - xdy^T$, the third largest singular value of A becomes the second largest singular value of $A - xdy^T$, and so on. It is easy to see that for any distinct indexes $i_1, \ldots, i_k$,

$$\sum_{j=1}^{k}\sigma_{i_j}^2(A - xdy^T) = \sum_{j=1}^{k}\sigma_{i_j+1}^2(A).$$

Therefore it is reasonable to expect an $O(\epsilon)$ estimation:

$$\sum_{j=1}^{k}\sigma_{i_j}^2(A - xdy^T) = \sum_{j=1}^{k}\sigma_{i_j+1}^2(A) + O(\epsilon) \tag{3.6}$$

when the triplet (x, d, y) is an $O(\epsilon)$ approximation of (u, σ, v). We now want to make (3.6) more precise and prove it rigorously. To this end, we first present several technical lemmas.

LEMMA 3.3. *Denote* $\hat{d} = u_+^T A_{11} v_+ / (\|u_+\| \cdot \|v_+\|)^2$ *and* $\sigma = \sigma_1(A)$. *If* $\|u_-\|^2 + \|v_-\|^2 \leq 2\epsilon^2$, *assuming* $\epsilon^2 \leq 1/\sqrt{5}$, *we then have*

$$\left|\frac{\sigma - \hat{d}}{\sigma}\right| \leq c_1\epsilon^2, \quad c_1 = \frac{1+\epsilon^2}{(1-\epsilon)^2}. \tag{3.7}$$

Proof. By (3.2) and (3.3) we have

$$\hat{d} = \sigma + \frac{u_-^T A_{22} v_- - \sigma(\|u_-\|^2\|v_-\|^2)}{\|u_+\|^2\|v_+\|^2}.$$

Hence

$$|\hat{d} - \sigma| \le \sigma \frac{\|u_-\|\|v_-\| + \|u_-\|^2\|v_-\|^2}{\|u_+\|^2\|v_+\|^2}. \tag{3.8}$$

Writing $\|u_-\| = \sqrt{2}a\cos(\theta)$, $\|v_-\| = \sqrt{2}a\sin(\theta)$ for certain $\theta \in [0, \pi/2]$ and $0 < a \le \epsilon$, and, furthermore, denoting $t = a^2\sin(2\theta) \in [0, a^2]$, we have

$$\frac{\|u_-\|\|v_-\| + \|u_-\|^2\|v_-\|^2}{\|u_+\|^2\|v_+\|^2} = \frac{a^2\sin(2\theta) + a^4\sin^2(2\theta)}{1 - 2a^2 + a^4\sin^2(2\theta)} = \frac{t + t^2}{1 - 2a^2 + t^2}.$$

It can be shown that the function $g(t) \equiv (t + t^2)/(1 - 2a^2 + t^2)$ is monotonically increasing in the interval $[0, a^2]$ if $a^2 \le \epsilon^2 \le 1/\sqrt{5}$. Therefore,

$$\begin{aligned} \frac{\|u_-\|\|v_-\| + \|u_-\|^2\|v_-\|^2}{\|u_+\|^2\|v_+\|^2} &\le g(a^2) = \frac{a^2 + a^4}{1 - 2a^2 + a^4} \\ &\le \frac{\epsilon^2 + \epsilon^4}{1 - 2\epsilon^2 + \epsilon^4} = c_1\epsilon^2. \end{aligned} \tag{3.9}$$

Equation (3.7) then follows from (3.8). ∎

LEMMA 3.4. *Let $\{u, \sigma = \sigma_1(A), v\}$ be the largest singular triplet of A. Denote $E = u\sigma v^T - xdy^T$. If $\|u_-\|^2 + \|v_-\|^2 \le 2\epsilon^2$, assuming $\epsilon^2 < 1/3$, then*

$$\|E\|_F \le \sigma_1(A)(\sqrt{2} + \epsilon^2)\epsilon \tag{3.10}$$

and

$$|\sigma_j(A - xdy^T) - \sigma_{j+1}(A)| \le \sigma_1(A)(\sqrt{2} + \epsilon^2)\epsilon. \tag{3.11}$$

Proof. Let $\hat{h} = (\sigma - \hat{d})/\sigma$ with $\hat{d}$ defined in Lemma 3.3. Then $\hat{d} = \sigma(1 - \hat{h})$ and

$$P_1 E P_2^T = \sigma \begin{bmatrix} \hat{h}\|v_+\|u_+ & \|v_-\|u_+ \\ \|v_+\|u_- & \|v_-\|u_- \end{bmatrix} \begin{bmatrix} v_+/\|v_+\| & 0 \\ 0 & v_-/\|v_-\| \end{bmatrix}^T.$$

Hence, by (3.8) we obtain that

$$\begin{aligned} \|E\|_F^2/\sigma^2 &= (\hat{h}^2\|u_+\|^2 + \|u_-\|^2)\|v_+\|^2 + \|v_-\|^2 \\ &= \hat{h}^2\|u_+\|^2\|v_+\|^2 + \|u_-\|^2 + \|v_-\|^2 - \|u_-\|^2\|v_-\|^2 \\ &\le \|u_-\|^2 + \|v_-\|^2 + \|u_-\|^2\|v_-\|^2\left(\frac{(1 + \|u_-\|\|v_-\|)^2}{\|u_+\|^2\|v_+\|^2} - 1\right) \\ &= \|u_-\|^2 + \|v_-\|^2 + \|u_-\|^2\|v_-\|^2\left(\frac{\|u_-\| + \|v_-\|}{\|u_+\|\|v_+\|}\right)^2 \\ &\le 2\epsilon^2 + \epsilon^4\frac{4\epsilon^2}{(1 - \epsilon^2)^2} \\ &\le \epsilon^2(\sqrt{2} + \epsilon^2)^2. \end{aligned}$$

Here we have used the inequality

$$\left(\frac{\|u_-\| + \|v_-\|}{\|u_+\|\|v_+\|}\right)^2 \le \left(\frac{2\epsilon}{1 - \epsilon^2}\right)^2,$$

which is valid for $\epsilon^2 \le 1/3$. This inequality can be proved using the same technique as when we prove (3.9). The standard perturbation bounds for singular values [5, section 8.6.1] now give

$$\begin{aligned}\sigma_j(A - xdy^T) &= \sigma_j(A - u\sigma v^T + E)\\ &\le \sigma_j(A - u\sigma v^T) + \|E\|\\ &= \sigma_{j+1}(A) + \|E\|\\ &\le \sigma_{j+1}(A) + \sigma_1(A)\big(\sqrt{2} + \epsilon^2\big)\epsilon,\end{aligned}$$

completing the proof. □

Remark. It can be shown that if $\|u_-\| \le \epsilon$ and $\|v_-\| \le \epsilon$, then

$$|\sigma_j(A - xdy^T) - \sigma_{j+1}(A)| \le \sigma_1(A)\left(1 + \frac{2\epsilon}{\sqrt{1-\epsilon^2}}\right)\epsilon.$$

Using the well-known Wielandt–Hofmann theorem [5, section 8.6.1] and Lemma 3.4, one can prove that

$$\left(\sum_{i=k}^{n} \sigma_j^2(A - xdy^T)\right)^{1/2} \le \left(\sum_{i=k+1}^{n} \sigma_j^2(A)\right)^{1/2} + \sigma_1(A)\big(\sqrt{2} + \epsilon^2\big)\epsilon.$$

Therefore it is not difficult to show that

$$\|A - X_k D_k Y_k^T\|_F \le (1 + c_k\epsilon)\|A - U_k\Sigma_k V_k^T\|_F, \tag{3.12}$$

with

$$c_k = \sqrt{2}\sum_{i=1}^{k} \sigma_i(A) \Big/ \left(\sum_{i=k+1}^{n} \sigma_j^2(A)\right)^{1/2} + O(\epsilon). \tag{3.13}$$

However, the coefficient c_k seems to give a less tight bound. To derive a much tighter bound for $\|A - X_k D_k Y_k^T\|_F$, we need the following key lemma.

LEMMA 3.5. *Use the notation of Step* 2.1 *of Algorithm SLRA and assume that* $\|u_-\|^2 + \|v_-\|^2 \le 2\epsilon^2$ *with* $\epsilon^2 < 1/3$. *Then for any distinct indexes* $i_1, \ldots, i_k$,

$$\sum_{j=1}^{k} \sigma_{i_j}^2(A - xdy^T) \le \sum_{j=1}^{k} \sigma_{i_j+1}^2(A) + \sigma_1(A)\sigma_2(A)\epsilon + c\sigma_1^2(A)\epsilon^2, \tag{3.14}$$

where $c = c_2$ *for* $k = 1$ *and* $c = 2c_2$ *for* $k > 1$, *and* $c_2 = (1 + c_1\epsilon^2)^2(3 + \sqrt{2}c_1\epsilon)$ *with* c_1 *defined in Lemma* 3.3.

Proof. Let the SVD of A be $A = U\Sigma V^T$ with $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$. To simplify the notation, denote $u = u_1$, $v = v_1$, $\sigma = \sigma_1$, and $\Sigma_2 = \operatorname{diag}(\sigma_2, \ldots, \sigma_n)$. Denote $B = U^T(A - xdy^T)V$. We also assume that $\|u_-\| \le \|v_-\|$, which implies $\|u_-\| \le \epsilon$. (Otherwise we can consider BB^T instead of B^TB in what follows.) The proof of this lemma consists of the following three parts.

(1) We first show that the matrix B^TB is a rank-3 modification of $\operatorname{diag}(0, \Sigma_2^2)$, i.e.,

$$B^TB = \operatorname{diag}(0, \Sigma_2^2) + F, \quad \operatorname{rank}(F) \le 3. \tag{3.15}$$

Thus it follows from [16, p. 202] that for distinct indexes $i_1, \ldots, i_k$,

$$\sum_{j=1}^{k} \lambda_{i_j}(B^T B) \leq \sum_{j=1}^{k} \lambda_{i_j}(\mathrm{diag}(0, \Sigma_2^2)) + \sum_{j=1}^{k} \lambda_j(F),$$

with the notation $\lambda_j(\cdot)$ denoting the jth *largest* eigenvalue of a symmetric matrix. To write the above in another way, we have

$$\sum_{j=1}^{k} \sigma_{i_j}^2(A - xdy^T) \leq \sum_{j=1}^{k} \sigma_{i_j+1}^2(A) + \sum_{j=1}^{k} \lambda_j(F). \tag{3.16}$$

To this end, partition

$$P_1 U = \left[\begin{bmatrix} u_+ \\ u_- \end{bmatrix}, U_2 \right], \quad P_2 V = \left[\begin{bmatrix} v_+ \\ v_- \end{bmatrix}, V_2 \right], \quad \Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_2 \end{bmatrix}.$$

(See Step 2.1 of SLRA for the definition of the permutation matrices P_1 and P_2.) It can be verified that

$$U^T x d y^T V = \hat{d}\Big(e_1 e_1^T - e_1 w_2^T - w_1 e_1^T + w_1 e_2^T \Big),$$

where

$$w_1 = (P_1 U)^T \begin{bmatrix} 0 \\ u_- \end{bmatrix} = \begin{bmatrix} \|u_-\|^2 \\ U_2^T \begin{bmatrix} 0 \\ u_- \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix},$$

$$w_2 = (P_2 V)^T \begin{bmatrix} 0 \\ v_- \end{bmatrix} = \begin{bmatrix} \|v_-\|^2 \\ V_2^T \begin{bmatrix} 0 \\ v_- \end{bmatrix} \end{bmatrix} \equiv \begin{bmatrix} w_{12} \\ w_{22} \end{bmatrix}.$$

Furthermore, we have

$$w_{11} = \|u_-\|^2 = \|w_1\|^2 \leq \epsilon^2, \quad w_{12} = \|v_-\|^2 = \|w_2\|^2 \leq 2\epsilon^2. \tag{3.17}$$

Therefore, we can write

$$\begin{aligned} B &\equiv U^T (A - x d y^T) V \\ &= \mathrm{diag}(0, \Sigma_2) + \hat{d}\Big(h e_1 e_1^T + e_1 w_2^T + w_1 e_1^T - w_1 e_2^T \Big) \\ &= \mathrm{diag}(0, \Sigma_2) + \hat{d} [e_1, w_1] \begin{bmatrix} h & 1 \\ 1 & -1 \end{bmatrix} [e_1, w_2]^T, \end{aligned}$$

i.e., B is a rank-2 modification of $\mathrm{diag}(0, \Sigma_2)$. Here

$$\hat{d} = d/(\|u_+\| \cdot \|v_+\|), \quad h = (\sigma - \hat{d})/\hat{d}.$$

To show that $B^T B$ is a rank-3 modification of $\mathrm{diag}(0, \Sigma_2^2)$, let

$$w_3 = \begin{bmatrix} 0 \\ \Sigma_2 w_{21} \end{bmatrix}, \quad \Delta_1 = \begin{bmatrix} h & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & w_{11} \\ w_{11} & w_{11} \end{bmatrix} \begin{bmatrix} h & 1 \\ 1 & -1 \end{bmatrix}.$$

Then it can be verified that

$$B^T B = \operatorname{diag}(0, \Sigma_2^2) + [e_1, w_2, w_3]\Delta[e_1, w_2, w_3]^T \equiv \operatorname{diag}(0, \Sigma_2^2) + F,$$

where

$$\Delta = \hat{d}\begin{bmatrix} \hat{d}\Delta_1 & \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ \begin{bmatrix} 1 & -1 \end{bmatrix} & 0 \end{bmatrix} = \hat{d}\begin{bmatrix} \hat{d}(h^2 + 2hw_{11} + w_{11}) & \hat{d}h(1 - w_{11}) & 1 \\ \hat{d}h(1 - w_{11}) & \hat{d}(1 - w_{11}) & -1 \\ 1 & -1 & 0 \end{bmatrix}.$$

Therefore, (3.15) holds.

(2) We now prove that the matrix F has a negative eigenvalue, which implies that the last term of (3.16) is a sum of at most two largest eigenvalues of F. First $\operatorname{rank}([e_1, w_2, w_3]) \geq 2$ since e_1 is orthogonal to w_3. Without loss of generality, we assume that $\operatorname{rank}([e_1, w_2, w_3]) = 3$. (The case when $\operatorname{rank}([e_1, w_2, w_3]) = 2$ is simpler and can be similarly handled.) Thus by Sylvester's law of inertia [5, Theorem 8.1.17], the number of positive eigenvalues of F is equal to the number of positive eigenvalues of Δ. Therefore it is enough to show that Δ has only two positive eigenvalues. Clearly, Δ has at least one positive eigenvalue since it has a positive diagonal element. It can be shown that the determinant of Δ is negative: $\det(\Delta) = -\hat{d}^2(1 + h)^2 < 0$. It implies that Δ has one and only negative eigenvalue because Δ is obviously not negative definite. Therefore, Δ has exactly two positive eigenvalues, and so does F. Hence we can write (3.16) as

$$\sum_{j=1}^{k} \sigma_{i_j}^2(A - xdy^T) \leq \sum_{j=1}^{k} \sigma_{i_j+1}^2(A) + \sum_{j=1}^{\min\{k,2\}} \lambda_j(F). \tag{3.18}$$

(3) We finally derive upper bounds for $\lambda_1(F)$ and $\lambda_2(F)$ which lead to the inequality (3.14). To this end, we write

$$\begin{aligned} F &= \hat{d}[e_1, w_3]\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}[e_1, w_3]^T \\ &\quad + [e_1, w_2, w_3]\begin{bmatrix} \hat{d}^2\Delta_2 & [0, -\hat{d}]^T \\ [0, -\hat{d}] & 0 \end{bmatrix}[e_1, w_2, w_3]^T \equiv H + \tilde{F}. \end{aligned}$$

It is easy to see that $\lambda(H) = \{\hat{d}\|w_3\|, 0, \ldots, 0, -\hat{d}\|w_3\|\}$. By (3.7) and the inequality $\|w_3\| \leq \sigma_2\epsilon$, we thus have

$$\lambda_1(F) \leq \hat{d}\|w_3\| + \|\tilde{F}\| \leq \sigma_1\sigma_2\epsilon(1 + c_1\epsilon^2) + \|\tilde{F}\|, \tag{3.19}$$

$$\lambda_2(F) \leq \|\tilde{F}\|. \tag{3.20}$$

To estimate $\|\tilde{F}\|$, we normalize w_2 and w_3 and let $\hat{w}_2 = w_2/\|w_2\|$ and $\hat{w}_3 = w_3/\|w_3\|$. It is easy to see that

$$\|[e_1, \hat{w}_2, \hat{w}_3]\| \leq \sqrt{2}, \quad \|\tilde{F}\| \leq 2\|\hat{F}\|$$

with $\hat{d}h = \sigma - \hat{d}$, and

$$\begin{aligned} \hat{F} &\equiv (\hat{f}_{ij})_{i,j=1}^3 \\ &= \begin{bmatrix} (\sigma - \hat{d})^2 + \hat{d}\big(2\sigma - \hat{d}\big)w_{11} & \hat{d}(\sigma - \hat{d})(1 - w_{11})\|w_2\| & 0 \\ \hat{d}(\sigma - \hat{d})(1 - w_{11})\|w_2\| & \hat{d}^2(1 - w_{11})\|w_2\|^2 & -\hat{d}\|w_2\| \cdot \|w_3\| \\ 0 & -\hat{d}\|w_2\| \cdot \|w_3\| & 0 \end{bmatrix}. \end{aligned}$$

By Lemma 5.2 of [17] and the fact that $\hat{f}_{13} = \hat{f}_{31} = \hat{f}_{33} = 0$, we have

$$\begin{aligned}\|\hat{F}\| &\le \max\left\{|\hat{f}_{11}|, \left\|\begin{bmatrix} \hat{f}_{22} & \hat{f}_{23} \\ \hat{f}_{32} & \hat{f}_{33} \end{bmatrix}\right\|\right\} + |\hat{f}_{12}| \\ &\le \max\left\{|\hat{f}_{11}|, |\hat{f}_{22}| + |\hat{f}_{23}|\right\} + |\hat{f}_{12}|.\end{aligned}$$

By (3.7) and (3.17), it is easy to see that $\|\hat{F}\| = O(\epsilon^2)$. Furthermore, it can be verified that

$$|\hat{f}_{11}| \le |\hat{f}_{22}| + |\hat{f}_{23}| \le 3\sigma^2(1 + c_1\epsilon^2)^2\epsilon^2, \quad |\hat{f}_{12}| \le \sqrt{2}c_1\sigma^2(1 + c_1\epsilon^2)\epsilon^3,$$

which leads to

$$\|\hat{F}\| \le (1 + c_1\epsilon^2)^2(3 + \sqrt{2}c_1\epsilon)\sigma^2\epsilon^2 \equiv c_2\sigma^2\epsilon^2$$

and

$$\lambda_1(F) \le \sigma_1\sigma_2\epsilon + c_2\sigma^2\epsilon^2, \quad \lambda_2(F) = c_2\sigma^2\epsilon^2.$$

Combining the above bounds with (3.18) yields the result (3.14). ∎

Now we are ready to prove our main theorem.

THEOREM 3.6. *Use the notation in Step* 2.1 *of Algorithm SLRA and assume in each iteration of Step* 2.1 *that* $\|u_-\|^2 + \|v_-\|^2 \le 2\epsilon^2$ *with* $\epsilon^2 < 1/3$. *Then*

$$\|A - U_k\Sigma_k V_k^T\|_F \le \|A - X_k D_k Y_k^T\|_F \le \sqrt{1 + b_k\epsilon}\,\|A - U_k\Sigma_k V_k^T\|_F,$$

where

$$b_k = \frac{\sum_{j=1}^{k}\sigma_j(A)\sigma_{j+1}(A)}{\sum_{j=k+1}^{n}\sigma_j^2(A)} + O(\epsilon).$$

Proof. Let $A_k = A - X_k D_k Y_k^T$ with $A_0 = A$, and

$$X_k = [x_1, \ldots, x_k], \quad D_k = \operatorname{diag}(d_1, \ldots, d_k), \quad Y_k = [y_1, \ldots, y_k].$$

Then $A_k = A_{k-1} - x_k d_k y_k^T$, where x_k and y_k are the sparsified version of the largest left and right singular vectors $u^{(k-1)}$ and $v^{(k-1)}$ of A_{k-1}, respectively. Specifically, we choose permutation matrices $P_1^{(k-1)}$ and $P_2^{(k-1)}$ such that

$$P_1^{(k-1)}u^{(k-1)} = \begin{bmatrix} u_+^{(k-1)} \\ u_-^{(k-1)} \end{bmatrix}, \quad P_2^{(k-1)}v^{(k-1)} = \begin{bmatrix} v_+^{(k-1)} \\ v_-^{(k-1)} \end{bmatrix}$$

with $\|u_-^{(k-1)}\|^2 + \|v_-^{(k-1)}\|^2 \le 2\epsilon^2$. Then

$$x_k = (P_1^{(k-1)})^T \begin{bmatrix} u_+^{(k-1)} \\ 0 \end{bmatrix} / \|u_+^{(k-1)}\|, \quad y_k = (P_2^{(k-1)})^T \begin{bmatrix} v_+^{(k-1)} \\ 0 \end{bmatrix} / \|v_+^{(k-1)}\|.$$

Since $A_k = A - X_k D_k Y_k^T$, applying Lemma 3.5 to $A_k = A_{k-1} - x_k d_k y_k^T$, we have

$$\begin{aligned}\|A - X_k D_k Y_k^T\|_F^2 &= \sum_{j=1}^{n}\sigma_j^2(A_k) \\ &\le \sum_{j=2}^{n}\sigma_j^2(A_{k-1}) + \sigma_1(A_{k-1})\sigma_2(A_{k-1})\epsilon + c\sigma_1^2(A_{k-1})\epsilon^2 \\ &\le \sum_{j=k+1}^{n}\sigma_j^2(A) + \sum_{j=0}^{k-1}\sigma_1(A_j)\sigma_2(A_j)\epsilon + c\sum_{j=0}^{k-1}\sigma_1^2(A_j)\epsilon^2.\end{aligned} \tag{3.21}$$

On the other hand, by Lemma 3.4 we have, with $c_3 = \sqrt{2} + \epsilon^2$,

$$\begin{aligned}\sigma_1(A_j) &\le \sigma_2(A_{j-1}) + c_3\sigma_1(A_{j-1})\epsilon \\ &\le \sigma_3(A_{j-2}) + c_3(\sigma_1(A_{j-2}) + \sigma_1(A_{j-1}))\epsilon \\ &\le \cdots \\ &\le \sigma_{j+1}(A) + c_3\sum_{i=0}^{j-1}\sigma_1(A_i)\epsilon.\end{aligned} \tag{3.22}$$

Let $s_j = \sum_{i=0}^{j-1}\sigma_1(A_i)$. Then by (3.22)

$$\begin{aligned}s_j &= \sigma_1(A_{j-1}) + s_{j-1} \le \sigma_j(A) + (1 + c_3\epsilon)s_{j-1} \\ &\le \sigma_j(A) + (1 + c_3\epsilon)(\sigma_{j-1}(A) + (1 + c_3\epsilon)s_{j-2}) \\ &\le \cdots \\ &\le \sum_{i=1}^{j}(1 + c_3\epsilon)^{j-i}\sigma_i(A).\end{aligned} \tag{3.23}$$

Substituting (3.23) into (3.22) gives

$$\sigma_1(A_j) \le \sigma_{j+1}(A) + c_3\sum_{i=1}^{j}(1 + c_3\epsilon)^{j-i}\sigma_i(A)\epsilon \equiv \sigma_{j+1}(A) + \phi_j\epsilon,$$

where $\phi_j = c_3\sum_{i=1}^{j}(1 + c_3\epsilon)^{j-i}\sigma_i(A)$. Similarly, we have

$$\sigma_2(A_j) \le \sigma_{j+2}(A) + \phi_j\epsilon.$$

Therefore,

$$\begin{aligned}&\sum_{j=0}^{k-1}\sigma_1(A_j)\sigma_2(A_j) \\ &\le \sum_{j=1}^{k}\left(\sigma_j(A)\sigma_{j+1}(A) + (\sigma_j(A) + \sigma_{j+1}(A) + \phi_{j-1}\epsilon)\phi_{j-1}\epsilon\right)\end{aligned} \tag{3.24}$$

and

$$\sum_{j=0}^{k-1}\sigma_1^2(A_j) \le \sum_{j=1}^{k}\left(\sigma_j^2(A) + 2\sigma_j(A)\phi_{j-1}\epsilon + \phi_{j-1}^2\epsilon^2\right). \tag{3.25}$$

Combining (3.21), (3.24), and (3.25) we obtain that

$$\begin{aligned}\|A - X_kD_kY_k^T\|_F &\le \sum_{j=k+1}^{n}\sigma_j^2(A) + \sum_{j=1}^{k}\sigma_j(A)\sigma_{j+1}(A)\epsilon + \tilde{b}_k\epsilon^2 \\ &= (1 + b_k\epsilon)\|A - U_k\Sigma_kV_k^T\|_F^2,\end{aligned}$$

where

$$\tilde{b}_k = \sum_{j=1}^{k}\left\{c\sigma_j^2(A) + ((1 + 2c\epsilon)\sigma_j(A) + \sigma_{j+1}(A) + (1 + c\epsilon)\phi_{j-1}\epsilon)\,\phi_{j-1}\right\},$$

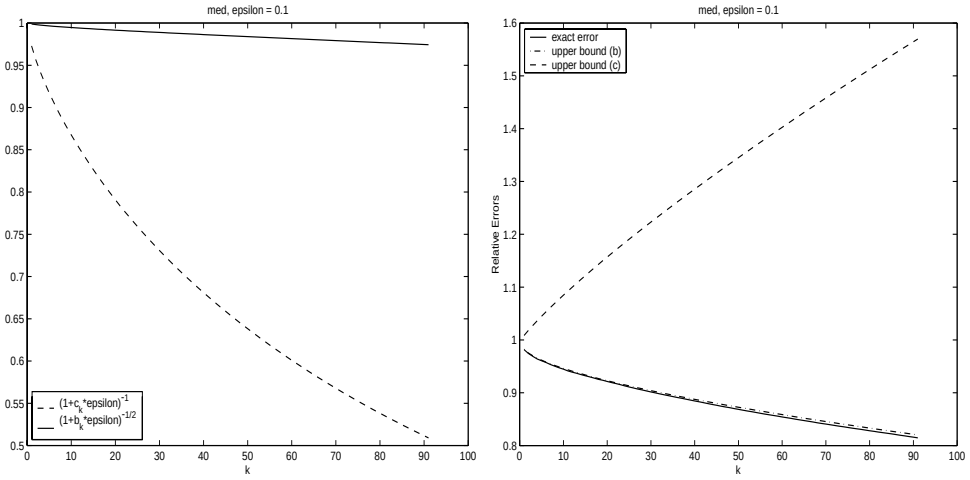


FIG. 3. $(1 + c_k * \epsilon)^{-1}$ *and* $(1 + b_k * \epsilon)^{-1/2}$ *(left) and the relative errors (right).*

completing the proof. □

The bound proved in the above theorem usually is much tighter than the bound in (3.12). In Figure 3, for various k, we plot the quantities $(1 + c_k\epsilon)^{-1}$ and $(1 + b_k\epsilon)^{-1/2}$ with the $O(\epsilon)$ terms omitted for the matrix `med` (cf. section 5) on the left and the relative error

$$\mathrm{err}_{\mathrm{best}}(k) = \frac{\|A - \mathrm{best}_k(A)\|_F}{\|A\|_F}$$

and the upper bounds

$$(1 + c_k\epsilon)\mathrm{err}_{\mathrm{best}}(k) \quad \text{and} \quad (1 + b_k\epsilon)^{1/2}\mathrm{err}_{\mathrm{best}}(k)$$

on the right.

4. Computational variations. In this section, we first discuss several computational variations of Algorithm SLRA; in particular we look at two approaches for sparsifying vectors in Step 2.1 of Algorithm SLRA. We first briefly discuss how to find approximations to the largest singular triplet of a matrix.

Computing largest singular triplets. As we mentioned in section 2, the largest singular triplet $\{u, \sigma, v\}$ does not need to be computed to high accuracy because a sparsification process that follows will introduce errors by discarding certain nonzero elements of u and v. There are several approaches for approximating the largest singular triplets, such as the power method and the Lanczos bidiagonalization process [5, 13]. Using the power method, we suggest performing several steps of power iteration, as follows:

$$\begin{aligned} v &\leftarrow (A^T A)^\alpha v_0, \\ v &\leftarrow v/\|v\|_2, \\ u &\leftarrow Av/\|Av\|_2, \end{aligned}$$

where v_0 is an initial guess, for example, $v_0 = (1, \ldots, 1)^T$, and α is a small integer, for example, $\alpha = 3$.

For Lanczos bidiagonalization, we can run several Lanczos iterations to generate a pair of orthogonal bases $\{u_1, \ldots, u_\beta\}$ and $\{v_1, \ldots, v_\beta\}$, and a lower bidiagonal matrix B_β satisfying

$$A[v_1, \ldots, v_\beta] = [u_1, \ldots, u_\beta]B_\beta + b_\beta u_{\beta+1},$$
$$A^T[u_1, \ldots, u_\beta] = [v_1, \ldots, v_\beta]B_\beta^T.$$

The largest singular vectors a and b of B_β will be used to obtain approximations u and v:

$$v = [v_1, \ldots, v_\beta]a, \quad u = [u_1, \ldots, u_\beta]b.$$

Sorting and sparsification. This corresponds to how to partition the computed approximate singular vectors u and v for the sparsification process to come later. By Theorems 3.1 and 3.2 the reconstruction error $\|A - xdy^T\|_F$ of the sparse rank-one approximation depends on the size of the discarded sections $\|u_-\|_2$ and $\|v_-\|_2$. Therefore it makes sense to sort vectors u and v in decreasing order by their absolute values so that the number of discarded elements is largest under the constraints $\|u_-\|_2 \le \epsilon$ and $\|v_-\|_2 \le \epsilon$, or $\|u_-\|_2^2 + \|v_-\|_2^2 \le 2\epsilon^2$. In particular, we find permutations P_1 and P_2 such that $\tilde{u} \equiv P_1 u = \left[\begin{smallmatrix} u_+ \\ u_- \end{smallmatrix}\right]$, $\tilde{v} \equiv P_2 v = \left[\begin{smallmatrix} v_+ \\ v_- \end{smallmatrix}\right]$ with

$$|\tilde{u}_1| \ge |\tilde{u}_2| \ge \cdots \ge |\tilde{u}_m|, \quad |\tilde{v}_1| \ge |\tilde{v}_2| \ge \cdots \ge |\tilde{v}_n|.$$

Let k_u and k_v be the lengths of sections u_+ and v_+, respectively. Thus $u_+ = \tilde{u}(1 : \mathtt{k_u})$ and $v_+ = \tilde{v}(1 : \mathtt{k_v})$. We then choose

$$x = P_1^T \begin{bmatrix} \tilde{u}(1 : \mathtt{k_u}) \\ 0 \end{bmatrix} / \|\tilde{u}(1 : \mathtt{k_u})\|, \quad y = P_2^T \begin{bmatrix} \tilde{v}(1 : \mathtt{k_v}) \\ 0 \end{bmatrix} / \|\tilde{v}(1 : \mathtt{k_v})\|.$$

The integers k_u and k_v can be determined by the following two different schemes.

- *Separated scheme.* In this approach, we sort the elements of u and v *separately*, and k_u and k_v are defined by

$$k_u = \min\left\{ k \,\Big|\, \sum_{j=1}^{k} \tilde{u}_j^2 \ge 1 - \epsilon^2 \right\}, \quad k_v = \min\left\{ k \,\Big|\, \sum_{j=1}^{k} \tilde{v}_j^2 \ge 1 - \epsilon^2 \right\}$$

  for a given tolerance ϵ.

- *Mixed scheme.* Another approach is to set $w = [u^T, v^T]^T$ and find a permutation P such that $Pw = \tilde{w}$, $|\tilde{w}_1| \ge |\tilde{w}_2| \ge \cdots \ge |\tilde{w}_{m+n}|$. We determine k_w such that

$$k_w = \min\left\{ k \ge k_0 \,\Big|\, \sum_{j=1}^{k} \tilde{w}_j^2 \ge 2\epsilon^2 \right\},$$

  where k_0 is the smallest integer such that the section $w(1 : \mathtt{k_0})$ contains both u-components and v-components. Obviously, the order of the u-components of vector $\tilde{w}$ implies the permutation P_1, as does the order of the v-components for P_2. Therefore the main section $\tilde{w}(1 : \mathtt{k_w})$ also determines $\tilde{u}(1 : \mathtt{k_u})$ and $\tilde{v}(1 : \mathtt{k_v})$, where k_u and k_v are, respectively, the number of u-components and v-components of $\tilde{w}(1 : \mathtt{k_w})$.

Remark. In general, our experiments show that the mixed scheme performs better than the separated scheme.

Choice of tolerance ϵ. At each iteration step of Algorithm SLRA, the tolerance ϵ can be a predetermined constant or be chosen dynamically during the iteration process. We will use, for variable tolerance, at the kth iteration

$$\epsilon_k = \frac{\|A_{k-1}\|_F}{\|A\|_F}\epsilon,$$

which depends on the approximation computed by previous iterations.

Choice of k. Notice that the norm of error matrix A_k at step k can be written as

$$\|A_k\|_F^2 = \|A - X_k D_k Y_k\|_F^2 = \|A\|_F^2 - \sum_{j=1}^{k} d_j^2.$$

In fact, we have

$$\|A_k\|_F^2 = \|A_{k-1}\|_F^2 - d_k^2.$$

It is quite convenient to use this recurrence as a stopping criterion for Algorithm SLRA:

$$\|A_k\|_F \leq \texttt{tol}$$

for the given user-specified tolerance `tol`.

Self-correcting mechanism. This is certainly an area that deserves further research, and in the following we can only touch the tip of the iceberg. When we use a rank-one matrix $u\sigma v^T$ that is constructed from the exact largest singular triplet $\{u, \sigma, v\}$ of A, the difference $A - u\sigma v^T$ will not have any components in the two one-dimensional subspaces spanned by u and v, respectively. Notice that $\|A - u\sigma v^T\|_F^2 = \|A\|_F^2 - \sigma^2$, and the amount of reduction in the Frobenius norm is the largest possible by a rank-one modification. Now when we use an inaccurate rank-one approximation xdy^T, in general, it is true that $\hat{A} \equiv A - xdy^T$ will have some components left in the directions of u and v. Also $\|\hat{A}\|_F^2 = \|A\|_F^2 - d^2$, and the reduction in Frobenius norm will be smaller. The question now is the following: if we compute the rank-one approximation $\hat{x}\hat{d}\hat{y}^T$ for $\hat{A}$, will $\hat{x}\hat{d}\hat{y}^T$ pick up some of the components in u and v that are left by the previous rank-one approximation xdy^T? The answer seems to be yes even though we do not have a formal proof. This indicates that Algorithm SLRA has a self-correcting mechanism: errors made in early deflation steps can be corrected by later deflation steps. We now give an example that illustrates this phenomenon. Table 1 lists the first 10 diagonals $\{d_j\}$ and the singular values $\{\sigma_j\}$ of matrix A, respectively. In this example, those steps j for which $d_j > \sigma_j$ show the self-correcting process at work.

A combinatorial optimization problem. Now we reexamine the optimization problem (2.1) for $k = 1$. We can impose the following constraints on the number of nonzeros of x and y: $\texttt{nnz(x)} = \texttt{n}_\texttt{x}$, $\texttt{nnz(y)} = \texttt{n}_\texttt{y}$, where $n_x \leq m$ and $n_y \leq n$ are fixed. Let $i_1, \ldots, i_{n_x}$ and $j_1, \ldots j_{n_y}$ be the indexes of the nonzero elements of x and y, respectively. Then it is easy to see that the optimization problem (2.1) is reduced to

$$\min_{\hat{x}\in\mathcal{R}^{n_x}, \hat{y}\in\mathcal{R}^{n_y}} \|A([i_1, \ldots, i_{n_x}], [j_1, \ldots, j_{n_y}]) - \hat{x}\hat{d}\hat{y}^T\|_F, \tag{4.1}$$

TABLE 1
Self-correction phenomenon.

j	d_j	σ_j
1	4.5595e+05	4.5808e+05
2	3.8998e+05	4.5762e+05
3	4.5482e+05	4.5761e+05
4	3.7309e+05	3.9093e+05
5	4.4721e+05	3.9050e+05
6	3.5648e+05	3.9049e+05
7	2.2148e+05	2.2090e+05
8	1.8609e+05	2.2046e+05
9	2.3341e+05	2.2044e+05
10	2.2075e+05	1.1472e+05

where $\tilde{A} \equiv A([i_1, \ldots, i_{n_x}], [j_1, \ldots j_{n_y}])$ is the submatrix of A consisting of the intersection of rows $i_1, \ldots, i_{n_x}$ and columns $j_1, \ldots j_{n_y}$. Therefore, by Theorem 1.1 we need to find the largest singular triplet of $\tilde{A}$. Hence, the optimization problem (2.1) for $k = 1$ is equivalent to the following problem:

> Find n_x rows and n_y columns of A such that the largest singular value of $\tilde{A}$ is maximized.

This is a *combinatorial* optimization problem, and we do not know any good, i.e., polynomial-time, solution method for it. Step 2.1 of Algorithm SLRA does seem to provide a heuristic for its solution. Now we give an example to illustrate this point.

Example. Consider the matrix

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

The goal is to compare the computed SLRA with the *optimal* solution of the combinatorial optimization problem (4.1) computed by exhaustive search.

We first compute the sparse approximation $X_k D_k Y_k^T$ for $k = 2$ using Algorithm SLRA with $\epsilon = 0.3$ and $\beta = 4$ for computing the approximate largest singular triplet using Lanczos bidiagonalization. The computed vectors x_i and y_i have the numbers of nonzeros listed below.

$$\mathtt{nnz}(\mathtt{x}_1) = 5, \quad \mathtt{nnz}(\mathtt{y}_1) = 4, \quad \mathtt{nnz}(\mathtt{x}_2) = 3, \quad \mathtt{nnz}(\mathtt{y}_2) = 3.$$

Next we compute the best rank-one approximation $u_1 s_1 v_1^T$ to A with the constraints $\mathtt{nnz}(\mathtt{u}_1) = \mathtt{nnz}(\mathtt{x}_1)$ and $\mathtt{nnz}(\mathtt{v}_1) = \mathtt{nnz}(\mathtt{y}_1)$, and then the best rank-one approximation $u_2 s_2 v_2^T$ to matrix $A - u_1 s_1 v_1^T$ with the constraints $\mathtt{nnz}(\mathtt{u}_2) = \mathtt{nnz}(\mathtt{x}_2)$ and $\mathtt{nnz}(\mathtt{v}_2) = \mathtt{nnz}(\mathtt{y}_2)$. The above two steps for computing u_i and v_i are carried out using exhaustive search. Below we list the computed components of vectors x_i, y_i, u_i, and v_i. The two approximations give the same sparsity patterns, i.e., wherever x_i (or y_i) has a zero element, u_i (or v_i) also has a zero element in the same position, and vice versa.

However, notice that the values of nonzero elements are different but very close:

x_1	x_2	u_1	u_2
0.4058	0.3245	0.4111	0.3118
0.6146	0	0.6362	0
0.4058	0.3245	0.4111	0.3118
0.3583	0	0.3587	0
0.4058	−0.8885	0.3587	−0.8975
0	0	0	0

y_1	y_2	v_1	v_2
0.4508	0.5423	0.4905	0.5066
0	−0.6170	0	−0.6322
0.3075	0	0.3346	0
0.7734	0	0.7318	0
0.3226	−0.5702	0.3346	−0.5863

5. Numerical experiments. In this section, we present several numerical experiments to illustrate the effectiveness and efficiency of our approach for computing SLRAs. We will compare the performance of Algorithm SLRA with that of SVD and the approach proposed in [15] with respect to the following two issues:

(1) the reconstruction errors, and

(2) the computational complexity and storage required.

For the numerical experiments, we generate a collection of test matrices, which are listed below together with some relevant statistics: matrices 3, 4, 5, and 6 are term-document matrices from the SMART information retrieval system, and the rest of the matrices are selected from Matrix Market [2, 9]. We do not claim that the collection is comprehensive.

	Matrix	m	n	`nnz(A)`	Density(%)
1	ash958	958	292	19196	0.68
2	illc1033	1033	320	4732	1.43
3	cisi	5081	1469	66241	0.89
4	cacm	3510	3204	70339	0.63
5	med	5504	1033	51096	0.90
6	npl	4322	11429	224918	0.46
7	watson4	467	468	2836	1.30
8	orsirr2	886	886	5970	0.76
9	e20r1000	4241	4241	131430	0.73

Some explanation of the notation we used is in order here: m and n represent the row and column dimensions, respectively, of the given matrix. As used before, `nnz(A)` denotes the number of nonzero elements of A. Density is computed as `nnz(A)/(mn)`, the percentage of nonzero elements of a matrix.

In order to compare our algorithm with SPQR (sparse pivoted QR algorithm) in [15], for each matrix A, we first use SPQR to compute a rank-k approximation $B = A_c M A_r^T$. We use $k = 300$ if $\min(m, n) > 500$, otherwise we use $k = 100$. Then we let $\text{tol}(A) = \|A - B\|_F$, and we seek to find a low-rank approximation using SLRA such that

$$\|A - X_k D_k Y_k^T\|_F \leq \text{tol}(A).$$

Test 1. We compare the low-rank approximations computed by Algorithm SLRA with constant tolerance $\epsilon = 0.1$ and those computed by SVD. The dimension used for

Lanczos bidiagonalization for computing the approximate largest singular vectors is $\beta = 4$ (see the definition of β in the previous section). To illustrate the reconstruction error $\|A - X_k D_k Y_k^T\|_F$, we use the error ratio $\mathrm{er}(k)$ defined by

$$\mathrm{er}(k) = \frac{\|A - \mathrm{best}_k(A)\|_F}{\|A - X_k D_k Y_k^T\|_F}$$

to measure the effectiveness of Algorithm SLRA. It is easy to see that $0 \le \mathrm{er}(k) \le 1$. The larger the error ratio is, the more effective SLRA is. Below we list the error ratios of SLRA with constant tolerance $\epsilon = 0.1$ using the *separated sorting scheme*. The rank k is chosen to be $5 \sim 20\%$ of the size $l = \min(m, n)$ of a given matrix A. We also computed the average error ratio defined as

$$\mathrm{Average} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{er}(i),$$

where k is the smallest integer satisfying $\|A - X_k D_k Y_k^T\|_F \le \epsilon$.

Matrix	$k = 5\%$	10%	15%	20%	Average
ash958	0.9946	0.9896	0.9876	0.9845	0.9908
illc1033	0.3622	0.9160	0.9226	0.8984	0.8595
cisi	0.9866	0.9771	0.9690	0.9612	0.9778
cacm	0.9774	0.9625	0.9427	0.9221	0.9596
med	0.9882	0.9790	0.9699	0.9617	0.9790
watson4	0.9784	0.9374	0.4833	0.3166	0.7809
orsirr2	0.9217	0.8942	0.9136	0.9206	0.9274

For these matrices, Theorem 3.6 gives tight bounds for the ratios. Figure 4 plots, with respect to k, the lower bounds $(1 + b_k\epsilon)^{-1/2}$ (dashed lines) given in Theorem 3.6 and the ratio quantities $\mathrm{er}(k)$ (solid line) computed by the separated sorting SLRA with $\epsilon = 0.1$ for all the nine matrices. These examples show that SLRA has very high error ratios for most of the test matrices, especially for the term-document matrices.

Test 2. In general, the *mixed sorting scheme* gives a smaller number of nonzero elements for the sparse factors X_k and Y_k, i.e., less storage required, than the *separated sorting scheme* if we use the same tolerance sequence while the rank k of the low-rank approximations computed by the different schemes are about the same. We computed the low-rank approximations using Algorithm SLRA with the same variable tolerance scheme for both the separated and mixed sorting schemes. Different starting tolerances `ε = 0.05:0.05:0.5` are used for each test matrix. In Figure 5 we plot the ranks (left) and the total number of nonzero elements of X_k and Y_k (right) computed by SLRA with separated (top) and mixed (bottom) sorting schemes. For each test matrix, the ranks computed by the two sorting schemes are about the same while the mixed sorting scheme gives a smaller number of nonzero elements; this is especially the case for the starting tolerances around $\epsilon = 0.15$.

Test 3. In this test we compare, respectively, the ranks of the low-rank approximations, the computation cost in flops, and storage required for SVD, SPQR, and SLRA using variable tolerance and the mixed sorting scheme. For SLRA, we use $\epsilon = 0.1$ as the starting tolerance and $\beta = 6$ iterations for Lanczos bidiagonalization. The low-rank approximations computed by the three approaches have the same reconstruction errors for each test matrix. In general, as we mentioned before, SVD produces dense factors even when A is sparse. Therefore the low-rank approximation computed by SVD requires at least $(m + n + 1)k$ storage for its associated factors.

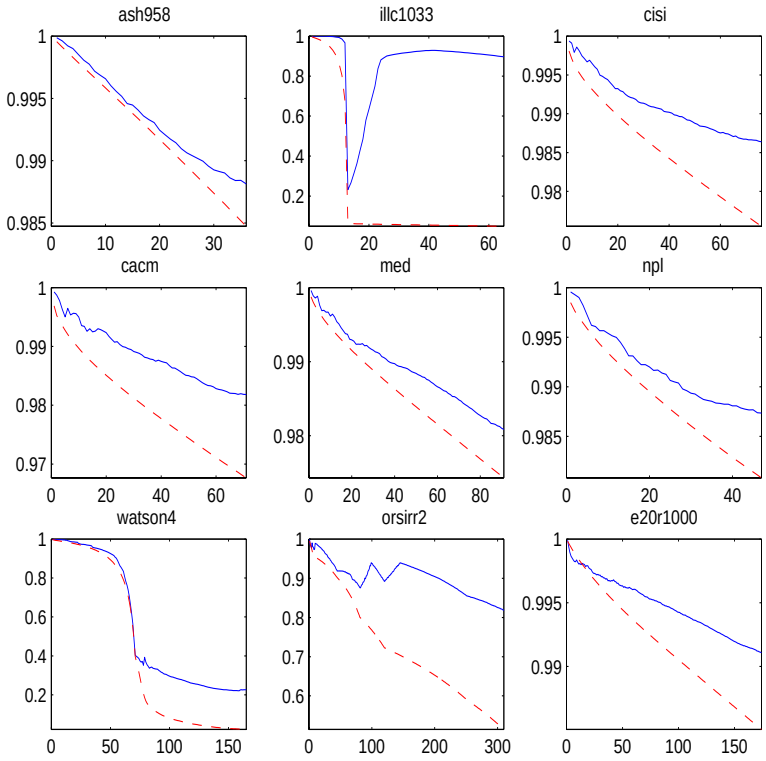


FIG. 4. *The computed* $\mathrm{er}(k)$ *(solid lines) and the lower bounds* $(1+b_k\epsilon)^{-1/2}$ *(dashed lines).*

For SPQR, the rank k of the low-rank approximation $B_k = A_c M A_r^T$ is usually quite large compared with the rank of the optimal low-rank approximation generated by SVD. Since the matrix M is generally dense, the storage required is dominated by M resulting in larger than k^2 storage requirement. In contrast, SLRA can produce low-rank approximations with small rank k and good degree of sparsity of the factors X_k and Y_k. (The number of nonzeros can be reduced by increasing the starting tolerance ϵ, which also increases the flops and ranks.) We list below the comparison for the term-document matrices in the test collection:

Matrix		Rank	Total nnz	Flops
cisi	TSVD	68	449412	6925863163
	SLRA	72	217401	523406959
	SPQR	300	129720	568382817
cacm	TSVD	63	426951	5390479001
	SLRA	67	216982	478032905
	SPQR	300	133784	463854304
med	TSVD	79	522664	9598485598
	SLRA	84	278456	658852943
	SPQR	300	120444	469695010
npl	TSVD	41	647472	6208537332
	SLRA	44	384118	616205165
	SPQR	300	227567	588513394

However, we should mention that the performance of SLRA is not as good as SPQR when the matrix A is close to a highly rank-deficient matrix. For example, let

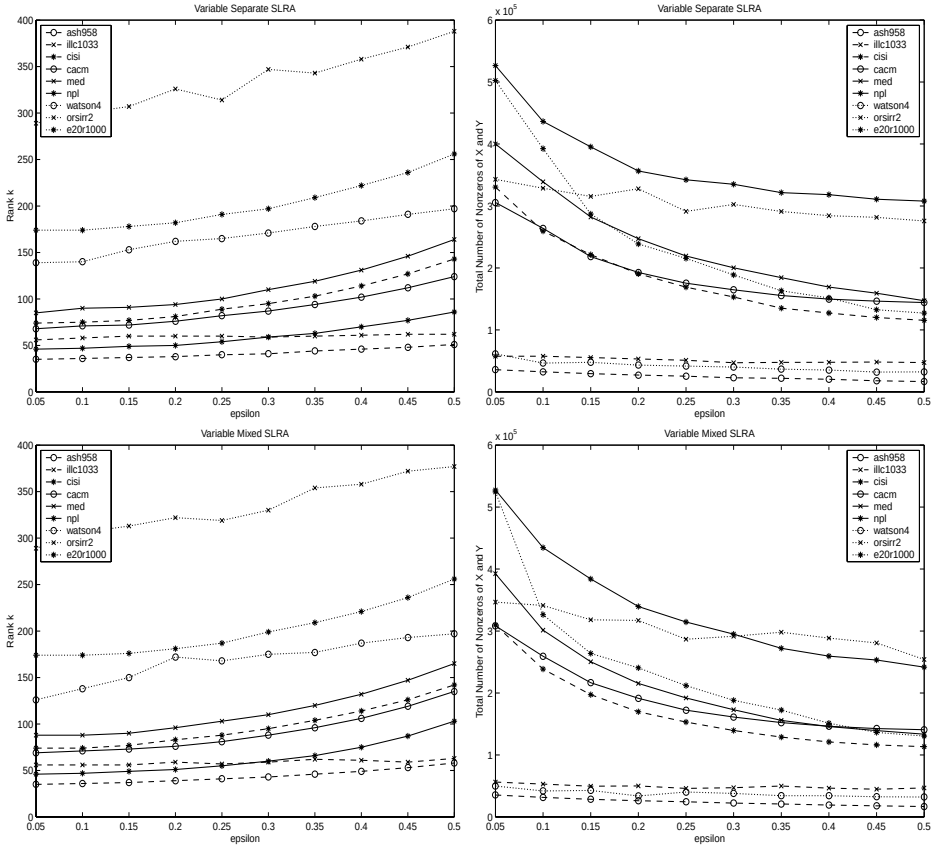


FIG. 5. *Plots for ranks (left) and numbers of nonzero elements of X_k and Y_k (right) vs. starting epsilon for the variable tolerance, separated (top) and mixed (bottom) sorting approaches.*

A be the matrix `illc1033` in the test collection. We compute, using SPQR, a rank-100 approximation $B = A_c M A_r^T$. The storage required (the number of nonzeros) for the computed low-rank approximation is about 20% of that for the best approximation B^* computed by SVD that achieves the same reconstruction error. SLRA with $\epsilon = 0.1$ gives an approximation B_k that has the same reconstruction error as that of SPQR, and the storage required is 85% of that for B^*, though the rank of B_k is close to the optimal rank and much smaller than the rank of B. SPQR also requires less flops for computing the low-rank approximation. In general, SPQR is very effective for sparse matrices that are close to highly rank-deficient, and the rank of the low-rank approximation can be predetermined. However, using SPQR is not convenient if the user just imposes an upper bound on reconstruction error.

6. Concluding remarks. We have presented algorithms for computing matrix low-rank approximations with sparse factors. We also gave a detailed error analysis comparing the reconstruction errors for the low-rank approximations computed by SVD and the low-rank approximations computed by our sparse low-rank algorithms. Our algorithms are flexible in the sense that users can balance the tradeoff of high sparsity levels of the computed low-rank factors and the reduced reconstruction error. Several issues deserve further investigation: (1) We need to develop better ways for

computing sparse rank-one approximations. As we mentioned, for example, if we fix the number of nonzero elements in x and y, say p and q, then $\min \|A - xdy^T\|_F$ is equivalent to the following *combinatorial optimization* problem: find p rows and q columns of A such that the largest singular value of their intersection is maximized. We are in the process of finding heuristics for solving this problem and investigating their relationships to the sorting approach of Algorithm SLRA. (2) Once a low-rank approximation A_k is computed, a certain refinement procedure needs to be developed to reduce its reconstruction error and/or the number of nonzeros of its sparse factors. (3) It will also be of great interest to consider reconstruction errors in norms other than $\|\cdot\|_F$.

Acknowledgments. The authors want to thank the anonymous referees for their comments and suggestions, which greatly improved the presentation of the paper.

REFERENCES

[1] M.W. Berry, S.T. Dumais, and G.W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Rev., 37 (1995), pp. 573–595.
[2] Cornell SMART System, ftp://ftp.cs.cornell.edu/pub/smart.
[3] C. Couvreur and Y. Bresler, *On the optimality of the backward greedy algorithm for the subset selection problem*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 797–808.
[4] M. Evans, Z. Gilula, and I. Guttman, *Latent class analysis of two-way contingency tables by Bayesian methods*, Biometrika, 76 (1989), pp. 557–563.
[5] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[6] T. Hofmann. *Probabilistic latent semantic indexing*, in Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99), ACM Press, New York, 1999, pp. 50–55.
[7] T. Kolda and D. O'Leary, *A semidiscrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Trans. Information Systems, 16 (1998), pp. 322–346.
[8] D. Lee and S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.
[9] Matrix Market, http://math.nist.gov/MatrixMarket/.
[10] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM J. Comput., 24 (1995), pp. 227–234.
[11] D.P. O'Leary and S. Peleg, *Digital image compression by outer product expansion*, IEEE Trans. Comm., 31 (1983), pp. 441–444.
[12] H. Park, L. Zhang, and J. Ben Rosen, *Low rank approximation of a Hankel matrix by structured total least norm*, BIT, 39 (1999), pp. 757–779.
[13] B.N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.
[14] H. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.
[15] G.W. Stewart, *Four algorithms for the efficient computation of truncated pivoted QR approximation to a sparse matrix*, Numer. Math., 83 (1999), pp. 313–323.
[16] G.W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, MA, 1990.
[17] 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. 23, No. 3, pp. 728–743
© 2002 Society for Industrial and Applied Mathematics

NEW BAND TOEPLITZ PRECONDITIONERS FOR ILL-CONDITIONED SYMMETRIC POSITIVE DEFINITE TOEPLITZ SYSTEMS*

D. NOUTSOS† AND P. VASSALOS†

Abstract. It is well known that preconditioned conjugate gradient (PCG) methods are widely used to solve ill-conditioned Toeplitz linear systems $T_n(f)x = b$. In this paper we present a new preconditioning technique for the solution of symmetric Toeplitz systems generated by nonnegative functions f with zeros of even order. More specifically, f is divided by the appropriate trigonometric polynomial g of the smallest degree, with zeros the zeros of f, to eliminate its zeros. Using rational approximation we approximate $\sqrt{f/g}$ by $\frac{p}{q}$, p, q trigonometric polynomials and consider $\frac{p^2 g}{q^2}$ as a very satisfactory approximation of f. We propose the matrix $M_n = B_n^{-1}(q)B_n(p^2 g)B_n^{-1}(q)$, where $B(\cdot)$ denotes the associated band Toeplitz matrix, as a preconditioner whence a good clustering of the spectrum of its preconditioned matrix is obtained. We also show that the proposed technique can be very flexible, a fact that is confirmed by various numerical experiments so that in many cases it constitutes a much more efficient strategy than the existing ones.

Key words. low rank correction, Toeplitz matrix, conjugate gradient, rational interpolation and approximation, preconditioner

AMS subject classifications. 65F10, 65F15

PII. S0895479800376314

1. Introduction.

In this paper we use and analyze band Toeplitz matrices as preconditioners for the solution of the $n \times n$ ill-conditioned symmetric and positive definite Toeplitz system

$$T_n(f)x = b \tag{1.1}$$

by the preconditioned conjugate gradient (PCG) method, where the matrix $T_n(f) \in \mathbb{R}^{n\times n}$ is produced by a real-valued, even, 2π-periodic function defined in the fundamental interval $[-\pi, \pi]$. Then, the (j, k) element of $T_n(f)$ is given by the Fourier coefficient of f, i.e.,

$$T_n(f)_{j,k} = T_{j-k} = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-\mathbf{i}(j-k)x}dx, \qquad 1 \le j, k \le n,$$

where $\mathbf{i}$ is the imaginary unit.

Toeplitz matrices arise very often in a wide variety of applications, as, e.g., in the numerical solution of differential equations using finite differences, in statistical problems (linear prediction), in Wiener–Hopf kernels, in Markov chains, in image and signal processing, etc. (see [13], [6], [25]). The generating function f plays a significant role in the location and distribution of the eigenvalues of Toeplitz matrix [13], [7] and in many cases is a priori known. As it is known for the spectrum of $T_n(f)$ there holds $\sigma(T_n(f)) \subseteq [\inf f, \sup f]$.

*Received by the editors August 4, 2000; accepted for publication (in revised form) by L. Eldén August 2, 2001; published electronically January 11, 2002.
http://www.siam.org/journals/simax/23-3/37631.html

†Department of Mathematics, University of Ioannina, GR-451 10, Ioannina, Greece (dnoutsos@cc.uoi.gr, pvassal@cc.uoi.gr). The research of the second author was supported by the Hellenic Foundation of Scholarships (HFS).

Superfast direct methods can solve system (1.1) in $O(n \log^2 n)$ operations, but their stability properties for ill-conditioned Toeplitz matrices are still unclear; see, for instance, [6].

Classical iterative methods such as Jacobi, Gauss–Seidel, and SOR are not effective since the associated spectral radius tends to 1 for large n. The method which is widely used for the solution of such systems is the PCG method. The factors that affect the convergence features of this method are the magnitude of the condition number $\kappa_2(T_n(f))$ and the distribution of the eigenvalues. So a good preconditioner must cluster the eigenvalues of the preconditioned system as much as possible and make the eigenvalues that might lie outside the cluster be bounded by nonzero constants independent of n.

If the generating function is continuous and positive, then problem (1.1) will not be ill-conditioned and the condition number cannot increase proportionally to n, although it can be very large. In this case system (1.1) can be handled by using a preconditioner belonging to some trigonometric matrix algebras (circulant, τ, Hartley; see [24], [2], [3], [23], [14]) or by band Toeplitz preconditioners with weakly increasing bandwidth defined by a polynomial operator $\mathcal{S}_n$, as was proposed in [22]. Theoretically, the latter class of preconditioners seems to perform better as $n \to \infty$ since the number of PCG iterations tends to 1, while in the former cases this number tends to a constant.

When f has a finite number of zeros, each one of finite multiplicity, then system (1.1) is ill-conditioned and the condition number $\kappa_2(T_n(f))$ increases proportionally to n^α where α is the largest number of the multiplicities of the zeros of f [7], [20]. To best handle this case it is necessary to know the number of multiplicities of each one. If this number is not even, then the most suitable technique for this situation [19] fails to make the condition number of the preconditioned matrix independent of its dimension n, and the problem is still open. On the other hand things dramatically change when the multiplicity of each zero is even.

In this case, it was Chan [7] who first proposed as a preconditioner for system (1.1) the Toeplitz band matrix $B_n(g)$ whose generating function g is a trigonometric polynomial that has the same zeros with the same multiplicities as those of f. Next, in [9], not only was g considered as having the zeros of f, but its degree was also increased so that it provided additional degrees of freedom to approximate f and to minimize the relative error $\|\frac{f-g}{g}\|_\infty$ over all trigonometric polynomials g of a fixed degree l. The generating function g is then computed by the Remez algorithm, which can be very expensive from a computational point of view, especially when f has a large number of zeros.

Recently, Serra [21] extended this method by proposing alternative techniques to minimize $\|\frac{f-g}{g}\|_\infty$. More specifically, he chose as g, $z_k g_{l-k}$, where z_k is the trigonometric polynomial of minimum degree k that has all the zeros of f with their multiplicities and g_{l-k} is the trigonometric polynomial of degree $l-k$ which is the best Chebyshev approximation of $\hat{f} = \frac{f}{z_k}$ from the space $\mathcal{P}_{l-k}$ of all trigonometric polynomials of degree at most $l-k$. In addition, in the same work [21], another way was proposed of constructing g_{l-k} by interpolating $\hat{f}$ at the $l-k+1$ zeros of the $(l-k+1)$st degree Chebyshev polynomial of the first kind.

We remark that it has been proved [12] that preconditioners belonging to the aforementioned matrix algebra, when they can be defined, produce weak clustering; i.e., the eigenvalues of the preconditioned matrix are such that for every $\epsilon > 0$ there exists a positive β so that, except for rare exceptions, $O(n^\beta)$ of the eigenvalues lie in

the interval $(0, \epsilon)$.

Further preconditioning techniques based on inverses of Toeplitz matrices can be found in [8], [11], [15].

In this paper we extend the previous methods in order to achieve a better clustering for the eigenvalues of the preconditioned matrix and propose a way of constructing a class of preconditioners based on rational approximation or on interpolation to the positive and continuous function $\sqrt{f/z_k}$, with z_k defined previously.

The outline of the present work is as follows. In section 2 we recall some useful issues about the rational approximation, while in section 3 we introduce the technique of constructing the new class of preconditioners based on rational approximation to $\sqrt{f/z_\rho}$ with z_k and analyze the convergence of the PCG method. In section 4 we study the flexibility and possible modification of our method, analyze its cost per iteration, and compare it with that of previous techniques. Finally, in section 5, results of illustrative numerical experiments are exhibited and concluding remarks are made.

2. Preliminaries. In what follows we assume that the generating function f is defined in $[-\pi, \pi]$, is 2π-periodic, continuous, nonnegative, and has zeros of even order.

We define by z_k a trigonometric polynomial of minimum degree k containing all the zeros of f with their multiplicities. Then we define $r_{lm} = \frac{p_l}{q_m}$ as the best rational approximation of $\hat{f} = \sqrt{f/z_k}$ in the uniform norm, i.e.,

$$\|\hat{f} - r_{lm}\|_\infty = \min_{r \in \mathcal{R}(l,m)} \|\hat{f} - r\|_\infty,$$

where $\mathcal{R}(l, m)$ denotes the set of rational functions r, with $p \in \mathcal{P}_l$, $q \in \mathcal{P}_m$, and r irreducible, that is, p and q have no zeros in common.

It is known that when f belongs to some special class of functions [16] then the order of magnitude of the maximum error of an approximation from the space $\mathcal{R}(l, m)$ is better than the corresponding error in the space $\mathcal{P}(l + m)$. In general, we hope that by taking advantage of the flexible nature of rational functions, this set will be a stronger tool than its competitor, the polynomial one. For example, it is obvious that polynomials are not suitable for approximating functions having sharp peaks near the center of their ranges and are slowly varying when $|x|$ increases. Such behavior can be obtained by continuous functions which are not differentiable at some points. However, it is easy to overcome this difficulty by using rational functions.

The next theorem establishes the fact that rational approximation of continuous functions in $[-\pi, \pi]$ is always possible and unique.

THEOREM 2.1. *Let f be in $C[-\pi, \pi]$. Then there exists $r^* \in \mathcal{R}(l, m)$ such that*

$$\|f - r^*\|_\infty < \|f - r\|_\infty$$

for all $r \in \mathcal{R}(l, m)$, $r \neq r^*$.

Proof. See [18, pp. 121, 125] for the proof. ∎

3. Construction of the preconditioner. Let f be a 2π-periodic nonnegative function belonging to $C[-\pi, \pi]$ with zeros $x_1, x_2, \ldots, x_s$ of multiplicities $2\mu_1, 2\mu_2, \ldots, 2\mu_s$, respectively, and $2\mu_1 + 2\mu_2 + \cdots + 2\mu_s = \rho$. First, we define

$$z_\rho = \prod_{i=1}^{s} (1 - \cos(x - x_i))^{\mu_i},$$

which is the trigonometric polynomial of minimum degree ρ having all the zeros of f. By dividing f by z_ρ, all its zeros are eliminated and the ratio $\frac{f}{z_\rho}$ becomes a real positive function.

Then, we define the function $\hat{f} = \sqrt{f/z_\rho}$ and approximate it with the rational trigonometric function $r_{lm} = \frac{p_l}{q_m}$, where l, m are the degrees of the numerator and the denominator, respectively. Since $\frac{p_l}{q_m}$ is the best rational approximation of $\sqrt{f/z_\rho}$ for certain l and m, we are led to the conclusion that $\frac{p_l^2}{q_m^2}$ may be a good approximation of $\frac{f}{z_\rho}$. This means that there exists a small $\epsilon > 0$ such that

$$\left\| \frac{f}{z_\rho} - \frac{p_l^2}{q_m^2} \right\|_\infty < \epsilon$$

or, equivalently, that there exists a small $\delta > 0$ such that

$$\left\| \frac{q_m^2}{z_\rho p_l^2} f - 1 \right\|_\infty < \delta.$$

The last inequality means that the values of $\frac{q_m^2}{z_\rho p_l^2} f$ are clustered in a small region near the constant number 1. In terms of matrices, this means that taking $T_n(\frac{z_\rho p_l^2}{q_m^2})$ as a preconditioner matrix for the solution of (1.1), the eigenvalues of $T_n^{-1}(\frac{z_\rho p_l^2}{q_m^2})T_n(f)$ are clustered in a small region near 1 [7] and the PCG method will become very fast. Unfortunately, because this matrix is a full Toeplitz matrix, is hard to construct, and is costly to invert, it is useless as a preconditioner. Instead, we are led to the idea of separating the numerator and the denominator of the ratio $\frac{z_\rho p_l^2}{q_m^2}$ and use as a preconditioner matrix the product of three matrices. More specifically, the preconditioner we propose for the solution of system (1.1) is

$$M_n = B_{nm}^{-1}(q) B_{n\hat{l}}(p^2 z_\rho) B_{nm}^{-1}(q), \qquad \hat{l} = 2l + \rho, \tag{3.1}$$

where the second index in the matrices represents their halfbandwidth, while the first one represents their dimension. The notation $B_{nm}(\cdot)$ will be used instead of $T_n(\cdot)$ for band Toeplitz matrices to emphasize their bandness. The following statements prove the basic assumptions a preconditioner must satisfy and also describe the spectrum of the preconditioned matrix $M_n^{-1}T_n$.

THEOREM 3.1. *The matrix M_n is symmetric and positive definite for every n.*

Proof. Its symmetry is implied directly from the definition (3.1). On the other hand, the eigenvalues of $B_{n\hat{l}}(p^2 z_\rho)$ belong to the interval $(\min p_l^2 z_\rho, \max p_l^2 z_\rho)$, where $0 = \min p_l^2 z_\rho < \max p_l^2 z_\rho \leq 2^\rho \max p_l^2$. Therefore, $B_n(p_l^2 z_\rho)$ is symmetric and positive definite. Furthermore, q_m has no zeros in $[-\pi, \pi]$ because it results from the rational approximation to a function which is strictly positive in $[-\pi, \pi]$. So, $B_{nm}(q)$ is symmetric and invertible. Then, for every $x \in \mathbb{R}^n$, $x \neq 0$, we have

$$x^T M_n x = x^T B_{nm}^{-1}(q) B_{n\hat{l}}(p^2 z_\rho) B_{nm}^{-1}(q) x = y^T B_{n\hat{l}}(p^2 z_\rho) y > 0,$$

where $y = B_{nm}^{-1}(q)x$. Hence M_n is symmetric and positive definite. ∎

Theorem 3.1 suggests that the matrix M_n can be taken as a preconditioner matrix. It then remains to study the convergence rate of the PCG method or, equivalently, how the eigenvalues of the matrix $M_n^{-1}T_n$ are distributed. For this, we give without proof the following lemma and then state and prove our main result in Theorem 3.2.

LEMMA 3.1. *Suppose $A, B \in \mathbb{R}^{n\times n}$ are symmetric matrices such that*

$$A = B + \epsilon cc^T,$$

where $c \in \mathbb{R}^n$, $c^T c = 1$. If $\epsilon > 0$, then

$$\lambda_1(B) \le \lambda_1(A) \le \lambda_2(B) \le \cdots \le \lambda_n(B) \le \lambda_n(A),$$

while if $\epsilon \le 0$, then

$$\lambda_1(A) \le \lambda_1(B) \le \lambda_2(A) \le \cdots \le \lambda_n(A) \le \lambda_n(B),$$

provided that the eigenvalues are labeled in nondecreasing order of magnitude. In either case

$$\lambda_k(A) = \lambda_k(B) + t_k\epsilon, \qquad k = 1, 2, \ldots, n,$$

where $t_k \ge 0$, $k = 1, 2, \ldots, n$, and $\sum_{k=1}^n t_k = 1$.

Proof. See Wilkinson [26, pp. 97–98] for the proof. ∎

THEOREM 3.2. *Let $\lambda_i(M_n^{-1}T_n)$, $i = 1(1)n$, and denote the eigenvalues of $M_n^{-1}T_n$ and m the degree of the denominator q_m of the rational approximation. Then, at least $n - 4m$ eigenvalues of the preconditioned matrix lie in $(h_{\min}, h_{\max})$, at most $2m$ are greater than $h_{\max}$, and at most $2m$ are in $(0, h_{\min})$, where $h = \frac{fq^2}{p^2 z_\rho}$.*

Proof. Obviously the matrix

$$M_n^{-1}T_n = B_{nm}(q)B_{n\hat{l}}^{-1}(p^2 z_\rho)B_{nm}(q)T_n(f)$$

is similar to the matrix

$$B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z_\rho)B_{nm}(q)T_n(f)B_{nm}(q)B(p^2 z_\rho)_{n\hat{l}}^{-\frac{1}{2}}. \tag{3.2}$$

Then, since $B_{nm}(q)$ is a band matrix with halfbandwidth m, the matrix $B_{nm}(q)T_n(f)$ differs from $T_n(qf)$ only in the m first and last rows, and the matrix $B_{nm}(q)T_n(f)B_{nm}(q)$ differs from $T_n(q^2 f)$ only in the first and last m rows and columns. So it can be written as a sum of a Toeplitz matrix and a low rank correction matrix, i.e.,

$$B_{nm}(q)T_n(f)B_{nm}(q) = T_n(q^2 f) + \Delta, \tag{3.3}$$

where Δ is a symmetric "border" matrix with nonzero elements only in the first and last m rows and columns. So $\operatorname{rank}(\Delta) \le 4m$ is independent of n. Then, from (3.2) and (3.3) we obtain that

$$\begin{aligned}\overbrace{B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z_\rho)B_{nm}(q)T_n(f)B_{nm}(q)B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z_\rho)}^{\mathrm{E}} = {} & \overbrace{B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z_\rho)T_n(q^2 f)B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z_\rho)}^{\widetilde{\mathrm{E}}} \\ & + B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z)\Delta B_{n\hat{l}}^{-\frac{1}{2}}(p^2 z).\end{aligned} \tag{3.4}$$

Since a matrix product does not have rank larger than that of each of the factors involved, there exist $\alpha_i > 0$, $c_i \in \mathbb{R}^n$, $i = 1(1)m_+$, and $\beta_i > 0$, $d_i \in \mathbb{R}^n, i = 1(1)m_-$, with $m_+ + m_- \le 4m$, such that (3.4) can be written as

$$E - \widetilde{E} = \sum_{i=1}^{m_+} \alpha_i c_i c_i^T - \sum_{i=1}^{m_-} \beta_i d_i d_i^T.$$

So applying successively $m_+ + m_-$ times Lemma 3.1 gives

$$h_{\min} \le \lambda_i(E) \le h_{\max}, \qquad m_- < i \le n - m_+,$$

and the theorem is proved. ∎

It is clear from the previous analysis and statements that contrary to what happens with other band Toeplitz preconditioners, the one we propose of the "premultiplier" matrix $B_{nm}(q)$ may make some of the eigenvalues lie outside the approximation interval $[h_{\min}, h_{\max}]$. We will prove now that the spectral radius of the preconditioned matrix is bounded by a constant number independent of n. For this, first, we state and prove the following lemma.

LEMMA 3.2. *Let B_n be an $n \times n$ symmetric and positive definite band Toeplitz matrix with halfbandwidth s. Then the $k \times k$ principal and trailing submatrices of B_n^{-1} as well as the $k \times k$ submatrices consisting from the first k rows and the last k columns (right upper corner) or from the last k rows and the first k columns (left lower corner) of B_n^{-1} are componentwise bounded for every fixed k independent of n.*

Proof. For principal and trailing submatrices, this property has been proved in [10] for $k = s$. We will prove the validity of this property for $k = s + 1$ and the proof of every fixed k can be completed by induction. From the fundamental relation

$$\sum_{l=1}^{s+1} b_{1l}(B_n^{-1})_{lj} = \delta_{1j},$$

where δ_{1j} is the Kronecker δ, we obtain successively that

$$(B_n^{-1})_{s+1,j} = \frac{1}{b_{1,s+1}} \left(\delta_{1j} - \sum_{l=1}^{s} b_{1l}(B_n^{-1})_{lj} \right), \qquad j = 1, 2, \ldots, s. \tag{3.5}$$

Since all the elements in the right-hand side of (3.5) are bounded, so are the elements $(B_n^{-1})_{s+1,j}$, $j = 1, 2, \ldots, s$. From the symmetry of B_n^{-1} we obtain that the elements $(B_n^{-1})_{j,s+1}$, $j = 1, 2, \ldots, s$, are also bounded. One more application of (3.5) for $j = s + 1$ gives us that the element $(B_n^{-1})_{s+1,s+1}$ is bounded, and the proof for the principal submatrices is complete. Since B_n^{-1} is a persymmetric matrix the elements of the trailing matrix are the same as those of the principal one in reverse order. So the $k \times k$ trailing matrix is also bounded.

It remains to prove the validity of the property for the submatrices in the right upper corner and in the left lower corner of B_n^{-1}. These matrices are transposes of each other due to the symmetry of B_n^{-1}. From the positive definiteness of B_n^{-1} we have that

$$|(B_n^{-1})_{ij}| < \frac{(B_n^{-1})_{ii} + (B_n^{-1})_{jj}}{2}, \qquad i = 1, \ldots, k, \quad j = n - k + 1, \ldots, n.$$

The elements in the right-hand side are the diagonal elements of the $k \times k$ principal and trailing submatrices, respectively, which are bounded, and the proof is complete. ∎

The following theorem proves that the eigenvalues of $M^{-1}T$ have an upper bound.

THEOREM 3.3. *Under the assumptions of Theorem* 3.2 *there exists a constant c, independent of n, such that $\rho\left(M_n^{-1}T_n(f)\right) \le c$ for every n.*

Proof. We begin the proof by using some relations connecting the spectral radii and the Rayleigh quotients of symmetric matrices. The fact that all the matrices are

positive definite is also used.

$$
\begin{aligned}
\rho\left(M_n^{-1}T_n(f)\right) &= \rho\left(B_{nm}(q)B_{n\hat{l}}^{-1}(p^2z_\rho)B_{nm}(q)T_n(f)\right)\\
&= \rho\left(B_{n\hat{l}}^{-\frac{1}{2}}(p^2z_\rho)B_{nm}(q)T_n(f)B_{nm}(q)B_{n\hat{l}}^{-\frac{1}{2}}(p^2z_\rho)\right)\\
&= \max_{x\neq 0}\frac{x^TB_{n\hat{l}}^{-\frac{1}{2}}(p^2z_\rho)B_{nm}(q)T_n(f)B_{nm}(q)B_{n\hat{l}}^{-\frac{1}{2}}(p^2z_\rho)x}{x^Tx}\\
&= \max_{x\neq 0}\left(\frac{x^TT_n(f)x}{x^TB_{nm}^{-1}(q)B_{n\hat{l}}(p^2z_\rho)B_{nm}^{-1}(q)x}\cdot\frac{x^TB_{n\hat{l}}(p^2z_\rho)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\right)\\
&= \max_{x\neq 0}\left(\frac{x^TT_n(f)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\cdot\frac{x^TB_{n\hat{l}}(p^2z_\rho)x}{x^TB_{nm}^{-1}(q)B_{n\hat{l}}(p^2z_\rho)B_{nm}^{-1}(q)x}\right) \qquad (3.6)\\
&\leq \max_{x\neq 0}\frac{x^TT_n(f)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\cdot\max_{x\neq 0}\frac{x^TB_{n\hat{l}}(p^2z_\rho)x}{x^TB_{nm}^{-1}(q)B_{n\hat{l}}(p^2z_\rho)B_{nm}^{-1}(q)x}\\
&= M_1\max_{x\neq 0}\frac{x^TB_{nm}(q)B_{n\hat{l}}(p^2z_\rho)B_{nm}(q)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\\
&= M_1\max_{x\neq 0}\frac{x^T\left(B_{n\hat{l}+2m}(q^2p^2z_\rho)+\Delta\right)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\\
&\leq M_1\left(M_2+\max_{x\neq 0}\frac{x^T\Delta x}{x^TB_{n\hat{l}}(p^2z_\rho)x}\right)\\
&\leq M_1\left(M_2+\rho\left(B_{n\hat{l}}^{-1}(p^2z_\rho)\Delta\right)\right).
\end{aligned}
$$

In (3.6) we have taken

$$M_1=\max_{x\neq 0}\frac{x^TT_n(f)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}=\rho\left(B_{n\hat{l}}^{-1}(p^2z_\rho)T_n(f)\right)$$

and

$$M_2=\max_{x\neq 0}\frac{x^TB_{n\hat{l}+2m}(q^2p^2z_\rho)x}{x^TB_{n\hat{l}}(p^2z_\rho)x}=\rho\left(B_{n\hat{l}}^{-1}(p^2z_\rho)B_{n\hat{l}+2m}(q^2p^2z_\rho)\right),$$

which are bounded, since the generating functions $\frac{f}{p^2z_\rho}$ and $\frac{q^2p^2z_\rho}{p^2z_\rho}=q^2$, respectively, are bounded functions in $[-\pi,\pi]$. In (3.6), the matrix product $B_{nm}(q)B_{n\hat{l}}(p^2z_\rho)B_{nm}(q)$ was written as the band Toeplitz matrix $B_{n\hat{l}+2m}(q^2p^2z_\rho)$, generated by the function $q^2p^2z_\rho$, plus the low rank correction matrix Δ.

It is known [5] that the matrix Δ is given by

$$
\begin{aligned}
\Delta = {} & B_{nm}(q)H(q)H(p^2z_\rho)+B_{nm}(q)H^R(q)H^R(p^2z_\rho)\\
& +H(q)H(qp^2z_\rho)+H^R(q)H^R(qp^2z_\rho),
\end{aligned}
$$

where $H(q)$, $H(p^2z_\rho)$, and $H(qp^2z_\rho)$ are Hankel matrices produced by the trigonometric polynomials q, p^2z_ρ, and qp^2z_ρ, respectively, while H^R denotes the matrix obtained from H by reversing the order of its rows and columns.

It is obvious that Δ is a low rank correction matrix that has nonzero elements only in the upper left and lower right triangles, as illustrated below:

$$\Delta = \begin{pmatrix} * & \cdots & * & 0 & \dots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 & \vdots \\ * & 0 & \ddots & 0 & & 0 \\ 0 & \ddots & 0 & \ddots & 0 & * \\ \vdots & 0 & & 0 & \ddots & \vdots \\ 0 & \dots & 0 & * & \dots & * \end{pmatrix}.$$

It is clear that the elements of Δ are bounded and the size of the triangles depends only on the bandwidths m and $\hat{l}$ and are independent of n.

It remains to prove that $\rho(B_{n\hat{l}}^{-1}(p^2 z_\rho)\Delta)$ is bounded. For this, we write the matrices in the following block forms:

$$B_{n\hat{l}}^{-1}(p^2 z_\rho) = \begin{pmatrix} B_1 & * & B_2 \\ * & * & * \\ B_2^T & * & B_1^R \end{pmatrix}, \qquad \Delta = \begin{pmatrix} D & & \\ & O & \\ & & D^R \end{pmatrix},$$

where B_1, B_2 are $k \times k$ matrices if D has k nonzero antidiagonals.

Since the only nonzero columns of the matrix $B_{n\hat{l}}^{-1}(p^2 z_\rho)\Delta$ are its first k and last k ones, the nonidentically zero eigenvalues of $B_{n\hat{l}}^{-1}(p^2 z_\rho)\Delta$ will be the eigenvalues of the matrix

$$\begin{pmatrix} B_1 D & B_2 D^R \\ B_2^T D & B_1^R D^R \end{pmatrix}.$$

In view of Lemma 3.2 this matrix is bounded, and so are its eigenvalues, which proves the present statement. ∎

So, the eigenvalues that are greater than $h_{\max}$ have an upper bound.

To study the behavior of the eigenvalues that lie in the interval $(0, h_{min})$ we prove the following Lemma.

LEMMA 3.3. *The smallest eigenvalue of the matrix $M_n^{-1}T_n(f)$ has a bound from below a constant number $c_1 > 0$, independent of n, iff the smallest eigenvalue of the matrix $B_{n\rho}^{-1}(z_\rho)B_{nm}(q)B_{n\rho}(z_\rho)B_{nm}(q)$ has lower bound a constant number $c_2 > 0$, independent of n.*

Proof. As in Theorem 3.3 we use the relation connecting the eigenvalues of a symmetric positive definite matrix with the Rayleigh quotient:

$$\begin{aligned} &\min_i \lambda_i \left(M_n^{-1}T_n(f)\right) = \min_i \lambda_i \left(B_{n\hat{l}}^{-1}(p^2 z_\rho)B_{nm}(q)T_n(f)B_{nm}(q)\right) \\ &= \min_{x\neq 0} \left(\frac{x^T B_{nm}(q)T_n(f)B_{nm}(q)x}{x^T B_{n\hat{l}}(p^2 z_\rho)x} \cdot \frac{x^T B_{nm}(q)B_{n\rho}(z_\rho)B_{nm}(q)x}{x^T B_{nm}(q)B_{n\rho}(z_\rho)B_{nm}(q)x} \cdot \frac{x^T B_{n\rho}(z_\rho)x}{x^T B_{n\rho}(z_\rho)x} \right) \\ &\geq \min_{x\neq 0} \frac{x^T T_n(f)x}{x^T B_{n\rho}(z_\rho)x} \cdot \min_{x\neq 0} \frac{x^T B_{nm}(q)B_{n\rho}(z_\rho)B_{nm}(q)x}{x^T B_{n\rho}(z_\rho)x} \cdot \min_{x\neq 0} \frac{x^T B_{n\rho}(z_\rho)x}{x^T B_{n\hat{l}}(p^2 z_\rho)x} \\ &\geq \min \frac{f}{z_\rho} \cdot \min \frac{1}{p^2} \cdot \min_{x\neq 0} \frac{x^T B_{nm}(q)B_{n\rho}(z_\rho)B_{nm}(q)x}{x^T B_{n\rho}(z_\rho)x}. \end{aligned}$$

Since the functions $\frac{f}{z_\rho}$ and $\frac{1}{p^2}$ have both lower bounds independent of n, the spectrum of the preconditioned matrix has such a bound iff the Rayleigh quotient $\frac{x^T B_{nm}(q) B_{n\rho}(z_\rho) B_{nm}(q) x}{x^T B_{n\rho}(z_\rho) x}$ does. □

The above equivalent problem that the matrix $B_{n\rho}^{-1}(z_\rho) B_{nm}(q) B_{n\rho}(z_\rho) B_{nm}(q)$ has a spectrum bounded from below by a positive constant c independent of n remains in this paper an open question for general values of the bandwidths m and ρ. Despite that, strong numerical evidence shows that this holds. To make our conjecture stronger we present the proof for the special cases where $m = 1$ and $\rho = 1, 2$.

THEOREM 3.4. *The matrix $B_{n\rho}^{-1}(z_\rho) B_{nm}(q) B_{n\rho}(z_\rho) B_{nm}(q)$ has its smallest eigenvalue λ_1 bounded from below by a constant number $c > 0$ which is independent of n for $m = 1$ and $\rho = 1, 2$*

Proof. The case $m = \rho = 1$ is quite obvious and is based on the fact that all the tridiagonal symmetric Toeplitz matrices have the same eigenvectors. More specifically, the matrix $B_{n1}(z_1)$ is the Laplace matrix with its eigenvalues and the corresponding normalized eigenvectors being given by

$$\lambda_i = z_1(\theta_i) = 4 \sin^2 \frac{\theta_i}{2}, \qquad x^{(i)} = \sqrt{\frac{2}{n+1}} \left(\sin\theta_i \ \ \sin 2\theta_i \ \ \sin 3\theta_i \ \ \ldots \ \ \sin n\theta_i \right)^T,$$

respectively, where $\theta_i = \frac{\pi i}{n+1},\ i = 1(1)n$. The matrix $B_{n1}(q)$ is a tridiagonal Toeplitz matrix of the form $tridiag(\beta,\ \alpha,\ \beta)$. Since $B_{n1}(q)$ and $B_{n1}(z_1)$ have the same eigenvectors we can write any arbitrary vector $x \in \mathbb{R}^n$ as a convex combination $x = \sum_{i=1}^n c_i x^{(i)},\ c_i \in \mathbb{R},\ i = 1(1)n$. With these assumptions and using the orthogonal properties of $x^{(i)}$'s the Rayleigh quotient gives

$$\begin{aligned} \frac{x^T B_{n1}(q) B_{n1}(z_1) B_{n1}(q) x}{x^T B_{n1}(z_1) x} &= \frac{\left(\sum_{i=1}^n c_i x^{(i)}\right)^T B_{n1}(q) B_{n1}(z_1) B_{n1}(q) \left(\sum_{i=1}^n c_i x^{(i)}\right)}{\left(\sum_{i=1}^n c_i x^{(i)}\right)^T B_{n1}(z_1) \left(\sum_{i=1}^n c_i x^{(i)}\right)} \\ &= \frac{\sum_{i=1}^n c_i^2 q^2(\theta_i) 4 \sin^2 \frac{\theta_i}{2}}{\sum_{i=1}^n c_i^2 4 \sin^2 \frac{\theta_i}{2}} \geq \min_i q^2(\theta_i) \geq \min_{\theta \in [-\pi, \pi]} q^2(\theta). \end{aligned} \tag{3.7}$$

The proof is complete since the function q is strictly positive.

For the case where $(m, \rho) = (1, 2)$ we write the matrix $B_{n2}(z_2)$ as a function of $B_{n1}(z_1)$ and the corresponding Hankel matrices [5], i.e.,

$$B_{n2}(z_2) = \left(B_{n1}(z_1)\right)^2 + \left(H(z_1) + H^R(z_1)\right)^2,$$

where the notations H and H^R are the same as in Theorem 3.3. For simplicity we denote $H = H(z_1) + H^R(z_1)$, so $H = \mathrm{diag}(-1,\ 0,\ 0, \ldots,\ 0,\ -1)$.

By considering the same convex combination of the vector x, the Rayleigh quotient gives

$$\begin{aligned} \frac{x^T B_{n1}(q) B_{n2}(z_2) B_{n1}(q) x}{x^T B_{n2}(z_2) x} &= \frac{x^T B_{n1}(q) \left(B_{n1}^2(z_1) + H^2\right) B_{n1}(q) x}{x^T \left(B_{n1}^2(z_1) + H^2\right) x} \\ &= \frac{16 \sum_{i=1}^n c_i^2 q^2(\theta_i) \sin^4 \frac{\theta_i}{2} + \frac{2}{n+1} \left(\sum_{i=1}^n c_i q(\theta_i) \sin\theta_i\right)^2 + \frac{2}{n+1} \left(\sum_{i=1}^n c_i q(\theta_i) \sin n\theta_i\right)^2}{16 \sum_{i=1}^n c_i^2 \sin^4 \frac{\theta_i}{2} + \frac{2}{n+1} \left(\sum_{i=1}^n c_i \sin\theta_i\right)^2 + \frac{2}{n+1} \left(\sum_{i=1}^n c_i \sin n\theta_i\right)^2}. \end{aligned} \tag{3.8}$$

First, we suppose that the first term of the denominator in (3.8) is greater than or

equal to the second or the third one in order of magnitude. In that case we obtain that the ratio in (3.8), similar to (3.7), has a lower bound the value $\min_{\theta\in[-\pi,\pi]} q^2(\theta)$. Otherwise, we suppose that the second term is greater than the others in order of magnitude. Since the numerator is a sum of quadratic terms, the ratio will tend to zero if all the terms in the numerator decrease with a higher rate. So, we consider the case where the term $\frac{2}{n+1}\left(\sum_{i=1}^{n} c_i q(\theta_i)\sin\theta_i\right)^2$ has an order of magnitude less than that of $\frac{2}{n+1}\left(\sum_{i=1}^{n} c_i \sin\theta_i\right)^2$. By substituting $q(\theta)=\alpha+2\beta\cos\theta_i=\alpha+2\beta(1-2\sin^2\frac{\theta_i}{2})$, we have

$$\sum_{i=1}^{n} c_i q(\theta_i)\sin\theta_i=(\alpha+2\beta)\sum_{i=1}^{n} c_i\sin\theta_i-4\beta\sum_{i=1}^{n} c_i\sin^2\frac{\theta_i}{2}\cdot\sin\theta_i,$$

which means that the terms $\sum_{i=1}^{n} c_i\sin\theta_i$ and $\sum_{i=1}^{n} c_i\sin^2\frac{\theta_i}{2}\cdot\sin\theta_i$ must have the same orders of magnitude. Applying the Cauchy–Schwarz inequality on the second sum we obtain that

$$\left(\sum_{i=1}^{n} c_i\sin^2\frac{\theta_i}{2}\cdot\sin\theta_i\right)^2\le\sum_{i=1}^{n} c_i^2\sin^4\frac{\theta_i}{2}\cdot\sum_{i=1}^{n}\sin^2\theta_i=\frac{n+1}{2}\sum_{i=1}^{n} c_i^2\sin^4\frac{\theta_i}{2}.$$

So, the order of magnitude of the term $\frac{2}{n+1}\left(\sum_{i=1}^{n} c_i\sin\theta_i\right)^2$ must be less than or equal to the one of $\sum_{i=1}^{n} c_i^2\sin^4\frac{\theta_i}{2}$, which is a contradiction. The assumption that the third term is the greater one, in order of magnitude, gives similarly the same contradiction. So, the ratio in (3.8) does not tend to zero as n tends to infinity. □

We remark that the same idea to split the matrix $B_{n\rho}(z_\rho)$ into $(B_{n1}(z_1))^\rho$ plus a sum of Hankel matrices can be used for the proof of the above property in the case of $\rho>2$. In the case of $m>1$, first the matrix $B_{nm}(q)$ is written as a sum of the terms $B_{nj}(z_j)$ $j=0(1)m$, $(B_{n0}(z_0)=I_n)$ and the above idea can be applied. In both cases the analysis becomes more and more complicated. Figures 5.1(b)–(d), 5.2(b), 5.3(b) fully confirm the above properties. Moreover they show that the main interval eigenvalues appear in pairs and the elements of each pair tend to each other as n tends to infinity. In view of this observation, the convergence analysis of the PCG method in [1] assures us that our method will not be seriously affected and the convergence of it will remain superlinear, which is the optimal cost for this method.

4. Computational analysis and modification of the method. In this section we will try to compare, from the computational point of view, our preconditioner with the most recent band Toeplitz preconditioner proposed in [21]. The latter has in general the best performance from all the previous ones, when the generating function f is nonnegative and has zeros of even order.

The main computational cost in every PCG iteration is due to the Toeplitz matrix-vector product $T_n(f)x$ and to the solution of a system with coefficient matrix the preconditioner itself. The first one is the same for both methods and can be computed by means of the fast Fourier transform (FFT) in $10(n\log 2n)$ operations (ops) in a sequential machine, or in $O(\log 2n)$ steps in the parallel PRAM model of computation, when $O(n)$ processors are used. For the inversion of the preconditioners, things slightly change. If we use band Toeplitz preconditioners, then their halfbandwidth $\hat{l}_1$ represents the degree l_1 of the Chebyshev approximation plus the degree ρ of the

trigonometric polynomial, which eliminates the zeros of f. The inversion of such type of matrices can be achieved using the LDL^T factorization method in $n(\hat{l}_1^2 + 8\hat{l}_1 + 1)$ ops. We mention that this method is more preferable than the band Cholesky factorization because the latter requires the computation of n square roots, which is quite expensive when n is large.

In the case of our preconditioner the inversion requires two band matrix vector products of total cost $n(8m + 4)$ ops, where m is the halfbandwidth and coincides with the degree of the denominator in the rational approximation. In addition, the inversion of $B_{n\hat{l}_2}$, as in the previous case, can be performed in $n(\hat{l}_2^2 + 8\hat{l}_2 + 1)$ ops, where $\hat{l}_2 = \rho + 2l_2$ and l_2 represents the degree of the numerator of the rational approximation. So the total cost per iteration for this step of the algorithm of the PCG method is about

$$Cost_{it} = n(\hat{l}_2^2 + 8\hat{l}_2 + 8m + 5).$$

We must mention here that more sophisticated techniques reduce the cost of approximating the solution of such systems, up to within an $O(\epsilon)$ error, in $O(n \log m + m \log^2 m \log\log \epsilon^{-1})$ [4]. In both cases, when n is large, the complexity of the method is strongly dominated by that of the first step, which requires $O(n \log 2n)$ ops since $\hat{l}_2, m$ are independent of n. So the methods are essentially equivalent in complexity per iteration. Thus the costs of finding $B_{n\hat{l}_1}^{-1}$ and $B_{nm}B_{n\hat{l}_2}^{-1}B_{nm}$, where $l_1 = l_2 + m$, are comparable.

In case n is not large enough, taking $l_2 = \frac{l_1}{2} - 1$ and making some calculations, we can see that the two preconditioning strategies are approximately equivalent even when $m = \rho l_1$.

According to this observation, if we have two candidates of rational approximations of f with almost the same relative error and degrees (l_1, m_1), (l_2, m_2) with $l_1 + m_1 \approx l_2 + m_2$, it is preferable, from the computation point of view, to choose as the generating function for our preconditioner the one which has the larger m and the smaller l.

Finally, we will focus on the calculation of rational approximation of degree (l, m) of a positive continuous function f. In the recent literature many different strategies that produce this kind of approximation [17] can be found. Each of them is most suitable for certain classes of functions, but the one which is based on the Remez algorithm seems to be, in general, the most appropriate for a large variety of functions. The starting point of this category of algorithms is to construct a rational approximation using rational interpolation, and then this rational approximation is used to generate a better approximation until an alternative set of $m + l + 2$ points is reached. This procedure consists of adjusting the choice of the interpolation points in such a way as to ensure that the relative error decreases. In practice this method can fail in some cases. Usually, problems occur either because the extreme values of the relative error occur more than $m + l + 2$ times, or because the starting rational interpolation has zeros in the interval in which this approximation is sought. The first difficulty is usually overcome by seeking a rational approximation of a different degree or by designing a more robust algorithm. A trick that often works in the latter case is, instead of seeking again for a rational approximation of a different degree, to start with an approximation that is valid over a shorter interval and to use it as a starting point for an approximation on a slightly larger interval. Iterative application of this procedure may enable us to obtain a final approximation in the desired interval.

For the convergence rate of the approximation method we cannot give a theoretical result, but the facts that its computational cost is independent of n and the computations are done only once for a given function make us believe that this issue does not play an important role in the whole procedure.

4.1. Modification of the method. The idea of constructing a preconditioner from a rational approximation of a function can be used in exactly the same way in case of rational interpolation at the Chebyshev points. The advantage of this modification is the simplicity of its calculation. Nevertheless, it is worth noticing that we cannot ensure that this interpolation would not have zeros in the interval of approximation. Despite this, whenever the preconditioning gives us poor results, this technique may give, at least for certain classes of f, results similar to the corresponding ones by the best Chebyshev approximation.

5. Numerical examples and concluding remarks. In this section, we present some numerical examples. The aim of these examples is twofold: (i) to show, by numerical evidence, the correctness of our observations regarding the asymptotical spectral analysis of the preconditioned matrices, and (ii) to compare the convergence rate of our preconditioner with that of the band Toeplitz preconditioner proposed in [21]. We use the latter to compare it with ours because it is the most efficient technique for preconditioning Toeplitz matrices generating by functions with zeros of even order. Our test functions are the following:

$$\text{(i)} \quad f_1(x) = x^4,$$
$$\text{(ii)} \quad f_2(x) = \frac{2x^4}{1+25x^2},$$

and

$$\text{(iii)} \quad f_3(x) = \begin{cases} (x-3)^4(x-1)^2, & 0 \le x \le \pi, \\ (x+3)^4(x+1)^2, & -\pi \le x \le 0. \end{cases}$$

An effort was made to choose functions of different behaviors which produce ill-conditioned matrices T_n. The Toeplitz matrices produced have Euclidean condition numbers of order $O(n^4)$. In our experiments we solve the system $T_n(f)x = b$, where b is the vector having all its components equal to 1. As a starting initial guess of solution the zero vector is used and as a stopping criterion the validity of $\frac{\|r_k\|_2}{\|r_0\|_2} \le 10^{-7}$ is considered, where r_k is the residual vector after k iterations. The construction of matrices and the rational approximations were performed using *Mathematica* in order to have more accurate results, while all the other computations were performed using MATLAB.

In Tables 5.1, 5.2, and 5.3 we report the number of iterations needed until convergence is achieved in each case; B_n^{*l} denotes the optimal band Toeplitz preconditioner [21] which is generated by the trigonometric polynomial $z_\rho g_l$, with g_l being the best Chebyshev approximation of $\frac{f}{z_\rho}$ out of $\mathcal{P}_l$, $\hat{B}_n^l$ is the band Toeplitz preconditioner where $\hat{g}_l$ is the interpolation polynomial at the Chebyshev points, $M_n^{l,m}$ denotes our main proposed preconditioner obtained by the best rational approximation procedure of degree (l,m), and $R_n^{l,m}$ denotes the preconditioner that results after applying rational interpolation of degree (l,m).

TABLE 5.1
Number of iterations for $f_1(x)$.

n	B_n^{*1}	$\hat{B}_n^1$	B_n^{*3}	$\hat{B}_n^3$	B_n^{*4}	$\hat{B}_n^4$	$M_n^{0,1}$	$R_n^{0,1}$	$M_n^{1,1}$	$R_n^{1,1}$	$M_n^{1,2}$	$R_n^{1,2}$
16	9	8	9	7	7	6	8	7	6	6	5	5
32	10	10	11	8	9	7	10	9	7	7	6	6
64	13	12	11	10	9	8	11	11	9	9	8	8
128	15	15	12	11	10	10	12	13	11	11	10	10
256	16	16	12	13	10	10	13	13	12	12	11	11
512	16	16	13	13	10	11	13	14	13	13	11	12

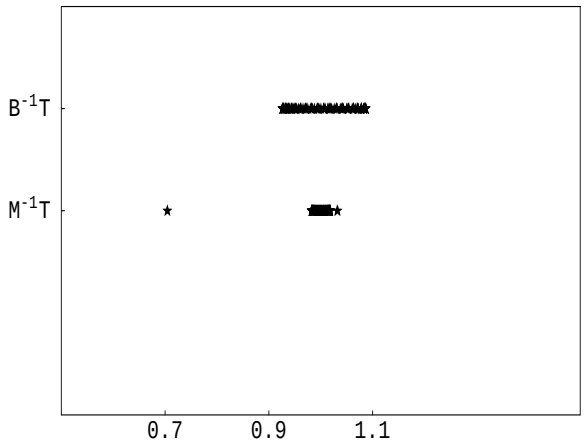
TABLE 5.2
Number of iterations for $f_2(x)$.

n	B_n^{*3}	B_n^{*4}	B_n^{*5}	B_n^{*6}	$M_n^{1,1}$	$R_n^{2,2}$
16	8	8	7	8	8	6
32	13	13	12	11	11	7
64	19	18	15	13	12	9
128	24	19	17	14	12	11
256	25	21	18	15	13	13
512	27	22	18	16	14	14

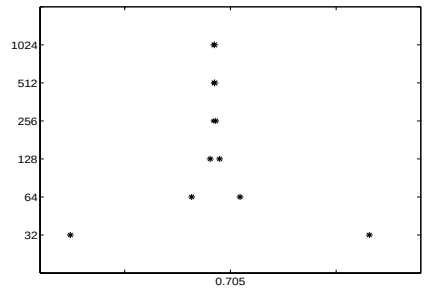
TABLE 5.3
Number of iterations for $f_3(x)$.

n	B_n^{*3}	B_n^{*5}	B_n^{*7}	$M_n^{1,2}$	$R_n^{(1,2)}$
16	9	7	7	9	8
32	17	14	13	18	11
64	34	28	22	21	14
128	65	48	36	21	20
256	111	69	54	23	24
512	152	93	66	23	27

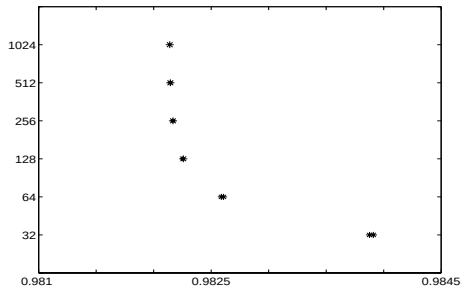
In Figures 5.1(a), 5.2(a), 5.3(a), the spectra of the matrices $M_n^{-1}T_n(f_i)$, $i = 1, 2, 3$, are illustrated, while in Figures 5.1(b)-(d), 5.2(b), 5.3(b) we focus on the behavior of the pairs of eigenvalues of the matrix lying outside the interval $[h_{\min}, h_{\max}]$ for different values of n. The boundness and the convergence in pairs is obvious in all figures. Especially, we stress the case of Figures 5.1 and 5.3, where as we expected from the theory at most eight eigenvalues would lie outside the interval $[h_{\min}, h_{\max}]$, but in practice, for the first test function, only three pairs of eigenvalues lie outside this interval, one of which (the second lower pair) moves very close to the lower bound $h_{\min} = 0.98214$, while, for the third test function, only two pairs lie outside this interval. Finally, we remark that in the case of f_3 and for $n = 512$, the preconditioning by band Toeplitz B^{*3} "clusters" the eigenvalues of the preconditioned matrix in $[0.5, 584.3]$, B^{*5} does so in $[0.36, 104.7]$, while $M^{1,2}$ collects the main mass of them in $[0.67, 1.65]$ and $R^{1,2}$ collects it in $[0.95, 14.25]$.



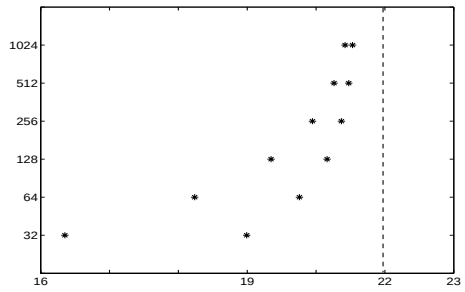
(a) The main mass of the eigenvalues of the preconditioned matrices.



(b) The lower extreme pair.

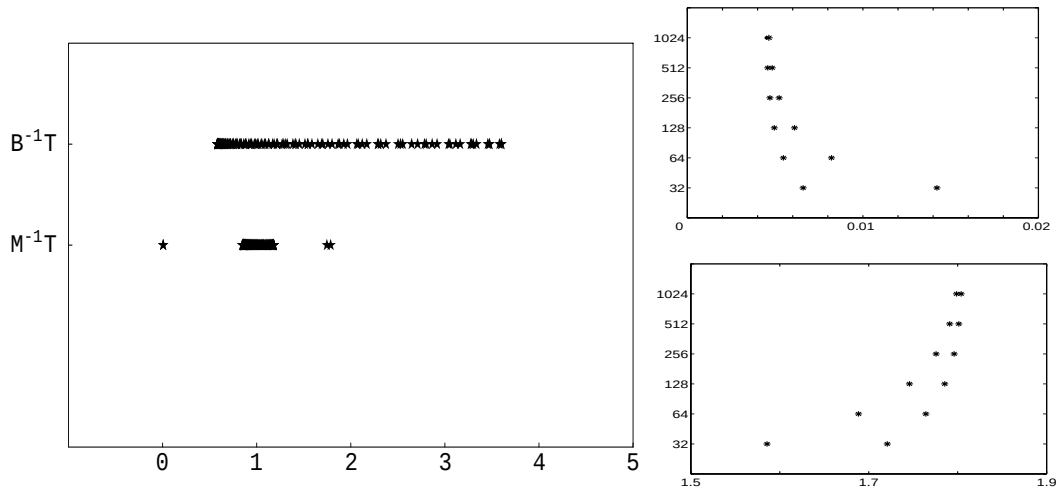


(c) The second upper pair.



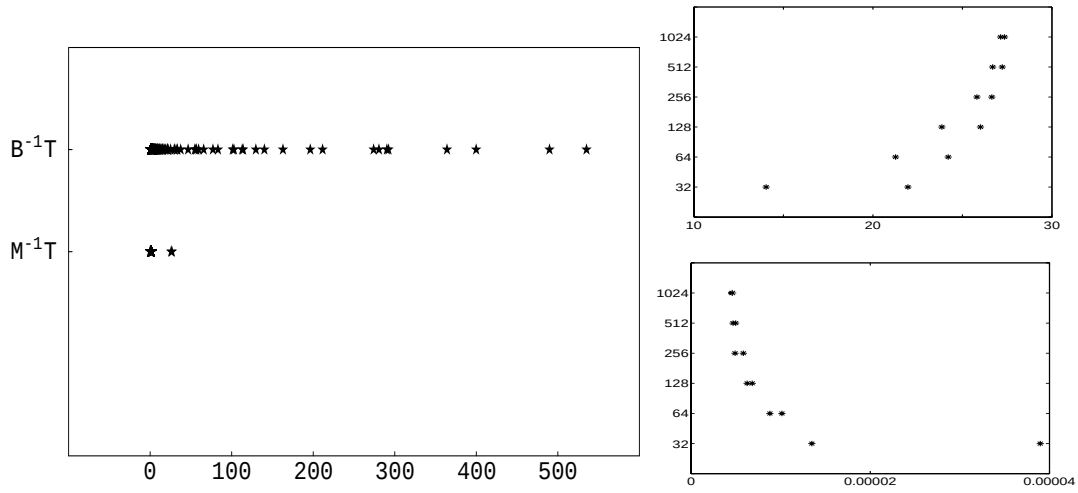
(d) The upper extreme pair.

FIG. 5.1. *Spectra of* $(M_n^{2,2})^{-1}T_n(f_1)$ *and* $(B_n^{*5})^{-1}T_n(f_1)$ *for* $n = 128$ *and behavior of the pairs of eigenvalues that lie outside the interval* $[h_{\min}, h_{\max}]$ *with* $h_{\min} = 0.98214$.



(a) (b)

FIG. 5.2. *Spectra of* $(M_n^{1,1})^{-1}T_n(f_2)$ *and* $(B_n^{*3})^{-1}T_n(f_2)$ *for* $n = 128$ *and behavior of the pairs of eigenvalues that lie outside the interval* $[h_{min}, h_{max}]$.



(a) (b)

FIG. 5.3. *Spectra of* $(M_n^{1,2})^{-1}T_n(f_3)$ *and* $(B_n^{*3})^{-1}T_n(f_3)$ *for* $n = 256$ *and behavior of the pairs of eigenvalues that lie outside the interval* $[h_{min}, h_{max}]$.

REFERENCES

[1] O. Axelsson and G. Lindskog, *On the rate of convergence of the preconditioned conjugate gradient method*, Numer. Math., 48 (1986), pp. 499–523.

[2] D. Bini and F. Di Benedetto, *A new preconditioner for parallel solution of positive definite Toeplitz systems*, in Proceedings of the Second Annual Symposium on Parallel Algorithms and Architectures, Crete, Greece, 1990, pp. 220–223.

[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. A. Bini and B. Meini, *Effective methods for solving banded Toeplitz systems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 700–719.

[5] A. Böttcher and B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices*, Springer-Verlag, New York, 1998.

[6] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 349–364.

[7] R. H. Chan, *Preconditioners for Toeplitz systems with nonnegative generating functions*, IMA J. Numer. Anal., 11 (1991), pp. 333–345.

[8] R. H. Chan and K. P. Ng, *Toeplitz preconditioners for Hermitian Toeplitz systems*, Linear Algebra Appl., 190 (1993), pp. 181–208.

[9] R. H. Chan and P. T. P. Tang, *Fast band-Toeplitz preconditioners for Hermitian Toeplitz systems*, SIAM J. Sci. Comput., 15 (1994), pp. 164–171.

[10] F. Di Benedetto, *Analysis of preconditioning techniques for ill-conditioned Toeplitz matrices*, SIAM J. Sci. Comput., 16 (1995), pp. 682–697.

[11] F. Di Benedetto, G. Fiorentino, and S. Serra, *C.G. preconditioning of Toeplitz matrices*, Comput. Math. Appl., 25 (1993), pp. 35–45.

[12] F. Di Benedetto and S. Serra, *A unifying approach to abstract matrix algebra preconditioning*, Numer. Math., (1999), pp. 57–90.

[13] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*, 2nd ed., Chelsea, New York, 1984.

[14] X. Q. Jin, *Hartley preconditioners for Toeplitz systems generated by positive continuous functions*, BIT, 34 (1994), pp. 367–371.

[15] T. K. Ku and C. C. Kuo, *A minimum-phase LU factorization preconditioner for Toeplitz matrices*, SIAM J. Sci. Comput., 13 (1992), pp. 1470–1487.

[16] G. Lorentz, *Approximation of Functions*, 2nd ed., Chelsea, New York, 1986.

[17] M. Powell, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, UK, 1982.

[18] T. Rivlin, *Introduction to the Approximation of Functions*, Dover, New York, 1981.

[19] S. Serra, *New PCG based algorithms for the solution of Hermitian Toeplitz systems*, Calcolo, 32 (1995), pp. 154–176.

[20] S. Serra, *On the extreme spectral properties of Toeplitz matrices generated by l^1 functions with several minima (maxima)*, BIT, 36 (1996), pp. 135–142.

[21] S. Serra, *Optimal, quasi-optimal and superlinear band-Toeplitz preconditioners for asymptotically ill-conditioned positive definite Toeplitz systems*, Math. Comput., 66 (1997), pp. 651–665.

[22] S. Serra Capizzano, *Toeplitz preconditioners constructed from linear approximation processes*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 446–465.

[23] S. Serra, *A Korovkin-type theory for finite Toeplitz operators via matrix algebras*, Numer. Math., 82 (1999), pp. 117–142.

[24] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[25] H. Widom, *Toeplitz matrices*, in Studies in Real and Complex Analysis, Stud. Math. 3, Math. Assoc. Amer., Buffalo, NY, 1965, pp. 179–209.

[26] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 744–760
© 2002 Society for Industrial and Applied Mathematics

CONVERGENCE ANALYSIS OF THE LATOUCHE–RAMASWAMI ALGORITHM FOR NULL RECURRENT QUASI-BIRTH-DEATH PROCESSES*

CHUN-HUA GUO†

Abstract. The minimal nonnegative solution G of the matrix equation $G = A_0 + A_1G + A_2G^2$, where the matrices A_i $(i = 0, 1, 2)$ are nonnegative and $A_0 + A_1 + A_2$ is stochastic, plays an important role in the study of quasi-birth-death processes (QBDs). The Latouche–Ramaswami algorithm is a highly efficient algorithm for finding the matrix G. The convergence of the algorithm has been shown to be quadratic for positive recurrent QBDs and for transient QBDs. In this paper, we show that the convergence of the algorithm is linear with rate 1/2 for null recurrent QBDs under mild assumptions. This new result explains the experimental observation that the convergence of the algorithm is still quite fast for nearly null recurrent QBDs.

Key words. matrix equations, minimal nonnegative solution, Markov chains, cyclic reduction, iterative methods, convergence rate

AMS subject classifications. 15A24, 15A51, 60J10, 60K25, 65U05

PII. S0895479800381872

1. Introduction.

A discrete-time quasi-birth-death process (QBD) is a Markov chain with state space $\{(i, j) \,|\, i \geq 0, 1 \leq j \leq m\}$, which has a transition probability matrix of the form

$$P = \begin{pmatrix} B_0 & B_1 & 0 & 0 & \cdots \\ A_0 & A_1 & A_2 & 0 & \cdots \\ 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where B_0, B_1, A_0, A_1, and A_2 are $m \times m$ nonnegative matrices such that P is stochastic. In particular, $(A_0 + A_1 + A_2)e = e$, where e is the column vector with all components equal to one. The matrix P is also assumed to be irreducible. Thus, $A_0 \neq 0$ and $A_2 \neq 0$.

The matrix equation

$$G = A_0 + A_1G + A_2G^2 \tag{1.1}$$

plays an important role in the study of the QBD (see [12] and [16]). It is known that (1.1) has at least one solution in the set $\{G \geq 0 \mid Ge \leq e\}$ (i.e., the set of substochastic matrices). The desired solution G is the minimal nonnegative solution.

We assume that $A = A_0 + A_1 + A_2$ is irreducible. Then, by the Perron–Frobenius theorem (see [17]), there exists a unique vector $\alpha > 0$ with $\alpha^T e = 1$ and $\alpha^T A = \alpha^T$. The vector α is called the stationary probability vector of A. By Theorem 7.2.3 in [12], the QBD is null recurrent if $\alpha^T A_0 e = \alpha^T A_2 e$; positive recurrent if $\alpha^T A_0 e > \alpha^T A_2 e$;

*Received by the editors December 4, 2000; accepted for publication (in revised form) by D. O'Leary July 13, 2001; published electronically January 11, 2002. This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.
http://www.siam.org/journals/simax/23-3/38187.html
†Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca).

and transient if $\alpha^T A_0 e < \alpha^T A_2 e$. For our purpose, we may use this criterion as an alternative definition for the three classes of QBDs.

The minimal nonnegative solution of (1.1) can be found by any of the following three fixed-point iterations (see [3], [5], [9], [10], [14], [15], [18]):

$$G_{n+1} = A_0 + A_1 G_n + A_2 G_n^2, \quad G_0 = 0, \tag{1.2}$$

$$G_{n+1} = (I - A_1)^{-1}(A_0 + A_2 G_n^2), \quad G_0 = 0, \tag{1.3}$$

$$G_{n+1} = (I - A_1 - A_2 G_n)^{-1} A_0, \quad G_0 = 0. \tag{1.4}$$

Among the three iterations, iteration (1.4) has the fastest rate of convergence. An inversion-free version of (1.4) has also been proposed in [1] and analyzed in [1] and [5]. These four iterations are adequate for most situations. However, the convergence of all four iterations is sublinear when the QBD is null recurrent (see [5]). The convergence of these methods is also extremely slow if the QBD is nearly null recurrent.

The algorithm proposed by Latouche and Ramaswami [11] is a little more complicated. However, it works very well even for nearly null recurrent QBDs.

The algorithm is as follows.

ALGORITHM 1.1. *Set*

$$\begin{aligned} H_0 &= (I - A_1)^{-1} A_2; \\ L_0 &= (I - A_1)^{-1} A_0; \\ G_0 &= L_0; \\ T_0 &= H_0. \end{aligned}$$

For $k = 0, 1, \ldots,$ *compute*

$$\begin{aligned} U_k &= H_k L_k + L_k H_k; \\ H_{k+1} &= (I - U_k)^{-1} H_k^2; \\ L_{k+1} &= (I - U_k)^{-1} L_k^2; \\ G_{k+1} &= G_k + T_k L_{k+1}; \\ T_{k+1} &= T_k H_{k+1}. \end{aligned}$$

It is shown in [11] that the matrices H_k and L_k are well defined and nonnegative and that the sequence $\{G_k\}$ converges quadratically to the matrix G for positive recurrent QBDs and for transient QBDs. The algorithm is called a logarithmic reduction algorithm in [11]. We will call it the LR algorithm (for logarithmic reduction or for Latouche–Ramaswami). A similar method is proposed in [2] for positive recurrent QBDs.

Since the LR algorithm has the greatest advantage over the fixed-point iterations when the QBD is nearly null recurrent, it is important to know the convergence rate of the LR algorithm when the QBD is null recurrent.

Before we can determine the convergence rate, we will take a closer look into the LR algorithm and present some preliminary results.

2. Preliminaries. It was mentioned in [11] that G. W. Stewart pointed out that the LR algorithm is related to the cyclic reduction technique. We will make this point more transparent and derive two equations relating H_k and L_k.

Let G and F be the minimal nonnegative solution of (1.1) and

$$F = A_2 + A_1 F + A_0 F^2, \tag{2.1}$$

respectively. We have the following fundamental result (see [12], for example).

THEOREM 2.1. *If the QBD is positive recurrent, then G is stochastic and F is substochastic with spectral radius $\rho(F) < 1$. If the QBD is transient, then F is stochastic and G is substochastic with $\rho(G) < 1$. If the QBD is null recurrent, then G and F are both stochastic.*

It is clear that the matrix G is also the minimal nonnegative solution of $G = L_0 + H_0G^2$. Thus, we have the infinite system

$$\begin{pmatrix} W_0 & -H_0 & & 0 \\ -L_0 & I & -H_0 & \\ & -L_0 & I & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} I \\ G \\ G^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} K_0 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \tag{2.2}$$

for appropriate K_0 and W_0.

As in [2], we apply the cyclic reduction algorithm to (2.2) and get a reduced system. Multiplying both sides of the reduced system by a proper block diagonal matrix, we get an infinite system with the same structure as (2.2), but with G replaced by G^2. After repeated application of the cyclic reduction algorithm and the block diagonal scaling, we obtain for each $n \geq 0$

$$\begin{pmatrix} W_n & -H_n & & 0 \\ -L_n & I & -H_n & \\ & -L_n & I & \ddots \\ 0 & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} I \\ G^{2^n} \\ G^{2\cdot 2^n} \\ \vdots \end{pmatrix} = \begin{pmatrix} K_n \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \tag{2.3}$$

where H_n and L_n are as in the LR algorithm.

From (2.3), we have

$$-L_n + G^{2^n} - H_nG^{2\cdot 2^n} = 0 \tag{2.4}$$

for each $n \geq 0$. Therefore,

$$G = L_0 + H_0G^2 = L_0 + H_0(L_1 + H_1G^4) = L_0 + H_0L_1 + H_0H_1(L_2 + H_2G^8) = \cdots.$$

In general,

$$G = G_k + \Big(\prod_{0\leq i\leq k} H_i\Big)G^{2\cdot 2^k}, \tag{2.5}$$

where G_k is as in the LR algorithm.

It is clear that the matrix F is also the minimal nonnegative solution of $F = H_0 + L_0F^2$. By repeating the whole process leading to (2.4), we get for each $n \geq 0$

$$-H_n + F^{2^n} - L_nF^{2\cdot 2^n} = 0. \tag{2.6}$$

From (2.6), we can see that $H_n \leq F^{2^n}$ for each $n \geq 0$. Thus, we have by (2.5)

$$0 \leq G - G_k \leq F^{2\cdot 2^k - 1}G^{2\cdot 2^k}. \tag{2.7}$$

Therefore, if the QBD is positive recurrent or transient, the quadratic convergence of $\{G_k\}$ is an immediate consequence of Theorem 2.1. In this situation, it is also very

easy to determine the limits of the sequences $\{H_k\}$ and $\{L_k\}$. The following result is necessary.

THEOREM 2.2. *Let Q be a stochastic matrix. If Q^r has a positive column for some integer $r \geq 1$, then there is a unique vector $q \geq 0$ such that $q^T Q = q^T$ and $q^T e = 1$ (the vector q is called the stationary probability vector of Q). Moreover, there are constants $K > 0$ and $\beta \in (0,1)$ such that*

$$\|Q^n - eq^T\| \leq K\beta^n$$

for all $n \geq 0$. In particular, $\lim_{n\to\infty} Q^n = eq^T$.

For a proof of this result, see [6]. See also [7] for the special case when Q^r is positive for some integer $r \geq 1$. Obviously, the condition that Q^r has a positive column for some $r \geq 1$ is necessary for $\lim_{n\to\infty} Q^n = eq^T$.

If the QBD is positive recurrent, then G is stochastic and $\rho(F) < 1$. Assuming that G^p has a positive column for some integer $p \geq 1$, we get from (2.4) and (2.6) that $\lim_{n\to\infty} H_n = 0$ and $\lim_{n\to\infty} L_n = eg^T$, where g is the stationary probability vector of G. If the QBD is transient, then $\rho(G) < 1$ and F is stochastic. Assuming that F^p has a positive column for some integer $p \geq 1$, we have $\lim_{n\to\infty} L_n = 0$ and $\lim_{n\to\infty} H_n = ef^T$, where f is the stationary probability vector of F. The limits of $\{H_n\}$ and $\{L_n\}$ were determined in [11] in a different way.

If the QBD is null recurrent, then $\rho(G) = 1$ and $\rho(F) = 1$. In this case, (2.7) tells us nothing about the convergence rate of the LR algorithm. It is also much more difficult to determine the limits of the sequences $\{H_n\}$ and $\{L_n\}$. These issues will be resolved in the next section.

3. Convergence rate of the LR algorithm for the null recurrent case. We start with an algebraic proof of a basic result about the LR algorithm. A probabilistic proof was given in [11].

LEMMA 3.1. *For each $k \geq 0$, $(H_k + L_k)e = e$.*

Proof. First, $(H_0+L_0)e = (I-A_1)^{-1}(A_0+A_2)e = e$. Assuming that $(H_k+L_k)e = e\,(k \geq 0)$, we have $(H_k+L_k)^2 e = e$. So, $(I-H_kL_k-L_kH_k)e = (H_k^2+L_k^2)e$. Therefore, $(H_{k+1}+L_{k+1})e = (I - H_kL_k - L_kH_k)^{-1}(H_k^2 + L_k^2)e = e$. We have thus proved the result by induction. ∎

In the above proof, we have used the fact that the sequences $\{H_k\}$ and $\{L_k\}$ are well defined (i.e., the matrices $I - H_kL_k - L_kH_k$ are nonsingular).

It is noted in [11] that when the QBD is null recurrent, it is not true in general that one of the two sequences $\{H_k\}$ and $\{L_k\}$ converges to 0. Our next result shows that neither of the two sequences can converge to 0 for null recurrent QBDs.

LEMMA 3.2. *For the null recurrent QBD, there is a sequence $\{\alpha_k\}$ such that for all $k \geq 0$, $\alpha_k \geq 0$, $\alpha_k^T e = 1$, $\alpha_k^T(H_k + L_k) = \alpha_k^T$, and $\alpha_k^T H_k e = \alpha_k^T L_k e = \frac{1}{2}$.*

Proof. Recall that α is the stationary probability vector of $A_0 + A_1 + A_2$. So,

$$\alpha^T(I - A_1) = \alpha^T(A_0 + A_2) = \alpha^T(I - A_1)(H_0 + L_0).$$

Let $\hat{\alpha}^T = \alpha^T(I - A_1)$. Since $\alpha > 0$ and $A_0 \neq 0$, $\hat{\alpha}^T = \alpha^T(A_0 + A_2) \geq 0$ and $c_0 = \hat{\alpha}^T e > 0$. Since the QBD is null recurrent, we have $\alpha^T A_2 e = \alpha^T A_0 e$ and thus $\hat{\alpha}^T H_0 e = \hat{\alpha}^T L_0 e$. Let $\alpha_0 = \hat{\alpha}/c_0$. It is clear that α_0 has all the properties in the lemma, noting that $\alpha_0^T H_0 e + \alpha_0^T L_0 e = \alpha_0^T e = 1$.

Assuming that an α_i $(i \geq 0)$, with all the properties in the lemma, has been found, we are going to find an α_{i+1} satisfying these properties.

Since

$$\alpha_i^T = \alpha_i^T(H_i + L_i) = \alpha_i^T(H_i + L_i)^2 = \alpha_i^T(H_i^2 + L_i^2 + H_iL_i + L_iH_i),$$

we have

$$\alpha_i^T(I - H_iL_i - L_iH_i) = \alpha_i^T(H_i^2 + L_i^2) = \alpha_i^T(I - H_iL_i - L_iH_i)(H_{i+1} + L_{i+1}).$$

Since $\alpha_i \neq 0$ and $I - H_iL_i - L_iH_i$ is nonsingular, $\alpha_i^T(I - H_iL_i - L_iH_i) = \alpha_i^T(H_i^2 + L_i^2) \neq 0$. Thus, $\alpha_i^T(H_i^2 + L_i^2)e > 0$, and we can define

$$\alpha_{i+1}^T = \frac{\alpha_i^T(H_i^2 + L_i^2)}{\alpha_i^T(H_i^2 + L_i^2)e} = \frac{\alpha_i^T(I - H_iL_i - L_iH_i)}{\alpha_i^T(H_i^2 + L_i^2)e}.$$

It remains to prove $\alpha_{i+1}^T H_{i+1}e = \alpha_{i+1}^T L_{i+1}e$, which is equivalent to $\alpha_i^T H_i^2 e = \alpha_i^T L_i^2 e$. Note that

$$\begin{aligned}\alpha_i^T H_i^2 e - \alpha_i^T L_i^2 e &= \alpha_i^T H_i(e - L_ie) - \alpha_i^T L_i(e - H_ie)\\ &= -\alpha_i^T H_iL_ie + \alpha_i^T L_iH_ie\\ &= -\alpha_i^T(I - L_i)L_ie + \alpha_i^T(I - H_i)H_ie\\ &= \alpha_i^T L_i^2 e - \alpha_i^T H_i^2 e.\end{aligned}$$

Thus, $\alpha_i^T H_i^2 e = \alpha_i^T L_i^2 e$. □

Remark 3.3. The result in the above lemma has also been obtained independently by Ye [19]. In [19] it is assumed that $\alpha_i^T(H_i^2 + L_i^2)e \neq 0$ for each i.

In the first version of this paper, the author used the assumption that $(H_i^2 + L_i^2)e > 0$ for each i. Without this assumption, the short argument (in the proof of the lemma) showing $\alpha_i^T(H_i^2 + L_i^2)e > 0$ for each i was pointed out by two referees.

Our further analysis will rely on Theorem 2.2. In order to apply Theorem 2.2, we make the following assumption:

(3.1) $\det(A_0 + zA_1 + z^2A_2 - zI)$ has no zeros on the unit circle other than $z = 1$.

This assumption may be verified easily when the matrices A_0, A_1, A_2 have special structures (see [13], for example). From [4] we know that in the null recurrent case, assumption (3.1) is equivalent to the assumption that $\lambda = 1$ is a simple eigenvalue of G and F and there are no other eigenvalues of G or F on the unit circle. It is easy to show that the latter assumption is in turn equivalent to the next assumption:

(3.2) G^p and F^q have each a positive column for some $p \geq 1$ and some $q \geq 1$.

Note that assumption (3.2) for G is satisfied if G_k in the LR algorithm has a positive column for some $k \geq 0$, since $G \geq G_k$. In particular, assumption (3.2) for G is satisfied if L_0 has a positive column. Similar comments can be made on assumption (3.2) for F. Since assumptions (3.1) and (3.2) are equivalent, Theorem 2.2 can be applied to G and F under assumption (3.1). We let f and g be the unique stationary probability vector of F and G, respectively.

Since $(H_k + L_k)e = e$ for all $k \geq 0$, the sequences $\{H_k\}$ and $\{L_k\}$ are bounded and hence have convergent subsequences. Let $\{H_{n_k}\}$ and $\{L_{n_k}\}$ be convergent with

$$\lim_{k\to\infty} H_{n_k} = H, \quad \lim_{k\to\infty} L_{n_k} = L.$$

Then, by (2.4) and (2.6) and Theorem 2.2,

$$-L + eg^T - Heg^T = 0, \quad -H + ef^T - Lef^T = 0.$$

Therefore, $H = af^T$ with $a = e - Le$, and $L = bg^T$ with $b = e - He$. Note that $a + b = 2e - (H + L)e = e$.

We have thus proved the following result.

LEMMA 3.4. *For the null recurrent QBD with assumption* (3.1)*, if* (H, L) *is a limit point of* $\{(H_k, L_k)\}$*, then* $H = af^T$ *and* $L = bg^T$ *with* $a \geq 0$*,* $b \geq 0$*, and* $a + b = e$*.*

To prove that the convergence of the LR algorithm is linear with rate 1/2, we will need to show that

$$\lim_{k\to\infty} H_k = \frac{1}{2}ef^T, \quad \lim_{k\to\infty} L_k = \frac{1}{2}eg^T.$$

Lemma 3.4 is only one small step towards this goal. Many other auxiliary results will be needed.

Although we are unable to show the convergence of the sequences $\{H_k\}$ and $\{L_k\}$ at the moment, it is fairly easy to show that the sequence $\{\alpha_k\}$ in Lemma 3.2 converges.

LEMMA 3.5. *For the null recurrent QBD with assumption* (3.1)*,*

$$\lim_{k\to\infty} \alpha_k = \frac{1}{2}(f + g).$$

Proof. Let α^* be any limit point of $\{\alpha_k\}$ and $\lim_{k\to\infty} \alpha_{n_k} = \alpha^*$. We will prove that $\alpha^* = \frac{1}{2}(f + g)$. We may assume without loss of generality that

$$\lim_{k\to\infty} H_{n_k} = af^T, \quad \lim_{k\to\infty} L_{n_k} = bg^T$$

for some $a, b \geq 0$ with $a + b = e$. By taking limits in

$$\alpha_{n_k}^T = \alpha_{n_k}^T(H_{n_k} + L_{n_k}), \quad \alpha_{n_k}^T H_{n_k} e = \alpha_{n_k}^T L_{n_k} e,$$

we get

$$(\alpha^*)^T = (\alpha^*)^T(af^T + bg^T), \quad (\alpha^*)^T a = (\alpha^*)^T b.$$

Thus, $(\alpha^*)^T a = (\alpha^*)^T(e - a) = 1 - (\alpha^*)^T a$. So, $(\alpha^*)^T a = (\alpha^*)^T b = 1/2$ and $(\alpha^*)^T = \frac{1}{2}(f^T + g^T)$, or $\alpha^* = \frac{1}{2}(f + g)$. □

As we have already seen, in the null recurrent case, the two equations (2.4) and (2.6) are not sufficient to determine the convergence of the sequences $\{H_n\}$ and $\{L_n\}$. We have to seek additional information from the recursions for the sequences $\{H_n\}$ and $\{L_n\}$. The next result is one such finding.

LEMMA 3.6. *For the null recurrent QBD with assumption* (3.1)*, if*

$$\lim_{k\to\infty} H_{n_k} = af^T, \quad \lim_{k\to\infty} L_{n_k} = bg^T,$$

and $g^T a \neq 1$*, then*

$$\lim_{k\to\infty} H_{n_k+1} = \hat{a}f^T, \quad \lim_{k\to\infty} L_{n_k+1} = \hat{b}g^T,$$

with

$$\hat{a} = \frac{1+g^Ta}{1+2g^Ta}a + \frac{g^Ta}{1+2g^Ta}b, \quad \hat{b} = \frac{g^Ta}{1+2g^Ta}a + \frac{1+g^Ta}{1+2g^Ta}b.$$

Proof. Let $(\tilde{a}f^T, \tilde{b}g^T)$ be any limit point of $\{(H_{n_k+1}, L_{n_k+1})\}$ and let

$$\lim_{k\to\infty} H_{n_{s_k}+1} = \tilde{a}f^T, \quad \lim_{k\to\infty} L_{n_{s_k}+1} = \tilde{b}g^T.$$

Since

$$(I - H_{n_{s_k}}L_{n_{s_k}} - L_{n_{s_k}}H_{n_{s_k}})H_{n_{s_k}+1} = (H_{n_{s_k}})^2,$$

we get, by letting $k \to \infty$,

$$(I - af^Tbg^T - bg^Taf^T)\tilde{a}f^T = af^Taf^T.$$

Postmultiplying the above equality by e gives

$$\tilde{a} = (f^Ta + f^Tbg^T\tilde{a})a + (g^Taf^T\tilde{a})b \equiv \lambda a + \mu b.$$

By Lemma 3.2,

$$\alpha_{n_k}^T H_{n_k}e = \alpha_{n_k}^T L_{n_k}e = \frac{1}{2}, \quad (\alpha_{n_{s_k}+1})^T H_{n_{s_k}+1}e = (\alpha_{n_{s_k}+1})^T L_{n_{s_k}+1}e = \frac{1}{2}.$$

By taking limits in the above identities and using Lemma 3.5, we have

$$(f^T+g^T)a = (f^T+g^T)b = (f^T+g^T)\tilde{a} = (f^T+g^T)\tilde{b} = 1.$$

So, $f^Ta = 1 - g^Ta = g^Te - g^Ta = g^Tb$. Similarly, $f^Tb = g^Ta$, $f^T\tilde{a} = g^T\tilde{b}$, $f^T\tilde{b} = g^T\tilde{a}$. Thus,

$$\lambda + \mu = f^Ta + f^Tbg^T\tilde{a} + g^Taf^T\tilde{a} = f^Ta + f^Tb(g^T\tilde{a} + f^T\tilde{a}) = f^Ta + f^Tb = f^Te = 1.$$

Now,

$$\begin{aligned}\mu &= g^Taf^T\tilde{a} = g^Taf^T(\lambda a + \mu b)\\ &= (1-\mu)g^Taf^Ta + \mu g^Taf^Tb = (1-\mu)g^Ta(1-g^Ta) + \mu(g^Ta)^2.\end{aligned}$$

Thus,

$$(1+2g^Ta)(1-g^Ta)\mu = g^Ta(1-g^Ta).$$

Since $g^Ta \neq 1$, we have $\mu = g^Ta/(1+2g^Ta)$ and $\lambda = 1-\mu = (1+g^Ta)/(1+2g^Ta)$. So,

$$\tilde{a} = \frac{1+g^Ta}{1+2g^Ta}a + \frac{g^Ta}{1+2g^Ta}b,$$

and

$$\tilde{b} = e - \tilde{a} = a + b - \tilde{a} = \frac{g^Ta}{1+2g^Ta}a + \frac{1+g^Ta}{1+2g^Ta}b.$$

The proof is completed since the limit point is uniquely determined by a and b. ∎

We can now move a little closer to our goal.

LEMMA 3.7. *For the null recurrent QBD with assumptions* (3.1) *and*

(3.3) *Each limit point* af^T *of the sequence* $\{H_n\}$ *is such that* $0 < g^T a < 1$,

the sequence $\{(H_n, L_n)\}$ *has a limit point* $(\frac{1}{2}ef^T, \frac{1}{2}eg^T)$.

Proof. Take any subsequence $\{(H_{n_k}, L_{n_k})\}$ such that

$$\lim_{k\to\infty}(H_{n_k}, L_{n_k}) = (a_0 f^T, b_0 g^T).$$

By the previous lemma, for each integer $r \geq 1$,

$$\lim_{k\to\infty}(H_{n_k+r}, L_{n_k+r}) = (a_r f^T, b_r g^T),$$

where

$$a_{k+1} = \frac{1 + g^T a_k}{1 + 2g^T a_k} a_k + \frac{g^T a_k}{1 + 2g^T a_k} b_k, \tag{3.4}$$

and $b_{k+1} = e - a_{k+1}$ for each integer $k \geq 0$. Let $p_k = g^T a_k$. We have by (3.4)

$$p_{k+1} = \frac{(1 + p_k)p_k}{1 + 2p_k} + \frac{p_k(1 - p_k)}{1 + 2p_k} = \frac{2p_k}{1 + 2p_k}.$$

Since $p_0 = g^T a_0 > 0$ by assumption, it is easy to show that $\lim_{k\to\infty} p_k = \frac{1}{2}$. By (3.4) we have

$$a_{k+1} = \frac{g^T a_k}{1 + 2g^T a_k} e + \frac{1}{1 + 2g^T a_k} a_k,$$

which can be rewritten as

$$a_{k+1} - \frac{1}{2}e = \frac{1}{1 + 2g^T a_k}\left(a_k - \frac{1}{2}e\right).$$

Since $\lim_{k\to\infty} g^T a_k = \frac{1}{2}$, we have

$$\left|a_{k+1} - \frac{1}{2}e\right| \leq \frac{2}{3}\left|a_k - \frac{1}{2}e\right|$$

for k large enough. Thus

$$\lim_{r\to\infty} a_r = \lim_{r\to\infty} b_r = \frac{1}{2}e.$$

Therefore, we can find a subsequence $\{(H_{m_k}, L_{m_k})\}$ such that

$$\lim_{k\to\infty}(H_{m_k}, L_{m_k}) = \left(\frac{1}{2}ef^T, \frac{1}{2}eg^T\right).$$

This completes the proof. ∎

The next result is quite straightforward.

LEMMA 3.8. *Let the relation between (H_{k+1}, L_{k+1}) and (H_k, L_k) in the LR algorithm be denoted by*

$$(H_{k+1}, L_{k+1}) = \mathcal{T}(H_k, L_k).$$

Then $(\frac{1}{2}ef^T, \frac{1}{2}eg^T)$ is a fixed point of $\mathcal{T}$.

Proof. It is easy to verify that

$$\left(I - \frac{1}{2}ef^T\frac{1}{2}eg^T - \frac{1}{2}eg^T\frac{1}{2}ef^T\right)\frac{1}{2}ef^T = \left(\frac{1}{2}ef^T\right)^2,$$
$$\left(I - \frac{1}{2}ef^T\frac{1}{2}eg^T - \frac{1}{2}eg^T\frac{1}{2}ef^T\right)\frac{1}{2}eg^T = \left(\frac{1}{2}eg^T\right)^2.$$

The result follows since

$$M \equiv I - \frac{1}{2}ef^T\frac{1}{2}eg^T - \frac{1}{2}eg^T\frac{1}{2}ef^T = I - \frac{1}{4}e(f^T + g^T)$$

is a nonsingular M-matrix (note that $Me = e/2$). □

Thus, we have shown that the sequence $\{(H_n, L_n)\}$ defined by $(H_{k+1}, L_{k+1}) = \mathcal{T}(H_k, L_k)\,(k \geq 0)$ has a limit point $(\frac{1}{2}ef^T, \frac{1}{2}eg^T)$ that is a fixed point of $\mathcal{T}$. By a theorem on general fixed-point iterations (see [8, p. 21], for example), we can conclude that the whole sequence $\{(H_n, L_n)\}$ converges to this fixed point *if* the spectral radius of the Fréchet derivative of the operator $\mathcal{T}$ at the fixed point is less than 1. But, unfortunately, the spectral radius is *not* less than 1 in our case. (The spectral radius is equal to 4 when the matrices A_0, A_1, A_2 are 1×1.) However, the sequence $\{(H_n, L_n)\}$ can still converge since (H_n, L_n) may approach $(\frac{1}{2}ef^T, \frac{1}{2}eg^T)$ in a special way. A delicate analysis for the error $(H_n - \frac{1}{2}ef^T, L_n - \frac{1}{2}eg^T)$ is necessary.

For notational convenience, let $H = \frac{1}{2}ef^T$ and $L = \frac{1}{2}eg^T$. It is easy to see that

$$H^2 = LH = \frac{1}{2}H, \quad L^2 = HL = \frac{1}{2}L.$$

We start with expressing $H_{k+1} - H$ in terms of $H_k - H$ and $L_k - L$:

$$\begin{aligned}
H_{k+1} - H &= (I - H_kL_k - L_kH_k)^{-1}H_k^2 - (I - HL - LH)^{-1}H^2 \\
&= (I - H_kL_k - L_kH_k)^{-1}(H_k^2 - H^2) \\
&\quad + \big((I - H_kL_k - L_kH_k)^{-1} - (I - HL - LH)^{-1}\big)H^2 \\
&= (I - H_kL_k - L_kH_k)^{-1}(H_k^2 - H^2) + (I - H_kL_k - L_kH_k)^{-1} \\
&\quad \big((I - HL - LH) - (I - H_kL_k - L_kH_k)\big)(I - HL - LH)^{-1}H^2 \\
&= (I - H_kL_k - L_kH_k)^{-1}\big(H_k^2 - H^2 + (H_kL_k - HL + L_kH_k - LH)H\big) \\
&= (I - H_kL_k - L_kH_k)^{-1}\big(H_k(H_k - H) + (H_k - H)H \\
&\quad + (H_k(L_k - L) + (H_k - H)L + L_k(H_k - H) + (L_k - L)H)H\big).
\end{aligned}$$

To simplify the expression, observe that

$$(H_k - H + L_k - L)H = \frac{1}{2}\big((H_k + L_k)e - (H + L)e\big)f^T = 0.$$

Thus,

$$(L_k - L)H = -(H_k - H)H$$

and

$$\big((H_k - H)L + (L_k - L)H\big)H = \frac{1}{2}(H_k - H + L_k - L)H = 0.$$

Therefore,

$$\begin{aligned} H_{k+1} - H = (I - H_kL_k - L_kH_k)^{-1}\big(&H_k(H_k - H) + (H_k - H)H \\ &-H_k(H_k - H)H + L_k(H_k - H)H\big). \end{aligned}$$

Similarly, we can get

$$\begin{aligned} L_{k+1} - L = (I - H_kL_k - L_kH_k)^{-1}\big(&L_k(L_k - L) + (L_k - L)L \\ &-L_k(L_k - L)L + H_k(L_k - L)L\big). \end{aligned}$$

Now, for any $\epsilon \in (0, \frac{1}{4})$, we can find $\delta > 0$ such that whenever $\|H_k - H\|_\infty \le \delta$ and $\|L_k - L\|_\infty \le \delta$,

$$\begin{aligned} H_{k+1} - H = (I - HL - LH)^{-1}\big(&H(H_k - H) + (H_k - H)H \\ &-H(H_k - H)H + L(H_k - H)H\big) + W_k \end{aligned} \tag{3.5}$$

with $\|W_k\|_\infty \le \epsilon\|H_k - H\|_\infty$, and

$$\begin{aligned} L_{k+1} - L = (I - HL - LH)^{-1}\big(&L(L_k - L) + (L_k - L)L \\ &-L(L_k - L)L + H(L_k - L)L\big) + Z_k \end{aligned}$$

with $\|Z_k\|_\infty \le \epsilon\|L_k - L\|_\infty$.

To get rid of the inverse in (3.5), we use

$$(I - HL - LH)^{-1} = \left(I - \frac{1}{2}(H + L)\right)^{-1} = I + \frac{1}{2}(H + L) + \frac{1}{2^2}(H + L)^2 + \cdots. \tag{3.6}$$

Since

$$\begin{aligned} &(H + L)\big(H(H_k - H) + (H_k - H)H - H(H_k - H)H + L(H_k - H)H\big) \\ &= H(H_k - H) + (H + L)(H_k - H)H - H(H_k - H)H + L(H_k - H)H \\ &= H(H_k - H) + 2L(H_k - H)H \end{aligned}$$

and

$$(H + L)^i\big(H(H_k - H) + 2L(H_k - H)H\big) = H(H_k - H) + 2L(H_k - H)H$$

for all $i \ge 1$, we get by (3.5) and (3.6) that

$$\begin{aligned} H_{k+1} - H &= H(H_k - H) + (H_k - H)H - H(H_k - H)H + L(H_k - H)H \\ &\quad + H(H_k - H) + 2L(H_k - H)H + W_k \\ &= (H_k - H)H + 2H(H_k - H) - H(H_k - H)H + 3L(H_k - H)H + W_k. \end{aligned}$$

Similarly,

$$L_{k+1} - L = (L_k - L)L + 2L(L_k - L) - L(L_k - L)L + 3H(L_k - L)L + Z_k.$$

For the scalar case, $H = L = \frac{1}{2}$. So, the estimate for $H_{k+1} - H$ becomes $H_{k+1} - H \approx 2(H_k - H)$. If we could replace $3L(H_k - H)H$ by $-3H(H_k - H)H$ in the estimate, we would have $H_{k+1} - H \approx \frac{1}{2}(H_k - H)$ instead. Thus, we should try to show that $3(H + L)(H_k - H)H$ is "small" in the general case. Let $\alpha^* = (f + g)/2$. We have

$$3(H + L)(H_k - H)H = \frac{3}{2}e(\alpha^*)^T(H_k - H)ef^T = \frac{3}{2}e(\alpha^* - \alpha_k)^T(H_k - H)ef^T,$$

since $\alpha_k^T(H_k - H)e = \frac{1}{2} - \alpha_k^T(\frac{1}{2}e) = 0$ by Lemma 3.2. Similarly,

$$3(H + L)(L_k - L)L = \frac{3}{2}e(\alpha^* - \alpha_k)^T(L_k - L)eg^T.$$

Since $\lim \alpha_k = \alpha^*$ by Lemma 3.5, we can find integer k_1 such that for all $k \geq k_1$,

$$3(H + L)(H_k - H)H = P_k, \quad 3(H + L)(L_k - L)L = Q_k$$

with $\|P_k\|_\infty \leq \epsilon\|H_k - H\|_\infty$ and $\|Q_k\|_\infty \leq \epsilon\|L_k - L\|_\infty$. Now we have

$$\begin{aligned} H_{k+1} - H &= (H_k - H)H + 2H(H_k - H) - 4H(H_k - H)H + P_k + W_k \\ &= \frac{1}{2}(H_k - H) - \frac{1}{2}(I - 4H)(H_k - H)(I - 2H) + P_k + W_k \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} L_{k+1} - L &= (L_k - L)L + 2L(L_k - L) - 4L(L_k - L)L + Q_k + Z_k \\ &= \frac{1}{2}(L_k - L) - \frac{1}{2}(I - 4L)(L_k - L)(I - 2L) + Q_k + Z_k. \end{aligned} \tag{3.8}$$

Next we will estimate the term $(H_k - H)(I - 2H)$ in (3.7) and the term $(L_k - L)(I - 2L)$ in (3.8). By (2.4) and (2.6), we have

$$H_k(I - G^{2\cdot 2^k}F^{2\cdot 2^k}) = F^{2^k} - G^{2^k}F^{2\cdot 2^k}, \quad L_k(I - F^{2\cdot 2^k}G^{2\cdot 2^k}) = G^{2^k} - F^{2^k}G^{2\cdot 2^k}.$$

Now,

$$\begin{aligned} (H_k - H)(I - 2H) &= H_k(I - ef^T) \\ &= F^{2^k} - G^{2^k}F^{2\cdot 2^k} + H_k(G^{2\cdot 2^k}F^{2\cdot 2^k} - ef^T) \\ &= F^{2^k} - ef^T - (G^{2^k} - eg^T)F^{2\cdot 2^k} - eg^T(F^{2\cdot 2^k} - ef^T) \\ &\quad + H_k\big((G^{2\cdot 2^k} - eg^T)F^{2\cdot 2^k} + eg^T(F^{2\cdot 2^k} - ef^T)\big). \end{aligned}$$

Similarly,

$$\begin{aligned} (L_k - L)(I - 2L) = G^{2^k} - eg^T - (F^{2^k} - ef^T)G^{2\cdot 2^k} - ef^T(G^{2\cdot 2^k} - eg^T) \\ + L_k\big((F^{2\cdot 2^k} - ef^T)G^{2\cdot 2^k} + ef^T(G^{2\cdot 2^k} - eg^T)\big). \end{aligned}$$

By Theorem 2.2, there are constants $C_1 > 0$ and $\beta \in (0, 1)$ such that

$$\|(H_k - H)(I - 2H)\|_\infty \leq C_1\beta^{2^k}, \quad \|(L_k - L)(I - 2L)\|_\infty \leq C_1\beta^{2^k}$$

for all $k \geq 0$. Now, by (3.7) and (3.8), we have

$$\|H_{k+1} - H\|_\infty \leq \left(\frac{1}{2} + 2\epsilon\right)\|H_k - H\|_\infty + C_2\beta^{2^k}, \tag{3.9}$$

$$\|L_{k+1} - L\|_\infty \leq \left(\frac{1}{2} + 2\epsilon\right)\|L_k - L\|_\infty + C_2\beta^{2^k} \tag{3.10}$$

for any $k \geq k_1$ with $\|H_k - H\|_\infty < \delta$ and $\|L_k - L\|_\infty < \delta$. Let $r = \frac{1}{2} + 2\epsilon < 1$. Since (H, L) is a limit point of $\{(H_k, L_k)\}$ by Lemma 3.7, we can find $l \geq k_1$ such that $\|H_l - H\|_\infty < \delta$, $\|L_l - L\|_\infty < \delta$, $r\delta + C_2\beta^{2^l} < \delta$, and $\beta^{2^l} \leq r$. Now, it is clear that (3.9) and (3.10) are valid for all $k \geq l$ and that $\beta^{2^{l+j-1}} \leq r^j$ for all $j \geq 0$. Thus, we can obtain for any $m \geq 1$ that

$$\begin{aligned}\|H_{l+m} - H\|_\infty &\leq r^m \|H_l - H\|_\infty + C_2(r^{m-1}\beta^{2^l} + r^{m-2}\beta^{2^{l+1}} + \cdots + \beta^{2^{l+m-1}}) \\ &\leq r^m \|H_l - H\|_\infty + C_2 m r^m,\end{aligned}$$

and that

$$\|L_{l+m} - L\|_\infty \leq r^m \|L_l - L\|_\infty + C_2 m r^m.$$

Therefore, $\lim_{k\to\infty} H_k = H$ and $\lim_{k\to\infty} L_k = L$. Moreover, since $\epsilon > 0$ can be arbitrarily small, we also have

$$\limsup_{k\to\infty} \sqrt[k]{\|H_k - H\|_\infty} \leq \frac{1}{2}, \quad \limsup_{k\to\infty} \sqrt[k]{\|L_k - L\|} \leq \frac{1}{2}.$$

In summary, we have proved the following result.

THEOREM 3.9. *For the null recurrent QBD with assumptions* (3.1) *and* (3.3)*, we have*

$$\lim_{k\to\infty} H_k = \frac{1}{2} e f^T, \quad \lim_{k\to\infty} L_k = \frac{1}{2} e g^T.$$

It is clear that assumption (3.3) is necessary for the conclusion of the above theorem. Since the assumption cannot be verified directly, we will give a sufficient condition that is easier to verify.

PROPOSITION 3.10. *Let the components of f and g be f_i and g_i $(i = 1, 2, \ldots, m)$, respectively, and let*

$$S_f = \{i \,|\, 1 \leq i \leq m, f_i = 0\}, \quad S_g = \{i \,|\, 1 \leq i \leq m, g_i = 0\}.$$

If assumption (3.1) *and the assumption that*

$$S_f \subset S_g \text{ or } S_g \subset S_f \tag{3.11}$$

are satisfied, then assumption (3.3) *is also satisfied.*

Proof. Let

$$\lim_{k\to\infty} H_{n_k} = a f^T, \quad \lim_{k\to\infty} L_{n_k} = b g^T.$$

It is shown in the proof of Lemma 3.6 that

$$(f^T + g^T)a = (f^T + g^T)b = 1, \quad f^T a = g^T b, \quad f^T b = g^T a.$$

If $g^T a = 1$, then $g^T b = f^T a = 0$. By assumption (3.11), we would have $(f^T + g^T)a = 0$ or $(f^T + g^T)b = 0$, which is a contradiction. Similarly, we get a contradiction if $g^T a = 0$. □

Remark 3.11. Assumption (3.11) is certainly satisfied if one of F and G is irreducible (in particular, if one of H_0 and L_0 is irreducible) since one of S_f and S_g is an empty set in this case.

We are now ready to determine the convergence rate of the LR algorithm for the null recurrent case. Recall that for the sequence $\{G_k\}$ generated by the LR algorithm,

$$G - G_k = H_0 H_1 \cdots H_k G^{2^{k+1}}. \tag{3.12}$$

PROPOSITION 3.12. *For each $k \geq 0$, $H_0 H_1 \cdots H_k \neq 0$.*

Proof. Equation (8.47) in [12] states that $(H_0 H_1 \cdots H_k)_{ij}$ is the probability of first passage to the state $(2^{k+1}, j)$ before any of the states $(0, \cdot)$ starting from $(1, i)$. If all of these entries were equal to 0, it would be impossible to reach the states $(2^{k+1}, \cdot)$ and the transition probability matrix P would not be irreducible. ∎

Remark 3.13. The above proof was provided by a referee. In the first version of the paper, the author gave the statement in Proposition 3.12 as an assumption.

The next theorem is our main result. It shows that the sequence $\{G_k\}$ converges to the minimal nonnegative solution of (1.1) at precisely the rate of 1/2.

THEOREM 3.14. *For the null recurrent QBD with assumptions* (3.1) *and* (3.3),

$$\lim_{k\to\infty} \sqrt[k]{\|G_k - G\|_\infty} = \frac{1}{2}.$$

Proof. Since $\lim_{k\to\infty} H_k = \frac{1}{2} e f^T$, we have $\lim_{k\to\infty} H_k e = \frac{1}{2} e$. Therefore, for any $\epsilon \in (0, \frac{1}{2})$, we can find an integer k_0 such that $(\frac{1}{2} - \epsilon)e \leq H_k e \leq (\frac{1}{2} + \epsilon)e$ for all $k > k_0$.

Note that, by (3.12),

$$\|G - G_k\|_\infty = \|(G - G_k)e\|_\infty = \|H_0 \cdots H_{k_0} H_{k_0+1} \cdots H_k e\|_\infty$$

for $k > k_0$. Thus,

$$\left(\frac{1}{2} - \epsilon\right)^{k-k_0} \|H_0 \cdots H_{k_0} e\|_\infty \leq \|G - G_k\|_\infty \leq \left(\frac{1}{2} + \epsilon\right)^{k-k_0} \|H_0 \cdots H_{k_0} e\|_\infty.$$

In view of Proposition 3.12, it follows readily that

$$\limsup_{k\to\infty} \sqrt[k]{\|G_k - G\|_\infty} \leq \frac{1}{2} + \epsilon, \quad \liminf_{k\to\infty} \sqrt[k]{\|G_k - G\|_\infty} \geq \frac{1}{2} - \epsilon.$$

Since ϵ can be arbitrarily small, we have $\lim_{k\to\infty} \sqrt[k]{\|G_k - G\|_\infty} = \frac{1}{2}$. ∎

4. Improvement of the approximate solution in the null recurrent case. By (3.12) and Theorem 3.9, it is easy to get a much better approximation to the matrix G from the sequence $\{G_k\}$ generated by the LR algorithm. In fact, we have by (3.12)

$$G - G_k - 2(G - G_{k+1}) = H_0 \cdots H_k (G^{2^{k+1}} - G^{2^{k+2}}) + H_0 \cdots H_{k-1}(H_k - 2H_k H_{k+1}) G^{2^{k+2}}.$$

The first term converges to zero quadratically by Theorem 2.2 since $G^{2^{k+1}} - G^{2^{k+2}} = (G^{2^{k+1}} - eg^T) - (G^{2^{k+2}} - eg^T)$. The second term is also much smaller than $G - G_{k+1}$ since $\lim_{k\to\infty}(H_k - 2H_k H_{k+1}) = 0$ and $\lim_{k\to\infty}(H_k H_{k+1}) = \frac{1}{4} e f^T$ by Theorem 3.9. Therefore, $\tilde{G}_{k+1} = 2G_{k+1} - G_k = G_{k+1} + (G_{k+1} - G_k)$ can be a much better approximation to G in the null recurrent case. Of course, improvements may also be achieved for nearly null recurrent QBDs by using the above strategy.

5. Examples. In this section, we will present a few examples to illustrate the theoretical results in section 3 and the simple strategy described in section 4 for the improvement of the approximate solution. For all examples, assumption (3.1) is checked through the equivalent assumption (3.2).

Example 5.1. Consider (1.1) with

$$A_0 = \begin{pmatrix} 0.6 & 0.4 & 0 \\ 0.1 & 0 & 0.1 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0.2 & 0 & 0.1 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0.26 & 0 & 0.24 \\ 0.2 & 0 & 0.8 \end{pmatrix}.$$

It is easy to verify that the corresponding QBD is null recurrent. We also find that $G_1 = L_0 + H_0L_1$ is irreducible and has a positive column and that $F_1 = H_0 + L_0H_1$ has a positive column. Since $G \geq G_1$ and $F \geq F_1$, assumptions (3.1) and (3.11) are satisfied. By Proposition 3.10, assumption (3.3) is also satisfied.

For this example, the exact minimal nonnegative solutions of (1.1) and (2.1) can be found to be

$$G = \frac{1}{2300}\begin{pmatrix} 1380 & 920 & 0 \\ 1320 & 680 & 300 \\ 1357 & 828 & 115 \end{pmatrix}, \quad F = \frac{1}{25}\begin{pmatrix} 5 & 0 & 20 \\ 9 & 0 & 16 \\ 5 & 0 & 20 \end{pmatrix}.$$

Accordingly, we have

$$g^T = (1431, 874, 120)/2425, \quad f^T = (1, 0, 4)/5.$$

For the matrices H_{18} and L_{18}, found by the LR algorithm using double precision, we have

$$H_{18} - \frac{1}{2}ef^T = 10^{-5}\begin{pmatrix} -0.0916 & 0 & -0.3665 \\ 0.0037 & 0 & 0.0149 \\ 0.0991 & 0 & 0.3964 \end{pmatrix},$$

$$L_{18} - \frac{1}{2}eg^T = 10^{-5}\begin{pmatrix} 0.3091 & 0.1888 & 0.0259 \\ 0.0277 & 0.0169 & 0.0023 \\ -0.2537 & -0.1549 & -0.0213 \end{pmatrix}.$$

Note that H_{18} and L_{18} are already very close to $\frac{1}{2}ef^T$ and $\frac{1}{2}eg^T$, respectively. We also find that for the matrices G_k computed by the LR algorithm,

$$G - G_{17} = 10^{-5}\begin{pmatrix} 0 & 0 & 0 \\ 0.474676 & 0.289914 & 0.039805 \\ 1.091754 & 0.666802 & 0.091552 \end{pmatrix}$$

and

$$G - G_{18} = 10^{-5}\begin{pmatrix} 0 & 0 & 0 \\ 0.237339 & 0.144957 & 0.019903 \\ 0.545879 & 0.333402 & 0.045776 \end{pmatrix}.$$

Note that $G - G_{18} \approx \frac{1}{2}(G - G_{17})$. For $\tilde{G}_{18} = 2G_{18} - G_{17}$, we have

$$G - \tilde{G}_{18} = 10^{-10}\begin{pmatrix} 0 & 0 & 0 \\ 0.1959 & 0.1196 & 0.0164 \\ 0.4505 & 0.2752 & 0.0378 \end{pmatrix}.$$

So, $\tilde{G}_{18}$ is a much better approximation for G.

For the next example, assumption (3.11) is satisfied even though neither of S_f and S_g is empty.

Example 5.2. Consider (1.1) with

$$A_0 = \begin{pmatrix} 0 & 0.5 \\ 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0.5 \\ 0 & 0 \end{pmatrix}.$$

The corresponding QBD is clearly null recurrent. Since

$$H_0 = L_0 = \begin{pmatrix} 0 & 0.5 \\ 0 & 0.5 \end{pmatrix}$$

for this example, assumptions (3.1) and (3.11) are satisfied. It is easy to find that

$$G = F = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

So, we actually have $S_f = S_g = \{1\}$. For this example, we have $H_k = \frac{1}{2}ef^T$ and $L_k = \frac{1}{2}eg^T$ for all $k \geq 0$. We also have for each $k \geq 0$

$$G_k = \begin{pmatrix} 0 & 1 - 1/2^{k+1} \\ 0 & 1 - 1/2^{k+1} \end{pmatrix}.$$

So, $\{G_k\}$ converges to G linearly with rate 1/2 and $\tilde{G}_k = 2G_k - G_{k-1} = G$ for all $k \geq 1$.

We can also find examples for which (3.11) is not satisfied.

Example 5.3. Consider (1.1) with

$$A_0 = \begin{pmatrix} 0.25 & 0 \\ 0.25 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0.25 \\ 0 & 0.25 \end{pmatrix}.$$

The corresponding QBD is clearly null recurrent. Since

$$H_0 = \begin{pmatrix} 0 & 0.5 \\ 0 & 0.5 \end{pmatrix}, \quad L_0 = \begin{pmatrix} 0.5 & 0 \\ 0.5 & 0 \end{pmatrix}$$

for this example, assumption (3.1) is satisfied. It is easy to find that

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad F = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

So, we have $S_f = \{1\}$ and $S_g = \{2\}$. Therefore, assumption (3.11) is not satisfied. However, the conclusions in our main results in section 3 still hold. In fact, we have $H_k = \frac{1}{2}ef^T$ and $L_k = \frac{1}{2}eg^T$ for all $k \geq 0$. We also have for each $k \geq 0$

$$G_k = \begin{pmatrix} 1 - 1/2^{k+1} & 0 \\ 1 - 1/2^{k+1} & 0 \end{pmatrix}.$$

So, $\{G_k\}$ converges to G linearly with rate 1/2 and $\tilde{G}_k = 2G_k - G_{k-1} = G$ for all $k \geq 1$.

There are also examples for which assumption (3.1) is not satisfied. The next example was provided by a referee.

Example 5.4. Consider (1.1) with

$$A_0 = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}, \quad A_1 = 0, \quad A_2 = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}.$$

The corresponding QBD is clearly null recurrent. In this case,

$$G = F = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

So, (3.11) is true, but (3.1) is not satisfied. It is easy to find that $H_k = L_k = \frac{1}{2}I$ for each $k \geq 1$ and that

$$G_k = \begin{pmatrix} 0 & 1 - 1/2^{k+1} \\ 1 - 1/2^{k+1} & 0 \end{pmatrix}$$

for each $k \geq 0$. Thus, $\{G_k\}$ converges to G linearly with rate $1/2$ and $\tilde{G}_k = 2G_k - G_{k-1} = G$ for all $k \geq 1$.

We do not have any examples of null recurrent QBDs for which the convergence of the LR algorithm is not linear with rate $1/2$.

Acknowledgment. The author is grateful to the three referees for their very helpful comments.

REFERENCES

[1] Z.-Z. Bai, *A class of iteration methods based on the Moser formula for nonlinear equations in Markov chains*, Linear Algebra Appl., 266 (1997), pp. 219–241.
[2] D. Bini and B. Meini, *On the solution of a nonlinear matrix equation arising in queueing problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 906–926.
[3] P. Favati and B. Meini, *On functional iteration methods for solving nonlinear matrix equations arising in queueing problems*, IMA J. Numer. Anal., 19 (1999), pp. 39–49.
[4] H. R. Gail, S. L. Hantler, and B. A. Taylor, *Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains*, Adv. in Appl. Probab., 28 (1996), pp. 114–165.
[5] C.-H. Guo, *On the numerical solution of a nonlinear matrix equation in Markov chains*, Linear Algebra Appl., 288 (1999), pp. 175–186.
[6] D. J. Hartfiel, *Markov Set-Chains*, Lecture Notes in Math. 1695, Springer, Berlin, 1998.
[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[8] M. A. Krasnoselskii, G. M. Vainikko, P. P. Zabreiko, Ya. B. Rutitskii, and V. Ya. Stetsenko, *Approximate Solution of Operator Equations*, Wolters-Noordhoff, Groningen, The Netherlands, 1972.
[9] G. Latouche, *Newton's iteration for non-linear equations in Markov chains*, IMA J. Numer. Anal., 14 (1994), pp. 583–598.
[10] G. Latouche, *Algorithms for evaluating the matrix G in Markov chains of $PH/G/1$ type*, Cahiers Centre Études Rech. Opér., 36 (1994), pp. 251–258.
[11] G. Latouche and V. Ramaswami, *A logarithmic reduction algorithm for quasi-birth-death processes*, J. Appl. Probab., 30 (1993), pp. 650–674.
[12] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, SIAM, Philadelphia, PA, 1999.
[13] G. Latouche and P. G. Taylor, *Level-phase independence for $GI/M/1$-type Markov chains*, J. Appl. Probab., 37 (2000), pp. 984–998.
[14] B. Meini, *New convergence results on functional iteration techniques for the numerical solution of $M/G/1$ type Markov chains*, Numer. Math., 78 (1997), pp. 39–58.
[15] M. F. Neuts, *Moment formulas for the Markov renewal branching process*, Adv. in Appl. Probab., 8 (1976), pp. 690–711.
[16] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Johns Hopkins University Press, Baltimore, MD, 1981.

[17] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[18] Q. Ye, *High accuracy algorithms for solving nonlinear matrix equations in queueing models*, in Advances in Algorithmic Methods for Stochastic Models, Proceedings of the 3rd International Conference on Matrix Analytic Methods, G. Latouche and P. G. Taylor, eds., Notable Publications, Neshanic Station, NJ, 2000, pp. 401–415.
[19] Q. Ye, *On Latouche-Ramaswami's Logarithmic Reduction Algorithm for Quasi-Birth-and-Death Processes*, Research Report 2001-05, Department of Mathematics, University of Kentucky, Lexington, KY, 2001.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 761–798
© 2002 Society for Industrial and Applied Mathematics

MORE ACCURATE BIDIAGONAL REDUCTION FOR COMPUTING THE SINGULAR VALUE DECOMPOSITION*

JESSE L. BARLOW†

Abstract. Bidiagonal reduction is the preliminary stage for the fastest stable algorithms for computing the singular value decomposition (SVD) now available. However, the best-known error bounds on bidiagonal reduction methods on any matrix are of the form

$$A + \delta A = UBV^T,$$
$$\|\delta A\|_2 \leq \varepsilon_M f(m,n)\|A\|_2,$$

where B is bidiagonal, U and V are orthogonal, ε_M is machine precision, and $f(m,n)$ is a modestly growing function of the dimensions of A.

A preprocessing technique analyzed by Higham [*Linear Algebra Appl.*, 309 (2000), pp. 153–174] uses orthogonal factorization with column pivoting to obtain the factorization

$$A = Q \begin{pmatrix} C^T \\ 0 \end{pmatrix} P^T,$$

where Q is orthogonal, C is lower triangular, and P is permutation matrix. Bidiagonal reduction is applied to the resulting matrix C.

To do that reduction, a new Givens-based bidiagonalization algorithm is proposed that produces a bidiagonal matrix B that satisfies $C + \delta C = U(B + \delta B)V^T$ where δB is bounded *componentwise* and δC satisfies a columnwise bound (based upon the growth of the lower right corner of C) with U and V orthogonal to nearly working precision. Once we have that reduction, there is a good menu of algorithms that obtain the singular values of the bidiagonal matrix B to relative accuracy, thus obtaining an SVD of C that can be much more accurate than that obtained from standard bidiagonal reduction procedures. The additional operations required over the standard bidiagonal reduction algorithm of Golub and Kahan [*J. Soc. Indust. Appl. Math. Ser. B Numer. Anal.*, 2 (1965), pp. 205–224] are those for using Givens rotations instead of Householder transformations to compute the matrix V, and $2n^3/3$ flops to compute column norms.

Key words. orthogonal reduction, bidiagonal form, singular values, accuracy

AMS subject classifications. 65F15, 65F35, 15A18

PII. S0895479898343541

1. Introduction.

We consider the problem of reducing $A \in \Re^{m\times n}$ to bidiagonal form. Without loss of generality, assume that $m \geq n$. We find orthogonal matrices $U \in \Re^{n\times n}$ and $V \in \Re^{m\times m}$ such that

$$U^T A^T V = \begin{array}{c} \begin{matrix} n & m-n \end{matrix} \\ \begin{pmatrix} B & 0 \end{pmatrix} \end{array}, \tag{1.1}$$

where

$$B = \begin{pmatrix} \gamma_1 & \phi_2 & 0 & \cdot & \cdot & \cdot \\ 0 & \gamma_2 & \phi_3 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \gamma_{n-1} & \phi_n \\ 0 & \cdot & \cdot & \cdot & 0 & \gamma_n \end{pmatrix}. \tag{1.2}$$

*Received by the editors July 27, 1998; accepted for publication (in revised form) by N.J. Higham September 18, 2001; published electronically January 23, 2002. This research was supported by the National Science Foundation under grants CCR–9424435 and CCR–9732081. Part of this work was done while the author was visiting the Department of Mathematics, University of Manchester, Manchester, UK.

http://www.siam.org/journals/simax/23-3/34354.html

†Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802-6106 (barlow@cse.psu.edu, http://www.cse.psu.edu/∼barlow).

To denote B in (1.2) we use the shorthand

$$B = \text{bidiag}(\gamma_1, \gamma_2, \ldots, \gamma_n; \phi_2, \ldots, \phi_n)$$

or use the MATLAB-like [33] form

$$B = \text{bidiag}(\gamma(1\colon n); \phi(2\colon n)).$$

We will also use MATLAB notation for submatrices. Thus $A(i\colon j, k\colon \ell)$ denotes the submatrix of A consisting of rows i through j and columns k through ℓ. Likewise, $A(:, k\colon \ell)$ denotes all of columns k through ℓ and $A(i\colon j, :)$ denotes all of rows i through j. In some of the error analysis proofs and discussions, we use $(A+B)(i\colon j, k\colon \ell)$ to denote that section of $A+B$.

For a matrix $X \in \Re^{m\times n}$, $m \geq n$, let

$$\sigma_i(X)$$

denote the ith singular value of X and let

$$\sigma_{min}(X) = \min_{\|\mathbf{w}\|_2=1} \|X\mathbf{w}\|_2$$

denote the smallest singular value of X. We let $J(i, j, \theta_{ij})$ be a Givens rotation through an angle θ_{ij} applied to columns i and j, that is, a matrix which is the identity except for the four entries (i,i),(j,i),(i,j), and (j,j) given by $\left(\begin{smallmatrix} cn & sn \\ -sn & cn \end{smallmatrix}\right)$, with $cn = \cos\,\theta_{ij}$ and $sn = \sin\,\theta_{ij}$.

The error analysis bounds are stated as

$$error \leq \varepsilon_M g(n) * factor + O(\varepsilon_M^2).$$

This arise out of simplify expressions of the form

$$(1+\varepsilon_M)^n factor - 1 = n * \varepsilon_M * factor + O(\varepsilon_M^2 factor) = n * \varepsilon_M * factor + O(\varepsilon_M^2).$$

Thus they would be properly stated as

$$error \leq \varepsilon_M g(n) * factor + O(\varepsilon_M^2 factor),$$

but we leave out "factor" in $O(\varepsilon_M^2)$ to make statements less tedious.

The reduction (1.1) is usually done as a preliminary stage for computing the singular value decomposition (SVD) of A. There are now several very good algorithms for computing the SVD of bidiagonal matrices. We know that the "zero-shift" QR algorithm [14], bisection [3, 20], and the dqds algorithm [21] can compute all of the singular values of B to relative accuracy. We also know that it is not reasonable to expect any algorithm to compute all of the singular values of a matrix to relative accuracy unless that matrix has an acyclic graph [13] or is totally sign compound [12].

Thus, it is not surprising that no algorithm can be expected to produce the bidiagonal form of a general matrix to relative accuracy in finite precision arithmetic.

The Jacobi algorithm [28] has a stronger error bound for finding the singular values of a general matrix than any algorithm that requires bidiagonal reduction [15]. Unfortunately, the Jacobi algorithm is usually slower. For simplicity, assume that $m = n$. Bidiagonal reduction followed by the QR algorithm can produce that SVD in

about $20n^3$ flops. One Jacobi sweep requires about $7n^3$ flops. Thus, for Jacobi to be competitive, it must converge in about three sweeps, and that rarely happens.

We note that efforts to improve the performance of Jacobi have been quite successful [17]. However, there have also been significant improvements in speed in obtaining the singular values and vectors of bidiagonal matrices from the dqds algorithms [21, 19, 18, 35, 36]; thus exact complexity comparisons of Jacobi and methods using bidiagonal reduction are a bit elusive. Nonetheless, as of the present writing, algorithms requiring bidiagonal reduction tend to be significantly faster than Jacobi-based algorithms.

In this paper, we present a bidiagonal reduction method that can be expected to preserve more of the accuracy in the SVD.

The reduction is computed in two stages. In the first stage, discussed in section 3, using a Householder factorization method described by Björck [7, pp. 165–169] and analyzed by Cox and Higham [11], it is possible to reduce A to a lower triangular matrix $C \in \Re^{n\times n}$. In floating point arithmetic with machine precision ε_M, for some permutation matrix P and orthogonal matrix $\bar{V}_0$, the first stage reduction satisfies

$$A + \delta A = \bar{V}_0 \begin{pmatrix} C^T \\ 0 \end{pmatrix} P^T, \tag{1.3}$$

where

$$\|\delta A(i,:)\|_2 \le \varepsilon_M \rho_A f(m,n) \|A(i,:)\|_2 + O(\varepsilon_M^2), \tag{1.4}$$

$$i = 1, \ldots, m.$$

There is also a well-known columnwise error bound given in section 3. Here $f(m,n)$ is a modestly sized function and ρ_A is a growth factor given in [11]. A reduction recommended by Demmel and Veselić [15] applied to $A^T A$ before using the Jacobi method produces the same C (in exact arithmetic). The advantage of the row ordering is that if A is well-conditioned after row scalings, the singular values of A will be preserved more accurately in C (see, for instance, [26], or an analysis can be constructed from the discussion in section 2). The discussion below assumes that $\operatorname{rank}(C) = n$, but as shown in section 3, the case $\operatorname{rank}(C) < n$ requires only a postprocessing procedure from Hanson and Lawson [25], and the rest of the algorithm will proceed in the same manner.

Most of the attention in this paper will be focused upon C. Our goal in the above reduction is similar to that in [15], to produce a matrix C that is well-conditioned under column scalings and then apply an SVD routine that has columnwise backward error.

Therefore in the second stage, we apply a new bidiagonal reduction algorithm to C. We show that (see Theorems 6.1 and 6.2) this algorithm produces a bidiagonal matrix B such that for some $\delta B, \delta C \in \Re^{n\times n}$, some modestly sized functions $g_i(n), i = 1\colon 3$, and matrices U and V, a growth factor $\hat{\rho}_V^{(k)}$ (specified in (5.10)), we have

$$C + \delta C = U(B + \delta B)V^T, \tag{1.5}$$

$$\|U^T U - I\|_2, \|V^T V - I\|_2 \le g_1(n)\varepsilon_M + O(\varepsilon_M^2), \tag{1.6}$$

$$|\delta B| \le \varepsilon_M g_2(n)|B| + O(\varepsilon_M^2), \quad |\cdot| \text{ and } \le \text{ entry-wise}, \tag{1.7}$$

$$\|\delta C(:,j)\|_2 \le g_3(n)\varepsilon_M \hat{\rho}_V^{(j-1)} \|C(:,j\!:\!n)\|_F + O(\varepsilon_M^2), \tag{1.8}$$

where

$$\hat{\rho}_V^{(j-1)} \le \|C\|_F / \|C(:,j\!:\!n)\|_F$$

and is usually much smaller. The upper bound on $\hat{\rho}_V^{(j-1)}$ is just a statement that standard normwise bounds apply. Moreover, the algorithm can be implemented in less than $2n^3 + O(n^2)$ more flops than the Lawson–Hanson–Chan SVD [30, pp. 107–120], [9]. In the worst case, the algorithm produces the same error as the standard algorithm; in practice, it appears to be much better.

Our procedure for bidiagonal reduction of C has three important differences from the Golub–Kahan Householder-transformation-based procedure [22]:

- Givens transformations are used in the construction of the right orthogonal matrix V. (Clearly, 2×2 Householder transformations could also be used.) (See section 5.1.)
- The computation of the matrices U and V are interleaved in a different manner to preserve accuracy in the small columns. (See again section 5.1.)
- The rotations are accumulated in a different manner from that normally used. (See section 5.2.)

In the next section, we give the perturbation theory that justifies (1.5). The appropriate algorithm for producing the factorization (1.3)–(1.4) is discussed in section 3. A template for bidiagonal reduction procedures and an outline for the development of our procedure is given in section 4. The material in section 5 provides justification for the choices made in creating Algorithm 6.1 in section 6 from Algorithm 4.1 in section 4. The motivation for using Givens rotations, the different manner in which they are accumulated, and the way in which the computation of U and V are interleaved is justified in section 5. That motivation includes the error analysis of one major step of the algorithm (see Lemma 5.7 and Theorem 5.8 in section 5.2). A full description of the algorithm is the subject of section 6, followed by the error bounds (1.5)–(1.8), formally stated as Theorem 6.1 in section 6.2. The tedious proofs of Lemma 5.5 and Theorem 5.8 from section 5.2 are in the technical report [2].

In section 7, we give some numerical tests. The tests seem to indicate that our error bounds on bidiagonal reduction of C in (1.3) are pessimistic. We could find no examples where the singular values obtained by our procedure differ from those of the Jacobi method by more than those from the Jacobi method are expected to differ from the exact singular values. The Golub–Kahan Householder-based procedure also seems to produce very accurate singular values of such matrices very often. However, the changes made in this paper to the algorithm in [22] were needed in our proofs of the results in section 6.2. The effect of these changes is demonstrated in an example in section 5.3. Moreover, we produce a class of examples of lower triangular matrices of the same form as C for which our procedure is more accurate than the Golub–Kahan procedure. Our conclusion is given in section 8.

2. Necessary perturbation theory. We now give some perturbation theory results to motivate obtaining error bounds of the form (1.5)–(1.8).

The following result shows that the nonorthogonality of U and V in (1.5) causes only a small relative change in the singular values. This lemma is given in [27, problem 18, pp. 423–424].

LEMMA 2.1. *Let $A \in \Re^{m\times n}$ and let $B \in \Re^{n\times n}$. Then for $i = 1, 2, \ldots, n$,*

$$\sigma_i(A)\sigma_n(B) \le \sigma_i(AB) \le \sigma_i(A)\sigma_1(B).$$

Standard bounds on eigenvalue perturbation [23, Chapter 8] lead to

$$1 - g_1(n)\varepsilon_M + O(\varepsilon_M^2) \le \sigma_n^2(U) \le \sigma_1^2(U) \le 1 + g_1(n)\varepsilon_M + O(\varepsilon_M^2),$$

$$1 - g_1(n)\varepsilon_M + O(\varepsilon_M^2) \le \sigma_n^2(V) \le \sigma_1^2(V) \le 1 + g_1(n)\varepsilon_M + O(\varepsilon_M^2),$$

where $g_1(n)$ is the function in (1.6). Thus the singular values of $B + \delta B$ represent those of $C + \delta C$ to relative accuracy.

The effect of the bound in (1.7) is addressed by a result of Demmel and Kahan [14].

LEMMA 2.2. *Let* $B = \text{bidiag}(\gamma(1\colon n); \phi(2\colon n)) \in \Re^{n\times n}$, *let* $\tilde{B} \equiv B + \delta B = \text{bidiag}(\tilde{\gamma}(1\colon n); \tilde{\phi}(2\colon n)) \in \Re^{n\times n}$, *and let* $\zeta \ge 1$. *If*

$$\frac{1}{\zeta} \le \frac{\tilde{\gamma}_j}{\gamma_j}, \frac{\tilde{\phi}_i}{\phi_i} \le \zeta, \qquad \begin{matrix} i = 2, \ldots, n, \\ j = 1, 2, \ldots, n, \end{matrix}$$

with the convention that $0/0 = 1$, *then*

$$\frac{1}{\zeta^{2n-1}} \le \frac{\sigma_j(\tilde{B})}{\sigma_j(B)} \le \zeta^{2n-1}, \quad j = 1, 2, \ldots, n.$$

Thus in (1.5)–(1.7), the singular values of B represent those of $\tilde{B}$ to relative accuracy.

The important effect is that of the error δC which is bounded in (1.5)–(1.8).

To characterize that error, we make the simplifying assumption that C and $C+\delta C$ are both nonsingular. A relaxation of the assumption that $C + \delta C$ is nonsingular is possible; see Barlow and Slapničar [5].

We define two parameters,

$$\tau = \sup_{\mathbf{x}\neq 0} \frac{\|\delta C\mathbf{x}\|_2}{\|C\mathbf{x}\|_2}, \tag{2.1}$$

$$\hat{\tau} = \max\left\{\tau, \sup_{\mathbf{x}\neq 0} \frac{\|\delta C\mathbf{x}\|_2}{\|(C+\delta C)\mathbf{x}\|_2}\right\}. \tag{2.2}$$

Assuming that $\tau < 1$, we have that

$$\hat{\tau} \le \frac{\tau}{1-\tau}.$$

Two simple lemmas illustrate the relative error in the singular values and vectors of C in terms of τ. This result is proved in Demmel and Veselić [15].

LEMMA 2.3. *Let* $C, \delta C \in \Re^{n\times n}$ *be such that both* C *and* $C + \delta C$ *are nonsingular. Let* τ *be given by* (2.1). *Then the singular values of* C *and* $C + \delta C$ *satisfy*

$$\frac{|\sigma_i(C+\delta C) - \sigma_i(C)|}{\sigma_i(C)} \le \tau.$$

We now bound the error in the vectors in terms of $\hat{\tau}$. This is a generalization of a bound in Barlow and Demmel [3] and an elaboration of a bound in Demmel and

Veselić [15]. The analysis is very similar to that used to bound error in subspaces in ULV decomposition in [6]. This result can be generalized to clusters of singular values in a well-known manner; see, for instance, [32, 5].

LEMMA 2.4. *Assume the hypothesis and terminology of Lemma* 2.3. *Let* $(\sigma_i, \mathbf{y}_i, \mathbf{w}_i)$ *denote the ith singular triplet of* C *and let* $(\tilde{\sigma}_i, \tilde{\mathbf{y}}_i, \tilde{\mathbf{w}}_i)$ *denote the ith singular triplet of* $C + \delta C$ *for* $i = 1, \ldots, n$. *Then we have*

$$|\tilde{\mathbf{w}}_j^T \mathbf{w}_i| \leq \hat{\tau} \frac{2\sigma_i \tilde{\sigma}_j}{|\sigma_i^2 - \tilde{\sigma}_j^2|}, \tag{2.3}$$

and

$$|\tilde{\mathbf{y}}_j^T \mathbf{y}_i| \leq \hat{\tau} \frac{\sigma_i^2 + \tilde{\sigma}_j^2}{|\sigma_i^2 - \tilde{\sigma}_j^2|}. \tag{2.4}$$

Proof. To prove (2.3), we simply use the fact that $\mathbf{w}_i$ is an eigenvector of $C^T C$ and $\tilde{\mathbf{w}}_j$ is an eigenvector of $(C + \delta C)^T (C + \delta C)$. Thus

$$\mathbf{w}_i^T (C + \delta C)^T (C + \delta C) \tilde{\mathbf{w}}_j = \tilde{\sigma}_j^2 \mathbf{w}_i^T \tilde{\mathbf{w}}_j,$$

which leads to

$$(\sigma_i^2 - \tilde{\sigma}_j^2) \mathbf{w}_i^T \tilde{\mathbf{w}}_j = -\mathbf{w}_i^T \delta C^T (C + \delta C) \tilde{\mathbf{w}}_j - \mathbf{w}_i^T C^T \delta C \tilde{\mathbf{w}}_j.$$

The use of the Cauchy–Schwarz inequality and the definition of $\hat{\tau}$ in (2.2) yields

$$\begin{aligned} |\sigma_i^2 - \tilde{\sigma}_j^2| |\mathbf{w}_i^T \tilde{\mathbf{w}}_j| &\leq \|\delta C \mathbf{w}_i\|_2 \|(C + \delta C) \tilde{\mathbf{w}}_j\|_2 + \|C \mathbf{w}_i\|_2 \|\delta C \tilde{\mathbf{w}}_j\|_2. \\ &\leq 2\hat{\tau} \sigma_i \tilde{\sigma}_j. \end{aligned}$$

Thus we have (2.3). A nearly identical derivation from $(C + \delta C)\ (C + \delta C)^T$ yields

$$|\sigma_i^2 - \tilde{\sigma}_j^2| |\mathbf{y}_i^T \tilde{\mathbf{y}}_j| \leq \hat{\tau} (\sigma_i^2 + \tilde{\sigma}_j^2).$$

Thus (2.4) holds. ∎

We note that the bound in the error of the right singular vectors is stronger than that for left singular vectors since

$$2\sigma_i \tilde{\sigma}_j \leq \sigma_i^2 + \tilde{\sigma}_j^2.$$

Adding the two bounds together yields Demmel and Veselić's [15] bound

$$|\tilde{\mathbf{w}}_j^T \mathbf{w}_i| + |\tilde{\mathbf{y}}_j^T \mathbf{y}_i| \leq \hat{\tau} \frac{\sigma_i + \tilde{\sigma}_j}{|\sigma_i - \tilde{\sigma}_j|}.$$

The standard error bound on δC is

$$\|\delta C\|_2 \leq \varepsilon_M g_4(n) \|C\|_2 + O(\varepsilon_M^2) \tag{2.5}$$

for some modestly growing function $g_4(n)$. The advantage of (1.8) over (2.5) is a smaller value of τ in (2.1), leading to more accurate singular values and vectors.

A bound on τ is given by

$$\begin{aligned} \tau &= \sup_{\mathbf{x} \neq 0} \frac{\|\delta C \mathbf{x}\|_2}{\|C \mathbf{x}\|_2} = \sup_{\mathbf{y} \neq 0} \frac{\|\delta C C^{-1} \mathbf{y}\|_2}{\|\mathbf{y}\|_2} \\ &= \|\delta C C^{-1}\|_2. \end{aligned}$$

If the only bound that we have on δC is (2.5), then

$$\|\delta C C^{-1}\|_2 \leq \varepsilon_M g_4(n)\kappa_2(C) + O(\varepsilon_M^2), \tag{2.6}$$

where $\kappa_2(C) = \|C\|_2\|C^{-1}\|_2$.

To see the effect of (1.8) we rewrite it as

$$\delta C = E_C R_C D_C, \quad \|E_C\|_2 \leq g_4(n)\varepsilon_M + O(\varepsilon_M^2), \tag{2.7}$$

where

$$D_C = \operatorname{diag}(\|C(:,1)\|_2, \ldots, \|C(:,n)\|_2), \tag{2.8}$$

$$R_C = \operatorname{diag}(\hat{\rho}_V^{(0)}, \ldots, \hat{\rho}_V^{(n-1)}). \tag{2.9}$$

We define

$$\hat{\rho}_V = \|R_C\|_2 = \max_{0\leq k\leq n-1} \hat{\rho}_V^{(k)}. \tag{2.10}$$

A bound of the form (2.7) yields

$$\|\delta C C^{-1}\|_2 \leq \|E_C R_C D_C C^{-1}\|_2 \leq \varepsilon_M \hat{\rho}_V g_4(n)\kappa_2(C D_C^{-1}) + O(\varepsilon_M^2). \tag{2.11}$$

Van der Sluis [38] showed that D_C satisfies

$$\kappa_2(C D_C^{-1}) \leq \sqrt{n}\min\{\kappa_2(CD^{-1}) : D \text{ diagonal and nonsingular}\}. \tag{2.12}$$

Thus the choice of D_C in (2.8) is within a factor of $\sqrt{n}\hat{\rho}_V$ of optimizing the second bound in (2.11) over all nonsingular diagonal scaling matrices.

Noting that the bound in (1.8) implies that in (2.5) the bound in (2.11) cannot be significantly worse than the one in (2.6). Very often it will be significantly better, since it states that the scaling of the columns can be ignored.

3. Reduction to triangular form. Before performing bidiagonal reduction procedure, we assume that A has been reduced to a lower triangular matrix C. We recommend using a Householder-transformation-based procedure discussed by Björck [7, pp. 165–169] and analyzed by Cox and Higham [11]. For more about the use of this procedure in SVD computation, see Higham [26].

The procedure is as follows.

1. Reorder the rows of A so that
$$\|A(1,:)\|_\infty \geq \cdots \geq \|A(m,:)\|_\infty.$$
2. Using the maximal column pivoting algorithm of Businger and Golub [8], factor A into
$$A = V_0 \begin{pmatrix} C^T \\ 0 \end{pmatrix} P^T, \tag{3.1}$$
where $V_0 \in \Re^{m\times m}$ is orthogonal, P is a permutation matrix, and $C \in \Re^{n\times n}$ is lower triangular.

This particular Householder factorization algorithm has very strong numerical stability properties. If we let C be the computed lower triangular factor, then Cox and Higham [11] (based on the analysis of a more complicated pivoting strategy by Powell and Reid [37]) showed that for some orthogonal matrix $\bar{V}_0$ and some matrix δA, we have

$$A + \delta A = \bar{V}_0 \begin{pmatrix} C^T \\ 0 \end{pmatrix} P^T, \tag{3.2}$$

$$\|\delta A(:,j)\|_2 \le c_1(m,n)\|A(:,j)\|_2 \varepsilon_M, \quad j = 1,2,\ldots,n, \tag{3.3}$$

and

$$\|\delta A(i,:)\|_2 \le c_2(m,n)\rho_A \|A(i,:)\|_2 \varepsilon_M + O(\varepsilon_M^2), \quad i = 1,\ldots,m, \tag{3.4}$$

where ρ_A is a growth factor bounded by $\sqrt{n}2^{n-1}$ and $c_k(m,n) = O(mn)$, $k = 1,2$. The column oriented error bound (3.3) holds for standard Householder factorization [39, pp. 152–162]. The second rowwise error bound (3.4) can be shown only for algorithms that do some kind of row and column permutations for stability.

Similar results for Givens-based algorithms have been given [1, 4]. Cox and Higham demonstrate that Householder's original version of Householder transformations must be used for these bounds to hold. The bound does not hold if Parlett's version [34] of the Householder transformation is used.

If the matrix C in (3.2) has rank n, we use that matrix. If $\operatorname{rank}(C) = r < n$, then C has the form

$$C = \begin{array}{c} \\ r \\ n-r \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{pmatrix} C_{11} & 0 \\ C_{12} & 0 \end{pmatrix} \end{array},$$

and we can use a procedure of Hanson and Lawson [25] to produce an orthogonal matrix U_0 from a product of Householder transformations such that

$$U_0^T C = \begin{array}{c} \\ r \\ n-r \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{pmatrix} \tilde{C} & 0 \\ 0 & 0 \end{pmatrix} \end{array}, \tag{3.5}$$

where $\tilde{C}$ remains lower triangular. Our algorithms will use $\tilde{C}$ in place of C, and thus we assume that C is square nonsingular matrix.

Since maximal column pivoting assures us that

$$|C(j,j)| \ge \|C(k,j{:}n)\|_2, \qquad \begin{array}{l} j = 1,\ldots,n-1, \\ k = j+1,\ldots,n, \end{array}$$

and C is lower triangular, we have that the columns of the matrix C satisfy

$$\|C(:,j)\|_2 \ge \frac{1}{\sqrt{n-j}} \|C(:,j+1{:}n)\|_F, \quad j = 1,2,\ldots,n. \tag{3.6}$$

If we are using $\tilde{C}$ in (3.5) in place of C, this bound is satisfied for n, not $r = \operatorname{rank}(C)$.

For the rest of this paper, any reduction of A that produces C satisfying the property (3.6) will be a suitable preprocessing step and will lead to the results given here.

We now give a class of algorithms for computing the bidiagonal reduction of C.

4. A template for bidiagonal reduction. The usual bidiagonal reduction algorithm is that given by Golub and Kahan [22, Theorem 1, pp. 208–210]; see also Golub and Reinsch [24, pp. 404–405]. It is given below.

ALGORITHM 4.1 (Golub–Kahan bidiagonal reduction).

1. *Construct an orthogonal transformation* U_1 *such that*

$$U_1^T C(:,1) = \gamma_1 \mathbf{e}_1,$$

$$C \leftarrow U_1^T C;\ U \leftarrow U_1;\ V \leftarrow V_1 \equiv I;$$

2. ***for*** $k = 2 : n-1$
   (a) *If* $C(k-1, k\colon n) \neq 0$, *construct an orthogonal transformation* V_k *such that*

$$V_k^T C(k-1, k\colon n)^T = \phi_k \mathbf{e}_1,$$

$$C(k\colon n, k\colon n) \leftarrow C(k\colon n, k\colon n) V_k, \quad V(:, k\colon n) \leftarrow V(:, k\colon n) V_k.$$

   else $\phi_k = 0$, *and* $V_k = I$ *(implicitly).*
   (b) *Construct an orthogonal transformation* U_k *such that*

$$U_k^T C(k\colon n, k) = \gamma_k \mathbf{e}_1,$$

$$C(k\colon n, k\colon n) \leftarrow U_k^T C(k\colon n, k\colon n), \quad U(:, k\colon n) \leftarrow U(:, k\colon n) U_k.$$

3.

$$\gamma_n \leftarrow C(n,n), \quad \phi_n \leftarrow C(n-1, n).$$

The bidiagonal reduction of C *is given by*

$$C = UBV^T,$$

where

$$B = \text{bidiag}(\gamma_1, \ldots, \gamma_n; \phi_2, \ldots, \phi_n).$$

Golub and Kahan [22] use Householder transformations to describe this algorithm.

Algorithm 4.1 allows a number of options for its implementation. Most important is how we choose U_k and V_k for each step in the transformation. We will follow the convention in [22] and choose $U_k, k = 1, \ldots, n-1$, to be Householder transformations. In section 5.1, we justify choosing $V_k, k = 2, \ldots, n-1$, to be Givens rotations in standard order and show why the application of U_k and V_k must be interleaved differently from described above. In section 5.2, we justify an unusual method for applying V_k.

5. The use and application of Givens rotations in constructing V_k.

5.1. Eliminating columns from the right and element growth. To formulate a new algorithm for bidiagonal reduction, we consider the effect of the orthogonal transformations used to form V in Algorithm 4.1. Let $U_1, \ldots, U_{n-1}$ and $V_1, \ldots, V_{n-1}$ be the orthogonal transformations from Algorithm 4.1. Define

$$\tilde{U}_k = \hat{U}_1 \cdots \hat{U}_k, \quad \hat{U}_k = \operatorname{diag}(I_{k-1}, U_k), \tag{5.1}$$

$$\tilde{V}_k = \hat{V}_1 \cdots \hat{V}_k, \quad \hat{V}_k = \operatorname{diag}(I_{k-1}, V_k), \quad k = 1, \ldots, n-1. \tag{5.2}$$

Define

$$F^{(k)} = \tilde{U}_k^T C, \quad k = 1, \ldots, n-1.$$

By orthogonal equivalence,

$$\|F^{(k)}\mathbf{e}_i\|_2 = \|C\mathbf{e}_i\|_2.$$

If we let

$$C^{(k)} = F^{(k)}\tilde{V}_k = \tilde{U}_k^T C \tilde{V}_k, \tag{5.3}$$

then $C^{(k)}$ has the form

$$C^{(k)} = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} C_{11}^{(k)} & C_{12}^{(k)} \\ 0 & C_{22}^{(k)} \end{pmatrix} \end{array},$$

where

$$C_{11}^{(k)} = \operatorname{bidiag}(\gamma(1\colon k); \phi(2\colon k)),$$

and $C_{12}^{(k)}$ is zero except for the last row. Therefore, $\tilde{V}_k$, in effect, zeros out the block $F_{21}^{(k)}$.

The following lemma shows the effect of a large class of orthogonal transformations from the right.

LEMMA 5.1. *Let $F \in \Re^{m\times n}$ and $V \in \Re^{n\times n}$ be partitioned according to*

$$F = \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} \end{array}, \qquad V = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \end{array}, \tag{5.4}$$

where V_{11} is nonsingular. Let

$$G = \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} G_{11} & G_{12} \\ 0 & G_{22} \end{pmatrix} \end{array} = FV. \tag{5.5}$$

Then

$$F_{21} = -F_{22}V_{21}V_{11}^{-1}, \quad G_{22} = F_{22}\tilde{V}_{22}, \tag{5.6}$$

where

$$\tilde{V}_{22} = V_{22} - V_{21}V_{11}^{-1}V_{12}. \tag{5.7}$$

Proof. Matching blocks in (5.4) and (5.5) yields

$$F_{21}V_{11} + F_{22}V_{21} = 0, \quad F_{21}V_{12} + F_{22}V_{22} = G_{22}.$$

Using the fact that V_{11} is nonsingular, block Gaussian elimination yields (5.6). ∎

The following generalization of the result of Lemma 5.1 provides a key portion of the proof of Theorem 5.4.

LEMMA 5.2. *Let $F \in \Re^{m\times n}$ be partitioned according to*

$$F = \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \end{pmatrix} \end{matrix}.$$

Let $V = \bar{V}_1\bar{V}_2$, where

$$\bar{V}_1 = \begin{matrix} \\ k \\ j-k \\ n-j \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} V_{11}^{(1)} & V_{12}^{(1)} & 0 \\ V_{21}^{(1)} & V_{22}^{(1)} & 0 \\ 0 & 0 & I_{n-j} \end{pmatrix} \end{matrix}, \qquad \bar{V}_2 = \begin{matrix} \\ k \\ j-k \\ n-j \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} V_{11}^{(2)} & 0 & V_{12}^{(2)} \\ 0 & I_{j-k} & 0 \\ V_{21}^{(2)} & 0 & V_{22}^{(2)} \end{pmatrix} \end{matrix},$$

and V has the form in Lemma 5.1*, assume $V_{11} = V_{11}^{(1)}V_{11}^{(2)}$ is nonsingular, and let $\tilde{V}_{22}$ be given by* (5.7)*. Let $G = FV$ be partitioned,*

$$G = \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} G_{11} & G_{12} & G_{13} \\ 0 & G_{22} & G_{23} \end{pmatrix} \end{matrix}. \tag{5.8}$$

Then

$$G_{23} = F_{23}\tilde{V}_{22}^{(2)}, \tag{5.9}$$

where

$$\tilde{V}_{22}^{(2)} = V_{22}^{(2)} - V_{21}^{(2)}[V_{11}^{(2)}]^{-1}V_{12}^{(2)}$$

and $\|\tilde{V}_{22}^{(2)}\|_2 \le \|\tilde{V}_{22}\|_2$.

Proof. We note that

$$V = \bar{V}_1\bar{V}_2 = \begin{matrix} \\ k \\ j-k \\ n-j \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} V_{11}^{(1)}V_{11}^{(2)} & V_{12}^{(1)} & V_{11}^{(1)}V_{12}^{(2)} \\ V_{21}^{(1)}V_{11}^{(2)} & V_{22}^{(1)} & V_{21}^{(1)}V_{12}^{(2)} \\ V_{21}^{(2)} & 0 & V_{22}^{(2)} \end{pmatrix} \end{matrix}.$$

If we partition V according to (5.4), then

$$V_{11} = V_{11}^{(1)}V_{11}^{(2)}, \qquad V_{12} = \begin{pmatrix} V_{12}^{(1)} & V_{11}^{(1)}V_{12}^{(2)} \end{pmatrix},$$

$$V_{21} = \begin{pmatrix} V_{21}^{(1)} V_{11}^{(2)} \\ V_{21}^{(2)} \end{pmatrix}, \qquad V_{22} = \begin{pmatrix} V_{22}^{(1)} & V_{21}^{(1)} V_{12}^{(2)} \\ 0 & V_{22}^{(2)} \end{pmatrix}.$$

First we note that V_{11} is nonsingular if and only if $V_{11}^{(i)}, i = 1, 2$, are nonsingular. Evaluating $\tilde{V}_{22}$ according to (5.7) leads to

$$\tilde{V}_{22} = \begin{pmatrix} V_{22}^{(1)} - V_{21}^{(1)}[V_{11}^{(1)}]^{-1} V_{12}^{(1)} & 0 \\ -V_{21}^{(2)}[V_{11}^{(2)}]^{-1}[V_{11}^{(1)}]^{-1} V_{12}^{(1)} & V_{22}^{(2)} - V_{21}^{(2)}[V_{11}^{(2)}]^{-1} V_{12}^{(2)} \end{pmatrix}.$$

Thus $\|\tilde{V}_{22}^{(2)}\|_2 \le \|\tilde{V}_{22}\|_2$. Now we have that

$$F^{(1)} = F\bar{V}_1 = \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & j-k & n-j \end{matrix} \\ \begin{pmatrix} F_{11}^{(1)} & F_{12}^{(1)} & F_{13} \\ F_{21}^{(1)} & F_{22}^{(1)} & F_{23} \end{pmatrix} \end{matrix},$$

where the (1,3) and (2,3) blocks of F are unaffected. We then apply Lemma 5.1 to

$$\begin{pmatrix} F_{11}^{(1)} & F_{13} \\ F_{21}^{(1)} & F_{23} \end{pmatrix} \begin{pmatrix} V_{11}^{(2)} & V_{12}^{(2)} \\ V_{21}^{(2)} & V_{22}^{(2)} \end{pmatrix} = \begin{pmatrix} G_{11} & G_{13} \\ 0 & G_{23} \end{pmatrix}$$

to obtain (5.9). ∎

The following lemma gives us an alternative method for bounding the growth of the $(2, 3)$ block.

LEMMA 5.3. *Assume the hypothesis and terminology of Lemma* 5.2. *Assume also that V is orthogonal. Then*

$$\|G_{23}\|_2 \le \rho_V^{(k)} \|F_{23}\|_2, \quad \rho_V^{(k)} \equiv \|\tilde{V}_{22}\|_2 = \left(1 + \|V_{21}V_{11}^{-1}\|_2^2\right)^{1/2}, \tag{5.10}$$

$$\sigma_{min}(\tilde{V}_{22}) \ge 1.$$

Proof. From (5.7), we have

$$\begin{matrix} \\ \end{matrix} \begin{matrix} k & n-k \end{matrix}$$
$$\begin{pmatrix} 0 & \tilde{V}_{22} \end{pmatrix} = \begin{pmatrix} 0 & V_{22} - V_{21}V_{11}^{-1}V_{12} \end{pmatrix} = \begin{pmatrix} -V_{21}V_{11}^{-1} & I_{n-k} \end{pmatrix} V.$$

By orthogonal equivalence

$$\|\tilde{V}_{22}\|_2 = \left\| \begin{pmatrix} -V_{21}V_{11}^{-1} & I_{n-k} \end{pmatrix} \right\|_2 = \left(1 + \|V_{21}V_{11}^{-1}\|_2^2\right)^{1/2}.$$

This equivalence is also true for all of the other singular values of $\tilde{V}_{22}$, and thus

$$\sigma_{min}(\tilde{V}_{22}) = \sigma_{min}[\begin{pmatrix} -V_{21}V_{11}^{-1} & I_{n-k} \end{pmatrix}^T] \ge 1.$$

From (5.9) and the result of Lemma 5.2, we have (5.10). ∎

The value of $\hat{\rho}_V^{(k)}$ used in (1.8) and again in (2.9) is given by

$$\hat{\rho}_V^{(k)} = \min\{\rho_V^{(k)}, \|C\|_F / \|C(:, k+1:n)\|_F\}. \tag{5.11}$$

We now assume that the kth right orthogonal transformation $\hat{V}_k$ has the form

$$\hat{V}_k = V_{k,k+1} \cdots V_{k,n}, \quad k = 1, \ldots, n-1, \tag{5.12}$$

where $V_{k,j} = J(k, j, \theta_{k,j})$ is a Givens rotation. The rotation $V_{k,j}$ sets entry $(k-1, j)$ of C (at step k of the algorithm) to zero. We note that we could also assume the use of 2×2 Householder transformations without changing any of the results below. The following is a key result on the growth of the elements of C through the course of Algorithm 4.1.

THEOREM 5.4. *Let Algorithm* 4.1 *be applied to* C *with matrix* V_k *given by the product of Givens rotations in* (5.12). *For* $k = 1, \ldots, n-1$, *let* $\tilde{U}_k$ *be defined by* (5.1), *let* $\tilde{V}_k$ *be defined by* (5.2), *and let* $C^{(k)}$ *be defined by* (5.3). *Let* $\rho_V^{(k)}$ *be defined as in Lemma* 5.3 *and let* $\hat{\rho}_V^{(j-1)}$ *be as defined in* (5.11). *Then for* $k = 1, \ldots, n-1$, *and* $j = k+1, \ldots, n$, *we have*

$$\|C^{(k)}(k+1\colon n, j\colon n)\|_F \le \min\{\rho_V^{(k)}, \hat{\rho}_V^{(j-1)}\} \|C(:, j\colon n)\|_F. \tag{5.13}$$

Proof. Since $\tilde{V}_k$ is the product of Givens rotations in the standard order, we directly apply the results in Lemma 5.2.

Since $\tilde{V}_k = \prod_{i=1}^{k} \prod_{\ell=i+1}^{n} V_{i,\ell}$, for each $j > k$, taking advantage of rotations that commute, we can write

$$\tilde{V}_k = \bar{V}_1 \bar{V}_2,$$

where

$$\bar{V}_1 = \prod_{i=1}^{k} \prod_{\ell=i+1}^{j-1} V_{i,\ell}, \quad \bar{V}_2 = \prod_{i=1}^{k} \prod_{\ell=j}^{n} V_{i,\ell}.$$

Thus, $\bar{V}_1$ and $\bar{V}_2$ have the structure in the hypothesis of Lemma 5.2. Using the terminology of that lemma, we have

$$C^{(k)}(k+1\colon n, j\colon n) = F^{(k)}(k+1\colon n, j\colon n)\tilde{V}_{22}^{(2)} \tag{5.14}$$

and that

$$\begin{aligned} \|C^{(k)}(k+1\colon n, j\colon n)\|_F &\le \|F^{(k)}(k+1\colon n, j\colon n)\|_F \|\tilde{V}_{22}^{(2)}\|_2 \\ &\le \|C(k+1\colon n, j\colon n)\|_F \|\tilde{V}_{22}^{(2)}\|_2 \le \rho_V^{(k)} \|C(:, j\colon n)\|_F, \end{aligned}$$

which is the first half of (5.13). To show that the above bound holds with $\hat{\rho}_V^{(j-1)}$, assume $j > k+1$. We note that

$$\tilde{V}_{j-1} = Z_1 Z_2,$$

where

$$Z_1 = \prod_{i=1}^{j-2} \prod_{\ell=i+1}^{j-1} V_{i,\ell}, \quad Z_2 = \prod_{i=1}^{j-1} \prod_{\ell=j}^{n} V_{i,\ell},$$

and

$$Z_1 = \begin{array}{c} \\ j-1 \\ n-j+1 \end{array} \begin{array}{c} \begin{array}{cc} j-1 & n-j+1 \end{array} \\ \begin{pmatrix} Z_{11}^{(1)} & 0 \\ 0 & I \end{pmatrix} \end{array}, \qquad Z_2 = \begin{array}{c} \\ j-1 \\ n-j+1 \end{array} \begin{array}{c} \begin{array}{cc} j-1 & n-j+1 \end{array} \\ \begin{pmatrix} Z_{11}^{(2)} & Z_{12}^{(2)} \\ Z_{21}^{(2)} & Z_{22}^{(2)} \end{pmatrix} \end{array}.$$

It is easily shown that

$$\rho_V^{(j-1)} = \|\tilde{Z}_{22}\|_2 = \|\tilde{Z}_{22}^{(2)}\|_2.$$

Thus we need only show that $\|\tilde{Z}_{22}^{(2)}\|_2 \geq \|\tilde{V}_{22}^{(2)}\|_2$.

We have that

$$Z_2 = \bar{V}_2 W,$$

where

$$W = \prod_{i=k+1}^{j-1} \prod_{\ell=j}^{n} V_{i,\ell}.$$

Here W has the structure

$$W = \begin{array}{c} \\ k \\ j-k-1 \\ n-j+1 \end{array} \begin{array}{c} \begin{array}{ccc} k & j-k-1 & n-j+1 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & W_{11} & W_{12} \\ 0 & W_{21} & W_{22} \end{pmatrix} \end{array}.$$

Therefore the blocks of Z_2 may be written

$$Z_{11}^{(2)} = \begin{pmatrix} V_{11}^{(2)} & V_{12}^{(2)} W_{21} \\ 0 & W_{11} \end{pmatrix}, \quad Z_{12}^{(2)},$$

$$Z_{21}^{(2)} = \begin{pmatrix} V_{21}^{(2)} & V_{22}^{(2)} W_{21} \end{pmatrix}, \quad Z_{22}^{(2)} = V_{22}^{(2)} W_{22}.$$

Some algebra shows that

$$\tilde{Z}_{22}^{(2)} = Z_{22}^{(2)} - Z_{21}^{(2)} [Z_{11}^{(2)}]^{-1} Z_{12}^{(2)}$$

$$= \tilde{V}_{22}^{(2)} \tilde{W}_{22}.$$

We have that

$$\rho_V^{(j-1)} = \|\tilde{Z}_{22}^{(2)}\|_2 \geq \|\tilde{V}_{22}^{(2)}\|_2 \sigma_{min}(\tilde{W}_{22}).$$

By Lemma 5.3, $\sigma_{min}(\tilde{W}_{22}) \geq 1$, so

$$\rho_V^{(j-1)} \geq \|\tilde{V}_{22}^{(2)}\|_2.$$

Using (5.14), we get

$$\|C^{(k)}(k+1\colon n, j\colon n)\|_F \leq \rho_V^{(j-1)} \|C(\colon, j\colon n)\|_F.$$

Since orthogonal equivalence gives us

$$\|C^{(k)}(k+1\colon n, j\colon n)\|_F \leq \|C\|_F$$

we have (5.13). ∎

As will be shown later in Theorem 6.1, if we can control the growth of the elements in the lower corner $C^{(k)}$, then we can obtain a columnwise backward error bound on our implementation of Algorithm 4.1.

Immediately, we make two changes to Algorithm 4.1. The first is the use of Givens rotations in standard order to compose the transformations $V_k, k = 1, \ldots, n-1$, as in (5.12). The second is to note that for the entries of $C(k+1\colon n, k+1\colon n)$ to never exceed the bound on $C^{(k)}(k+1\colon n, k+1\colon n)$ in (5.13), $C^{(k)}$ should be computed by the recurrence

$$C^{(0)} = C, \tag{5.15}$$

$$M^{(k)} \equiv \hat{U}_k^T C^{(k-1)}, \quad k = 1, \ldots, n-1, \tag{5.16}$$

$$C^{(k)} = M^{(k)} \hat{V}_k. \tag{5.17}$$

Thus, even though V_k is formulated before U_k, it must be applied to C after U_k.

If we use the order of application in Algorithm 4.1, then we would produce

$$N^{(k)} \equiv C^{(k-1)} V_k,$$

$$C^{(k)} = U_k^T N^{(k)},$$

but there is the possibility that

$$\|N^{(k)}(k+1\colon n, j\colon n)\|_F \gg \|C^{(k)}(k+1\colon n, j\colon n)\|_F \tag{5.18}$$

for some $j > k \geq 1$, and our error analysis depends upon the ability to bound the growth in these values. In the example in section 5.3, we show that exactly this phenomenon occurs and that it can affect the accuracy of the small singular values.

The main loop of Algorithm 4.1 is modified as follows. If $C(k-1, k\colon n) \neq 0$, construct V_k such that

$$V_k^T C(k-1, k\colon n)^T = \phi_k \mathbf{e}_1,$$

$$\mathbf{y}_k = C(k\colon n, k\colon n) V_k \mathbf{e}_1.$$

Find U_k such that

$$U_k^T \mathbf{y}_k = \gamma_k \mathbf{e}_1, \tag{5.19}$$

$$C(k\colon n, k\colon n) \leftarrow U_k^T C(k\colon n, k\colon n),$$

$$C(k\colon n, k\colon n) \leftarrow C(k\colon n, k\colon n) V_k.$$

To ensure that $C^{(k)}(k\colon n, k) = \gamma_k \mathbf{e}_1$ for some γ_k with appropriate backward error, we modify the manner in which the orthogonal transformation V_k is applied to C. Those modifications are discussed in the next section.

5.2. Application of the U_k and V_k matrices. We now explain how to implement one step of Algorithm 4.1 in the order (5.16)–(5.17) while preserving the identity

$$C^{(k)}(k\colon n, k) = \gamma_k \mathbf{e}_1. \tag{5.20}$$

We assume that $|\phi_k| = \|C^{(k-1)}(k-1, k\colon n)\|_2 \neq 0$, otherwise $V_k = I$, and the steps in this section are unnecessary.

Let $M^{(k)}$ be as defined in (5.16). We also define

$$\mathbf{v}_1^{(k)} = V_k \mathbf{e}_1 \equiv (v_{kk}^{(k)}, \ldots, v_{nk}^{(k)})^T. \tag{5.21}$$

Since, by assumption, $C^{(k-1)}(k-1, k\colon n) \neq 0$, $\mathbf{v}_1^{(k)} = C^{(k-1)}(k-1, k\colon n)^T/\phi_k$, this first column will determine the Givens rotations $V_{k,k+1}, \ldots, V_{k,n}$. We adopt three conventions in our discussions in this section:

- To avoid greater notational complexity, we assume that $M^{(k)}$ and $C^{(k)}$ are the computed values in (5.16)–(5.17) rather than the exact ones. We also use this assumption throughout section 6.
- The dimensions of the matrix $V_{k,j}$ will be different in different contexts, but will always refer to a Givens rotation whose nontrivial portion is applied to columns k and j of C.
- We ignore the rounding errors in applying the rotations in $\hat{V}_k$ to $M^{(k)}$, but not the errors in forming the rotations. Theorem 5.8, at the end of this section, shows that assumption can be lifted and still obtain the necessary error bounds.

The important concern in (5.20) is to maintain

$$C^{(k)}(k+1\colon n, k) = 0, \tag{5.22}$$

and thus row k of $C^{(k)}$ may be computed according to

$$C^{(k)}(k, k\colon n) = M^{(k)}(k, k\colon n) V_{k,k+1} \cdots V_{k,n} \tag{5.23}$$

in the standard manner. Our discussion will center upon how to compute rows $k+1$ through n of $C^{(k)}$.

In exact arithmetic, the statement (5.20) implies that

$$M^{(k)}(k\colon n, k\colon n) V_k \mathbf{e}_1 = M^{(k)}(k\colon n, k\colon n)\mathbf{v}_1^{(k)} = \gamma_k \mathbf{e}_1 \tag{5.24}$$

for some γ_k. Unfortunately, since U_k and V_k are applied in the opposite order from which they are defined in (5.15)–(5.17), we do not expect (5.24) to hold in floating point arithmetic.

We begin the following lemma which is proven in [2]. It shows that a columnwise perturbation of $M^{(k)}$ satisfies (5.24). That result is a building block for developing our procedure for computing $C^{(k)}$ from $M^{(k)}$.

LEMMA 5.5. *Let $M^{(k)} \in \Re^{n\times n}$ be the computed result of applying the Householder transformation U_k to $C^{(k-1)}$ in* (5.16). *Then for some $\delta M_0^{(k)} \in \Re^{n\times n}$ and modestly growing function $g_0(n)$ we have that*

$$(M^{(k)} + \delta M_0^{(k)})(k\colon n, k\colon n)\mathbf{v}_1^{(k)} = \gamma_k \mathbf{e}_1,$$

where, for $j = k, \ldots, n$,

$$\|\delta M_0^{(k)}(:, j)\|_2 \leq \varepsilon_M g_0(n-k+1)\|M^{(k)}(k\colon n, j)\|_2 + O(\varepsilon_M^2)$$

and

$$g_0(n) = n^2 + 14n. \tag{5.25}$$

The assumption that U_k is a Householder transformation determines the function $g_0(n)$, but any stable implementation of orthogonal transformations such that (5.19) holds will yield this bound with a different modestly growing function $g_0(n)$.

We note that Lemma 5.5 gives us that

$$M^{(k)}(k\colon n, k\colon n)\mathbf{v}_1^{(k)} = \gamma_k \mathbf{e}_1 - \delta\mathbf{m}_1, \quad \delta\mathbf{m}_1 = \delta M_0^{(k)}(k\colon n, k\colon n)\mathbf{v}_1^{(k)}. \tag{5.26}$$

When applying V_k to $M^{(k)}$ the computation (5.26) is not actually performed. Instead we accept $\gamma_k \mathbf{e}_1$ as the correct result for $C^{(k)}(k\colon n, k)$, thereby enforcing (5.20).

However, (5.26) tells us that the exact relation between $M^{(k)}$ and $C^{(k)}$ is

$$M^{(k)}(k\colon n, k\colon n) - \delta\mathbf{m}_1[\mathbf{v}_1^{(k)}]^T = C^{(k)}(k\colon n, k\colon n)V_k^T$$

or

$$(M^{(k)} + \delta M_1^{(k)})(k\colon n, k\colon n) = C^{(k)}(k\colon n, k\colon n)V_k^T, \tag{5.27}$$

where

$$\delta M_1^{(k)}(k\colon n, k\colon n) = \delta M_0^{(k)}(k\colon n, k\colon n)\mathbf{v}_1^{(k)}[\mathbf{v}_1^{(k)}]^T.$$

Thus applying $\hat{V}_k$ to $M^{(k)}$ and enforcing (5.20) creates the backward error $\delta M_1^{(k)}$ (*not* $\delta M_0^{(k)}$). Unfortunately, $\delta M_1^{(k)}$ has only the normwise bound

$$\|\delta M_1^{(k)}\|_F \leq \|\delta M_0^{(k)}\|_F \leq \varepsilon_M g_0(n-k+1)\|M^{(k)}\|_F + O(\varepsilon_M^2). \tag{5.28}$$

From the analysis of section 2*, we need columnwise error bounds; thus* (5.28) *is not acceptable.*

Instead, we compute with a matrix $C^{(k,k)}$ such that

$$C^{(k,k)} \equiv M^{(k)} + \delta M_2^{(k)}, \tag{5.29}$$

$$C^{(k,k)}(k+1\colon n, k\colon n)\mathbf{v}_1^{(k)} = 0, \tag{5.30}$$

and for a modestly growing function $\tilde{g}_0(n)$,

$$\delta M_2^{(k)} = \delta\tilde{\mathbf{m}}_s \mathbf{e}_s^T, \tag{5.31}$$

$$\|\delta\tilde{\mathbf{m}}_s\|_2 \leq \varepsilon_M \tilde{g}_0(n-k+1)\|M^{(k)}(k+1\colon n, s)\|_2 + O(\varepsilon_M^2) \tag{5.32}$$

for some $s \in \{k, k+1, \ldots, n\}$. Thus $C^{(k,k)}(:, j) = M^{(k)}(:, j)$ for $j \neq s$, and

$$C^{(k,k)}(:, s) = M^{(k)}(:, s) + \delta\tilde{\mathbf{m}}_s. \tag{5.33}$$

There is no reason to perturb rows $1, 2, \ldots, k$, since there we accept the computed result of (5.16) and (5.17), so $\delta\tilde{\mathbf{m}}_s(1\colon k) = 0$.

We then obtain $C^{(k)}$ from the computation

$$C^{(k,j)} \equiv C^{(k,j-1)} V_{k,j}, \quad j = k+1, \ldots, n, \tag{5.34}$$

$$C^{(k)} = C^{(k,n)} = C^{(k,k)} V_k = (M^{(k)} + \delta M_2^{(k)}) V_k.$$

The perturbation $\delta M_2^{(k)}$ introduces only a columnwise backward error but guarantees (5.20). Below, we show how to obtain a $C^{(k,k)}$ that satisfies (5.30)–(5.32), and thereby a procedure to obtain $C^{(k-1)}$ from $C^{(k)}$ with an appropriate error bound.

First, we show how to get $C^{(k,k)}(k+1\colon n, s)$ for any $s \in \{k, \ldots, n\}$. We then show how to choose s to enforce (5.32). To construct $C^{(k,k)}$, we enforce (5.30). Since

$$C^{(k,j)} = C^{(k,k)} V_{k,k+1} \cdots V_{k,j}, \quad j = k+1, \ldots, n,$$

the equation (5.30) is equivalent to the statement

$$\begin{pmatrix} C^{(k,j)}(k+1\colon n, k) & C^{(k,k)}(k+1\colon n, j\colon n) \end{pmatrix} V_{k,j+1} \cdots V_{k,n} \tag{5.35}$$

$$= \begin{pmatrix} 0 & C^{(k)}(k+1\colon n, j\colon n) \end{pmatrix} \tag{5.36}$$

for $j = k+1, \ldots, n$.

In terms of the components of the Givens rotations, column k of recurrence (5.35)–(5.36) is written

$$C^{(k,j)}(k+1\colon n, k)$$

$$= (cn)_{k,j} C^{(k,j-1)}(k+1\colon n, k) + (sn)_{k,j} C^{(k,k)}(k+1\colon n, j), \tag{5.37}$$

$$j = k+1, \ldots, n,$$

with $C^{(k,n)} = C^{(k)}$ and $C^{(k,n)}(k+1\colon n, k) = 0$.

Writing the recurrence (5.37) in reverse we have

$$C^{(k,n)}(k+1\colon n, k) = C^{(k)}(k+1\colon n, k) = 0, \tag{5.38}$$

$$C^{(k,j-1)}(k+1\colon n, k)$$

$$= (C^{(k,j)}(k+1\colon n, k) - (sn)_{k,j} C^{(k,k)}(k+1\colon n, j))/(cn)_{k,j}, \tag{5.39}$$

$$j = n, \ldots, k+1.$$

To obtain the value of $C^{(k,k)}(k+1\colon n, s)$ we need both the recurrence (5.37) and the recurrence (5.38)–(5.39). We consider two separate cases, $s = k$ and $s > k$.

For $s = k$, we construct $C^{(k,j)}(k+1\colon n, k)$, $j = n, \ldots, k$, using (5.38)–(5.39), thus constructing $C^{(k,k)}(k+1\colon n, k)$.

For $s > k$, we construct $C^{(k,j)}(k+1\colon n,k), j = k,\dots,s-1$, using (5.37), then construct $C^{(k,j)}(k+1\colon n,k), j = n,\dots,s$, using (5.38)–(5.39). Then compute $C^{(k,k)}(k+1\colon n,s)$ from

$$C^{(k,k)}(k+1\colon n,s) = C^{(k,s-1)}(k+1\colon n,s)$$

$$= (C^{(k,s)}(k+1\colon n,k) - (cn)_{k,s}C^{(k,s-1)}(k+1\colon n,k))/(sn)_{k,s}. \tag{5.40}$$

For both cases, the jth column of $C^{(k)}$ is computed from

$$C^{(k)}(k+1\colon n,j) = (cn)_{k,j}C^{(k,k)}(k+1\colon n,j) - (sn)_{k,j}C^{(k,j-1)}(k+1\colon n,k) \tag{5.41}$$

when both terms of (5.41) are available.

The value of $C^{(k,k)}(k\colon n,s)$ may not satisfy (5.32) and (5.33) for every value of s. Fortunately, as Proposition 5.6 shows, there is always at least one value of s such that (5.32) and (5.33) are enforced.

PROPOSITION 5.6. *Assume the hypothesis and terminology of Lemma* 5.5. *Let* $C^{(k,k)}$ *be defined by* (5.29) *and satisfy* (5.30). *If* s *is chosen so that*

$$\|M^{(k)}(k\colon n,s)\|_2|v_{sk}^{(k)}| = \max_{k\le j\le n}\|M^{(k)}(k\colon n,j)\|_2|v_{jk}^{(k)}|, \tag{5.42}$$

then $\delta\tilde{\mathbf{m}}_s$ *satisfies* (5.32) *with* $\tilde{g}_0(n) = ng_0(n)$. *With this choice, excluding the rounding error from applying the Givens rotations, the computed matrix* $C^{(k)}$ *satisfies*

$$C^{(k)} = (M^{(k)} + \delta M_2^{(k)})V_k,$$

where

$$\|\delta M_2^{(k)}(:,j)\|_2 \le \begin{cases} \tilde{g}_0(n-k+1)\varepsilon_M\|M_2^{(k)}(:,s)\|_2 + O(\varepsilon_M^2), & j = s,\\ 0, & j \ne s. \end{cases} \tag{5.43}$$

Proof. From Lemma 5.5 and (5.30) we must have

$$\delta M_0^{(k)}(k+1\colon n,k\colon n)\mathbf{v}_1^{(k)} = \delta M_2^{(k)}(k+1\colon n,k\colon n)\mathbf{v}_1^{(k)},$$

thus

$$\sum_{j=k}^{n}\delta M_0^{(k)}(k+1\colon n,j)v_{jk}^{(k)} = \delta\tilde{\mathbf{m}}_s(k+1\colon n)\mathbf{e}_{s-k+1}^T\mathbf{v}_1^{(k)} = \delta\tilde{\mathbf{m}}_s(k+1\colon n)v_{sk}^{(k)}.$$

Therefore,

$$\|\delta\tilde{\mathbf{m}}_s(k+1\colon n)\|_2|v_{sk}^{(k)}| \le \sum_{j=k}^{n}|v_{jk}^{(k)}|\|\delta M^{(k)}(k+1\colon n,j)\|_2 \tag{5.44}$$

$$\le \varepsilon_M g_0(n-k+1)\sum_{j=k}^{n}|v_{jk}^{(k)}|\|M^{(k)}(k\colon n,j)\|_2 + O(\varepsilon_M^2). \tag{5.45}$$

If we choose s so that (5.42) holds, then (5.45) becomes

$$\|\delta\tilde{\mathbf{m}}_s(k+1\colon n)\|_2 \le \varepsilon_M\tilde{g}_0(n-k+1)\|M^{(k)}(k+1\colon n,s)\|_2 + O(\varepsilon_M^2), \tag{5.46}$$

which satisfies the second expression in (5.31) with $\tilde{g}_0(n) = ng_0(n)$. Thus $\delta M_2^{(k)}$ satisfies (5.43). ∎

Note that Proposition 5.6 requires the computation of the column norms

$$\|M^{(k)}(k\colon n, j)\|_2, \quad j = k, \ldots, n. \tag{5.47}$$

These must be recomputed for each k, thus adding $2(n-k)^2 + O(n)$ flops at each step of the bidiagonal reduction algorithm.

Our procedure for applying V_k is summarized as follows.

ALGORITHM 5.1 (algorithm for applying V_k).

1. *Compute the column norms in* (5.47)*, then determine s that satisfies* (5.42).
2. *Compute $C^{(k)}(k, k\colon n) = M^{(k)}(k, k\colon n)V_k$ in the ordinary manner.*
3. *If $s = k$, compute all $C^{(k,j)}(k+1\colon n, k)$ using the backward recurrence* (5.38)–(5.39)*, thus implicitly replacing $M^{(k)}(k+1\colon n, k)$ with $C^{(k,k)}(k+1\colon n, k)$.*
4. *If $s \neq k$, for $j = k, \ldots, s-1$ compute $C^{(k,j)}(k+1\colon n, j)$ using the forward recurrence* (5.37)*. We then compute $C^{(k,k)}(k+1\colon n, s) = M^{(k)}(k+1\colon n, s) + \delta\tilde{\mathbf{m}}_s$ from* (5.40).
5. *For either* (3) *or* (4)*, compute $C^{(k)}(k+1\colon n, k+1\colon n)$ according to* (5.41) *as the appropriate columns are available.*

The development above provides most of a proof of the following lemma.

LEMMA 5.7. *Let $C^{(k)}, M^{(k)} \in \Re^{n\times n}, k = 1, \ldots, n-1$, be as defined in* (5.16)–(5.17) *and let $C^{(k,j)}, j = k+1, \ldots, n$, be as defined in* (5.34)*. Excluding the rounding error in applying the Givens rotations in V_k to $C^{(k,k)}$, Algorithm* 5.1 *produces a matrix $C^{(k)}$ such that the matrix satisfies* (5.24) *and*

$$C^{(k)}(k\colon n, k\colon n) = U_k^T[(C^{(k-1)} + \delta C_1^{(k-1)})(k\colon n, k\colon n)]V_k,$$

$$\|\delta C_1^{(k-1)}(k\colon n, j)\|_2 \leq \varepsilon_M h(n-k+1)\|C^{(k-1)}(k\colon n, j)\|_2 + O(\varepsilon_M^2),$$

where $h(n) = ng_0(n) + 12n$ and $g_0(n)$ is defined in Lemma 5.5.

Proof. An interpretation of the error bounds due to Wilkinson [39, pp. 152–162, 236] on the formation and application of a single Householder transformation gives us that $M^{(k)}$ and $C^{(k-1)}$ satisfy

$$(C^{(k-1)} + \delta C_0^{(k-1)})(k\colon n, k\colon n) = U_k M^{(k)}(k\colon n, k\colon n),$$

where

$$\|\delta C_0^{(k-1)}(k\colon n, j)\|_2 \leq 12(n-k+1)\varepsilon_M\|C^{(k-1)}(k\colon n, j)\|_2.$$

Since

$$C^{(k)}(k\colon n, k\colon n) = C^{(k,k)}(k\colon n, k\colon n)V_k = (M^{(k)} + \delta M_2^{(k)})(k\colon n, k\colon n)V_k,$$

we have

$$C^{(k)}(k\colon n, k\colon n) = U_k^T[(C^{(k-1)} + \delta C_1^{(k-1)})(k\colon n, k\colon n)]V_k,$$

where

$$\delta C_1^{(k-1)}(k\colon n, k\colon n) = \delta M_0^{(k)}(k\colon n, k\colon n) + U_k\delta M_2^{(k)}(k\colon n, k\colon n).$$

procedure *compute_pivot*$(C, n, \mathbf{v}, \mathbf{cn}, \mathbf{sn}, s)$

```
%
% Input arguments -- v -- First column of orthogonal transformation.
%                    C -- matrix to be transformed
%                    n -- dimension of C, C is n x n
%
% Output arguments -- cn,sn -- defining values for Givens rotations
%                         s -- column to be replaced implicitly in
%                              orthogonal transformations
```

$maxval = |v(1)| * \|C(:,1)\|_2$; $s = 1$;
for $j = 2{:}\, n$
  $newval = |v(j)| * \|C(:,j)\|_2$;
  $rotg(v(1), v(j), \mathbf{cn}(j-1), \mathbf{sn}(j-1))$;
  if $newval > maxval$
    $maxval = newval$;
    $s = j$;
  end;
end;
end*compute_pivot*

FIG. 5.1. *The compute_pivot procedure.*

Thus from Lemma 5.5, Proposition 5.6, and orthogonal equivalence, we conclude that

$$\|\delta C_1^{(k-1)}(k{:}\, n, j)\|_2 \le \|\delta C_0^{(k-1)}(k{:}\, n, j)\|_2 + \|\delta M_2^{(k)}(k{:}\, n, j)\|_2$$

$$\le \varepsilon_M[(n-k+1)g_0(n-k+1) + 12(n-k+1)]\|C^{(k-1)}(k{:}\, n, j)\|_2 + O(\varepsilon_M^2),$$

thereby establishing the necessary result □

We encapsulate Algorithm 5.1 into two procedures given in a MATLAB-like pseudolanguage. The first, called *compute_pivot* (see Figure 5.1), computes the rotations and the value s, thus doing step 1 of Algorithm 5.1. The second, called *rot_back* (see Figure 5.2), implements the above procedure for applying the Givens rotations, thus doing steps 2–5 of Algorithm 5.1.

Unlike MATLAB, we have procedures, and arguments are called by reference.

REMARK 5.1. *We also use routines* rotg *and* rot*, which correspond to the BLAS* 1 *routines of the same names that generate and apply Givens rotations* [31, 16]. *The call* rotg(a, b, cn, sn) *inputs* a *and* b *to produce* cn *and* sn *such that*

$$\begin{pmatrix} cn & sn \\ -sn & cn \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \rho \\ 0 \end{pmatrix}, \qquad \begin{matrix} \rho = \pm\|(a,b)^T\|_2, \\ cn^2 + sn^2 = 1 \end{matrix}$$

except that rotg$(0, 0, cn, sn)$ *produces* $cn = 0$ *and* $sn = 1$. *For two* n*-vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *the call* $rot(\mathbf{x}, \mathbf{y}, cn, sn)$ *performs the rotation*

$$\mathbf{x} \leftarrow (cn)\mathbf{x} + (sn)\mathbf{y}, \quad \mathbf{y} \leftarrow -(sn)\mathbf{x} + (cn)\mathbf{y}.$$

If the condition in Proposition 5.6 and Lemma 5.7 that excludes the rounding error in applying V_k is lifted, we still obtain a good backward error on the step (5.16)–(5.17).

```
procedure rot_back(C, n, cn, sn, s)

%
% Input arguments -- cn,sn -- Vectors defining Givens rotations
%                        C -- matrix to be transformed
%                        n -- dimension of C, C is n x n
%                        s -- column to be modified in
%                             orthogonal transformations
%
% Transform first row in standard fashion as in step 2.
%
for j = 2:n
    rot(C(1,1), C(1,j), cn(j-1), sn(j-1));
end;
if  s == 1 % The case s = k
    w = 0; % w computes the backward recurrence for column 1.
    for j = n:-1:2
        w = (w - sn(j-1) * C(2:n,j))/cn(j-1);
        C(2:n,j) = cn(j-1) * C(2:n,j) - sn(j-1) * w;
    end;
else % The case s > k
    w = 0;
    for j = n:-1:s+1
        w = (w - sn(j-1) * C(2:n,j))/cn(j-1);
        C(2:n,j) = cn(j-1) * C(2:n,j) - sn(j-1) * w;
    end;
    for j = 2:s-1
        rot(C(2:n,1), C(2:n,j), cn(j-1), sn(j-1));
    end;
    % Construct column s
    C(2:n,s) = (w - cn(s-1) * C(2:n,1))/sn(s-1);
    % Use new column s
    C(2:n,s) = cn(s-1) * C(2:n,s) - sn(s-1) * C(2:n,1);
end;
endrot_back
```

FIG. 5.2. *The rot_back procedure.*

For that, we state the following theorem, which is proved in [2]. Lemma 5.7 is used in the proof of this theorem.

THEOREM 5.8. *Suppose that* $C^{(k)}$ *is computed from* $C^{(k-1)}$ *as in* (5.16)–(5.17) *using Algorithm* 5.1 *to compute* $C^{(k)}$ *from* $M^{(k)}$. *For* $1 \le k < j \le n$, *let*

$$\beta_{k,j} = \max\{\|C^{(k-1)}(k\colon n, j\colon n)\|_F, \|C^{(k)}(k+1\colon n, j\colon n)\|_F\}, \tag{5.48}$$

and $f_1(n) = n^3 + 20n^2 + 12n$. *Then, in floating point arithmetic with machine unit* ε_M, *the computed matrices* $C^{(k-1)}$ *and* $C^{(k)}$ *satisfy*

$$C^{(k)} + \mathbf{e}_{k-1} \begin{pmatrix} \delta\gamma_{k-1} & \delta\phi_k \end{pmatrix} \begin{pmatrix} \mathbf{e}_{k-1}^T \\ \mathbf{e}_k^T \end{pmatrix}$$

$$= E_k \hat{U}_k^T [C^{(k-1)}(k-1\colon n, k\colon n) + \delta C^{(k-1)}] \hat{Q}_k, \tag{5.49}$$

where $\hat{U}_k$ *is orthogonal, the matrices* E_k, $\delta C_k^{(k-1)}$, *and* $\hat{Q}_k$ *and the scalars* $\delta\gamma_{k-1}$ *and* $\delta\phi_k$ *satisfy*

$$\|\delta C^{(k)}(:, j\colon n)\|_F \le \varepsilon_M f_1(n-k+1)\beta_{k,j} + O(\varepsilon_M^2),$$

$$\|E_k - I\|_2 \le 3(n-k+1)^2 \varepsilon_M + O(\varepsilon_M^2),$$

$$\|\hat{Q}_k^T \hat{Q}_k - I\|_2 \le 6(n-k+1)^2 \varepsilon_M + O(\varepsilon_M^2),$$

$$\delta\gamma_0 = \delta\phi_1 = 0,$$

$$|\delta\gamma_{k-1}| \le 3(n-k+1)^2 \varepsilon_M + O(\varepsilon_M^2), \quad |\delta\phi_k| \le (n-k+1)\varepsilon_M + O(\varepsilon_M^2), \quad k \ge 2.$$

The diagonal matrix E_k, the slight nonorthogonality of $\hat{Q}_k$, and errors introduced to γ_{k-1} and ϕ_k are the results of the error analysis techniques necessary to get a columnwise backward error bound on $C^{(k-1)}$. Algorithm 5.1 is critical in ensuring this error bound.

In the next section, we give a simple example that shows the effect of using Algorithm 5.1 in the implementation of one step of bidiagonal reduction.

5.3. A 4 × 4 example. The following 4 × 4 example illustrates how this new Givens-based procedure preserves small singular values better than the Golub–Kahan Householder-based procedure.

EXAMPLE 5.1. *Let* A *be the* 4×4 *matrix*

$$A = \begin{pmatrix} 1 & \zeta_1 & \zeta_1 & 2\zeta_1 \\ 0 & 1/\sqrt{3} & \zeta_2 & \zeta_2 \\ 0 & 1/\sqrt{3} & 2\zeta_2 & \zeta_2 \\ 0 & 1/\sqrt{3} & 3\zeta_2 & 3\zeta_2 \end{pmatrix}, \tag{5.50}$$

where ζ_1 *and* ζ_2 *are small parameters. Using the MATLAB value* $\varepsilon_M = 2.2204e-16$, *we chose* $\zeta_1 = 10\varepsilon_M$ *and* $\zeta_2 = \varepsilon_M/1000$. *That yields a matrix that has two singular values clustered at* 1, *and two distinct singular values smaller than* ε_M.

To the digits displayed,

$$A = \begin{pmatrix} 1 & 2.22045e-15 & 2.22045e-15 & 4.44089e-15 \\ 0 & 0.57735 & 2.22045e-19 & 2.22045e-19 \\ 0 & 0.57735 & 4.44089e-19 & 2.22045e-19 \\ 0 & 0.57735 & 6.66134e-19 & 6.66134e-19 \end{pmatrix}.$$

The singular values are well-conditioned under column scalings. If we let

$$\Delta = \operatorname{diag}(\|A(:,1)\|_2, \|A(:,2)\|_2, \|A(:,3)\|_2, \|A(:,4)\|_2),$$

then

$$\kappa_2(A\Delta^{-1}) = \|A\Delta^{-1}\|_2 \|\Delta A^{-1}\|_2 = 5.1962e+04.$$

Thus according to the theory in [15]*, we should expect that the Jacobi method will compute the singular values of* A *to at least* 11 *digit accuracy. The Jacobi method (coded by Yoon for* [6]*) obtains the SVD*

$$A = U\Sigma V^T,$$

where (to the digits displayed)

$$\Sigma = \mathrm{diag}(1, 1, 4.63692e-19, 1.22778e-19),$$

$$U = \begin{pmatrix} 2.22045e-15 & 1 & -1.01546e-51 & -4.70505e-53 \\ 0.57735 & 1.47911e-33 & -0.553113 & -0.600611 \\ 0.57735 & 1.97215e-33 & -0.243588 & 0.779315 \\ 0.57735 & 4.43734e-33 & 0.7967 & -0.178704 \end{pmatrix},$$

$$V = \begin{pmatrix} -4.55449e-33 & 1 & -4.82373e-15 & 1.17624e-15 \\ 1 & 2.22602e-51 & -9.86268e-19 & -1.72585e-19 \\ 7.69185e-19 & 2.22045e-15 & 0.646375 & 0.76302 \\ 6.40988e-19 & 4.44089e-15 & 0.76302 & -0.646375 \end{pmatrix}.$$

For the bidiagonal reduction procedures below, we did none of the preprocessing in section 3*. The singular values of the bidiagonal matrices were obtained by the bisection procedure described in* [3]*; the singular vectors were obtained by using oqd iterations as in* [21] *until the matrices decoupled into* 2×2 *blocks.*

The dramatic difference between the application of our algorithm and the standard bidiagonal reduction procedure occurs with the application of U_2 *and* V_2*. Algorithm* 4.1 *implemented with Householder transformations yields*

$$N^{(2)} = \begin{pmatrix} 1 & 5.438960e-15 & 0 & 0 \\ 0 & -.2357023 & -.2357023 & -.4714045 \\ 0 & -.2357023 & -.2357023 & -.4714045 \\ 0 & -.2357023 & -.2357023 & -.4.714045 \end{pmatrix} = AV_2,$$

where

$$\|N^{(2)}(3\colon 4, 3\colon 4)\|_F = 0.7454.$$

Note that columns of $N^{(2)}(2\colon 4, 2\colon 4)$ *are all nearly multiples of one another. Thus when we compute*

$$C^{(2)} = \begin{pmatrix} 1 & 5.438960e-15 & 0 & 0 \\ 0 & .4082483 & .4082483 & .8164966 \\ 0 & 0 & -2.775558e-17 & -5.551115e-17 \\ 0 & 0 & -2.775558e-17 & -5.551115e-17 \end{pmatrix} = U_2^T N^{(2)},$$

for which

$$\|C^{(2)}(3\colon 4, 3\colon 4)\|_F = 8.7771e-17,$$

the lower 2×2 *corner of* $C^{(2)}$ *is dominated by the rounding error in this step, and its two columns are colinear. Notice that* $\|N^{(2)}(3\colon 4, 3\colon 4)\|_F$ *is much larger than*

$\|C^{(2)}(3{:}\,4, 3{:}\,4)\|_F$. *This is the concern raised in the text surrounding* (5.18) *and that motivated Algorithm* 5.1.

The final bidiagonal matrix obtained by this procedure is

$$B = \begin{pmatrix} -1 & 5.43896e-15 & 0 & 0 \\ 0 & 0.408248 & -0.912871 & 0 \\ 0 & 0 & -8.77708e-17 & -3.48631e-32 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

It has the singular values

$$\Sigma = \mathrm{diag}(1, 1, 3.58323e-17, 0).$$

The computed singular vector matrices are

$$U = \begin{pmatrix} -0.596931 & -0.802293 & 1.7791e-31 & 0 \\ -0.463204 & 0.344638 & -0.816497 & 0 \\ -0.463204 & 0.344638 & 0.408248 & -0.707107 \\ -0.463204 & 0.344638 & 0.408248 & 0.707107 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.596931 & -0.802293 & -4.96507e-15 & -4.83077e-30 \\ -0.802293 & 0.596931 & -1.11022e-16 & 3.33067e-16 \\ -1.52586e-15 & -1.63233e-15 & 0.447214 & -0.894427 \\ -3.31895e-15 & -3.06585e-15 & 0.894427 & 0.447214 \end{pmatrix}.$$

The invariant subspaces for the double singular value at 1 *and for singular values* 3 *and* 4 *are correct. However, the individual singular vectors for singular values* 3 *and* 4 *are wrong.*

Our algorithm computes the same step

$$M^{(2)} = \begin{pmatrix} 1 & -2.220446e-15 & -2.220446e-15 & -4.440892e-15 \\ 0 & -1 & -7.691851e-18 & -6.409876e-18 \\ 0 & -1.110223e-16 & 8.127397e-19 & -9.384709e-19 \\ 0 & -1.110223e-16 & 3.033186e-18 & 3.502421e-18 \end{pmatrix} = U_2^T A,$$

where

$$\|M^{(2)}(3{:}\,4, 3{:}\,4)\|_F = 4.7967e-18.$$

Then the procedure of Algorithm 5.1 *produces*

$$C^{(2)} = \begin{pmatrix} 1 & -5.438960e-15 & 0 & 0 \\ 0 & -.4082483 & .7071068 & .5773503 \\ 0 & 0 & -1.778109e-19 & -1.625479e-18 \\ 0 & 0 & 9.242744e-18 & 6.066371e-18 \end{pmatrix} = M^{(2)}V_2,$$

which has

$$\|C^{(2)}(3{:}\,4, 3{:}\,4)\|_F = 1.1176e-17,$$

and the columns of $C^{(2)}(3{:}\,4, 3{:}\,4)$ *are linearly independent. Mathematically,* $\|M^{(2)}(3{:}\,4, 2{:}\,4)\|_F = \|C^{(2)}(3{:}\,4, 3{:}\,4)\|_F$, *but, in fact,*

$$\|M^{(2)}(3{:}\,4, 2{:}\,4)\|_F = 1.5708e-16,$$

which is more than 10 *times as large. The value of* s *in Algorithm* 5.1 *is* 2 *for this step. The values in* $M^{(2)}(3\colon 4, 2)$ *are mostly rounding error from computing* $M^{(2)}$*. Step* 3 *of Algorithm* 5.1 *recomputes these components of column* 2 *from columns* 3 *and* 4 *so that*

$$C^{(2,2)}(3\colon 4, 2\colon 4)C^{(2)}(1, 2\colon 4)^T = 0,$$

as is expected.

The resulting bidiagonal matrix for our Givens procedure (to the digits displayed) is

$$B = \begin{pmatrix} -1 & -5.43896e-15 & 0 & 0 \\ 0 & -0.408248 & 0.912871 & 0 \\ 0 & 0 & 1.10577e-18 & -1.01936e-19 \\ 0 & 0 & 0 & -1.26113e-19 \end{pmatrix}.$$

The matrix B *has the same singular values as those obtained by Jacobi to* 15 *significant digits. Thus there can be no important difference in their quality. The corresponding singular vectors are*

$$U = \begin{pmatrix} -0.633989 & -0.773342 & 2.23672e-33 & -1.44416e-34 \\ -0.446489 & 0.366034 & 0.553113 & 0.600611 \\ -0.446489 & 0.366034 & 0.243588 & -0.779315 \\ -0.446489 & 0.366034 & -0.7967 & 0.178704 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.633989 & -0.773342 & 4.82373e-15 & -1.17624e-15 \\ -0.773342 & 0.633989 & -1.10294e-16 & 4.3525e-17 \\ -1.53653e-15 & -1.61158e-15 & -0.646375 & -0.76302 \\ -2.72962e-15 & -3.50472e-15 & -0.76302 & 0.646375 \end{pmatrix}.$$

The first two singular vectors correspond to a cluster at 1*; thus they cannot be expected to be the same, but the subspaces associated with the leading two singular values are computed correctly. The third and fourth singular vectors are the same as obtained by Jacobi to* 15 *digits.*

If $C^{(2)}$ *above is computed from* $M^{(2)}$ *using an ordinary implementation of Givens rotations, we obtain*

$$C^{(2)} = \begin{pmatrix} 1 & -5.43896e-15 & 0 & 0 \\ 0 & -0.408248 & 0.707107 & 0.57735 \\ 0 & -4.53681e-17 & 7.85621e-17 & 6.39977e-17 \\ 0 & -4.49149e-17 & 7.87191e-17 & 6.41258e-17 \end{pmatrix}.$$

Of course, we set $C^{(2)}(3\colon 4, 2) = 0$*, but that neglects a column of* 2*-norm* $6.3841e-17$ *which is of the magnitude of the singular values in lower* 2×2 *corner of* C*.*

If we continue the computation this way, the resulting singular values are

$$\Sigma = \operatorname{diag}(1, 1, 5.85614e-17, 1.62027e-22).$$

Singular values 3 *and* 4 *are wrong. The singular vectors matrices are*

$$U = \begin{pmatrix} -0.615412 & -0.788205 & 0 & 0 \\ -0.455071 & 0.355308 & 0.816496 & 0.00081583 \\ -0.455071 & 0.355308 & -0.407542 & -0.707514 \\ -0.455071 & 0.355308 & -0.408955 & 0.706699 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.615412 & -0.788205 & 4.96505e-15 & -1.37484e-17 \\ -0.788205 & 0.615412 & -1.11022e-16 & 2.1684e-19 \\ -1.49775e-15 & -1.64768e-15 & -0.449689 & -0.893185 \\ -2.64547e-15 & -3.56866e-15 & -0.893185 & 0.449689 \end{pmatrix}.$$

As in the case of the Householder algorithm, the singular vectors associated with the clustered singular value at 1 *are correct, but the individual singular vectors associated with singular values* 3 *and* 4 *are wrong. This suggests that Algorithm* 5.1 *makes a difference in how well the bidiagonal reduction is computed.*

We also tried this example with Householder transformations where the transformation U_k and V_k were performed in reverse order. The results were similar to those for Givens method implemented without using Algorithm 5.1.

We have now specified everything in Algorithm 4.1, and thus we give our proposed bidiagonal reduction algorithm in the next section.

6. A Givens-based bidiagonal reduction algorithm and its backward error.

6.1. Statement of the algorithm. We now present a Givens-based bidiagonal reduction procedure for an $n \times n$ matrix C satisfying (3.6). In section 6.2, we show that this new algorithm will achieve error bounds of the form (1.5)–(1.8).

ALGORITHM 6.1 (new procedure for bidiagonal reduction).

1. *Let U_1 be an orthogonal transformation such that*

$$U_1^T C(:,1) = \gamma_1 \mathbf{e}_1,$$

$$C \leftarrow U_1^T C, \quad U \leftarrow U_1, \quad V \leftarrow V_1 \equiv I.$$

2. ***for*** $k = 2{:}\, n-1$

   (a) *If $C(k-1, k{:}\, n) \neq 0$, let V_k be the product of Givens rotations*

$$V_k = V_{k,k+1} \cdots V_{k,n}, \quad V_{k,j} = J(k, j, \theta_{kj})$$

   that satisfies

$$V_k^T C(k-1, k{:}\, n)^T = \pm \phi_k \mathbf{e}_1,$$

   else $V_k = I$ (implicitly). Compute

$$\mathbf{y}_k = C(k{:}\, n, k{:}\, n) V_k \mathbf{e}_1, \quad V(:, k{:}\, n) \leftarrow V(:, k{:}\, n) V_k.$$

   (b) *Find an orthogonal transformation U_k such that*

$$U_k^T \mathbf{y}_k = \gamma_k \mathbf{e}_1,$$

$$C(k{:}\, n, k{:}\, n) \leftarrow U_k^T C(k{:}\, n, k{:}\, n), \quad U(:, k{:}\, n) \leftarrow U(:, k{:}\, n) U_k.$$

   (c) *If $C(k-1, k{:}\, n) \neq 0$, let $\mathbf{v}_1^{(k)} = C(k-1, k{:}\, n)^T / \phi_k$ and compute*

$$C(k{:}\, n, k{:}\, n) \leftarrow C(k{:}\, n, k{:}\, n) V_k$$

   using the two calls

compute_pivot$(C(k\colon n, k\colon n), n-k+1, \mathbf{v}_1^{(k)}, \mathbf{cn}, \mathbf{sn}, s)$;
rot_back$(C(k\colon n, k\colon n), n-k+1, \mathbf{cn}, \mathbf{sn}, s)$;

3.

$$\gamma_n \leftarrow C(n,n), \quad \phi_n \leftarrow C(n-1,n)$$

The bidiagonal reduction of C is given by

$$C = UBV^T,$$

where

$$B = \text{bidiag}(\gamma_1, \ldots, \gamma_n; \phi_2, \ldots, \phi_n).$$

Table 6.1 summarizes the complexity of the two bidiagonal reduction algorithms. The extra operations in Algorithm 6.1 over Algorithm 4.1 are the computation of the column norms hidden in the routine *compute_pivot*, and the use of Givens rotations instead of Householder transformations.

TABLE 6.1
Complexity of bidiagonal reduction algorithms.

Compute U,V?	Algorithm 4.1(H)	Algorithm 6.1
Yes	$\frac{20}{3}n^3$	$\frac{26}{3}n^3$
No	$\frac{8}{3}n^3$	$4n^3$

H – with Householder transformations

6.2. Error bounds on Algorithm 6.1. The error bounds for this paper are stated in two theorems.

THEOREM 6.1. *Let $C \in \Re^{n\times n}$ and let $B = \text{bidiag}(\gamma(1\colon n); \phi(2\colon n)) \in \Re^{n\times n}$ be the bidiagonal matrix computed by Algorithm* 6.1 *in floating point arithmetic with machine precision ε_M. Let $C^{(k)}, k = 1, \ldots, n-1$, be the contents of C after k passes thorough the main loop of Algorithm* 6.1. *Then there exist $U, V \in \Re^{n\times n}$ and modestly growing functions $g_i(\cdot)$, $i = 1, 2, 3$, such that*

$$C + \delta C = U(B + \delta B)V^T, \tag{6.1}$$

where

$$\|U^TU - I\|_2, \|V^TV - I\|_2 \le g_1(n)\varepsilon_M + O(\varepsilon_M^2), \tag{6.2}$$

$$|\delta B| \le g_2(n)\varepsilon_M|B| + O(\varepsilon_M^2), \quad |\cdot| \text{ and } \le \text{ entry-wise}, \tag{6.3}$$

and for $j = 1, \ldots, n$

$$\|\delta C(:, j\colon n)\|_F \le \varepsilon_M g_3(n) \max_{0\le k<j} \|C^{(k)}(k+1\colon n, j\colon n)\|_F \tag{6.4}$$

$$\le \varepsilon_M g_3(n)\hat{\rho}_V^{(j-1)}\|C(:, j\colon n)\|_F + O(\varepsilon_M^2),$$

where $\hat{\rho}_V^{(j-1)}$ *is defined by* (2.9).

Proof. This proof is an induction argument on Theorem 5.8. Using the notation from that theorem, we let

$$\tilde{U}_k = \hat{U}_1 E_2^{-1} \hat{U}_2 \cdots \hat{U}_k E_k^{-1},$$

$$\bar{U}_k = \hat{U}_k E_k \hat{U}_{k+1} E_{k+1} \cdots \hat{U}_{n-1} E_{n-1},$$

$$U = \tilde{U}_{n-1} = \bar{U}_1,$$

$$\tilde{Q}_k = Q_2^{-T} \cdots Q_k^{-T},$$

$$\bar{Q}_k = Q_{k+1} \cdots Q_{n-1}, \quad \bar{Q}_n = I,$$

$$V = \tilde{Q}_{n-1} = \bar{Q}_1^{-T}.$$

Clearly, by an induction argument on the bounds on Q_k and E_k from Theorem 5.8, we have that

$$\|V^T V - I\|_2, \|U^T U - I\|_2 \leq 2n^3 \varepsilon_M + O(\varepsilon_M^2),$$

which establishes (6.2).

An induction argument on the backward error bound in Theorem 5.8 yields

$$C + \delta C = U(B + \delta B)V^T,$$

where

$$\delta C = \sum_{k=1}^{n-1} \tilde{U}_k \delta C^{(k)} \tilde{Q}_{k-1}^T$$

and

$$\delta B = \sum_{k=1}^{n-1} \bar{U}_k^T \mathbf{e}_{k-1} \begin{pmatrix} \delta\gamma_{k-1} & \delta\phi_k \end{pmatrix} \begin{pmatrix} \mathbf{e}_{k-1}^T \\ \mathbf{e}_k^T \end{pmatrix} \bar{Q}_k.$$

Clearly, $\bar{U}_k$ and $\bar{Q}_k$ have no effect on the terms of the sum constituting δB, so

$$\delta B = \sum_{k=1}^{n-1} \mathbf{e}_{k-1} \begin{pmatrix} \delta\gamma_{k-1} & \delta\phi_k \end{pmatrix} \begin{pmatrix} \mathbf{e}_{k-1}^T \\ \mathbf{e}_k^T \end{pmatrix}$$

$$= \text{bidiag}(\delta\gamma_1, \ldots, \delta\gamma_{n-1}, 0; \delta\phi_2, \ldots, \delta\phi_n),$$

where from Theorem 5.8,

$$|\delta\gamma_k| \leq 3(n-k+2)^2 \varepsilon_M |\gamma_k| + O(\varepsilon_M^2), \quad |\delta\phi_k| \leq 3(n-k+1)\varepsilon_M |\gamma_k| + O(\varepsilon_M^2)$$

for all appropriate k. We may write (conservatively) that

$$|\delta B| \leq 3(n+1)^2 \varepsilon_M |B| + O(\varepsilon_M^2),$$

thereby establishing (6.3).

To prove (6.4), we note that

$$\begin{aligned}\|\delta C(:, j{:}n)\|_F &\leq \sum_{k=1}^{n-1} \|\tilde{U}_k \delta C^{(k)} \tilde{Q}_k^T \begin{pmatrix} 0 \\ I_{n-j+1} \end{pmatrix} \|_F + O(\varepsilon_M^2) \\ &= \sum_{k=1}^{n-1} \|\delta C^{(k)} \tilde{Q}_k^T \begin{pmatrix} 0 \\ I_{n-j+1} \end{pmatrix} \|_F + O(\varepsilon_M^2).\end{aligned}$$

The structure of the Givens rotations in the algorithm give us that

$$\tilde{Q}_k^T = Q_{k-1}^{-1} \cdots Q_1^{-1} \begin{pmatrix} 0 \\ I_{n-j+1} \end{pmatrix} = \begin{matrix} k-1 \\ j-k+1 \\ n-j+1 \end{matrix} \begin{pmatrix} J_1 \\ 0 \\ J_2 \end{pmatrix} = J. \tag{6.5}$$

Noting that

$$\delta C(1{:}\,k-1, :) = 0, \quad \delta C(:, 1{:}\,k-1) = 0,$$

we have that

$$\|\delta C^{(k)}(:, j{:}n) Q_{k-1}^T \begin{pmatrix} 0 \\ I_{n-j+1} \end{pmatrix} \|_F = \|\delta C^{(k)}(:, j{:}n) J_2\|_F \leq \|\delta C^{(k)}(:, j{:}n)\|_F.$$

Therefore,

$$\|\delta C(:, j{:}n)\|_F \leq \sum_{k=1}^{n} \|\delta C^{(k)}(:, j{:}n)\|_F.$$

Using the bound on $\|\delta C^{(k)}(:, j{:}n)\|_F$ and the definition of $\beta_{k,j}$ from Theorem 5.8, we have

$$\begin{aligned}\|\delta C(:, j{:}n)\|_F &\leq \varepsilon_M \sum_{k=1}^{n-1} f_1(n-k+1)\beta_{k,j} + O(\varepsilon_M^2) \\ &\leq \varepsilon_M g_3(n) \max_{1 \leq k < j \leq n} \beta_{k,j} + O(\varepsilon_M^2),\end{aligned}$$

where

$$g_3(n) = \frac{n^4}{4} + \frac{20n^3}{3} + 6n.$$

From the definition of $\beta_{k,j}$ we have that

$$\max_{1 \leq k < j} \beta_{k,j} = \max_{1 \leq k < j} \|C^{(k)}(k+1{:}\,n, j{:}n)\|_F \leq \hat{\rho}_V^{(j-1)} \|C(:, j{:}n)\|_F.$$

That establishes (6.4) and the theorem. ∎

Theorem 6.1 uses none of the properties of C from section 3, the backward error bound above applies to Algorithm 6.1 applied to any nonsingular matrix. Equation (3.6) allows us to obtain a columnwise bound expressed in the next theorem.

THEOREM 6.2. *Let $C \in \Re^{n\times n}$ satisfy* (3.6)*, and let $B = \text{bidiag}(\gamma(1\colon n); \phi(2\colon n)) \in \Re^{n\times n}$ be the bidiagonal matrix computed by Algorithm* 6.1 *in floating point arithmetic with machine precision ε_M. Let $C^{(k)}, k = 1, \ldots, n-1$, be the contents of C after k passes through the main loop of Algorithm* 6.1*, and let $\hat{\rho}_V^{(k)}$ be defined by* (5.11)*. Then there exist $U, V \in \Re^{n\times n}$ satisfying* (6.2) *and a modestly growing function $g_4(\cdot)$ such that B and C satisfy* (6.1)*, δB is as in Theorem* 6.1*, and*

$$\|\delta C(:,j)\|_2 \le \varepsilon_M \hat{\rho}_V^{(j-1)} g_4(n) \|C(:,j)\|_2 + O(\varepsilon_M^2), \tag{6.6}$$

where

$$\hat{\rho}_V^{(j-1)} = \min\{\rho_V^{(j-1)}, \|C\|_F / \|C(:,j\colon n)\|_F\}.$$

Proof. The result follows from our bounds on

$$\|C^{(k)}(k+1\colon n, j\colon n)\|_F.$$

A combination of Theorem 6.1 and (3.6) leads to

$$\begin{aligned}
\|\delta C(:,j)\|_2 &\le \|\delta C(:,j\colon n)\|_F \\
&\le \varepsilon_M g_3(n) \|C^{(k)}(k+1\colon n, j\colon n)\|_F + O(\varepsilon_M^2) \\
&\le \varepsilon_M \hat{\rho}_V^{(j-1)} g_3(n) \|C(:,j\colon n)\|_F + O(\varepsilon_M^2) \\
&\le \varepsilon_M \hat{\rho}_V^{(j-1)} \sqrt{n} g_3(n) \|C(:,j)\|_2 + O(\varepsilon_M^2) \\
&= \varepsilon_M \hat{\rho}_V^{(j-1)} g_4(n) \|C(:,j)\|_2, \quad g_4(n) = \sqrt{n} g_3(n) + O(\varepsilon_M^2). \qquad \square
\end{aligned}$$

From Theorem 6.2, we have the conditional columnwise error bound necessary to establish error bounds on the singular values and vectors as discussed in section 2.

7. Numerical tests. We performed three sets of numerical tests on the bidiagonal reduction algorithms. Three separate routines were used to find the SVD of the matrices in the test sets, as follows.

- *The Jacobi algorithm.* The Jacobi method described in [15]. In Figures 7.1 and 7.2, this is referred to as *Jacobi alg.*
- *The Givens algorithm.* The bidiagonalization method of Algorithm 6.1 followed by the bisection routine of Demmel and Kahan [14]. In Figures 7.1, 7.2, and 7.3, this is referred to as *Givens alg.*
- *The Householder algorithm.* The bidiagonalization method of Algorithm 4.1 using Householder transformations followed by the same bisection routine. In Figures 7.1, 7.2, and 7.3, this is referred to as *Householder alg.*

Our first two sets of examples are constructed using the Kahan matrices [29]. Let $\hat{C}$ be the $n \times n$ lower triangular matrix

$$\hat{C} = (\hat{c}_{ij}), \quad \hat{c}_{ij} = \begin{cases} \alpha^{i-1}, & i = j, \\ -\alpha^{i-1}\beta, & i > j, \end{cases} \tag{7.1}$$

where $\alpha^2 + \beta^2 = 1$ and $\alpha, \beta > 0$. If we chose α bounded away from zero or one, we obtain a matrix that is unaltered by QR factorization with column pivoting, has slowly decaying diagonals, but has a condition number that grows rapidly with n.

This matrix has one isolated small singular value [40], $\sigma_n(\hat{C})$ in (7.3). The other singular values $\sigma_j(\hat{C}), k = 1, \ldots, n-1$, are close enough to $\|\hat{C}\|_2$ that standard SVD software will compute them to nearly machine relative accuracy. The smallest singular value $\sigma_n(\hat{C})$ may be recovered from

$$\sigma_n(\hat{C}) = \frac{|\det(\hat{C})|}{\prod_{j=1}^{n-1} \sigma_j(\hat{C})} = \frac{|\prod_{j=1}^{n} \hat{c}_{jj}|}{\prod_{j=1}^{n-1} \sigma_j(\hat{C})}. \tag{7.2}$$

Since the formula (7.2) uses only multiplications and divisions, in the absence of underflow, it will compute $\sigma_n(\hat{C})$ to nearly machine relative accuracy.

We used the formula (7.2) to construct both of the first two test sets given next.

For each matrix, we also computed the values

$$\rho_V = \max_{1 \le k \le n-1} \rho_V^{(k)}, \quad \hat{\rho}_V = \max_{1 \le k \le n-1} \hat{\rho}_V^{(k)},$$

$$totratio = \max_{1 \le k < j \le n} \frac{\|C^{(k)}(k+1\colon n, j\colon n)\|_F}{\|C(\colon, j\colon n)\|_F}.$$

Of course, we expect that $\rho_V \ge \hat{\rho}_V \ge totratio$. In practice, the gaps separating these three quantities were very large.

EXAMPLE 7.1 (test set 1). *We let $\hat{C}$ in* (7.1) *be constructed with $\beta = 0.3$ and $n = 50$. Let* $\mathbf{w}$ *be the right singular vector such that*

$$\hat{C}\mathbf{w} = \sigma_n(\hat{C})\mathbf{y}, \quad \|\mathbf{w}\|_2 = \|\mathbf{y}\|_2 = 1, \tag{7.3}$$

where $\sigma_n(\hat{C})$. We then constructed the twenty (20) 51×51 matrices defined by

$$C_j = \begin{pmatrix} \hat{C} & 0 \\ \zeta \mathbf{w}^T & \xi_j \end{pmatrix}, \quad j = 1, \ldots, 20, \tag{7.4}$$

where

$$\zeta = 0.5\hat{c}_{50,50}, \quad \xi_j = \sigma_{50}(\hat{C})/100^{j-1}.$$

The smallest singular value of C_j is also the smallest singular value of the 2×2 matrix

$$D_j = \begin{pmatrix} \sigma_{50}(\hat{C}) & 0 \\ \zeta & \xi_j \end{pmatrix}.$$

Given $\sigma_{50}(\hat{C})$ to nearly machine relative accuracy, we can find the smallest singular value of D_j to nearly machine relative accuracy. We used this value as the exact value of $\sigma_{51}(C_j)$.

We then computed $\sigma_{51}(C_j)$ using the all three of the algorithms given above. The results are represented in Figure 7.1. *The values posted are*

$$\log_{10}\left(\frac{|\tilde{\sigma}_{51}(C_j) - \sigma_{51}(C_j)|}{\sigma_{51}(C_j)}\right),$$

where $\tilde{\sigma}_{51}(C_j)$ and $\sigma_{51}(C_j)$ are the computed and "exact" 51st singular values of C_j. From Figure 7.1, *we can see that the Jacobi and Givens algorithms always compute σ_{51} to at least* 10 *digit accuracy in IEEE double precision. There is no significant*

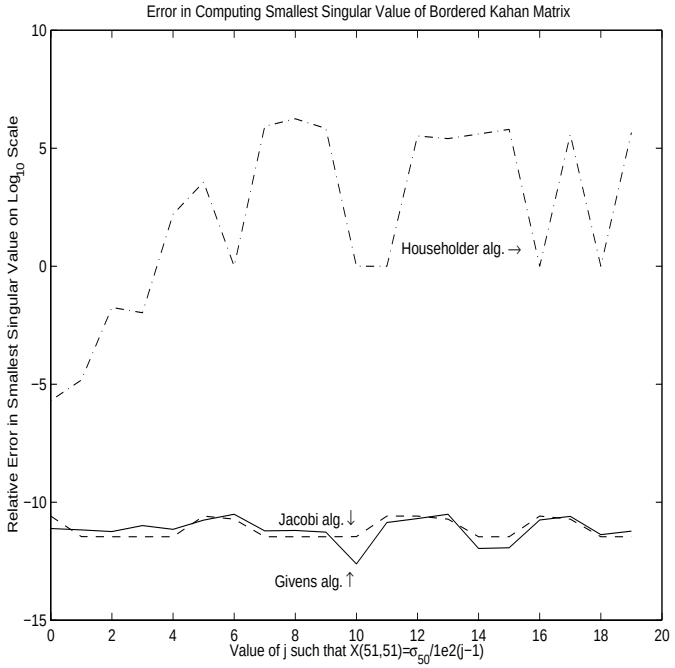


FIG. 7.1. *Relative error from Example* 7.1.

difference in the accuracy of the Jacobi and Givens algorithms on this test set. The Householder algorithm often obtains no accurate digits at all.

The values of ρ_V,$\hat{\rho}_V$ *and* totratio *are given in Table* 7.1. *In this case, the values of* ρ_V *are gross overestimates of the growth factor* totratio *which never exceeds* 4.

EXAMPLE 7.2 (test set 2). *For the second test set, we computed the lower triangular matrices* $C_n, n = 50, 60, \ldots, 200$, *from the QR factorization*

$$\hat{C}_n = QC_n^T,$$

where $\hat{C}_n$ *is a Kahan matrix of size* n *with* $\beta = 0.3$. *Once again this matrix has exactly one small singular value which we compute using the formula* (7.2) *applied to* C_n.

The results represented in Figure 7.2 *are the values*

$$\log_{10}\left(\frac{|\tilde{\sigma}_n(C_n) - \sigma_n(C_n)|}{\sigma_n(C_n)}\right),$$

where $\sigma_n(C_n)$ *is the value computed from the formula* (7.2) *(presumed to be exact) and* $\tilde{\sigma}_n(C_n)$ *computed by one of the three algorithms.*

Similar to the results for Example 7.1, *the Jacobi and Givens algorithms compute* $\sigma_n(C_n)$ *to* 11*-digit accuracy and, once again, there is no significant difference in their accuracy. The Householder algorithm fares a bit better than in Example* 7.1, *but still obtains no more than 5-digit accuracy.*

The values of ρ_V, $\hat{\rho}_V$, *and* totratio *are given in Table* 7.2. *In this class of examples* totratio *was large for some of the smaller problems—it was at* 1.7×10^5 *for* $n = 50$*—but is much smaller for the larger dimensions. Again both* ρ_V *and* $\hat{\rho}_V$ *are gross*

TABLE 7.1
Growth factors for bordered Kahan matrices.

j	ρ_V	$\hat{\rho}_V$	$totratio$
1	1.25434e+009	1.45144e+007	3.86412
2	1.25325e+009	9.7244e+008	3.40234
3	1.25299e+009	8.80033e+006	3.40234
4	1.25299e+009	88066.1	3.40234
5	1.25299e+009	880.661	3.40234
6	1.25299e+009	36.4059	3.40234
7	1.25299e+009	36.4059	3.40234
8	1.25299e+009	36.4059	3.40234
9	1.25299e+009	36.4059	3.40234
10	1.25299e+009	36.4059	3.40234
11	1.25299e+009	36.4059	3.40234
12	1.25299e+009	36.4059	3.40234
13	1.25299e+009	36.4059	3.40234
14	1.25299e+009	36.4059	3.40234
15	1.25299e+009	36.4059	3.40234
16	1.25299e+009	36.4059	3.40234
17	1.25299e+009	36.4059	3.40234
18	1.25299e+009	36.4059	3.40234
19	1.25299e+009	36.4059	3.40234
20	1.25299e+009	36.4059	3.40234

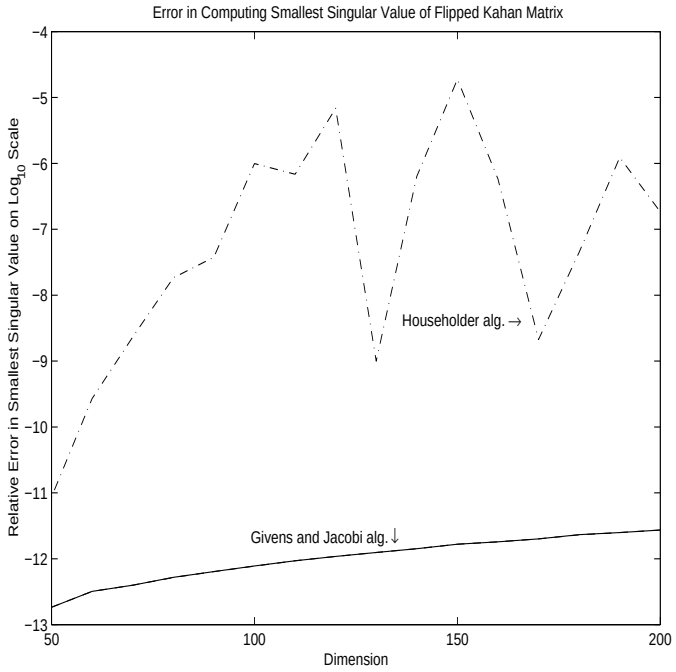


FIG. 7.2. *Relative error from Example* 7.2.

overestimates. For all of the matrices Algorithm 6.1 *computes the smallest singular value correctly.*

The values of $\hat{\rho}_V$ *behave somewhat erratically. It is at about* 10^5 *or* 10^6 *for the first three matrices and hovers about* 10^4 *for the remaining examples. We do not have*

TABLE 7.2
Growth factors for flipped Kahan matrices.

n	ρ_V	$\hat{\rho}_V$	*totratio*
50	840495	279473	171710
60	3.99297e+006	3.99297e+006	89542
70	9.55389e+006	395549	6858.24
80	2.88479e+007	12120.4	507.934
90	5.73345e+008	239.548	25.5923
100	3.13879e+009	404.636	1.39048
110	1.42799e+009	1430.55	1.39033
120	8.87921e+008	2507.13	1.61819
130	5.31115e+008	4181.68	1.61266
140	1.1211e+009	6096.04	1.61983
150	1.06992e+009	11534.8	1.65989
160	1.85538e+009	7503.35	1.58612
170	2.94431e+008	17075.8	1.52791
180	6.5911e+008	20437	1.64403
190	1.03768e+010	33647.4	1.61659
200	8.45255e+008	55320.1	1.55874

a ready explanation for this behavior and it does not seem to affect the accuracy of the algorithm.

These two test sets clearly demonstrate that significant accuracy may be gained through the use of Algorithm 6.1. We found no class of examples where the Jacobi algorithm obtained significantly better accuracy for matrices C of the form (3.1) than did the Givens algorithm.

Interestingly, there were many examples of badly scaled matrices resulting from (3.1) with very small singular values where the Householder algorithm computed the singular values very accurately. The following test set is such a case.

EXAMPLE 7.3 (test set 3). *We used the set*

$$R_k, \quad k = 25, \ldots, 90,$$

where R_k was the Cholesky factor of the Hilbert matrix of dimension k. That is, R_k was the upper triangular matrix with positive diagonals that satisfied

$$H_k = R_k^T R_k,$$

where H_k is a $k \times k$ matrix whose (i, j) entry is $h_{ij} = (i + j - 1)^{-1}$. The matrix R_k is computed from formulas given by Choi [10]. *We then computed the matrices $C_k, k = 25, \ldots, 90$, for input to the three algorithms using the Cox–Higham* [11] *procedure in section* 3 *applied to R_k^T.*

We note that for every matrix C_j in this test set the value $\kappa_2(CD_C^{-1})$ in (2.12) *is less than* 200. *From* [15], *the Jacobi method will compute all of singular values of each C_j to near machine relative accuracy.*

For this test set, we do not present a table of the growth factors ρ_V, $\hat{\rho}_V$, totratio. Instead we simply note that for each matrix in this set, $\rho_V < 212$, $\hat{\rho}_V < 13.5$, and totratio < 4. Thus no significant growth occurred in the columns of these matrices.

For the matrices in Example 7.3, there are no fast formulas for computing the smallest singular value to relative accuracy, thus we use the results of Jacobi algorithm as exact to test the two bidiagonal reduction algorithms.

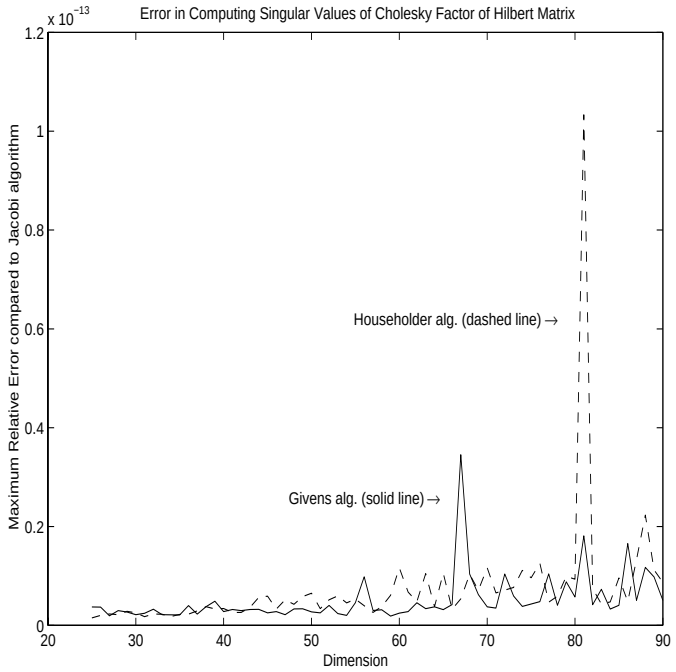


FIG. 7.3. *Relative error from Example* 7.3.

For each matrix in that set, we calculated the two ratios

$$max\_error_G = \max_{1 \le i \le n} \frac{|\sigma_i^G - \sigma_i^J|}{\sigma_i^J},$$

$$max\_error_H = \max_{1 \le i \le n} \frac{|\sigma_i^H - \sigma_i^J|}{\sigma_i^J},$$

where σ_i^J, σ_i^G, and σ_i^H are the ith singular values as calculated by Jacobi, Givens, and Householder algorithms, respectively. Thus we were trying to measure how well the SVDs calculated from the two bidiagonal reduction algorithms agreed with that from the Jacobi algorithm. It can be quickly seen from Figure 7.3 that there is no important difference in the accuracy of the three algorithms on this test set. The good behavior of the Jacobi and Givens algorithms can be explained, but to the author's knowledge there is no satisfactory explanation in the literature for the good behavior of the Householder algorithm.

8. Conclusion. We have constructed a new bidiagonal reduction algorithm (Algorithm 6.1) that allows us to compute the SVD of matrices resulting from (3.1) with more guaranteed accuracy. The accuracy guarantee is not quite as good as the one for Jacobi methods, but we give up little of the speed advantage of bidiagonal reduction methods.

The results given above raise an important unanswered question. The fact that Algorithm 6.1 is based upon 2×2 orthogonal transformations in standard order is used in the proofs of Theorems 5.4 and 6.1. It is not known whether an algorithm

based upon Householder transformations (of size greater than 2×2) or one based on other Givens orderings could yield similar error bounds.

Acknowledgments. The author would like to acknowledge helpful suggestions from Ivan Slapničar, Zlatko Drmač, and Alicja Smoktunowicz. Nick Higham and two anonymous and patient referees made many useful suggestions on earlier versions of this paper.

REFERENCES

[1] J.L. Barlow, *Stability analysis of the G-algorithm and a note on its application to sparse least squares problems*, BIT, 25 (1985), pp. 507–520.

[2] J.L. Barlow, *More Accurate Bidiagonal Reduction for Computing the Singular Value Decomposition*, Technical report, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, 2001; also available online from http://www.cse.psu.edu/∼barlow/bidiag-final4.ps.

[3] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[4] J.L. Barlow and S.L. Handy, *The direct solution of weighted and equality constrained least-squares problems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 704–716.

[5] J.L. Barlow and I. Slapničar, *Optimal perturbation bounds for the Hermitian eigenvalue problem*, Linear Algebra Appl., 309 (2000), pp. 19–43.

[6] J.L. Barlow, P.A. Yoon, and H. Zha, *An algorithm and a stability theory for downdating the ULV decomposition*, BIT, 36 (1996), pp. 14–40.

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

[8] P.A. Businger and G.H. Golub, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–278.

[9] T.F. Chan, *An improved algorithm for computing the singular value decomposition*, ACM Trans. Math. Software, 8 (1982), pp. 72–83.

[10] M.D. Choi, *Tricks or treats with the Hilbert matrix*, Amer. Math. Monthly, 90 (1983), pp. 301–312.

[11] 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 Conference, D.F. Griffiths, D.J. Higham, and G.A. Watson, eds., Addison-Wesley-Longman, Harlow, Essex, UK, 1998, pp. 57–73.

[12] 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, 1999.

[13] J.W. Demmel and W.B. Gragg, *On computing accurate singular values and eigenvalues of matrices with acyclic graphs*, Linear Algebra Appl., 185 (1993), pp. 203–217.

[14] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

[15] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[16] D.S. Dodson and R.G. Grimes, *Remark on algorithm* 539, ACM Trans. Math. Software, 8 (1982), pp. 403–405.

[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] K.V. Fernando, *Accurate BABE Factorisation of Tri-Diagonal Matrices for Eigenproblems*, Technical Report TR5, Numerical Algorithms Group Ltd., Oxford, UK, 1995.

[19] K.V. Fernando, *On computing an eigenvector of a tridiagonal matrix. Part* I: *Basic results*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1013–1034.

[20] K.V. Fernando, *Accurately counting singular values of bidiagonal matrices and eigenvalues of skew-symmetric tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 373–399.

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

[22] 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.

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

[24] G.H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numer. Math, 14 (1970), pp. 403–20.
[25] R.J. Hanson and C.L. Lawson, *Extensions and applications of the Householder algorithm for solving linear least squares problems*, Math. Comp., 23 (1969), pp. 787–812.
[26] N.J. Higham, *QR factorization with complete pivoting and accurate computation of the SVD*, Linear Algebra Appl., 309 (2000), pp. 153–174.
[27] R.A. Horn and C.A. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[28] C.G.J. Jacobi, *Über ein Leichtes Verfahren Die in der Theorie der Sacularstorungen Vorkommendern Gleichungen Numerisch Aufzulosen*, Crelle's J., 30 (1848), pp. 51–94.
[29] W. Kahan, *Numerical linear algebra*, Canad. Math. Bull., 9 (1966), pp. 757–801.
[30] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[31] C.L. Lawson, R.J. Hanson, D.R. Kincaid, and F.T. Krogh, *Basic linear algebra subprograms for FORTRAN usage*, ACM Trans. Math. Software, 5 (1979), pp. 308–325.
[32] R.-C. Li, *Relative perturbation theory:* II. *Eigenspace and singular value variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.
[33] The Mathworks, Inc., *MATLAB: The Language of Technical Computing—Using MATLAB*, The Mathworks, Inc., Natick, MA, 1996.
[34] B.N. Parlett, *Analysis of algorithms for reflectors in bisectors*, SIAM Rev., 13 (1971), pp. 197–208.
[35] B.N. Parlett and I.S. Dhillon, *Relatively robust representations of symmetric tridiagonals*, Linear Algebra Appl., 309 (2000), pp. 121–151.
[36] B.N. Parlett and O.A. Marques, *An implementation of the dqds algorithm (positive case)*, Linear Algebra Appl., 309 (2000), pp. 217–259.
[37] M.J.D. Powell and J.K. Reid, *On applying Householder transformations to linear least squares problems*, in Information Processing, Proceedings of the IFIP Congress, Edinburgh, 1968, Vol. 1, North-Holland, Amsterdam, 1969, pp. 122–126.
[38] A. van der Sluis, *Condition numbers and equilibrium matrices*, Numer. Math., 15 (1969), pp. 74–86.
[39] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.
[40] H. Zha, *Singular values of a classical matrix*, Amer. Math. Monthly, 104 (1997), pp. 172–173.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 799–818
© 2002 Society for Industrial and Applied Mathematics

APPROXIMATION OF THE DETERMINANT OF LARGE SPARSE SYMMETRIC POSITIVE DEFINITE MATRICES*

ARNOLD REUSKEN†

Abstract. This paper is concerned with the problem of approximating $\det(A)^{1/n}$ for a large sparse symmetric positive definite matrix A of order n. It is shown that an efficient solution of this problem is obtained by using a sparse approximate inverse of A. The method is explained and theoretical properties are discussed. The method is ideal for implementation on a parallel computer. Numerical experiments are described that illustrate the performance of this new method and provide a comparison with Monte Carlo–type methods from the literature.

Key words. determinant, sparse approximate inverse, preconditioning

AMS subject classifications. 6F10, 65F10, 65F50

PII. S089547980036869X

1. Introduction.

Throughout this paper, A denotes a real symmetric positive definite matrix of order n with eigenvalues

$$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \ .$$

In a number of applications, for example, in lattice quantum chromodynamics [14, 8, 16, 17] certain functions of the determinant of A, such as $\det(A)^{\frac{1}{2}}$ or $\ln(\det(A))$, are of interest. It is well known (cf. also section 2) that for large n the function $A \to \det(A)$ has poor scaling properties and can be very ill-conditioned for certain matrices A. In this paper we consider the function

$$d: \ A \to \det(A)^{\frac{1}{n}} \ . \tag{1.1}$$

A few basic properties of this function are discussed in section 2. In this paper we present a new method for approximating $d(A)$ for large sparse matrices A. The method is based on using a matrix which is in a certain sense close to A^{-1} and for which the determinant can be computed with low computational costs. One popular method for approximating A is based on the construction of an incomplete Cholesky factorization. This incomplete factorization is often used as a preconditioner when solving linear systems with matrix A. In this paper we use another preconditioning technique, namely, that of factorized sparse approximate inverses (cf. [1, 7, 10, 12]). With such a method a lower triangular matrix G_E with a prescribed sparsity structure E can be constructed such that $G_E A G_E^T$ is in a certain sense close to the identity. We then use $\det(G_E)^{-2/n} = \prod_{i=1}^n (G_E)_{ii}^{-2/n}$ as an approximation for $d(A)$. In section 3 we explain the construction of G_E and discuss theoretical properties of this sparse approximate inverse. For example, such a sparse approximate inverse can be shown to exist for any symmetric positive definite A and has an interesting *optimality property* related to $d(A)$. From this optimality property it immediately follows that $d(A) \leq \det(G_E)^{-2/n}$ holds and that the approximation of $d(A)$ by $\det(G_E)^{-2/n}$ becomes

*Received by the editors March 2, 2000; accepted for publication (in revised form) by N. J. Higham August 17, 2001; published electronically January 23, 2002.
http://www.siam.org/journals/simax/23-3/36869.html
†Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, D-52056 Aachen, Germany (reusken@igpm.rwth-aachen.de).

better if we take a larger sparsity pattern E. Besides this optimality property the method we present has two other interesting properties. The method is *ideal for a parallel implementation* and has *very low storage requirements*.

To make a comparison with other methods for approximating $d(A)$ we describe two known Monte Carlo–type methods (from [3] and [16]). We present results of a few numerical experiments. In these experiments the new method and the Monte Carlo methods are applied to a few model examples of large sparse symmetric positive definite matrices.

2. Preliminaries. In this section we discuss a few elementary properties of the function d. We give a comparison between the conditioning of the function d and of the function $A \to d(A)^n = \det(A)$. We use the notation $\|\cdot\|_2$ for the Euclidean norm, and $\kappa(A) = \lambda_n/\lambda_1$ denotes the spectral condition number of A. The trace of the matrix A is denoted by $\mathrm{tr}(A)$.

LEMMA 2.1. *Let A and $A+\delta A$ be symmetric positive definite matrices of order n. The following inequalities hold:*

$$\lambda_1 \le d(A) \le \lambda_n \ , \tag{2.1a}$$

$$d(A) \le \frac{1}{n}\mathrm{tr}(A) \ , \tag{2.1b}$$

$$\Big|\frac{d(A+\delta A)-d(A)}{d(A)}\Big| \le \kappa(A)\frac{\|\delta A\|_2}{\|A\|_2} \ . \tag{2.1c}$$

Proof. The result in (2.1a) follows from

$$\lambda_1 \le \left(\prod_{i=1}^{n}\lambda_i\right)^{\frac{1}{n}} \le \lambda_n \ .$$

The result in (2.1b) follows from the inequality between the geometric and arithmetic mean:

$$d(A) = \left(\prod_{i=1}^{n}\lambda_i\right)^{\frac{1}{n}} \le \frac{1}{n}\sum_{i=1}^{n}\lambda_i = \frac{1}{n}\mathrm{tr}(A) \ .$$

Now note that

$$\frac{d(A+\delta A)-d(A)}{d(A)} = \big(\det(I+A^{-1}\delta A)\big)^{\frac{1}{n}} - 1 = \left(\prod_{i=1}^{n}(1+\lambda_i(A^{-1}\delta A))\right)^{\frac{1}{n}} - 1 \ .$$

From $\lambda_i(A^{-1}\delta A) \le \|A^{-1}\|_2\|\delta A\|_2$ it follows that

$$\left(\prod_{i=1}^{n}(1+\lambda_i(A^{-1}\delta A))\right)^{\frac{1}{n}} - 1 \le \left(\prod_{i=1}^{n}(1+\|A^{-1}\|_2\|\delta A\|_2)\right)^{\frac{1}{n}} - 1 = \|A^{-1}\|_2\|\delta A\|_2 \ .$$

Using $1+\lambda_i(A^{-1}\delta A) > 0$ and $\lambda_i(A^{-1}\delta A) \ge -\|A^{-1}\|_2\|\delta A\|_2$ we obtain

$$\left(\prod_{i=1}^{n}(1+\lambda_i(A^{-1}\delta A))\right)^{\frac{1}{n}} - 1 \ge \left(\prod_{i=1}^{n}\max\{0,\ 1-\|A^{-1}\|_2\|\delta A\|_2\}\right)^{\frac{1}{n}} - 1 \ge -\|A^{-1}\|_2\|\delta A\|_2.$$

Thus we have

$$\Big|\frac{d(A+\delta A)-d(A)}{d(A)}\Big| \le \|A^{-1}\|_2\|\delta A\|_2 = \kappa(A)\frac{\|\delta A\|_2}{\|A\|_2} ,$$

and the result in (2.1c) is proved. □

The result in (2.1c) shows that the function $d(A)$ is well-conditioned for matrices A which have a not-too-large condition number $\kappa(A)$.

We now briefly discuss the difference in conditioning between the functions $A \to d(A)$ and $A \to \det(A)$. For any symmetric positive definite matrix B of order n we have

$$d'(A)B := \lim_{t\to 0}\frac{d(A+tB)-d(A)}{t} = \frac{d(A)}{n}\mathrm{tr}(A^{-1}B) .$$

From the Courant–Fischer eigenvalue characterization, we obtain, for all i, $\lambda_i(A^{-1}B) \le \lambda_i(A^{-1})\|B\|_2$. Hence

$$\|d'(A)\|_2 := \max_{B \text{ is SPD}}\frac{|d'(A)B|}{\|B\|_2} = \frac{d(A)}{n}\max_{B \text{ is SPD}}\frac{\mathrm{tr}(A^{-1}B)}{\|B\|_2} \le \frac{d(A)}{n}\mathrm{tr}(A^{-1}) ,$$

with equality for $B = I$. Thus for the condition number of the function d we have

$$\frac{\|A\|_2\|d'(A)\|_2}{d(A)} = \frac{1}{n}\|A\|_2\mathrm{tr}(A^{-1}) \le \kappa(A) . \tag{2.2}$$

Note that for the diagonal matrix $A = \mathrm{diag}(A_{ii})$ with $A_{11} = 1,\ A_{ii} = \alpha$ for $i > 1$, the inequality in (2.2) is sharp if $0 < \alpha \ll 1$ and n is large. For this A and with $\delta A = \varepsilon I$, $0 < \varepsilon \ll \alpha$, the bound in (2.1c) is sharp, too.

For $\tilde{d}(A) = \det(A) = d(A)^n$ the condition number is given by

$$\frac{\|A\|_2\|\tilde{d}'(A)\|_2}{\tilde{d}(A)} = \frac{\|A\|_2 n d(A)^{n-1}\|d'(A)\|_2}{d(A)^n} = \|A\|_2\mathrm{tr}(A^{-1}) , \tag{2.3}$$

i.e., n times larger than the condition number in (2.2). The condition numbers for d and $\tilde{d}$ give an indication of the sensitivity if the perturbation $\|\delta A\|_2$ is sufficiently small. Note that the bound in (2.1c) is valid for arbitrary symmetric positive definite perturbations δA. The bound shows that even for larger perturbations the function $d(A)$ is well-conditioned at A if $\kappa(A)$ is not too large. For the function $\tilde{d}(A)$ the effect of relatively large perturbations can be much worse than for the asymptotic case ($\delta A \to 0$), which is characterized by the condition number in (2.3). Consider, for example, for $0 < \varepsilon < \frac{1}{2}$ a perturbation $\delta A = \varepsilon A$, i.e., $\|\delta A\|_2/\|A\|_2 = \varepsilon$. Then

$$\frac{\tilde{d}(A+\delta A)-\tilde{d}(A)}{\tilde{d}(A)} = (1+\varepsilon)^n - 1 \ge e^{\frac{1}{2}n\varepsilon} - 1 ,$$

which is very large if, for example, $\varepsilon = 10^{-3},\ n = 10^5$.

The results in this section show that the numerical approximation of the function $d(A)$ can be considered to be an easier task than the numerical approximation of $A \to \det(A)$.

Remark 2.2. The results on conditioning derived above and the fact that in the analysis of the sparse approximate inverse the function $d(A)$ plays a natural role (cf.

section 3) are the main motivation for considering $d(A)$ instead of $A \to \det(A)$. Of course, an algorithm for approximating $d(A)$ yields an approximation for $\det(A)$ or $\ln(\det(A))$, too. Note that the functions $x \to x^n$ and $x \to \ln(x^n)$ have condition numbers n and $1/\ln(x)$, respectively. Hence, if $\hat{d}$ is an approximation of $d(A)$ with relative error $|\hat{d}-d(A)|/d(A) \leq$ eps, then it follows that $|\hat{d}^n - \det(A)|/\det(A) \lesssim n\,\text{eps}$ and $|\ln(\hat{d}^n) - \ln(\det(A))|/|\ln(\det(A))| \lesssim \text{eps}/|\ln(d(A))|$.

3. Sparse approximate inverse. In this section we explain and analyze the construction of a sparse approximate inverse of the matrix A. Let $A = LL^T$ be the Cholesky factorization of A, i.e., L is lower triangular and $L^{-1}AL^{-T} = I$. Note that $d(A) = d(L)^2 = \prod_{i=1}^n L_{ii}^{2/n}$. We will construct a sparse lower triangular approximation G of L^{-1} and approximate $d(A)$ by $d(G)^{-2} = \prod_{i=1}^n G_{ii}^{-2/n}$. The construction of a sparse approximate inverse that we use in this paper was introduced in [10, 11, 12] and can also be found in [1]. Some of the results derived in this section are also presented in [1].

3.1. Introduction. We first introduce some notation. Let $E \subset \{(i,j) \mid 1 \leq i,j \leq n\}$ be a given sparsity pattern. By $\#E$ we denote the number of elements in E. Let S_E be the set of $n \times n$ matrices for which all entries are set to zero if the corresponding index is *not* in E:

$$S_E = \{M \in \mathbb{R}^{n\times n} \mid M_{ij} = 0 \text{ if } (i,j) \notin E\} .$$

For $1 \leq i \leq n$ let $E_i = E \cap \{(i,j) \mid 1 \leq j \leq n\}$. If $n_i := \#E_i > 0$, we use the representation

$$E_i = \{(i,j_1),(i,j_2),\ldots,(i,j_{n_i})\}, \quad 1 \leq j_1 < j_2 < \cdots < j_{n_i} \leq n . \tag{3.1}$$

For $n_i > 0$ we define the projection

$$P_i : \mathbb{R}^n \to \mathbb{R}^{n_i} , \quad P_i(x_1,x_2,\ldots,x_n)^T = (x_{j_1},x_{j_2},\ldots,x_{j_{n_i}})^T . \tag{3.2}$$

Note that the matrix

$$P_i A P_i^T : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$$

is symmetric positive definite. To facilitate the analysis below, we first discuss the construction of a approximate sparse inverse $M_E \in S_E$ in a general framework. For $M_E \in S_E$ we use the representation

$$M_E = \begin{bmatrix} m_1^T \\ m_2^T \\ \vdots \\ m_n^T \end{bmatrix} , \quad m_i \in \mathbb{R}^n .$$

Note that if $n_i = 0$, then $m_i^T = (0,0,\ldots,0)$.

For given $A, B \in \mathbb{R}^{n\times n}$ with A symmetric positive definite, we consider the following problem:

$$\text{determine } M_E \in S_E \text{ such that } (M_E A)_{ij} = B_{ij} \text{ for all } (i,j) \in E . \tag{3.3}$$

In (3.3) we have $\#E$ equations to determine $\#E$ entries in M_E. We first give two basic lemmas which will play an important role in the analysis of the sparse approximate inverse defined in (3.9) below.

LEMMA 3.1. *The problem* (3.3) *has a unique solution* $M_E \in S_E$. *If* $n_i > 0$, *then the* i*th row of* M_E *is given by* m_i^T *with*

$$m_i = P_i^T (P_i A P_i^T)^{-1} P_i b_i \ , \tag{3.4}$$

where b_i^T *is the* i*th row of* B.

Proof. The equations in (3.3) can be represented as

$$(m_i^T A)_{j_k} = (b_i^T)_{j_k} \text{ for all } i \text{ with } n_i > 0 \text{ and all } k = 1, 2, \ldots, n_i \ ,$$

where m_i^T is the ith row of M_E. Consider an i with $n_i > 0$. Note that $M_E \in S_E$, and hence $P_i^T P_i m_i = m_i$. For the unknown entries in m_i we obtain the system of equations

$$(A P_i^T P_i m_i)_{j_k} = (b_i)_{j_k} \ , \quad k = 1, 2, \ldots, n_i \ ,$$

which is equivalent to

$$P_i A P_i^T P_i m_i = P_i b_i \ .$$

The matrix $P_i A P_i^T$ is symmetric positive definite and thus m_i must satisfy

$$P_i m_i = (P_i A P_i^T)^{-1} P_i b_i \ .$$

Using $P_i^T P_i m_i = m_i$ we obtain the result in (3.4). The construction in this proof shows that the solution is unique. ∎

Below we use the Frobenius norm, denoted by $\| \cdot \|_F$:

$$\|B\|_F^2 = \sum_{i,j=1}^{n} B_{ij}^2 = \operatorname{tr}(BB^T) \ , \quad B \in \mathbb{R}^{n \times n}. \tag{3.5}$$

LEMMA 3.2. *Let* $A = LL^T$ *be the Cholesky factorization of* A *and let* $M_E \in S_E$ *be the unique solution of* (3.3). *Then* M_E *is the unique minimizer of the functional*

$$M \to \|(B - MA)L^{-T}\|_F^2 = \operatorname{tr}((B - MA)A^{-1}(B - MA)^T), \quad M \in S_E. \tag{3.6}$$

Proof. Let e_i be the ith basis vector in $\mathbb{R}^n$. Take $M \in S_E$. The ith rows of M and B are denoted by m_i^T and b_i^T, respectively. Now note

$$\begin{aligned} \operatorname{tr}((B - MA)A^{-1}(B - MA)^T) &= \sum_{i=1}^{n} e_i^T (BA^{-1}B^T - MB^T - BM^T + MAM^T) e_i \\ &= \operatorname{tr}(BA^{-1}B^T) + \sum_{i=1}^{n} (-2 m_i^T b_i + m_i^T A m_i) \ . \end{aligned} \tag{3.7}$$

The minimum of the functional (3.6) is obtained if in (3.7) we minimize the functionals

$$m_i \to -2 m_i^T b_i + m_i^T A m_i \ , \quad m_i \in \mathcal{R}(P_i^T) \tag{3.8}$$

for all i with $n_i > 0$. If we write $m_i = P_i^T \hat{m}_i \ , \hat{m}_i \in \mathbb{R}^{n_i}$, then for $n_i > 0$ the functional (3.8) can be rewritten as

$$\hat{m}_i \to -2 \hat{m}_i^T P_i b_i + \hat{m}_i^T P_i A P_i^T \hat{m}_i \ , \quad \hat{m}_i \in \mathbb{R}^{n_i}.$$

The unique minimum of this functional is obtained for $\hat{m}_i = (P_i A P_i^T)^{-1} P_i b_i$, i.e., $m_i = P_i^T (P_i A P_i^T)^{-1} P_i b_i$ for all i with $n_i > 0$. Using Lemma 3.1 it follows that M_E is the unique minimizer of the functional (3.6). ∎

3.2. Sparse approximate inverse for approximating $d(A)$. We now introduce the sparse approximate inverse that will be used as an approximation for L^{-1}. For this we chose a lower triangular pattern $E^l \subset \{(i,j) \mid 1 \le j \le i \le n\}$ and we assume that $(i,i) \in E^l$ for all i. The sparse approximate inverse is constructed in two steps:

$$1. \quad \hat{G}_{E^l} \in S_{E^l} \text{ such that } (\hat{G}_{E^l} A)_{ij} = \delta_{ij} \text{ for all } (i,j) \in E^l \ , \tag{3.9a}$$

$$2. \qquad G_{E^l} := (\mathrm{diag}(\hat{G}_{E^l}))^{-\frac{1}{2}} \hat{G}_{E^l} \ . \tag{3.9b}$$

The construction of G_{E^l} in (3.9) was first introduced in [10]. A theoretical background for this factorized sparse inverse is given in [12]. The approximate inverse $\hat{G}_{E^l}$ in (3.9a) is of the form (3.3) with $B = I$. From Lemma 3.1 it follows that in (3.9a) there is a unique solution $\hat{G}_{E^l}$. Note that because E^l is lower triangular and $(i,i) \in E^l$ we have $n_i = \#E^l > 0$ for all i and $j_{n_i} = i$ in (3.1). Hence it follows from Lemma 3.1 that the ith row of $\hat{G}_{E^l}$, denoted by g_i^T, is given by

$$\begin{aligned} g_i &= P_i^T (P_i A P_i^T)^{-1} P_i e_i, \qquad i = 1,2,\ldots,n, \\ &= P_i^T (P_i A P_i^T)^{-1} \hat{e}_i, \quad \text{with } \ \hat{e}_i = (0,\ldots,0,1)^T \in \mathbb{R}^{n_i} \ . \end{aligned} \tag{3.10}$$

The ith entry of g_i, i.e., $e_i^T g_i$, is given by $\hat{e}_i^T (P_i A P_i^T)^{-1} \hat{e}_i$, which is strictly positive because $P_i A P_i^T$ is symmetric positive definite. Hence $\mathrm{diag}(\hat{G}_{E^l})$ contains only strictly positive entries and the second step (3.9b) is well-defined. Define $\hat{g}_i = P_i g_i$. The sparse approximate inverse $\hat{G}_{E^l}$ in (3.9a) can be computed by solving the low-dimensional symmetric positive definite systems

$$P_i A P_i^T \hat{g}_i = \hat{e}_i := (0,\ldots,1)^T, \quad i = 1,2,\ldots,n. \tag{3.11}$$

For the approximation of $d(A)$ we propose to use $d(G_{E^l})^{-2}$. Due to

$$d(G_{E^l})^{-2} = d(\hat{G}_{E^l})^{-1} = \prod_{i=1}^{n} (\hat{G}_{E^l})_{ii}^{-\frac{1}{n}}$$

we only need the diagonal entries of $\hat{G}_{E^l}$. In the systems $P_i A P_i^T \hat{g}_i = \hat{e}_i$ we then only have to compute the last entry of $\hat{g}_i$, i.e., $(\hat{g}_i)_{n_i}$. If these systems are solved using the Cholesky factorization $P_i A P_i^T =: L_i L_i^T$ (L_i lower triangular) we only need the (n_i, n_i) entry of L_i, since $(\hat{g}_i)_{n_i} = (L_i)_{n_i n_i}^{-2}$ and thus

$$d(G_{E^l})^{-2} = \prod_{i=1}^{n} (L_i)_{n_i n_i}^{\frac{2}{n}} \ .$$

This leads to the following algorithm.

ALGORITHM 3.3. *Let $A \in \mathbb{R}^{n \times n}$ and a lower triangular pattern E^l be given.*
For $i = 1,\ldots,n$ do:
1. *Construct the matrix $A_i := P_i A P_i^T \in \mathbb{R}^{n_i \times n_i}$.*
2. *Compute the Cholesky factorization $A_i = L_i L_i^T$ and set $\gamma_i := (L_i)_{n_i n_i}$.*

End. Compute

$$\prod_{i=1}^{n} \gamma_i^{\frac{2}{n}} \ . \tag{3.12}$$

3.3. Analysis of the method. We now derive some interesting properties of the sparse approximate inverse as in (3.9). We start with a minimization property of $\hat{G}_{E^l}$.

THEOREM 3.4. *Let $A = LL^T$ be the Cholesky factorization of A and $D := \mathrm{diag}(L)$, $\hat{L} := LD$. $\hat{G}_{E^l}$ as in* (3.9a) *is the unique minimizer of the functional*

$$G \to \|(I - G\hat{L})D^{-1}\|_F^2 = \mathrm{tr}((I - G\hat{L})D^{-2}(I - G\hat{L})^T), \quad G \in S_{E^l}. \tag{3.13}$$

Proof. The construction of $\hat{G}_{E^l}$ in (3.9a) is as in (3.3) with $E = E^l$, $B = I$. Hence Lemma 3.2 is applicable with $B = I$. It follows that $\hat{G}_{E^l}$ is the unique minimizer of

$$G \to \|(I - GA)L^{-T}\|_F^2 , \quad G \in S_{E^l} . \tag{3.14}$$

Decompose L^{-T} as $L^{-T} = D^{-1} + R$ with R strictly upper triangular. We then obtain

$$\begin{aligned} \|(I - GA)L^{-T}\|_F^2 &= \|(I - GLL^T)L^{-T}\|_F^2 = \|D^{-1} + R - GL\|_F^2 \\ &= \|D^{-1} - GL\|_F^2 + \|R\|_F^2 = \|(I - G\hat{L})D^{-1}\|_F^2 + \|R\|_F^2 . \end{aligned}$$

Hence the minimizers in (3.14) and (3.13) are the same. ∎

Remark 3.5. From the result in Theorem 3.4 we see that in a scaled Frobenius norm (scaling with D^{-1}) $\hat{G}_{E^l}$ is the optimal approximation of $\hat{L}^{-1}$ in the set S_{E^l}, in the sense that $\hat{G}_{E^l}\hat{L}$ is closest to the identity. A seemingly more natural minimization problem is

$$\min_{G \in S_{E^l}} \|I - GL\|_F , \tag{3.15}$$

i.e., we directly approximate L^{-1} (instead of $\hat{L}^{-1}$) and do not use the scaling with D^{-1}. The minimization problem (3.15) is of the form as in Lemma 3.2 with $B = L^T$, $E = E^l$. Hence the unique minimizer in (3.15), denoted by $\tilde{G}_{E^l}$, must satisfy (3.3) with $B = L^T$:

$$(\tilde{G}_{E^l}A)_{ij} = L_{ji} \quad \text{for all } (i,j) \in E^l . \tag{3.16}$$

Because E^l contains only indices (i,j) with $i \geq j$ and $L_{ji} = 0$ for $i > j$, it follows that $\tilde{G}_{E^l} \in S_{E^l}$ must satisfy

$$(\tilde{G}_{E^l}A)_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ L_{ii} & \text{if } i = j \end{cases} \qquad \text{for all } (i,j) \in E^l . \tag{3.17}$$

This is similar to the system of equations in (3.9a), which characterizes $\hat{G}_{E^l}$. However, in (3.17) one needs the values L_{ii}, which in general are not available. Hence opposite to the minimization problem related to the functional (3.13) the minimization problem (3.15) is in general not solvable with acceptable computational costs. ∎

The following lemma will be used in the proof of Theorem 3.8.

LEMMA 3.6. *Let $\hat{G}_{E^l}$ be as in* (3.9a). *Decompose $\hat{G}_{E^l}$ as $\hat{G}_{E^l} = \hat{D}(I - \hat{L})$, with $\hat{D}$ diagonal and $\hat{L}$ strictly lower triangular. Define $E^l_- := E^l \setminus \{(i,i) \mid 1 \leq i \leq n\}$. Then $\hat{L}$ is the unique minimizer of the functional*

$$L \to \mathrm{tr}((I - L)A(I - L^T)) , \qquad L \in S_{E^l_-} , \tag{3.18}$$

and also of the functional

$$L \to \det[\mathrm{diag}((I-L)A(I-L^T))] \ , \qquad L \in S_{E^l_-} \ . \tag{3.19}$$

Furthermore, for $\hat{D}$ *we have*

$$\hat{D} = [\mathrm{diag}((I-\hat{L})A(I-\hat{L}^T))]^{-1} \ . \tag{3.20}$$

Proof. From the construction in (3.9a) it follows that

$$((I-\hat{L})A)_{ij} = 0 \quad \text{for all } (i,j) \in E^l_- \ ,$$

i.e., $\hat{L} \in S_{E^l_-}$ is such that $(\hat{L}A)_{ij} = A_{ij}$ for all $(i,j) \in S_{E^l_-}$. This is of the form (3.3) with $B = A$, $E = E^l_-$. From Lemma 3.2 we obtain that $\hat{L}$ is the unique minimizer of the functional

$$L \to \mathrm{tr}((A-LA)A^{-1}(A-LA)^T) = \mathrm{tr}((I-L)A(I-L^T)) \ , \quad L \in S_{E^l_-} \ ,$$

i.e., of the functional (3.18). From the proof of Lemma 3.2, with $B = A$, it follows that the minimization problem

$$\min_{L \in S_{E^l_-}} \mathrm{tr}((I-L)A(I-L^T))$$

decouples into separate minimization problems (cf. (3.8)) for the rows of L:

$$\min_{l_i \in \mathcal{R}(P_i^T)} \{-2l_i^T a_i + l_i^T A l_i\} \tag{3.21}$$

for all i with $n_i > 0$. Here l_i^T and a_i^T are the ith rows of L and A, respectively. The minimization problem corresponding to (3.19) is

$$\min_{L \in S_{E^l_-}} \prod_{i=1}^{n} ((I-L)A(I-L^T))_{ii} = \min_{L \in S_{E^l_-}} \prod_{i=1}^{n} (A_{ii} - 2l_i^T a_i + l_i^T A l_i) \ .$$

This decouples into the same minimization problems as in (3.21). Hence the functionals in (3.18) and (3.19) have the same minimizer.

Let $J = \mathrm{diag}((I-\hat{L})A(I-\hat{L}^T))$. Using the construction of $\hat{G}_{E^l}$ in (3.9a) we obtain

$$\begin{aligned} \hat{D}_{ii}^2 J_{ii} &= (\hat{D}(I-\hat{L})A(I-\hat{L}^T)\hat{D})_{ii} = (\hat{G}_{E^l} A \hat{G}_{E^l}^T)_{ii} \\ &= \sum_{k=1}^{n} (\hat{G}_{E^l} A)_{ik} (\hat{G}_{E^l})_{ik} = \sum_{k=1,(i,k)\in E^l}^{n} \delta_{ik} (\hat{G}_{E^l})_{ik} \\ &= (\hat{G}_{E^l})_{ii} = \hat{D}_{ii} \ . \end{aligned}$$

Hence $\hat{D}_{ii} = J_{ii}^{-1}$ holds for all i, i.e., (3.20) holds. ∎

Corollary 3.7. *From* (3.20) *it follows that* $\mathrm{diag}(\hat{G}_{E^l} A \hat{G}_{E^l}) = \mathrm{diag}(\hat{G}_{E^l})$ *and thus, using* (3.9b), *we obtain*

$$\mathrm{diag}(G_{E^l} A G_{E^l}) = I \tag{3.22}$$

for the approximate inverse G_{E^l}.

The following theorem gives a main result in the theory of approximate inverses. It was first derived in [12]. A proof can be found in [1], too.

THEOREM 3.8. *Let* G_{E^l} *be the approximate inverse in* (3.9). *Then* G_{E^l} *is the unique minimizer of the functional*

$$G \to \frac{\frac{1}{n}\mathrm{tr}(GAG^T)}{\det(GAG^T)^{\frac{1}{n}}}\ , \qquad G \in S_{E^l}\ . \tag{3.23}$$

Proof. For $G \in S_{E^l}$ we use the decomposition $G = D(I-L)$, with D diagonal and $L \in S_{E^l_-}$. Furthermore, for $L \in S_{E^l_-}$, $J_L := \mathrm{diag}((I-L)A(I-L^T))$. Now note

$$\begin{aligned}\frac{\frac{1}{n}\mathrm{tr}(GAG^T)}{\det(GAG^T)^{\frac{1}{n}}} &= \det(A)^{-\frac{1}{n}}\frac{\frac{1}{n}\mathrm{tr}(D(I-L)A(I-L^T)D)}{\det(G^2)^{\frac{1}{n}}} = \det(A)^{-\frac{1}{n}}\frac{\frac{1}{n}\mathrm{tr}(D^2J_L)}{\det(D^2)^{\frac{1}{n}}}\\ &= \det(A)^{-\frac{1}{n}}\frac{\frac{1}{n}\mathrm{tr}(D^2J_L)}{\det(D^2J_L)^{\frac{1}{n}}}\det(J_L)^{\frac{1}{n}} \geq \det(A)^{-\frac{1}{n}}\det(J_L)^{\frac{1}{n}}\ . \end{aligned}\tag{3.24}$$

The inequality in (3.24) follows from the inequality between the arithmetic and geometric mean: $\frac{1}{n}\sum_{i=1}^n \alpha_i \geq (\prod_{i=1}^n \alpha_i)^{1/n}$ for $\alpha_i \geq 0$.

For $\hat{G}_{E^l}$ in (3.9a) we use the decomposition $\hat{G}_{E^l} = \hat{D}(I-\hat{L})$. For the approximate inverse G_{E^l} we then have $G_{E^l} = (\mathrm{diag}(\hat{G}_{E^l}))^{-\frac{1}{2}}\hat{G}_{E^l} = \hat{D}^{\frac{1}{2}}(I-\hat{L})$. From (3.19) of Lemma 3.6 it follows that $\det(J_L) \geq \det(J_{\hat{L}})$ for all $L \in S_{E^l_-}$. Furthermore, from (3.20) of Lemma 3.6 we obtain that for $G_{E^l} = \hat{D}^{\frac{1}{2}}(I-\hat{L})$ we have $(\hat{D}^{\frac{1}{2}})^2 J_{\hat{L}} = I$ and thus equality in (3.24) for $G = G_{E^l}$. We conclude that G_{E^l} is the unique minimizer of the functional in (3.23). ∎

Remark 3.9. The quantity

$$K(A) = \frac{\frac{1}{n}\mathrm{tr}(A)}{\det(A)^{\frac{1}{n}}}$$

can be seen as a nonstandard condition number (cf. [1, 10]). Properties of this quantity are given in [1, Theorem 13.5]. One elementary property is

$$1 \leq K(A) \leq \frac{\lambda_n}{\lambda_1} = \kappa(A)\ .$$

COROLLARY 3.10. *For the approximate inverse* G_{E^l} *as in* (3.9) *we have* (*cf.* (3.22))

$$1 \leq K(G_{E^l}AG_{E^l}^T) = \frac{1}{\det(G_{E^l}AG_{E^l}^T)^{\frac{1}{n}}}\ ,$$

i.e.,

$$d(A) \leq \det(G_{E^l}^2)^{-\frac{1}{n}} = \prod_{i=1}^n (G_{E^l})_{ii}^{-\frac{2}{n}} = \prod_{i=1}^n (\hat{G}_{E^l})_{ii}^{-\frac{1}{n}} = \prod_{i=1}^n \gamma_i^{\frac{2}{n}}\ , \tag{3.25}$$

where γ_i *is as in* (3.12). *Let* $\tilde{E}^l$ *be a lower triangular sparsity pattern that is larger than* E^l, *i.e.,* $E^l \subset \tilde{E}^l \subset \{(i,j) \mid 1 \leq j \leq i \leq n\}$. *From the optimality result in Theorem* 3.8 *it follows that*

$$1 \leq K(G_{\tilde{E}^l}AG_{\tilde{E}^l}^T) \leq K(G_{E^l}AG_{E^l}^T)\ . \tag{3.26}$$

In the following remark we summarize the main properties of the new method for approximating $d(A)$ that is formulated in Algorithm 3.3.

Remark 3.11. The method of approximating $d(A)$ by $d(G_{E^l})^{-2} = d(\hat{G}_{E^l})^{-1}$ boils down to choosing a sparsity pattern E^l and computing the Cholesky decomposition of the low-dimensional matrices A_i in step 2 of Algorithm 3.3. We note the following related to this algorithm:

1. The sparse approximate inverse exists for every symmetric positive definite A. Note that such an existence result does not hold for the incomplete Cholesky factorization.

2. The construction of the matrices $A_i = P_i A P_i^T$ and the computation of the Cholesky factorization $A_i = L_i L_i^T$ can be realized for all i *in parallel*. Hence the method has a very high potential for parallelism.

3. If for a given i the number $\gamma_i = (L_i)_{n_i n_i}$ in (3.12) has been computed the matrices A_i and L_i are not needed anymore. Hence the storage requirements for the method are very low.

4. The sparse approximate inverse has an optimality property related to the determinant: The functional $G \to K(GAG^T),\ G \in S_{E^l}$, is minimal for G_{E^l}. From this the inequality (3.25) and the monotonicity result (3.26) follow.

5. From (3.25) it follows that $\prod_{i=1}^n \gamma_i^{\frac{2}{n}}$ is an upper bound for $d(A)$. ∎

4. Monte Carlo methods for approximating $d(A)$. In this section we describe two methods for approximating $d(A)$ that are known from the literature. Both methods are based on the following proposition [9, 3].

PROPOSITION 4.1. *Let H be a symmetric matrix of order n with $\mathrm{tr}(H) \neq 0$. Let V be the discrete random variable which takes the values 1 and -1 each with probability 0.5 and let z be a vector of n independent samples from V. Then $z^T H z$ is an unbiased estimator of $\mathrm{tr}(H)$:*

$$E(z^T H z) = \mathrm{tr}(H)$$

and

$$var(z^T H z) = 2 \sum_{i \neq j} h_{ij}^2 \ .$$

Using the identity

$$d(A) = \det(A)^{\frac{1}{n}} = \exp\left(\frac{1}{n} \mathrm{tr} \ln(A)\right)$$

leads to the following Monte Carlo algorithm.

ALGORITHM 4.2.

For $j = 1, 2, \ldots, M$

1. *Generate $z_j \in \mathbb{R}^n$ with entries which are uniformly distributed in $(0, 1)$.*
2. *If $(z_j)_i < 0.5$, then $(z_j)_i := -1$, otherwise $(z_j)_i := 1$.*
3. *Compute an approximation*

$$d_j \approx z_j^T \ln(A) z_j \ . \tag{4.1}$$

End. Compute

$$\hat{d}_M(A) = \exp\left(\frac{1}{n} \frac{1}{M} \sum_{j=1}^M d_j\right) \ .$$

In the following two subsections we describe methods for computing the approximation $d_j \approx z_j^T \ln(A) z_j$ in (4.1).

4.1. Approximation of $z^T \ln(A) z$ using Chebyshev polynomials. We describe a method that is presented in [16]. We assume that A is scaled by a factor $0 < \frac{1}{b} \le \frac{1}{\lambda_n}$. Then $\sigma(\frac{1}{b}A) \subset [\varepsilon, 1]$ holds with $0 < \varepsilon \le \frac{\lambda_1}{b}$. For ease of notation this scaled matrix is denoted by A, too.

Let T_k, $k \ge 0$, be the Chebyshev polynomials on $[0,1]$:

$$T_{-1}(x) = 2x - 1, \quad T_0(x) = 1, \quad T_{k+1}(x) = (4x-2)T_k(x) - T_{k-1}(x) \quad \text{for} \quad k \ge 1 \ .$$

The method is based on the following expansion for $\ln x$:

$$\ln x = \sum_{k=0}^{m+1} b_k T_k\left(\frac{1-x}{1-\varepsilon}\right) + \delta \ln x \quad \text{for} \quad x \in [\varepsilon, 1], \tag{4.2}$$

$$|\delta| \le 2e^{-2(m+1)\sqrt{\varepsilon}} \ . \tag{4.3}$$

We show that this result holds and derive a simple and cheap algorithm for the computation of the coefficients b_k. The starting point is the identity

$$\frac{1}{y}\Big(1 + \rho T_{m+1}\Big(\frac{1-y}{1-\varepsilon}\Big)\Big) = \sum_{k=0}^{m} c_k T_k\Big(\frac{1-y}{1-\varepsilon}\Big) \ , \quad y \in [\varepsilon, 1] \ , \tag{4.4}$$

with parameters ρ and c_k, $0 \le k \le m$. With $z := \frac{1-y}{1-\varepsilon} \in [0,1]$ this is equivalent to

$$1 + \rho T_{m+1}(z) = (1 - (1-\varepsilon)z) \sum_{k=0}^{m} c_k T_k(z) \ . \tag{4.5}$$

Substituting $zT_k(z) = \frac{1}{4}T_{k+1}(z) + \frac{1}{2}T_k(z) + \frac{1}{4}T_{k-1}(z)$ in (4.5) and comparing the coefficients of T_k on both sides of the equality results in a linear system of $m+2$ equations for the unknowns $c := (c_0, \ldots, c_m)^T$ and ρ. A simple calculation shows that the solution of this system is given by

$$c = \frac{4}{1-\varepsilon} B^{-1} e_1 \ , \tag{4.6}$$

$$\rho = -e_{m+1}^T B^{-1} e_1 \ , \tag{4.7}$$

with e_1 and e_{m+1} the first and $(m+1)$st basis vector in $\mathbb{R}^{m+1}$, respectively, and

$$B = \begin{pmatrix} 2\gamma & -1 & & & & \\ -2 & 2\gamma & -1 & & \emptyset & \\ & -1 & 2\gamma & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & \emptyset & & \ddots & \ddots & -1 \\ & & & & -1 & 2\gamma \end{pmatrix} \in \mathbb{R}^{(m+1)\times(m+1)} \ , \quad \gamma := \frac{1+\varepsilon}{1-\varepsilon} \ .$$

Hence, the LU-decomposition of B results in an efficient algorithm for computing the coefficients c and ρ in (4.6), (4.7). Elementary manipulations with difference equations

yield explicit formulas for $B^{-1}e_1$. For example, for the last component of this vector one can derive the expression

$$-\rho = e_{m+1}^T B^{-1} e_1 = \frac{-2}{\lambda^{m+1} + \lambda^{-(m+1)}} \ , \quad \lambda := \gamma + \sqrt{\gamma^2 - 1} \ . \tag{4.8}$$

Such explicit expressions are given in [16] and offer an alternative (but probably somewhat less efficient) approach for computing c and ρ.

From (4.4) and $|T_{m+1}(z)| \leq 1$ it follows that

$$-|\rho|\frac{1}{y} \leq \frac{1}{y} - \sum_{k=0}^{m} c_k T_k\Big(\frac{1-y}{1-\varepsilon}\Big) \leq |\rho|\frac{1}{y} \ , \qquad y \in [\varepsilon, 1] \ .$$

Integrating between $y = x \in [\varepsilon, 1]$ and $y = 1$ we obtain

$$|\rho| \ln x \leq -(1-\varepsilon)\sum_{k=0}^{m} c_k \int_0^{\frac{1-x}{1-\varepsilon}} T_k(z)\, dz - \ln x \leq -|\rho| \ln x \ , \quad x \in [\varepsilon, 1] \ . \tag{4.9}$$

Using $\int T_0 = \frac{1}{2}(T_0 + T_1)$, $\int T_1 = \frac{1}{8}(T_2 - T_0)$, $\int T_k = \frac{1}{4}(\frac{T_{k+1}}{k+1} - \frac{T_{k-1}}{k-1})$, $k \geq 2$, a straightforward computation yields

$$\begin{aligned}
&-(1-\varepsilon)\sum_{k=0}^{m} c_k \int_0^{\frac{1-x}{1-\varepsilon}} T_k(z)\, dz = \sum_{k=0}^{m+1} b_k T_k\Big(\frac{1-x}{1-\varepsilon}\Big) \ , \\
&\text{with} \quad b_k = -\frac{1-\varepsilon}{4k} c_{k-1} \ , \quad k = m, m+1 \ , \\
&b_k = -\frac{1-\varepsilon}{4k}(c_{k-1} - c_{k+1}) \ , \quad 2 \leq k \leq m-1 \ , \\
&b_1 = -\frac{1-\varepsilon}{4}(2c_0 - c_2) \ , \\
&b_0 = -\sum_{k=1}^{m+1} (-1)^k b_k \ .
\end{aligned} \tag{4.10}$$

Hence, using the values for the coefficients $c = (c_0, \ldots, c_m)^T$ from (4.6) the coefficients b_k in (4.2) directly follow from (4.10). The bound on δ in (4.3) is a consequence of (4.9) and

$$|\rho| = \frac{2}{\lambda^{m+1} + \lambda^{-(m+1)}} \leq 2\lambda^{-(m+1)} \leq 2e^{-2(m+1)\sqrt{\varepsilon}} \ .$$

Now assume that the coefficients b_k have been computed. For $z_j \in \mathbb{R}^n$ it follows that

$$z_j^T \ln(A) z_j \approx \sum_{k=0}^{m+1} b_k z_j^T T_k\Big(\frac{I-A}{1-\varepsilon}\Big) z_j =: d_j \tag{4.11}$$

can be used as an approximation in (4.1). The terms $z_j^T T_k\big(\frac{I-A}{1-\varepsilon}\big) z_j$ in (4.11) can be computed using the recursion for T_k. In our applications we have $n = \dim(A) \gg m$, and the costs for computing d_j in (4.11) are dominated by the costs for the $m+1$ matrix-vector multiplications with the matrix A. These matrix-vector computations are easy to parallelize. Note, however, that the Monte Carlo Algorithm 4.2 and the computation of the sum in (4.11) are purely sequential processes.

4.2. Approximation of $z^T \ln(A) z$ using quadrature. In this subsection we recall the method from [3] for approximating $z^T \ln(A) z$, $z \in \mathbb{R}^n$. Let $Q^T \Lambda Q = A$ be the eigendecomposition of A with Q orthogonal, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_1 \le \cdots \le \lambda_n$. For $z \in \mathbb{R}^n$ let $\tilde{z} = \frac{Qz}{\|z\|_2}$. Then we have

$$\frac{z^T \ln(A) z}{\|z\|_2^2} = \tilde{z}^T \ln(\Lambda) \tilde{z} = \sum_{i=1}^{n} \ln \lambda_i \, \tilde{z}_i{}^2 = \int_{\lambda_1}^{\lambda_n} \ln \lambda \, d\mu(\lambda) =: J \ , \tag{4.12}$$

where the measure $\mu(\lambda)$ is given by

$$\mu(\lambda) = \begin{cases} 0 & \text{if} \quad \lambda < \lambda_1 \ , \\ \sum_{j=1}^{i} \tilde{z}_j^2 & \text{if} \quad \lambda_i \le \lambda < \lambda_{i+1}, \quad 1 \le i \le n-1 \ , \\ 1 & \text{if} \quad \lambda_n \le \lambda \ . \end{cases}$$

For approximating the integral in (4.12) one can use a Gauss-type quadrature rule. Several possibilities are treated in [3]. Here we use a Gauss–Radau method:

$$Q_N := \sum_{j=1}^{N} \omega_j \ln \theta_j + \nu_\tau \ln \tau \ ,$$

where the node τ is prescribed. We will consider $\tau \approx \lambda_1$ and $\tau \approx \lambda_n$. The weights ω_j, ν_τ and the nodes θ_j are unknown and to be determined. It is well known that the nodes and weights in the Gauss quadrature can be computed using the Lanczos method (cf. [4]). The Gauss–Radau quadrature is treated in [5]. For $f(x) = \ln x$ we have $f^{(2N+1)}(x) > 0$ for all $x > 0$, and from [5] it then follows that if $\tau \le \lambda_1$ ($\tau \ge \lambda_n$), the approximation Q_N is a lower bound (upper bound) for J. In [3] the following algorithm for approximating J is proposed. We assume that $\nu_1 \le \lambda_1$ and $\nu_2 \ge \lambda_n$ are given.

ALGORITHM 4.3.

$x_0 = z/\|z\|_2, \quad x_{-1} = 0, \quad \gamma_0 = 0$;

For $k = 1, 2, \ldots$ *do:*

1. $\alpha_k = x_{k-1}^T A x_{k-1}$.
2. $r_k = A x_{k-1} - \alpha_k x_{k-1} - \gamma_{k-1} x_{k-2}$.
3. $\gamma_k = \|r_k\|_2$.
4. *Let*

$$T_k = \begin{pmatrix} \alpha_1 & \gamma_1 & & \emptyset & \\ \gamma_1 & \alpha_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{k-1} \\ & \emptyset & & \gamma_{k-1} & \alpha_k \end{pmatrix} \ ,$$

$$\delta_m = \gamma_k^2 e_k^T (T_k - \nu_m I)^{-1} e_k, \quad m = 1, 2,$$

$$\hat{T}_k^{(m)} = \begin{pmatrix} T_k & \gamma_k e_k \\ \gamma_k e_k^T & \phi_m \end{pmatrix} \ , \quad \phi_m = \nu_m + \delta_m \ , \quad m = 1, 2 \ .$$

5. *Compute the eigenvalues* $\theta_\ell^{(m)}$ *and the first elements* $\omega_\ell^{(m)}$ *of the normalized eigenvectors of* $\hat{T}_k^{(m)}$ *($m = 1, 2$; $1 \le \ell \le k+1$).*

6. $Q_k^{(m)} = \sum_{\ell=1}^{k+1} (\omega_\ell^{(m)})^2 \ln \theta_\ell^{(m)}, \;\; m = 1, 2.$

7. *If* $\frac{Q_k^{(2)} - Q_k^{(1)}}{|Q_k^{(1)}|} \leq \text{eps}$ *(user specified tolerance), then Stop.*

8. $x_k = r_k / \gamma_k$.

End. Compute

$$d_z = \frac{1}{2}(Q_k^{(1)} + Q_k^{(2)}) \|z\|_2^2 \; . \tag{4.13}$$

For $z = z_j$ as in Algorithm 4.2 the value $d_j := d_{z_j}$ from (4.13) is taken as the approximation in (4.1). As for the method in the previous subsection we have an outer (Monte Carlo) and inner iteration which are purely sequential operations. In our applications the dimensions of the eigenvalue problems that occur in Algorithm 4.3 are very small compared to $n = \dim(A)$, and the costs for one iteration in this algorithm are dominated by the matrix-vector multiplication with the matrix A.

Both in the algorithm in this subsection and in the algorithm in subsection 4.1 we need approximations of λ_1 and λ_n. It turns out that the performance of the algorithms is less sensitive to the accuracy of these approximations. In the numerical experiments we used a fixed (small) number of Lanczos iterations to compute these approximations.

5. Numerical experiments. In this section we present some results of numerical experiments with the methods introduced in sections 3 and 4. All experiments are done using a MATLAB implementation.

Experiment 1 (discrete two-dimensional Laplacian). We consider the standard 5-point discrete Laplacian on a uniform square grid with N mesh points in both directions, i.e., $n = N^2$. For this symmetric positive definite matrix the eigenvalues are known:

$$\lambda_{\nu\mu} = 4(N+1)^2 \left(\sin^2 \left(\frac{\nu\pi}{2(N+1)} \right) + \sin^2 \left(\frac{\mu\pi}{2(N+1)} \right) \right), \quad 1 \leq \nu, \mu \leq N \; . \tag{5.1}$$

For the choice of the sparsity pattern E^l we use a simple approach:

$$E^l(k) := \{(i,j) \mid i \geq j \;\text{ and }\; (A^k)_{ij} \neq 0\} \; , \quad k = 1, 2, \ldots \; . \tag{5.2}$$

We first describe some features of the methods for the case $N = 30$, $k = 2$, and after that we will vary N and k. Let A denote the discrete Laplacian for the case $N = 30$. For the matrices $A_i = P_i A P_i^T \in \mathbb{R}^{n_i \times n_i}$ $(i = 1, \ldots, n)$ the dimensions n_i are between 1 and 7; the mean of these dimensions is 6.7. Algorithm 3.3 yields an approximation,

$$d(G_{E^l(2)})^{-2} = d(\hat{G}_{E^l(2)})^{-1} = \prod_{i=1}^{n} \gamma_i^{\frac{2}{n}} = 3.2526 \; 10^3$$

for $d(A) = 3.1379 \; 10^3$. Hence the relative error is 3.5%. For the computation of the Cholesky factorizations $A_i = L_i L_i^T$, $i = 1, 2, \ldots, n$, approximately 41 10^3 flops are needed (in the MATLAB implementation). If we compare this with the costs of one matrix-vector multiplication $A * x$ (8760 flops), denoted by MATVEC, it follows that for computing this approximation of $d(A)$, with error 3.5%, we need arithmetic work comparable to only 5 MATVEC. In Table 5.1 we give results for the discrete two-dimensional Laplacian with $N = 30$ $(n = 900)$, $N = 100$ $(n = 10000)$ and $N = 200$ $(n = 40000)$. We use the sparsity patterns $E^l(2)$ and $E^l(4)$. In the third column

TABLE 5.1
Results for two-dimensional discrete Laplacian with $E^l = E^l(2)$.

n	$d(A)$	$d(G_{E^l(2)})^{-2}$ (error)	Costs for $d(G_{E^l(2)})^{-2}$	$d(G_{E^l(4)})^{-2}$ (error)	Costs for $d(G_{E^l(4)})^{-2}$
900	$3.138\ 10^3$	$3.253\ 10^3$ (3.5%)	5 MV	$3.177\ 10^3$ (1.2%)	41 MV
10000	$3.292\ 10^4$	$3.434\ 10^4$ (4.1%)	5 MV	$3.347\ 10^3$ (1.6%)	45 MV
40000	$1.300\ 10^5$	$1.359\ 10^5$ (4.3%)	5 MV	$1.323\ 10^3$ (1.7 %)	46 MV

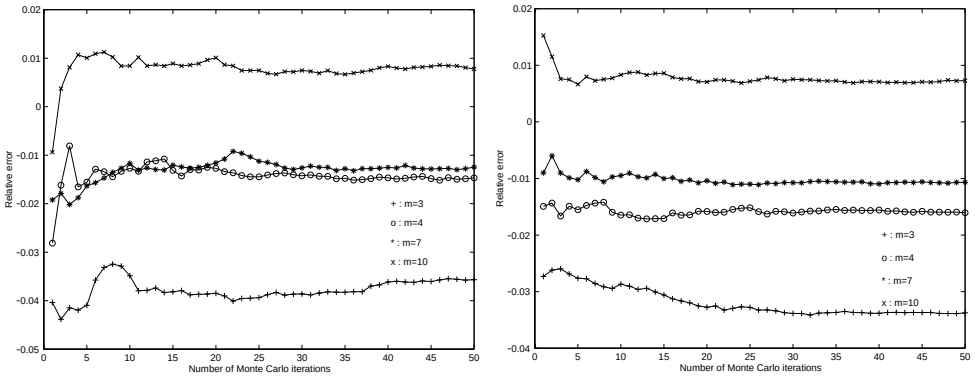


FIG. 5.1. *Algorithm* 4.2 *combined with the method from section* 4.1: $n = 10000$ *(left),* $n = 40000$ *(right).*

of this table we give the computed approximation of $d(A)$ and the corresponding relative error. In the fourth column we give the total arithmetic costs for the Cholesky factorization of the matrices A_i, $i = 1, 2, \ldots, n$. In the columns 5 and 6 we give the results and corresponding arithmetic costs for the case with larger sparsity pattern $E^l(4)$.

Related to these numerical results we note the following. From the third and fourth column in Table 5.1 we see that using this method we can obtain an approximation of $d(A)$ with relative error only a few percent and arithmetic costs only a few MATVEC. Moreover, this efficiency hardly depends on the dimension n. Comparison of the third and fifth columns in Table 5.1 shows that the approximation significantly improves if we enlarge the pattern from $E^l(2)$ to $E^l(4)$. The corresponding arithmetic costs increase by a factor of about 9. This is caused by the fact that the mean of the dimensions of the systems A_i, $i = 1, 2, \ldots, n$, increases from approximately 7 (for $E^l(2)$) to approximately 20 (for $E^l(4)$).

We also applied the Monte Carlo Algorithm 4.2, with $M = 50$, to this problem. If, for the approximation of $z_j^T \ln(A) z_j$ in (4.1), we use the approach based on Chebyshev polynomials, we obtain the results in Figure 5.1. It turns out that the bound in (4.3) is very pessimistic and should not be used to determine a value for the parameter m. In the experiments we used the values $m = 3, 4, 7, 10$. Note that the arithmetic costs in the inner Chebyshev iteration (4.11) are comparable to $m + 1$ MATVEC. From Figure 5.1 we see that for a relative error of approximately 1.5% it suffices to take 10–15 Monte Carlo iterations with $m = 4$. The arithmetic costs are then

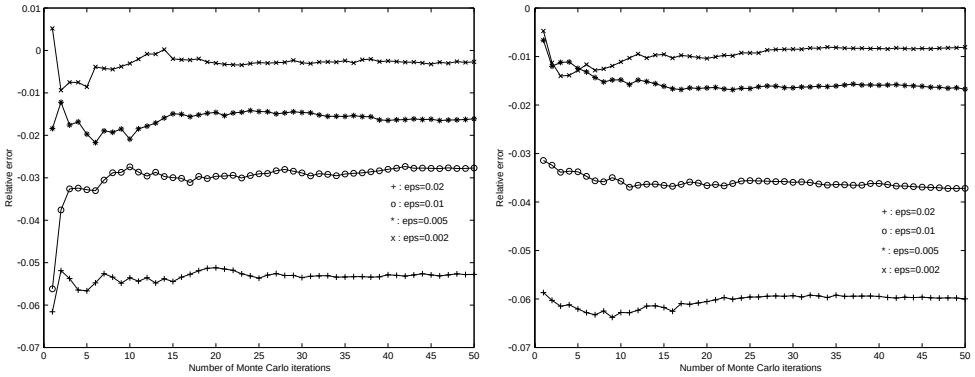


FIG. 5.2. *Algorithm* 4.2 *combined with the method from section* 4.2: $n = 10000$ *(left),* $n = 40000$ *(right).*

roughly 50–75 MATVEC. In Figure 5.2 results are shown if $z_j^T \ln(A) z_j$ in (4.1) is approximated using Algorithm 4.3. We used different tolerances in step 7 in this algorithm: eps = 0.02, 0.01, 0.005, 0.002. The corresponding total number of matrix-vector multiplications is 188, 250, 306, 449 (for $n = 10000$) and 200, 250, 350, 501 (for $n = 40000$). We observe that for a relative error of approximately 1.5% about 10–15 Monte Carlo iterations with eps = 0.005 are sufficient. The arithmetic costs are then roughly 60–95 MATVEC.

Note that both Monte Carlo methods (in Figures 5.1 and 5.2) perform similarly. In both methods we need estimates for the extreme eigenvalues of the matrix A. We used the known values of these extreme eigenvalues given in (5.1).

Experiment 2 (MATLAB random sparse matrix). The sparsity structure of the matrices considered in Experiment 1 is very regular. In this experiment we consider matrices with a pattern of nonzero entries that is very irregular. We used the MATLAB generator (SPRAND$(n, n, 2/n)$) to generate a matrix B of order n with approximately $2n$ nonzero entries. These are uniformly distributed random entries in $(0, 1)$. The matrix $B^T B$ is then sparse symmetric positive semidefinite. In the generic case this matrix has many eigenvalues zero. To obtain a positive definite matrix we generated a random vector d with all entries chosen from a uniform distribution on the interval $(0, 1)$ ($d :=$RAND$(n, 1)$). As a test matrix we used $A := B^T B + \mathrm{diag}(d)$. We performed numerical experiments similar to those in Experiment 1 above. We consider only the case with sparsity pattern $E^l = E^l(2)$. Results obtained with Algorithm 3.3 are shown in Table 5.2. From these results it is clear that for this random matrix A the approximation of $d(A)$ based on the sparse approximate inverse is much better than for the discrete Laplacian in Experiment 1. This is related to the fact that for the random matrices considered in this example the preconditioned matrix $G_{E^l} A G_{E^l}$ turns out to be very well-conditioned.

We also apply the same Monte Carlo methods as discussed in Experiment 1 to these matrices. To allow a fair comparison we first rescaled the matrix A with a diagonal matrix D such that the absolute row sums of the matrix DAD are all equal to one. Estimates of the extreme eigenvalues that are needed in these algorithms are obtained by applying 20 iterations of the Lanczos method (with starting vector $(1, \ldots, 1)^T$). The performance of these methods is similar for the three cases $n =$ 900, 10000, 40000. In Figure 5.3 we show the results for the case $n = 10000$.

TABLE 5.2
Results for MATLAB random sparse matrices with $E^l = E^l(2)$.

n	$d(A)$	$d(G_{E^l})^{-2}$ (error)	Costs for $d(G_{E^l})^{-2}$
900	0.82453	0.82521 ($8.2\ 10^{-4}$)	23 MV
10000	0.80985	0.81053 ($8.4\ 10^{-4}$)	18 MV

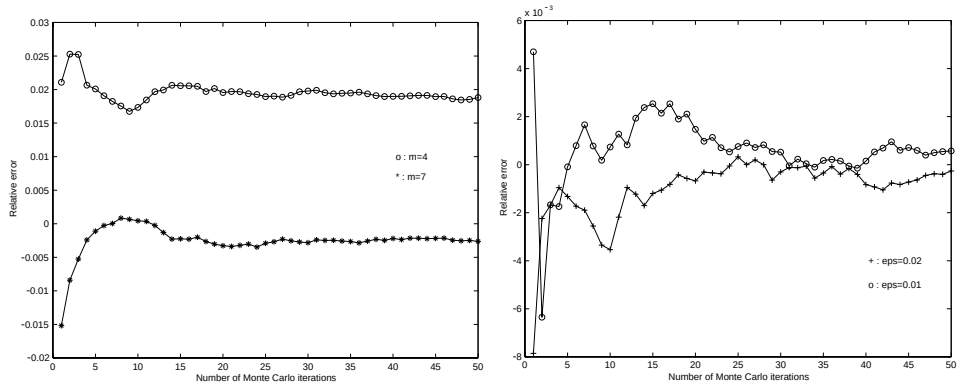


FIG. 5.3. *Algorithm* 4.2 *combined with the methods from section* 4.1 *(left) and from section* 4.2 *(right).*

For the Monte Carlo method using the approach based on Chebyshev polynomials the result after 20 iterations and $m = 7$ (cf. Figure 5.3, left) has a relative error ≈ 0.001. For computing this result, approximately 180 MATVEC are needed. If we use the Gauss–Radau quadrature (cf. Figure 5.3, right) with eps $= 0.02$, then after 20 Monte Carlo iterations the result also has a relative error ≈ 0.001. The total costs are about 100 MATVEC. Hence, in this example the method based on the sparse approximate inverse is more efficient than the Monte Carlo methods.

Experiment 3 (quantum chromodynamics (QCD) type matrix). In this experiment we consider a complex Hermitian positive definite matrix with a regular sparsity structure. This matrix is motivated by applications from the QCD field. In QCD simulations the determinant of the so-called Wilson fermion matrix is of interest. These matrices and some of their properties are discussed in [14, 13]. The Wilson fermion matrix $A = I - \kappa D$ describes a nearest neighbor coupling with periodic boundary conditions on a four-dimensional regular space-time lattice with lattice sites

$$\Omega_N = \{\, (x_1, x_2, x_3, x_4) \mid x_i = 1, \ldots, n_i, \quad n_i = 2^{N_i} \,\} \ .$$

The so-called hopping matrix D has the form

$$D_{x,y} = \sum_{\mu=1}^{4} \big((I - \gamma_\mu) \otimes U_\mu(x)\big)\delta_{x,y-e_\mu} + \big((I + \gamma_\mu) \otimes U_\mu^H(x - e_\mu)\big)\delta_{x,y+e_\mu} \ , \tag{5.3}$$

where x, y are lattice sites from Ω_N, e_μ is the μth basisvector in $\mathbb{R}^4$, and $\delta_{x,y} = 1$ (0) if $x = y$ ($x \neq y$). The matrices $I \pm \gamma_\mu \in \mathbb{C}^{4\times 4}$ are projectors onto two-dimensional subspaces and the matrices $U_\mu(x) \in \mathbb{C}^{3\times 3}$ are from $SU(3)$ (see [13] for

TABLE 5.3
Results for QCD type matrix with $E^l = E^l(2)$.

n	$d(A)$	$d(G_{E^l})^{-2}$ (error)	Costs for $d(G_{E^l})^{-2}$
1024	0.8032	0.8248 (2.7%)	22 MV
4096	0.8037	0.8254 (2.7%)	21 MV

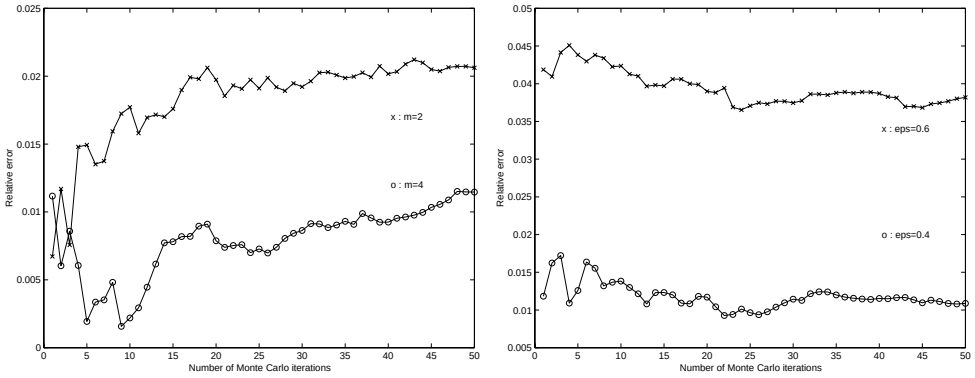


FIG. 5.4. *Algorithm* 4.2 *combined with the methods from section* 4.1 *(left) and from section* 4.2 *(right).*

details). Usually, these matrices $U_\mu(x)$ are generated randomly. In this model the matrix D has a block structure with blocks $D_{x,y} \in \mathbb{C}^{12\times 12}$, $x, y \in \Omega_N$. Here we consider a very simple variant of this model. We take $\gamma_\mu = 0$, $I = 1$, $U_\mu(x) = \exp(2i\pi\alpha_\mu(x))$, where $\alpha_\mu(x)$ is chosen from a uniform distribution on the interval $(0, 1)$. Hence the couplings $D_{x,y}$ in (5.3) are complex scalars. Note that the matrix D is Hermitian. Due to the randomly generated functions $\alpha_\mu(x)$ the couplings $D_{x,y}$ show a strong fluctuation as a function of x and y. In QCD simulations the parameter κ is taken such that the Wilson fermion matrix A is positive definite and close to singular. In the experiment here we computed the largest eigenvalue ρ_D of D (using the MATLAB function EIGS) and set $\kappa := (1.01\,\rho_D)^{-1}$. We performed numerical experiments as in Experiment 1 with $E^l = E^l(2)$ for two cases: $(n_1, n_2, n_3, n_4) = (4, 4, 8, 8)$ and $(n_1, n_2, n_3, n_4) = (8, 8, 8, 8)$. The results are presented in Table 5.3. We also used the Monte Carlo methods. As in Experiment 2 we applied 20 Lanczos iterations to obtain estimates for the extreme eigenvalues. The results are shown in Figure 5.4. From this figure we see that after 20 Monte Carlo iterations using the method from subsection 4.1 with $m = 2$ the result has a relative error of about 2%. For computing this result approximately 80 MATVEC are needed. Using the method from subsection 4.2 with eps = 0.4 the result after 20 Monte Carlo iterations has a relative error of about 1%. The total costs for computing this result are about 80 MATVEC.

Note that in all three experiments the performance of the methods hardly depends on the dimension n. In all measurements for the arithmetic costs we did not take into account the costs of determining the sparsity pattern $E^l(k)$ and of building the matrices $P_i A P_i^T$.

We conclude that at least for these few model problems the new method can

compete, even on a sequential machine, with the two Monte Carlo methods proposed in the literature. We believe that in a (massively) parallel environment the method based on the sparse approximate inverse can be expected to be much more efficient than the Monte Carlo techniques because the former is ideally suited for a parallel implementation.

Remark 5.1. In this paper we do not discuss the topic of error estimation. For the Monte Carlo method, error estimation techniques are treated in [3]. Related to the method based on the sparse approximate inverse (Algorithm 3.3) we briefly discuss one possible technique for a posteriori error estimation. From (3.25) we have the a priori error bound

$$\frac{d(A)}{d(G_{E^l})^{-2}} \leq 1 \ .$$

The exact error is given by

$$\frac{d(A)}{d(G_{E^l})^{-2}} = d(G_{E^l} A G_{E^l}^T) = d(\mathcal{E}_{E^l}) \ ,$$

where $\mathcal{E}_{E^l} := G_{E^l} A G_{E^l}^T$ is a sparse symmetric positive definite matrix. For ease of presentation we assume that the pattern E^l is sufficiently large such that $\rho(I - \mathcal{E}_{E^l}) < 1$ holds. In [12] it is proved that if A is an M-matrix or a (block) H-matrix, then this condition is satisfied for every lower triangular pattern E^l. For the exact error we obtain, using a Taylor expansion of $\ln(I - B)$ for $B \in \mathbb{R}^{n \times n}$ with $\rho(B) < 1$ (see [6]),

$$\begin{aligned} d(\mathcal{E}_{E^l}) &= \exp\left(\frac{1}{n}\ln(\det(\mathcal{E}_{E^l}))\right) = \exp\left(\frac{1}{n}\mathrm{tr}(\ln(\mathcal{E}_{E^l}))\right) \\ &= \exp\left(\frac{1}{n}\mathrm{tr}(\ln(I - (I - \mathcal{E}_{E^l})))\right) = \exp\left(-\frac{1}{n}\mathrm{tr}\left(\sum_{k=1}^{\infty} \frac{(I - \mathcal{E}_{E^l})^k}{k}\right)\right) \ . \end{aligned} \tag{5.4}$$

Hence, an error estimation can be based on estimates for the partial sums $S_m := \sum_{k=1}^{m} \frac{1}{k}\mathrm{tr}((I - \mathcal{E}_{E^l})^k)$. The construction of G_{E^l} is such that $\mathrm{diag}(\mathcal{E}_{E^l}) = I$ (cf. (3.22)) and thus $\mathrm{tr}(\mathcal{E}_{E^l}) = n$ and $S_1 = 0$. For S_2 we have

$$S_2 = \frac{1}{2}\mathrm{tr}((I - \mathcal{E}_{E^l})^2) = \frac{1}{2}\mathrm{tr}(I - 2\mathcal{E}_{E^l} + \mathcal{E}_{E^l}^2) = -\frac{1}{2}n + \frac{1}{2}\mathrm{tr}(\mathcal{E}_{E^l}^2) \ . \tag{5.5}$$

For approximating the trace quantity $\mathrm{tr}(\mathcal{E}_{E^l}^2)$ in S_2 we can use the following Monte Carlo algorithm based on Proposition 4.1.

ALGORITHM 5.2.

For $j = 1, 2, \ldots, M$

1. *Generate* $z_j \in \mathbb{R}^n$ *with entries which are uniformly distributed in* $(0, 1)$.
2. *If* $(z_j)_i < 0.5$, *then* $(z_j)_i := -1$, *otherwise* $(z_j)_i := 1$.
3. $y_j := \mathcal{E}_{E^l} z_j, \quad \alpha_j := y_j^T y_j$.

End.

This then yields

$$\hat{S}_2 := -\frac{1}{2}n + \frac{1}{2M}\sum_{j=1}^{M} \alpha_j \tag{5.6}$$

as an approximation for S_2. The corresponding error estimate is given by

$$E_2 = \exp\left(-\frac{1}{n}\hat{S}_2\right). \tag{5.7}$$

It turns out that, at least in our experiments, this technique yields satisfactory results. One clear disadvantage of this approach is that the matrix G_{E^l} must be available (and thus stored). Note that for the computation of the approximation $d(G_{E^l})^{-2}$ of $d(A)$ we do not have to store the matrix G_{E^l}.

Acknowledgment. The author thanks the referees for a number of valuable comments that considerably improved the paper.

REFERENCES

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, 1994.
[2] Z. Bai and G. H. Golub, *Bounds on the trace of the inverse and the determinant of symmetric positive definite matrices*, Ann. Numer. Math., 4 (1997), pp. 29–38.
[3] Z. Bai, M. Fahey, and G. H. Golub, *Some large scale matrix computation problems*, J. Comput. Appl. Math., 74 (1996), pp. 71–89.
[4] P. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1984.
[5] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[7] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
[8] M. Hasenbusch, *Speeding up finite step-size updating of full QCD on the lattice*, Phys. Rev. D, 59 (1999); available online as article 054505 from http://prd.aps.org/.
[9] M. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Statist. Simulation Comput., 18 (1989), pp. 1059–1076.
[10] I. E. Kaporin, *An alternative approach to estimating the convergence rate of the CG method*, in Numerical Methods and Software, Yu. A. Kuznetsov, ed., Department of Numerical Mathematics, USSR Academy of Sciences, Moscow, 1990, pp. 55–72 (in Russian).
[11] L. Yu. Kolotilina and A. Yu. 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.
[12] L. Yu. Kolotilina and A. Yu. Yeremin, *Factorized sparse approximate inverse preconditionings.* I: *Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
[13] B. Medeke, *On algebraic multilevel preconditioners in lattice gauge theory*, in Numerical Challenges in Lattice Quantum Chromodynamics, Lecture Notes in Comput. Sci. and Engrg. 15, A. Frommer, T. Lippert, B. Medeke, and K. Schilling, eds., Springer-Verlag, Berlin, 2000, pp. 99–114.
[14] I. Montvay and G. Münster, *Quantum Fields on a Lattice*, Cambridge University Press, Cambridge, 1994.
[15] D. Pollard, *Convergence of Stochastic Processes*, Springer-Verlag, New York, 1984.
[16] J. Sexton and D. Weingarten, *Error estimate for the valence approximation and for a systematic expansion of full QCD*, Phys. Rev. D, 55 (1997), pp. 4025–4035.
[17] C. Thron, S. J. Dong, K. F. Liu, and H. P. Ying, *Padé-Z_2 estimator of determinants*, Phys. Rev. D, 57 (1998), pp. 1642–1653.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 819–839
© 2002 Society for Industrial and Applied Mathematics

ADAPTIVE EIGENVALUE COMPUTATIONS USING NEWTON'S METHOD ON THE GRASSMANN MANIFOLD*

EVA LUNDSTRÖM† AND LARS ELDÉN†

Abstract. We consider the problem of updating an invariant subspace of a large and structured Hermitian matrix when the matrix is modified slightly. The problem can be formulated as that of computing stationary values of a certain function with orthogonality constraints. The constraint is formulated as the requirement that the solution must be on the Grassmann manifold, and Newton's method on the manifold is used. In each Newton iteration a Sylvester equation is to be solved. We discuss the properties of the Sylvester equation and conclude that for large problems preconditioned iterative methods can be used. Preconditioning techniques are discussed. Numerical examples from signal subspace computations are given in which the matrix is Toeplitz and we compute a partial singular value decomposition corresponding to the largest singular values. Further we solve numerically the problem of computing the smallest eigenvalues and corresponding eigenvectors of a large sparse matrix that has been slightly modified.

Key words. conjugate gradient method, differential geometry, eigenvalue, eigenvector, Grassmann manifold, Newton's method, preconditioner, signal subspace problem, singular values and vectors, sparse matrix, Toeplitz matrix

AMS subject classifications. 65F15, 49M15, 53B20

PII. S0895479899354688

1. Introduction.

In many applications there are problems which involve computing several partial matrix eigendecompositions successively. This is often called the subspace tracking problem and arises, for instance, when a system varies with time and new approximations are requested at regular intervals. Also, when new information is added and the available quantities need to be adjusted, the partial eigendecomposition has to be updated. In various signal processing applications [8, 9, 10, 41, 38] the matrix is modified by a rank-one matrix consisting of recently arrived data. A related problem is to update the singular value decomposition (SVD) when a new row is appended to the data matrix. In [22, Chapter 8] a method for adaptive eigenvalue computations was presented that was specifically designed for this case.

Here we are interested in the case when the elements of a Hermitian matrix $A(t)$ are assumed to vary slowly with time such that $A(t)$ is a continuous function of t. Since the eigenvalue problem for Hermitian matrices is robust, the eigendecompositions of the successive matrices will not vary substantially. This situation appears, for instance, in airborne radar applications [43], in medicine [44], and in electronic structure computations (nonlinear eigenvalue problems) [16].

In this paper we will describe a general approach for adaptive eigenvalue computations, subspace tracking, that was originally proposed by Edelman, Arias, and Smith in [15]. The basic idea of the method is to take advantage of the geometry of the constraint that the eigenvectors in an invariant subspace of a Hermitian matrix are orthogonal. This can be formulated in the language of differential geometry, and the algorithm we will use is Newton's method on the Grassmann manifold. The pur-

*Received by the editors April 20, 1999; accepted for publication (in revised form) by A. Edelman August 24, 2001; published electronically January 23, 2002.
http://www.siam.org/journals/simax/23-3/35468.html

†Department of Mathematics, Linköping University, S-581 883 Linköping, Sweden (evlun@math.liu.se, laeld@math.liu.se). The work of the first author was supported by a grant from the Swedish Research Council for Engineering Sciences.

pose of the paper is to investigate the usefulness of the Newton–Grassmann approach for adaptive eigenvalue computations for two classes of problems:

1. *Signal subspace problem:* $A = T^H T$, where $T \in \mathbb{C}^{m\times n}$ is a Toeplitz matrix, and we want to compute a partial SVD of T corresponding to the largest singular values. Often there is a gap between the d largest singular values and the rest. Note that the eigenvalues of A are the squares of the singular values of T. Also note that matrix–vector multiplication by T and T^H can be performed in $O((m+n)\log(m+n))$ operations using the fast Fourier transform (FFT); see, e.g., [42, 22].
2. *Smallest eigenvalues of a large, sparse, positive definite matrix:* We have in mind applications, e.g., from computational physics and structural mechanics, and as a model problem we consider a matrix close to the discrete Laplace operator.

The Hermitian matrix A may change arbitrarily over time, but we assume that it varies slowly enough so that the previous subspace approximation is a good approximation of the present.

The Newton–Grassmann method is related to several other methods in the literature, in particular Newton-based approaches [39, 14, 7, 12, 21, 29, 2] and the Jacobi–Davidson method [35]. In connection with a taxonomy of methods in [15] it is stated that several existing methods can be thought of as approximations of Newton's method on the Grassmann manifold. With this in mind it is of some interest whether or not the basic method can be used efficiently in actual computations.

In section 2 of this paper we give a short introduction to the maximization problem on the Grassmann manifold that gives as a solution an orthogonal basis of the invariant subspace corresponding to the largest eigenvalues. When Newton's method is applied to the maximization (or minimization) problem on the manifold, a certain Sylvester equation results. We discuss some aspects of the numerical solution of this equation in section 3. When only a small number of eigenvalues and eigenvectors are to be determined, one of the matrices in the Sylvester equation is of small dimension, which means that we can easily transform the matrix equation into independent linear systems. In the case when the eigenvalues of largest modulus are required, e.g., the signal subspace problem, the systems are demonstrated to be Hermitian positive definite and well-conditioned. Therefore iterative methods, e.g., the conjugate gradient (CG) method, will converge fast. In other contexts, for instance when a number of the smallest eigenvalues are to be computed, it may be necessary to precondition the linear systems. Also in this case the linear systems are Hermitian and positive definite. We discuss the implementation of preconditioning in section 4. For Toeplitz matrices a natural choice is to precondition by circulant matrices. For sparse matrices, when the eigenvalues of smallest modulus are sought, one could try incomplete factorizations as preconditioners. A few numerical experiments are reported in section 5, and in an appendix some similarities and differences between the present approach and the Jacobi–Davidson method are pointed out.

2. Newton's method on the Grassmann manifold. Let A have the eigenvalues $(\lambda_i)_{i=1}^n$ and assume that they are ordered $\lambda_1 \geq \cdots \geq \lambda_n > 0$. Let $Y \in \mathbb{C}^{n\times d}$, where $d < n$. From the Courant–Fischer minimax theorem (see, e.g., [17, Theorem 8.1.2]) it is seen that the solution of the maximization problem

$$\max_{Y^HY=I} F(Y) = \max_{Y^HY=I} \frac{1}{2}\operatorname{tr}(Y^H A Y) \tag{2.1}$$

is the sum of the d largest eigenvalues of A, and any Y consisting of d orthonormal vectors that span the corresponding invariant subspace will give the optimum. If the smallest eigenvalues are to be determined, then in (2.1) maximization is to be replaced by minimization. This is a well-known approach for symmetric eigenvalue problems; see, e.g., [1].

In [15] algorithms with orthogonality constraints are studied using ideas and techniques from differential geometry. It is shown that standard methods for optimization in Euclidean space, such as Newton's method and the CG method, can be generalized to manifolds and realized numerically. In particular, (2.1) can be considered as a maximization problem on the Grassmann manifold, and by using, e.g., Newton's method on this manifold, the constraint $Y^H Y = I$ is implemented automatically. One important contribution of [15] is that explicit, computable expressions are given for tangent spaces, geodesics on the Grassmann manifold, etc., as opposed to the practice of the standard literature in differential geometry, where usually a coordinate-free approach is used. This makes it possible to identify some existing eigenvalue algorithms as approximations of Newton's algorithm on the Grassmann manifold.

We provide here a very brief—and incomplete—review of some basic concepts from differential geometry that are relevant in our context. For more details, see [15]. First we need to define the Stiefel manifold $\mathcal{V}_{n,d}$ of "tall, skinny" matrices with orthonormal columns:

$$\mathcal{V}_{n,d} = \{Y \in \mathbb{C}^{n\times d} \mid Y^H Y = I\}.$$

The Grassmann manifold $\mathcal{G}_{n,d}$ is then obtained by identifying those matrices in $\mathcal{V}_{n,d}$ whose columns span the same subspace. Thus, a point on the Grassmann manifold is an equivalence class of $n \times d$ matrices with orthonormal columns, where two matrices are equivalent if they are related by right multiplication of a unitary $d \times d$ matrix. Therefore, $\mathcal{G}_{n,d} = \mathcal{V}_{n,d}/\mathcal{O}_d$, where $\mathcal{O}_d$ is the group of $d \times d$ unitary matrices. If $Y \in \mathbb{C}^{n\times d}$ is a matrix with orthonormal columns, then $[Y]$ denotes the corresponding equivalence class, i.e., the point on the Grassmann manifold.

For Newton's method on the Grassmann manifold we must compute the gradient of the function $F(Y)$ defined on the manifold. It can be shown to be

$$\nabla F = F_Y - YY^H F_Y = \Pi AY,$$

where $\Pi = I - YY^H$ projects onto the tangent space at $[Y]$. Let $Y_\perp \in \mathbb{C}^{n\times(n-d)}$ be a matrix such that $(Y\ Y_\perp)$ is unitary. Then $\Pi = Y_\perp Y_\perp^H$, and $\mathcal{R}(Y_\perp)$ is the tangent space T_Y at $[Y]$.

In Newton's method we determine a correction $\Delta = -(\text{Hessian})^{-1}\nabla F$, which here becomes the linear problem

$$\Pi A\Delta - \Delta Y^H AY = -\Pi AY, \tag{2.2}$$

with the requirement that Δ is in the tangent space T_Y at $[Y]$,

$$Y^H \Delta = 0. \tag{2.3}$$

Now define $H = Y^H AY$. Obviously, H is the matrix from which Ritz approximations of the eigenvalues are computed [26, Chapter 11], and

$$R = R(Y) = \Pi AY = (AY - YH)$$

is the residual matrix for the eigenvalue approximation.

The requirement that (2.3) is satisfied is equivalent to $\Pi\Delta = \Delta$, and therefore it follows that (2.2) is equivalent to the *Hermitian Sylvester equation:*

$$\Pi A\Pi\Delta - \Delta H = -R. \tag{2.4}$$

For Newton's method in Euclidean space the correction is added to the present iterate. On the manifold the corresponding operation is to follow a geodesic path in the direction Δ. This can be written

$$Y := YV_\Delta \cos(\Sigma_\Delta)V_\Delta^H + U_\Delta \sin(\Sigma_\Delta)V_\Delta^H, \tag{2.5}$$

where $U_\Delta\Sigma_\Delta V_\Delta^H$ is the compact singular decomposition of Δ (i.e., $U_\Delta \in \mathbb{C}^{n\times d}$ and $\Sigma_\Delta \in \mathbb{C}^{d\times d}$).

We can now formulate the algorithm [15].

ALGORITHM 1. NEWTON'S METHOD FOR MAXIMIZING $F(Y)$ ON THE GRASSMANN MANIFOLD.

Given Y such that $Y^HY = I$.

repeat until convergence

1. Compute $H := Y^HAY$ and $R = R(Y) := (AY - YH)$.
2. Solve $\Pi A\Pi\Delta - \Delta H = -R$.
3. Compute updated Y using (2.5).

end

We will see later that in some cases it is advantageous to update columns of Y in a sequential manner.[1] Assume, e.g., that $\Delta = (0 \cdots 0\, d_i\, 0 \cdots 0)$ for some vector d_i. It is easy to show that in this case the update along a geodesic becomes

$$Y := (\, y_1 \quad \cdots y_{i-1} \quad \cos\delta\, y_i + \sin\delta\, u_i \quad y_{i+1} \quad \cdots \quad y_d\,), \tag{2.6}$$

where $\delta = \|d_i\|_2$ (Euclidean norm), and u_i is the ith column of U_Δ.

It is shown in [15] that several algorithms for simultaneously improving a set of eigenvector approximations can be considered as variations of Newton's method on the Grassmann manifold. In particular, a simultaneous Rayleigh quotient iteration can be defined by letting Z be the solution of the Sylvester equation

$$AZ - ZH = Y \tag{2.7}$$

and then putting the correction equal to $D = -Y + Z(Y^HZ)^{-1}$. For $d = 1$ this is mathematically equivalent to Newton's method on the Grassmann manifold, but for $d > 1$ the procedures are different in general. (However, D is always in the tangent space T_Y.)

The equation (2.4) can also be derived based on eigenvalue perturbation theory [39] and is used for the refinement of eigenvalue approximations in [33].

3. Solving the Sylvester equation. The major work computationally in Algorithm 1 is solving (2.4). Sylvester equations arise in many areas of applied mathematics, and several efficient methods to solve them have been developed. When the matrices in the equation are small, direct methods can be used, but if one or both matrices are large, then iterative methods must be used to compute an approximate solution. Extensive reference lists of methods for solving the Sylvester equation can be found in [19, 31].

[1] In analogy to the Gauss–Seidel method for linear systems.

Obviously, since we assume that d is small, it is cheap to decouple (2.4) into d separate linear systems. Let $H = G\Phi G^H$ be the eigendecomposition of H, where $\Phi = \mathrm{diag}(\tau_1, \tau_2, \ldots, \tau_d)$, and $G = (\, g_1 \quad g_2 \quad \ldots \quad g_d \,)$ is unitary. Inserting this in (2.4) we get

$$\Pi A \Pi \Delta - \Delta G \Phi G^H = -R.$$

Postmultiplying by G gives

$$(\Pi A \Pi)(\Delta G) - (\Delta G)\Phi = -RG.$$

Equating each column separately yields the d equations

$$(\Pi A \Pi - \tau_i I)\delta_i = -r_i, \quad i = 1, 2, \ldots, d, \tag{3.1}$$

where we have used the notation $\delta_i = \Delta g_i$ and $r_i = Rg_i$. Note that, since Δ and R are orthogonal to Y, the same is true for δ_i and r_i. When these systems have been solved, we obtain the solution Δ simply by postmultiplying[2] the matrix $(\, \delta_1 \quad \delta_2 \quad \ldots \quad \delta_d \,)$ by G^H.

In the discussions here and in section 4, it is important to keep in mind that the equations (3.1) are not really defined on the whole space but rather on the tangent space, $T_Y = \mathcal{R}(Y_\perp)$. Using the facts that $\delta^H Y = 0$, implying $\delta = \Pi\delta$, and that $\Pi^2 = \Pi$, we can rewrite (3.1) in the form

$$\Pi(A - \tau_i I)\Pi\delta_i = -r_i. \tag{3.2}$$

Since Π projects onto $\mathcal{R}(Y_\perp)$, and since the right-hand side is in this space, (3.2) is seen to be an equation on the tangent space.

The systems (3.1) will be solved using iterative methods for linear systems. By keeping the equation matrix $\Pi A \Pi - \tau_i I$ in factored form, the matrix–vector multiplication is fast, provided that it is fast for A. Note that the matrix $\Pi A \Pi - \tau_i I$ is Hermitian, and if it is also positive definite we can use the CG method to solve these linear systems. If $\Pi A \Pi - \tau_i I$ is indefinite, which will be the case when we want to compute eigenvalues in the middle of the spectrum, then MINRES or GMRES can be used instead. We refer, e.g., to [27, 18] for comprehensive surveys of iterative methods for linear systems.

For the signal subspace problem we assume that $\mathcal{R}(Y)$ is close to the space spanned by the eigenvectors of A corresponding to the d largest eigenvalues. Then, in $\Pi A \Pi$, these eigenvalues are replaced by d zero eigenvalues (exactly), with eigenvectors equal to the columns of Y. The remaining $n-d$ eigenvalues of $\Pi A \Pi$ are approximately equal to

$$\{\lambda_{d+1}, \lambda_{d+2}, \ldots, \lambda_n\},$$

with eigenvectors that are orthogonal to Y and are approximations of the eigenvectors of A with the same eigenvalues. Since the quantities τ_i, $i = 1, 2, \ldots, d$, are Ritz values of A from the space $\mathcal{R}(Y)$, they will approximate the d largest eigenvalues of A well if Y is a good approximation. Therefore the matrix $(\Pi A \Pi - \tau_i I)$ can be expected to be negative definite. Moreover, if we assume that there is a significant gap between

[2]However, this is not necessary, since the solution is defined on the Grassmann manifold, where postmultiplication by a unitary matrix does not matter.

λ_d and λ_{d+1}, then the Ritz values τ_i will be large compared to the small eigenvalues λ_i, $i = d+1, d+2, \ldots n$, and the eigenvalues of $\Pi A\Pi - \tau_i I$ will be fairly well clustered around $-\tau_i$ (cf. Figure 5.2 below).

STATEMENT 3.1. *Assume that we want to compute the d largest eigenvalues of A and that Y is a good approximation of the basis for the corresponding invariant subspace of A. Let τ be a Ritz value for one of the requested eigenvalues. If the separation between λ_d and λ_{d+1} is large enough, then the matrix $\Pi A\Pi - \tau I$ is negative definite on the tangent space T_Y.*

In the beginning of the Newton iterations, when $\mathcal{R}(Y)$ is only a moderately good approximation to the subspace spanned by the first d eigenvectors of A, it may happen that τ_i is smaller than the largest eigenvalue of $\Pi A\Pi$. Then the matrix $(\Pi A\Pi - \tau_i I)$ will be indefinite and the CG method is not certain to converge. In our experiments we have used CG for the equations (3.1) and it never failed to converge. This is probably due to the fact that the requested subspace is only slightly modified between two successive matrices, which means that we always had a good starting approximation.

Next consider the case when we want to compute the d smallest eigenvalues, and assume that Y is a good approximation of a basis of the invariant subspace. The corresponding eigenvalues of $\Pi A\Pi$ are exactly equal to zero, with eigenvectors equal to the columns of Y. Therefore, if τ is a good approximation of one of the smallest eigenvalues of A, then the matrix $\Pi A\Pi - \tau I$ will have an eigenvalue of multiplicity d at $-\tau$, with eigenvectors equal to the columns of Y. If the d smallest eigenvalues of A are well separated from the rest, and if Y is a good approximation, then we can expect the $n-d$ remaining eigenvalues of $\Pi A\Pi - \tau I$ to be positive. Thus, the matrix $\Pi A\Pi - \tau I$ is indefinite on the whole space but positive definite on the tangent space, $T_Y = \mathcal{R}(Y_\perp)$.

STATEMENT 3.2. *Assume that we want to compute the d smallest eigenvalues of A and that Y is a good approximation of the basis for the corresponding invariant subspace of A. Let τ be a Ritz value for one of the requested eigenvalues. If the separation between λ_{n-d+1} and λ_{n-d} is large enough, then the matrix $\Pi A\Pi - \tau I$ is positive definite on the tangent space T_Y.*

We conclude this section by pointing out one seemingly significant difference between the Grassmann manifold and simultaneous Rayleigh quotient approaches. The equations (2.7) are inherently ill-conditioned, and when direct methods are used, it is the ill-conditioning that ensures success; see, e.g., [17, pp. 362–364]. It is less clear how to solve the equations iteratively.[3] If the separation between the wanted and unwanted eigenvalues is large enough (and this is often the case; see Example 2 in section 5), then (3.2) is well-conditioned, which should make it accessible to iterative solution.

4. Preconditioning the linear systems. This section is devoted to preconditioning techniques for the system of equations (3.1). For clarity of presentation, we omit the subscripts here and in the following section. The difficulty with finding a preconditioner for the matrix $\Pi A\Pi - \tau I$ is that even if A possesses some particular structure (e.g., Toeplitz), which makes a good preconditioner easy to find, the matrix $\Pi A\Pi$ does not in general have that structure. The first step toward circumventing this problem is to rewrite (3.1) into the more convenient form (3.2). By keeping the three factors in the matrix $\Pi(A - \tau I)\Pi$ separate, the structure of A can be utilized.

[3] However, it is shown in [32] that in the case $d = 1$ the Newton–Grassmann method is equivalent to Rayleigh quotient iteration, and the convergence of iterative methods can be quite fast.

An equation similar to (3.2) (where Y consists of one single vector only) arises in the Jacobi–Davidson method as the correction equation for the new update. The problem of preconditioning this equation has been studied by Sleijpen et al. in [34, 36]. They propose three methods. In this section we will outline the procedure in [36]. It is straightforward to generalize it to the case when Y consists of d orthogonal columns (see [36]).

Assume that we have a Hermitian preconditioner P for the matrix $A-\tau I$, where $Px = y$ can be solved easily. In what follows, when we refer to P^{-1}, we actually mean solving a system $Px = y$. For a preconditioner to work for (3.2), it must be restricted to the subspace $\mathcal{R}(Y_\perp)$. We will therefore use the projected preconditioner

$$\tilde{P} = \Pi P \Pi. \tag{4.1}$$

The preconditioner will be applied from the left and we will solve

$$\tilde{P}^{-1}\Pi(A-\tau I)\Pi\delta = -\tilde{P}^{-1}r. \tag{4.2}$$

Here and in the following, when we use the notation $\tilde{P}^{-1}$, we mean the inverse of $\tilde{P}$ restricted to $\mathcal{R}(Y_\perp)$. As long as P itself is nonsingular, the restricted inverse exists trivially. At first we assume that P is nonsingular, and then we discuss the case when P is singular or close-to-singular.

In the experiments in section 5 we will use the CG method and GMRES. Preconditioning for these two methods is discussed in detail in [27, Chapter 9]. Regardless of which iterative method is used, the only operation in which the projected preconditioner appears is to find the solution $w \in \mathcal{R}(Y_\perp)$ of a linear system

$$\Pi P \Pi w = z, \tag{4.3}$$

where $z \in \mathcal{R}(Y_\perp)$. We outline the procedure for how this is done in Algorithm 2. For more details, see [36]. Note that the vectors generated in the iterative procedures are always in the tangent space $\mathcal{R}(Y_\perp)$.

ALGORITHM 2. PRECONDITIONER SOLVE IN $\mathcal{R}(Y_\perp)$.
Given $z \in \mathcal{R}(Y_\perp)$, $N = Y^H\hat{Y}$, $\hat{Y} = P^{-1}Y$, solve $\Pi P\Pi w = z$.

1. Solve the linear system $P\hat{z} = z$.
2. Compute $f = Y^H\hat{z}$.
3. Solve $Na = -f$.
4. Compute $w = \hat{z} + \hat{Y}a$.

Since Y does not change during the iterations for solving the linear system (3.2), the matrices $\hat{Y}$ and N have to be computed only once at the beginning of each Newton step. Note that the matrix N is of dimension $d \times d$ and its Cholesky factorization can be computed cheaply once before the iterations start.

As the Newton iterations proceed, the approximate eigenvalue τ becomes closer and closer to an exact eigenvalue of the matrix A. This means that the matrix $A-\tau I$ turns more and more into a singular matrix. If the preconditioner P approximates $A-\tau I$ well, then also P will be close-to-singular. Actually we do not want to exclude the possibility that P is exactly singular. We will now assume that P is exactly singular. Of course, for $\tilde{P} = \Pi P\Pi$ to be nonsingular on $\mathcal{R}(Y_\perp)$ we must require that

$$\mathcal{N}(P) \cap \mathcal{R}(Y_\perp) = \{0\}.$$

In step 1 of Algorithm 2, we can use the Moore–Penrose pseudoinverse of P. However, by doing this, we cannot be certain that the result fulfills the requirement $Y^Hw = 0$.

Further, it may happen that P does not have full rank on $\mathcal{R}(Y)$, which implies that $Y^H P^+ Y$ is singular. To circumvent such problems one may proceed as follows. Rewriting (4.3) using that $w, z \in \mathcal{R}(Y_\perp)$ we obtain

$$\Pi(Pw - z) = 0,$$

implying that $Pw - z \in \mathcal{R}(Y)$. Thus we have the equation

$$Pw = z + Ya.$$

From the requirement that $(z + Ya) \in \mathcal{R}(P)$ or, equivalently, $(z + Ya) \perp \mathcal{N}(P)$, we get

$$V^H Ya = -V^H z,$$

where the columns of $V \in \mathbb{C}^{n\times r}$ are orthonormal basis vectors which span $\mathcal{N}(P)$. We can write $w = P^+(z + Ya) + Vc$ for some vector c, to be determined. Then by imposing $Y^H w = 0$ we get the linear system

$$\begin{pmatrix} Y^H P^+ Y & Y^H V \\ V^H Y & 0 \end{pmatrix} \begin{pmatrix} a \\ c \end{pmatrix} = - \begin{pmatrix} Y^H P^+ z \\ V^H z \end{pmatrix}.$$

If this matrix is nonsingular, then we get a unique solution w. Nonsingularity can be proved if P^+ is positive semidefinite. This can be the case when one computes the smallest singular values and chooses to work with a preconditioner which approximates the matrix A instead of $A - \tau I$. The case when P^+ is indefinite needs more research to give a better understanding of the problem.

To check whether the preconditioner is close-to-singular might not be a simple task, since it involves investigating its eigenvalues, which in general are unknown. However, there are preconditioners where this can be directly verified. One such situation is when circulant preconditioners are used. We refer to [20] or section 4.1 for discussions of circulant preconditioners for this type of problem. To see if the matrix N is singular is easy using its SVD. Both P and N can be treated as singular if the smallest eigenvalue is below a certain tolerance.

In each Newton iteration of Algorithm 1, d linear systems of the form

$$\Pi(A - \tau_i I)\Pi\delta_i = -r_i, \quad i = 1, 2, \ldots, d,$$

must be solved. The fact that the matrix $A - \tau_i I$ varies with i complicates matters if a new preconditioner is required for each system. Assume, for instance, that A is sparse and that an incomplete LU (ILU) factorization (see [23] or, e.g., [17, p. 535]) is used as a preconditioner. Then one alternative is to recalculate the ILU factorization for each τ_i. However, to decrease the cost for solving these linear systems one can use the same preconditioner computed for one particular value of τ in all Newton steps; cf. sections 4.2 and 5.3. This would probably slow down the convergence, requiring more steps, but each step would be considerably cheaper. Besides the gain in only having to compute one matrix factorization at each Newton step, we would also need to compute the matrices $\hat{Y}$ and N (and its inverse) only once per step.

In the signal subspace problem, when the matrix is $A = T^H T$, where T is a Toeplitz matrix, the situation is considerably simplified. In this case circulant matrices can be used as preconditioners. Not only can a circulant matrix be easily inverted, but it is also easy to find a new preconditioner for each matrix $A - \tau_i I$. This will be discussed in the next section.

4.1. Circulant preconditioners for Toeplitz matrices. Circulant matrices are suitable as preconditioners for two reasons: (a) they are easily inverted, and (b) they are completely described by the first column and need only n places of storage. Recall that a circulant matrix C is diagonalized by the discrete Fourier matrix, $C = F^H \Theta F$, with $\Theta = \mathrm{diag}(Fc)$, where c is the first column of C [11, 42, 22]. Hence, the matrix C can easily be inverted by using the FFT, $w = C^{-1}z = F^H \Theta^{-1} F z = F^H((1./(Fc)).*(Fz))$, where "./" and ".*" denote elementwise division and multiplication, respectively.

Circulant preconditioners for linear systems are discussed in [40, 4, 5]. In our application we cannot use such a preconditioner directly, since the matrix $A - \tau I = T^H T - \tau I$ is not Toeplitz in general. In [20] (see also [22]) the matrix T is partitioned

$$T = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_l \end{pmatrix},$$

where each T_j is a square $n \times n$ Toeplitz matrix. For all blocks T_j, $j = 1, 2, \ldots, l$, construct circulant preconditioners C_j. Using these we can approximate $T^H T$ by

$$T^H T = \sum_{j=1}^{l} T_j^H T_j \approx \sum_{j=1}^{l} C_j^H C_j = F^H \left(\sum_{j=1}^{l} |\Theta_j|^2 \right) F = F^H \Theta F =: C,$$

where $C_j = F^H \Theta_j F$ is the eigendecomposition of C_j, $j = 1, 2, \ldots, l$. Here $|\cdot|$ denotes elementwise absolute value. Since Θ is a diagonal matrix, the matrix C is also a circulant and can be used as a preconditioner for $A = T^H T$. Note that C depends only on A and can be computed once at the beginning of the Newton iterations ($O(ln \log_2(n))$ operations). It can then be updated for $A - \tau I$ by simply subtracting τI from Θ. Also, a (nearly) singular or indefinite preconditioner can easily be detected by checking $\Theta - \tau I$.

4.2. Preconditioner based on incomplete Cholesky factorization. In this subsection we will briefly discuss preconditioning for the problem of computing a few of the smallest eigenvalues and corresponding eigenvectors of a sparse Hermitian, positive definite matrix. Incomplete factorizations are standard preconditioners for solving sparse linear systems. There is a vast literature on this subject; we here refer only to the presentations in [17, Chapter 10.3.2], [27, Chapter 10.3], and [13, Chapter 9.3].

From [36] we quote: "For the discrete Poisson operator A it has been shown that an incomplete decomposition K has almost the same eigenvectors as A for a few of the smallest eigenvalues.... If such a preconditioner is used for values of τ close to zero, then the ILU-process will yield a preconditioner that is also effective for a number of nearby small values." Based on this, we propose to let the matrix P in (4.1) be an incomplete Cholesky preconditioner of A. Furthermore, we will use the same preconditioner for all linear equations (3.1).

Obviously, if P is positive definite, then its restriction to the tangent space $\mathcal{R}(Y_\perp)$ is also positive definite.

5. Numerical experiments. In this section a few experiments are carried out to show how the Newton method on the Grassmann manifold can perform when

computing the largest (signal subspace problems) and the smallest eigenvalues. A few tests of preconditioners in the inner iterations are done. The results are only preliminary and more research is needed for a complete understanding of the methods. However, the experiments still give some hints of the advantages and disadvantages of each method.

The first two examples were run on a Dec Alpha $200^{4/166}$ and the third on a Sun Ultra 1. We used MATLAB 5.2, in IEEE double precision with unit roundoff $\mathbf{u}$ approximately equal to $2 \cdot 10^{-16}$. As starting matrix Y in the Newton iterations for the succeeding matrices, the approximations from the previous matrix was used. For the first matrix $A(1)$, the requested eigenvectors were computed with the routine `eig` in MATLAB. In the iterative methods for solving the linear systems (3.1) we used the starting vector $(\,0 \quad 0 \quad \cdots \quad 0\,)^T$. The reason for this is that Δ is a correction to the matrix Y. When Y is a good approximation we can expect the elements in Δ to be small in magnitude, i.e., the vectors δ_i are close to the zero-vector.

The linear systems (3.1) were solved with the CG method. Recall from the discussion in section 3 that when the subspace corresponding to the d largest eigenvalues is requested and the separation to the unwanted eigenvalues is large, we can expect good performance when a fairly good approximation is available from the start. Besides the ordinary CG method we used a variant designed for multiple linear systems [6]. These will be called NCG and NCG-seed, respectively. In the procedure NCG-Ymod, we update a column of Y (2.6) as soon as a linear system (3.1) has been solved. (This affects the projection matrix in the linear systems (3.2) only; the shifts τ_i are kept fixed during one whole Newton step.)

Also different preconditioners were tried. In the first two examples we used GMRES with a circulant preconditioner, which we allowed to be indefinite; cf. section 4.1. In Example 5.3 we used CG with an incomplete Cholesky preconditioner. We refer to these three procedures as NCG, NGM-Circ, and NCG-ICh, respectively.

When the unpreconditioned CG method is used the residual norm $\|-r_i-(\Pi A\Pi-\tau_i)\delta_i^{(j)}\|_2$, where $\delta_i^{(j)}$ is the jth iterate of δ_i, is available in every iteration and is thus suitable to use in a convergence check. The same is not true for the case when a left preconditioner is used. Instead the norm $\|\tilde{P}^{-1}(-r_i-(\Pi A\Pi-\tau_i)\delta_i^{(j)})\|_2$ can be used to guard the convergence. Thus, the unpreconditioned iterations were terminated when

$$\| - r_i - (\Pi A\Pi - \tau_i)\delta_i^{(j)}\|_2 \leq \epsilon,$$

and the preconditioned iterations when

$$\|\tilde{P}^{-1}(-r_i - (\Pi A\Pi - \tau_i)\delta_i^{(j)})\|_2 \leq \epsilon_{\text{prec}}.$$

In the experiments the values of these two norms were quite different, so we had to calibrate the tolerances ϵ and ϵ_{pgmres} to make the stopping criteria comparable.

When the eigenvectors corresponding to the largest eigenvalues were requested, the Newton iterations were terminated when the inequalities

$$\frac{\|Ay_i - \tau_i y_i\|_2}{|\tau_i|} < \varepsilon, \quad i = 1, 2, \ldots, d, \tag{5.1}$$

were fulfilled. Since $|\tau_i| \leq \|A\|_2$, $i = 1, 2, \ldots, 5$, this stopping criterion ensures that, for each Ritz pair, the backward error is less than ε [30]. If the smallest eigenvalues are sought, the norm of the absolute residual $\|Ay_i - \tau_i y_i\|_2 < \varepsilon$ is more appropriate. Note

that all quantities in the criterion are used in the next Newton step, so essentially no extra cost is required for the convergence check. In our experiments we used the tolerance $\varepsilon = 1 \cdot 10^{-7}$.

5.1. Example 1: Rank-ten update in a signal subspace problem. In this experiment we determined the right singular vectors of a sequence of Toeplitz matrices $T(t)$, $t = 1, 2, \ldots$, by computing the corresponding Ritz vectors of the Hermitian matrix $A(t) = T(t)^H T(t)$.

We used a model problem [22, section 2.1] to investigate how the Newton method performs on matrices which differ by rank-ten updates. The signal consists of a sum of exponentially damped sinusoids,

$$\hat{x}_j = \sum_{l=1}^{k} c_l \exp[(\alpha_l + i\omega_l)t_j + i\phi_l] + \theta_j, \quad j = 1, 2, \ldots, m+n-1, \tag{5.2}$$

where θ_j is additive Gaussian noise. We used $k = 5$.

The values of the parameters can be found in Table 5.1, and the noise variance was set to $\nu = 1$. The signal was sampled for $t = 0 : 10^{-4} : (2n-1)10^{-4}$ for different values of n, and the samples were put together in a square Toeplitz matrix T. Two sizes of the matrix were used. The smaller was a 256×256 matrix and the larger was of size 512×512.

TABLE 5.1
Sinusoid parameters for Example 1.

c_l	α_l	$\omega_l/(2\pi)$	$\phi_l/(\pi/180)$
6.1	-208	-1397	15
9.9	-256	-685	15
6.0	-197	-271	15
2.8	-117	353	15
17.0	-808	478	15

In the experiments we computed the invariant subspace of dimension 5 corresponding to the largest singular values of 15 successive matrices. Each matrix differed from the previous one by 10 extra top rows. For the Hermitian matrix $A(t)$ this corresponds to adding a rank-ten matrix to the previous matrix $A(t-1)$. The first matrix $T(1)$ consisted of the $n-140$ bottom rows of T, and the last matrix $T(15)$ was equal to T. The first 25 singular values of $T(1)$ and $T(15)$, respectively, are shown in Figure 5.1.

The termination tolerances in the inner iterations were set to $\epsilon = 10^{-4}$ and $\epsilon_{\text{prec}} = 10^{-7}$.

The results of computing the approximate signal subspaces of dimension 5 are shown in Table 5.2. Only the last 14 matrices were used in the measurements, since for the first matrix `eig` was used. The results in Table 5.2 show the average number of Newton steps, Toeplitz matrix–vector multiplications, inner iterations per eigenvalue at each Newton step, and floating point operations, respectively, for these 14 matrices.

In our experiments Toeplitz matrix–vector multiplications were carried out in $O(n \log n)$ operations using FFT (see [42, 22]). Since exactly the same number of Newton steps were carried out and the same number of multiplications were performed per inner iteration for the first three methods, the difference in the number of matrix–vector multiplications is due to the difference in the number of inner iterations. The results in Table 5.2 show that the preconditioning helps to decrease

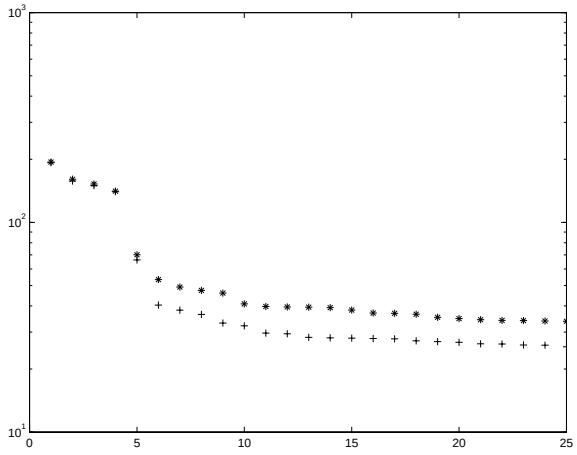


FIG. 5.1. *The* 25 *largest singular values of* $T(1) \in \mathbb{C}^{116\times 256}$ ("+") *and for* $T(15) \in \mathbb{C}^{256\times 256}$ ("∗") *for Example* 1 *with noise level* $\nu = 1$.

TABLE 5.2

Average results from 15 *successive matrices in Example* 1 *where signal subspaces of dimension* $k = d = 5$ *were approximated. "Toeplitz multiplications" refers to the number of multiplications of* T *or* T^H *by a vector, "inner iterations" to the average number of inner iterations per eigenvalue at each Newton step, and "flops" to real floating point operations.*

$m \times n$	256×256			
Method	NCG	NCG-seed	NGM-Ymod	NCG-Circ
Newton steps	2	2	2	2
Toeplitz mult.	101	90	98	97
Inner iter.	3.6	3	3.4	3.4
Flops (10^6)	3.6	3.9	3.0	4.9
$m \times n$	512×512			
Newton steps	2.3	2.3	2	2.3
Toeplitz mult.	161	147	146	138
Inner iter.	5.6	5	5.8	4.6
Flops (10^6)	10.2	12.6	8.3	13.7

the number of iterations somewhat compared to NCG. However, the extra overhead still makes NGM-Circ more expensive. We gain some Toeplitz matrix–vector multiplications by using the seed method, but again the overhead was noticeable in the total cost. Finally, when Y was modified during a Newton step, i.e., when the most recent approximations were used, the number of Newton steps was smaller for the larger matrices and the flop count was decreased substantially. Also for the smaller problem, when the number of Newton step was the same, the total cost decreased.

5.2. Example 2: The Cadzow algorithm. In the second example we used the Cadzow algorithm [3] to extract a signal from noisy data. In our model problem (5.2) the Toeplitz matrix T is of rank k when no noise is present but has larger rank when the signal is contaminated by noise. The aim in this procedure is to produce a Toeplitz matrix with rank equal to k from the noisy data and from this matrix obtain an approximation to the noise-free signal. This is an optimization problem; to find a matrix with the Toeplitz and low rank properties as close as possible in Frobenius

norm to the original matrix T. In Cadzow's algorithm this optimum is searched for[4] by optimizing with respect to one property at the time. More specifically, starting with the Toeplitz matrix $T(1) = T$, we look for the closest possible matrix of rank k. This matrix is given by

$$S(1) = \sum_{j=1}^{k} \sigma_j^{(1)} u_j^{(1)} (v_j^{(1)})^H,$$

where $(u_j^{(1)}, \sigma_j^{(1)}, v_j^{(1)})$, $j = 1, 2, \ldots, k$, are the singular triplets of $T(1)$ corresponding to the k largest singular values.

The matrix $S(1)$ is in general no longer a Toeplitz matrix. Therefore we want to find the Toeplitz matrix which is closest to $S(1)$ in Frobenius norm. This matrix is obtained by computing the mean value of the elements in each diagonal and letting the corresponding diagonal in the Toeplitz matrix $T(2)$ have this value, i.e., the elements in $T(2)$ are

$$\hat{x}_{ij}^{(2)} = \begin{cases} \frac{1}{n-(i-j)} \sum_{l=1}^{n-(i-j)} s_{l+(i-j),l}^{(1)}, & i \geq j, \\ \frac{1}{n-(j-i)} \sum_{l=1}^{n-(j-i)} s_{l,l+(j-i)}^{(1)}, & i < j, \end{cases}$$

where $s_{ij}^{(1)}$ are elements in $S(1)$. Here we have assumed that the matrices are square, but it is straightforward to generalize this to rectangular matrices.

The Cadzow algorithm continues in this way by alternately computing the closest rank-k matrix to a Toeplitz matrix and vice versa. Note that there is no simple structure in the way the successive Toeplitz matrices are updated, and in the beginning their approximate signal subspaces may be far apart. However, the more the method converges, the closer the requested subspaces of the matrices will be, and fewer iterations are likely to be needed in the Newton algorithm. Moreover, when the Toeplitz matrices converge to a rank-k matrix, the gap between the singular values $\sigma_k^{(j)}$ and $\sigma_{k+1}^{(j)}$ increases. We thus expect that the cost for the singular value problems in the algorithm decreases the closer to convergence we get.

The Toeplitz matrices used in these experiments were the same as those in Example 5.1, i.e., we used matrices of dimension 256 and 512. Two succeeding Toeplitz matrices were considered to be equal enough, and the Cadzow iteration thus stopped, when

$$\|\hat{x}(t) - \hat{x}(t-1)\|_2 \leq \epsilon_{\text{cad}},$$

where $\hat{x}(j) = \begin{pmatrix} \hat{x}_1^{(j)} & \hat{x}_2^{(j)} & \cdots & \hat{x}_{2n-1}^{(j)} \end{pmatrix}^T$ for $j = t-1, t$. The superscript stresses that the elements are from $T(j)$. The vectors in the criterion consist of the elements which completely describe the Toeplitz matrices. This is a natural way of computing the distance between two Toeplitz matrices, since they only have $2n-1$ degrees of freedom instead of n^2. We thus can use a vector norm instead of a matrix norm. The tolerance in the experiments was set to $\epsilon_{\text{cad}} = 1 \cdot 10^{-2}$. The other termination tolerances were the same as in Example 1.

The results in Table 5.3 are from the experiments with the Cadzow method. The number of matrices refer to the number of Toeplitz partial SVDs computed, the first

[4]To our knowledge there is no proof that Cadzow's algorithm finds a global minimum, or even that it is convergent.

TABLE 5.3
Results for the singular value problems of the matrices in the Cadzow experiment. A rank-5 matrix was requested. We give the total number of Newton steps, etc., for the whole experiment.

$m \times n$	256×256			
Method	NCG	NCG-seed	NGM-Ymod	NCG-Circ
Matrices	20	20	20	20
Newton steps	27	27	27	27
Toeplitz mult.	852	836	852	1302
Flops (10^6)	36	37	30	67
$m \times n$	512×512			
Matrices	21	21	21	21
Newton steps	31	31	31	31
Toeplitz mult.	1068	1000	1046	1220
Flops (10^6)	86	88	71	141

one excluded. Considering the number of Newton steps, matrix–vector multiplications, and flops, we display the total number for all the succeeding matrices. These measurements are for the singular value computations only, since this is what is of interest to us.

For both test matrices, the experiments with the CG method show the best performance. To explain this we computed the eigenvalues of $\Pi T^H T \Pi - \tau I$, where τ is the approximation of the smallest eigenvalue in the signal subspace. In Figure 5.2 we show the eigenvalues in the first and last steps of the Cadzow procedure. It is seen that already in the first step the eigenvalues are well clustered, leading to fast convergence of the CG method, as discussed in section 3. When the signal subspaces of two successive matrices are very close, as is the case in the last step, the eigenvalues are almost coincident, and only one iteration in the CG method is needed. Thus, preconditioning cannot improve the rate of convergence.

For these kind of problems, the gain in using the seed method or in modifying the matrix Y during the Newton steps is only done for the first matrices. Therefore the difference in terms of matrix–vector multiplications is quite small between NCG, NCG-seed, and NCG-Ymod.

Figure 5.3 shows the number of matrix–vector multiplications and flops required for each matrix in the experiment with the smaller matrix. We see in the figure that less work is needed for all methods the more the Cadzow method converges. The jumps occur when the number of Newton steps is decreased by one. The number of Newton steps was the same for all methods. For the first matrix three steps were required to reach convergence, while only one step was needed for the last matrices.

For comparison we computed the singular values only (i.e., no singular vectors) of a random, complex matrix using the MATLAB function `svd`. A matrix of dimension 256 required $181 \cdot 10^6$ flops and one of dimension 512 needed $1.44 \cdot 10^9$ flops.

To see that the amount of noise indeed was decreased by these computations we show in Figure 5.4 the first 25 singular values of the first matrix $T(1) = T$ ("$*$") and the final matrix $T(21)$ ("+") for the smaller problem. The gap between σ_d and σ_{d+1} has clearly grown and the numerical rank is more distinct.

5.3. Example 3: Smallest eigenvalues and incomplete Cholesky preconditioning. We assume that the d smallest eigenvalues and corresponding eigenvectors of a large sparse matrix $A(0)$ have been computed. We then want to compute the corresponding eigenvalues and eigenvectors of a slightly modified matrix $A(s)$. Here we have in mind a situation, where, e.g., in a structural analysis application one

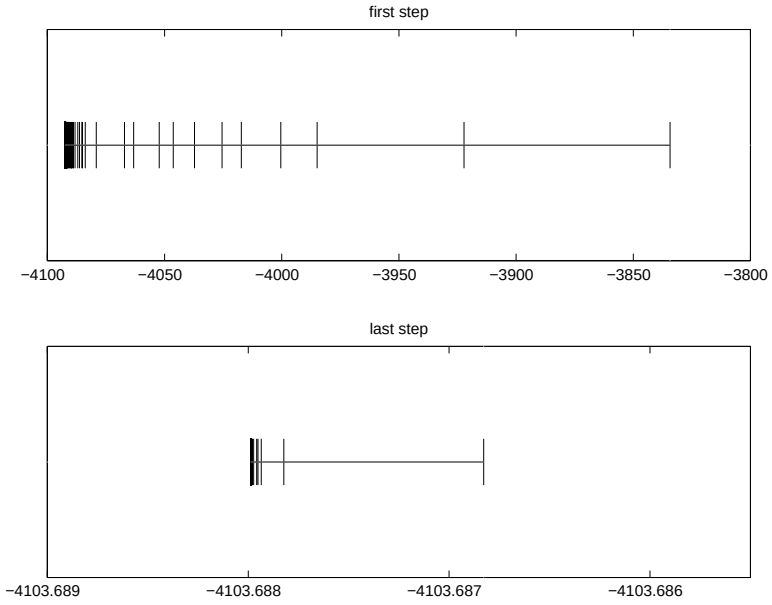


FIG. 5.2. *The eigenvalues of* $\Pi T^H T \Pi - \tau I$, *where* τ *is the approximation of the smallest eigenvalue in the signal subspace. The top graph is for the first step of the Cadzow procedure, and the bottom is for the last. Note that the scale of the horizontal axes is different in the two graphs. The matrix dimension is* 256.

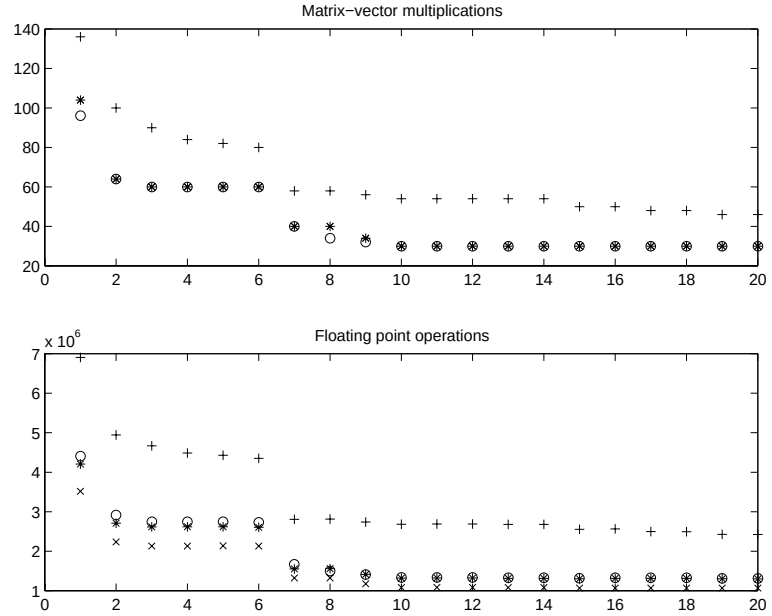


FIG. 5.3. *The work needed for the approximate signal subspace of each matrix in Example* 2 *to converge. The symbols "*$*$*," "*o*," "*$+$*," and "*$\times$*" refer to NCG, NCG-seed, NGM-Circ, and NCG-Ymod, respectively. The matrix dimension is* 256. *Top figure: the number of matrix–vector multiplications* Tx. *Bottom figure: the number of floating point operations.*

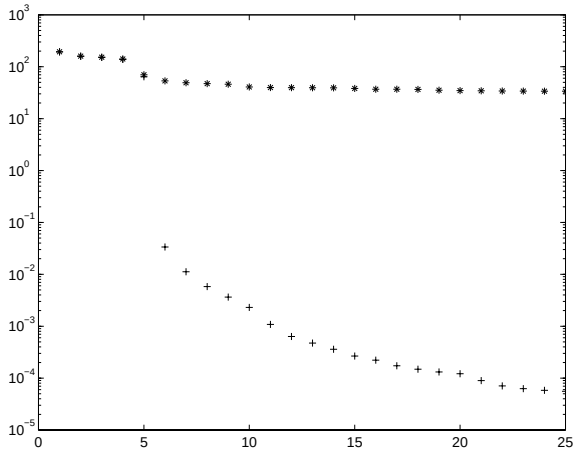


FIG. 5.4. *The* 25 *largest singular values of* $T(1) = T \in \mathbb{C}^{256\times 256}$ ("$*$") *and for* $T(21)$ ("$+$") *for Example* 2.

wants to modify the material properties somewhat and recompute the eigenmodes. For simplicity we take $A(0)$ to be the discrete Laplacian (five-point stencil; see, e.g., [27, p. 50]). The matrix $A(s)$ is a symmetry-preserving finite difference discretization of the self-adjoint, separable elliptic operator

$$L(s)u = ((1 + sx)u_x)_x + ((1 + sy)u_y)_y$$

on an equidistant grid on the unit square; see, e.g., [24, p. 114]. Thus, for small s the matrix $A(s)$ is equal to the discrete Laplacian plus a small symmetric perturbation.

In our experiment we chose $s = 0.15$ and $d = 6$, and the dimension of the matrices was 1156. The six smallest eigenvalues of the matrices are given in Table 5.4.

TABLE 5.4
The six smallest eigenvalues of $A(0)$ *and* $A(0.15)$ *in Example* 3.

A(0)	0.0161	0.0402	0.0402	0.0643	0.0801	0.0801
A(0.15)	0.0173	0.0432	0.0432	0.0690	0.0860	0.0860

We used the CG method to solve the linear systems, and as preconditioner an incomplete Cholesky decomposition [17, p. 535] of $A(0.15) - \tau(1)I$, where $\tau(1)$ is the initial Ritz approximation of the smallest eigenvalue. In the incomplete Cholesky algorithm (the MATLAB function `cholinc`) the drop tolerance was chosen equal to 10^{-1}. The same preconditioner was used throughout.

For comparison we also solved the linear equations without preconditioning. We refer to the different methods as NCG-IC (incomplete Cholesky preconditioning) and NCG (no preconditioning).

The residual tolerance for terminating the CG iterations for NCG and NCG-IC was taken equal to $0.5{\cdot}10^{-6}$ and $0.5{\cdot}10^{-7}$ for two Newton steps. This gave comparable accuracy in the computed eigenvalues for both methods.

In Table 5.5 we give the results from two Newton iterations. Reference approximations of the eigenvalues were computed using the MATLAB function `eig` that implements the QR method. The results show that the preconditioner reduces the

TABLE 5.5

Results from two Newton steps. "Iterations" denotes the total number of CG iterations in a Newton step for six eigenvalues. "Acc. flops" is the total number of flops after the first and second steps. The maximum relative error in the eigenvalues before the first step was $0.2 \cdot 10^{-2}$.

	First		Second	
	NCG-IC	NCG	NCG-IC	NCG
Iterations	81	229	45	137
Max. rel. err.	$0.3 \cdot 10^{-7}$	$0.3 \cdot 10^{-7}$	$0.9 \cdot 10^{-13}$	$0.8 \cdot 10^{-13}$
Acc. flops	$11 \cdot 10^6$	$22 \cdot 10^6$	$18 \cdot 10^6$	$35 \cdot 10^6$

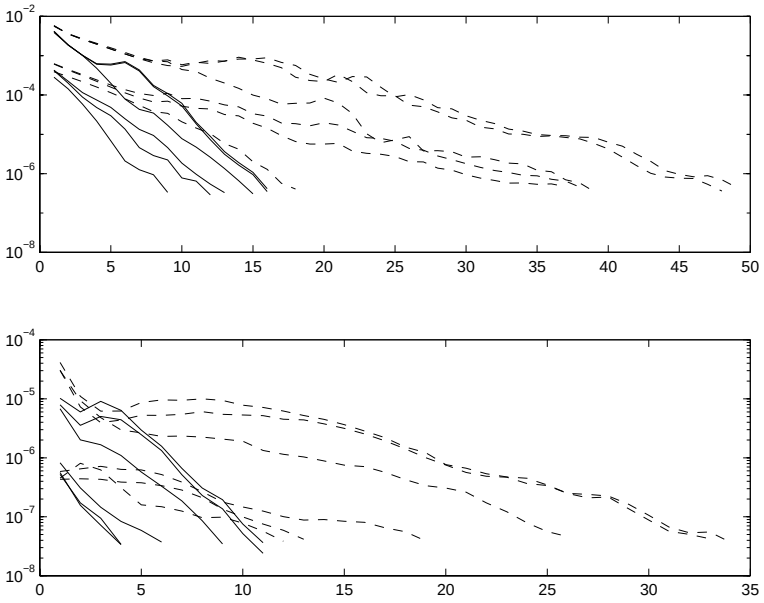


FIG. 5.5. *Convergence history (residuals) for the CG iterations in Example* 3 *for the six smallest eigenvalues in the two Newton iterations (top and bottom). The solid lines are with preconditioning and the dashed without.*

number of iterations and the work significantly. The convergence history is illustrated in Figure 5.5

In this experiment the Grassmann geodesic update (2.5) led to symmetric errors (one positive and one negative) around the double eigenvalues. This did not occur when we replaced (2.5) by the Q factor in the compact QR decomposition of $Y + \Delta$.

As a comparison, we computed the six smallest eigenvalues and corresponding eigenvectors from scratch, using the MATLAB sparse eigenvalue solver `eigs`, which is an implementation of the Arnoldi algorithm [37]. This took $16 \cdot 10^6$ flops, i.e., less than NCG-IC. However, this test matrix of dimension 1156 is probably too small for a method based on iterative solution of linear systems to be more efficient than a method that performs an exact decomposition of the matrix. Note that for very large problems the fill-in created in such a decomposition may be prohibitively high. We also remark that in one run `eigs` failed to detect a double eigenvalue.

6. Final remarks. One attractive feature of the Newton–Grassmann method for subspace tracking and eigenvalue computations is that it is based on a well-known principle, namely, Newton's method. As such it has a solid theoretical underpinning,

which can be used in the actual numerical implementation of the method. A necessary and crucial question is whether this general approach can be used efficiently for solving problems that are of interest in scientific computing. This paper gives a partial answer to this question, and our tests indicate that the Newton–Grassmann method is useful for solving certain structured problems in signal processing (subspace tracking). Concerning eigenvalue computations, more work is needed to ascertain whether this method is really competitive with existing algorithms.

In our experiments with signal subspace problems the linear equations were so well-conditioned that preconditioning did not improve the speed of convergence. It may turn out to be different if the matrices are not so close, and for such cases other preconditioners may have to be developed.

For the sparse matrix case, preconditioning reduced the work, but more research is needed to study incomplete factorization preconditioners in this context. There are different options available concerning the amount of fill-in allowed. One could also try incomplete LU factorization for $A - \tau I$ if care is taken to control the conditioning of the LU factors.

There are related methods, which should be compared to the manifold approach. We have recently [32] investigated the simultaneous Rayleigh quotient iteration based on (2.7), using acceleration [25, 28]. A systematic comparison to the Jacobi–Davidson method should also be performed (see the appendix, where some similarities and differences are pointed out). In future work it is essential to code the methods in Fortran or C to obtain timing comparisons.

Appendix. Comparison to the Jacobi–Davidson method. The Newton–Grassmann is similar in some respects to the Jacobi–Davidson method [35] for computing a set of eigenvalues and eigenvectors [36]. Consider first the case when only one eigenpair is computed, the largest, say. The Jacobi–Davidson algorithm [35, Algorithm 1] involves a parameter m, which is the maximum dimension of a search space that is used before a restart is made. If m is chosen equal to 2, then the search space is $V = (\, y \quad t\,)$, where y is the previous eigenvector approximation, and t is computed from the equation

$$(I - yy^H)(A - \tau I)(I - yy^H)t = -(Ay - \tau y), \quad \tau = y^H Ay. \tag{A.1}$$

Let (θ, v) be the largest eigenpair of $V^H AV$. Then θ is the new eigenvalue approximation, and the new eigenvector approximation is $v_1 y + v_2 t$. Note that since t is orthogonal to y and since $\|v\|_2 = 1$, this is a step along a geodesic curve on the unit sphere from y in the direction t.

Of course, the equation for determining t in the Jacobi–Davidson method is the same as the corresponding equation for the correction in the Newton–Grassmann method. It is now easily seen that in this case the Jacobi–Davidson method is equivalent to a modified Newton–Grassmann method, where, instead of the standard update, a "line search" is performed along the geodesic curve.[5]

For a numerical comparison of the methods, we used a test problem from [35, section 4.1]. The largest eigenvalue of the matrix A of dimension 1000 was computed, where the diagonal elements were $a(i,i) = i$, the super- and subdiagonals were equal to 0.5, as well as the elements $a(1000, 1)$ and $a(1, 1000)$. As starting vector we used $y_1 = (0.01,\, 0.01, \ldots, 0.01,\, 1)^T$. For both methods the linear systems were solved

[5] However, it is not a line search in the strict sense, i.e., such that the residual $\|(A - \theta I)Vv\|_2$ is minimized; see [26, section 11.4].

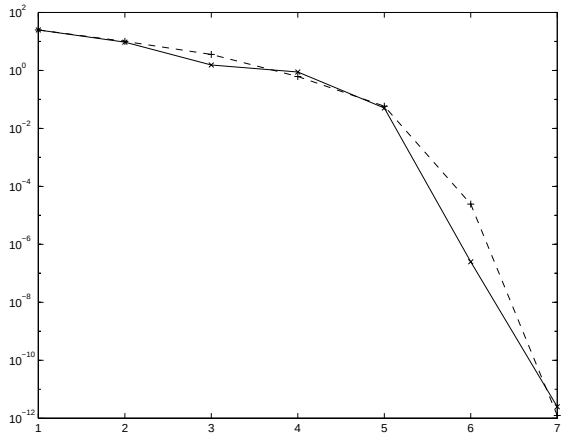


FIG. A.1. *Error in eigenvalue approximation as a function of iteration number for the Jacobi–Davidson method (solid line) and modified Newton–Grassmann (dashed).*

exactly using sparse LU factorization. It turned out that the Newton–Grassmann method converged to an intermediate eigenvalue, while both the Jacobi–Davidson method (without restart) and the modified Newton–Grassmann method (with "line search") converged in seven iterations.[6] The approximation errors for the eigenvalue are illustrated in Figure A.1.

Next consider the case $d > 1$, i.e., several eigenpairs are wanted. Assume that the columns of $Y_{l-1} = (y_1, \ldots, y_{l-1})$ are eigenvectors that have already been computed using the Jacobi–Davidson method. Put $Y_l^{(0)} = (Y_{l-1}\, y_l^{(0)})$, where $y_l^{(0)}$ is a first approximation of the eigenvector y_l, satisfying $y_l^{(0)} \perp Y_{l-1}$. Then new approximations $y_l^{(k)}$ are computed using a corrector equation,

$$\Pi_l^{(k)}(A - \theta_l^{(k)} I)\Pi_l^{(k)} t = -r_l^{(k)}, \tag{A.2}$$

where

$$\Pi_l^{(k)} = I - Y_l^{(k)}(Y_l^{(k)})^H, \qquad r_l^{(k)} = (I - Y_{l-1}Y_{l-1}^H)(A - \theta_l^{(k)} I)\, y_l^{(k)},$$

and $\theta_l^{(k)}$ is a Ritz approximation from a certain search subspace; for details see [35, 36]. Obviously, (A.2) has a structure very similar to (2.4). However, in the Jacobi–Davidson method the eigenvalues and eigenvectors are computed *sequentially*: Previously computed eigenvectors are deflated out of the problem. On the other hand, in the Newton–Grassmann method approximations of the whole set of eigenvectors are used to form the projection matrix in the correction equation, and corrections for all eigenvectors are computed from (2.4); thus new approximate eigenvectors are computed *in parallel*.

Even if there are differences between the two methods, the computational techniques used in connection with the Jacobi–Davidson method can also be applied to the Newton–Grassmann method, since the structures of the correction equations (2.4) and (A.2) are similar.

[6] We can also note that for this example using a search subspace of dimension larger than 2 in the Jacobi–Davidson method did not pay off.

Acknowledgment. Helpful discussions with Valeria Simoncini are acknowledged.

REFERENCES

[1] M. W. Berry, *Large-scale sparse singular value computations*, Internat. J. Supercomput. Appl., 6 (1992), pp. 13–49.
[2] J. Brandts, *Riccati Algorithms for Eigenvalues and Invariant Subspaces*, Technical Report, Preprint 1150, Mathematics Institute, University of Utrecht, Utrecht, 2000.
[3] J. A. Cadzow, *Signal enhancement: A composite property mapping algorithm*, IEEE Trans. Acoust. Speech Signal Process., 36 (1988), pp. 49–62.
[4] R. H. Chan, J. G. Nagy, and R. J. Plemmons, *Displacement preconditioner for Toeplitz least squares iterations*, Electron. Trans. Numer. Anal., 2 (1994), pp. 44–56.
[5] T. F. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.
[6] T. F. Chan and M. K. Ng, *Galerkin Projection Methods for Solving Multiple Linear Systems*, Technical Report 96-31, Department of Mathematics, University of California, Los Angeles, 1996.
[7] F. Chatelin, *Simultaneous Newton's iteration for the eigenproblem*, Comput. Suppl. 5, Springer-Verlag, Vienna, 1984, pp. 67–74.
[8] P. Comon, *Adaptive computation of a few extreme eigenpairs of a positive definite Hermitian matrix*, in Signal Processing IV: Theories and Applications, J. L. Lacoume, A. Chehikian, N. Martin, and J. Malbos, eds., Elsevier Science, Amsterdam, 1988, pp. 647–650.
[9] P. Comon, *An array processing technique using the first principal component*, in SVD and Signal Processing; Algorithms, Applications and Architectures, E. F. Deprettere, ed., Elsevier Science, Amsterdam, 1988, pp. 301–316.
[10] P. Comon and G. H. Golub, *Tracking a few extreme singular values and vectors in signal processing*, Proc. IEEE, 78 (1990), pp. 1327–1343.
[11] P. J. Davis, *Circulant Matrices*, John Wiley, New York, 1979.
[12] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.
[13] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, 1998.
[14] 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.
[15] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353.
[16] J.-L. Fattebert, *A block-Rayleigh quotient iteration with local quadratic convergence*, Electron. Trans. Numer. Anal., 7 (1998), pp. 56–74.
[17] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[18] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[19] D. Y. Hu and L. Reichel, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.
[20] J. Kamm and J. G. Nagy, *A total least squares method for Toeplitz systems of equations*, BIT, 38 (1998), pp. 560–582.
[21] R. Lösche, H. Schwetlick, and G. Timmerman, *A modified block Newton iteration for approximating an invariant subspace of a symmetric matrix*, Linear Algebra Appl., 275–276 (1998), pp. 381–400.
[22] E. Lundström, *Singular Value Computations for Toeplitz Matrices and Subspace Tracking*, Ph.D. thesis, University of Linköping, Linköping, Sweden, 1998.
[23] J. A. Meijerink and H. van der Vorst, *An iterative solution method for linear equation systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.
[24] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, John Wiley, Chichester, 1980.
[25] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[26] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.
[27] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.
[28] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.

[29] A. Sameh and Z. Tong, *The trace minimization method for the symmetric generalized eigenvalue problem*, J. Comput. Appl. Math., 123 (2000), pp. 155–175.
[30] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse, real, unsymmetric matrices*, ACM Trans. Math. Software, 21 (1995), pp. 432–475.
[31] V. Simoncini, *On the numerical solution of $AX - XB = C$*, BIT, 36 (1996), pp. 814–830.
[32] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT, 42 (2002), pp. 159–182.
[33] V. Simoncini and M. Sadkane, *Arnoldi-Riccati method for large eigenvalue problems*, BIT, 36 (1996), pp. 579–594.
[34] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. Van Der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.
[35] 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.
[36] G. L. G. Sleijpen, H. A. van der Vorst, and E. Meijerink, *Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems*, Electron. Trans. Numer. Anal., 7 (1998), pp. 75–89.
[37] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[38] J. M. Speiser, *Signal processing computational needs*, in Advances Algorithms and Architectures for Signal Processing, Vol. 696, J. M. Speiser, ed., SPIE, Bellingham, WA, 1986, pp. 2–6.
[39] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[40] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.
[41] P. Strobach, *Low-rank adaptive filters*, IEEE Trans. Signal Process., 44 (1996), pp. 2932–2947.
[42] C. F. Van Loan, *Computational Framework for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
[43] J. Ward, *Space-time adaptive processing for airborne radar*, in Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing, Vol. 5, IEEE Press, Los Alamitos, CA, 1995, pp. 2809–2812.
[44] G. P. Zientara, L. P. Panych, and F. A. Jolesz, *Dynamically adaptive MRI with encoding by singular value decomposition*, Magnetic Resonance in Medicine, 32 (1994), pp. 268–274.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 840–862
© 2002 Society for Industrial and Applied Mathematics

GENERALIZED POLAR DECOMPOSITIONS FOR THE APPROXIMATION OF THE MATRIX EXPONENTIAL*

A. ZANNA† AND H. Z. MUNTHE-KAAS†

Abstract. In this paper we describe the use of the theory of generalized polar decompositions [H. Munthe-Kaas, G. R. W. Quispel, and A. Zanna, *Found. Comput. Math.*, 1 (2001), pp. 297–324] to approximate a matrix exponential. The algorithms presented have the property that, if $Z \in \mathfrak{g}$, a Lie algebra of matrices, then the approximation for $\exp(Z)$ resides in G, the matrix Lie group of $\mathfrak{g}$. This property is very relevant when solving Lie-group ODEs and is not usually fulfilled by standard approximations to the matrix exponential.

We propose algorithms based on a splitting of Z into matrices having a very simple structure, usually one row and one column (or a few rows and a few columns), whose exponential is computed very cheaply to machine accuracy.

The proposed methods have a complexity of $\mathcal{O}(\kappa n^3)$, with constant κ small, depending on the order and the Lie algebra $\mathfrak{g}$. The algorithms are recommended in cases where it is of fundamental importance that the approximation for the exponential resides in G, and when the order of approximation needed is not too high. We present in detail algorithms up to fourth order.

Key words. matrix exponential, Lie algebra, Lie-group integrator

AMS subject classification. 65M06

PII. S0895479800377551

1. Introduction.

With the recent developments in the theory of Lie-group integration schemes for ordinary differential equations (ODEs) [10], the problem of approximating the matrix exponential has lately received renewed attention. Most Lie-group methods require a number of computations of matrix exponentials from a Lie algebra $\mathfrak{g} \subseteq \mathbb{R}^{n\times n}$ to a Lie group $G \subseteq \mathrm{GL}(n,\mathbb{R})$ that usually constitutes a bottleneck in the numerical implementation of the schemes [5].

The matrix exponentials need to be approximated to the order of the underlying ODE method (hence exact computation is not an issue); however, it is of fundamental importance that such approximations reside in G. In general, this property is not fulfilled by many standard approximations to the exponential function [14] unless the exponential is evaluated exactly.

In some few cases (usually for small dimension) the exponential of a matrix can be evaluated exactly. This happens, for instance, for 3×3 skew-symmetric matrices, whose exponential can be calculated exactly by means of the well-known *Euler–Rodriguez* formula

$$\exp(Z) = I + \frac{\sin\alpha}{\alpha} Z + \frac{1}{2}\left(\frac{\sin(\alpha/2)}{\alpha/2}\right)^2 Z^2, \tag{1.1}$$

where

$$Z = \begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix}, \qquad \alpha = (z_1^2 + z_2^2 + z_3^3)^{1/2}$$

*Received by the editors September 5, 2000; accepted for publication (in revised form) by A. Edelman August 24, 2001; published electronically January 23, 2002.
http://www.siam.org/journals/simax/23-3/37755.html
†Institutt for Informatikk, University of Bergen, Høyteknologisenteret, Thormøhlensgate 55, N-5020 Bergen, Norway (hans@ii.uib.no, anto@ii.uib.no).

[13]. Exact formulas for skew-symmetric matrices and matrices in $\mathfrak{so}(p,q)$ can be derived up to dimension eight making use of the *Cayley–Hamilton* theorem [9] with significant savings with respect to approximation techniques [1, 12]. However, for several reasons the algorithms are not practical for larger dimensions. First, they require high powers of the matrix in question (and each matrix-matrix multiplication amounts to $\mathcal{O}(n^3)$ computations). Second, it is well known that the direct use of the characteristic polynomial, for large-scale matrices, may lead to computational instabilities.

The problem of approximating the exponential of a matrix from a Lie algebra to its corresponding Lie group has been recently considered by [4, 3]. In the first paper, the authors construct the approximation by first splitting the matrix $X \in \mathfrak{g}$ as the sum of bordered matrices. Strang-type splittings of order 2 are considered, so that one could apply a Yoshida technique [24], based on a symmetric composition of a basic scheme whose error locally expands in odd powers of time only, to increase the order. In the second paper, the authors consider techniques based on canonical coordinates of the second kind (CCSK) [22]. To follow that approach, it is necessary to choose a basis of the Lie algebra $\mathfrak{g}$. The choice of the basis plays a significant role in the computational complexity of the algorithms [19], and, by choosing *Chevalley bases* [2] which entail a large number of zero structure constants, it is possible to significantly reduce the cost of the methods.

In this paper we consider the problem of approximating to a given order of accuracy

$$F(t,Z) \approx \exp(tZ) \in G, \qquad Z \in \mathfrak{g}, \tag{1.2}$$

so that $F(t,Z) \in G$, where $\mathfrak{g} \subseteq \mathfrak{gl}(\mathbb{R},n)$ and $G \subseteq \mathrm{GL}(\mathbb{R},n)$. The techniques we introduce consist of a Lie-algebra splitting of the matrix Z by means of an iterated *generalized polar decomposition* induced by an appropriate involutive automorphism $\sigma : G \to G$, as discussed in [16]. We introduce a general technique for approximations of arbitrary high order and discuss practical algorithms of order 2, 3, and 4. For large n, these algorithms are very competitive with standard approximations of the exponential function (for example, diagonal Padé approximants).

The paper is organized as follows. In section 2 we discuss the background theory of the polar decomposition on Lie groups and its symmetric version. Such polar decomposition can be used to induce a splitting in the Lie algebra $\mathfrak{g}$. As long as this splitting is *practical* to compute, together with the exponential of each "split" part, it leads to splitting methods for the approximation of the exponential of practical interest.

In section 3 we use the theory developed in section 2 to derive approximations of the exponential function for some relevant matrix Lie groups as $\mathrm{SO}(\mathbb{R},n)$ and $\mathrm{SL}(\mathbb{R},n)$. Methods of order 2, 3, and 4 are discussed in greater detail, together with their computational complexity. The methods are based on splittings in bordered matrices, whose exact exponentials are very easy to compute.

Section 4 is devoted to some numerical experiments in which we illustrate the results derived in this paper; in section 5 we discuss the relation between our approach and another method for the approximation of the exponential in terms of eigenspace and Schur decompositions; and finally section 6 is devoted to some concluding remarks.

2. Background theory. It is usual in differential geometry to denote Lie-group elements with lowercase letters and Lie-algebra elements with uppercase letters,

whether they represent matrices, vectors, or scalars [7]. We adopt this convention throughout this section.

Let G be a Lie group with Lie algebra $\mathfrak{g}$. We restrict our attention to matrix groups, i.e., to the case when $G \subseteq \mathrm{GL}(\mathbb{R}, n)$.

It is known that, provided $\sigma : G \to G$ is an involutive automorphism of G, every element $z \in G$ sufficiently close to the identity can be decomposed in the product

$$z = xy, \tag{2.1}$$

where $y \in G^\sigma = \{w \in G : \sigma(w) = w\}$, the subgroup of elements of G fixed under σ, and $x \in G_\sigma = \{w : \sigma(w) = w^{-1}\}$ is the subset of antifixed points of σ [11, 16]. The set G_σ has the structure of a *symmetric space* [7] and is closed under the product

$$x_1 \cdot x_2 = x_1 x_2^{-1} x_1,$$

as can be easily verified by application of σ to the right-hand side of the above relation. The decomposition (2.1) is called the *generalized polar decomposition* of z in analogy with the case of real matrices with the choice of automorphism $\sigma(z) = z^{-\mathrm{T}}$.

Next set $z = \exp(tZ)$ with $Z \in \mathfrak{g}$. The automorphism σ induces an involutive automorphism $\mathrm{d}\sigma$ on $\mathfrak{g}$ in a natural manner,

$$\mathrm{d}\sigma(Z) = \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} \sigma(\exp(tZ)),$$

and it defines a splitting of the algebra $\mathfrak{g}$ into the sum of two linear spaces,

$$\mathfrak{g} = \mathfrak{p} \oplus \mathfrak{k}, \tag{2.2}$$

where $\mathfrak{k} = \{Z \in \mathfrak{g} : \mathrm{d}\sigma(Z) = Z\}$ is a subalgebra of $\mathfrak{g}$, while $\mathfrak{p} = \{Z \in \mathfrak{g} : \mathrm{d}\sigma(Z) = -Z\}$ has the structure of a Lie-triple system, a set closed under the double commutator,

$$A, B, C \in \mathfrak{p} \Longrightarrow [A, [B, C]] \in \mathfrak{p}.$$

To keep our presentation relevant to the argument matter of this paper, we refer the reader to [16, 15] and the references therein for a more extensive treatment of such decompositions. However, it is of fundamental importance to note that the sets $\mathfrak{k}$ and $\mathfrak{p}$ possess the following properties:

$$\begin{aligned} [\mathfrak{k}, \mathfrak{k}] &\subseteq \mathfrak{k}, \\ [\mathfrak{k}, \mathfrak{p}] &\subseteq \mathfrak{p}, \\ [\mathfrak{p}, \mathfrak{p}] &\subseteq \mathfrak{k}. \end{aligned} \tag{2.3}$$

We denote by $\Pi_\mathfrak{p} : \mathfrak{g} \to \mathfrak{p}$ the canonical projection onto the subspace $\mathfrak{p}$ and by $\Pi_\mathfrak{k} : \mathfrak{g} \to \mathfrak{k}$ the projection onto $\mathfrak{k}$. Then,

$$Z = \Pi_\mathfrak{p} Z + \Pi_\mathfrak{k} Z = P + K,$$

where

$$\Pi_\mathfrak{p} Z = \frac{1}{2}(Z - \mathrm{d}\sigma(Z)), \qquad \Pi_\mathfrak{k} Z = \frac{1}{2}(Z + \mathrm{d}\sigma(Z)).$$

Assume that x and y in (2.1) are of the form $x = \exp(X(t))$ and $y = \exp(Y(t))$. Then $X(t) \in \mathfrak{p}$, $Y(t) \in \mathfrak{k}$ and they can be expanded in series

$$X(t) = \sum_{i=1}^{\infty} X_i t^i, \qquad Y(t) = \sum_{i=1}^{\infty} Y_i t^i,$$

where the X_i and Y_i can be explicitly calculated by means of the following recurrence relations:

$$\begin{aligned} X_1 &= P, \\ (i+1)X_{i+1} &= -[X_i, K] + \sum_{\substack{\ell \geq 1 \\ 2\ell \leq i}} c_{2\ell} \sum_{\substack{\ell_1, \ldots, \ell_{2\ell} > 0 \\ \ell_1 + \cdots + \ell_{2\ell} = i}} [X_{\ell_1}, [X_{\ell_2}, \ldots, [X_{\ell_{2\ell}}, P]]], \quad i = 1, 2, \ldots, \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} Y_1 &= K, \\ Y_{2i} &= O, \qquad i = 0, 1, 2, \ldots, \\ 2(2i+1)Y_{2i+1} &= -2\sum_{q=1}^{i} \sum_{\substack{k \geq 1 \\ k \leq q}} \frac{1}{(2k+1)!} \sum_{\substack{k_1, \ldots, k_{2k} > 0 \\ k_1 + \cdots + k_{2k} = 2q}} [Y_{k_1}, \ldots, [Y_{k_{2k}}, Y_{2(i-q)+1}] \ldots] \\ &\quad - \sum_{m=1}^{i} \frac{2(i-m)+1}{(2m)!} \mathrm{ad}_Z^{2m} Y_{2(i-m)+1} \\ &\quad - \sum_{q=0}^{2(i-1)} \sum_{j=0}^{2(i-1)-q} \frac{(-1)^{2i-q-j-1}(j+1)}{(2i-q-j-1)!} \mathrm{ad}_Z^{2i-j-q-1} \\ &\qquad \sum_{\substack{k \geq 1 \\ k \leq q+1}} \frac{1}{(k+1)!} \sum_{\substack{j_1, \ldots, j_k > 0 \\ j_1 + \cdots + j_k = q+1}} [Y_{j_1}, \ldots, [Y_{j_k}, Y_{j+1}] \ldots] \\ &\quad - \sum_{\substack{\ell \geq 1 \\ \ell \leq i}} \frac{1}{(2\ell)!} \sum_{\substack{\ell_1, \ldots, \ell_{2\ell} > 0 \\ \ell_1 + \cdots + \ell_{2l} = 2i}} [Y_{\ell_1}, \ldots, [Y_{\ell_{2\ell}}, P - K] \ldots] \end{aligned} \tag{2.5}$$

(see [25]). Note that $Y(t)$ expands in odd powers of t only. The first terms in the expansions of $X(t)$ and $Y(t)$ are

$$\begin{aligned} X = {} & Pt - \tfrac{1}{2}[P, K]t^2 - \tfrac{1}{6}[K, [P, K]]t^3 \\ & + \Big(\tfrac{1}{24}[P, [P, [P, K]]] - \tfrac{1}{24}[K, [K, [P, K]]] \Big) t^4 \\ & + \Big(\tfrac{7}{360}[K, [P, [P, [P, K]]]] - \tfrac{1}{120}[K, [K, [K, [P, K]]]] - \tfrac{1}{180}[[P, K], [P, [P, K]]] \Big) t^5 \\ & + \mathcal{O}\big(t^6\big), \end{aligned}$$

$$\begin{aligned} Y = {} & Kt - \tfrac{1}{12}[P, [P, K]]t^3 + \Big(\tfrac{1}{120}[P, [P, [P, [P, K]]]] \\ & + \tfrac{1}{720}[K, [K, [P, [P, K]]]] - \tfrac{1}{240}[[P, K], [K, [P, K]]] \Big) t^5 + \mathcal{O}\big(t^7\big). \end{aligned} \tag{2.6}$$

We also consider a symmetric-type generalized polar decomposition,

$$z = xyx, \qquad z = \exp(tZ), \quad x = \exp(X(t)), \quad y = \exp(Y(t)), \tag{2.7}$$

where, as above, $X(t) \in \mathfrak{p}$ and $Y(t) \in \mathfrak{k}$. To compute $X(t)$, we apply σ to both sides of (2.7) to obtain

$$\sigma(z) = \exp(-X(t)) \exp(Y(t)) \exp(-X(t)). \tag{2.8}$$

Isolating the y term in (2.8) and (2.7) and equating the result, we obtain

$$\exp(tZ) = \exp(2X(t)) \exp(tW) \exp(2X(t)), \qquad W = \mathrm{d}\sigma(Z). \tag{2.9}$$

This leads to a differential equation for X which is very similar to the one obeyed by Y in (2.5) [25]. Using the recursions in [25] we obtain recursions for $X(t)$ and $Y(t)$. The first terms are given as

$$\begin{aligned} X(t) = {} & \tfrac{1}{2}Pt + \tfrac{1}{24}[K,[P,K]]t^3 - \Big(\tfrac{1}{1440}[K,[P,[P,[P,K]]]] \\ & + \tfrac{1}{240}[K,[K,[P,K]]]] + \tfrac{1}{360}[[P,K],[P,[P,K]]]\Big)t^5 + \cdots, \end{aligned} \tag{2.10}$$

$$\begin{aligned} Y(t) = {} & Kt + \tfrac{1}{24}[P,[P,K]]t^3 + \Big(\tfrac{1}{1920}[P,[P,[P,[P,K]]]] \\ & - \tfrac{13}{1440}[K,[K,[P,[P,K]]]] - \tfrac{1}{240}[[P,K],[K,[P,K]]]\Big)t^5 + \cdots \end{aligned} \tag{2.11}$$

and both $X(t)$ and $Y(t)$ expand in odd powers of t only.

3. Generalized polar decomposition and its symmetric version for the approximation of the exponential. Assume now that we wish to approximate $\exp(tZ)$ for some $Z \in \mathfrak{g}$, and that σ_1 is an involutive automorphism so that the exponential of terms in $\mathfrak{p}_1 = \{X \in \mathfrak{g} : \mathrm{d}\sigma_1 X = -X\}$ as well as analytic functions of $\mathrm{ad}_P = [P, \cdot]$ are easy to compute. Then $\mathfrak{g} = \mathfrak{p}_1 \oplus \mathfrak{k}_1$, and we can approximate

$$\exp(tZ) \approx \exp(X^{[1]}(t)) \exp(Y^{[1]}(t)), \tag{3.1}$$

where $X^{[1]}$ and $Y^{[1]}$ obey the order conditions (2.4)–(2.6) to suitable order.

Alternatively, we can approximate

$$\exp(tZ) \approx \exp(X^{[1]}(t)) \exp(Y^{[1]}(t)) \exp(X^{[1]}(t)), \tag{3.2}$$

where $X^{[1]}$ and $Y^{[1]}$ now obey the order conditions (2.10)–(2.11) to given accuracy.

The same mechanism can be applied to split $\mathfrak{k}_1$ in $\mathfrak{p}_2 \oplus \mathfrak{k}_2$ by means of a suitable automorphism σ_2. The procedure can be iterated and, provided that the exponential of $\mathfrak{k}_m$ is easy to compute, we have an algorithm to approximate $\exp(tZ)$ to a given order of accuracy. In this circumstance, (3.1) will read

$$\exp(tZ) \approx F(t,Z) = \exp(X^{[1]}(t)) \cdots \exp(X^{[m]}(t)) \exp(Y^{[m]}(t)), \tag{3.3}$$

while the analogue of (3.2) is

$$\begin{aligned} \exp(tZ) & \approx F(t,Z) \\ & = \exp(X^{[1]}(t)) \cdots \exp(X^{[m]}(t)) \exp(Y^{[m]}(t)) \exp(X^{[m]}(t)) \cdots \exp(X^{[1]}(t)), \end{aligned} \tag{3.4}$$

both corresponding to the algebra splitting

$$\mathfrak{g} = \mathfrak{p}_1 \oplus \cdots \oplus \mathfrak{p}_m \oplus \mathfrak{k}_m. \tag{3.5}$$

3.1. On the choice of the automorphisms σ_i. In what follows, we will consider automorphisms σ of the form

$$\sigma(z) = SzS, \qquad z \in G, \tag{3.6}$$

where S is an idempotent matrix, i.e., $S^2 = I$ [18]. Clearly,

$$\mathrm{d}\sigma(Z) = SZS,$$

and, for simplicity, we will abuse notation by writing σZ in place of $\mathrm{d}\sigma Z$, given that all our computations take place in the space of matrices.

Since $S^2 = I$, all the eigenvalues of S are either $+1$ or -1. Thus, powers of matrices $P = \Pi_{\mathfrak{p}}(Z)$ as well as powers of ad_P are easy to evaluate by means of the $(+1)$- and (-1)-eigenspace of S [18].

Note that automorphisms of the type (3.6) are defined a priori, with respect to a fixed basis chosen independently of the data in Z. In section 5 we shall discuss automorphisms based on approximate eigenspace decompositions of the matrix Z, a case in which the splitting depends dynamically on the given matrix.

3.2. Automorphisms that lead to bordered matrix splittings. Let $Z \in \mathfrak{gl}(n, \mathbb{R})$ be an $n \times n$ matrix and consider the automorphism

$$\sigma_1 Z = S_1 Z S_1,$$

where S_1 is the idempotent matrix

$$S_1 = \left(\begin{array}{c|ccc} -1 & 0 & \cdots & 0 \\ \hline 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{array}\right).$$

It is easy to verify that

$$\Pi_{\mathfrak{p}_1} Z = \frac{1}{2}(Z - S_1 Z S_1) = \left(\begin{array}{c|ccc} 0 & z_{1,2} & \cdots & z_{1,n} \\ \hline z_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & 0 & \cdots & 0 \end{array}\right), \tag{3.7}$$

while

$$\Pi_{\mathfrak{k}_1} Z = \frac{1}{2}(Z + S_1 Z S_1) = \left(\begin{array}{c|ccc} z_{1,1} & 0 & \cdots & 0 \\ \hline 0 & z_{2,2} & \cdots & z_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & z_{n,2} & \cdots & z_{n,n} \end{array}\right). \tag{3.8}$$

In general, assume that, at the jth step, the space $\mathfrak{k}_{j-1}$ consists of matrices of the form

$$W = \left(\begin{array}{ccc|ccc} w_{1,1} & & O & & & \\ & \ddots & & & O & \\ O & & w_{j-1,j-1} & & & \\ \hline & & & w_{j,j} & \cdots & w_{j,n} \\ & O & & \vdots & \ddots & \vdots \\ & & & w_{n,j} & \cdots & w_{n,n} \end{array}\right). \tag{3.9}$$

Then, the obvious choice is

$$S_j = \left(\begin{array}{c|c} I_{j-1} & O \\ \hline O & \tilde{S}_j \end{array}\right), \qquad \tilde{S}_j = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \tag{3.10}$$

where I_{j-1} denotes the $(j-1)\times(j-1)$ identity matrix and $\tilde{S}_j$ is an $(n-j+1)\times(n-j+1)$ block, so that the subspace $\mathfrak{p}_j$ consists of matrices of the form

$$\Pi_{\mathfrak{p}_j} W = \left(\begin{array}{c|c} O_{j-1} & O \\ \hline O & \tilde{P}_j \end{array}\right), \qquad \tilde{P}_j = \left(\begin{array}{c|ccc} 0 & w_{j,j+1} & \cdots & w_{j,n} \\ \hline w_{j+1,j} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,j} & 0 & \cdots & 0 \end{array}\right), \tag{3.11}$$

where O_{j-1} denotes the $(j-1)\times(j-1)$ zero matrix.

Exponentials of matrices of the form (3.11) are very easy to compute: in effect,

$$\exp\left(\begin{array}{c|c} O_{j-1} & O \\ \hline O & \tilde{P}_j \end{array}\right) = \left(\begin{array}{c|c} I_{j-1} & O \\ \hline O & \exp(\tilde{P}_j) \end{array}\right),$$

where $\exp(\tilde{P}_j)$ can be computed *exactly* with a formula analogous to the Euler–Rodriguez formula (1.1): denote $\boldsymbol{a}_j = [w_{j+1,j}, \ldots, w_{n,j}]^{\mathrm{T}}$ and $\boldsymbol{b}_j = [w_{j,j+1}, \ldots w_{j,n}]^{\mathrm{T}}$. Then,

$$\exp(\tilde{P}_j) = \begin{cases} I_{n-j+1} + \frac{\sinh\alpha_j}{\alpha_j}\tilde{P}_j + \frac{1}{2}\left(\frac{\sinh(\alpha_j/2)}{\alpha_j/2}\right)^2 \tilde{P}_j^2 & \text{if } \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{b}_j > 0, \alpha_j = \sqrt{\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{b}_j}, \\ I_{n-j+1} + \tilde{P}_j + \frac{1}{2}\tilde{P}_j^2 & \text{if } \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{b}_j = 0, \\ I_{n-j+1} + \frac{\sin\alpha_j}{\alpha_j}\tilde{P}_j + \frac{1}{2}\left(\frac{\sin(\alpha_j/2)}{\alpha_j/2}\right)^2 \tilde{P}_j^2 & \text{if } \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{b}_j < 0, \alpha_j = \sqrt{-\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{b}_j}. \end{cases} \tag{3.12}$$

Note that

$$\tilde{P}_j^2 = \left(\begin{array}{c|c} \alpha_j^2 & \mathbf{0}^{\mathrm{T}} \\ \hline \mathbf{0} & \boldsymbol{a}_j\boldsymbol{b}_j^{\mathrm{T}} \end{array}\right).$$

Another alternative for the exact exponential of $\tilde{P}_j$ is the one proposed in [4]:

$$\exp(\tilde{P}_j) = I_{n-j+1} + [\boldsymbol{k}, \boldsymbol{e}_1]\varphi(D)\begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \boldsymbol{l}^{\mathrm{T}} \end{bmatrix}, \tag{3.13}$$

where

$$\boldsymbol{k} = \begin{bmatrix} 0 \\ \boldsymbol{a}_j \end{bmatrix}, \quad \boldsymbol{l} = \begin{bmatrix} 0 \\ \boldsymbol{b}_j \end{bmatrix}, \quad D = \begin{pmatrix} 0 & 1 \\ \boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{a}_j & 0 \end{pmatrix},$$

$\boldsymbol{e}_1$ is the vector $[1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^{n-j+1}$ and finally $\varphi(z) = (\mathrm{e}^z - 1)/z$. The latter formula (3.13), as we shall see in what follows, leads to significant savings in the computation and assembly of the exponentials.

Moreover, given that

$$\Pi_{\mathfrak{k}_j} W = \left(\begin{array}{c|c} D_{j-1} & O \\ \hline O & \tilde{K}_j \end{array}\right), \qquad D_{j-1} = \operatorname{diag}(w_{1,1}, \ldots, w_{j-1,j-1}), \tag{3.14}$$

where

$$\tilde{K}_j = \left(\begin{array}{c|c} w_{j,j} & \mathbf{0}^{\mathrm{T}} \\ \hline \mathbf{0} & \bar{K}_j \end{array}\right),$$

then

$$[\tilde{P}_j, \tilde{K}_j] = \left(\begin{array}{c|c} 0 & \boldsymbol{b}_j^{\mathrm{T}} \bar{K}_j - w_{j,j} \boldsymbol{b}_j^{\mathrm{T}} \\ \hline w_{j,j} \boldsymbol{a}_j - \bar{K}_j \boldsymbol{a}_j & O \end{array}\right). \tag{3.15}$$

Next, if $Z \in \mathfrak{g}$, to obtain an approximation of the exponential in G by these automorphisms we shall require that the σ_i's, defined by the above matrices S_i, map $\mathfrak{g}$ into $\mathfrak{g}$. Clearly, this is the case for

- $\mathfrak{so}(n, \mathbb{R})$, since $\sigma_i Z = S_i Z S_i = \mathrm{Ad}_{S_i} Z$ is a map from $\mathfrak{so}(n) \to \mathfrak{so}(n)$, given that each S_i is an orthogonal matrix;
- $\mathfrak{sl}(n, \mathbb{R})$, since σ_i leaves the diagonal elements of Z (hence its trace) unchanged;
- quadratic Lie algebras $\mathfrak{g} = \{Z : ZJ + JZ^{\mathrm{T}} = O, J \text{ nonsingular}\}$ provided that J and the S_i's commute. This is, for instance, the case when J is diagonal; hence our formulas are valid for $\mathfrak{so}(p, q)$, $p+q = n$, but not for the symplectic algebra $\mathfrak{sp}(n, \mathbb{R})$. In the latter situation, we consider different choices for the automorphisms σ_i, discussed at a greater length in [18].

3.3. Splittings of order 2 to 4, their implementation and complexity. In this section we describe in more detail the algorithms, the implementation, and the complexity of the splittings induced by the automorphisms described above. Note that the complexity counts refer to the computation of the splitting alone and depends on the desired order of approximation, and further work is required for the assembly of the approximation to the exponential. The total cost of some schemes (splitting and assembly of the approximated exponential) is therefore presented in section 4.

The cases of a polar-type representation, $z = xy$, or a symmetric polar-type representation, $z = xyx$, are discussed separately.

ALGORITHM 1 (polar-type splitting, order 2). *Based on the iterated generalized polar decomposition* (3.3). Note that the $\Pi_{\mathfrak{p}_j}$ and $\Pi_{\mathfrak{k}_j}$ projections need not be stored in separate matrices but can be stored in places of the rows and columns of the matrix Z. We truncate the expansions (2.6) to order 2, and hence at each step only the $\mathfrak{p}_j$-part needs correction. Taking in mind (3.3), the matrices $X^{[j]}$ are low-rank matrices with nonzero entries only on the jth row, column $j+1$ to n, and jth column, row $j+1$ to n, for $j = 1, \ldots, n-2$, which are stored in place of the corresponding Z entries. The matrix $Y^{[n-1]}$ is diagonal and is stored in the diagonal entries of Z.

% Purpose: second-order approximation of the splitting (3.3)
% In: $n \times n$ matrix Z
% Out: Z overwritten with the nonzero elements of $X^{[i]}$ and $Y^{[m]}$ as:
% $Z(i+1:n, i) = X^{[i]}(i+1:n, i), \quad Z(i, i+1:n) = X^{[i]}(i, i+1:n),$
% $\operatorname{diag}(Z) = \operatorname{diag}(Y^{[m]})$

```
%
for j = 1 : n − 1
   a_j := Z(j + 1 : n, j),
   b_j := Z(j, j + 1 : n)^T,
   K̄_j := Z(j + 1 : n, j + 1 : n),
   c_j := z_{j,j} a_j − K̄_j a_j,
   d_j := −z_{j,j} b_j + K̄_j^T b_j,
   Z(j + 1 : n, j) := a_j − (1/2) c_j,
   Z(j, j + 1 : n) := (b_j − (1/2) d_j)^T
end
```

The computation of the splitting requires at each step two matrix-vector multiplications, each amounting to $\mathcal{O}\big(2(n-j+1)^2\big)$ floating point operations (we count both multiplications and additions), as well as two vector updates, which are $\mathcal{O}(n-j+1)$ operations. Hence, for large n, the cost of computing the splitting is of the order

- $\frac{4}{3}n^3$ for $\mathfrak{so}(p,q)$, $p+q=n$ and $\mathfrak{sl}(n)$,
- $\frac{2}{3}n^3$ for $\mathfrak{so}(n)$, taking into account that $\boldsymbol{b}_j = -\boldsymbol{a}_j$.

Note that for both $\mathfrak{so}(p,q)$ and $\mathfrak{so}(n)$ the matrix $Y^{[n-1]}$ is the zero matrix.

ALGORITHM 2 (symmetric polar-type splitting, order 2). *Based on the iterated generalized polar decomposition* (3.4). We truncate the expansions (2.10)–(2.11) to order 2. The storing of the entries is as above.

```
% Purpose: second-order approximation of the splitting (3.4)
% In: n × n matrix Z
% Out: Z overwritten with the nonzero elements of X^[i] and Y^[m] as:
%    Z(i + 1 : n, i) = X^[i](i + 1 : n, i),    Z(i, i + 1 : n) = X^[i](i, i + 1 : n),
%    diag(Z) = diag(Y^[m])
%
% Computation of the splitting
for j = 1 : n − 1
   a_j := Z(j + 1 : n, j),
   b_j := Z(j, j + 1 : n)^T,
   Z(j + 1 : n, j) := (1/2) a_j,
   Z(j, j + 1 : n) := (1/2) b_j^T
end
```

This splitting costs only

- $n(n-1)$ for $\mathfrak{so}(p,q)$, $p+q=n$ and $\mathfrak{sl}(n)$,
- $\frac{n(n-1)}{2}$ for $\mathfrak{so}(n)$, because of skew-symmetry.

ALGORITHM 3 (polar-type splitting, order 3). We truncate (2.6)–(2.7) to include $\mathcal{O}\big(t^3\big)$ terms. Note that the term $[K,[P,K]]$ is of the form (3.15). We need to include also the term of the form $[P,[P,K]]$. We observe that

$$[\tilde{P}_j,[\tilde{P}_j,\tilde{K}_j]] = \left(\begin{array}{c|c} 2\boldsymbol{b}_j^{\mathrm{T}}(z_{j,j}I-\bar{K}_j)\boldsymbol{a}_j & \mathbf{0}^{\mathrm{T}} \\ \hline \mathbf{0} & -\boldsymbol{a}_j\boldsymbol{b}_j^{\mathrm{T}}(z_{j,j}I-\bar{K}_j)-(z_{j,j}I-\bar{K}_j)\boldsymbol{a}_j\boldsymbol{b}_j^{\mathrm{T}} \end{array}\right). \tag{3.16}$$

```
% Purpose: third-order approximation of the splitting (3.3)
% In: n × n matrix Z
% Out: Z overwritten with the nonzero elements of X^[i] and Y^[m] as:
```

% $Z(i+1:n,i) = X^{[i]}(i+1:n,i), \quad Z(i,i+1:n) = X^{[i]}(i,i+1:n),$
% $\operatorname{diag}(Z) = \operatorname{diag}(Y^{[m]})$
%
% Computation of the splitting
for $j = 1:n-1$
$\boldsymbol{a}_j := Z(j+1:n,j)$,
$\boldsymbol{b}_j := Z(j,j+1:n)^{\mathrm{T}}$,
$\bar{K}_j := Z(j+1:n,j+1:n)$,
$\boldsymbol{c}_j := z_{j,j}\boldsymbol{a}_j - \bar{K}_j\boldsymbol{a}_j$,
$\boldsymbol{d}_j := -z_{j,j}\boldsymbol{b}_j + \bar{K}_j^{\mathrm{T}}\boldsymbol{b}_j$,
$Z(j+1:n,j) := \boldsymbol{a}_j - \frac{1}{2}\boldsymbol{c}_j + \frac{1}{6}(z_{j,j}I - \bar{K}_i)\boldsymbol{c}_j$,
$Z(j,j+1:n) := (\boldsymbol{b}_j - \frac{1}{2}\boldsymbol{d}_j - \frac{1}{6}(z_{j,j}I - \bar{K}_j)\boldsymbol{d}_j)^{\mathrm{T}}$,
$Z(j+1:n,j+1:n) := Z(j+1:n,j+1:n) + \frac{1}{12}(-\boldsymbol{a}_j\boldsymbol{d}_j^{\mathrm{T}} + \boldsymbol{c}_j\boldsymbol{b}_j^{\mathrm{T}})$,
$Z(j,j) := Z(j,j) - \frac{1}{6}\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{c}_j$
end

Analyzing the computations involved, we find that the most costly part is constituted by the matrix-vector products in the computations in $\boldsymbol{c}_j, \boldsymbol{d}_j$, $Z(j+1:n,j)$, $Z(j,j+1:n)$ and vector-vector products in the update of $Z(j+1:n,j+1:n)$ and $z(j,j)$. The computation of $\boldsymbol{c}_j, \boldsymbol{d}_j, Z(j+1:n,j)$, and $Z(j,j+1:n)$ amounts to $\frac{8}{3}n^3$ in the whole process. For the update of $Z(j+1:n,j+1:n)$, we need to compute two vector-vector products ($\mathcal{O}\big((n-j+1)^2\big)$ each) plus $3(n-j+1)^2$ operations to update the elements of the matrix. Thus, the whole cost of updating the matrix $Z(j+1:n,j+1:n)$ is $\frac{5}{3}n^3$. The update of $z_{j,j}$ requires $2(n-j+1)^2$ operations per step, which give a $\frac{2}{3}n^3$ contribution to the total cost of the splitting.

In summary, the total cost of the splitting is

- $5n^3$ for $\mathfrak{so}(p,q)$ and $\mathfrak{sl}(n)$,
- for $\mathfrak{so}(n)$, note that $\boldsymbol{d}_j$ need not be calculated as well as $z_{j,j} = 0$. Similarly, we take into account that $\boldsymbol{b}_j = -\boldsymbol{a}_j$ and that only half of the elements of $Z(j+1:n,j+1:n)$ need to be updated. The total amounts to $2\frac{1}{2}n^3$ operations.

It is easy to modify the splitting above to obtain order 4. Note that

$$[\tilde{P}_j,[\tilde{P}_j,[\tilde{P}_j,\tilde{K}_j]]] = \boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{a}_j[\tilde{P}_j,\tilde{K}_j] + 3(\boldsymbol{b}_j^{\mathrm{T}}(z_{j,j}I - \bar{K}_j)\boldsymbol{a}_j)\tilde{S}_j\tilde{P}_j, \tag{3.17}$$

which requires the computation of the scalar $\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{a}_j$ only, costing $2/3n^3$ operations in the whole process. However, all the other powers $\operatorname{ad}_{\tilde{P}_j}^i \tilde{K}_j$ for $i = 4,5,\ldots$ require no further computation. The next term, $[\tilde{K}_j,[\tilde{K}_j,[\tilde{P}_j,\tilde{K}_j]]]$, can be computed with just two (one) extra matrix-vector computations for $\mathfrak{sl}(n)$ (resp., $\mathfrak{so}(n)$), which contribute $\frac{4}{3}n^3$ (resp., $\frac{2}{3}n^3$) to the cost of the splitting, so that the splitting of order 4 costs a total of $7n^3$ operations for $\mathfrak{sl}(n)$ (resp., $4n^3$ for $\mathfrak{so}(n)$).

ALGORITHM 4 (symmetric polar-type splitting, order 4). We truncate (2.10)–(2.11) to include $\mathcal{O}\big(t^3\big)$ terms. Also in this case, the term $[K,[P,K]]$ is of the form (3.15), while the term $[P,[P,K]]$ is computed according to (3.16).

% Purpose: fourth-order approximation of the splitting (3.4)
% In: $n \times n$ matrix Z
% Out: Z overwritten with the nonzero elements of $X^{[i]}$ and $Y^{[m]}$ as:
% $Z(i+1:n,i) = X^{[i]}(i+1:n,i), \quad Z(i,i+1:n) = X^{[i]}(i,i+1:n),$

```
%   diag(Z) = diag(Y^[m])
%
for j = 1 : n − 1
   a_j := Z(j + 1 : n, j),
   b_j := Z(j, j + 1 : n)^T,
   K̄_j := Z(j + 1 : n, j + 1 : n),
   c_j := (z_{j,j} I − K̄_j)^2 a_j,
   d_j := (z_{j,j} I − K̄_j^T)^2 b_j,
   Z(j + 1 : n, j) := (1/2) a_j − (1/24) c_j,
   Z(j, j + 1 : n) := ((1/2) b_j − (1/24) d_j)^T,
   Z(j + 1 : n, j + 1 : n) := Z(j + 1 : n, j + 1 : n) − (1/24)(a_j b_j^T (z_{j,j} I − K̄_j),
                                  + (z_{j,j} I − K̄_j) a_j b_j^T),
   z(j, j) := z(j, j) + (1/12) b_j^T (z_{j,j} I − K̄_j) a_j
end
```

We need to compute a total of four matrix-vector products, yielding $\frac{8}{3}n^3$ operations. The update of the block $Z(j+1:n, j+1:n)$ costs $\frac{5}{3}n^3$ operations, while the update of $z(j,j)$ costs $\frac{2}{3}n^3$ operations, for a total of

- $5n^3$ operations for $\mathfrak{sl}(n)$ and $\mathfrak{so}(p+q)$,
- $2\frac{1}{2}n^3$ operations for $\mathfrak{so}(n)$.

3.4. On higher-order splittings. The costs of implementing splittings following (3.3) or (3.4) depend on the type of commutation involved: commutators of the form $[P, K]$ and $[P_1, P_2]$, $P, P_1, P_2 \in \mathfrak{p}$, $K \in \mathfrak{k}$, contribute as an $\mathcal{O}(n^3)$ term to the total complexity of the splitting; however, commutators of the form $[K_1, K_2]$ for $K_1, K_2 \in \mathfrak{k}$ can easily contribute an $\mathcal{O}(n^4)$ to the total complexity of the splittings if the special structure of the terms involved is not taken into consideration. If carefully implemented, these terms can also be computed with only matrix-vector and vector-vector products, contributing $\mathcal{O}(n^3)$ operations to the total cost of the splitting. For example, let us consider the term $[\tilde{K}_j, [\tilde{K}_j, [\tilde{P}_j, [\tilde{P}_j, \tilde{K}_j]]]]$, which appears in the $\mathcal{O}(t^5)$ contribution in the expansion of the Y part, for both the polar-type and symmetric polar-type splittings. One has

$$
\begin{aligned}
[\tilde{K}_j, [\tilde{K}_j, [\tilde{P}_j, [\tilde{P}_j, \tilde{K}_j]]]] &= \left(\begin{array}{c|c} 0 & \mathbf{0}^{\mathrm{T}} \\ \hline \mathbf{0} & \bar{Q}_j \end{array}\right), \\
\bar{Q}_j &= -\Big((\bar{K}_j(\bar{K}_j \boldsymbol{a}_j))(\boldsymbol{b}_j^{\mathrm{T}} \Delta_j)\Big) - \Big((\bar{K}_j(\bar{K}_j(\Delta_j \boldsymbol{a}_j)))\boldsymbol{b}_j^{\mathrm{T}}\Big) \\
&\quad +2\Big(((\bar{K}_j \boldsymbol{a}_j)((\boldsymbol{b}_j^{\mathrm{T}} \Delta_j)\bar{K}_j)) + ((\bar{K}_j(\Delta_j \boldsymbol{a}_j))(\boldsymbol{b}_j^{\mathrm{T}} \bar{K}_j))\Big) \\
&\quad -\Big(\boldsymbol{a}_j(((\boldsymbol{b}_j^{\mathrm{T}} \Delta_j)\bar{K}_j)\bar{K}_j)\Big) - \Big((\Delta_j \boldsymbol{a}_j)(\boldsymbol{b}_j^{\mathrm{T}} \bar{K}_j)\bar{K}_j\Big),
\end{aligned} \tag{3.18}
$$

where Δ_j denotes the matrix $z_{j,j} I - \bar{K}_j$. The parentheses indicate the correct order in which the operations should be executed to obtain the right complexity ($\mathcal{O}\big((n-j+1)^2\big)$ per step, hence a total of $\mathcal{O}(n^3)$ for the splitting). Many of the terms are already computed for the lower-order conditions, yet the complexity arises significantly. Therefore we recommend these splitting-type techniques when a moderate order of approximation is required.

To construct higher-order approximations with these splitting techniques, one

could use our symmetric polar-type splittings together with a Yoshida-type symmetric combination.

3.5. Assembly of the approximation $F(t, Z)$ to the exponential. For each algorithm that computes the approximation to the exponential, we distinguish two cases: when the approximation is applied to a vector $\boldsymbol{v}$, and when instead the matrix exponential $\exp(Z)$ is required. Since the matrices $X^{[j]}$ are never constructed explicitly and are stored as vectors, computation of the exponentials $\exp(X^{[j]})$ is also never performed explicitly, but is implemented as in the case of the *Householder reflections* [6] when applied to a vector.

First, let us return to (3.13). It is easy to verify that, if we denote by $\alpha_j = \sqrt{\boldsymbol{b}_j^T \boldsymbol{a}_j}$, then $\exp(D)$ has the exact form

$$\exp(D) = \left(1 + 2\sinh^2\left(\frac{\alpha_j}{2}\right)\right) I + \frac{\sinh \alpha_j}{\alpha_j} D,$$

where I is the 2×2 identity matrix. Similar remarks hold about the matrix D^{-1}. Thus, the computation of $\varphi(D) = D^{-1}(\exp(D) - I)$ can be done in very few flops that "do not contribute" to the total cost of the algorithm. Next, if $\boldsymbol{v}, \boldsymbol{k}, \boldsymbol{l}, \boldsymbol{e}_1 \in \mathbb{R}^j$, the assembly of $\exp(\tilde{P}_j)\boldsymbol{v}$ according to (3.13) can be computed in $6j$ operations. If we let j vary between 1 and n, the total cost of the multiplications is hence $3n^2$. This is precisely the complexity of the assembly of the exponential for polar-type splittings, which has the form as in (3.3).

ALGORITHM 5 (polar-type approximation).

% Purpose: Computing the approximant (3.3) applied to a vector $\boldsymbol{v}$
% In: $\boldsymbol{v}$: n-vector
% Z: $n \times n$ matrix containing the nonzero elements of $X^{[i]}$ and $Y^{[m]}$ as:
% $Z(i+1:n, i) = X^{[i]}(i+1:n, i), \quad Z(i, i+1:n) = X^{[i]}(i, i+1:n)$,
% $\operatorname{diag}(Z) = \operatorname{diag}(Y^{[m]})$
% Out: $\boldsymbol{v} := \exp(X^{[1]}) \cdots \exp(X^{[m]}) \exp(Y^{[m]})\boldsymbol{v}$
%
for $k = 1 : n$
  $v_k := \exp(z_{k,k}) v_k$
end
for $j = n - 1 : -1 : 1$
  $\boldsymbol{a}_j := [0; Z(j+1:n, j)]$,
  $\boldsymbol{b}_j := [0; Z(j, j+1:n)^{\mathrm{T}}]$,
  $\boldsymbol{v}_{\text{old}} := \boldsymbol{v}(j:n)$,
  $\alpha_j := \sqrt{\boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{a}_j}$, $\beta_j = \frac{\sinh \alpha_j}{\alpha_j}$ and $\gamma_j = 2\sinh^2(\alpha_j/2)$,
  $D := \left(\begin{smallmatrix} 0 & 1 \\ \alpha_j^2 & 0 \end{smallmatrix}\right)$,
  $\varphi(D) := \gamma_j D^{-1} + \beta_j I$,
  $\boldsymbol{w} := \varphi(D) \left[\begin{smallmatrix} v_{\text{old},1} \\ \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{v}_{\text{old}} \end{smallmatrix}\right]$,
  $\boldsymbol{v}_{\text{new}} := [\boldsymbol{a}_j, \boldsymbol{e}_1]\boldsymbol{w} = w_1 \boldsymbol{a}_j + w_2 \boldsymbol{e}_1$,
  $\boldsymbol{v}(j:n) := \boldsymbol{v}_{\text{old}} + \boldsymbol{v}_{\text{new}}$
end

In the case in which the output needs to be applied to an $n \times n$ matrix B, we can apply the above algorithm to each column of B, for a total of $3n^3$ operations. This complexity can be reduced to about $2n^3$ taking into account that the vector

$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{n-1}]^{\mathrm{T}}$ can be calculated once and for all, depending only on the splitting of the matrix Z and not in any manner on the columns of B. Also $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{n-1}]^{\mathrm{T}}$ and $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_{n-1}]^{\mathrm{T}}$ can be computed once and stored for later use.

ALGORITHM 6 (symmetric polar-type approximation). The approximation to the exponential is carried out in a manner very similar to that described above in Algorithm 5, except that, since (3.4) is based on a Strang-type splitting, the assembly is also performed in reverse order.

% Purpose: Computing the approximant (3.4) applied to a vector $\boldsymbol{v}$
% In: $\boldsymbol{v}$: n-vector
% Z: $n \times n$ matrix containing the nonzero elements of $X^{[i]}$ and $Y^{[m]}$ as:
% $Z(i+1:n,i) = X^{[i]}(i+1:n,i), \quad Z(i,i+1:n) = X^{[i]}(i,i+1:n),$
% $\mathrm{diag}(Z) = \mathrm{diag}(Y^{[m]})$
% Out: $\boldsymbol{v} := \exp(X^{[1]}) \cdots \exp(X^{[m]}) \exp(Y^{[m]}) \exp(X^{[m]}) \cdots \exp(X^{[1]}) \boldsymbol{v}$
%

for $j = 1 : n-1$
  $\boldsymbol{a}_j := [0; Z(j+1:n, j)]$,
  $\boldsymbol{b}_j := [0; Z(j, j+1:n)^{\mathrm{T}}]$,
  $\boldsymbol{v}_{\mathrm{old}} := \boldsymbol{v}(j:n)$,
  $\alpha_j := \sqrt{\boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{a}_j}, \beta_j = \frac{\sinh \alpha_j}{\alpha_j}, \gamma_j = 2 \sinh^2(\alpha_j/2)$,
  $D := \left(\begin{smallmatrix} 0 & 1 \\ \alpha_j^2 & 0 \end{smallmatrix}\right)$,
  $\varphi(D) := \gamma_j D^{-1} + \beta_j I$,
  $\boldsymbol{w} := \varphi(D) \left[\begin{smallmatrix} v_{\mathrm{old},1} \\ \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{v}_{\mathrm{old}} \end{smallmatrix}\right]$,
  $\boldsymbol{v}_{\mathrm{new}} := [\boldsymbol{a}_j, \boldsymbol{e}_1] \boldsymbol{w} = w_1 \boldsymbol{a}_j + w_2 \boldsymbol{e}_1$,
  $\boldsymbol{v}(j:n) := \boldsymbol{v}(j:n) + \boldsymbol{v}_{\mathrm{new}}$
end

for $k = 1 : n$
  $v_k := \exp(z_{k,k}) v_k$
end

for $j = n-1 : -1 : 1$
  $\boldsymbol{a}_j := [0; Z(j+1:n, j)]$,
  $\boldsymbol{b}_j := [0; Z(j, j+1:n)^{\mathrm{T}}]$,
  $\boldsymbol{v}_{\mathrm{old}} := \boldsymbol{v}(j:n)$,
  $D := \left(\begin{smallmatrix} 0 & 1 \\ \alpha_j^2 & 0 \end{smallmatrix}\right)$,
  $\varphi(D) := \gamma_j D^{-1} + \beta_j I$,
  $\boldsymbol{w} := \varphi(D) \left[\begin{smallmatrix} v_{\mathrm{old},1} \\ \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{v}_{\mathrm{old}} \end{smallmatrix}\right]$,
  $\boldsymbol{v}_{\mathrm{new}} := [\boldsymbol{a}_j, \boldsymbol{e}_1] \boldsymbol{w} = w_1 \boldsymbol{a}_j + w_2 \boldsymbol{e}_1$,
  $\boldsymbol{v}(j:n) := \boldsymbol{v}(j:n) + \boldsymbol{v}_{\mathrm{new}}$
end

The vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ need to be calculated only once and stored for later use in the reverse-order multiplication. The cost of the assembly is roughly twice the cost of the assembly in Algorithm 1; hence it amounts to $5n^2$ operations. (We save n^2 operations omitting the computation of $\boldsymbol{\alpha}$.)

When the result is applied to a matrix B, again we apply the same algorithm to each column of B, which yields n^3 operations. Also in this case the vector $\boldsymbol{\alpha}$ does not depend on B and can be computed once and for all, reducing the cost to $4n^3$

operations. The same remark holds for the vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

It is important to mention that the matrix D might be singular or close to singular (for example, when $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ are close to be orthogonal); hence the computation of $\exp(\tilde{P}_j)$ according to (3.13) may lead to instabilities. In this case, it is recommended to use (3.12) instead of (3.13) or to compute (3.13) by means of singular value decompositions. The cost of (3.12) is twice the cost of (3.13) ($5n^2$ for polar-type assemblies and $9n^2$ for symmetric assemblies for $F(t, Z)$ applied to a vector).

4. Numerical experiments.

4.1. Nonsymmetric polar-type approximations to the exponential. We commence comparing the polar-type order-2 splitting of Algorithm 1 combined with the assembly of the exponential in Algorithm 5 with the $(1, 1)$-Padé approximant for matrices in $\mathfrak{sl}(n)$ and $\mathfrak{so}(n)$, with corresponding groups $\mathrm{SL}(n)$ and $\mathrm{SO}(n)$. We choose diagonal Padé approximants as a benchmark because they are easy to implement, they are the rational approximant with highest order of approximation at the origin, and it is well known that they map quadratic Lie algebras into quadratic Lie groups (but not necessarily other Lie algebras into the corresponding Lie groups). Furthermore, there exists a well-established error analysis for diagonal Padé approximants, and this makes them the standard against which other methods are compared. For using the Padé approximant, we refer to [23].

Table 1 reports the complexity of the method 1+5. A $(1, 1)$-Padé approximant costs $\mathcal{O}\big(2/3n^3\big)$ floating point operations when applied to a vector (essentially the cost of LU-factorizing a linear system) and $\mathcal{O}\big(2\frac{2}{3}n^3\big)$ operations when applied to $n \times n$ matrices. (Note that $(I - Z/2)^{-1}(I + Z/2) = -I + 2(I - Z/2)^{-1}$; hence we reduce to solve a single linear system: $\frac{2}{3}n^3$ flops come from the LU factorization and $2n^3$ from the n forward and backward solution of triangular systems.)

In Figure 1 we compare the number of floating point operations scaled by n^3 for matrices Z up to size 500 as obtained in MATLAB for our polar-type order-2 algorithm (method 1+5) and the $(1, 1)$-Padé approximant both applied to a matrix. We consider the cases when Z is in $\mathfrak{sl}(n)$ and $\mathfrak{so}(n)$. The costs of computing both approximations clearly converges to the theoretical estimates (which in the plot are represented by solid lines) given in Table 1 for large n.

In Figure 2 we compare the accuracy of the two approximations (top plot) for the exponential of a random 10×10 traceless matrix hZ, where Z is normalized so that $\|Z\|_2 = 1$ and $h = \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{64}$. Both methods show a local truncation error of $\mathcal{O}\big(h^3\big)$, revealing that the order of approximation to the exact exponential is 2. The bottom plot shows the error in the determinant as a function of h: the Padé approximant has an error that behaves like h^3, while our method preserves the determinant equal to 1 to machine accuracy.

TABLE 1
Complexity for a polar-type order-2 approximant.

Algorithm	$\mathfrak{sl}(n), \mathfrak{so}(p, q)$		$\mathfrak{so}(n)$	
1+5	Vector	Matrix	Vector	Matrix
Splitting	$1\frac{1}{3}n^3$	$1\frac{1}{3}n^3$	$\frac{2}{3}n^3$	$\frac{2}{3}n^3$
Assembly exp	$3n^2$	$2n^3$	$3n^2$	$2n^3$
Total	$1\frac{1}{3}n^3$	$3\frac{1}{3}n^3$	$\frac{2}{3}n^3$	$2\frac{2}{3}n^3$

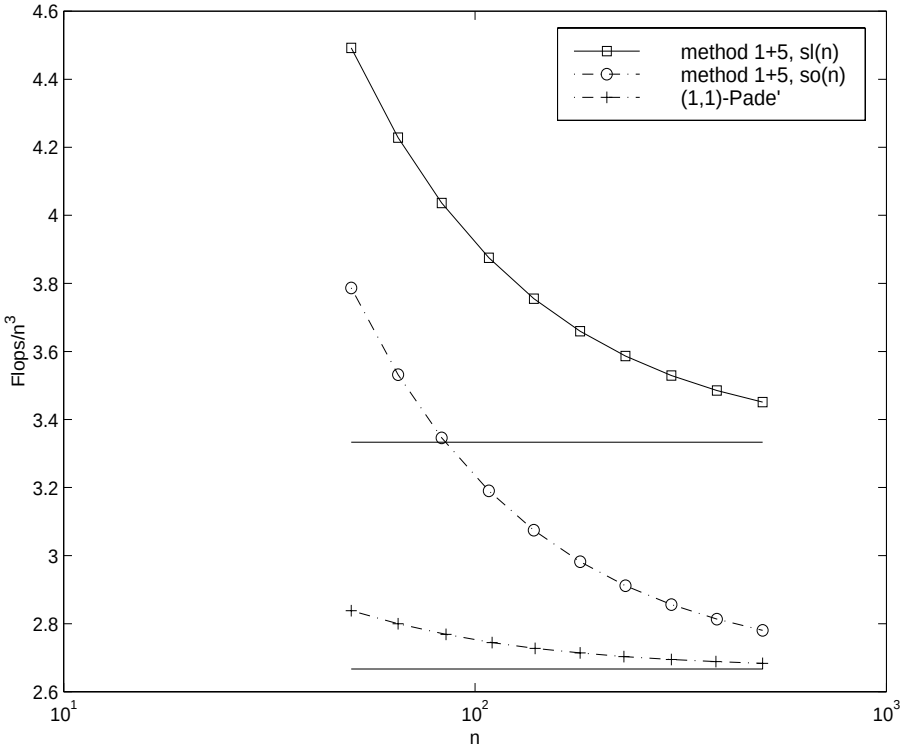


FIG. 1. *Floating point operations (scaled by n^3) versus size for the approximation of the exponential of a matrix in $\mathfrak{sl}(n)$ and in $\mathfrak{so}(n)$ applied to a matrix with the order-2 polar-type algorithm (method $1+5$) and $(1,1)$-Padé approximant.*

In Table 2 we report the complexity of the method 3+5, which yields an approximation to the exponential of order 3. The numbers in parentheses refer to the cost of the algorithm with order-4 corrections.

4.2. Symmetric polar-type approximations to the exponential. We commence comparing our method 2+6, yielding an approximation of order 2, with the $(1,1)$-Padé approximant. Table 3 reports the complexity of the method 2+6.

Clearly, in the matrix-vector case, our methods are one order of magnitude cheaper than the Padé approximant and are definitively to be preferred (see Figure 3 for matrices in $\mathfrak{sl}(n)$). Furthermore, our method maps the approximation in $\mathrm{SL}(n)$, while the Padé approximant does not. When comparing approximations of the matrix exponential applied to a vector, it is a must to consider Krylov subspace methods [20]. We compare the method 2+6 with a Krylov subspace method when Z is a matrix in $\mathfrak{sl}(n)$, normalized so that $\|Z\|_2 = 1$ and $\boldsymbol{v} \in \mathbb{R}^n$ is a vector of unit norm. The Krylov subspaces are obtained by Arnoldi iterations, whose computational cost amounts to about $2mn^2 + 2nm(m-1)$ operations, counting both multiplications and additions. Here m is the dimension of the subspace $K_m \equiv \mathrm{span}\{\boldsymbol{v}, Z\boldsymbol{v}, \ldots, Z^{m-1}\boldsymbol{v}\}$. To obtain the total cost of a Krylov method, we have to add $\mathcal{O}(m^3)$ computations arising from the evaluation of the exponential of the Hessenberg matrix obtained with the Arnoldi iteration, plus $2nm$ operations arising from the multiplication of the latter with the orthogonal basis. However, when n is large and $m \ll n$, these costs are

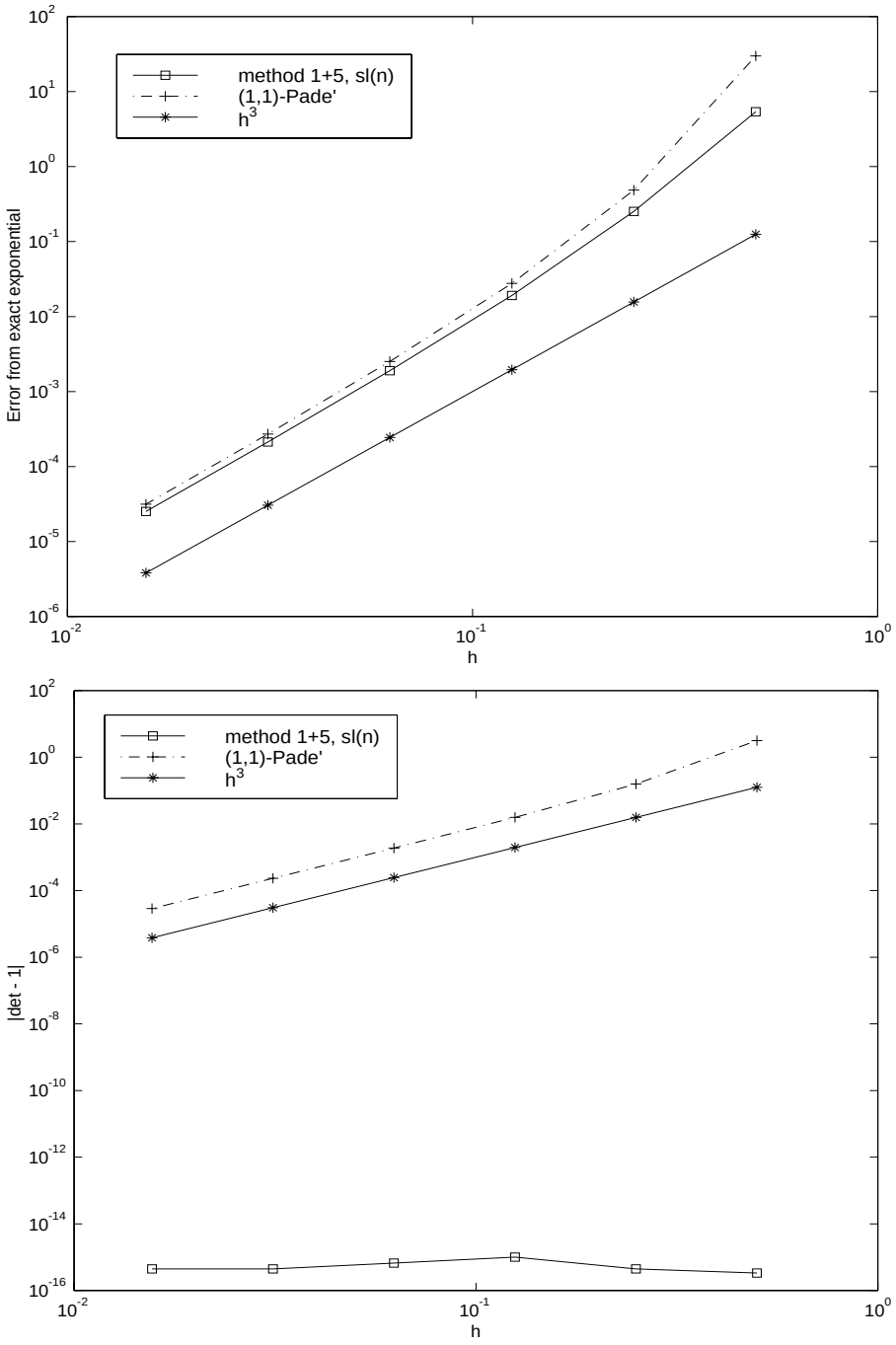


FIG. 2. *Error in the approximation (top) and in the determinant (bottom) versus* h *for the approximation of the exponential of a traceless* 10×10 *matrix of unit norm with the order-*2 *polar-type algorithm (method* $1+5$*) and* $(1,1)$*-Padé approximant.*

TABLE 2
Complexity for a polar-type order-3 approximant. The numbers in parentheses correspond to the coefficients for an order-4 approximation.

Algorithm	$\mathfrak{sl}(n), \mathfrak{so}(p,q)$		$\mathfrak{so}(n)$	
3+5	Vector	Matrix	Vector	Matrix
Splitting	$5(7)n^3$	$5(7)n^3$	$2\frac{1}{2}(4)n^3$	$2\frac{1}{2}(4)n^3$
Assembly exp	$3n^2$	$2n^3$	$3n^2$	$2n^3$
Total	$5(7)n^3$	$7(9)n^3$	$2\frac{1}{2}(4)n^3$	$4\frac{1}{2}(6)n^3$

TABLE 3
Complexity for a symmetric polar-type order-2 approximant.

Algorithm	$\mathfrak{sl}(n), \mathfrak{so}(p,q)$		$\mathfrak{so}(n)$	
2+6	Vector	Matrix	Vector	Matrix
Splitting	n^2	n^2	$\frac{1}{2}n^2$	$\frac{1}{2}n^2$
Assembly exp	$5n^2$	$4n^3$	$5n^2$	$4n^3$
Total	$6n^2$	$4n^3$	$5\frac{1}{2}n^2$	$4n^3$

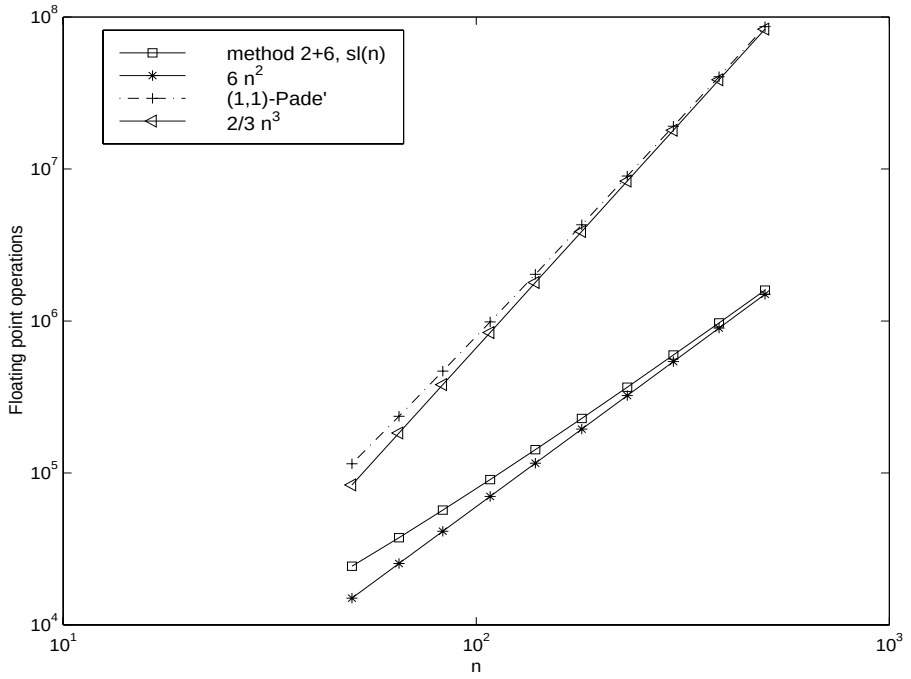


FIG. 3. *Floating point operations versus size for the approximation of the exponential of a matrix in $\mathfrak{sl}(n)$ applied to a vector with the order-2 symmetric polar-type algorithm (method $2+6$) and $(1,1)$-Padé approximant.*

subsumed in that of the Arnoldi iteration, and the leading factor is $2mn^2$ ($2mn^3$ for matrices).

The error, computed as $\|F(1,Z)\boldsymbol{v} - \exp(Z)\boldsymbol{v}\|_2$, and the floating point operations of both approximations for $n = 100, 200, 300$ are given in Table 4. The Krylov method

TABLE 4
Krylov subspace approximations versus the method $2+6$ *for the approximation of* $\exp(Z)\mathsf{v}$.

	Krylov			2+6	
Size n	Error	m	Flops	Error	Flops
100	0.74	1	21041	0.05	66239
	0.01	2	42887		
	4.4e-15	9	219123		
200	0.79	1	82041	0.05	242589
	0.01	2	165477		
	7.1e-15	8	690653		
300	0.75	1	183041	0.06	528939
	0.01	2	368077		
	7.7e-15	8	1510653		

converges very fast: in all three cases 89 iterations are sufficient to obtain almost machine accuracy, while two iterations yield an error which is of the order of method 2+6, at about two-thirds $(0.64, 0.68, 0.69$, respectively) the cost. On the other hand, Krylov methods do not produce an $\mathrm{SL}(n)$ approximation to the exponential unless the computation is performed to machine accuracy, which, in our particular example, is 3.30, 2.84, and 2.85—about three times more costly than the 2+6 algorithm. For the $\mathrm{SO}(n)$ case, it should be noted that if $Z \in \mathfrak{so}(n)$, then the approximation $\boldsymbol{w} \approx \exp(Z)\boldsymbol{v}$ produced by the Krylov method has the feature that $\|w\|_2 = \|v\|_2$ independently of the number m of iterations: in this case, the Hessenberg matrix produced by the Arnoldi iterations is tridiagonal and skew-symmetric, hence its exponential orthogonal. Thus, Krylov methods are the method of choice for actions of $\mathrm{SO}(n)$ on $\mathbb{R}^n$ [17]. One might extrapolate that, if we wish to compute the exponential $\exp(Z)Q$, where $Q \in \mathrm{SO}(n)$, one could perform only a few iterations of the Krylov method to compute $\boldsymbol{w}_i \approx \exp(Z)\boldsymbol{q}_i$, for $i = 1, 2, \ldots, n$, the $\boldsymbol{q}_i$'s being columns of Q. Unfortunately, the approximation $[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]$ ceases to be orthogonal: although $\|\boldsymbol{w}_i\|_2 = 1$, the vectors $\boldsymbol{w}_i$ cease to be linearly independent and the final approximation is not in $\mathrm{SO}(n)$. Similar analysis yields for Stiefel manifolds, unless Krylov methods are implemented to approximate the exponential to machine accuracy.

In passing, we recall that our methods based on a symmetric polar-type decomposition are time-symmetric. Hence it is possible to compose a basic scheme in a symmetric manner, following a technique introduced by Yoshida [24], to obtain higher-order approximations: two orders of accuracy can be obtained at three times the cost of the basic method. For instance, we can use the method 2+6 as a basic algorithm to obtain an approximation of order 4. Thus an approximation of order 4 applied to a vector can be obtained in $17n^2$ operations for $\mathfrak{sl}(n)$ (two splittings and three assemblies), compared to $\mathcal{O}(n^3)$ operations required by the method 4+6.

To conclude our gallery, we compare the method 4+6, an order-4 scheme, whose complexity is described in Table 5, with a $(2,2)$-Padé approximant, which requires $2\frac{2}{3}n^3$ floating point operations when applied to vectors ($2n^3$ for the assembly and $\frac{2}{3}n^3$ for the LU factorization) and $6\frac{2}{3}n^3$ for matrices (since we have to resolve for multiple right-hand sides). The figures obtained by numerical simulations for matrices in $\mathfrak{sl}(n)$ and $\mathrm{SO}(n)$ clearly agree with the theoretical asymptotic values (plotted as solid lines), as shown in Figure 4. The costs of both methods are very similar, as is the error from the exact exponential, although, in the $\mathrm{SL}(n)$ case, the 4+6 scheme preserves the determinant to machine accuracy, while the Padé scheme does not (see Figure 5). Note that a Krylov method iterated to convergence would cost $\approx 18n^3$ operations (assuming that 9 iterations are sufficient to obtain machine accuracy), but it is a

TABLE 5
Complexity for a symmetric polar-type order-4 approximant.

Algorithm	$\mathfrak{sl}(n), \mathfrak{so}(p,q)$		$\mathfrak{so}(n)$	
4+6	Vector	Matrix	Vector	Matrix
Splitting	$5n^3$	$5n^3$	$2\frac{1}{2}n^3$	$2\frac{1}{2}n^3$
Assembly exp	$5n^2$	$4n^3$	$5n^2$	$4n^3$
Total	$5n^3$	$9n^3$	$2\frac{1}{2}n^3$	$6\frac{1}{2}n^3$

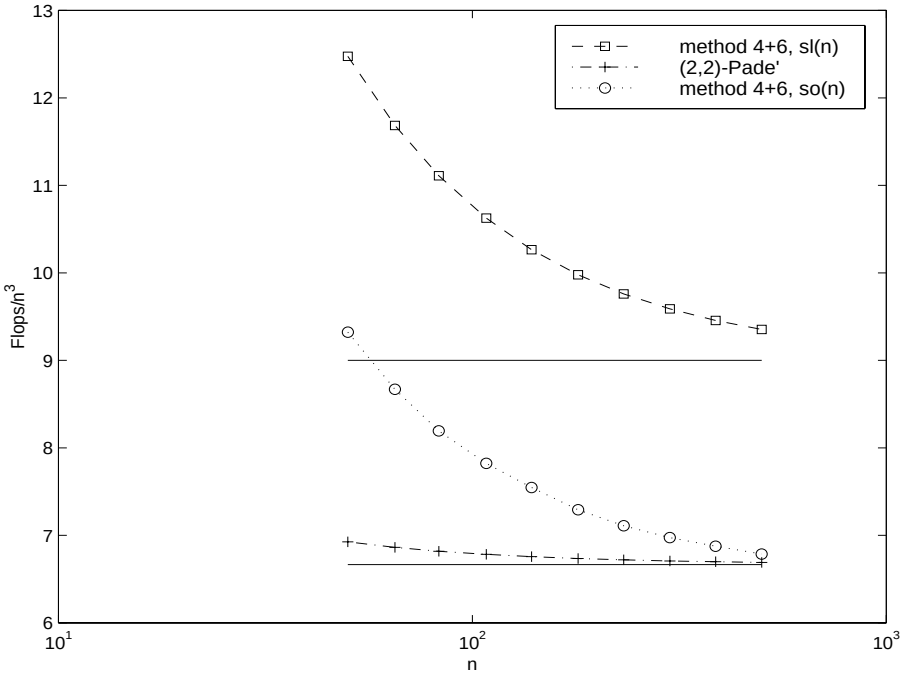


FIG. 4. *Floating point operations (scaled by n^3) versus size for the approximation of the exponential of a matrix in $\mathfrak{sl}(n)$ applied to an $n \times n$ matrix with the order-4 symmetric polar-type algorithm (method $4+6$) and $(2,2)$-Padé approximant.*

clear winner when the exponential is applied to a vector, since our method is an $\mathcal{O}(n^3)$ scheme even in the vector case.

5. Automorphisms based on approximate eigenspace and Schur decompositions. One of the anonymous referees of this paper pointed to possible connections between our theory and the family of splitting methods proposed by Stickel in [21]. Although the original formulation of Stickel does not exactly fit into the framework of this paper, we will see that a modification along the lines of this paper leads to splittings with very interesting properties.

Stickel's approach is based on a commutative splitting derived from the *matrix sign function* $\chi(Z)$. The matrix $\chi(Z)$ has the same eigenvectors as Z, and its eigenvalues are ± 1 or 0 according to whether the corresponding eigenvalues of Z have negative or positive real part, or are purely imaginary.

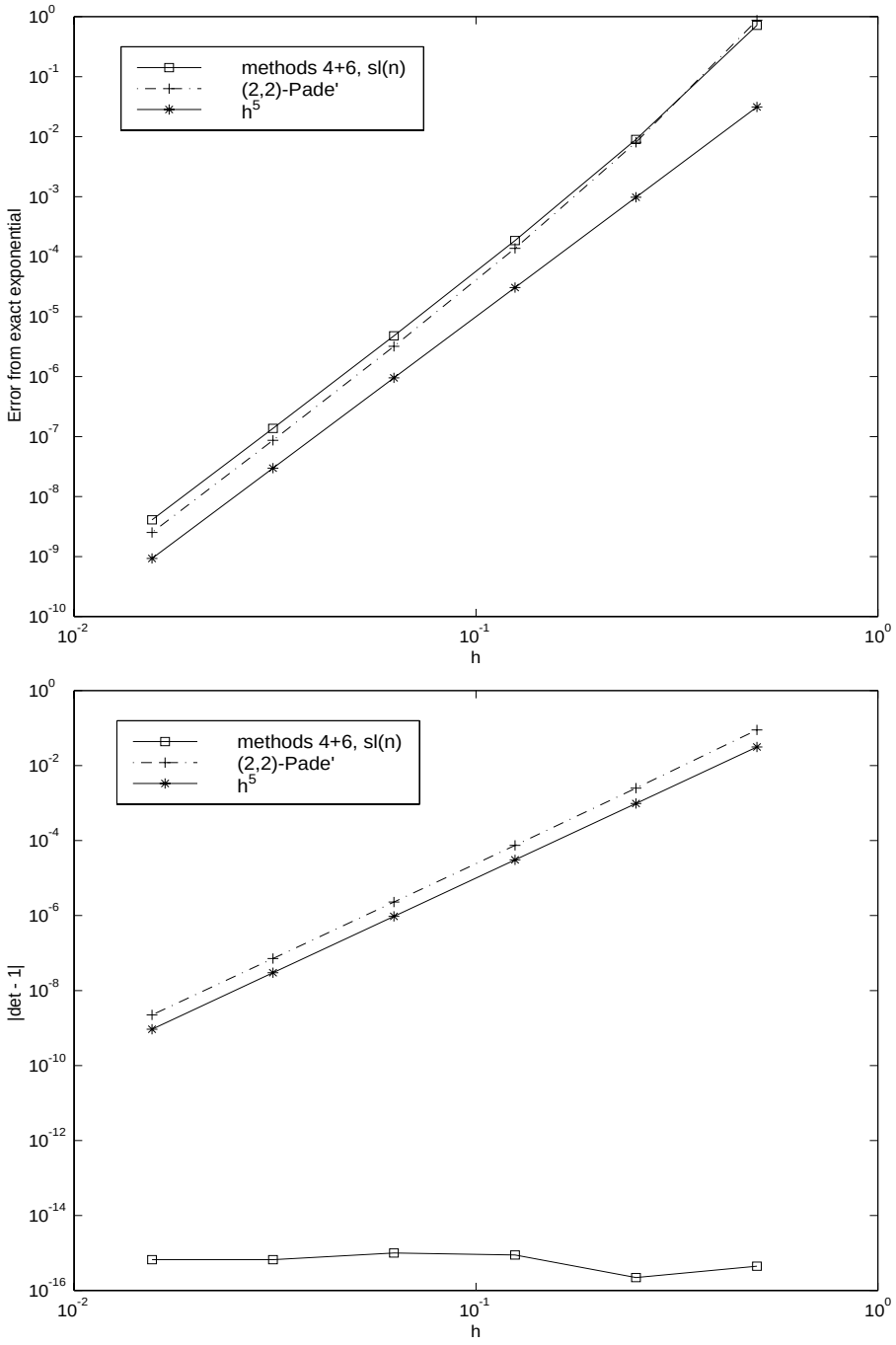


FIG. 5. *Error in the approximation (top) and in the determinant (bottom) versus* h *for the approximation of the exponential of a traceless* 10×10 *matrix of unit norm with the order-*4 *symmetric polar-type algorithm (method* $4+6$*) and* $(2,2)$*-Padé approximant.*

Suppose Z has no purely imaginary eigenvalues.[1] Stickel considers the splitting $Z = Z_1 + Z_2$, where $Z_1 = \frac{1}{2}(Z - Z\chi(Z))$ and $Z_2 = \frac{1}{2}(Z + Z\chi(Z))$. Since $Z\chi(Z) = \chi(Z)Z$, it is clear that Z_1 and Z_2 commute, and hence $\exp(Z) = \exp(Z_1)\exp(Z_2)$ exactly. The same approach may now be employed recursively on Z_1 and Z_2. Note that the map $Z \mapsto Z\chi(Z)$ is an involution, but it is not an involutive automorphism at the algebra level, since it is not linear. Hence the theory of our paper does not directly apply to the splitting methods of Stickel.

A major difficulty in the approach described above is the cost of computing the matrix sign function. It would be of interest to perform similar splittings using approximations S of the sign function. This may lead to a destruction of the commutativity of Z and S, and it should be useful to correct the result by the technique we study in this paper.

Given a matrix S such that $S^2 = I$, the map $Z \mapsto ZS$ is *not* an involutive algebra automorphism. (It is an involution and a linear map; however, in general it is not true that $[Z_1, Z_2]S = [Z_1S, Z_2S]$.) On the other hand, one can consider the map $Z \mapsto SZS$, where S is some approximation of $\chi(Z)$ such that $S^2 = I$, and hence a splitting as we have studied in this paper. Thus, $Z = P + K$, where $P = \frac{1}{2}(Z - SZS)$ and $K = \frac{1}{2}(Z + SZS)$. We can compute $\exp(Z)$ from either of the expansions (3.1) and (3.2), where X and Y are given in (2.4)–(2.6). Note that $XS = -SX$ and $YS = SY$, and that if S is exactly the sign of Z, then $X = 0$. If S is an approximation to $\chi(Z)$, then X and Y are close to commuting, and the expansions converge fast. The exponential of X reduces essentially to a matrix problem of size $p = \operatorname{rank}(S - I)$, while the exponential of Y reduces to a problem of size $n - p$.

How can we produce a suitable approximation $S \approx \chi(Z)$? Let ξ_i and η_j denote right and left eigenvectors of Z, $Z\xi_i = \lambda_i\xi_i$ and $\eta_j^{\mathrm{T}}Z = \lambda_j\eta_j^{\mathrm{T}}$, normalized such that $\eta_j^{\mathrm{T}}\xi_i = \delta_{i,j}$. Then the matrix sign function can be written as

$$\chi(Z) = 2 \sum_{\mathrm{Re}\,(\lambda_i)>0} \xi_i\eta_i^{\mathrm{T}} - I.$$

Given p approximate eigenvectors $\tilde{\xi}_i$ and $\tilde{\eta}_i$, we can use

$$S = 2\sum_{i=1}^{p} \tilde{\xi}_i\tilde{\eta}_i^{\mathrm{T}} - I.$$

Note that if we know p eigenvectors *exactly*, we may use deflation techniques to reduce the size of the problem. Instead, if the eigenvectors are not exactly known, we may use commutators to produce a problem that still can be deflated.

For numerical reasons, a similar approach based on Schur decompositions may be even more attractive. From approximate knowledge of the first p Schur vectors $\tilde{q}_1, \ldots, \tilde{q}_p$ one can employ splittings based on the involutive matrix

$$S = 2\sum_{i=1}^{p} \tilde{q}_i\tilde{q}_i^{\mathrm{T}} - I.$$

The numerical performance of these methods is presently unknown and will have to be addressed in future research.

[1] Stickel introduces shifts $Z - \alpha I$ in the general case, but to keep the exposition simple we avoid the discussion of shifts.

6. Conclusions. In this paper we have introduced numerical algorithms for approximating the matrix exponential. The methods discussed possess the feature that if $Z \in \mathfrak{g}$, then the output is in G, the Lie group of $\mathfrak{g}$, a property that is fundamental to the integration of ODEs by means of Lie-group methods.

The proposed methods have a complexity of $\mathcal{O}(\kappa n^3)$, where n denotes the size of the matrix whose exponential we wish to approximate. Typically, for moderate order (up to order 4), the constant κ is less than 10, whereas the exact computation of a matrix exponential in MATLAB (which employs the scaling and squaring with a Padé approximant) generally costs between $20n^3$ and $30n^3$.

We compare methods of the same order of accuracy applied to a vector $\boldsymbol{v} \in \mathbb{R}^n$ and to a matrix $B \in G$:

- *For the case $F(t,Z)\boldsymbol{v} \approx \exp(tZ)\boldsymbol{v}$, where $\boldsymbol{v}$ is a vector.* Symmetric polar-type methods are slightly cheaper than their nonsymmetric variant. For the $\mathrm{SO}(n)$ case, the complexity of symmetric methods is very comparable to that of diagonal Padé approximants of the same order.

  The complexity of the method 2+6 is $\mathcal{O}(n^2)$, while for the rest of our methods it is $\mathcal{O}(n^3)$. Krylov subspace methods do, however, have the complexity $\mathcal{O}(n^2)$ if the number of iterations is independent of n. Thus, if it is important to stay on the group, we recommend Krylov methods with iteration to machine accuracy for this kind of problem. If convergence of Krylov methods is slow, our methods might be good alternatives. See [8] for accurate bounds on the number m of iterations of Krylov methods.
- *For the case $F(t,Z)B \approx \exp(tZ)B$, with B an $n \times n$ matrix.* Nonsymmetric polar-type methods are marginally cheaper than their symmetric counterpart; however, the latter should be preferred when the underlying ODE scheme is time-symmetric. The proposed methods have a complexity very comparable to that of diagonal Padé approximants of the same order (asymptotically they require slightly less operations in the $\mathrm{SO}(n)$ case); in addition they map $\mathfrak{sl}(n)$ to $\mathrm{SL}(n)$, a property that is shared by neither Padé approximants nor Krylov iterations not carried to convergence.

  For these classes of problems our proposed methods seem to be very competitive.

It should also be noted that significant advantages arise when Z is a banded matrix. For instance, the cost of method 2+6 scales as $\mathcal{O}(nr)$ for $F(t,Z)$ applied to a vector and $\mathcal{O}(n^2 r)$ for $F(t,Z)$ applied to a matrix when Z has bandwidth $2r+1$. The savings are less striking for higher-order methods since commutation usually causes fill-in in the splitting.

As mentioned earlier in the paper, [3] recently proposed similar splitting methods that also produce an output in G when $Z \in \mathfrak{g}$. With respect to their schemes, ours display a slight computational gain: for the $\mathrm{SO}(n)$ case, Celledoni and Iserles propose an order-4 scheme whose complexity is $11\frac{1}{2}n^3$, while our order-4 schemes (method 3+5 with order-4 corrections and method 4+6) cost $6n^3$, $6\frac{1}{2}n^3$ operations—very comparable to the diagonal Padé approximant of the same order.

However, the novelty of our approach lies in the fact that it is based on a rather general theory that can include very different splitting methods already existent in literature (for instance, the method 2+6 is nothing else than a Strang-type splitting), obtaining in an elegant and neat way the otherwise complicated order conditions.

REFERENCES

[1] A. O. Barut, J. R. Zeni, and A. Laufer, *The exponential map for the conformal group* O(2,4), J. Phys. A, 27 (1994), pp. 5239–5250.
[2] R. Carter, G. Segal, and I. Macdonald, *Lectures on Lie Groups and Lie Algebras*, London Math. Soc. Stud. Texts 32, Cambridge University Press, Cambridge, UK, 1995.
[3] E. Celledoni and A. Iserles, *Methods for the approximation of the matrix exponential in a Lie-algebraic setting*, IMA J. Numer. Anal., 21 (2001), pp. 463–488.
[4] E. Celledoni and A. Iserles, *Approximating the exponential from a Lie algebra to a Lie group*, Math. Comp., 69 (2000), pp. 1457–1480.
[5] E. Celledoni, A. Iserles, S. P. Nørsett, and B. Orel, *Complexity Theory for Lie-Group Solvers*, Technical Report NA1999/20, DAMTP, University of Cambridge, Cambridge, UK, 1999.
[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.
[7] S. Helgason, *Differential Geometry, Lie Groups and Symmetric Spaces*, Academic Press, New York, 1978.
[8] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.
[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[10] A. Iserles, H. Munthe-Kaas, S. P. Nørsett, and A. Zanna, *Lie-group methods*, Acta Numer., 9 (2000), pp. 215–365.
[11] J. D. Lawson, *Polar and Ol'shanskii decompositions*, J. Reine Angew. Math., 448 (1994), pp. 191–219.
[12] F. Silva Leite and P. Crouch, *Closed forms for the exponential mapping on matrix Lie groups based on Putzer's method*, J. Math. Phys., 40 (1999), pp. 3561–3568.
[13] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*, Springer-Verlag, New York, 1994.
[14] C. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.
[15] H. Munthe-Kaas, G. R. W. Quispel, and A. Zanna, *Application of Symmetric Spaces and Lie Triple Systems in Numerical Analysis*, Technical Report 217, Department of Informatics, University of Bergen, Norway, 2001.
[16] H. Munthe-Kaas, G. R. W. Quispel, and A. Zanna, *Generalized polar decompositions on Lie groups with involutive automorphisms*, Found. Comput. Math., 1 (2001), pp. 297–324.
[17] H. Munthe-Kaas and A. Zanna, *Numerical integration of differential equations on homogeneous manifolds*, in Foundations of Computational Mathematics, F. Cucker, ed., Springer-Verlag, Berlin, 1997, pp. 305–315.
[18] H. Munthe-Kaas and A. Zanna, *Lie-Group Integrators Based on Generalized Polar Coordinates*, manuscript.
[19] B. Owren and A. Marthinsen, *Integration Methods Based on Canonical Coordinates of the Second Kind*, Technical Report Numerics 5/1999, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway, 1999.
[20] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.
[21] E. U. Stickel, *A splitting method for the calculation of the matrix exponential*, Analysis, 14 (1994), pp. 103–112.
[22] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representation*, Grad. Texts in Math. 102, Springer-Verlag, New York, 1984.
[23] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal., 14 (1977), pp. 600–610.
[24] H. Yoshida, *Construction of higher order symplectic integrators*, Phys. Lett. A, 150 (1990), pp. 262–268.
[25] A. Zanna, *Recurrence Relation for the Factors in the Polar Decomposition on Lie Groups*, Technical Report 192, Department of Informatics, University of Bergen, Norway, 2000.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 863–884
© 2002 Society for Industrial and Applied Mathematics

ON THE BEST RANK-1 APPROXIMATION OF HIGHER-ORDER SUPERSYMMETRIC TENSORS*

ELEFTHERIOS KOFIDIS† AND PHILLIP A. REGALIA‡

Abstract. Recently the problem of determining the best, in the least-squares sense, rank-1 approximation to a higher-order tensor was studied and an iterative method that extends the well-known power method for matrices was proposed for its solution. This higher-order power method is also proposed for the special but important class of supersymmetric tensors, with no change. A simplified version, adapted to the special structure of the supersymmetric problem, is deemed unreliable, as its convergence is not guaranteed. The aim of this paper is to show that a symmetric version of the above method converges under assumptions of convexity (or concavity) for the functional induced by the tensor in question, assumptions that are very often satisfied in practical applications. The use of this version entails significant savings in computational complexity as compared to the unconstrained higher-order power method. Furthermore, a novel method for initializing the iterative process is developed which has been observed to yield an estimate that lies closer to the global optimum than the initialization suggested before. Moreover, its proximity to the global optimum is a priori quantifiable. In the course of the analysis, some important properties that the supersymmetry of a tensor implies for its square matrix unfolding are also studied.

Key words. supersymmetric tensors, rank-1 approximation, higher-order power method, higher-order singular value decomposition

AMS subject classifications. 15A18, 15A57, 15A69

PII. S0895479801387413

1. Introduction.

A *tensor of order N* is an N-way array, i.e., its entries are accessed via N indices.[1] For example, a scalar is a tensor of order 0, a vector is a tensor of order 1, and a matrix is a second-order tensor. Tensors find applications in such diverse fields as physics, signal processing, data analysis, chemometrics, and psychology [4].

The notion of *rank* can also be defined for tensors of order higher than 2. The way this is done is via an extension of the well-known expansion of a matrix in a sum of rank-1 terms. Thus, the rank, R, of an Nth-order tensor $\boldsymbol{\mathcal{T}}$ is the *minimum* number of rank-1 tensors that sum up to $\boldsymbol{\mathcal{T}}$. A rank-1 tensor of order N is given by the generalized outer product of N vectors, $\boldsymbol{u}^{(i)}$, $i = 1, 2, \ldots, N$, i.e., its $(i_1, i_2, \ldots, i_N)$ entry is $\mathcal{T}_{i_1,i_2,\ldots,i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(N)}_{i_N}$. Despite the similarity in their definitions, the ranks of lower- ($N \leq 2$) and higher-order tensors exhibit important differences. For example, the rank of a higher-order tensor is not necessarily upper bounded by the tensor dimensions [9]. Furthermore, there is not a unique way of extending to higher orders the singular value decomposition (SVD) and its connection with least-squares low rank approximation. A multilinear generalization of the matrix SVD, called *higher-order singular value decomposition* (HOSVD), was recently proposed and studied [9] and may be understood as an extension of the so-called Tucker3 model for 3-way

*Received by the editors April 6, 2001; accepted for publication (in revised form) by A. H. Sayed October 17, 2001; published electronically March 5, 2002.

http://www.siam.org/journals/simax/23-3/38741.html

†Department of Informatics and Telecommunications, Division of Communications and Signal Processing, University of Athens, Panepistimioupoli, 157 84 Athens, Greece (kofidis@di.uoa.gr).

‡Département Communications, Image et Traitement de l' Information, Institut National des Télécommunications, 9 rue Charles Fourier, 91011 Évry cedex, France (Phillip.Regalia@int-evry.fr).

[1] Although the tensor admits a more rigorous definition in terms of a tensor product induced by a multilinear mapping [12], the above definition suffices for the purposes of this paper.

arrays (see, e.g., [18]). Despite the many similarities of this decomposition with the second-order SVD (e.g., orthogonality between the vectors $\boldsymbol{u}$ at different terms), its truncation does not provide the best (in the least-squares (LS) sense) low rank approximation to the tensor. Nevertheless, it has been used to initialize a higher-order equivalent of the power method, recently proposed [8, 10] for determining the best rank-1 approximation of Nth-order tensors.

Though an important problem per se, the LS reduced rank tensor approximation plays a central role in the context of blind source separation (BSS) based on higher-order statistics (HOS) [5]. The problem there is to separate and recover statistically independent random processes, $x_1, \ldots, x_K$, with the aid of observations of their linear mixture of the form $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, where the $M \times K$ mixing matrix $\boldsymbol{H}$, the source vector $\boldsymbol{x} = [x_1, \ldots, x_K]^T$, and the disturbance vector $\boldsymbol{n}$ are assumed unknown (and real, for simplicity). The noise vector is also commonly assumed to be Gaussian and independent of $\boldsymbol{x}$. A common method for recovering one of the sources is to project the observation vector $\boldsymbol{y}$ onto an $M \times 1$ vector $\boldsymbol{u}$, chosen so that the normalized kurtosis of the source estimate $z = \boldsymbol{u}^T\boldsymbol{y}$, given by $\frac{\mathrm{cum}_4(z)}{(\mathrm{cum}_2(z))^2}$, is absolutely maximized. It has been shown [8] that this maximization problem is equivalent to that of best approximating the fourth-order cumulant tensor of $\boldsymbol{y}$ by another of rank 1. Hence the higher-order power method (HOPM) of [8, 10] can be employed. However, it should be noticed that the above tensor is *supersymmetric*, i.e., its entries remain unchanged under any permutation of their indices [4]. Such a rich symmetry would be expected to permit a simplified version of the HOPM to be applicable to this kind of tensor. Unfortunately such a symmetric HOPM (S-HOPM) is not convergent for general supersymmetric tensors as claimed in [10] and demonstrated via an example in this paper. Yet, in many cases of practical interest, namely, when the function $g(\boldsymbol{u}) = \sum_{i_1,i_2,\ldots,i_N} \mathcal{T}_{i_1,i_2,\ldots,i_N} u_{i_1} u_{i_2} \cdots u_{i_N}$ is convex (or concave), this symmetric version of the HOPM can be shown to converge to a (local) maximum of the restriction of $|g|$ to the unit sphere. The gain from using this symmetric version comes mainly from the consequent reduction in computational complexity. Though more iterations are usually required for the S-HOPM to converge, the fact that they are N times cheaper than in the general HOPM more than compensates for that, resulting in significant computational complexity savings. The requirements for convexity (concavity) of the function g are always met in the BSS context when the source kurtoses are all of the same sign. For example in communications these kurtoses are negative and g turns out to be concave.[2]

The aim of this paper is to present the S-HOPM and prove that it converges for supersymmetric tensors whose induced polynomial forms enjoy the property of convexity or concavity. A novel scheme for initializing the S-HOPM for fourth-order tensors is also proposed, which has been observed to almost always outperform the HOSVD-based scheme of [10]. Moreover, it allows for an a priori quantification of its proximity to the globally optimal solution.

The rest of this paper is organized as follows. Section 2 introduces definitions of basic quantities and tensor operations. The problem is stated in section 3, where the equivalence of the rank-1 approximation problem with that of maximizing an associated functional and the characterization of the stationary points are recalled from [10, 16] for the symmetric case. The HOPM in both its general and symmetric

[2] In fact, the convergence of the S-HOPM for the BSS problem (where it coincides with the well-known superexponential algorithm [26, 14, 16]) has been shown for the case of mixed sign kurtoses as well, when there are at least as many sensors as sources ($M \geq K$) [24].

versions is given in section 4, and the S-HOPM is shown to converge under convexity/concavity assumptions. Some important properties that the supersymmetry of a tensor implies for its square matrix version are proved in section 5. These properties, to be used in the subsequent analysis, are important in their own right since they hold for *any* supersymmetric tensor, regardless of whether the associated function is convex/concave or not. Section 6 develops the new initialization method and derives bounds on its performance. The problem of computing a rank-R approximation ($R > 1$) is briefly discussed in section 7. Section 8 concludes the paper.

1.1. Notation. Vectors will be denoted by bold lowercase letters (e.g., $\boldsymbol{u}$) while bold uppercase letters (e.g., $\boldsymbol{T}$) will denote tensors of second order (i.e., matrices). Higher-order tensors will be denoted by bold, calligraphic, uppercase letters (e.g., $\boldsymbol{\mathcal{T}}$). The symbol $\boldsymbol{I}$ designates the identity matrix, where the dimensions will be understood from the context. The superscript ^{T} will be employed for transposition. The $(i, j, k, \ldots, l)$ element of a tensor $\boldsymbol{\mathcal{T}}$ is denoted by $\mathcal{T}_{i,j,k,\ldots,l}$. All indices are assumed to start from one. The symbol $\otimes$ will be used to denote the (right) Kronecker product. If $\boldsymbol{A}$ is an $m \times n$ matrix, $\mathrm{vec}(\boldsymbol{A})$ will signify the $mn \times 1$ vector that is built from the columns of $\boldsymbol{A}$ stacked one below another. The inverse operator that builds a matrix from a vector is called unvec. Finally, for the sake of simplicity, only real tensors will be considered. The extension of the results to tensors with Hermitian symmetry [9] is straightforward.

2. Basic definitions. This section contains some definitions that will be useful in what follows. Since they have been presented in detail in earlier works [9, 10, 16], they are only briefly recalled here.

DEFINITION 1 (supersymmetric tensor). *A tensor is called supersymmetric if its entries are invariant under any permutation of their indices.*

The notions of scalar product and norm are easily extended to higher orders, as follows.

DEFINITION 2 (tensor scalar product). *The scalar product of two tensors $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$, of the same order, N, and same dimensions, is given by*

$$\langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}} \rangle = \sum_{i_1, i_2, \ldots, i_N} \mathcal{S}_{i_1, i_2, \ldots, i_N} \mathcal{T}_{i_1, i_2, \ldots, i_N}.$$

DEFINITION 3 (Frobenius norm). *The Frobenius norm of a tensor $\boldsymbol{\mathcal{T}}$ of order N is defined as*

$$\|\boldsymbol{\mathcal{T}}\| = \sqrt{\langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{T}} \rangle} = \left(\sum_{i_1, i_2, \ldots, i_N} \mathcal{T}^2_{i_1, i_2, \ldots, i_N} \right)^{1/2}.$$

DEFINITION 4 (matrix unfoldings). *The n-mode matrix unfolding, $\boldsymbol{T}_{(n)}$, of an $M_1 \times M_2 \times \cdots \times M_N$ tensor $\boldsymbol{\mathcal{T}}$ of order N with entries $\mathcal{T}_{i_1, i_2, \ldots, i_N}$ is defined as the $M_n \times M_1 M_2 \cdots M_{n-1} M_{n+1} \cdots M_N$ matrix whose columns are the M_n-dimensional vectors obtained from $\boldsymbol{\mathcal{T}}$ by varying the index i_n and keeping the other indices fixed.*[3]

It is readily verified that, for a supersymmetric tensor, $\boldsymbol{\mathcal{T}}$, all n-mode matrix unfoldings are equal, that is, $\boldsymbol{T}_{(1)} = \boldsymbol{T}_{(2)} = \cdots = \boldsymbol{T}_{(N)}$. A square matrix version for supersymmetric tensors will also be used, as follows.

[3]The order of appearance of the n-mode vectors in $\boldsymbol{T}_{(n)}$ is irrelevant in our context.

DEFINITION 5 (square matrix unfolding). *The square matrix unfolding, $\boldsymbol{T}$, of an $M \times M \times \cdots \times M$ supersymmetric tensor $\boldsymbol{\mathcal{T}}$ of even order $N = 2L$ is given by*

$$T_{m,n} = \mathcal{T}_{i_1,i_2,\ldots,i_L,j_1,j_2,\ldots,j_L}, \tag{2.1}$$

where

$$m = M^{L-1}(i_1 - 1) + \cdots + M(i_{L-1} - 1) + i_L, \;\; 1 \le i_1, \ldots, i_L \le M, \tag{2.2}$$

$$n = M^{L-1}(j_1 - 1) + \cdots + M(j_{L-1} - 1) + j_L, \;\; 1 \le j_1, \ldots, j_L \le M. \tag{2.3}$$

The outer product can be generalized to higher-orders, as follows.

DEFINITION 6 (Tucker product). *The Tucker product of N matrices $\{\boldsymbol{U}^{(n)}\}_{n=1}^{N}$, each of dimension $M_n \times L$, yields an Nth-order tensor $\boldsymbol{\mathcal{T}}$ of dimensions $M_1 \times M_2 \times \cdots \times M_N$ as*

$$\mathcal{T}_{i_1,i_2,\ldots,i_N} = \sum_{l=1}^{L} U^{(1)}_{i_1,l} U^{(2)}_{i_2,l} \cdots U^{(N)}_{i_N,l}$$

and is denoted by

$$\boldsymbol{\mathcal{T}} = \boldsymbol{U}^{(1)} \star \boldsymbol{U}^{(2)} \star \cdots \star \boldsymbol{U}^{(N)}.$$

A weighted outer product will be defined as follows.[4]

DEFINITION 7 (weighted Tucker product). *The weighted Tucker product (or $\boldsymbol{\mathcal{S}}$-product), with core an $L_1 \times L_2 \times \cdots \times L_N$ tensor $\boldsymbol{\mathcal{S}}$, of N matrices $\{\boldsymbol{U}^{(n)}\}_{n=1}^{N}$ of dimensions $M_n \times L_n$ yields an Nth-order $M_1 \times M_2 \times \cdots \times M_N$ tensor as*

$$\mathcal{T}_{i_1,i_2,\ldots,i_N} = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \cdots \sum_{l_N=1}^{L_N} \mathcal{S}_{l_1,l_2,\ldots,l_N} U^{(1)}_{i_1,l_1} U^{(2)}_{i_2,l_2} \cdots U^{(N)}_{i_N,l_N}$$

and is denoted by

$$\boldsymbol{\mathcal{T}} = \boldsymbol{U}^{(1)} \overset{\mathcal{S}}{\star} \boldsymbol{U}^{(2)} \overset{\mathcal{S}}{\star} \cdots \overset{\mathcal{S}}{\star} \boldsymbol{U}^{(N)}.$$

It can easily be seen that the standard Tucker product is a weighted $\boldsymbol{\mathcal{I}}$-product, where $\boldsymbol{\mathcal{I}}$ is the identity tensor ($\mathcal{I}_{i_1,i_2,\ldots,i_N} = \delta(i_1, i_2, \ldots, i_N)$). The Tucker product with all $\boldsymbol{U}^{(n)}$ being vectors results in a rank-1 tensor, as follows.

DEFINITION 8 (tensor rank). *The rank, R, of an $M_1 \times M_2 \times \cdots \times M_N$ tensor $\boldsymbol{\mathcal{T}}$ is the minimal number of terms in a finite decomposition of $\boldsymbol{\mathcal{T}}$ of the form*

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \boldsymbol{u}^{(1)}_r \star \boldsymbol{u}^{(2)}_r \star \cdots \star \boldsymbol{u}^{(N)}_r,$$

where $\boldsymbol{u}^{(i)}_r$ are M_i-dimensional column vectors.

A way of extending the SVD from matrices to higher-order tensors is given in the following (see [9]).

THEOREM 1 (HOSVD). *Any $M_1 \times M_2 \times \cdots \times M_N$ tensor $\boldsymbol{\mathcal{T}}$ can be expressed as*

$$\boldsymbol{\mathcal{T}} = \boldsymbol{U}^{(1)} \overset{\mathcal{S}}{\star} \boldsymbol{U}^{(2)} \overset{\mathcal{S}}{\star} \cdots \overset{\mathcal{S}}{\star} \boldsymbol{U}^{(N)},$$

[4] This product is denoted in [9, 10] as $\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \cdots \times_N \boldsymbol{U}^{(N)}$, whereas it takes the form $\boldsymbol{\mathcal{S}} \bullet \boldsymbol{U}^{(1)} \bullet \cdots \bullet \boldsymbol{U}^{(N)}$ in [4].

where

- $\boldsymbol{U}^{(n)}$, $n = 1, 2, \ldots, N$, *are orthogonal* $M_n \times M_n$ *matrices, and*
- *the core tensor* $\boldsymbol{\mathcal{S}}$ *is of the same size as* $\boldsymbol{\mathcal{T}}$*, and its subtensors* $\boldsymbol{\mathcal{S}}_{i_n=\alpha}$*, obtained by fixing the* k*th index to* α*, have the properties of*
  - *all-orthogonality: two subtensors* $\boldsymbol{\mathcal{S}}_{i_n=\alpha}$ *and* $\boldsymbol{\mathcal{S}}_{i_n=\beta}$ *are orthogonal for any possible values of* n *and* $\alpha \neq \beta$*, in the sense that*

$$\langle \boldsymbol{\mathcal{S}}_{i_n=\alpha}, \boldsymbol{\mathcal{S}}_{i_n=\beta} \rangle = 0;$$

  - *ordering: for all* n*,*

$$\|\boldsymbol{\mathcal{S}}_{i_n=1}\| \geq \|\boldsymbol{\mathcal{S}}_{i_n=2}\| \geq \cdots \geq \|\boldsymbol{\mathcal{S}}_{i_n=M_n}\|.$$

The matrix $\boldsymbol{U}^{(n)}$ is computed as the matrix of the left singular vectors of the n-mode unfolding of $\boldsymbol{\mathcal{T}}$, $\boldsymbol{T}_{(n)}$ [9]. The core tensor is then determined as $\boldsymbol{\mathcal{S}} = (\boldsymbol{U}^{(1)})^T \overset{\mathcal{T}}{\star} (\boldsymbol{U}^{(2)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{U}^{(N)})^T$. In the supersymmetric case, all $\boldsymbol{U}^{(n)}$ are equal and, of course, the core tensor is supersymmetric as well. The above multilinear SVD then reduces to the so-called higher-order eigenvalue decomposition (HOEVD) [9]. We will sometimes use the notation $\boldsymbol{U}^{\overset{\mathcal{S}}{\star} N}$ to denote a symmetric $\boldsymbol{\mathcal{S}}$-product.

3. Problem statement. The best LS rank-1 tensor approximation problem is stated below along with its close connection to the maximization of the associated polynomial form on the unit sphere. This property is the higher-order equivalent of an analogous property holding for matrices [11] and is going to play a central role in the subsequent developments. Proofs can be found (for the more general, nonsymmetric case) in [7, 10, 16].

THEOREM 2 (tensor rank-1 approximation). *Given an* N*th-order supersymmetric* $M \times M \times \cdots \times M$ *tensor* $\boldsymbol{\mathcal{T}}$*, consider the problem of determining a scalar* λ *and a vector* $\boldsymbol{u} \in \mathbb{R}^M$ *such that the rank-1 tensor* $\hat{\boldsymbol{\mathcal{T}}} = \lambda \boldsymbol{u}^{\star N}$ *minimizes the function*

$$f(\hat{\boldsymbol{\mathcal{T}}}) = \|\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}\|^2 \tag{3.1}$$

subject to $\boldsymbol{u}$ *having unit norm. Then the unit-norm vector* $\boldsymbol{u}$ *corresponds to a (local) minimum of* (3.1) *if and only if it yields a (local) maximum of* $|g(\boldsymbol{u})|$*, with*

$$g(\boldsymbol{u}) = \sum_{i_1, i_2, \ldots, i_N} \mathcal{T}_{i_1, i_2, \ldots, i_N} u_{i_1} u_{i_2} \cdots u_{i_N} = \langle \boldsymbol{\mathcal{T}}, \boldsymbol{u}^{\star N} \rangle. \tag{3.2}$$

The corresponding value of λ *is* $\lambda = g(\boldsymbol{u})$.

To make the connection with similar results in the second-order case clearer, let us define the functional

$$h(\boldsymbol{u}) = \frac{\langle \boldsymbol{\mathcal{T}}, \boldsymbol{u}^{\star N} \rangle^2}{\langle \boldsymbol{u}, \boldsymbol{u} \rangle^N}, \tag{3.3}$$

whose maximization corresponds to maximizing $g(\boldsymbol{u})$ for $\|\boldsymbol{u}\| = 1$. The above can be viewed as an Nth-order Rayleigh quotient squared. Recall that the corresponding maximization problem for matrices is solved by the dominant eigenpair $(\lambda, \boldsymbol{u})$, where λ is the eigenvalue with the largest absolute value [28, 11]. The stationary points of the corresponding functional $\frac{\boldsymbol{u}^T \boldsymbol{T} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}}$ are the solutions to

$$\boldsymbol{T}\boldsymbol{u} = \frac{\boldsymbol{u}^T \boldsymbol{T} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} \boldsymbol{u}. \tag{3.4}$$

The analogous result for the Nth-order case is as follows [7, 10, 16].

THEOREM 3 (characterization of stationary points). *The unit-norm vector* $\boldsymbol{u}$ *is a stationary point of the functional* h *of* (3.3) *if and only if*

$$\sum_{i_2,\ldots,i_N} \mathcal{T}_{i_1,i_2,\ldots,i_N} u_{i_2} u_{i_3} \cdots u_{i_N} = \lambda u_{i_1} \text{ for all } i_1 \tag{3.5}$$

or, equivalently,

$$\boldsymbol{I} \overset{\mathcal{T}}{\star} \underbrace{(\boldsymbol{u}^T) \overset{\mathcal{T}}{\star} (\boldsymbol{u}^T) \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}^T)}_{N-1 \text{ times}} = \lambda \boldsymbol{u}, \tag{3.6}$$

with $\lambda = \langle \boldsymbol{\mathcal{T}}, \boldsymbol{u}^{\star N} \rangle$.

4. The HOPM.

4.1. General case. An iterative use of (3.4) leads to the well-known power method for determining the dominant eigenpair of a matrix $\boldsymbol{T}$ [28, 11]. A tensorial equivalent is suggested by (3.6) and will be analyzed in the next subsection. Let us first have a look at the HOPM as given in [8, 10] for a general, not necessarily supersymmetric, $M_1 \times M_2 \times \cdots \times M_N$ tensor $\boldsymbol{\mathcal{T}}$.

ALGORITHM 1. HIGHER-ORDER POWER METHOD (HOPM).

Initialization: $\boldsymbol{u}_0^{(n)} =$ *a unit-norm* $M_n -$ *vector,* $1 \le n \le N$

Iteration: for $k = 1, 2, \ldots$

$$\begin{aligned}
\tilde{\boldsymbol{u}}_k^{(1)} &= \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(2)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(N)})^T, \\
\lambda_k^{(1)} &= \|\tilde{\boldsymbol{u}}_k^{(1)}\|, \\
\boldsymbol{u}_k^{(1)} &= \frac{\tilde{\boldsymbol{u}}_k^{(1)}}{\lambda_k^{(1)}}, \\
\boldsymbol{u}_k^{(2)} &= (\boldsymbol{u}_k^{(1)})^T \overset{\mathcal{T}}{\star} \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(3)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(N)})^T, \\
\lambda_k^{(2)} &= \|\tilde{\boldsymbol{u}}_k^{(2)}\|, \\
\boldsymbol{u}_k^{(2)} &= \frac{\tilde{\boldsymbol{u}}_k^{(2)}}{\lambda_k^{(2)}}, \\
&\vdots \\
\tilde{\boldsymbol{u}}_k^{(n)} &= (\boldsymbol{u}_k^{(1)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_k^{(n-1)})^T \overset{\mathcal{T}}{\star} \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(n+1)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(N)})^T, \\
\lambda_k^{(n)} &= \|\tilde{\boldsymbol{u}}_k^{(n)}\|, \\
\boldsymbol{u}_k^{(n)} &= \frac{\tilde{\boldsymbol{u}}_k^{(n)}}{\lambda_k^{(n)}}, \\
&\vdots \\
\tilde{\boldsymbol{u}}_k^{(N)} &= (\boldsymbol{u}_k^{(1)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_k^{(N-1)})^T \overset{\mathcal{T}}{\star} \boldsymbol{I}, \\
\lambda_k^{(N)} &= \|\tilde{\boldsymbol{u}}_k^{(N)}\|, \\
\boldsymbol{u}_k^{(N)} &= \frac{\tilde{\boldsymbol{u}}_k^{(N)}}{\lambda_k^{(N)}}
\end{aligned}$$

end

Output: $\hat{\boldsymbol{\mathcal{T}}} = \lambda \boldsymbol{u}^{(1)} \overset{\mathcal{T}}{\star} \boldsymbol{u}^{(2)} \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} \boldsymbol{u}^{(N)}$.

The $\boldsymbol{\mathcal{T}}$-product $\tilde{\boldsymbol{u}}_k^{(n)} = (\boldsymbol{u}_k^{(1)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_k^{(n-1)})^T \overset{\mathcal{T}}{\star} \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(n+1)})^T \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^{(N)})^T$ can be implemented as[5]

$$\tilde{\boldsymbol{u}}_k^{(n)} = \boldsymbol{\mathcal{T}}_{(1)}(\boldsymbol{u}_k^{(1)} \otimes \cdots \otimes \boldsymbol{u}_k^{(n-1)} \otimes \boldsymbol{u}_{k-1}^{(n+1)} \otimes \cdots \otimes \boldsymbol{u}_{k-1}^{(N)}).$$

N such products have to be computed per iteration.

The above algorithm can be shown to always converge to a (local) maximum of (3.3), with the corresponding value of h given by λ^2. The convergence proof [22] relies on the fact that $\boldsymbol{u}^{(1)} \overset{\mathcal{T}}{\star} \cdots \overset{\mathcal{T}}{\star} \boldsymbol{u}^{(N)}$ is a multilinear function of $\boldsymbol{u}^{(n)}$'s, that is, it is linear with respect to each of them.

An initial estimate of $\boldsymbol{u}^{(n)}$, $1 \leq n \leq N$, which has been observed in [8, 10] to very often lie in the basin of attraction of a globally optimal solution, is given by setting $\boldsymbol{u}_0^{(n)}$ equal to the dominant left singular vector of the n-mode matrix unfolding, $\boldsymbol{\mathcal{T}}_{(n)}$. This is the first column of the matrix $\boldsymbol{U}^{(n)}$ in the HOSVD of $\boldsymbol{\mathcal{T}}$. This initialization method is inspired from what holds in the matrix case, where the best rank-1 approximant is provided by the dominant singular triple [11]. As shown in [10], however, this property does not hold anymore for higher-order arrays, and only some bounds can be derived on the approximation error. The similarities, though, of HOSVD with its second-order counterpart suggest its use to compute an initial estimate for HOPM.

4.2. Symmetric case. As pointed out in [8, 10], for a supersymmetric $\boldsymbol{\mathcal{T}}$, the convergence of Algorithm 1 is to a supersymmetric estimator $\hat{\boldsymbol{\mathcal{T}}}$, with all $\boldsymbol{u}^{(n)}$'s being equal to each other. However, the intermediate results are not necessarily symmetric. It is shown in [10] that, with $\boldsymbol{\mathcal{T}}$ being a supersymmetric $2 \times 2 \times \cdots \times 2$ tensor, the stationary points of the HOPM (solutions to (3.6)) can be determined as the roots of an appropriate Nth-order polynomial. For larger supersymmetric tensors the above algorithm is also proposed, as a constrained version suggested by (3.6) is deemed unreliable since it is not guaranteed to monotonically increase $|g|$. The algorithm suggested by (3.6) is as follows.

ALGORITHM 2. SYMMETRIC HIGHER-ORDER POWER METHOD (S-HOPM).

Initialization: $\boldsymbol{u}_0 = a\ unit\text{-}norm\ M - vector$

Iteration: for $k = 1, 2, \ldots$

$$\tilde{\boldsymbol{u}}_k = \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^T)^{\overset{\mathcal{T}}{\star}(N-1)},$$

$$\boldsymbol{u}_k = \frac{\tilde{\boldsymbol{u}}_k}{\|\tilde{\boldsymbol{u}}_k\|}$$

end

Output: $\hat{\boldsymbol{\mathcal{T}}} = g(\boldsymbol{u})\boldsymbol{u}^{\overset{\mathcal{T}}{\star} N}$

The expression $\tilde{\boldsymbol{u}}_k = \boldsymbol{I} \overset{\mathcal{T}}{\star} (\boldsymbol{u}_{k-1}^T)^{\overset{\mathcal{T}}{\star}(N-1)}$ can as before be rewritten as

$$\tilde{\boldsymbol{u}}_k = \boldsymbol{\mathcal{T}}_{(1)} \underbrace{(\boldsymbol{u}_{k-1} \otimes \cdots \otimes \boldsymbol{u}_{k-1})}_{N-1 \text{ times}}.$$

[5] Recall that all matrices $\boldsymbol{\mathcal{T}}_{(n)}$, $n = 1, \ldots, N$, are equal for a supersymmetric $\boldsymbol{\mathcal{T}}$.

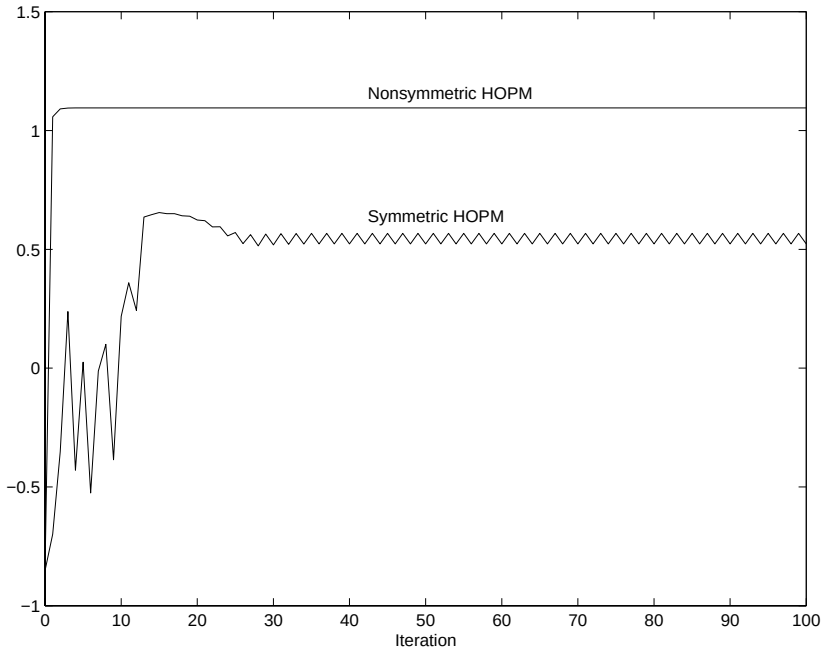


FIG. 4.1. *Results of the HOPM and S-HOPM for the supersymmetric tensor given in Example* 1.

For the special case of $N = 4$, which is the most common one in the BSS problem, the above can also take the following alternative form, in terms of the square matrix unfolding:

$$\tilde{\boldsymbol{u}}_k = \text{unvec}(\boldsymbol{T}(\boldsymbol{u}_{k-1} \otimes \boldsymbol{u}_{k-1}))\boldsymbol{u}_{k-1}.$$

Note that only one such product needs to be computed per iteration, as compared to N for the general HOPM. Thus, if this constrained version is applicable, an N-fold reduction in computational complexity results. However, the form to be optimized is now nonlinear with respect to the sought for vector, $\boldsymbol{u}$, thus rendering the convergence proof for Algorithm 1 not applicable. In fact, as the following example demonstrates, the S-HOPM does not converge for all supersymmetric tensors $\boldsymbol{\mathcal{T}}$.

Example 1. Consider a supersymmetric $3 \times 3 \times 3 \times 3$ tensor with entries

$$\begin{aligned}
&\mathcal{T}_{1111} = 0.2883,\ \mathcal{T}_{1112} = -0.0031,\ \mathcal{T}_{1113} = 0.1973,\ \mathcal{T}_{1122} = -0.2485,\ \mathcal{T}_{1123} = -0.2939,\\
&\mathcal{T}_{1133} = 0.3847,\ \mathcal{T}_{1222} = 0.2972,\ \mathcal{T}_{1223} = 0.1862,\ \mathcal{T}_{1233} = 0.0919,\ \mathcal{T}_{1333} = -0.3619,\\
&\mathcal{T}_{2222} = 0.1241,\ \mathcal{T}_{2223} = -0.3420,\ \mathcal{T}_{2233} = 0.2127,\ \mathcal{T}_{2333} = 0.2727,\ \mathcal{T}_{3333} = -0.3054.
\end{aligned}$$

The results from the application of the general HOPM and its symmetric version, S-HOPM, are depicted in Figure 4.1. The curve for the general HOPM depicts the values of $\langle \boldsymbol{\mathcal{T}}, \boldsymbol{u}_k^{(1)} \star \boldsymbol{u}_k^{(2)} \star \boldsymbol{u}_k^{(3)} \star \boldsymbol{u}_k^{(4)} \rangle$. $g(\boldsymbol{u}_k)$ is plotted for the S-HOPM. Both algorithms are initialized via HOSVD. It is seen that the S-HOPM iterations do not converge.

We will show, however, that Algorithm 2 is convergent if $N = 2L$ is even and $g(\cdot)$ is a *convex (or concave)* function of $\boldsymbol{u}$. Let us recall the meaning of this property [25].

DEFINITION 9 (convex (concave) function). *Let g be a function whose values are real or $\pm\infty$ and whose domain is a convex subset S of $\mathbb{R}^M$. Then g is said to be*

convex on S if its epigraph*,*

$$\text{epi}g \stackrel{\triangle}{=} \{(\boldsymbol{u}, \nu) | \boldsymbol{u} \in S, \nu \in \mathbb{R}, \nu \geq g(\boldsymbol{u})\},$$

is a convex subset of $\mathbb{R}^{M+1}$. A concave function on S is a function whose negative is convex.

From the definition of g, and using the square matrix unfolding of $\boldsymbol{\mathcal{T}}$, it is readily verified that $g(\boldsymbol{u})$ can be written as a polynomial matrix form:

$$g(\boldsymbol{u}) = \underbrace{(\boldsymbol{u} \otimes \boldsymbol{u} \otimes \cdots \otimes \boldsymbol{u})^T}_{L \text{ times}} \boldsymbol{T} \underbrace{(\boldsymbol{u} \otimes \boldsymbol{u} \otimes \cdots \otimes \boldsymbol{u})}_{L \text{ times}}.$$

A necessary and sufficient condition for a twice continuously differentiable function $g(\boldsymbol{u})$ to be convex (concave) on an open convex subset C of $\mathbb{R}^M$ is that its Hessian (i.e., second derivative) matrix be positive (negative) semidefinite on C [25]. Hence, g above is convex (concave) on $\mathbb{R}^M$ if and only if the matrix

$$(\boldsymbol{I} \otimes \underbrace{\boldsymbol{u} \otimes \boldsymbol{u} \otimes \cdots \otimes \boldsymbol{u}}_{L-1 \text{ times}})^T \boldsymbol{T} (\boldsymbol{I} \otimes \underbrace{\boldsymbol{u} \otimes \boldsymbol{u} \otimes \cdots \otimes \boldsymbol{u}}_{L-1 \text{ times}}) \tag{4.1}$$

is positive (negative) semidefinite for all $\boldsymbol{u} \in \mathbb{R}^M$. This implies that $g(\boldsymbol{u})$ has to be nonnegative (nonpositive) for all $\boldsymbol{u}$.

The above condition on the matrix (4.1) will be satisfied *if* $\boldsymbol{T}$ is positive (negative) semidefinite.[6] For example, this holds for the fourth-order cumulant tensor of the output of a linear mixing system, $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, whose sources have kurtoses of the same sign. In that case, $\boldsymbol{\mathcal{T}}$ is given by [16]

$$\boldsymbol{\mathcal{T}} = \boldsymbol{H} \stackrel{\mathcal{S}}{\star} \boldsymbol{H} \stackrel{\mathcal{S}}{\star} \boldsymbol{H} \stackrel{\mathcal{S}}{\star} \boldsymbol{H} \tag{4.2}$$

with $\boldsymbol{\mathcal{S}}$ denoting the (diagonal) tensor of the fourth-order cumulants of $\boldsymbol{x}$. The corresponding matrix $\boldsymbol{T}$ is given by

$$\boldsymbol{T} = (\boldsymbol{H} \odot \boldsymbol{H}) \text{diag}(\text{cum}_4(x_i))(\boldsymbol{H} \odot \boldsymbol{H})^T, \tag{4.3}$$

where $\odot$ is the Khatri–Rao (columnwise Kronecker) product [27].

Notice that even if g is convex (concave), this does not hold for the quotient

$$\frac{g(\boldsymbol{u})}{\|\boldsymbol{u}\|^N}$$

since the unit sphere

$$\Sigma \stackrel{\triangle}{=} \{\boldsymbol{u} \in \mathbb{R}^M | \|\boldsymbol{u}\| = 1\}$$

is not a convex set.

Example 2. Consider the supersymmetric $3 \times 3 \times 3 \times 3$ tensor $\boldsymbol{\mathcal{T}}$ that contains the fourth-order cumulants of the mixture observations for the 3×7 mixing matrix

$$\boldsymbol{H} = \begin{bmatrix} -0.3912 & 0.1427 & 0.3087 & 0.2511 & -0.5408 & 0.3692 & 0.4894 \\ -0.6743 & -0.3816 & -0.5317 & -0.1942 & -0.2120 & -0.0770 & -0.1687 \\ 0.4947 & -0.0364 & -0.3621 & 0.2594 & -0.6336 & 0.1911 & -0.3430 \end{bmatrix}$$

[6] For second-order tensors (matrices) this is also a necessary condition.

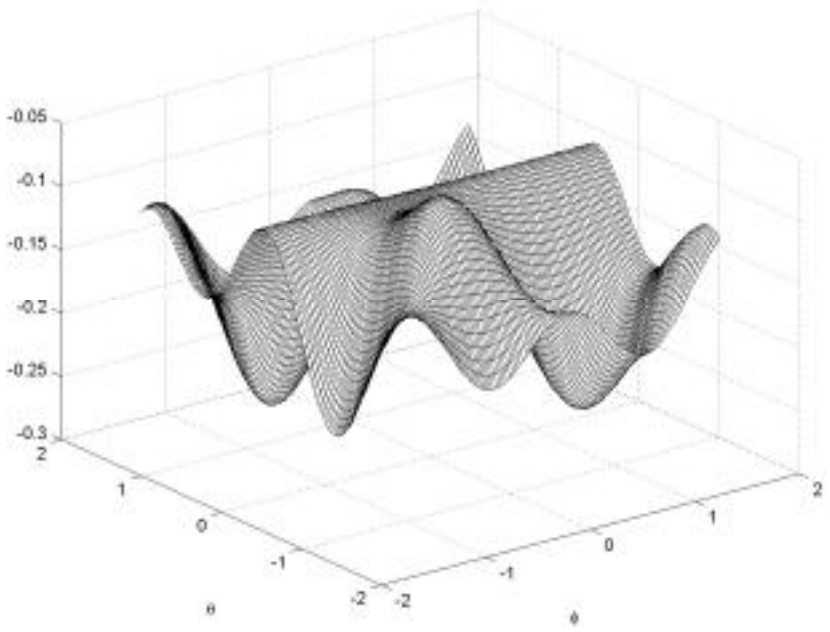


FIG. 4.2. *The function $g(\boldsymbol{u})$ for the tensor of Example* 2*, restricted to the unit sphere.*

and source fourth-order cumulants

$$(\mathrm{cum}_4(x_i))_{i=1}^7 = (-0.3753, -0.3087, -0.7600, -0.0227, -0.4633, -0.0143, -0.5470).$$

The resulting function g is concave and nonpositive. Its restriction to the unit sphere is shown in Figure 4.2. To allow a three-variate function to be plotted, we have parameterized the unit-norm vector $\boldsymbol{u}$ as $\boldsymbol{u} = \begin{bmatrix} \cos(\theta) & \sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) \end{bmatrix}^T$ where the angles θ, ϕ were normalized to the interval $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, i.e., the cosines were constrained to be nonnegative. This can be done since the value assumed by $g(\boldsymbol{u})$ is invariant to sign changes. It is clearly seen that this is no longer concave.

THEOREM 4 (convergence of S-HOPM). *For any supersymmetric Nth-order $M \times M \times \cdots \times M$ tensor $\boldsymbol{\mathcal{T}}$ such that N is even and the associated function g is convex (concave) on $\mathbb{R}^M$, Algorithm* 2 *converges to a local maximum (minimum) of the restriction of g to the unit sphere, Σ, for any initialization, except for saddle points and crest lines leading to such saddle points.*

Proof. Consider first the case that g is convex. This assumption implies that the set epi$g$ is a convex subset of $\mathbb{R}^{M+1}$, and hence a tangent hyperplane at any point $(\boldsymbol{v}, g(\boldsymbol{v}))$ is in fact a supporting hyperplane of epi$g$. This fact is expressed by the so-called *(sub)gradient inequality* [25]

$$g(\boldsymbol{v}_2) - g(\boldsymbol{v}_1) \geq \langle \boldsymbol{v}_2 - \boldsymbol{v}_1, \nabla g(\boldsymbol{v}_1) \rangle$$

holding *for any* vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^M$ (regardless of how distant they may be). To apply this to the problem at hand, set $\boldsymbol{v}_2 = \boldsymbol{u}_k$ and $\boldsymbol{v}_1 = \boldsymbol{u}_{k-1}$ to obtain

$$g(\boldsymbol{u}_k) - g(\boldsymbol{u}_{k-1}) \geq \langle \boldsymbol{u}_k, \nabla g(\boldsymbol{u}_{k-1}) \rangle - \langle \boldsymbol{u}_{k-1}, \nabla g(\boldsymbol{u}_{k-1}) \rangle. \tag{4.4}$$

What we want to show is that if we are not at a stationary point, g is increasing monotonically. It suffices then to show that the right-hand side of (4.4) is positive if

$\boldsymbol{u}_k \neq \boldsymbol{u}_{k-1}$. Note that, for any unit-norm vector $\boldsymbol{u}$, the Cauchy–Schwarz inequality yields

$$\langle \boldsymbol{u}, \nabla g(\boldsymbol{u}_{k-1})\rangle \leq \|\nabla g(\boldsymbol{u}_{k-1})\|, \tag{4.5}$$

where the equality holds if and only if $\boldsymbol{u} = \frac{\nabla g(u_{k-1})}{\|\nabla g(u_{k-1})\|}$. But this is precisely the formula that gives $\boldsymbol{u}_k$ in Algorithm 2. Hence

$$\langle \boldsymbol{u}_k, \nabla g(\boldsymbol{u}_{k-1})\rangle - \langle \boldsymbol{u}_{k-1}, \nabla g(\boldsymbol{u}_{k-1})\rangle > 0,$$

which, in view of (4.4), implies that $g(\boldsymbol{u}_k)$ is increasing with k. The convergence follows from the fact that the restriction of $|g|$ to Σ is bounded from above, namely,

$$\left|\frac{g(\boldsymbol{u})}{\|\boldsymbol{u}\|^N}\right| = \left|\frac{(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\cdots\otimes\boldsymbol{u})^T\boldsymbol{\mathcal{T}}(\boldsymbol{u}\otimes\boldsymbol{u}\otimes\cdots\otimes\boldsymbol{u})}{(\boldsymbol{u}\otimes\cdots\otimes\boldsymbol{u})^T(\boldsymbol{u}\otimes\cdots\otimes\boldsymbol{u})}\right| \leq |\lambda_1|, \tag{4.6}$$

with λ_1 denoting the eigenvalue of $\boldsymbol{\mathcal{T}}$ with largest modulus.

The case of g being concave can be treated as above, by replacing g with $-g$. The only point that we need to comment on is that the quantity $\langle \boldsymbol{u}, \nabla g(\boldsymbol{u}_{k-1})\rangle$ in (4.5) now has to take its minimum value, which occurs when $\boldsymbol{u} = -\frac{\nabla g(u_{k-1})}{\|\nabla g(u_{k-1})\|}$. However, the minus sign is not necessary as it does not affect the value of $g(\boldsymbol{u}_k)$ (recall that N is even). ∎

A distinction is made in [7] between the HOPM, as derived from the Lagrangian equations for the corresponding constrained minimization problem, and a gradient descent procedure. However, it can be shown, following similar arguments to those employed in [19] for the case of a norm function $g(\cdot)$, that the S-HOPM is in fact a gradient recursion using a strategic choice for the step-size parameter.

5. Properties of the matrix $\boldsymbol{T}$. In this section we will state and prove some properties that the supersymmetry of a tensor implies for its square matrix unfolding. The so-called *vec-permutation* matrix [13] plays a central role to the subsequent analysis.

DEFINITION 10 (vec-permutation matrix). *The vec-permutation matrix, $\boldsymbol{I}_{m,n}$, is defined as the $mn \times mn$ permutation matrix that satisfies the equality*

$$\boldsymbol{I}_{m,n}\mathrm{vec}(\boldsymbol{A}) = \mathrm{vec}(\boldsymbol{A}^T) \tag{5.1}$$

for all $m \times n$ matrices $\boldsymbol{A}$.

An explicit way of defining $\boldsymbol{I}_{m,n}$ is given below as a theorem [13].

THEOREM 5 (explicit characterization of $\boldsymbol{I}_{m,n}$). *The (i,j) entry of $\boldsymbol{I}_{m,n}$ is equal to unity if $k = l'$ and $l = k'$ with*

$$\begin{aligned} i &= (k-1)n + l, \quad 1 \leq k \leq m, \quad 1 \leq l \leq n, \\ j &= (k'-1)m + l', \quad 1 \leq k' \leq n, \quad 1 \leq l' \leq m, \end{aligned}$$

and is zero otherwise.

THEOREM 6 (properties of $\boldsymbol{\mathcal{T}}$ (general even order)). *The square matrix unfolding, $\boldsymbol{T}$, of an $M \times M \times \cdots \times M$ supersymmetric tensor $\boldsymbol{\mathcal{T}}$ of order $N = 2L$ satisfies the following properties:*

(i) $\boldsymbol{T}^T = \boldsymbol{T}$.

(ii) $\boldsymbol{I}_{M^{L-1},M}\boldsymbol{T} = \boldsymbol{T}$.

Proof. Recall the definition of $\boldsymbol{T}$ from (2.1)–(2.3). The symmetry of $\boldsymbol{T}$ follows easily from the fact that $\mathcal{T}_{i_1,i_2,\ldots,i_L,j_1,j_2,\ldots,j_L} = \mathcal{T}_{j_1,j_2,\ldots,j_L,i_1,i_2,\ldots,i_L}$.

Call $\boldsymbol{T}'$ the matrix $\boldsymbol{I}_{M^{L-1},M}\boldsymbol{T}$ and consider any of its entries, say $T'_{p,q}$. Write its indices as

$$p = M^{L-1}(k-1) + l, \quad 1 \le k \le M, \quad 1 \le l \le M^{L-1},$$
$$q = M^{L-1}(q_1 - 1) + \cdots + M(q_{L-1} - 1) + q_L, \quad 1 \le q_i \le M, \quad 1 \le i \le L.$$

Then, it is readily seen that Theorem 5 implies

$$T'_{p,q} = T_{M(l-1)+k,q}.$$

Writing further l in the form

$$l = M^{L-2}(l_1 - 1) + \cdots + M(l_{L-2} - 1) + l_{L-1}, \quad 1 \le l_i \le M,$$

and using the supersymmetry of $\boldsymbol{\mathcal{T}}$, the above yields

$$\begin{aligned} T'_{p,q} &= T_{M^{L-1}(l_1-1)+\cdots+M(l_{L-1}-1)+k,q} \\ &= \mathcal{T}_{l_1,l_2,\ldots,l_{L-1},k,q_1,q_2,\ldots,q_L} \\ &= \mathcal{T}_{k,l_1,l_2,\ldots,l_{L-1},q_1,q_2,\ldots,q_L} \\ &= T_{M^{L-1}(k-1)+M^{L-2}(l_1-1)+\cdots+l_{L-1},q} \\ &= T_{M^{L-1}(k-1)+l,q} \\ &= T_{p,q}. \end{aligned}$$

This proves (ii). ∎

Being symmetric, $\boldsymbol{T}$ admits an eigenvalue decomposition with M^L real eigenvalues and orthonormal eigenvectors [28],

$$\boldsymbol{T} = \sum_{i=1}^{M^L} \lambda_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T, \tag{5.2}$$

where

$$\boldsymbol{\xi}_i^T \boldsymbol{\xi}_j = \delta(i,j) \text{ for all } i,j$$

and the eigenvalues are numbered such that

$$|\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_{M^L}|.$$

Let us now confine our attention to fourth-order supersymmetric tensors ($L = 2$). We shall denote the corresponding permutation matrix $\boldsymbol{I}_{M,M}$ by $\boldsymbol{P}$. It is easy to see that $\boldsymbol{P}$ is symmetric. Let us also define the $M \times M$ matrices $\boldsymbol{\Xi}_i$ as the matrix versions of the corresponding eigenvectors of $\boldsymbol{T}$, i.e.,

$$\boldsymbol{\Xi}_i = \text{unvec}(\boldsymbol{\xi}_i).$$

For this special, yet important case, some more information on $\boldsymbol{T}$, involving its eigenstructure, can be revealed [23].

THEOREM 7 (properties of $\boldsymbol{T}$ ($N = 4$)). *The square matrix unfolding of any fourth-order supersymmetric tensor, $\boldsymbol{\mathcal{T}}$, satisfies the following properties:*

(i) $\boldsymbol{PT} = \boldsymbol{TP} = \boldsymbol{PTP} = \boldsymbol{T}$.

(ii) *It admits* $\frac{M(M+1)}{2}$ *eigenvectors* $\boldsymbol{\xi}$ *with* positive *symmetry, i.e.,* $\boldsymbol{P\xi} = +\boldsymbol{\xi}$, *and* $\frac{M(M-1)}{2}$ *eigenvectors with* negative *symmetry, i.e.,* $\boldsymbol{P\xi} = -\boldsymbol{\xi}$. *The corresponding matrices* $\boldsymbol{\Xi}$ *are symmetric* ($\boldsymbol{\Xi} = \boldsymbol{\Xi}^T$) *and skew-symmetric* ($\boldsymbol{\Xi}^T = -\boldsymbol{\Xi}$), *respectively.*

(iii) *All eigenvectors of* $\boldsymbol{T}$ *having negative symmetry must correspond to a zero eigenvalue.*

(iv) $\operatorname{rank}(\boldsymbol{T}) \leq \frac{M(M+1)}{2}$.

Proof.

(i) The proof of (i) follows easily from the symmetry of $\boldsymbol{T}, \boldsymbol{P}$ and property (ii) of Theorem 6.

(ii) We will first prove that all eigenvectors $\boldsymbol{\xi}$ of $\boldsymbol{T}$ enjoy one of the above symmetries. Take the ith eigenpair

$$\boldsymbol{T\xi}_i = \lambda_i \boldsymbol{\xi}_i.$$

Premultiplying the above equation by $\boldsymbol{P}$ and taking into account the equality $\boldsymbol{PT} = \boldsymbol{TP}$ yields

$$\boldsymbol{T} \cdot \boldsymbol{P\xi}_i = \lambda_i \cdot \boldsymbol{P\xi}_i, \tag{5.3}$$

which shows that $\boldsymbol{P\xi}_i$ is also an eigenvector of $\boldsymbol{T}$ for the eigenvalue λ_i. If λ_i is simple, then the corresponding eigenvector is unique up to a sign factor, and hence $\boldsymbol{P\xi}_i = \pm\boldsymbol{\xi}_i$. In the case of a multiple eigenvalue, one can always choose an eigenvector from its invariant space that has the desired symmetry. For example, a possible choice could be the normalized version of $\boldsymbol{\xi}_i' = \boldsymbol{P\xi}_i \pm \boldsymbol{\xi}_i$. This eigenvector is seen to be orthogonal to the rest.

It follows from the definition of $\boldsymbol{P}$ (cf. (5.1)) that the symmetries $\boldsymbol{P\xi}_i = \pm\boldsymbol{\xi}_i$ satisfied by the eigenvectors of $\boldsymbol{T}$ are equivalent to $\boldsymbol{\Xi}_i^T = \pm\boldsymbol{\Xi}_i$, respectively.

Introduce now the following two subspaces of $\mathbb{R}^{M^2}$:

$$S^+ = \{\boldsymbol{x} \in \mathbb{R}^{M^2} | \boldsymbol{Px} = +\boldsymbol{x}\},$$
$$S^- = \{\boldsymbol{x} \in \mathbb{R}^{M^2} | \boldsymbol{Px} = -\boldsymbol{x}\}.$$

These subspaces are orthogonal to each other since if $\boldsymbol{x} \in S^+$ and $\boldsymbol{y} \in S^-$, then the orthogonality of $\boldsymbol{P}$ [13] implies

$$\boldsymbol{x}^T\boldsymbol{y} = -\boldsymbol{x}^T\boldsymbol{P}^T\boldsymbol{Py} = -\boldsymbol{x}^T\boldsymbol{y};$$

hence $\boldsymbol{x}^T\boldsymbol{y} = 0$. It also follows that

$$\dim S^+ = \frac{M(M+1)}{2},$$
$$\dim S^- = \frac{M(M-1)}{2},$$

since, as shown above, parameterizing S^+ (resp., S^-) is equivalent to parameterizing the set of symmetric (resp., skew-symmetric) $M \times M$ matrices. Since $\dim S^+ + \dim S^- = M^2$, we can write the orthogonal decomposition as

$$\mathbb{R}^{M^2} = S^+ \oplus S^-.$$

The result then follows from the fact that the eigenvectors of $\boldsymbol{T}$ belong to either S^+ or S^- and form an orthonormal basis of $\mathbb{R}^{M^2}$.[7]

(iii) Using the property $\boldsymbol{T}\boldsymbol{P} = \boldsymbol{T}$ in (5.3) yields $\boldsymbol{T}\boldsymbol{\xi}_i = \lambda_i \boldsymbol{P}\boldsymbol{\xi}_i$. If $\boldsymbol{\xi}_i$ is such that $\boldsymbol{P}\boldsymbol{\xi}_i = -\boldsymbol{\xi}_i$, then it follows that $\lambda_i \boldsymbol{\xi}_i = -\lambda_i \boldsymbol{\xi}_i$, and $\lambda_i = 0$.

(iv) The proof of (iv) follows directly from (ii) and (iii). ∎

COROLLARY 1 (dominant eigenvector of $\boldsymbol{T}$ $(N = 4)$). *The dominant eigenvector, $\boldsymbol{\xi}_1$, of the square matrix unfolding of a nonzero supersymmetric fourth-order tensor satisfies $\boldsymbol{P}\boldsymbol{\xi}_1 = +\boldsymbol{\xi}_1$. Equivalently, its $M \times M$ matrix version, $\boldsymbol{\Xi}_1$, is symmetric.*

Proof. The proof follows from Theorem 7(iii) since $\lambda_1 \neq 0$. ∎

6. New initialization. We derive here an alternative initialization scheme for the S-HOPM for fourth-order tensors that is observed to be more effective than that based on the HOSVD in approaching the globally optimum point. The starting point is the inequality (4.6) that becomes an equality if and only if $\boldsymbol{u} \otimes \boldsymbol{u}$ coincides with a dominant eigenvector, $\pm\boldsymbol{\xi}_1$. That is, the global maximum of $h(\boldsymbol{u})$ (cf. (3.3)) would be attained if $\boldsymbol{\xi}_1$ could be written as a "Kronecker square," something which is in general not true.

Nonetheless, this remark suggests a way of computing an initial estimate for $\boldsymbol{u}$, namely, setting it equal to the best, in the LS sense, "Kronecker square root" of $\boldsymbol{\xi}_1$:

$$\boldsymbol{u} = \arg \min_{\varsigma \in \mathbb{R}, \|s\|=1} \|\boldsymbol{\xi}_1 - \varsigma \boldsymbol{s} \otimes \boldsymbol{s}\|.$$

Equivalently,

$$\boldsymbol{u} = \arg \min_{\varsigma \in \mathbb{R}, \|s\|=1} \|\boldsymbol{\Xi}_1 - \varsigma \boldsymbol{s}\boldsymbol{s}^T\|.$$

Since $\boldsymbol{\Xi}_1$ is symmetric (see Corollary 1), the latter problem is solved by setting $\boldsymbol{u}$ equal to the unit-norm eigenvector of $\boldsymbol{\Xi}_1$ that corresponds to its absolutely largest eigenvalue, say ς_1 [11]. Note that $\varsigma_1^2 \leq \|\boldsymbol{\Xi}_1\|^2 = \|\boldsymbol{\xi}_1\|^2 = 1$.

The proposed initialization method thus involves two symmetric matrix rank-1 approximation problems:

New Initialization

1. $\boldsymbol{\xi}_1$= dominant eigenvector of $\boldsymbol{T}$.
2. $\boldsymbol{u}_0$= dominant eigenvector of unvec$(\boldsymbol{\xi}_1)$.

Using (5.2), $g(\boldsymbol{u}_0)$ can be written as

$$\begin{aligned} g(\boldsymbol{u}_0) &= \langle \boldsymbol{u}_0 \otimes \boldsymbol{u}_0, \boldsymbol{T}(\boldsymbol{u}_0 \otimes \boldsymbol{u}_0) \rangle \\ &= \sum_{i=1}^{M^2} \lambda_i ((\boldsymbol{u}_0 \otimes \boldsymbol{u}_0)^T \boldsymbol{\xi}_i)^2 \\ &= \sum_{i=1}^{M(M+1)/2} \lambda_i (\boldsymbol{u}_0^T \boldsymbol{\Xi}_i \boldsymbol{u}_0)^2, \end{aligned}$$

[7] This also proves that the vec-permutation matrix $P \triangleq \boldsymbol{I}_{M,M}$ has eigenvalues ± 1 with multiplicities $\frac{M(M\pm1)}{2}$, respectively, a property stated in [13].

where use was made of the well-known identity $\mathrm{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^T \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{B})$ [13] and the fact that $\lambda_i = 0$ for $i > \frac{M(M+1)}{2}$. Choosing $\boldsymbol{u}_0$ as above yields

$$g(\boldsymbol{u}_0) = \lambda_1 \varsigma_1^2 + \sum_{i=2}^{M(M+1)/2} \lambda_i (\boldsymbol{u}_0^T \boldsymbol{\Xi}_i \boldsymbol{u}_0)^2.$$

If $\boldsymbol{T}$ is sign (semi)definite (g is then convex/concave), the latter relation implies

$$|g(\boldsymbol{u}_0)| \geq |\lambda_1| \varsigma^2. \tag{6.1}$$

This, in conjunction with (4.6), provides us with lower and upper bounds on the initial value of h, as follows.

THEOREM 8 (bounds on initial value). *The value assumed by h with the suggested initialization, when applied to a supersymmetric tensor $\boldsymbol{\mathcal{T}}$ with sign (semi)definite square matrix unfolding, $\boldsymbol{T}$, is bounded as*

$$\lambda_1^2 \varsigma_1^4 \leq h(\boldsymbol{u}_0) \leq \lambda_1^2,$$

where λ_1 and ς_1 are the absolutely largest eigenvalues of the matrices $\boldsymbol{T}$ and $\boldsymbol{\Xi}_1$, respectively. This initial value approaches the global maximum as the vector $\boldsymbol{\xi}_1$ approaches Kronecker decomposability (i.e., as $|\varsigma_1|$ approaches one).

Example 2 *(continued).* Consider the $3 \times 3 \times 3 \times 3$ tensor given before. The corresponding matrix $\boldsymbol{T}$ is found to have the following singular values:

$$(|\lambda_i|)_{i=1}^9 = (0.2841, 0.2617, 0.2305, 0.0353, 0.0020, 0.0001, 0, 0, 0),$$

agreeing with Theorem 7(iii).[8] We ran the S-HOPM for this tensor, using both the HOSVD-based and the new initialization methods. The results are shown in Figure 6.1. In both cases the iterations converge to a global minimum. Nevertheless, the new initialization scheme is seen to lie much closer to the globally optimal solution than the HOSVD-based scheme. In fact, for this example successive iterations are nearly superfluous. Extensive simulations have shown this to be the typical case for tensors with an associated functional that is convex/concave.

The superior performance of the new method can also be seen in Figure 6.2 where the position of the two initial estimates in the parameter space, as well as the trajectories followed in each case, are shown.

The lower and upper bounds given in Theorem 8 are 0.0444 and 0.0807, respectively, and are seen to be satisfied by the initial value assumed by h in the new initialization scheme, namely, $h(\boldsymbol{u}_0) = 0.0758$. Note that the initial value suggested by the HOSVD-based scheme, namely, 0.0183, does not meet the lower bound.

We have found, however, some examples where one of the initialization methods leads to a local extremum. These cases are rare and, moreover, in all of them h assumes at the suboptimal point a value quite close to the optimal one.

Example 3. Let the tensor $\boldsymbol{\mathcal{T}}$ be given by (4.2) with

$$\boldsymbol{H} = \begin{bmatrix} -0.1413 & -0.8318 & -0.0769 & -0.1434 & 0.4681 & 0.2054 & 0.0210 \\ 0.3194 & 0.0328 & 0.6555 & 0.1696 & 0.0224 & 0.6580 & 0.0716 \\ 0.4123 & -0.4371 & 0.1749 & -0.3828 & -0.6389 & -0.2315 & -0.0065 \end{bmatrix}$$

[8] Note that there are 6 nonzero eigenvalues, whereas there are 7 sources. The number of nonzero eigenvalues of $\boldsymbol{T}$ can reveal the number of (kurtic) sources only in the case that $K \leq \frac{M(M+1)}{2}$.

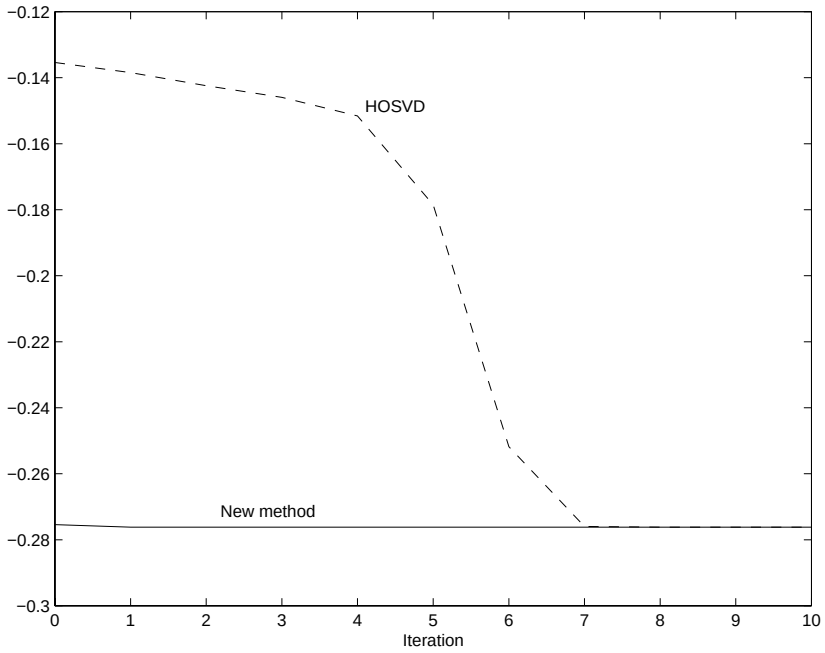


FIG. 6.1. *Results of the S-HOPM for both the HOSVD-based and new initializations.*

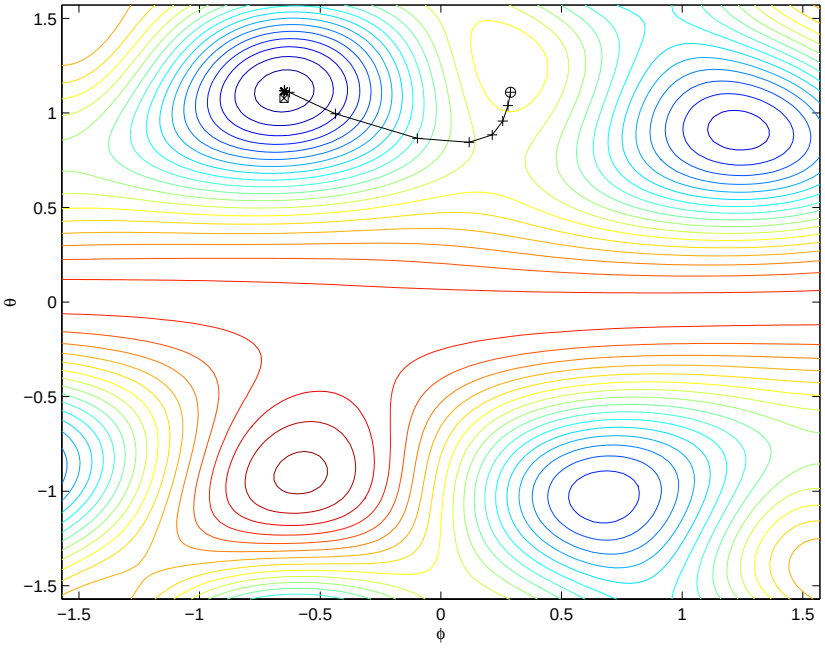


FIG. 6.2. *Visualization of the S-HOPM algorithm for both initialization methods (Example 2). The HOSVD-based initial estimate is denoted by a small circle and the trajectory followed by +'s. The initial estimate provided by the proposed method is denoted by a small square and the subsequent estimates by ×'s.*

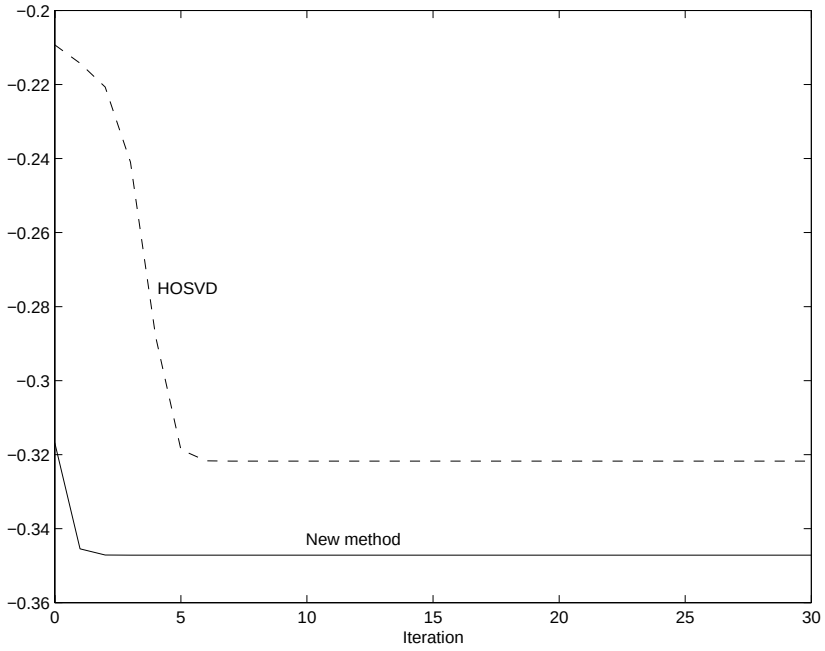


FIG. 6.3. *Results of S-HOPM for an initial estimate based on HOSVD and the new method. In the former case, the algorithm is trapped to a local minimum.*

and the source cumulants

$$(\mathrm{cum}_4(x_i))_{i=1}^7 = (-0.1204, -0.4336, -0.0961, -0.8479, -0.7684, -0.8408, -0.9204).$$

As shown in Figure 6.3, the S-HOPM, initialized with the aid of the HOSVD, is trapped to a local minimum.

The bounds of Theorem 8 are 0.0537 and 0.1272 and the new initialization method yields an initial value of 0.1004 for h. Again, the initial value provided by the HOSVD-based method, namely, 0.0438, does not satisfy the lower bound.

Example 4. The converse is seen to happen for the tensor built as in (4.2) with

$$\boldsymbol{H} = \begin{bmatrix} -0.5100 & 0.3056 & 0.2035 & 0.1959 & 0.4809 & 0.3216 & 0.4816 \\ 0.4881 & -0.4607 & 0.5045 & -0.2727 & 0.2863 & 0.2995 & 0.2211 \\ -0.0529 & -0.4287 & -0.2190 & 0.5228 & -0.3968 & 0.5673 & 0.1133 \end{bmatrix}$$

and

$$(\mathrm{cum}_4(x_i))_{i=1}^7 = (-0.4173, -0.3469, -0.2225, -0.2766, -0.5792, -0.4679, -0.7488).$$

Figure 6.4 shows that the new initialization method leads to a local minimum this time. Nevertheless, as we can see in Figure 6.5 there is no significant difference in the values of h at the local and global minima. Figure 6.6 shows the evolution of the algorithm on the parameter space for the two initializations. In this example, the bounds of Theorem 8, 0.0092 and 0.0387, are met by the initial values for h provided by both the new initialization method and the HOSVD-based one, namely, 0.0181 and 0.0174, respectively.

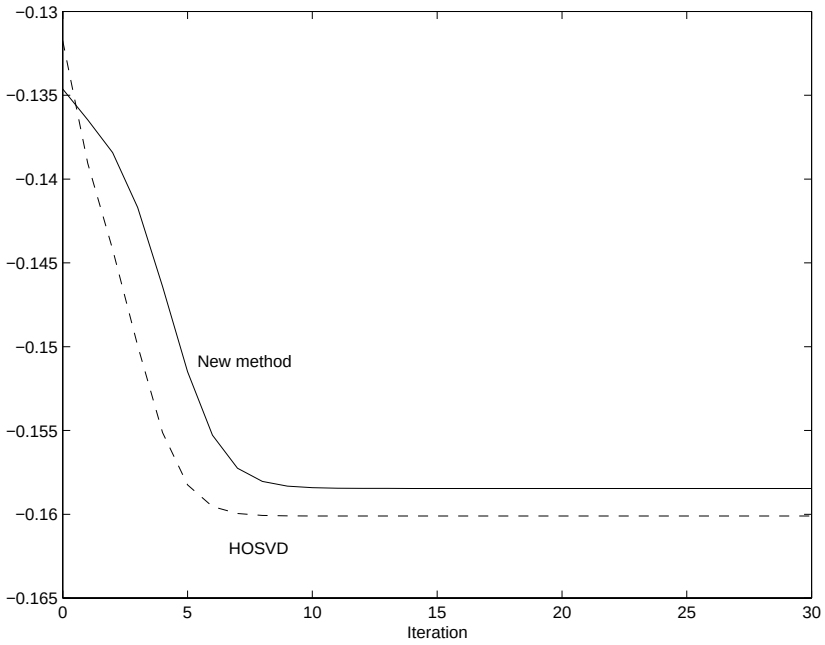


FIG. 6.4. *Results of S-HOPM for both the HOSVD-based and the new initialization methods. The latter leads to a local minimum.*

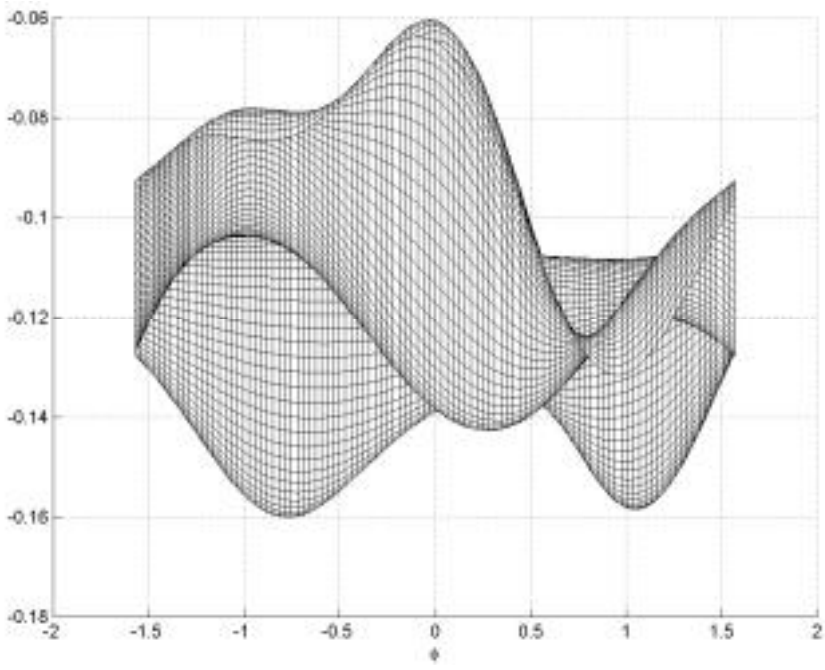


FIG. 6.5. *The function g for the tensor of Example* 4. *Notice the local minimum, corresponding to a value of the function quite close to its global minimum one.*

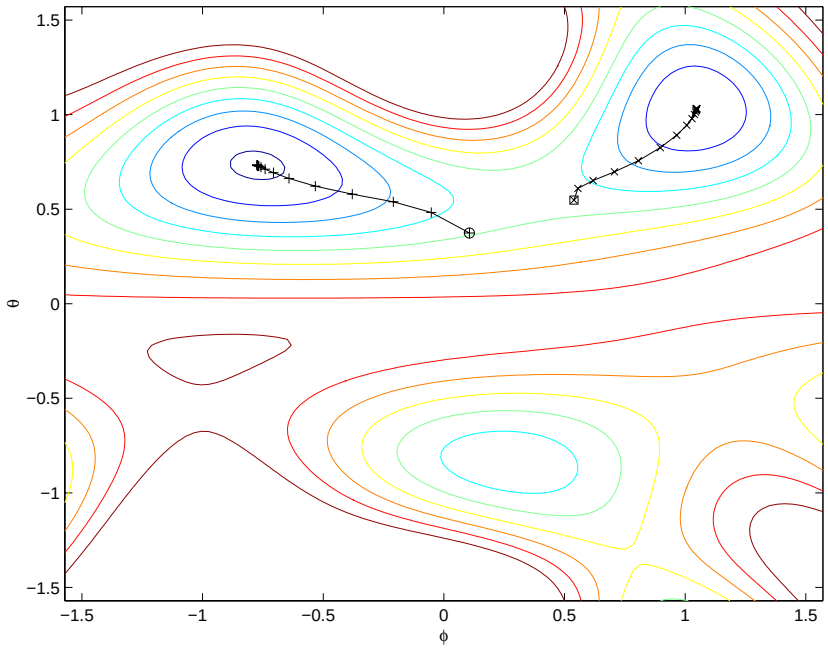


FIG. 6.6. *Visualization of the initial estimates and the trajectories followed by the S-HOPM in Example* 4*, for both initialization methods. Symbols are as in Figure* 6.2.

7. On the best rank-R approximation. It is known that by successively subtracting the LS rank-1 approximation from a given matrix R times results in its LS rank-R approximation [11]. One could wonder whether this fact still holds for higher-order tensors, as pointed out in [3]. Unfortunately, as the following (typical) example demonstrates, this is not the case.

Example 5. Take the tensor $\boldsymbol{\mathcal{T}}$ described in Example 2, normalized to unit norm, and determine its best rank-1 approximant, $\hat{\boldsymbol{\mathcal{T}}}$. Then do the same for the tensor $\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}$, and so on. As shown in Figure 7.1, the norm of the residue indeed decreases to practically zero; however, this is done in about 200 iterations, while $\boldsymbol{\mathcal{T}}$ has rank 7. It is also of interest to note that the rank of $\boldsymbol{\mathcal{T}}$ remains equal to 6 all the way through. ∎

The same iterative process, but with every new rank-1 term being constrained to be orthogonal to the previous ones, was recently studied in [17]. Depending on the definition of the orthogonality adopted, this fails or is not certain to provide a valid rank-R approximation scheme. The above process would work in the case of a tensor with rank less than or equal to its dimension, M. This is the case in the BSS context when the mixing matrix is "tall," i.e., $M \geq K$. In fact, in such a case, all rank($\boldsymbol{\mathcal{T}}$) rank-1 terms can be jointly determined by minimizing the norm $\|\boldsymbol{\mathcal{T}} - \overset{\mathcal{S}}{\boldsymbol{H}^{\star N}}\|$ subject to the constraint that $\boldsymbol{\mathcal{S}}$ is diagonal and $\boldsymbol{H}$ has full column rank (usually assumed to have orthonormal columns). Works coping with this problem include [15, 21]. An algebraic approach via joint diagonalization of the matrices $\boldsymbol{\Xi}_i$ as defined above is proposed in [2]. Note that step 2 of our initialization method is part of the diagonalization of $\boldsymbol{\Xi}_1$. Nonetheless, it is the problem of the recovery of a single source (rank-1 approximation) that our method addresses. Moreover, it is applicable to BSS problems with "fat" mixing matrices ($M < K$) as well (see Example 2).

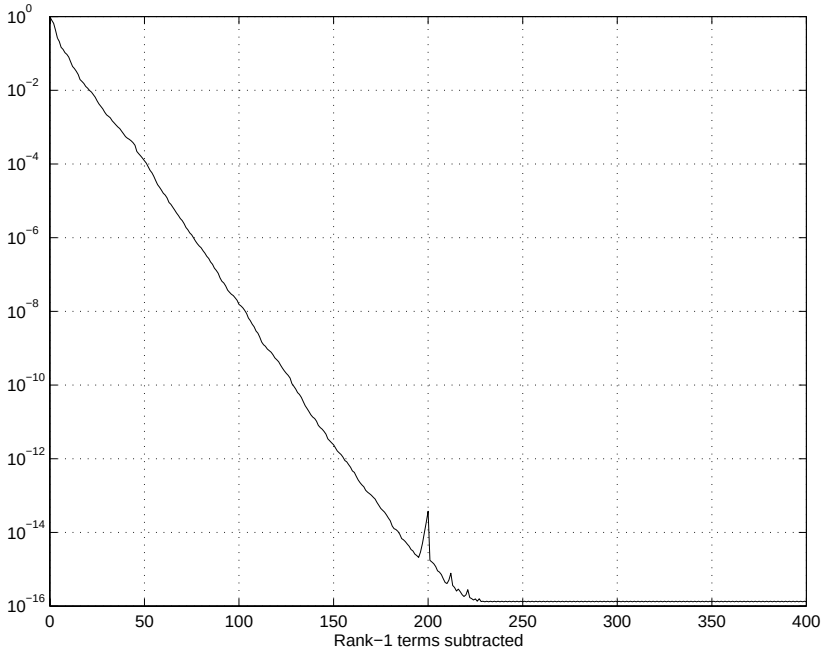


FIG. 7.1. *Norm of the residue remaining when successively subtracting rank-*1 *terms from the tensor of Example* 5. *The spike near iteration* 200 *may be due to numerical artifacts.*

For the latter more challenging problem, [1] develops an algebraic method for determining $\boldsymbol{H}$, based on the assumption of linear independence of the projectors on the spaces spanned by its columns. This can be seen to be equivalent to the matrix $\boldsymbol{H} \odot \boldsymbol{H}$, which occurs in (4.3), having full column rank.

Both algebraic- and optimization-based approaches for expanding a supersymmetric tensor in a sum of rank($\boldsymbol{\mathcal{T}}$) rank-1 terms have been developed in [6] based on its representation in terms of an homogeneous polynomial. The problem then becomes one of expressing the polynomial as a sum of powers of linear forms. Workable algorithms derived this way appear limited to small-sized tensors [3].

Finally, the HOPM can be viewed as a special case of the alternating least squares (ALS) iterative approach common in problems of multilinear model fitting (PARAFAC or CANDECOMP [18, 7, 27]) to multidimensional data. The ALS method can be used with no change to compute a rank-R approximation to a supersymmetric tensor as well, although this way the rich symmetry in the problem is not exploited.

8. Conclusions. The problem of computing the best, in the LS sense, rank-1 approximation to a given Nth-order supersymmetric tensor has been studied in this paper. A symmetric version of the higher-order power method, which was thought to be unreliable, has been shown to be convergent for tensors whose associated polynomial form is convex or concave. A new method for initializing the iterations has been developed for the fourth-order case and observed in extensive simulations to provide an estimate that lies closer to the globally optimal solution than that yielded by the HOSVD. Moreover, the proximity to the optimal solution is a priori quantifiable. It happens, though rarely, that the initial estimate provided by either the HOSVD-based

scheme or the one proposed here be in the basin of attraction of a locally optimal solution. However, this is not a serious problem since in all such cases encountered, the quality of approximation corresponding to the local optimum is observed to be quite close to the best attainable. As a byproduct of our study of the rank-1 approximation problem, some properties satisfied by the square matrix unfolding of *any* supersymmetric tensor have also been derived.

The applicability of the symmetric higher-order power method is accompanied by a significant reduction to the computational complexity of its general version. The convexity/concavity assumptions required to prove its convergence are plausible in many signal processing applications, such as in blind separation of multiuser communications channels, where all source signals have kurtoses of the same sign.

It is not an easy task to extend the results obtained here to the more general problem of computing the best rank-R approximation to a supersymmetric tensor, when $R > 1$. Simply imposing the symmetry constraint on the ALS method of fitting PARAFAC models to general tensors, in the same way that HOPM gave rise to the S-HOPM, does not always result in a convergent procedure.

REFERENCES

[1] J.-F. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, in Proceedings of the IEEE International Conference on Acoust., Speech, and Signal Processing (ICASSP'91), Toronto, Canada, 1991.

[2] J.-F. Cardoso and A. Souloumiac, *Blind beamforming for non-Gaussian signals*, Proc. IEE-F, 140 (1993), pp. 362–370.

[3] P. Comon, *Blind channel identification and extraction of more sources than sensors*, in Proc. SPIE Conference on Advanced Signal Processing VIII, San Diego, 1998.

[4] P. Comon, *Tensor decompositions—State of the art and applications*, keynote address in IMA Conf. Mathematics in Signal Processing, Warwick, UK, 2000.

[5] P. Comon and P. Chevalier, *Blind source separation: Models, concepts, algorithms, and performance*, in Unsupervised Adaptive Filtering, Vol. I, S. Haykin, ed., John Wiley, New York, 2000.

[6] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 96–107.

[7] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. dissertation, Katholieke Universiteit, Leuven, Belgium, 1997.

[8] L. De Lathauwer, P. Comon, B. De Moor, and J. Vandewalle, *Higher-order power method—Application in independent component analysis*, in Proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA'95), Las Vegas, NV, 1995, pp. 91–96.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[12] W. H. Greub, *Multilinear Algebra*, Springer-Verlag, Berlin, 1967.

[13] H. V. Henderson and S. R. Searle, *The vec-permutation matrix, the vec operator and Kronecker products: A review*, Linear and Multilinear Algebra, 9 (1981), pp. 271–288.

[14] A. Hyvärinen and E. Oja, *A fast fixed-point algorithm for independent component analysis*, Neural Comput., 9 (1997), pp. 1483–1492.

[15] A. Hyvärinen, *Fast and robust fixed-point algorithms for independent component analysis*, IEEE Trans. Neural Networks, 10 (1999), pp. 626–634.

[16] E. Kofidis and P. A. Regalia, *Tensor approximation and signal processing applications*, in Structured Matrices in Mathematics, Computer Science and Engineering, Vol. I, Contemp. Math. 280, V. Olshevsky, ed., AMS, Providence, RI, 2001.

[17] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.

[18] P. M. Kroonenberg, *Singular value decompositions of interactions in three-way contingency tables*, in Multiway Data Analysis, R. Coppi and S. Bolasco eds., Elsevier Science Publishers, North Holland, 1989, pp. 169–184.
[19] M. Mboup and P. A. Regalia, *A gradient search interpretation of the super-exponential algorithm*, IEEE Trans. Inform. Theory, 46 (2000), pp. 2731–2734.
[20] P. McCullagh, *Tensor Methods in Statistics*, Chapman and Hall, New York, 1987.
[21] C. B. Papadias, *Globally convergent blind source separation based on a multiuser kurtosis maximization criterion*, IEEE Trans. Signal Process., 48 (2000), pp. 3508–3519.
[22] P. A. Regalia and E. Kofidis, *The higher-order power method revisited: Convergence proofs and effective initialization*, in Proceedings of the IEEE International Conference on Acoust., Speech, and Signal Processing (ICASSP-2000), Istanbul, Turkey, 2000.
[23] P. A. Regalia and E. Kofidis, *A "unimodal" blind equalization criterion*, in Proceedings of the European Signal Processing Conference (EUSIPCO-2000), Tampere, Finland, 2000.
[24] P. A. Regalia and M. Mboup, *Properties of some blind equalization criteria in noisy multiuser environments*, IEEE Trans. Signal Process., 49 (2001), pp. 3112–3122.
[25] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[26] O. Shalvi and E. Weinstein, *Super-exponential methods for blind deconvolution*, IEEE Trans. Inform. Theory, 39 (1993), pp. 504–519.
[27] N. Sidiropoulos and R. Bro, *PARAFAC techniques for signal separation*, in Signal Processing Advances in Wireless & Mobile Communications—Trends in Single- and Multi-User Systems, G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, eds., Prentice-Hall, Englewood Cliffs, NJ, 2001.
[28] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 885–895
© 2002 Society for Industrial and Applied Mathematics

A SPECTRAL CHARACTERIZATION OF GENERALIZED REAL SYMMETRIC CENTROSYMMETRIC AND GENERALIZED REAL SYMMETRIC SKEW-CENTROSYMMETRIC MATRICES*

DAVID TAO† AND MARK YASUDA‡

Abstract. We show that the only real symmetric matrices whose spectrum is invariant modulo sign changes after either row or column reversal are the centrosymmetric matrices; moreover, we prove that the class of real symmetric centrosymmetric matrices can be completely characterized by this property. We also show that the only real symmetric matrices whose spectrum changes by multiplication by i after either row or column reversal are the skew-centrosymmetric matrices; here, too, we show that the class of real symmetric skew-centrosymmetric matrices can be completely characterized by this property of their eigenvalues. We prove both of these spectral characterizations as special cases of results for what we've called *generalized centrosymmetric K-matrices* and *generalized skew-centrosymmetric K-matrices*. Some results illustrating the application of the generalized centrosymmetric spectral characterization to other classes of real symmetric matrices are also given.

Key words. centrosymmetric matrices, skew-centrosymmetric matrices, eigenvalues

AMS subject classifications. 15A18, 15A57

PII. S0895479801386730

1. Introduction.

A *centrosymmetric* matrix A of order n is a square matrix whose elements $a_{i,j}$ satisfy the property

$$a_{i,j} = a_{n-i+1,n-j+1} \text{ for } 1 \leq i, j \leq n.$$

A is called *skew-centrosymmetric* if its elements $a_{i,j}$ satisfy the property

$$a_{i,j} = -a_{n-i+1,n-j+1} \text{ for } 1 \leq i, j \leq n.$$

Although they make a brief appearance in [1], centrosymmetric matrices received their first serious treatment in the 1962 work of Collar [4]. Collar's paper also introduces the notion of skew-centrosymmetric matrices (he uses the term *centroskew*).

The symmetric Toeplitz matrices form an important subclass of the class of symmetric centrosymmetric (sometimes called *doubly symmetric*) matrices. An $n \times n$ matrix T is said to be Toeplitz if there exist numbers $r_{-n+1}, \ldots, r_0, \ldots, r_{n-1}$ such that $t_{i,j} = r_{j-i}$ for $1 \leq i, j \leq n$. As such, Toeplitz matrices are sometimes described as being "constant along the diagonals." Toeplitz matrices occur naturally in digital signal processing applications as well as other areas [7]. Centrosymmetric matrices appear in their own right, for example, in the numerical solution of certain differential equations [2], in the study of some Markov processes [8], and in various physics and engineering problems [6].

In this paper, we establish spectral characterizations for both real symmetric centrosymmetric and real symmetric skew-centrosymmetric matrices as special cases of

*Received by the editors March 25, 2001; accepted for publication (in revised form) by L. Reichel September 27, 2001; published electronically March 5, 2002.
http://www.siam.org/journals/simax/23-3/38673.html
†Department of Defense, 9800 Savage Rd., Ft. Meade, MD 20755 (davidtao@hotmail.com).
‡Raytheon Systems Company, 8680 Balboa Avenue, San Diego, CA 92123 (myasuda@eskimo.com).

results for what we have called *generalized centrosymmetric K-matrices* and *generalized skew-centrosymmetric K-matrices* (defined below).[1] To emphasize the elementary nature of the techniques involved, all results used regarding centrosymmetric and skew-centrosymmetric matrices are developed within this paper.

2. Notation and terminology. Let J represent the *exchange matrix* of order n defined by $J_{i,j} = \delta_{i,n-j+1}$ for $1 \leq i, j \leq n$, where $\delta_{i,j}$ is the Kronecker delta (i.e., J is a matrix with ones on the cross-diagonal and zeros elsewhere). Left-multiplication by J against a matrix A reverses the row order of A. Right-multiplication by J against A reverses the column order of A. The properties of centrosymmetry and skew-centrosymmetry for a matrix can be written succinctly as $AJ = JA$ (equivalently, $A = JAJ$) and $AJ = -JA$ (equivalently, $A = -JAJ$), respectively.

We use K to denote an involutory (i.e., $K^2 = I$) matrix. The exchange matrix J belongs to the set of involutory matrices. We shall refer to matrices A satisfying $AK = KA$ as *generalized centrosymmetric K-matrices*,[2] and matrices A satisfying $AK = -KA$ as *generalized skew-centrosymmetric K-matrices*.

Following the terminology used in Andrew [2], when $x = Jx$ we say that the vector x is *symmetric*. When $x = -Jx$, we say that the vector x is *skew-symmetric*. We extend this terminology to the situation where J is replaced by an involutory matrix K by saying that when $x = Kx$ the vector x is *K-symmetric* and that when $x = -Kx$ the vector x is *K-skew-symmetric*.

Let R and S be multisets (i.e., elements can appear more than once in the collection). We write $R = \pm S$ if the elements of R are the same as those of S up to sign. We write $R = iS$ if $R = \{is \mid s \in S\}$, where $i = \sqrt{-1}$.

Let $\Lambda(A)$ denote the spectrum (eigenvalues) of A, $\{\lambda_i(A)\}_{1 \leq i \leq n}$. Our primary focus will be on the multisets $\Lambda(A)$, $\pm\Lambda(A)$, and $i\Lambda(A)$.

In what follows, $\|x\| = \sqrt{x^T x}$ will denote the Euclidean vector norm of a vector x.

3. Generalized centrosymmetric matrices. Although our focus is primarily on real symmetric matrices, we relax that restriction in the following proposition about generalized centrosymmetric K-matrices.

PROPOSITION 3.1. *Suppose $A \in F^{n\times n}$ and $K \in F^{n\times n}$, where F is a field of characteristic not equal to 2 and K is an involutory matrix. If $AK = KA$, then $\Lambda(A) = \pm\Lambda(KA)$.*

Note: The same theorems and proofs hold *mutatis mutandis* when $\Lambda(KA)$ is replaced with $\Lambda(AK)$ in this proposition and all subsequent results of this article.

Proof. Except for the trivial cases $K = \pm I$, the matrix K has minimal polynomial $m(x) = x^2 - 1$. Since the zeros of $m(x)$ have multiplicity one, there exists a matrix $X \in F^{n\times n}$ such that conjugation of K by X yields the block diagonal form

$$K' \equiv X^{-1}KX = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix},$$

where I represents a block identity matrix, and the sum of the dimensions of the I and $-I$ blocks is n (see [5], for example).

[1] A. Andrew obtained an eigenspace characterization for Hermitian centrosymmetric matrices in [2].

[2] A. Andrew also investigated this generalization in [2].

Conjugation of A by the same matrix X yields a matrix

$$A' \equiv X^{-1}AX = \begin{pmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{pmatrix},$$

where we assume the same partitioning as that for K'. A simple calculation shows that $AK = KA$ if and only if $A'K' = K'A'$ if and only if A'_{12} and A'_{21} are both zero matrices. Consequently,

$$A' = \begin{pmatrix} A'_{11} & 0 \\ 0 & A'_{22} \end{pmatrix} \text{ and } K'A' = \begin{pmatrix} A'_{11} & 0 \\ 0 & -A'_{22} \end{pmatrix}.$$

Since A is similar to A' and KA is similar to $K'A'$, the result is proved. ∎

Remark 3.2. When $K = J$, we can explicitly construct the eigenvector matrix X as follows. Denote the jth column of X by x_j. For $j \leq \lfloor \frac{n}{2} \rfloor$, let the vector x_j have components of 0 everywhere except for a 1 in components j and $n-j+1$. For $j > \lceil \frac{n}{2} \rceil$, let the vector x_j have components of 0 everywhere except for a 1 in component j and a -1 in component $n-j+1$. If n is odd, we let $x_{\lceil \frac{n}{2} \rceil}$ have components of 0 everywhere except for a 1 in component $\lceil \frac{n}{2} \rceil$. Note that the first $\lceil \frac{n}{2} \rceil$ eigenvectors are symmetric, while the remaining $\lfloor \frac{n}{2} \rfloor$ are skew-symmetric.

A centrosymmetric example. Consider the matrices

$$A_1 = \begin{pmatrix} 3 & -2 & -1 & 0 & 1 \\ -2 & 1 & -3 & 1 & 0 \\ -1 & -3 & 5 & -3 & -1 \\ 0 & 1 & -3 & 1 & -2 \\ 1 & 0 & -1 & -2 & 3 \end{pmatrix} \text{ and } JA_1 = \begin{pmatrix} 1 & 0 & -1 & -2 & 3 \\ 0 & 1 & -3 & 1 & -2 \\ -1 & -3 & 5 & -3 & -1 \\ -2 & 1 & -3 & 1 & 0 \\ 3 & -2 & -1 & 0 & 1 \end{pmatrix}.$$

A_1 is centrosymmetric and, consequently, so is JA_1.

$$\Lambda(A_1) = \left\{ -2, \quad 1-\sqrt{5}, \quad 1+\sqrt{5}, \quad 5, \quad 8 \right\}$$

and

$$\Lambda(JA_1) = \left\{ -1-\sqrt{5}, \quad -2, \quad -1+\sqrt{5}, \quad 5, \quad 8 \right\}.$$

A generalized centrosymmetric example. Let

$$A_2 = \begin{pmatrix} 8 & 2 & -5 \\ 2 & -4 & 1 \\ -5 & 1 & 2 \end{pmatrix} \text{ and } K = \begin{pmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-2}{3} \\ \frac{-1}{3} & \frac{2}{3} & \frac{-2}{3} \\ \frac{-2}{3} & \frac{-2}{3} & \frac{-1}{3} \end{pmatrix}.$$

Since $A_2K = KA_2$ and $K^2 = I$, we say that A_2 is a generalized centrosymmetric K-matrix.

$$\Lambda(A_2) = \Lambda(KA_2) = \left\{ 3-3\sqrt{7}, \quad 0, \quad 3+3\sqrt{7} \right\}.$$

Before proving the converse of the real symmetric case of Proposition 3.1, we establish a useful lemma.

LEMMA 3.3. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and nonzero, let $K \in \mathbb{R}^{n\times n}$ be a symmetric involutory matrix, and assume that the largest eigenvalue $\tilde{\lambda}(A)$ of A in magnitude equals the largest eigenvalue $\tilde{\lambda}(KA)$ of KA in magnitude up to sign (i.e.,*

$|\tilde{\lambda}(A)| \equiv \max_i \{|\lambda_i(A)|\}$ and $|\tilde{\lambda}(KA)| \equiv \max_i \{|\lambda_i(KA)|\}$ satisfy $\tilde{\lambda}(A) = \pm\tilde{\lambda}(KA)$). Then there is a nontrivial K-invariant subspace of the eigenspace of A corresponding to $\tilde{\lambda}(A)$. This subspace is also a subspace of the eigenspaces of KA and AK corresponding to their eigenvalues of largest magnitude.

Proof. Let x be a unit eigenvector of KA corresponding to λ, where $|\lambda| = \tilde{\lambda}(KA)$. Then $\lambda = x^T KAx$, and transposing this equation gives $\lambda = x^T AKx$. By the Cauchy–Schwarz inequality, we have that

$$|\lambda| = \left|x^T AKx\right| \leq \|AKx\|.$$

Since $|\lambda|$ is extremal for A, $\|AKx\| \leq |\lambda|$ and therefore $\|AKx\| = |\lambda|$.

Because the Cauchy–Schwarz inequality $|x^T(AKx)| \leq \|AKx\| \cdot \|x\|$ yields an equality only when vectors x and AKx have the same direction up to sign, we may write

$$AKx = \pm\lambda x. \tag{1}$$

Multiplying the equation $KAx = \lambda x$ by K gives

$$Ax = \lambda Kx. \tag{2}$$

Using (1) and (2), we obtain $A^2Kx = \pm\lambda Ax = \pm\lambda^2 Kx$. Since the eigenvalues of A^2 are all nonnegative, we can rewrite (1) as $AKx = \lambda x$.

If $x = \pm Kx$, we're clearly done. Assume this is not the case, so that $x \pm Kx \neq \vec{0}$. Adding and subtracting $AKx = \lambda x$ against $Ax = \lambda Kx$ gives the equations

$$A(x + Kx) = \lambda(x + Kx), \tag{3}$$

$$A(x - Kx) = -\lambda(x - Kx). \tag{4}$$

Observe that the A eigenvector $(x + Kx)$ is K-symmetric, while the A eigenvector $(A - Kx)$ is K-skew-symmetric (i.e., K invariance). Also, we have that the eigenvalue-eigenvector pair $(\lambda, x + Kx)$ of matrix A is simultaneously an eigenpair of the matrix KA (multiply (3) by K) and AK (factor out a K from (3)). Finally, we note that the eigenpair $(-\lambda, x - Kx)$ of matrix A corresponds to an eigenpair $(\lambda, x - Kx)$ of the matrix KA (multiply (4) above by K) and AK (factor out a K from (4) above). This completes the proof. ∎

Remark 3.4. In addition to establishing the lemma, we've also demonstrated that the K-invariant subspace above has a basis consisting of only K-symmetric and K-skew-symmetric vectors.

PROPOSITION 3.5. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, and suppose $\Lambda(A) = \pm\Lambda(KA)$, where $K \in \mathbb{R}^{n\times n}$ is a symmetric involutory matrix. Then $AK = KA$.*

Proof. Since the proposition holds trivially when A is the zero matrix, we may assume that A is nonzero in the following argument. Hence, A will have at least one nonzero eigenvalue and Lemma 3.3 will apply. Since A is symmetric, we are also guaranteed a full set of n independent eigenvectors.

Let S_0 be the nontrivial K invariant subspace of the eigenspace of A corresponding to the eigenvalue $\tilde{\lambda} \equiv \tilde{\lambda}(A)$ as defined in the statement of Lemma 3.3. Then for any $w_0 \in S_0$, we set $\tilde{w}_0 \equiv Kw_0 \in S_0$. If $w_1 \in S_0^{\perp}$, then

$$w_0^T Kw_1 = \left(K\tilde{w}_0\right)^T Kw_1 = \tilde{w}_0^T w_1 = 0.$$

Therefore, $S_0^\perp$ is also invariant under K. Since A is symmetric and maps S_0 to itself,

$$w_0^T A w_1 = (A w_0)^T w_1 = 0.$$

This shows that A also maps $S_0^\perp$ to itself. Therefore, so will KA.

If $x = Kx$ is an eigenvector of A corresponding to the eigenvalue $\tilde{\lambda}$, then

$$(KA - AK)x = (KAx - AKx) = (\tilde{\lambda} Kx - Ax) = (\tilde{\lambda} x - \tilde{\lambda} x) = \vec{0}.$$

Similarly, we can show that $(KA - AK)x = \vec{0}$ when $x = -Kx$. Making use of Remark 3.4, we conclude that $(KA - AK)\, w = \vec{0}$ for any $w \in S_0$.

Since $\mathbb{R}^n = S_0 \oplus S_0^\perp$, if $S_0^\perp$ is trivial, we're done. Otherwise, we apply the above argument to A restricted to $S_0^\perp$. That is, let $S_0^\perp = S_1 \oplus S_1^\perp$, where S_1 is the nontrivial K invariant subspace of the eigenspace corresponding to the largest eigenvalue in magnitude for A restricted to $S_0^\perp$. Then, just as before, we can show that $(KA - AK)\, w = \vec{0}$ for any $w \in S_1$. Continuing in this manner, we establish that $KA - AK$ maps each of the (say m total) nontrivial invariant subspaces S_j associated with A's nonzero eigenvalues to $\vec{0}$.

From above, we know that the eigenspace $S_m = S_{m-1}^\perp$ corresponding to A's 0 eigenvalues is K invariant ($S_m = \{\vec{0}\}$ if A is nonsingular). Therefore KA and AK will both be zero when restricted to S_m. Since $KA - AK$ maps $\mathbb{R}^n = \oplus_{j=0}^m S_j$ to zero, we conclude that $KA = AK$. ∎

Remark 3.6. In the course of proving Proposition 3.5, we have shown that the eigenspace of A corresponding to its nonzero eigenvalues has a basis consisting of K-symmetric and K-skew-symmetric eigenvectors. This is also true for the eigenspace corresponding to the eigenvalue 0.

Proof. As noted above, if $Ax = \vec{0}$, then $\vec{0} = \pm KAx = \pm AKx$. If $x = \pm Kx$, we're done, so assume $x \neq \pm Kx$. As in the proof of Lemma 3.3, we finish by noting that $A(x \pm Kx) = \vec{0}$ and that $x + Kx$ is K-symmetric and $x - Kx$ is K-skew-symmetric. ∎

Combining the real symmetric case of Proposition 3.1 with Proposition 3.5, we arrive at the following characterization of real symmetric generalized centrosymmetric matrices.

THEOREM 3.7. *Suppose $A \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{n\times n}$ are symmetric, and $K^2 = I$. Then $AK = KA$ if and only if $\Lambda(A) = \pm\Lambda(KA)$.*

COROLLARY 3.8. *Let $J \in \mathbb{R}^{n\times n}$ be the exchange matrix. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is centrosymmetric if and only if $\Lambda(A) = \pm\Lambda(JA)$.*

Proof. Let $K = J$ in the statement of Theorem 3.7. ∎

It is convenient at this stage to quantify the number of eigenvalues of A which differ (by sign) from those of KA, where $A \in \mathbb{R}^{n\times n}$ is symmetric generalized K-centrosymmetric. We begin by making the following observations.

LEMMA 3.9. *Suppose $K \in \mathbb{R}^{n\times n}$ is a symmetric involutory matrix, and $A \in \mathbb{R}^{n\times n}$ is a symmetric generalized centrosymmetric K-matrix. Assume that we have expressed A's eigenvector basis in terms of K-symmetric and K-skew-symmetric eigenvectors (Remark* 3.6 *guarantees that this can be done), and assume that*

(I) *for any $\{\lambda_i, \lambda_j\} \in \Lambda(A)$, $|\lambda_i| = |\lambda_j|$ implies $\lambda_i = \lambda_j$.*

Then the nonzero eigenvalues of A which differ by a sign from the eigenvalues of KA are precisely those corresponding to A's K-skew-symmetric eigenvectors.

Proof. If $Ax = \lambda x$, then $KAx = \lambda Kx$. If x is K-symmetric, then $KAx = \lambda x$. If x is K-skew-symmetric, we have that $KAx = -\lambda x$. Condition (I) precludes $-\lambda \in \Lambda(A)$ for $\lambda \neq 0$. ∎

Remark 3.10. Examples of matrices satisfying condition (I) include the positive definite and semidefinite matrices.

LEMMA 3.11. *Let $K \in \mathbb{R}^{n\times n}$ be a symmetric involutory matrix and let $A \in \mathbb{R}^{n\times n}$ be symmetric generalized K-centrosymmetric. Assume that K's eigenvalue 1 has multiplicity n_1 and that K's eigenvalue -1 has multiplicity n_2, where $n_1+n_2 = n$. If V is a basis for the eigenspace of A consisting entirely of K-symmetric and K-skew-symmetric eigenvectors, then V must contain precisely n_1 K-symmetric eigenvectors and n_2 K-skew-symmetric eigenvectors.*

Proof. The lemma is clearly true for $K = \pm I$, so assume this is not the case.

Let x be a K-symmetric eigenvector of A, and let X be the eigenvector matrix of K used in the proof of Proposition 3.1. Then we may express the K-symmetry of x as

$$KXy = Xy, \tag{5}$$

where $y = X^{-1}x$. Since $X^{-1}KX = \left(\begin{smallmatrix} I & 0 \\ 0 & -I \end{smallmatrix}\right)$, we may rewrite (5) as

$$\begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} y = y. \tag{6}$$

Equation (6) holds only if the last n_2 components of y are zero. Therefore, V cannot consist of more than n_1 K-symmetric eigenvectors without violating linear independence. Similarly, we can show that V cannot consist of more than n_2 K-skew-symmetric eigenvectors. Since $n = n_1 + n_2$, the basis V must consist of precisely n_1 K-symmetric eigenvectors and n_2 K-skew-symmetric eigenvectors. ∎

Remark 3.12. Lemma 3.11's quantification of the breakdown of V into K-symmetric eigenvectors and K-skew-symmetric eigenvectors generalizes a result in [3]. For real symmetric centrosymmetric matrices, Cantoni and Butler showed that V is composed of $\lceil \frac{n}{2} \rceil$ symmetric eigenvectors and $\lfloor \frac{n}{2} \rfloor$ skew-symmetric eigenvectors. This result follows from Lemma 3.11 applied to $K = J$, together with observations made in Remark 3.2.

PROPOSITION 3.13. *Let $K \in \mathbb{R}^{n\times n}$ be a symmetric involutory matrix and let $A \in \mathbb{R}^{n\times n}$ be symmetric generalized K-centrosymmetric. Assume that K's eigenvalue -1 has multiplicity n_2. If we let $d(X, Y)$ equal the number of eigenvalues of X which differ from those of Y, then $d(A, KA) \leq n_2$. If we further stipulate condition* (I) *above, we also have the lower bound* $\max\{n_2 - m, 0\} \leq d(A, KA)$, *where m is the multiplicity of A's zero eigenvalue.*

Proof. The proposition clearly holds for $K = \pm I$, so assume this is not the case.

The proof of Lemma 3.9 shows that the eigenpairs of A associated with the K-symmetric eigenvectors are also eigenpairs of KA. From Lemma 3.11, A has n_2 K-skew-symmetric eigenvectors and so it follows that $d(A, KA) \leq n_2$. Under the additional constraint of condition (I), Lemma 3.9 shows that the maximum amount by which $d(A, KA)$ can differ from n_2 is equal to the multiplicity of A's zero eigenvalue. ∎

Remark 3.14. The lower bound in Proposition 3.13 is sharp. For example, the

spectrum of the rank 2 centrosymmetric matrix

$$A = \begin{pmatrix} 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

satisfies condition (I), $\Lambda(A) = \Lambda(JA)$, and $n_2 - m = \lfloor \frac{n}{2} \rfloor - m = 0$. Here, the nonzero eigenvalues correspond to symmetric eigenvectors while the zero eigenvalues correspond to skew-symmetric eigenvectors. Of course, when condition (I) holds and the matrix A is nonsingular, the upper and lower bounds in Proposition 3.13 coincide.

We next prove a result that holds for *any* real symmetric matrix satisfying condition (I).

PROPOSITION 3.15. *Suppose $A \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{n\times n}$ are symmetric, with $K^2 = I$. Assume further that if $\{\lambda_i, \lambda_j\} \in \Lambda(A)$ that $|\lambda_i| = |\lambda_j|$ implies $\lambda_i = \lambda_j$. Then $\Lambda(A) = \Lambda(KA)$ if and only if $A = KA$.*

Proof. The $\Leftarrow$ direction is obvious.

$\Rightarrow$ From Proposition 3.5, $\Lambda(A) = \Lambda(KA)$ implies that A is a generalized centrosymmetric K-matrix. Therefore, Remark 3.6 shows that we can construct a basis for the eigenspace of A consisting entirely of K-symmetric and K-skew-symmetric eigenvectors. Assume we have done so.

If x is any K-symmetric eigenvector of A, then $Ax = \lambda x$ if and only if $KAx = \lambda x$. From Lemma 3.9, we know that all of the K-skew-symmetric eigenvectors of A correspond to an eigenvalue of 0 (a sign change would arise for any nonzero eigenvalue corresponding to a K-skew-symmetric eigenvector), so $Ay = KAy = \vec{0}$ for any K-skew-symmetric eigenvector y.

Since A and KA agree on a basis (e.g., the eigenvectors of A), they must in fact represent the same operator. □

Remark 3.16. In the case where $K = J$, Proposition 3.15 states that if one row reverses any real symmetric matrix satisfying condition (I), then the spectrum of the resulting matrix will always differ from that of the original matrix unless the original matrix is unchanged from the row reversal.

Remark 3.17. The reader may wish to confirm (if he or she has not already done so) that $A_2 = KA_2$ in the generalized centrosymmetric example given earlier.

Using the same type of argument as in the proof of Proposition 3.15, we can also show the following.

PROPOSITION 3.18. *Suppose $A \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{n\times n}$ are symmetric, with $K^2 = I$. Assume further that if $\{\lambda_i, \lambda_j\} \in \Lambda(A)$ that $|\lambda_i| = |\lambda_j|$ implies $\lambda_i = \lambda_j$. Then $\Lambda(A) = \Lambda(-KA)$ if and only if $A = -KA$.*

4. Generalized skew-centrosymmetric matrices. The following result is the generalized skew-centrosymmetric analogue of Proposition 3.1.

PROPOSITION 4.1. *Suppose $A \in F^{n\times n}$ and $K \in F^{n\times n}$, where F is a field of characteristic not equal to 2 and K is an involutory matrix. If $AK = -KA$, then $\Lambda(A) = i\Lambda(KA)$.*

Proof. If $K = \pm I$, then A must be the zero matrix and the result clearly holds.

Assume $K \neq \pm I$ and let $X \in F^{n\times n}$ be the same matrix used to diagonalize the matrix K in the proof of Proposition 3.1:

$$K' = X^{-1}KX = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}.$$

Notationally, let the I block be $n_1 \times n_1$ and let the $-I$ block be $n_2 \times n_2$ where $n_1 + n_2 = n$. Proceeding as we did in Proposition 3.1, we can show that conjugation of A by X yields a matrix of the form

$$A' \equiv X^{-1}AX = \begin{pmatrix} 0 & A'_{12} \\ A'_{21} & 0 \end{pmatrix}$$

and that

$$K'A' = \begin{pmatrix} 0 & A'_{12} \\ -A'_{21} & 0 \end{pmatrix},$$

where we have the same block partitioning for these matrices as for K'.

Consider the case where $n_1 \geq n_2$. Using elementary row operations on the matrices $A' - \lambda I$ and $KA' - \lambda I$, we can construct the block upper triangular matrices

$$\begin{pmatrix} -\lambda I & A'_{12} \\ 0 & \frac{1}{\lambda}A'_{21}A'_{12} - \lambda I \end{pmatrix} \tag{7}$$

and

$$\begin{pmatrix} -\lambda I & A'_{12} \\ 0 & -\frac{1}{\lambda}A'_{21}A'_{12} - \lambda I \end{pmatrix}.$$

Taking determinants, we obtain the characteristic polynomials for $A' - \lambda I$ and $KA' - \lambda I$ as

$$(-1)^{n_1}\lambda^{n_1-n_2} \det\left(A'_{21}A'_{12} - \lambda^2 I\right) \tag{8}$$

and

$$(-1)^{n_1}\lambda^{n_1-n_2} \det\left(A'_{21}A'_{12} + \lambda^2 I\right), \tag{9}$$

respectively. From (8) and (9), the similarity of A to A', and the similarity of KA to $K'A'$, we conclude that $\lambda \in \Lambda(A)$ if and only if $i\lambda \in \Lambda(KA)$.

When $n_1 < n_2$, one can apply elementary row operations to the matrices $A' - \lambda I$ and $KA' - \lambda I$ to obtain block lower triangular matrices analogous to those in (7). Taking determinants, one obtains the characteristic polynomials

$$(-1)^{n_2}\lambda^{n_2-n_1} \det\left(A'_{12}A'_{21} - \lambda^2 I\right)$$

for $A' - \lambda I$ and

$$(-1)^{n_2}\lambda^{n_2-n_1} \det\left(A'_{12}A'_{21} + \lambda^2 I\right)$$

for $K'A' - \lambda I$. Again, we conclude that $\lambda \in \Lambda(A)$ if and only if $i\lambda \in \Lambda(KA)$. □

A skew-centrosymmetric example. Consider the matrices

$$A_3 = \begin{pmatrix} 2 & -1 & 1 & -1 & 0 \\ -1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & -1 & -1 \\ -1 & 0 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -2 \end{pmatrix}$$

and

$$JA_3 = \begin{pmatrix} 0 & 1 & -1 & 1 & -2 \\ -1 & 0 & -1 & -1 & 1 \\ 1 & 1 & 0 & -1 & -1 \\ -1 & 1 & 1 & 0 & 1 \\ 2 & -1 & 1 & -1 & 0 \end{pmatrix}.$$

A_3 is skew-centrosymmetric and, consequently, so is JA_3.

$$\Lambda(A_3) = \left\{-\sqrt{10}, -\sqrt{3}, 0, \sqrt{3}, \sqrt{10}\right\}$$

and

$$\Lambda(JA_3) = \left\{-\sqrt{10}i, -\sqrt{3}i, 0, \sqrt{3}i, \sqrt{10}i\right\}.$$

A generalized skew-centrosymmetric example. Let

$$A_4 = \begin{pmatrix} -3\sqrt{2} & -\sqrt{2} & 2 \\ -\sqrt{2} & \sqrt{2} & -2 \\ 2 & -2 & 2\sqrt{2} \end{pmatrix}$$

and

$$K = \begin{pmatrix} \frac{1}{2} & \frac{-1}{2} & \frac{\sqrt{2}}{2} \\ \frac{-1}{2} & \frac{1}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{pmatrix}.$$

Since $A_4K = -KA_4$ and $K^2 = I$, we say that A_4 is a generalized skew-centrosymmetric K-matrix.

$$\Lambda(A_4) = \left\{-2\sqrt{6}, 0, 2\sqrt{6}\right\}$$

and

$$\Lambda(KA_4) = \left\{-2\sqrt{6}i, 0, 2\sqrt{6}i\right\}$$

We end this article by establishing the real symmetric converse to Proposition 4.1.

PROPOSITION 4.2. *Suppose $A \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{n\times n}$ are symmetric, with $K^2 = I$. If $\Lambda(A) = i\Lambda(KA)$, then $AK = -KA$.*

Proof. We can assume that A is nonzero, as the proposition clearly holds when A is the zero matrix.

Since A is real symmetric and $\Lambda(A) = i\Lambda(KA)$, the eigenvalues of KA must be imaginary. As noted earlier, symmetry of A guarantees a full set of n independent eigenvectors. Let x be an eigenvector of the matrix KA corresponding to the eigenvalue $i\lambda$, where $\lambda \in \mathbb{R}$ has the largest magnitude of KA's eigenvalues. We shall write x as $u + iv$ and $\bar{x} = u - iv$, where u and v are real n-vectors so that $u = \frac{x+\bar{x}}{2}$ and $v = \frac{x-\bar{x}}{2i}$.

Since $KAx = i\lambda x$, we have that $KA\bar{x} = -i\lambda\bar{x}$. Therefore $KA(x + \bar{x}) = i\lambda(x - \bar{x})$ or, equivalently,

$$Au = -\lambda Kv. \tag{10}$$

Using the symmetry of A, we obtain

$$u^T AKv = (Au)^T Kv = (-\lambda Kv)^T Kv = -\lambda \|v\|^2 . \tag{11}$$

Similarly, the equation $KA(x - \bar{x}) = i\lambda(x + \bar{x})$ yields

$$Av = \lambda Ku, \tag{12}$$

and so

$$v^T AKu = (Av)^T Ku = (\lambda Ku)^T Ku = \lambda \|u\|^2 . \tag{13}$$

If $\|u\| < \|v\|$, then (11) implies that $\|AKv\| > |\lambda| \cdot \|v\|$, which is impossible due to the extremality of λ. A similar argument applied to (13) demonstrates that $\|u\| > \|v\|$ is impossible. Hence, $\|u\| = \|v\|$.

Applying the Cauchy–Schwarz inequality to (11), we have that

$$\|u\| \cdot \|AKv\| \geq \left\|u^T AKv\right\| = |\lambda| \cdot \|v\|^2 .$$

We may freely take $\|u\| = \|v\| = 1$ and therefore write $\|AKv\| \geq |\lambda|$. Again, because of the extremality of λ, this must in fact be an equality. Therefore, we have shown that

$$\|u\| \cdot \|AKv\| = \left|u^T (AKv)\right| .$$

Since Cauchy–Schwarz implies equality only if the vectors in question have the same direction up to sign, we have

$$AKv = \pm\lambda u. \tag{14}$$

The same argument applied to (13) shows that

$$AKu = \pm\lambda v. \tag{15}$$

Multiplication of A against (14) and (15) and then using (10) and (12) gives $A^2 Kv = \mp\lambda^2 Kv$ and $A^2 Ku = \pm\lambda^2 Ku$. As A^2 has only nonnegative eigenvalues, we can dispense with the sign ambiguities in (14) and (15):

$$AKv = -\lambda u, \tag{16}$$

$$AKu = \lambda v. \tag{17}$$

Utilizing the relations (10), (12), (16), and (17), we obtain

$$A(Ku + v) = \lambda(Ku + v),$$

$$A(Ku - v) = -\lambda(Ku - v),$$

$$A(Kv + u) = -\lambda(Kv + u),$$

$$A(Kv - u) = \lambda(Kv - u).$$

So, starting with two eigenvectors of KA ($x = u + iv$ and $\bar{x} = u - iv$) corresponding to $\pm i\lambda$, we have obtained four (not necessarily independent) eigenvectors of A corresponding to $\pm\lambda$. More manipulation with (16) and (17) will show that x and $\bar{x}$ also generate two additional eigenvectors of KA corresponding to $\pm i\lambda$:

$$KA(Kv + iKu) = i\lambda(Kv + iKu),$$

$$KA(Ku + iKv) = -i\lambda(Ku + iKv).$$

Note that the real span of the eigenvectors of A obtained from x and $\bar{x}$ is the same as the real span of the eigenvectors of KA also obtained from x and $\bar{x}$, namely, the vector space

$$T_0 \equiv span_{\mathbb{R}} \{u, v, Ku, Kv\}.$$

Let $y_1 \in span_{\mathbb{R}} \{u + v, u - v\}$ and $y_2 \in span_{\mathbb{R}} \{Ku + Kv, Ku - Kv\}$. Then using (10), (12), (16), and (17), it is easy to see that $(KA+AK)y_1 = \vec{0}$ and $(KA+AK)y_2 = \vec{0}$. In other words, $KA + AK$ maps the space T_0 to zero.

T_0 is clearly K invariant. Using the same method as in the proof of Proposition 3.5, we can show that $T_0^{\perp}$ is also K invariant. Therefore, we can apply the same argument as above to the space $T_0^{\perp}$ and continue doing so (as needed) to show that $KA + AK$ maps each of the (say m total) eigenspaces corresponding to A's nonzero eigenvalues to zero. The last of these repeated arguments shows that the eigenspace $T_m = T_{m-1}^{\perp}$ corresponding to A's 0 eigenvalues (if any) is K invariant, so KA and AK will both be zero when restricted to T_m. Therefore $KA + AK$ maps each of the invariant subspaces T_j associated with A's eigenvalues to zero. Since $\mathbb{R}^n = \oplus_{j=0}^{m} T_j$, we conclude that $KA = -AK$. □

Together, Proposition 4.1 and Proposition 4.2 yield the following characterization of real symmetric generalized skew-centrosymmetric matrices.

THEOREM 4.3. *Suppose $A \in \mathbb{R}^{n\times n}$ and $K \in \mathbb{R}^{n\times n}$ are symmetric, with $K^2 = I$. Then $AK = -KA$ if and only if $\Lambda(A) = i\Lambda(KA)$.*

COROLLARY 4.4. *Let $J \in \mathbb{R}^{n\times n}$ be the exchange matrix. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is skew-centrosymmetric if and only if $\Lambda(A) = i\Lambda(JA)$.*

Proof. Let $K = J$ in the statement of Theorem 4.3. □

REFERENCES

[1] A. AITKEN, *Determinants and Matrices*, Oliver and Boyd, Edinburgh, 1956.

[2] A. ANDREW, *Eigenvectors of certain matrices*, Linear Algebra Appl., 7 (1973), pp. 157–162.

[3] A. CANTONI AND P. BUTLER, *Eigenvalues and eigenvectors of symmetric centrosymmetric matrices*, Linear Algebra Appl., 13 (1976), pp. 275–288.

[4] A. COLLAR, *On centrosymmetric and centroskew matrices*, Quart. J. Mech. Appl. Math., 15 (1962), pp. 265–281.

[5] C. CURTIS, *Linear Algebra: An Introductory Approach*, Springer-Verlag, New York, 1984.

[6] L. DATTA AND S. MORGERA, *On the reducibility of centrosymmetric matrices—applications in engineering problems*, Circuits Systems Signal Process, 8 (1989), pp. 71–96.

[7] P. DELSARTE AND Y. GENIN, *Spectral properties of finite Toeplitz matrices*, in Proceedings of the 1983 International Symposium on Mathematical Theory of Networks and Systems, Beer-Sheva, Israel, 1983, Springer-Verlag, Berlin, New York, 1984, pp. 194–213.

[8] J. WEAVER, *Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors*, Amer. Math. Monthly, 92 (1985), pp. 711–717.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 3, pp. 896–902
© 2002 Society for Industrial and Applied Mathematics

NORMAL MATRICES AND THE COMPLETION PROBLEM*

SHMUEL FRIEDLAND†

In memory of Binyamin Schwarz

Abstract. We show that there exists a normal matrix with an arbitrary upper triangular part.

Key words. normal matrices, completion problem

AMS subject classification. 15A57

PII. S0895479801386444

1. Introduction. Let $\mathbf{M}(n,\mathbb{F})$ be the algebra of $n \times n$ matrices over a field $\mathbb{F}$, where $\mathbb{F}$ is either the field of real or complex numbers ($\mathbb{R}$ or $\mathbb{C}$, respectively). An inverse problem for matrices (IPFM) is the existence of a matrix A of a certain class belonging to a given variety $\mathcal{A} \subset \mathbf{M}(n,\mathbb{F})$ of dimension m. Most of the IPFMs are given by l polynomial conditions

$$p(A) = \alpha, \quad \alpha \in \mathbb{F}^l. \tag{1.1}$$

($p : \mathbf{M}(n,\mathbb{F}) \to \mathbb{F}^l$ is a polynomial map.) The most common IPFM are the inverse eigenvalue problems (IEPs). As an example of IPFM (IEP) recall the result of Farahat and Ledermann [FL]: For a matrix $A \in \mathbf{M}(n,\mathbb{C})$ whose top left-hand corner is nonderogatory, every characteristic polynomial can be achieved through the choice of entries in the last row and column. Note that we have $2n-1$ parameters from which to choose to satisfy n polynomial conditions. If $l = m$ and $p|\mathcal{A}$ is dominant (the m polynomial conditions are algebraically independent over $\mathcal{A}$), then IPFM (IEP) is called *tight* IPFM (IEP) and is denoted by TIPFM (TIEP). The most well known TIEP is determination of a real symmetric tridiagonal matrix from two spectra, which was first solved by Krein [GK, Appendix II] and rediscovered by many other authors. This problem, as well as the TIEP discussed in [Fr3], can be solved in a constructive way by reducing the problem to a similar problem in $\mathbf{M}(n-1,\mathbb{R})$.

Assume first that $\mathbb{F} = \mathbb{C}$. Let $\mathcal{A}$ be a complex irreducible affine variety of complex dimension at least l. If $p|\mathcal{A}$ is a dominant map, then (1.1) is solvable for all $\alpha \in \mathbb{C}^l \backslash W$ for some subvariety $W \subset \mathbb{C}^l$. (IPFM is generically solvable.) The dominance of polynomial maps related to the IEP is studied in [HRW] and [RW]. To show that (1.1) is solvable for every $\alpha \in \mathbb{C}^l$ ($W = \emptyset$) requires stronger tools, e.g., the intersection theory from the algebraic geometry [Fr2] or the degree theory [Fr1].

Assume now that $\mathbb{F} = \mathbb{R}$. Then the theory of dominant maps does not apply. That is, even though $p|\mathcal{A}$ is dominant there may be an open (semialgebraic set) of $\alpha \in \mathbb{R}^l$ for which (1.1) is not solvable; see, e.g., [Fr2]. In the known nonconstructive solution to the TIEP for real symmetric Toeplitz matrices, one uses the degree theory [Lan] (see also [Fr4]).

*Received by the editors March 15, 2001; accepted for publication (in revised form) by M. Chu August 22, 2001; published electronically March 5, 2002. This research was partially supported by SFB 343 "Diskrete Strukturen in der Mathematik," Universität Bielefeld, and a Lady Davis Visiting Professorship at Technion, Fall 2000.

http://www.siam.org/journals/simax/23-3/38644.html

†Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7045 (friedlan@uic.edu).

In this paper we solve a TIPFM over $\mathbb{R}$ which deals with normal complex-valued matrices with prescribed upper triangular entries. Let $\mathcal{N}_n \subset \mathbf{M}(n,\mathbb{C})$ be the real variety of normal matrices. It is known that $\mathcal{N}_n$ is an irreducible variety of real dimension n^2+n [Ik2]. The purpose of this paper is twofold. First, we describe in more detail the structure of $\mathcal{N}_n$, the manifold of its regular (smooth) points $\mathcal{N}_n^r$, and its covering space induced by the spectral decomposition of a normal matrix. Second, using the degree theory, we show that any upper triangular $A \in \mathbf{M}(n,\mathbb{C})$ can be completed to a normal matrix $X \in \mathbf{M}(n,\mathbb{C})$. That is, for any $\binom{n+1}{2}$ complex numbers $a_{ij},\ i=1,\dots,n,\ j=i,\dots,n$, there exists $X=(x_{ij})_1^n \in \mathcal{N}_n$ such that $x_{ij}=a_{ij}$ for $i=1,\dots,n,\ j=i,\dots,n$. The cases $n=2,3$ were settled by Ikramov [Ik1]. The completion problem was raised by Elsner.

2. Remarks on the variety of normal matrices. Let $\mathbb{U}_n$ be the group of $n\times n$ unitary matrices. As $\mathbb{U}_n$ is a connected Lie group it follows that $\mathbb{U}_n$ is parallelizable and hence orientable. The symmetric group $\mathcal{S}_n$ acts on $\mathbb{U}_n$ by permuting the columns of $U \in \mathbb{U}_n$.

LEMMA 2.1. *The action of $\mathcal{S}_n$ on $\mathbb{U}_n$ preserves the orientation of $\mathbb{U}_n$.*

Proof. We identify the tangent space of $\mathbb{U}_n$ with the Lie algebra of skew Hermitian matrices:

$$\mathbb{A}_n := \{A=(a_{ij})_1^n \in \mathbf{M}(n,\mathbb{C}): \quad A^*=-A\}.$$

Fix a frame (basis) in $\mathbb{A}_n$ which is given by the unit vectors in the positive direction of the imaginary parts of the diagonal elements of A and the real and imaginary parts of the entries $a_{ij},\ 1\le i<j\le n$. We view this frame as $\mathbb{R}^n \times \mathbb{C}^{\frac{n(n-1)}{2}}$. Orient this frame by giving the standard orientations on $\mathbb{R}^n$ and $\mathbb{C}^{\frac{n(n-1)}{2}}$. The n^2 vectors fields of $\mathbb{U}_n$ at U are given by transporting the frame of $\mathbb{A}_n$, at the identity I_n, by the left multiplication by U. Let $\Pi_n \subset \mathbb{U}_n$ be the subgroup of permutation matrices. We identify $\mathcal{S}_n$ with Π_n. Then Π_n acts on $\mathbb{U}_n$ by the right multiplication:

$$U \mapsto UP, \quad P \in \Pi_n,\ U \in \mathbb{U}_n. \tag{2.1}$$

Observe next that

$$e^A P = P e^{P^T A P}, \quad P \in \Pi_n,\ A \in \mathbb{A}_n.$$

Thus the transformation (2.1) preserves the orientation iff the transformation

$$A \mapsto P^T A P, \quad P \in \Pi_n,\ A \in \mathbb{A}_n, \tag{2.2}$$

preserves the orientation of $\mathbb{A}_n$. As any element of $\mathcal{S}_n$ is a product of transpositions, it is enough to show that (2.2) preserves the orientation for any transposition P. Assume for simplicity that P corresponds to the transposition which transposes 1 and 2. Then the map (2.2) acts separately on $\mathbb{R}^n$ and $\mathbb{C}^{\frac{n(n-1)}{2}}$. On $\mathbb{R}^n$ it transposes a_{11} and a_{22}. Hence it reverses the orientation on $\mathbb{R}^n$. On $\mathbb{C}^{\frac{n(n-1)}{2}}$ the map (2.2) is a composition of the map $a_{12} \mapsto -\overline{a}_{12}$ and the permutations $a_{1j} \leftrightarrow a_{2j},\ j=3,\dots,n$, for $n>2$. Clearly, the above permutations preserve the orientation of $\mathbb{C}^{\frac{n(n-1)}{2}}$. The map $a_{12} \mapsto -\overline{a}_{12}$ reverses the orientation of $\mathbb{C}^{\frac{n(n-1)}{2}}$. Hence the action of the above transposition preserves the orientation of $\mathbb{U}_n$. The same arguments apply to any transposition P. ∎

Recall that $\mathbb{U}_n$ is a smooth compact real algebraic variety

$$\mathbb{U}_n = \{X + \sqrt{-1}Y : \quad XX^T + YY^T = I_n,\ -XY^T + YX^T = 0,\ X, Y \in \mathbf{M}(n, \mathbb{R})\}.$$

$\mathcal{S}_n$ acts freely on $\mathbb{U}_n$. Hence $\mathbb{U}_n/\Pi_n$ can be realized as a smooth compact real algebraic variety.

COROLLARY 2.2. *$\mathbb{U}_n/\Pi_n$ is orientable.*

Let $\mathcal{F}_n$ be the space of full flags in $\mathbb{C}^n$:

$$V_1 \subset V_2 \subset \cdots \subset V_n = \mathbb{C}^n. \tag{2.3}$$

Let $\mathbb{UD}_n < \mathbb{U}_n$ be the subgroup of diagonal unitary matrices. Then $\mathcal{F}_n$ is isomorphic to the homogeneous space $\mathbb{U}_n/\mathbb{UD}_n$. Indeed, to each coset $U\mathbb{UD}_n$ associate the following flag: V_k is the subspace spanned by the first k columns of U for $k = 1, \ldots, n$. Clearly, this flag is independent of the representative $U' \in U\mathbb{UD}_n$. Vice versa, assume that $\mathbb{C}^n$ is an inner product space with $\langle x, y\rangle = y^*x$. Then a flag (2.3) defines an orthonormal basis of column vectors $v_1, \ldots, v_n \in \mathbb{C}^n$ such that $V_k = \text{span}\,(v_1, \ldots, v_k)$ for $k = 1, \ldots, n$. Let $U = (v_1, \ldots, v_n)$. Then U defines a unique coset $U\mathbb{UD}_n$. As $P\mathbb{UD}_n = \mathbb{UD}_nP$ for any $P \in \Pi_n$ it follows that $\mathcal{S}_n$ acts on $\mathbb{U}_n/\mathbb{UD}_n$:

$$U\mathbb{UD}_n \mapsto UP\mathbb{UD}_n, \quad P \in \Pi_n.$$

Equivalently $\mathcal{S}_n$ acts on $\mathcal{F}_n$ by permuting the orthonormal basis $v_1, \ldots, v_n$ corresponding to the flag (2.3):

$$\{v_1, \ldots, v_n\} \mapsto \{v_{\sigma(1)}, \ldots, v_{\sigma(n)}\}, \quad \sigma \in \mathcal{S}_n.$$

The following lemma is probably well known and we present its proof for completeness.

LEMMA 2.3. *$\mathcal{F}_n$ is a fiber bundle with the basis $\mathbb{P}^{n-1}$ and the fiber $\mathcal{F}_{n-1}$. Hence $\mathcal{F}_n$ is an orientable manifold of dimension $n(n-1)$. The action of $\mathcal{S}_n$ on $\mathcal{F}_n$ does not preserve the orientation of $\mathcal{F}_n$.*

Proof. The choice of v_1 in the orthonormal basis $\{v_1, \ldots, v_n\}$ is equivalent to the choice of a one-dimensional subspace in $\mathbb{C}^n$ spanned by v_1. The space of one-dimensional subspaces is $\mathbb{P}^{n-1}$. Fix one-dimensional subspace $V_1 \subset \mathbb{C}^n$. Let $V_1^\perp$ be the orthogonal complement of V_1 in $\mathbb{C}^n$. Then the choices of all possible orthonormal bases in $V_1^\perp$ is $\mathcal{F}_{n-1}$. Hence $\mathcal{F}_n$ is a fiber bundle over $\mathbb{P}^{n-1}$ with the fiber $\mathcal{F}_{n-1}$. ($\mathcal{F}_1$ is a space consisting of one point.) We prove the rest of the lemma by induction. Let $n = 2$. Then $\mathcal{F}_2$ is identified with the Riemann sphere $\mathbb{P}$. Hence $\mathcal{F}_2$ is orientable of (real) dimension 2 . Let $V_1 \subset \mathbb{C}^2$ be a one-dimensional subspace which is not spanned either by $e_1 = (1, 0)^T$ or by $e_2 = (0, 1)^T$. Then

$$V_1 = \text{span}\,((1, z)^T), \quad V_1^\perp = \text{span}\,\left(\left(1, -\frac{1}{\overline{z}}\right)^T\right), \quad z \in \mathbb{C}\backslash\{0\}.$$

The action of the permutation $P \in \Pi_2$ is equivalent to the map $V_1 \mapsto V_1^T$. For V_1 of the above form this action is given by the map $z \mapsto -\frac{1}{\overline{z}}$. Clearly this map reverses the orientation of $\mathbb{P}$.

Assume by induction that $\mathcal{F}_n$ is orientable of dimension $n(n-1)$ and the action of $\mathcal{S}_n$ does not preserve the orientation of $\mathcal{F}_n$. Consider $\mathcal{F}_{n+1}$, which is a fiber bundle over $\mathbb{P}^n$ with the fiber $\mathcal{F}_n$. Clearly

$$\dim \mathcal{F}_{n+1} = \dim \mathbb{P}^n + \dim \mathcal{F}_n = 2n + n(n-1) = (n+1)n.$$

As $\mathbb{P}^n$ and $\mathcal{F}_n$ are orientable it follows directly that $\mathcal{F}_{n+1}$ is orientable. Fix $V_2 \subset V_3 \subset \cdots \subset V_{n+1} = \mathbb{C}^{n+1}$. Without loss of generality we identify V_2 with $\mathbb{C}^2$. Consider all possible choices of a one-dimensional subspace $V_1 \subset V_2$. Let $P \in \Pi_{n+1}$ be the transposition which transposes 1 and 2. Then the above arguments show that the action of P reverses the orientation of $\mathcal{F}_{n+1}$. ∎

Recall that

$$\mathcal{N}_n := \{A \in \mathbf{M}(n, \mathbb{C}) : \quad AA^* = A^*A\}.$$

By writing $A = X + \sqrt{-1}Y,\ X, Y \in \mathbf{M}(n, \mathbb{R})$ we obtain that

$$\begin{aligned}\mathcal{N}_n = \{A = X + \sqrt{-1}Y : XX^T + YY^T = X^TX + Y^TY,\\ -XY^T + YX^T = X^TY - Y^TX, \quad X, Y \in \mathbf{M}(n, \mathbb{R})\}.\end{aligned} \tag{2.4}$$

Let $\mathbb{D}(n, \mathbb{C})$ be the set of complex diagonal matrices. Let

$$\mathbb{D}^r(n, \mathbb{C}) = \{D = \operatorname{diag}(d_1, \ldots, d_n) \in \mathbb{D}(n, \mathbb{C}), \quad d_i \neq d_j \quad \text{for } i \neq j\}.$$

Recall that any normal A has the form UDU^*, where $D \in \mathbb{D}(n, \mathbb{C})$ and $U \in \mathbb{U}_n$. As $UDU^* = D$ for any $U \in \mathbb{UD}_n$ it follows that

$$\mathcal{N}_n = \{A : \quad A = UDU^*, \quad D \in \mathbb{D}(n, \mathbb{C}),\ U \in \mathbb{U}_n/\mathbb{UD}_n\}. \tag{2.5}$$

Some of the following results are stated explicitly in [Ik2].

LEMMA 2.4. *The set $\mathcal{N}_n$ is an irreducible real homogeneous algebraic variety of dimension $n(n+1)$. The set of smooth points $\mathcal{N}_n^r$ of $\mathcal{N}_n$ corresponds to all normal matrices with pairwise distinct eigenvalues. $\mathcal{N}_n^r$ is connected. The natural projection $p : \mathbb{D}(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n \to \mathcal{N}_n$ is a branched covering. The symmetric group $\mathcal{S}_n$ acts on $\mathbb{D}(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n$ by the permutation of the diagonal on the first factor and by column permutation on the second factor (simultaneously). This action commutes with p. Furthermore, for any $A \in \mathcal{N}_n^r$ $p^{-1}(A)$ is an $\mathcal{S}_n$-orbit consisting of exactly $n!$ elements.*

Proof. Clearly $p^{-1}(A)$ is a finite set iff A has n distinct eigenvalues. Hence $\mathcal{N}_n^r = p(\mathbb{D}^r(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n)$. Furthermore

$$p : \mathbb{D}^r(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n \to \mathcal{N}_n^r \tag{2.6}$$

is a covering map. Observe that $\mathbb{D}(n, \mathbb{C}) \backslash \mathbb{D}^r(n, \mathbb{C})$ is a complex algebraic subvariety. Hence $\mathbb{D}^r(n, \mathbb{C})$ and $\mathbb{D}^r(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n$ are connected sets. Hence $\mathcal{N}_n^r$ is a connected manifold. Therefore $\mathcal{N}_n$ is an irreducible variety. Let $A \in \mathcal{N}_n^r$. Then $p^{-1}(A)$ is an orbit of $\mathcal{S}_n$. The dimension of $\mathcal{N}_n$ is equal to the dimension of $\mathcal{N}_n^r$. As $\mathbb{D}^r(n, \mathbb{C}) \times \mathbb{U}_n/\mathbb{UD}_n$ is a finite cover of $\mathcal{N}_n^r$,

$$\dim \mathcal{N}_n^r = \dim \mathbb{D}^r(n, \mathbb{C}) + \dim \mathbb{U}_n/\mathbb{UD}_n = 2n + \dim \mathcal{F}_n = 2n + n(n-1) = n(n+1). \quad ∎$$

Recall Ikramov's result [Ik1] that $\mathcal{N}_n^r \cap \mathbf{M}(n, \mathbb{R})$ is not connected for $n \geq 2$. One can deduce this result by showing that there is no path in $\mathcal{N}_n^r \cap \mathbf{M}(n, \mathbb{R})$ connecting any symmetric matrix with any skew symmetric matrix.

LEMMA 2.5. *$\mathcal{N}_n^r$ is not orientable for $n \geq 2$.*

Proof. The action of $\mathcal{S}_n$ preserves the orientation of $\mathbb{D}^r(n, \mathbb{C})$. Lemma 2.3 claims that the action of $\mathcal{S}_n$ on $\mathbb{U}_n/\mathbb{UD}_n$ does not preserve the orientation of $\mathbb{U}_n/\mathbb{UD}_n$. Use Lemma 2.4 to deduce that $\mathcal{N}_n^r$ is not orientable. ∎

In what follows we need to consider compact models of $\mathcal{N}_n$. The first way is to consider the real projective variety $\mathcal{PN}_n \subset \mathbb{PR}^{2n^2-1}$, by identifying a line through the origin $L \subset \mathcal{N}_n$ with a point $\lambda \in \mathcal{PN}_n$. That is, $\mathcal{PN}_n$ is defined by the equations (2.4), where $X = (x_{ij})_1^n,\ Y = (y_{ij})_1^n$ are viewed as the homogeneous coordinates in the projective space $\mathbb{PR}^{2n^2-1}$. The second way is by considering the variety of normal matrices with the Frobenius norm equal to 1:

$$\mathcal{N}_{n,2} := \{A \in \mathcal{N}_n : \quad \text{trace } (AA^*) = 1\}.$$

Note that $\mathcal{N}_{n,2}$ is a double cover of $\mathcal{PN}_n$ obtained by identifying the points A and $-A$. Lemma 2.4 implies that the varieties $\mathcal{N}_{n,2},\ \mathcal{PN}_n$ have singular points. It is of interest to find a CW decomposition of these varieties, and in particular the homology groups of $\mathcal{N}_{n,2}$ and $\mathcal{PN}_n$. A useful observation that can help solve these problems is

$$p^{-1}(\mathcal{N}_{n,2}) = S^{2n-1} \times \mathbb{U}_n/\mathbb{UD}_n.$$

Here we identified the variety of all $n \times n$ complex diagonal matrices of Frobenius norm 1 with the $(2n-1)$-dimensional sphere S^{2n-1}. Hence $\mathcal{N}_{n,2}$ and $\mathcal{PN}_n$ are irreducible varieties, and the quasi-variety of its smooth points are connected nonorientable manifolds.

3. Main result.

THEOREM 3.1. *Let $a_{ij},\ i = 1, \dots, n,\ j = i, \dots, n$, be any set of $\frac{n(n+1)}{2}$ complex numbers. Then there exists a normal matrix $X = (x_{ij})_1^n \in \mathcal{N}_n$ such that $x_{ij} = a_{ij},\ i = 1, \dots, n,\ j = i, \dots, n$.*

Proof. Let $\tau : \mathcal{N}_n \to \mathbb{C}^{\frac{n(n+1)}{2}}$ be the projection

$$\tau((x_{ij})_1^n) = (x_{11}, \dots, x_{1n}, x_{22}, \dots, x_{2n}, \dots, x_{nn}).$$

We claim that

$$\tau^{-1}(\tau(D)) = D \quad \text{for all} \quad D \in \mathbb{D}(n, \mathbb{C}). \tag{3.1}$$

Suppose that

$$\tau(X) = \tau(D), \quad X = (x_{ij})_1^n,\ D = \text{diag } (d_1, \dots, d_n).$$

Then X is a lower triangular matrix with $x_{ii} = d_i,\ i = 1, \dots, n$. Hence $d_1, \dots, d_n$ are the eigenvalues of X. Since X is a lower triangular normal matrix it follows that

$$\sum_{i=1}^n |d_i|^2 = \text{trace } (XX^*) = \sum_{1 \le j \le i \le n} |x_{ij}|^2.$$

Hence $X = D$. In particular $\tau^{-1}(0) = 0$.

Let $\hat{\tau} : \mathcal{N}_{n,2} \to S^{n(n+1)-1}$ be given by

$$\hat{\tau}(X) = \frac{\tau(X)}{||\tau(X)||_2}, \quad X \in \mathcal{N}_{n,2}.$$

Here by $||x||_2$ we denote the l_2 norm of a vector $x \in \mathbb{C}^m$. Since $\mathcal{N}_n$ is real homogeneous variety our theorem is equivalent to the statement that $\hat{\tau}$ is onto map. To show that $\hat{\tau}$ is onto, we apply the degree theory. The standard mod 2 degree theory considers

a continuous map $\tilde{\tau} : \tilde{\mathcal{N}}_{n,2} \to S^{n(n+1)-1}$, where $\tilde{\mathcal{N}}_{n,2}$ is a smooth manifold. Recall that $\mathcal{N}_{n,2}$ is a real variety given by (2.4) and the equation trace $(XX^T + YY^T) = 1$. Hironaka's resolution of singularities [GH] implies that it is possible to blow up the singularities of $\mathcal{N}_{n,2}$ to obtain a compact smooth algebraic variety $\tilde{\mathcal{N}}_{n,2}$. That is, there is a projection $\pi : \tilde{\mathcal{N}}_{n,2} \to \mathcal{N}_{n,2}$ with the following properties. Let $\mathcal{N}^r_{n,2} := \mathcal{N}_{n,2} \cap \mathcal{N}^r_n$ be the quasi-variety of smooth points of $\mathcal{N}_{n,2}$. Then

$$\pi : \pi^{-1}(\mathcal{N}^r_{n,2}) \to \mathcal{N}^r_{n,2}$$

is a diffeomorphism. Furthermore, $\tilde{\mathcal{N}}_{n,2} \backslash \pi^{-1}(\mathcal{N}^r_{n,2})$ is a strict subvariety of $\tilde{\mathcal{N}}^r_{n,2}$. Set

$$\tilde{\tau} = \hat{\tau} \circ \pi : \tilde{\mathcal{N}}^r_{n,2} \to S^{n(n+1)-1}.$$

Let $D \in \mathbb{D}^r(n, \mathbb{C})$ and assume that D has the Frobenius norm 1. Then $\tau(D) \in S^{n(n+1)-1}$. Equation (3.1) yields that $\hat{\tau}^{-1}(\tau(D)) = D$. Since $D \in \mathcal{N}^r_{n,2}$ it follows that $\pi^{-1}(D)$ consists of one point. We claim that $\hat{\tau}|\mathcal{N}^r_{n,2}$ is an immersion at any $D = \operatorname{diag}(d_1, \dots, d_n) \in \mathbb{D}^r(n, \mathbb{C}) \cap \mathcal{N}^r_{n,2}$.

Indeed, let $A \in \mathcal{N}_n$ be a normal matrix close to D. Then A has eigenvalues $d_1 + z_1, \dots, d_n + z_n$ such that $d_i + z_i \neq d_j + z_j$ for $i \neq j$. Hence $A \in \mathcal{N}^r_n$. Let $Z := \operatorname{diag}(z_1, \dots, z_n)$. Then $A = U(D+Z)U^*$, $U \in \mathbb{U}_n$. Recall that $U = e^{-V}$, $V^* = -V \in \mathbf{M}(n, \mathbb{C})$. Let $V = (v_{ij})_1^n$. Hence the tangent space of $\mathcal{N}^r_n$ at D is given by the matrices of the form

$$A = D + Z + DV - VD = \operatorname{diag}(d_1 + z_1, \dots, d_n + z_n) + (v_{ij}(d_i - d_j))_{i,j=1}^n,$$
$$v_{11} = \cdots = v_{nn} = 0, \quad v_{ji} = -\overline{v}_{ij} \text{ for } j > i. \tag{3.2}$$

Here the $n(n+1)$ real coordinates of the tangent space are the real and the imaginary parts of $z_1, \dots, z_n$ and v_{ij} for $j > i$. Hence the tangent space of $\mathcal{N}^r_{n,2}$ at D is given by the matrices of the form (3.2) with an additional condition

$$\sum_{i=1}^n \operatorname{Re} z_i \overline{d}_i = 0.$$

Therefore $\hat{\tau}|\mathcal{N}^r_{n,2}$ is an immersion at D. This implies that the local degree of $\hat{\tau}|\mathcal{N}^r_{n,2}$ is ± 1. Hence the mod 2 degree of $\tilde{\tau}$ is 1 [Mil]. In particular $\tilde{\tau}$ is onto. Hence $\hat{\tau}$ is onto. □

REFERENCES

[FL] H. K. Farahat and W. Ledermann, *Matrices with prescribed characteristic polynomial*, Proc. Edinburgh Math. Soc., 11 (1958), pp. 143–146.

[Fr1] S. Friedland, *On inverse multiplicative eigenvalue problems for matrices*, Linear Algebra Appl., 12 (1975), pp. 127–137.

[Fr2] S. Friedland, *Inverse eigenvalue problems*, Linear Algebra Appl., 17 (1977), pp. 15–51.

[Fr3] S. Friedland, *The reconstruction of a symmetric matrix from the spectral data*, J. Math. Anal. Appl., 71 (1979), pp. 412–422.

[Fr4] S. Friedland, *Inverse eigenvalue problems for symmetric Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1142–1153.

[GK] F. R. Gantmacher and M. G. Krein, *Oszillationsmatrizen, Oszillationskerne und kleine Schwingungen mechanischer Systeme*, Akademie Verlag, Berlin, 1960.

[GH] P. Griffiths and J. Harris, *Principles of Algebraic Geometry*, Wiley, New York, 1978.

[HRW] W. Helton, J. Rosenthal, and X. Wang, *Matrix extensions and eigenvalue completions, the generic case*, Trans. Amer. Math. Soc., 349 (1997), pp. 3401–3408.

[Ik1] Kh. D. Ikramov, *Normal dilation of triangular matrices*, Mat. Zametki, 60 (1996), pp. 861–872.

[Ik2] Kh. D. Ikramov, *The dimension of the variety of normal matrices*, Comput. Math. Math. Phys., 38 (1998), pp. 1–6.

[Lan] H. J. Landau, *The inverse eigenvalue problem for real symmetric Toeplitz matrices*, J. Amer. Math. Soc., 7 (1994), pp. 749–767.

[Mil] J. Milnor, *Topology from the Differential Viewpoint*, University Press of Virginia, Charlottesville, VA, 1965.

[RW] J. Rosenthal and X. Wang, *Eigenvalue completion by affine varieties*, Proc. Amer. Math. Soc., 128 (1999), pp. 643–646.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 903–915
© 2002 Society for Industrial and Applied Mathematics

A NOTE ON THE LDL^T DECOMPOSITION OF MATRICES FROM SADDLE-POINT PROBLEMS*

MIROSLAV TŮMA†

Abstract. Sparse linear systems $Kx = b$ are considered, where K is a specially structured symmetric indefinite matrix. These systems arise frequently, e.g., from mixed finite element discretizations of PDE problems. The LDL^T factorization of K with diagonal D and unit lower triangular L is known to exist for natural ordering of K, but the resulting triangular factors can be rather dense. On the other hand, for a given permutation matrix P, the LDL^T factorization of P^TKP may not exist.

In this paper a new way to obtain a fill-in minimizing permutation based on initial fill-in minimizing ordering is introduced. For an important subclass of matrices arising from mixed and hybrid finite element discretizations, the existence of the LDL^T factorization of the permuted matrix is proved. Experimental results on practical problems indicate that the amount of computational savings can be substantial when compared with the approach based on Schur complement.

Key words. systems of sparse linear algebraic equations, symmetric indefinite systems, direct methods, LDL^T decomposition, supernodal solvers

AMS subject classifications. 65F10, 65F35, 65F50, 65Y05

PII. S0895479897321088

1. Introduction.

We consider the solution of symmetric linear systems of the form

$$Kx = b, \tag{1.1}$$

where the coefficient matrix $K \in \mathbb{R}^{(n+m)\times(n+m)}, n \geq m$, has the following block form:

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}, \tag{1.2}$$

where matrices $H \in \mathbb{R}^{n\times n}$ and $A \in \mathbb{R}^{m\times n}$ are large and sparse and H is symmetric and positive definite. Clearly, K is indefinite. We always assume in this paper that K is irreducible (being a technical assumption only) and also nonsingular, i.e., A^T has full column rank. Later in this paper, such matrices will be called *saddle-point* matrices.

If a sparse symmetric matrix K is factorized into LDL^T, it is likely that *fill-in* occurs, i.e., the number of nonzero entries in the factors is usually greater than the number of nonzeros in K. It is typically cheaper to factorize a permuted matrix $\bar{K} = PKP^T$ into $\bar{L}^T\bar{D}\bar{L}$ where P is chosen to reduce fill-in. The matrix K is not strongly factorizable in general. That is, not all the permutation matrices P provide LDL^T-factorizable matrices $\bar{K}$.

In recent years, a lot of work has been done on the stable decomposition of sparse symmetric *general indefinite* matrices and the efficient application of the decomposition to some practical problems (see [7], [8], [12], [23], [9]). The methods have been

*Received by the editors May 5, 1997; accepted for publication (in revised form) October 19, 2001; published electronically March 13, 2002. A part of the work was done while the author was visiting the University of Bergamo. This work was supported by a CNR Fellowship for young scientists, grant GA AS CR A1030103, and grant GA CR 101/00/1035.

http://www.siam.org/journals/simax/23-4/32108.html

†Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8 - Libeň, Czech Republic (tuma@cs.cas.cz).

based almost exclusively on the Bunch–Parlett factorization. In many situations, however, it is possible to make use of additional properties enjoyed by the saddle-point matrices. For example, we may use the fact that the left upper block H in (1.2) is positive definite. In this case the LDL^T decomposition of (1.2) can be computed by exploiting all the enhancements of supernodal Cholesky solvers. This approach seems to be a viable alternative to other approaches (see, e.g., [24], [22]). Let us mention that the LDL^T factorization applied to matrices with the same block structure arising in optimization was used in [24]. The approach is based on the notion of *tiers*, using the block structure of the system, rather than on structural properties of the symmetric Gaussian elimination. Zero diagonal elements are deferred until the end of the block phase by some static ordering. Therefore, this strategy is a variant of the block Schur complement approach.

The approach proposed in this paper is based on the LDL^T decomposition of the permuted matrix K. The reordering is done in two steps. First, we construct an initial fill-in minimizing permutation of the whole matrix. Second, we strive to modify the permutation in order to obtain a P such that P^TKP is factorizable. The process strongly relies on the structural properties of the symmetric Gaussian elimination. For an important subclass of matrices from mixed-hybrid finite element discretizations of the potential fluid flow problem we prove factorizability of permuted matrices. Though we are not able to prove formally the same result for more general systems with saddle-point matrices, we believe that the presented tools can be often successfully used even in these cases. Experimental results that compare the new strategy with the straightforward approach (using fill-in minimizing permutation for the first n rows and columns and then for the Schur complement) show the superiority of our strategy.

The outline of this paper is as follows. In section 2 we present an example showing the importance of mixing variables from both blocks of (1.2) in the fill-in reducing ordering of K. We will establish some factorizability conditions on saddle-point matrices in section 3. We also describe the class of $\mathcal{F}$-matrices which arise in mixed and hybrid finite element applications. For this class we propose an algorithm which determines a permutation matrix P in such a way that the factorizability of P^TKP is guaranteed. Experimental results in section 4 demonstrate the effectiveness of the adopted approach. We assume that the reader is familiar with the basic graph-theoretic terminology used in sparse decompositions (see [6], [10], [18], [1]).

2. Permutations and fill-in for saddle-point matrices. The common approach in many applications with saddle-point matrices is to reduce the matrix to the Schur complement and then to make use of its definiteness. (Schur complement $S = -AH^{-1}A^T$ of the matrix (1.2) is negative definite.) Unfortunately, S can become rather dense.

Consider the example in Figure 1. A structure of matrix K is depicted on the left. Let the block H be determined by its first 6 rows and columns. The Schur complement S is completely dense. On the right we have a structure where the 7th row and column were moved just after the 1st row and column, respectively. Then the principal submatrix of the permuted matrix determined by the last 10 rows and columns has the same pattern of nonzeros as K. Consequently, repeating this row and column exchange recursively we get a matrix with small fill-in in its LDL^T decomposition.

In this case, the recursive formation of the permutation resembles the choice of tile pivots in the ANALYZE phase of the MA47 code (see [8]).

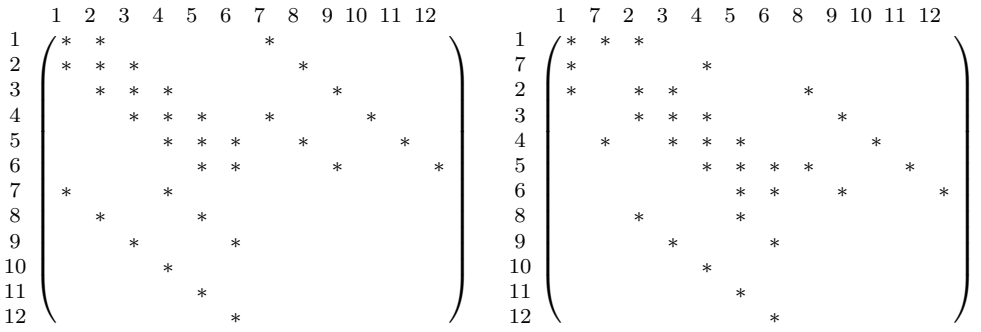


FIG. 1. *Matrix K providing a dense Schur complement and a sketch of the construction of its permutation inducing less fill-in in the LDL^T decomposition.*

3. LDL^T-factorizability of saddle-point matrices. If the right lower block of the matrix K is negative definite, then any symmetric permutation of K provides a factorizable matrix (see [24], [25]). However, small diagonal elements often restrict the set of theoretically feasible permutations to some subset in practical applications. Here we consider the problem of permutations from a different point of view. Given a saddle-point matrix K, we ask for the permutation matrices P such that P^TKP is factorizable. Such permutations (orderings) will be called *feasible*.

First, we introduce some notation. A given sparse symmetric matrix Q can be structurally represented by its *associated undirected graph* $G(Q) = (V(Q), E(Q))$, where nodes in $V(Q)$ correspond to rows and columns of Q and edges in $E(Q)$ correspond to offdiagonal nonzero entries in Q. This graph is usually used to find the initial node ordering. (In our experiments in section 4, we have used the multiple minimum degree (MMD) algorithm; see [11].)

Consider the partition of the vertices $V(Q)$ of $G(Q)$: $V(Q) = R(Q) \cup C(Q), R(Q) \cap C(Q) = \emptyset$. The submatrix $Q(R(Q), C(Q))$ of Q can be represented by its *associated bipartite graph* $B(R(Q), C(Q))$ where the two sets of nodes correspond to $R(Q)$ and $C(Q)$, respectively, and edges correspond to nonzero elements in Q in rows from $R(Q)$ and columns from $C(Q)$. This graph will be used to describe the structure of the offdiagonal block of the matrix (1.2). Let $G(K) = (V(K), E(K))$ be the associated undirected graph of the saddle-point matrix (1.2). Denote by $R \subset V(K)$ the set of nodes $\{1, \ldots, n\}$ corresponding to the first n rows of K and $C \subset V(K)$ the set of nodes $\{n+1, \ldots, n+m\}$ corresponding to the last m columns of K. This notation for the initial partition of vertices will be used throughout the paper. In our strategy, starting from K we transform it by a sequence of permutations. In general, the intermediate permuted matrices will be denoted by $\bar{K}$. If it is necessary to distinguish between the input and output matrices in some algorithm we will denote the output matrix in the new ordering by $\bar{\bar{K}}$. Let $T = T(\bar{K})$ denote an elimination tree of $\bar{K}$, i.e., the graph with n nodes $\{1, \ldots, n\}$ and with the arcs $(j, parent(j))$ where $parent(j) = \min\{i > j | l_{ij} \neq 0\}$. Note that that the elimination tree is well defined for any symmetric definite or indefinite matrix based on its structure completed by possibly missing diagonal entries. We use $T[i]$ to denote a subtree of T rooted in i. (We shall use $T[i]$ to denote both the subtree itself and the set of nodes in this subtree.) A simple necessary condition for a matrix to be factorizable can be described using the notion of an elimination tree.

PROPOSITION 3.1. *Matrix $\bar{K}$ is factorizable only if the leaves of the elimination*

tree $T(\bar{K})$ belong to R.

Proof. If some leaf i of $T(\bar{K})$ is not from R, then $a_{ii} = 0$ and $\bar{K}$ is not factorizable. □

Before we proceed further, we will show how to transform an input matrix $\bar{K}$ (K after some initial ordering) which may have some leaves of $T(\bar{K})$ in C to the permuted matrix $\bar{\bar{K}}$ satisfying the necessary condition from Proposition 3.1. The fact that the nodes of C form an independent set of $G(\bar{K})$ implies the following result.

LEMMA 3.1. *If $k \in C$ is a leaf of the elimination tree $T(\bar{K})$, then parent$(k) \notin C$.*

Therefore, if a node j has in $T(\bar{K})$ one or more leaves from C, then j is from R. Denote the leaves of j that belong to C by $k_1, \ldots, k_\omega$. It is well known that there exists a reordering of $\bar{K}$ (corresponding to a different topological numbering of $T(\bar{K})$) that provides the same fill-in in the factors as $\bar{K}$ and that orders the nodes $k_1, \ldots, k_\omega$ consecutively after all the other subtrees rooted in the children of j (see [18]). We assume that our initial ordering satisfies this property. Moreover, we assume that it is a postordering of the elimination tree, i.e., all the subtrees of $T(\bar{K})$ have their elements numbered consecutively. This fact will be used later. Denote by T_a the subgraph of the tree $T[j]$ given by $T[j]\backslash\{k_1, \ldots, k_\omega, j\}$. We need to reorder the set of nodes $\{j, k_1, \ldots, k_\omega\}$ so that $\bar{K}$ satisfies the necessary condition from Proposition 3.1. Namely, the elimination of the leaves $\{k_1, \ldots, k_\omega, j\}$ has to be delayed after the elimination of j and the new mutual order of the former leaves should be determined. The impact of delayed elimination on the size of fill-in was studied by Liu in [17]. In our special case of node delays we specify the ordering sequence of $\{j, k_1, \ldots, k_\omega\}$ directly. First j must be chosen. Then we take the nodes $k_i, i = 1, \ldots, \omega$, sequentially using locally the minimum degree strategy. In this way, we get the sequence $(j, \tilde{k}_1, \ldots, \tilde{k}_\omega)$. The situation is depicted in Figure 2. A "double-edge" is used to indicate a chain of nodes numbered consecutively in the tree.

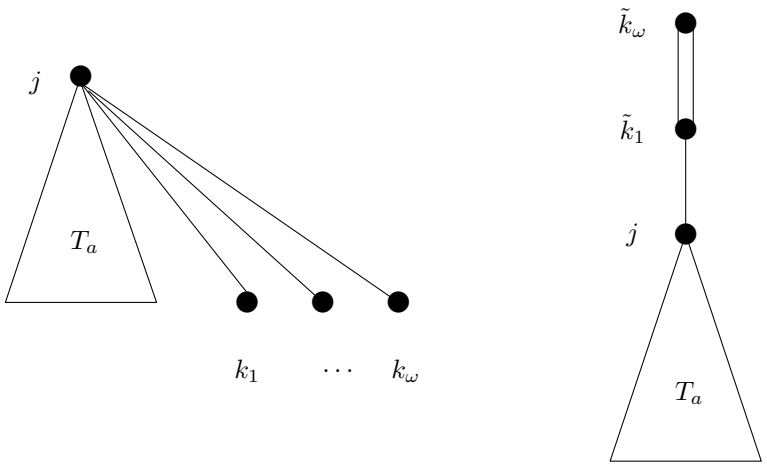


FIG. 2. *Elimination trees before and after local reordering of leaves.*

After reordering all the sets of leaves from C in this way, we get from $\bar{K}$ the permuted matrix $\bar{\bar{K}}$ satisfying the necessary condition in Proposition 3.1. In order to implement the procedure for finding a new reordering efficiently we must be able

to efficiently find the nodes having one or more leaves. The following result may be used.

LEMMA 3.2. *Let the vertices of the elimination tree T of $\bar{K}$ be postordered. Let the $size(i), i = 1, \ldots, n+m$, be defined as follows:*

$$size(i) = \begin{cases} |T[i]| & \text{if } i \text{ is not a leaf of } T, \\ 0 & \text{otherwise.} \end{cases}$$

Let $first(i) = \min\ T[i] \cap \{1, \ldots, n+m\}$. Then if the number of leaves of i in T is denoted by no_of_leaves(i) we have

$$no\_of\_leaves(i) = i - first(i) - \sum_{parent(c_j)=i} size(c_j).$$

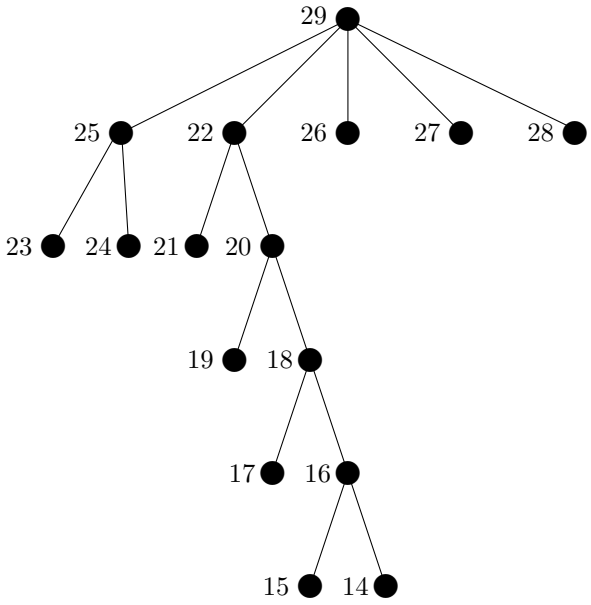


FIG. 3. *An example of a part of a postordered elimination tree.*

The lemma is a simple consequence of the basic properties of postorderings (see [1], [15]). An example is given in Figure 3, which shows a part of a postordered elimination tree. The number of leaves of root 29 of the tree from Figure 3 can be computed as $no\_of\_leaves(29) = 29 - first(29) - (size(T[22]) + size(T[25])) = 29 - 14 - (3+9) = 3$.

Lemma 3.2 suggests an efficient way to *implement* the whole modification of $\bar{K}$ to get $\bar{\bar{K}}$. First, we obtain a postordered elimination tree by standard algorithms (see [15]). Then we reorder it so that the leaves in each node are ordered consecutively after all the other subtrees rooted in children of the node. In fact, a similar transformation that reorders children sets in the elimination tree is a component of any up-to-date sparse symmetric Cholesky solver. Then, traversing the postordered elimination tree, we determine the modified permutation such that the output matrix $\bar{\bar{K}}$ satisfies the condition of Proposition 3.1.

Pseudocode of the algorithm *Leaves_Reordering* is provided below. The output of the algorithm is the new ordering sequence *newperm* (with respect to $\bar{K}$) which provides a $\bar{\bar{K}}$ satisfying the necessary condition from Proposition 3.1. Note that vector *size* keeps also the partial sums used in Lemma 3.2 for computations of *number_of_leaves*.

ALGORITHM 3.1. Leaves_Reordering.
input: The postordered elimination tree of an input matrix $\bar{K}$. The leaves in each node are ordered consecutively after all the larger subtrees rooted in children of the node.
output: New ordering sequence *newperm* such that the resulting matrix $\bar{\bar{K}}$ ordered according to the *newperm* satisfies the necessary condition from Proposition 3.1.

for j=1 *to* m+n
  first(j)=j; size(j)=0;
end for
for j=1 *to* $m+n$
  no_of_leaves=j-size(j)-first(j);
  if j does not have leaves from C **then**
    put j to *newperm*;
  else
    reorder j and its leaves which belong to C;
    put $j, \tilde{k}_1, \ldots, \tilde{k}_\omega$ to *newperm*;
  end if
  first(parent(j))= min*(first(parent(j)),j);*
  size(j)=size(j)+no_of_leaves(j)+1;
  size(parent(j))=size(parent(j))+size(j);
end for

Another condition for the factorizability of $\bar{K}$ that can be efficiently tested is given in Proposition 3.2.

PROPOSITION 3.2. *Matrix $\bar{K}$ is factorizable only if each of its leading principal submatrices have the number of indices from R greater or equal to the number of indices from C.*

In order to permute $\bar{K}$ into $\bar{\bar{K}}$ such that the latter matrix satisfies the necessary condition of Proposition 3.2 we traverse the elimination tree using its postordering. The nodes are either put into the new ordering sequence or delayed. We delay putting of the current node $i \in C$ into the new ordering sequence whenever the current number of the indices from C in the sequence is equal to the current number of the indices from R. A delayed node is stored in a stack and tested for inclusion into the new ordering sequence after its father has been processed. The postordering guarantees efficient stack search in the same way as for multifrontal implementation of Cholesky decomposition (see [7]). Clearly, the elimination tree traversal after which $\bar{\bar{K}}$ satisfies the necessary conditions from both Propositions 3.1 and 3.2 can be implemented in one routine. Nevertheless, fulfilling both these conditions still does not guarantee that our matrix is factorizable. A condition that is both necessary and sufficient is given in Proposition 3.3. First, some additional notation. Denote for $i = 1, \ldots, n+m$ a sequence of leading principal submatrices of $\bar{K}$ by $\bar{K}^{(i)}$. Let their corresponding associated unsymmetric graphs be denoted by $\bar{G}^{(i)} = (\bar{V}^{(i)}, \bar{E}^{(i)})$. Furthermore, let $\bar{R}^{(i)} = \bar{V}^{(i)} \cap R, \bar{C}^{(i)} = \bar{V}^{(i)} \cap C$.

PROPOSITION 3.3. *Matrix $\bar{K}$ is factorizable if and only if all its submatrices $\bar{K}(\bar{R}^{(i)}, \bar{C}^{(i)})$, $i = 1, \ldots, n+m$, have full column rank.*

Whether these conditions are satisfied depends (in general) on the numerical values of the offdiagonal block of K. In the following, we will concentrate on the matrices

arising in an important application of mixed-hybrid finite element discretizations of the problem of fluid flow in porous media (see, e.g., [5]), where we can guarantee the matrix factorizability.

DEFINITION 3.1. *Let K be a saddle-point matrix* (1.2). *Using the notation introduced above we say that K is an $\mathcal{F}$-matrix if the submatrix $K(R,C)$ contains at most two elements per row. Moreover, if there are two nonzeros in a row of $K(R,C)$, then their sum is zero.*

Mixed-hybrid finite element formulation of some important problems like fluid flow in porous media is of great interest in many fields including, e.g., oil recovery and groundwater pollution modeling. Effective numerical simulations of these phenomena often require accurate renderings of the fluid velocities. For this reason, such type of formulations was extensively studied during the past years. $\mathcal{F}$-matrices are often the result of the discretizations of fluid flow domains. The offdiagonal structure (expressed by $K(R,C)$) is determined by the fact that the hybridization technique introduces into the matrix a simple topological information through the Lagrangian multipliers. For more details about the relation of the matrix structure and problem formulation we refer, e.g., to [5], [20], [19].

Depth-first or breadth-first *search* is a standard technique of systematically exploring nodes in a graph (see [1], [18]). The search starts from a node, called the root, which is marked as visited. Then the unvisited nodes are recursively searched using edges from marked to new (unmarked) nodes. Each time we visit a new node v by an edge from a marked node u, we mark it as visited and the arc (u,v) is added to the tree edges. Visited nodes and induced oriented arcs form a search tree. The length (number of arcs) of the longest path in the search tree S from the root to a leaf is called its depth and is denoted by $depth(S)$. Nodes of S having the same distance $\ell, 0 \le \ell \le depth(S)$ from the root form so-called *ℓth level set*. Note that while the arcs of the search tree are not determined in a unique way, level sets form the same partition of the graph nodes for all the search trees rooted in a given node.

Let $\hat{C} \subseteq C$, $\hat{R} \subseteq R$. Let Adj denote the adjacency operator in the bipartite graph $B(\hat{R},\hat{C})$. Let $c \in \hat{C}$. Let $K(\hat{R},\hat{C}) = (B_{ij})_{i\in\hat{R},j\in\hat{C}}$. Define $Adj^{-1}(c) = \emptyset$, $Adj^0(c) = \{c\}$. Furthermore, for an integer α, $1 \le \alpha$, define $Adj^\alpha(c) = Adj(Adj^{\alpha-1}(c))$. The following lemma describes some properties of the search tree in the bipartite subgraph associated with an $\mathcal{F}$-matrix.

LEMMA 3.3. *Let $\bar{K}$ be an $\mathcal{F}$-matrix. Let $R_1 \subseteq R$, $C_1 \subseteq C$. Let S be a search tree of $B(R_1,C_1)$ rooted in $c \in C_1$, and let Adj be the adjacency operator in $B(R_1,C_1)$. Let α, $0 \le \alpha \le depth(S)$ be zero or an even integer, and assume that there are no paths of odd lengths from c to a leaf in S shorter than $\alpha+1$. Then we have*

$$\left(\sum_{k\in Adj^\alpha(c)} B_{*k}\right)_r \begin{cases} \neq 0 & \text{if } r \in Adj^{\alpha+1}(c)\backslash Adj^{\alpha-1}(c), \\ = 0 & \text{otherwise.} \end{cases}$$

In other words, $(\sum_{k\in Adj^\alpha(c)} B_{*k})$ has nonzeros only in the rows corresponding to vertices of the $(\alpha+1)$st level set of the search tree S.

Proof. For $\alpha = 0$ the assertion is clearly true. Assume that we have an even α, $2 \le \alpha \le depth(S)$. From the definition of adjacency set we have $(\sum_{k\in Adj^\alpha(c)} B_{*k})_r = 0$ for $r \notin Adj^{\alpha+1}(c)$. If $r \in Adj^{\alpha+1}(c) \cap Adj^{\alpha-1}(c)$, then, since each row of K contains at most two nonzeros, there exist uniquely determined indices k' and k'' and an even integer $\beta, 2 \le \beta \le \alpha$ such that (k',r) and (r,k'') are arcs of S and

$k' \in Adj^{\beta-2}(c)$, $k'' \in Adj^{\beta}(c)$. The zero sum condition on the $\mathcal{F}$-matrix then implies $(\sum_{k \in Adj^{\alpha}(c)} B_{*k})_r = 0$. □

The following theorem gives a necessary and sufficient condition for factorizability of the $\mathcal{F}$-matrix. Moreover, this condition can be tested cheaply.

THEOREM 3.1. *Let $\bar{K}$ be an $\mathcal{F}$-matrix. Let $R_1 \subseteq R$, $C_1 \cup \{c\} \subseteq C, c \notin C_1$, $Adj(c) \neq \emptyset$. Let S be a search tree of $B(R_1, C_1 \cup \{c\})$ rooted in c, and let Adj be the adjacency operator in $B(R_1, C_1 \cup \{c\})$. Assume that* $\mathrm{rank}(\bar{K}(R_1, C_1)) = |C_1|$. *Then*

$$\mathrm{rank}(\bar{K}(R_1, C_1 \cup \{c\})) < |C_1 \cup \{c\}|$$

if and only if there are no paths of odd length from the root to a leaf in S.

Proof. Suppose that $\mathrm{rank}(\bar{K}(R_1, C_1 \cup \{c\})) < |C_1 \cup \{c\}|$. Using the assumption on $\mathrm{rank}(\bar{K}(R_1, C_1))$, it follows that there exist a $\Gamma = \{\gamma_1, \ldots, \gamma_{|\Gamma|}\} \subseteq C_1$ and coefficients $c_{\gamma_1}, \ldots, c_{\gamma_{|\Gamma|}}$ such that

$$B_{*c} + \sum_{j=1}^{|\Gamma|} c_{\gamma_j} B_{*\gamma_j} = 0. \tag{3.1}$$

Matrix and search tree properties then imply that in order to satisfy (3.1), all nodes of S should be involved in this linear combination and all the coefficients c_{γ_j} are 1. Therefore, (3.1) transforms into $\sum_{\delta \in S} B_{*\delta} = 0$.

Assume now that there is a path of odd length from the root to a leaf in S. Let α be the length of one of such shortest paths. Denote by d its terminal node being the leaf node of S. Then we have $d \in Adj^{\alpha}(c)$. Using Lemma 3.3 we get that $(\sum_{k \in Adj^{\alpha-1}(c)} B_{*k})_d \neq 0$. From the search tree properties it follows then that $(\sum_{\delta \in S} B_{*\delta})_d \neq 0$, and we obtain a contradiction.

On the other hand, if there are no paths of odd length from the root to leaves in S, then $\sum_{k \in Adj^{depth(S)}(c)} B_{*k} = 0$ for even $depth(S) \geq 2$. Consequently, $\mathrm{rank}(\bar{K}(R_1, C_1 \cup \{c\})) < |C_1 \cup \{c\}|$, which completes the proof. □

Note that the technical assumption $Adj(c) \neq \emptyset$ used in Theorem 3.1 is very natural, as the following result shows.

LEMMA 3.4. *Take sequentially nodes of the postordered elimination tree $T(\bar{K})$. Let $R_1 \cup C_1$ for $R_1 \subseteq R, C_1 \subseteq C$ be the first $|R_1 \cup C_1|$ nodes and $c \in C$ be the $(|R_1 \cup C_1| + 1)$st node. If c is not a leaf of $T(\bar{K})$, then $Adj_{B(R_1, C_1 \cup \{c\})}(c) \neq 0$.*

The proof of Lemma 3.4 follows directly from Theorem 2.4 in [15]. Therefore, if our $\bar{K}$ satisfies the necessary conditions of Propositions 3.1 and 3.2, then the condition $Adj(c) \neq \emptyset$ is also satisfied using the sequential construction of R_1 and C_1.

Theorem 3.1 implies how to test the rank of a $\mathcal{F}$-matrix that is extended by adding a new column c and (symmetrically) a new row. The algorithm uses the graph of the offdiagonal block determined by current sets R_1, C_1 and by the column $c \in C_1$. This test is used within the procedure for modification of the current permutation of a $\mathcal{F}$-matrix $\bar{K}$. Namely, the nodes of a postordered elimination tree of the matrix $\bar{K}$ are traversed. The condition of Theorem 3.1 is used to test a new column c from C. The actual test strategy is based on the *breadth-first search*. In each such search we either find the above mentioned path of odd length or the whole search tree of an even length. In the former case we put c into the new ordering sequence. In the latter case we put c on the stack as the infeasible one. Then the *feasible* delayed columns are tested for inclusion into the new ordering sequence in the same way. Subsequent testing of a delayed column for inclusion into the new ordering sequence is performed

after its parent is processed. At this time we change its status to feasible. At the end of the procedure we get the factorizable matrix $\bar{\bar{K}}$.

The following two algorithms describe our strategy. The algorithm *Search* checks whether the condition of Theorem 3.1 is satisfied for a given node c. The algorithm *Test_Rank* then describes the whole procedure. In order to test the linear dependency of columns in the *Search* algorithm we use the bipartite graph $B(\bar{\bar{R}}, \bar{\bar{C}})$ formed from the reordered columns and rows which are temporarily stored in the new ordering sequence *newperm*. The columns which are temporarily delayed are kept in a heap.

ALGORITHM 3.2. Search($\bar{\bar{R}}, \bar{\bar{C}}$,p,i).
input: The bipartite graph $B(\bar{\bar{R}}, \bar{\bar{C}})$ constructed using the first p nodes in the new ordering *newperm*; node i.
output: Logical value *found* denoting whether the path of odd length from i to a leaf of the search tree S rooted in i was found.

Construct the breadth-first search tree of $B(\bar{\bar{R}}, \bar{\bar{C}})$ rooted in i;
if the above mentioned path was found **then**
found=*true;*
else
found=*false;*
end if

ALGORITHM 3.3. Test_Rank.
input: The input $\mathcal{F}$-matrix $\bar{K}$ and its postordered elimination tree.
output: The new permutation sequence *newperm* (related to $\bar{K}$) and $\bar{\bar{K}}$ (permuted $\bar{K}$) such that all the submatrices $\bar{\bar{K}}(\bar{\bar{R}}^{(i)}, \bar{\bar{C}}^{(i)})$ of $\bar{\bar{K}}$ have full column rank.

heap=$\emptyset$*;*
$p = 0$*;*
$\bar{\bar{R}} = \emptyset$
$\bar{\bar{C}} = \emptyset$
for j=1,$n + m$
do while *heap* $\neq \emptyset$
choose k from *heap* having minimal $after(k)$ among all nodes in *heap;*
if *after(k)* $> j$ **then**
break*;*
end if*;*
Search($\bar{\bar{R}}, \bar{\bar{C}}$,*p,k);*
if *(.not.found)* **then**
after(k)=parent(k);
else
remove k from *heap*;
put k to *newperm;*
$\bar{\bar{C}} = \bar{\bar{C}} \cup \{k\}$*;*
$p = p + 1$*;*
end if*;*
end do*;*
if $j \in R$ **then**
put j to *newperm;*
$\bar{\bar{R}} = \bar{\bar{R}} \cup \{j\}$*;*
$p = p + 1$*;*
else
Search($\bar{\bar{R}}, \bar{\bar{C}}$,*p,j);*
if *(.not.found)* **then**
after(j)=parent(j);
put j *to heap;*

```
        else
            put j to newperm;
            C̄ = C̄ ∪ {j};
            p = p + 1;
        end if;
    end if;
end for.
```

Note that the simple structure of the offdiagonal block suggests efficient breadth-first searches. We need to construct the search tree in the bipartite graph of the current offdiagonal block for each column from C and from the heap. The basic implementational trick is to reuse the constructed search tree if the new search tree can be obtained by extending the current one. We observed in our experiments that the search was finished rather early in most cases. This can be attributed to the fact that MMD reordering tends to construct wide elimination trees for problems arising from discretization of topologically regular meshes. Nevertheless, it may happen that some of the search trees have many levels.

The procedure *Test_Rank* can be used even for more general matrices. However, in this case we do not have guaranteed the factorizability of the output matrix and, in addition, the search can be more time consuming.

Consider the example matrix from Figure 1. MMD reordering of the matrix provides the elimination tree depicted in Figure 4 on the left. Here we used the reordering provided by the Matlab function *symmmd*. The matrix in Figure 4 on the right is then reordered by the output permutation coming from the algorithm *Leaves_Reordering* that was applied to the postordered elimination tree. We can see that the processing of a lot of leaves was delayed. Nevertheless, the overall fill-in is clearly modest. The algorithm *Test_Rank* then does not change the obtained order. This was a typical behavior in our experiments where the *Test_Rank* algorithm delays pivots only occasionally.

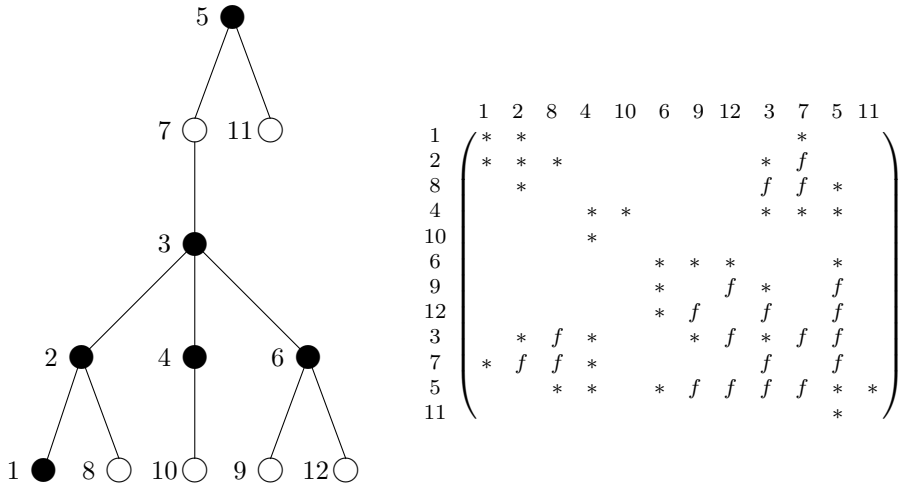


FIG. 4. *The elimination tree for the matrix from Figure* 1 *reordered by MMD (left) and the final reordered matrix after application of Leaves_Reordering algorithm (right). Circles which are not filled correspond to nodes from C.*

TABLE 1
Size of the factor L in the LDL^T decomposition using two different ordering strategies.

Matrix	$n+m$	n	nnz	Schur	LDL^T
S3P	477	270	1539	3151	2957
LI5	868	480	2208	5098	4686
LI6	3707	2100	9575	40269	34498
M3P	3744	2160	12312	46504	44002
DORT2	12992	7515	42935	296818	231312
DORT	22967	13360	76199	641828	551215

4. Implementation and experimental results. In this section, we present results of our experiments with the new ordering strategy on several $\mathcal{F}$-matrices arising in mixed-hybrid finite element discretizations of three-dimensional potential fluid flow problems (see [20]). Note that in practical calculations the matrix patterns for flow field computations are often used repeatedly, e.g., in the chemical transport outer loops. We are interested in the amount of fill-in in the factors because this crucially determines the time of numerical decomposition. The amount of fill-in was compared with that of a solver based on the Schur complement approach. In this case the MMD algorithm was applied to the principal leading submatrix H and then to the Schur complement system. This ordering strategy (ordering in stages) can be interpreted as using MMD ordering with the constraint that nodes from R are ordered first (see [16], [2]).

In our experiments we computed right-hand sides corresponding to a solution of all ones. We did not face any significant numerical instability in the computation. This seems to correspond to the results in [13]; cf. [22]. Namely, the stability of the computation is determined by a ratio of the maximal singular value of the offdiagonal block and of the maximal eigenvalue of the inverse of the diagonal block H of the $\mathcal{F}$-matrix. In our case, the ratio can be reasonably bounded using discretization parameter h of the problem. For details we refer to [19].

Our numerical procedure for solving the linear algebraic systems with saddle-point matrices consists of three stages:

- MMD ordering, elimination tree, and postordering construction;
- modification of permutations so that the permuted matrix satisfies conditions from Propositions 3.1 and 3.2 and Theorem 3.1;
- left-looking supernodal LDL^T solver (see [21]).

Table 1 compares the size of L factors of the final matrix $\bar{\bar{K}}$ for both approaches. Test matrices come from the mixed-hybrid finite element discretizations of three-dimensional potential fluid flow problem based on Raviart–Thomas elements of the lowest order. Namely, the velocity is approximated by vector functions linear on every element and piezometric potential is approximated by elementwise constant functions. The matrix dimension and the dimension of its left upper block are $n+m$ and n, respectively; nnz denotes number of nonzeros in the symmetric part of the original matrix. The column marked "Schur" contains factor size (number of the nonzeros) in the Schur complement approach. The column labeled "LDL^T" contains factor size required by LDL^T decomposition using the new strategy proposed in the paper.

It is clear that the new ordering provided better results. This is in agreement with general opinion that the ordering that is more general than the one used in the Schur complement approach may also provide less fill-in (see, e.g., [25]). An interesting

question is how much the original MMD ordering had to be modified for matrices in these experiments. It is natural that MMD tends to choose the leaves from C since these nodes have degrees at most two. In our test cases more than 80% of the leaves were taken from C and had to be delayed. This behavior can be observed in Figure 4 as well. The delay induced an increase in fill around 30%. Delay of nodes based on Theorem 3.1 was rather rare with only a few cases for each matrix. Note that there were no delays of nodes based on Theorem 3.1 in the two smallest matrices.

5. Conclusions. This paper presents a new ordering strategy for the class of symmetric saddle-point matrices. The approach is theoretically justified for an important subclass of matrices arising in mixed and hybrid formulations of the finite element method. The fill-in size obtained during the LDL^T decomposition of the matrices is compared with that obtained in the straightforward Schur complement strategy using efficient fill-in minimizing ordering. Our ordering is shown to be superior.

In spite of lack of more general theoretical results we would like to consider the techniques mentioned here as ordering routines that help to solve saddle-point systems in various applications. Namely, if we change the initial fill-in minimizing permutation so that $\bar{K}$ satisfies Propositions 3.1 and 3.2, the most important sources of instabilities of the decomposition will be removed. This can be combined with other practical rules, e.g., as a pivot modification (see [26]) in both direct solvers and preconditioner computations. We will study use of the algorithms as preprocessing schemes in the Bunch–Parlett sparse supernodal left-looking or right-looking (MA47) linear solvers for saddle-point matrices. Recently, an interesting class of preconditioners constructed from spanning trees of matrix graphs or their extensions and based on work of P. Vaidya was introduced (see [3], [4]). Construction of the preconditioners needs to test linear independence of sets of structurally simple vectors having thus connecting links to our work.

Acknowledgments. The author gratefully acknowledges the hospitality provided by Emilio Spedicato at the University of Bergamo. He would also like to thank Jiří Mužák and Dalibor Frydrych for preparing the test problems and to Michele Benzi and Zdeněk Strakoš for their comments on the previous version of the paper.

REFERENCES

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[2] C. Ashcraft, D. Pierce, D.K. Wah, and J. Wu, *The Reference Manual for SPOOLES, Release* 2.2: *An Object Oriented Software Library for Solving Sparse Linear Systems of Equations*, Software Description Report, Boeing Computer Services, available from www.netlib.org, 1999.

[3] M. Bern, J.R. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, *Support graph preconditioners*, SIAM J. Matrix Anal. Appl., submitted.

[4] E.G. Boman, D. Chen, B. Hendrickson, and S. Toledo, *Maximum-weight-basis preconditioners*, Numer. Linear Algebra Appl., submitted.

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

[6] I. Duff, A. Erisman, and J. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, UK, 1986.

[7] 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.

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

[9] R. Fourer and S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Math. Programming, 62 (1993), pp. 15–39.

[10] J.A. George and J.W.H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[11] J.A. George and J.W.H. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Rev., 31 (1989), pp. 1–19.
[12] P.E. Gill, W. Murray, D.B. Poncelen, and M.A. Saunders, *Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming*, Technical Report SOL 91-7, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1991.
[13] P.E. Gill, M.A. Saunders, and J.R. Shinnerl, *On the stability of Cholesky factorization for symmetric quasidefinite systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 35–46.
[14] G.H. Golub and C.F. Van Loan, *Unsymmetric positive definite linear systems*, Linear Algebra Appl., 28 (1979), pp. 85–98.
[15] J.W.H. Liu, *A compact row storage scheme for Cholesky factors using elimination trees*, ACM Trans. Math. Software, 12 (1986), pp. 127–148.
[16] J.W.H. Liu, *The minimum degree ordering with constraints*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1136–1145.
[17] J.W.H. Liu, *A tree model for sparse symmetric indefinite matrix factorization*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 26–39.
[18] J.W.H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
[19] J. Maryška, M. Rozložník, and M. Tůma, *The potential fluid flow problem and the convergence rate of the minimal residual method*, Numer. Linear Algebra Appl., 3 (1996), pp. 525–542.
[20] J. Maryška, M. Rozložník, and M. Tůma, *Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem*, SIAM J. Sci. Comput., 22 (2000), pp. 704–723.
[21] E.G. Ng and B.W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., 14 (1993), pp. 1034–1056.
[22] M.A. Saunders and J.A. Tomlin, *Stable Reduction to KKT Systems in Barrier Methods for Linear and Quadratic Programming*, preprint presented at the International Symposium on Optimization and Computation, Hayama, Kanagawa, Japan, August, 1996.
[23] K. Turner, *Computing projections for the Karmarkar algorithm*, Linear Algebra Appl., 152 (1991), pp. 141–154.
[24] R.J. Vanderbei, *Symmetric Quasi-definite Matrices*, Report SOR–91–10, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1991.
[25] R.J. Vanderbei, *Symmetric quasi-definite matrices*, SIAM J. Optim., 5 (1995), pp. 100–113.
[26] R.J. Vanderbei and T.J. Carpenter, *Symmetric indefinite systems for interior point methods*, Math. Programming, 58 (1993), pp. 1–32.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 916–928
© 2002 Society for Industrial and Applied Mathematics

EIGENVALUES OF WORDS IN TWO POSITIVE DEFINITE LETTERS*

CHARLES R. JOHNSON† AND CHRISTOPHER J. HILLAR‡

Abstract. The question of whether all words in two real positive definite letters have only positive eigenvalues is addressed and settled (negatively). This question was raised some time ago in connection with a long-standing problem in theoretical physics. A large class of words that do guarantee positive eigenvalues is identified, and considerable evidence is given for the conjecture that no other words do. In the process, a fundamental question about solvability of symmetric word equations is encountered.

Key words. positive definite matrix, word, nearly symmetric, positive eigenvalues, palindrome, trace conjecture

AMS subject classifications. 15A57, 15A90, 81Q99, 20F10, 15A42, 15A23

PII. S0895479801387073

Introduction. A *word* is a juxtaposed sequence of letters chosen (with repetition allowed) from a given alphabet. We shall be concerned here with an alphabet of two letters, $\{A, B\}$, so that a sample word would be $AABABBBAAB$; thus, hereafter "word" means one over a two-letter alphabet. The *length* of a word is the total number of letters present (including repetitions); the sample word has length 10. We shall be interested in the combinatorial structure of words as abstract objects, but, often, we will interpret a word as the matrix resulting from the substitution of two independent positive definite matrices for A and B. The eigenvalues and trace of the resulting matrix will be our primary interest.

The initial motivation comes from a chain of three questions raised by Lieb [L], stemming from issues in quantum physics [BMV]. In addition Pierce raised Question 3 below from an independent source [P]. The three questions are the following:

Question 1. Does the polynomial $p(t)$, defined by $p(t) = \mathrm{Tr}[(A + Bt)^m]$, have all positive coefficients whenever A and B are positive definite matrices?

Since the coefficient of t^k in $p(t)$ is the trace of the sum of all words in A and B with length m and k B's, the following, which could help answer Question 1, has also been asked [L].

Question 2. Is the trace of a given word positive for all positive definite A and B?

Since a matrix with positive eigenvalues necessarily has positive trace, a yet more precise question has also been raised [L], [P].

Question 3. Are all the eigenvalues of a given word positive for all positive definite A and B?

In addition, these particular questions and a number of natural issues they raise seem central to matrix analysis. Since we became interested in them (thanks to Lieb

*Received by the editors April 1, 2001; accepted for publication (in revised form) by R. Bhatia September 4, 2001; published electronically March 13, 2002. This research was conducted, in part, during the summer of 1999 at the College of William and Mary's Research Experiences for Undergraduates program and was supported by NSF REU grant DMS-96-19577.

http://www.siam.org/journals/simax/23-4/38707.html

†Mathematics Department, College of William and Mary, Williamsburg, VA 23187-8795 (crjohnso@math.wm.edu).

‡Mathematics Department, University of California, Berkeley, CA 94720 (chillar@math.berkeley.edu).

and Pierce), we have learned that a number of different investigators (including us) have tested them empirically by trying many different words and calculating the eigenvalues for many (tens of thousands) different randomly generated pairs of matrices of different sizes. To our knowledge, no one turned up a counterexample via such simulation, rendering Question 3 all the more interesting. Indeed, this apparent rarity of counterexamples surely means that something interesting is going on, and we have found that this area suggests many intriguing questions, a few, but not all, of which we discuss here.

We call a word *symmetric* if it reads the same right to left as left to right; e.g., $ABBABBA$ is symmetric, but $ABABBA$ is not (in other contexts, the name "palindromic" is also used). To simplify exposition, we shall often use exponents in the representation of a word; e.g., the symmetric word above might have been written AB^2AB^2A. We are principally concerned here with real symmetric positive definite matrices, though in many cases the complex Hermitian case is the same. We shall try to explicitly draw a distinction only when it is important. We intend to exploit differences in the complex Hermitian case in further work. Certain symmetries of a word do not change the eigenvalues, and, since eigenvalues are our interest, we shall freely use such symmetries and, often, only view two words as distinct if they are not equivalent via the following transformations:

(i) *Reversal.* Writing the letters of the words in reverse order. This corresponds to transposition of the matrix product and thus does not change eigenvalues.

(ii) *Cyclic permutation.* Movement of the first letter of the word to the end of the word. This can be realized as a similarity of the word via the first letter and, thus, also does not change eigenvalues.

(iii) *Interchange of A and B.* This may change the eigenvalues of a particular word, but, as A and B are both positive definite, it does not change the possible eigenvalues.

Note that a symmetric word is one that is identical to its own reversal. There are, for example, 20 words of length 6 with 3 A's, but only 3 that are distinct up to the above symmetries: $ABABAB$, A^3B^3, and ABA^2B^2.

Tangentially, we note that there is an algorithm for generating the equivalence class, relative to the above symmetries, of a word of length L or determining the number of distinct equivalence classes among N such words. Given a word W, another word V lies in its equivalence class if and only if V is the result of k cyclic permutations ($0 \leq k \leq L$), composed with (possibly) a reversal, composed with (possibly) an interchange, applied to W. This gives an algorithm of order $\mathrm{O}(NL)$.

Since a symmetric word may inductively be seen to be congruent [HJ, p. 223] to either the center letter (if the length is odd) or to I (if the length is even), we have by Sylvester's law of inertia the following.

LEMMA 1. *A symmetric word in two positive definite letters is positive definite and, thus, has positive eigenvalues.*

It follows that any symmetric word gives an affirmative answer to Question 3.

It has long been known [HJ] that a product of two positive definite matrices (e.g., the word AB) has positive eigenvalues and is diagonalizable. We call a diagonalizable matrix with positive eigenvalues *quasi-positive* and record here a slightly more complete observation.

LEMMA 2. *The n-by-n matrix Q is quasi-positive if and only if $Q = AB$, in which A and B are positive definite. Moreover if $Q = SDS^{-1}$, with D a positive diagonal matrix, then all factorizations AB of Q into positive definite matrices A and B are*

given by

$$A = SES^* \quad \textit{and} \quad B = S^{-1*}E^{-1}DS^{-1},$$

in which E is a positive definite matrix that commutes with D.

Proof. If $Q = AB$, with A and B positive definite matrices, then Q is similar to $A^{-1/2}ABA^{1/2} = A^{1/2}BA^{1/2}$, which is congruent to B and, therefore, positive definite. Thus, Q has positive eigenvalues and is diagonalizable, as is so for a positive definite matrix.

If Q is quasi-positive, $Q = SDS^{-1}$, with D positive diagonal, then $Q = AB$, with $A = SES^*$ and $B = S^{-1*}E^{-1}DS^{-1}$ (E is a positive definite matrix commuting with D), both positive definite. Suppose that $Q = AB$ is some other factorization into positive definite matrices. So $B = A^{-1}Q$ is Hermitian. Then, $A^{-1}Q = Q^*A^{-1}$ or $AQ^* = QA$ or $AS^{-1*}DS^* = SDS^{-1}A$, so that $S^{-1}AS^{-1*}D = DS^{-1}AS^{-1*}$. Thus, $S^{-1}AS^{-1*}$ commutes with D; call $E = S^{-1}AS^{-1*}$, and then $A = SES^*$. It follows that E is Hermitian and positive definite, as A is. Now, $B = A^{-1}Q = S^{-1*}E^{-1}DS^{-1}$, which is positive definite since $E^{-1}D$ is (because they commute). ∎

We now know that the nonsymmetric word AB also positively answers Question 3, but much more follows from Lemmas 1 and 2. We call a word *nearly symmetric* if it is either symmetric or the product (juxtaposition) of two symmetric words. It is an interesting exercise that the nearly symmetric words are unchanged by the three symmetries (i), (ii), and (iii). There is also a simple algorithm to check for near symmetry: left to right, parse a given word after each initial symmetric portion and check the remainder for symmetry (counting the empty word as symmetric). We then have the following.

THEOREM 3. *Every nearly symmetric word in two positive definite letters has only positive eigenvalues.*

Proof. The proof follows from Lemmas 1 and 2. ∎

Are all words nearly symmetric? No, but all sufficiently short words are.

THEOREM 4. *A word in which one of the letters appears at most twice is nearly symmetric.*

Proof. Without loss of generality, we examine the situation in which B appears at most twice. If a word contains only the letter A, the result is trivial. If the letter B appears only once, then the word will be of the form $A^pBA^q (p, q \geq 0)$. If $p \geq q$, then we have $A^pBA^q = A^{p-q}(A^qBA^q)$, and if $p \leq q$, we have $A^pBA^q = (A^pBA^p)A^{q-p}$. In both cases, the word is nearly symmetric. In the case of two B's, the word can be written as $A^pBA^qBA^t (p, q, t \geq 0)$, and so our word is one of the nearly symmetric words, $(A^pBA^qBA^p)A^{t-p}$ or $A^{p-t}(A^tBA^qBA^t)$. ∎

In order to not be nearly symmetric then, a word must have length at least 6 and 3 each of A and B. Among the 3 such equivalence classes of words of length 6, one is actually not nearly symmetric, ABA^2B^2, and this shows that Theorem 4 is best possible. This is the first interesting word relative to Question 3, and we have the following corollary.

COROLLARY 5. *Every nearly symmetric word, and thus every word of length < 6 has only positive eigenvalues.*

An interesting question one can ask is how many nearly symmetric words there are of a given length L. More importantly, what does the fraction of nearly symmetric words to the total number of words approach as L goes to infinity? The result can be found in [K], and it states that the number of nearly symmetric words of length L is $O(L \cdot 2^{(3/4)L})$. This gives us that the density of such words approaches 0, and therefore,

as L goes to infinity, there is a pool of potential negative answers to Questions 2 and 3 that ever increases in relative frequency.

The situation is much simpler for 2-by-2 matrices, and we note (as does Pierce [P] and Spitkovsky [S]) the following.

FACT 6. *Both eigenvalues of any word in two 2-by-2 positive definite matrices are positive.*

Proof. We will actually show something stronger. Let W be any finite product of real positive powers of A and B, in which A and B are 2-by-2 positive definite (complex) Hermitian matrices. (Here, we take principal powers, so that W is uniquely defined.) We first preprocess the word as follows. Make one letter diagonal via uniform unitary similarity, and then make the other letter entrywise nonnegative via a diagonal unitary similarity. This does not change the first letter. Now, the word is nonnegative (as it is clear from the spectral theorem that a positive power of a nonnegative 2-by-2 positive definite matrix is nonnegative). If it is diagonal, there is nothing more to do (the diagonal entries are positive). If not, apply the Perron–Frobenius theorem (which says a positive matrix must have a positive eigenvalue [HJ, p. 503]) and the fact that the determinant is positive to show that the other eigenvalue is positive as well. ∎

COROLLARY 7. *The polynomial $p(t)$, defined by $p(t) = \mathrm{Tr}[(A + Bt)^m]$, has all positive coefficients whenever A and B are 2-by-2 positive definite matrices.*

This all suggests that careful consideration of the word ABA^2B^2, or, equivalently, $(BA)(BA)(AB)$, for 3-by-3 positive definite A and B is warranted. This is equivalent, by Lemma 2, to the study of the expression C^2C^{T} for quasi-positive C. Since any real matrix with real eigenvalues may be upper triangularized by orthogonal similarity, it suffices to consider

$$C = \begin{bmatrix} a & x & z \\ 0 & b & y \\ 0 & 0 & c \end{bmatrix}$$

with $a, b, c > 0$. If a, b, and c are distinct, C is diagonalizable and thus quasi-positive. Using MAPLE, and with the assistance of Shaun Fallat, it was found that x, y, z and such a, b, c may be found so that $\mathrm{Tr}(C^2C^{\mathrm{T}}) < 0$. Consistent with prior empirical experience, choice of such x, y, z and a, b, c is delicate and falls in a very narrow range. Resulting A and B (see Lemma 2) that exhibit a negative answer to Question 2 (and, thus, 3) are, for example,

$$A_1 = \begin{bmatrix} 1 & 20 & 210 \\ 20 & 402 & 4240 \\ 210 & 4240 & 44903 \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} 36501 & -3820 & 190 \\ -3820 & 401 & -20 \\ 190 & -20 & 1 \end{bmatrix}.$$

The extreme and reverse diagonal progressions are typical of such examples. If the diagonal of one is "flattened" by orthogonal similarity, the progression on the diagonal of the other becomes more extreme.

We remark at this point that words giving a negative answer to Question 2 in the 3-by-3 case imply negative answers in the n-by-n case for $n > 3$. This allows us to restrict our attention to the 3-by-3 positive definite matrices. Simply direct sum a 3-by-3 example (giving a negative trace) with a sufficiently small positive multiple of the identity to get a larger example.

The idea of our first construction and some fortunate characteristics of the constructed pair allow the identification of several infinite classes of words giving negative answers to Questions 2 and 3. We indicate some of these next.

1. Any positive integer power of a word that does not guarantee positive eigenvalues also does not guarantee positive eigenvalues. For instance, this shows that $BABAABBABAAB$ can have a nonpositive eigenvalue. This is Theorem 8 below.

2. Suppose a word can be written in terms of another word T as $T^k(T^*)^j$ for $k \neq j$. Furthermore, suppose $T = S_1S_2$ is a product of two symmetric words S_1 and S_2. Then if the simultaneous word equations

$$S_1(A, B) = C,$$
$$S_2(A, B) = D$$

may be solved for positive definite A and B given positive definite C and D, then the original word can have negative trace. The first nontrivial application of this technique is the first counterexample, $(BA)^2AB$, in which $S_1 = B$, $S_2 = A$, $k = 2$, and $j = 1$. This result is Theorem 9 below.

3. Infinite classes involving single-letter length extension: this is a nice application of sign analysis. Our first result is the following.

(a) *The word, ABA^2B^{2+k} with k a nonnegative integer can have negative trace.*

Proof. A direct computation with A_1 and B_1 from above gives us that

$$(BABAAB)B = \begin{bmatrix} -164679899 & 17226460 & -856450 \\ 62354360 & -6523192 & 324340 \\ -5877450 & 614880 & -30573 \end{bmatrix}$$

has sign pattern

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

Next, notice that B_1 has the sign pattern

$$\begin{bmatrix} + & - & + \\ - & + & - \\ + & - & + \end{bmatrix}$$

and that

$$\begin{bmatrix} - & + & - \\ + & - & + \\ - & + & - \end{bmatrix} \begin{bmatrix} + & - & + \\ - & + & - \\ + & - & + \end{bmatrix}$$

is

$$\begin{bmatrix} - & + & - \\ + & - & + \\ - & + & - \end{bmatrix}$$

unambiguously.

Hence, multiplying the product $BABAABB$ by B on the right any number of times will preserve the negativity of the trace. Therefore, $BABAABB \cdot B^k$ gives a negative answer to Question 2 for all integers $k \geq 0$. □

Proofs using the same technique give us many infinite classes of counterexamples, some of which we list below:

(b) $ABABAAB^k$, $k \geq 2$.
(c) $ABBABAAB^k$, $k \geq 2$.
(d) $ABAABBAAB^k$, $k \geq 2$.

4. Recall the two matrices A_1 and B_1 giving $(BA)(BA)(AB)$ a negative trace. These matrices can also be used to prove that the words ABA^pB^q, $ABBABA^pB^q$, and $ABABA^pB^q$ can have a negative trace for all integers $p, q \geq 2$. Notice that (a), (b), and (c) above are corollaries to this result. This is Theorem 10 below.

We now present proofs of the three theorems mentioned above.

THEOREM 8. *Let W be any word for which there are positive definite A and B such that $W(A, B)$ has an eigenvalue that is not positive. Then, for any positive integer k, there are positive definite letters such that W^k has a nonpositive eigenvalue.*

Proof. Let A, B be positive definite matrices that give W a nonpositive eigenvalue, and let λ be such an eigenvalue. If $k \in \{1, 2, 3, \ldots\}$, then an eigenvalue of $W(A, B)^k$ is λ^k. If λ^k is nonpositive, we are done, so the problem lies in the possibility that $\lambda^k > 0$. It will be necessary, therefore, in this case to create a new pair of positive definite matrices A' and B' that give $W(A', B')^k$ a nonpositive eigenvalue.

We first offer a description of our approach before presenting the details that follow. The idea is to parameterize a pair of positive definite matrices in terms of a real variable t, $0 \leq t \leq 1$, and then examine the eigenvalues of the word W^k evaluated at those matrices. Using the continuity of eigenvalues on matrix entries, we then show that $W(A(t), B(t))^k$ cannot have positive eigenvalues for all $0 \leq t \leq 1$.

Let λ_A be the largest eigenvalue of A, and let λ_B be the largest eigenvalue of B. Define the following parameterization:

$$A(t) = t \cdot (\lambda_A I - A) + A \quad \text{and} \quad B(t) = t \cdot (\lambda_B I - B) + B \quad \text{for } 0 \leq t \leq 1.$$

We first note that $A(t)$ and $B(t)$ are positive definite for all such t since $(\lambda_A I - A)$ and $(\lambda_B I - B)$ are positive semidefinite by a simple eigenanalysis. Next, notice that $A(1) = \lambda_A I$ and $B(1) = \lambda_B I$, giving $W(A(1), B(1))$ positive eigenvalues. Additionally, $A(0) = A$ and $B(0) = B$, which shows that $W(A(0), B(0))$ has a nonpositive eigenvalue, by assumption. Since the eigenvalues of a matrix depend continuously on its entries [HJ, p. 539], the eigenvalues of $W(A(t), B(t))$ also depend continuously on t.

For $t \in [0, 1]$, the spectrum of $W(A(t), B(t))$ cannot contain 0 because each product, $W(A(t), B(t))$, has positive determinant. Now, let

$$\Gamma = \{t \in [0, 1] \mid W(A(t), B(t)) \text{ has a positive spectrum}\}.$$

Clearly, this set is not empty as $1 \in \Gamma$, and it is not the entire interval as $0 \notin \Gamma$. A straightforward continuity argument also shows that Γ is closed. Let t_M be the greatest lower bound of Γ, and notice that from above, $t_M \neq 0$ and $t_M \in \Gamma$. As a result, the eigenvalues of $W(A(t_M), B(t_M))$ are all positive. By continuity again, we can choose $t < t_M$ such that the eigenvalues of $W(A(t), B(t))$ are as close to the eigenvalues of $W(A(t_M), B(t_M))$ as we wish.

We are now ready to prove the theorem. Let k be a positive integer. By continuity, choose $t < t_M$ such that there is an eigenvalue, λ, of $W(A(t), B(t))$ with an argument θ satisfying $-\pi/k < \theta < \pi/k$ (see Figure 1). This guarantees that λ^k cannot be real. Our new pair $A(t), B(t)$ now proves the word W^k can have nonpositive eigenvalues. □

THEOREM 9. *If j and k are positive integers such that $j \neq k$, then there is a real, quasi-positive matrix T such that $T^k(T^*)^j$ has negative trace.*

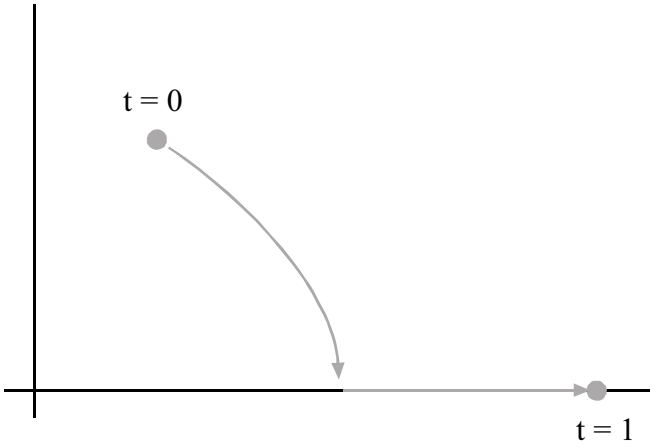


FIG. 1. *Tracking an eigenvalue of* $W(A(t), B(t))$.

Proof. We first note that we can assume $k > j$, since if $k < j$, we examine $[T^k(T^*)^j]^*$. We also assume without loss of generality that T has 1 for an eigenvalue and it is the smallest eigenvalue of T.

Using Schur triangularization, we suppose

$$T = \begin{bmatrix} 1 & x & z \\ 0 & a & y \\ 0 & 0 & b \end{bmatrix},$$

with $x, y, z \in \Re$ and $b > a > 1$.

Since it is necessary to compute powers of T, we note that

$$T^k = \begin{bmatrix} 1 & X_k & Z_k \\ 0 & a^k & Y_k \\ 0 & 0 & b^k \end{bmatrix} = \begin{bmatrix} 1 & X_{k-1} & Z_{k-1} \\ 0 & a^{k-1} & Y_{k-1} \\ 0 & 0 & b^{k-1} \end{bmatrix} \begin{bmatrix} 1 & x & z \\ 0 & a & y \\ 0 & 0 & b \end{bmatrix},$$

in which $X_k(Y_k;\, Z_k)$ is the 1,2 (2,3; 1,3) entry of T^k, $k > 0$.

The above expression allows us to find formulae for the entries of T^k by way of the following obvious recurrences:

$$X_k = x + aX_{k-1}; \quad Y_k = ya^{k-1} + bY_{k-1}; \quad Z_k = z + yX_{k-1} + bZ_{k-1}.$$

An easy induction gives us that

$$X_k = x\frac{a^k - 1}{a - 1}; \quad Y_k = y\frac{a^k - b^k}{a - b};$$

$$Z_k = xy\frac{1}{a-1} \cdot \left(\frac{a^k - b^k}{a - b} - b^{k-1} - \frac{b^{k-1} - 1}{b - 1} \right) + z\frac{b^k - 1}{b - 1} = xyC_k + zD_k,$$

in which $C_k = \frac{1}{a-1} \cdot (\frac{a^k - b^k}{a-b} - b^{k-1} - \frac{b^{k-1}-1}{b-1})$, $D_k = \frac{b^k - 1}{b-1}$ depend only on a, b, and k.

Thus, the trace of $T^k(T^*)^j$ can be computed explicitly in terms of x, y, z, a, b, k, j. It is

$$\begin{aligned}\mathrm{Tr}[T^k(T^*)^j] &= \mathrm{Tr}\left[\begin{pmatrix}1 & X_k & Z_k\\ 0 & a^k & Y_k\\ 0 & 0 & b^k\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ X_j & a^j & 0\\ Z_j & Y_j & b^j\end{pmatrix}\right]\\ &= (1 + X_kX_j + Z_kZ_j) + (a^{k+j} + Y_kY_j) + b^{k+j}\\ &= 1 + a^{k+j} + b^{k+j} + x^2\frac{a^k-1}{a-1}\cdot\frac{a^j-1}{a-1} + y^2\frac{a^k-b^k}{a-b}\cdot\frac{a^j-b^j}{a-b}\\ &\quad + x^2y^2C_kC_j + xyz(C_kD_j + C_jD_k) + z^2D_kD_j.\end{aligned}$$

Fix $a, b > 1$ and set $y = x$. Now, view $\mathrm{Tr}[T^k(T^*)^j]$ as a quadratic polynomial in z. For this polynomial to take on negative values, it is necessary and sufficient for its discriminant to be positive. This discriminant is a quartic polynomial in x; therefore, if we can show that its leading coefficient is always positive, this will demonstrate that for large enough values of x, the discriminant will also be positive. The coefficient of x^4 in this discriminant is

$$\begin{aligned}&(C_kD_j + C_jD_k)^2 - 4D_kD_j(C_kC_j)\\ &= C_k^2D_j^2 + 2C_kC_jD_kD_j + C_j^2D_k^2 - 4C_kC_jD_kD_j = (C_kD_j - C_jD_k)^2.\end{aligned}$$

When $k = j$, the expression above is 0, so it is necessary to prove that whenever $k \neq j$, $C_kD_j \neq C_jD_k$. Examining $C_kD_j - C_jD_k$, this is equivalent to proving that

$$a^jb^{k+1} - a^k + b^k + a^kb - b^{k+1} + a^kb^j - a^kb^{j+1} - a^jb^k - b^j + a^j - a^jb + b^{j+1}$$

is never zero unless $k = j$. Factoring out $(b-1)$, we need only prove that

$$f(a,b) = a^jb^k + a^k - b^k - a^kb^j + b^j - a^j$$

is never zero unless $k = j$. Examine the following polynomial in x:

$$g(x) = f(a,x) = x^k(a^j - 1) + x^j(1 - a^k) + a^k - a^j.$$

It is easy to see that $g(1) = 0$ and $g(a) = 0$. From Descartes's rule of signs, it is clear (since $a > 1$) that g has either 0 or 2 positive real roots. Since a and 1 are two such roots, g has no more positive ones. Hence, $g(b) \neq 0$ for $b \neq 1$, a.

This concludes the proof that $T^k(T^*)^j$ will have negative trace for some quasi-positive matrix T. Note that a description of all 3-by-3 quasi-positive T that give $T^k(T^*)^j$ a negative trace is implicit in the proof. □

Our first corollary to this theorem is that the word $(BA)^2AB$ gives a negative answer to Question 2; but moreover, we also now have a description of all 3-by-3 positive definite A and B that give $(BA)^2AB$ a negative trace. Theorem 9 describes all 3-by-3 quasi-positive matrices T that give T^2T^* a negative trace, and hence all positive definite matrices A and B are given by $T = BA$ from Lemma 2.

We now prove the following.

THEOREM 10. *For integers $p, q \geq 2$ and the word $W = ABA^pB^q$, there exist positive definite matrices A and B such that $W(A,B)$ has a negative trace.*

Proof. We first record a few preliminaries.

Let $F(p,q) = \mathrm{Tr}[ABA^pB^q] = \mathrm{Tr}[BAB^qA^p]$ be the desired trace of the word W. Now, suppose $A = U^*DU$ and $B = V^*EV$ are fixed positive definite matrices with U, V (real) orthogonal, and let $D = \mathrm{diag}(a,b,c)$, $E = \mathrm{diag}(r,s,t)$, $a,b,c,r,s,t > 0$. Then we can write

$$F(p,q) = \mathrm{Tr}[UBAB^qU^*D^p] = \mathrm{Tr}[VABA^pV^*E^q].$$

From these two expressions, it is clear that

$$F(p,q) = g_1(q)a^p + g_2(q)b^p + g_3(q)c^p, \tag{1}$$

$$F(p,q) = h_1(p)r^q + h_2(p)s^q + h_3(p)t^q, \tag{2}$$

where $g_i(q)$, $h_i(p)$ are linear functions in r^q, s^q, t^q and a^p, b^p, c^p, respectively. Equations (1) and (2) can be viewed as a generalization of the well-known expression for computing Fibonacci numbers. In fact, these equations imply the recurrence relations

$$F(p,q) = (a+b+c)F(p-1,q) - (ab+bc+ac)F(p-2,q) + (abc)F(p-3,q), \tag{3}$$

$$F(p,q) = (r+s+t)F(p,q-1) - (rs+rt+st)F(p,q-2) + (rst)F(p,q-3). \tag{4}$$

We are now ready to prove the result. It turns out that A_1 and B_1 (as described above) will prove the claim

$$A_1 = \begin{bmatrix} 1 & 20 & 210 \\ 20 & 402 & 4240 \\ 210 & 4240 & 44903 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 36501 & -3820 & 190 \\ -3820 & 401 & -20 \\ 190 & -20 & 1 \end{bmatrix}.$$

The values of $(a+b+c)$, $(ab+bc+ac)$, (abc), $(r+s+t)$, $(rs+st+rt)$, and (rst) are obtained from the characteristic polynomials of A and B. These polynomials are easy to compute as $P_A(t) = t^3 - 45306t^2 + 74211t - 6$ and $P_B(t) = t^3 - 36903t^2 + 44903t - 1$. Therefore, (3) and (4) become

$$F(p,q) = 45306 \cdot F(p-1,q) - 74211 \cdot F(p-2,q) + 6 \cdot F(p-3,q), \tag{5}$$

$$F(p,q) = 36903 \cdot F(p,q-1) - 44903 \cdot F(p,q-2) + F(p,q-3). \tag{6}$$

To prove the theorem, we must show that $F(p,q) < 0$ for all $p,q \geq 2$. First notice that for the base cases of $2 \leq p, q \leq 4$, we have that $F(p,q)$ are given by the following table:

	$q = 2$	$q = 3$	$q = 4$
$p = 2$	-3164	-171233664	-6318893781764
$p = 3$	-219049002	-10537988104302	-388873536893369802
$p = 4$	-9923997300324	-477421308542380824	-17617832833924812095724

To prove the result using the recurrences above, we will invoke induction and prove something stronger. Namely, we claim that for all $p,q \geq 2$, $F(p,q) < 0$ and also the following inequalities hold:

$$F(p,q) < 10 \cdot F(p-1,q) \quad \text{for } p > 2,$$
$$F(p,q) < 10 \cdot F(p,q-1) \quad \text{for } q > 2.$$

Suppose the result is true for all $2 \le p, q < N$ (from the table above, we can also suppose $N \ge 5$); then we want to show it true for $2 \le p,\ q \le N$. For $2 \le p, q < N$, examine $F(N, q), F(p, N)$, and $F(N, N)$. From (5) and (6), we have

$$\begin{aligned} F(N,q) &= 45306 \cdot F(N-1,q) - 74211 \cdot F(N-2,q) + 6 \cdot F(N-3,q) \\ &< 45306 \cdot F(N-1,q) - 7421.1 \cdot F(N-1),q) \\ &= 37884.9 \cdot F(N-1,q) < 10 \cdot F(N-1,q), \end{aligned} \tag{7}$$

$$\begin{aligned} F(p,N) &= 36903 \cdot F(p,N-1) - 44903 \cdot F(p,N-2) + F(p,N-3) \\ &< 36903 \cdot F(p,N-1) - 4490.3 \cdot F(p,N-1) \\ &= 32412.7 \cdot F(p,N-1) < 10 \cdot F(p,N-1). \end{aligned} \tag{8}$$

But to complete the induction, we must also show that $F(N, N) < 10 \cdot F(N-1, N)$ and $F(N, N) < 10 \cdot F(N, N-1)$. Substituting (6) into the right-hand side of (5) with $p = N$, $q = N$, we have

$$\begin{aligned} F(N,N) &= 1671927318 \cdot F(N-1,N-1) - 2034375318 \cdot F(N-1,N-2) \\ &\quad + 45306 \cdot F(N-1,N-3) \\ &\quad - 2738608533 \cdot F(N-2,N-1) + 3332296533 \cdot F(N-2,N-2) \\ &\quad - 74211 \cdot F(N-2,N-3) \\ &\quad + 221418 \cdot F(N-3,N-1) - 269418 \cdot F(N-3,N-2) \\ &\quad + 6 \cdot F(N-3,N-3) \\ &< 1671927318 \cdot F(N-1,N-1) - 203437531.8 \cdot F(N-1,N-1) \\ &\quad - 273860853.3 \cdot F(N-1,N-1) - 74.211 \cdot F(N-1,N-1) \\ &\quad - 269.418 \cdot F(N-1,N-1) \\ &= 1194628589.271 \cdot F(N-1,N-1). \end{aligned} \tag{9}$$

But from (8) with $p = N - 1$, we have

$$\begin{aligned} 36903 \cdot F(N-1,N-1) &= F(N-1,N) + 44903 \cdot F(N-1,N-2) - F(N-1,N-3) \\ &< F(N-1,N) - (1/100) \cdot F(N-1,N-1). \end{aligned}$$

Therefore, $36903.01 \cdot F(N-1, N-1) < F(N-1, N)$, which gives us easily (from (9)) that

$$F(N,N) < 1194628589.271 \cdot F(N-1,N-1) < 10 \cdot F(N-1,N).$$

To arrive at $F(N, N) < 10 \cdot F(N, N-1)$, we perform the same examination, this time with (7):

$$F(N,N-1) = 45306 \cdot F(N-1,N-1) - 74211 \cdot F(N-2,N-1) + 6 \cdot F(N-3,N-1),$$

giving us the inequality $45306.06 \cdot F(N-1, N-1) < F(N, N-1)$.

So again, from (9), we see that $F(N, N) < 10 \cdot F(N, N-1)$. This completes the induction and shows that for all $p, q \ge 2$, $F(p, q) < 0$. The proof also bounds the growth from below, but the factor of 10 is obviously not the best possible. □

At this point, we should remark that the proof for Theorem 10 above could be generalized to a certain extent. Namely, suppose W is a word that can be written as

$W_1 A^p W_2 B^q$ for some words W_1, W_2 in A and B. Then, A_1 and B_1 give this word negative trace for all integers $p, q \geq 2$ provided that for the base cases of $2 \leq p,\ q \leq 4$,

$$F(p,q) < 0; \quad F(p,q) < 10 \cdot F(p-1,q); \quad \text{and } F(p,q) < 10 \cdot F(p,q-1).$$

As an example, a calculation gives us that for the word $W = ABABA^p B^q$ the first 9 values of $F(p,q)$ are given by[1]

	$q=2$	$q=3$	$q=4$
$p=2$	−32302	−1319655482	−48697748014592
$p=3$	−1748875224	−70292975950848	−2.59394099689082e+018
$p=4$	−79232137801728	−3.18459541653658e+018	−1.17517468821039e+023

The word $W = ABBABA^p B^q$ also satisfies the base case conditions as the $F(p,q)$ are

	$q=2$	$q=3$	$q=4$
$p=2$	−222790424	−10720038844524	−3.95591587257758e+017
$p=3$	−10103386100406	−4.86025787321779e+017	−1.79353558546523e+022
$p=4$	−4.57727477164142e+017	−2.20190887755731e+022	−8.12549875102683e+026

It should now be clear that we conjecture the following.

Conjecture 4. A word has positive trace for every pair of positive definite letters if and only if the word is nearly symmetric.

Using the results and ideas we have discussed, it is possible to verify this conjecture for words of lengths less than 11. Before listing these results, we remark on how to find specific A and B for which a word has negative trace. One difficulty is how to view the set of positive definite matrices A and B. We explain a helpful parametric approach for the sample word $BAABBAAA$ and the generalization will be clear. Notice that we do not yet know that this word can have a negative trace using any of the methods thus far.

First set $Q = AB$, and recall that all solutions A, B to such an equation are given by Lemma 2 as $Q = SDS^{-1}$, $A = SES^*$, $B = S^{-1*}E^{-1}DS^{-1}$, in which D is a positive diagonal matrix, and E is a positive definite matrix commuting with D. For simplicity, we seek a positive diagonal E. Using these substitutions and some simplification, our original word has the same eigenvalues as the following expression: $DPDP^{-1}DPEPE$, in which $P = S^*S$.

Next, fix a positive definite matrix P and view the positive diagonal matrices D and E parametrically, hoping now to minimize the trace of the product above. These minimizations are easier to perform because now we have a simple parametric description of positive definite pairs. Notice that it is not necessary to find A and B to show that they exist and give the word a negative trace. However, it is useful to have explicit examples, as they may be later used to show that other (not nearly symmetric) words admit negative trace. After finding D, E, and P, we recover these letters from the equations $S^*S = P$, $A = SES^*$, $B = S^{-1*}E^{-1}DS^{-1}$. An example

[1] While values are integers, they are shown only to the first 15 significant digits.

TABLE 1
All words that are not nearly symmetric of length < 11 *admit negative trace.*

$AABABB$	Original solution A_1, B_1 using C^2C^T
$AAABABB$	Theorem 10
$AAAABABB$	Theorem 10
$AAABAABB$	Using A_2, B_2
$AAABABBB$	Theorem 10
$AABABABB$	Equivalent to $(AB)^3BA$
$AAAAABABB$	Theorem 10
$AAAABAABB$	Equivalent to $(A^2B)^2BA^2$
$AAAABABBB$	Theorem 10
$AAABAABAB$	Using A_3, B_3 produced by the technique above
$AAABAABBB$	Using A_2, B_2
$AAABABABB$	Theorem 10
$AABAABABB$	Theorem 10
$AAAAAABABB$	Theorem 10
$AAAAABAABB$	Using A_2, B_2
$AAAAABABBB$	Theorem 10
$AAAABAAABB$	Using A_4, B_4 produced by the technique above
$AAAABAABAB$	Using A_3, B_3
$AAAABAABBB$	Using A_2, B_2
$AAAABABABB$	Theorem 10
$AAAABABBBB$	Theorem 10
$AAAABBABBB$	Using A_2, B_2 (interchanging A and B)
$AAABAABABB$	Theorem 10
$AAABAABBAB$	Using A_1, B_1
$AAABABAABB$	Using A_2, B_2
$AAABABABBB$	Using A_2, B_2
$AAABABBABB$	Theorem 10
$AAABBAABBB$	Using A_5, B_5 produced by the technique above
$AABABABABB$	Equivalent to $(AB)^4BA$
$AABABABBAB$	Equivalent to $(AB)^3(BA)^2$
$AABABBAABB$	(d)

solution found using this technique for the word $BAABBAAA$ is given by

$$A_2 = \begin{bmatrix} 4351/479 & 4856/399 & 18421/62 \\ 4856/399 & 16073/64 & 3784/21 \\ 18421/62 & 3784/21 & 89917/9 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 2461/149 & -297/641 & -757/1569 \\ -297/641 & 179/6146 & 50/3767 \\ -757/1569 & 50/3767 & 269/19081 \end{bmatrix}.$$

It is easily verified that the trace of the word $BAABBAAA$ is a negative rational number given approximately by $\mathrm{Tr}(BAABBAAA) \approx -143370.8471$.

In Table 1 we list all the equivalence classes of words that are not nearly symmetric

and are of length less than 11. Next to each word, we describe the method of finding the A and B that proves they can have a negative trace.

Acknowledgment. The first author would like to acknowledge the pleasant and sometimes useful conversations with several mathematicians about this problem—in particular David Yopp and Tom Laffey.

REFERENCES

[BMV] D. Bessis, P. Moussa, and M. Villani, *Monotonic converging variational approximations to the functional integrals in quantum statistical mechanics*, J. Math. Phys., 16 (1975), pp. 2318–2325.

[HJ] R. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[K] R. Kemp, *On the number of words in the language* $\{w \in \Sigma^* \mid w = w^R\}^2$, Discrete Math., 40 (1982), pp. 225–234.

[L] E. Lieb, *private communication*.

[P] S. Pierce, *private communication*.

[S] I. Spitkovsky, *private communication*, Williamsburg, VA, 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 929–947
© 2002 Society for Industrial and Applied Mathematics

THE MULTISHIFT QR ALGORITHM. PART I: MAINTAINING WELL-FOCUSED SHIFTS AND LEVEL 3 PERFORMANCE*

KAREN BRAMAN†, RALPH BYERS†, AND ROY MATHIAS‡

Abstract. This paper presents a small-bulge multishift variation of the multishift QR algorithm that avoids the phenomenon of shift blurring, which retards convergence and limits the number of simultaneous shifts. It replaces the large diagonal bulge in the multishift QR sweep with a chain of many small bulges. The small-bulge multishift QR sweep admits nearly any number of simultaneous shifts—even hundreds—without adverse effects on the convergence rate. With enough simultaneous shifts, the small-bulge multishift QR algorithm takes advantage of the level 3 BLAS, which is a special advantage for computers with advanced architectures.

Key words. QR algorithm, implicit shifts, level 3 BLAS, eigenvalues, eigenvectors

AMS subject classifications. 65F15, 15A18

PII. S0895479801384573

1. Introduction.

This paper presents a small-bulge multishift variation of the multishift QR algorithm [4] that avoids the phenomenon of shift blurring, which retards convergence and limits the number of simultaneous shifts that can be used effectively. The small-bulge multishift QR algorithm replaces the large diagonal bulge in the multishift QR sweep with a chain of many small bulges. The small-bulge multishift QR sweep admits nearly any number of simultaneous shifts—even hundreds—without adverse effects on the convergence rate. It takes advantage of the level 3 BLAS by organizing nearly all the arithmetic work into matrix-matrix multiplies. This is particularly efficient on most modern computers and especially efficient on computers with advanced architectures.

The QR algorithm is the most prominent member of a large and growing family of bulge-chasing algorithms [11, 14, 15, 17, 28, 30, 51, 39, 40, 44, 57, 59]. It is remarkable that after thirty-five years, the original QR algorithm [25, 26, 35] with few modifications is still the method of choice for calculating all eigenvalues and (optionally) eigenvectors of small, nonsymmetric matrices. Despite being a dense matrix method, it is arguably still the method of choice for computing all eigenvalues of moderately large, nonsymmetric matrices, i.e., at this writing, matrices of order greater than 1,000. Its excellent rounding error properties and convergence behavior are both theoretically and empirically satisfactory [9, 16, 19, 32, 41, 42, 52, 62, 63, 64] despite surprising convergence failures [8, 10, 19, 56, 58].

*Received by the editors February 5, 2001; accepted for publication (in revised form) by D. Boley May 15, 2001; published electronically March 27, 2002. This work was partially supported by the National Computational Science Alliance under award DMS990005N and utilized the NCSA SGI/Cray Origin2000.

http://www.siam.org/journals/simax/23-4/38457.html

†Department of Mathematics, University of Kansas, 405 Snow Hall, Lawrence, KS 66045 (braman@math.ukans.edu, byers@math.ukans.edu). The research of these authors was partially supported by National Science Foundation awards CCR-9732671, DMS-9628626, CCR-9404425, and MRI-9977352 and by the NSF EPSCoR/K*STAR program through the Center for Advanced Scientific Computing.

‡Department of Mathematics, College of William & Mary, Williamsburg, VA 23187 (mathias@wm.edu). The research of this author was partially supported by NSF grants DMS-9504795 and DMS-9704534.

It has proven difficult to implement the QR algorithm in a way that takes full advantage of the potentially high execution rate of computers with advanced architectures—particularly hierarchical memory architectures. This is so much the case that some high-performance algorithms work around the slow but reliable QR algorithm by using faster but less reliable methods to tear or split off smaller submatrices on which to apply the QR algorithm [2, 5, 6, 7, 23, 33]. An exception is the successful high-performance pipelined Householder QZ algorithm in [18]. Although this paper is not directly concerned with distributed memory computation, it is worth noting that there are distributed memory implementations of the QR algorithm [31, 45, 48, 50].

Readers of this paper need to be familiar with the double implicit shift QR algorithm [25, 26, 35]. See, for example, any of the textbooks [20, 27, 47, 53, 62]. Familiarity with the multishift QR algorithm [4] as implemented in LAPACK version 2 [1] is helpful. We will refer to the iterative step from bulge-introduction to bulge-disappearance as a QR sweep.

1.1. Notation and the BLAS.

1.1.1. The BLAS. The basic linear algebra subprograms (BLAS) are a set of frequently required elementary matrix and vector operations introduced first in [38, 37] and later extensively developed [21, 22]. The level 3 BLAS are a small set of matrix-matrix operations like matrix-matrix multiply [22]. The level 2 BLAS are a set of matrix-vector operations like matrix-vector multiplication. The level 1 BLAS are a set of vector-vector operations like dot-products and scaled vector addition ($x \leftarrow ax+y$). Because they are relatively simple, have regular patterns of memory access, and have a high ratio of arithmetic work to data, the level 3 BLAS can be organized to make near optimal use of hierarchical cache memory [20, section 2.6], execution pipelining, and parallelism. It is only a small exaggeration to say that executing level 3 BLAS is what modern computers do best. Many manufacturers supply hand-tuned, extraordinarily efficient implementations of the BLAS. Automatically tuned versions of the BLAS [61] also perform well. It is the ability to exploit matrix-matrix multiplies that makes spectral splitting methods attractive competitors to the QR algorithm [2, 5, 6, 7].

The small-bulge multishift QR algorithm which we propose attains much of its efficiency through the level 3 BLAS.

1.1.2. Notation. Throughout this paper we use the following notation and definitions.

1. We will use the "colon notation" to denote submatrices: $H_{i:j,k:l}$ is the submatrix of matrix H in rows i–j and columns k–l inclusively. The notation $H_{:,k:l}$ indicates the submatrix in columns k–l inclusively (and all rows). The notation $H_{i:j,:}$ indicates the submatrix in rows i–j inclusively (and all columns).

2. A *quasi-triangular* matrix is a real, block triangular matrix with 1-by-1 and 2-by-2 blocks along the diagonal.

3. A matrix $H \in \mathbf{R}^{n\times n}$ is in Hessenberg form if $h_{ij} = 0$ whenever $i > j+1$. The matrix H is said to be unreduced if, in addition, the subdiagonal entries are nonzero, i.e., $h_{ij} \neq 0$ whenever $i = j+1$.

4. Following [27, p. 19], we define a "flop" as a single floating point operation, i.e., either a floating point addition or a floating point multiplication together with its associated subscripting. The Fortran statement

$$\texttt{C(I,J)} = \texttt{C(I,J)} + \texttt{A(I,K)} * \texttt{B(K,J)}$$

involves two flops.

In some of the examples in section 3, we report an automatic hardware count of the number of floating point instructions executed. Note that a trinary multiply-add instruction counts as just one executed instruction in the hardware count even though it executes two flops. Thus, depending on the compiler and optimization level, the above Fortran statement could be executed using either one floating point instruction or two floating point instructions (perhaps along with some integer subscripting calculations) even though it involves two flops.

2. A small-bulge multishift *QR* algorithm. If there are many simultaneous shifts, then most of the work in a QR sweep may be organized into matrix-vector and matrix-matrix multiplies that take advantage of the higher level BLAS [4]. However, to date, the performance of the large-bulge multishift QR has been disappointing [24, 55, 56, 58]. Accurate shifts are essential to accelerate the convergence of the QR algorithm. In the large-bulge multishift QR algorithm rounding errors "blur" the shifts and retard convergence [56, 58]. The ill effects of shift blurring grow rapidly with the size of the diagonal bulge, effectively limiting the large-bulge multishift QR algorithm to roughly 10 simultaneous shifts—too few to take full advantage of level 3 BLAS. See Figure 1.

Watkins explains much of the mechanics of shift blurring with the next theorem [56, 58].

THEOREM 2.1 (see [58]). *Consider a Hessenberg-with-a-bulge matrix that occurs during a multishift QR sweep using m pairs of simultaneous shifts. Let $\tilde{B}$ be the $2(m+1)$-by-$2(m+1)$ principal submatrix containing the bulge. Obtain B from $\tilde{B}$ by dropping the first row and the last column. Let N be a $(2m+1)$-by-$(2m+1)$ Jordan block with eigenvalue zero. The shifts are the $2m$ finite eigenvalues of the bulge pencil*

$$B - \lambda N. \tag{2.1}$$

The theorem shows that the shift information is transferred through the matrix by the bulge pencils (2.1). Watkins observes empirically that the eigenvalues of $B - \lambda N$ tend be ill-conditioned and that the ill-conditioning grows rapidly with the size of the bulge. The observation suggests that in order to avoid shift blurring, the bulges must be confined to very small order. This is borne out by many successful small bulge versions of the QR algorithm including the one described in this paper. See, for example, [17, 24, 31, 29, 34, 36, 54, 55].

In order to apply complex conjugate shifts in real arithmetic, one must allow bulges big enough to transmit at least two shifts. The smallest such bulges occupy full 4-by-4 principal submatrices. We sometimes refer to 4-by-4 bulges as "double implicit shift bulges" because they are used in the double implicit shift QR algorithm [25, 26], [27, Algorithm 7.5.1]. This paper uses double implicit shift bulges exclusively.

The QR algorithm needs to use many simultaneous shifts in order to generate substantial level 3 BLAS operations, but, in order to avoid shift blurring, it is restricted to small bulges. Small bulges transmit only a few simultaneous shifts. One way to resolve these superficially contradictory demands is to use a chain of m tightly packed two-shift bulges as illustrated in Figure 2. This configuration allows many simultaneous shifts while keeping each pair "well focused" inside a small bulge.

The processes of chasing the chain of bulges along the diagonal is illustrated in Figure 3. Near the diagonal, "3-by-3" Householder reflections are needed to chase the bulges one at a time along the diagonal. The reflections are applied individually in the light shaded region near the diagonal in a sequence of level 1 BLAS operations.

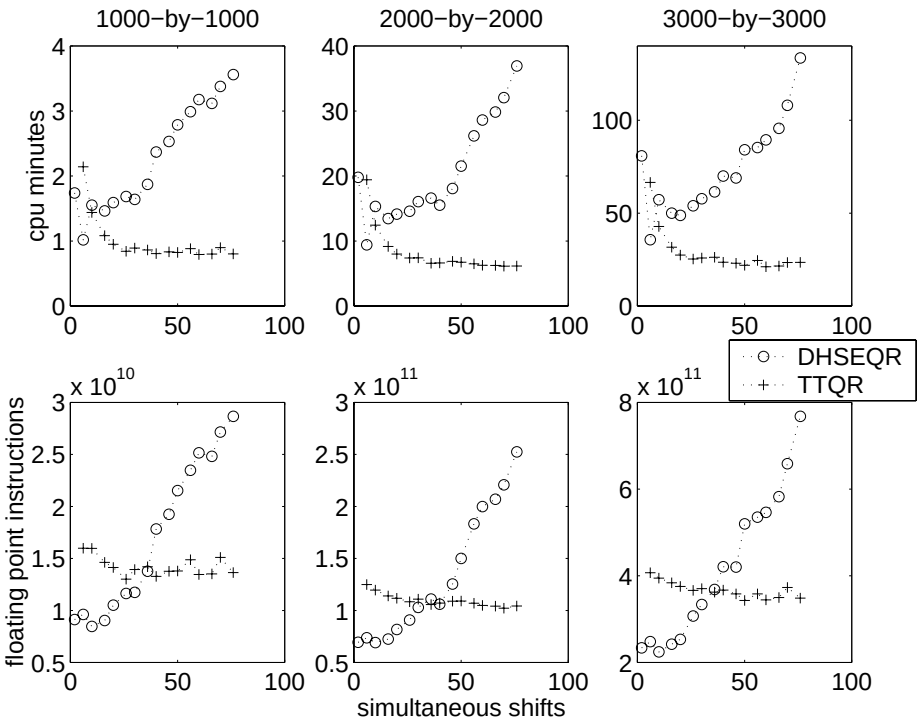


FIG. 1. *Effect of varying numbers of simultaneous shifts on the large-bulge multishift QR algorithm as implemented in* `DHSEQR`*, from LAPACK version* 2 *and on our experimental implementation of the small-bulge multishift QR,* `TTQR`*. The two programs calculated the Schur decompositions (including both the orthogonal and quasi-triangular factors) of* 1,000*-by-*1,000*,* 2,000*-by-*2,000 *and* 3,000*-by-*3,000 *pseudorandom Hessenberg matrices. (The pseudorandom test matrices, computational environment, and other details are described in section* 3*.) The first row plots number of simultaneous shifts versus cpu-minutes. The second row plots number of simultaneous shifts versus a hardware count of floating point instructions executed. (A trinary multiply-add instruction counts as one instruction but executes two flops.) The first, second, and third columns report data from matrices of order* $n = 1{,}000$*,* $n = 2{,}000$ *and* $n = 3{,}000$*, respectively.*

Periodically, a group of reflections are accumulated into a thickly banded orthogonal matrix. The orthogonal matrix is then used to update the dark shaded region in Figure 3 using matrix-matrix multiplication or other level 3 BLAS operations.

We refer to this as a "two-tone" QR sweep. The level 1 BLAS operations form one tone; the level 3 operations form the other.

Using the implicit Q theorem [27, Theorem 7.4.3], it is easy to show that a two-tone small-bulge multishift QR sweep with m pairs of simultaneous shifts, $(s_j, t_j) \in \mathbf{C} \times \mathbf{C}$, $j = 1, 2, 3, \ldots, m$, is equivalent to m Francis double implicit shift QR sweeps, i.e., a two-tone QR sweep overwrites H by $Q^T H Q$ where $\prod_{j=1}^{m}(H - s_j)(H - t_j) = QR$ is a QR (orthogonal-triangular) factorization. An examination of the detailed description below shows that, in exact arithmetic, the two-tone QR sweep generates the same bulges and the same sequence of similarity transformations by "3-by-3" Householder reflections as m double implicit QR sweeps using the same shifts.

Modified QR algorithms that chase more than one bulge at a time have been proposed several times [17, 24, 31, 29, 34, 36, 54, 55]. In particular, Henry, Watkins, and Dongarra [31] recently developed a distributed memory parallel QR algorithm

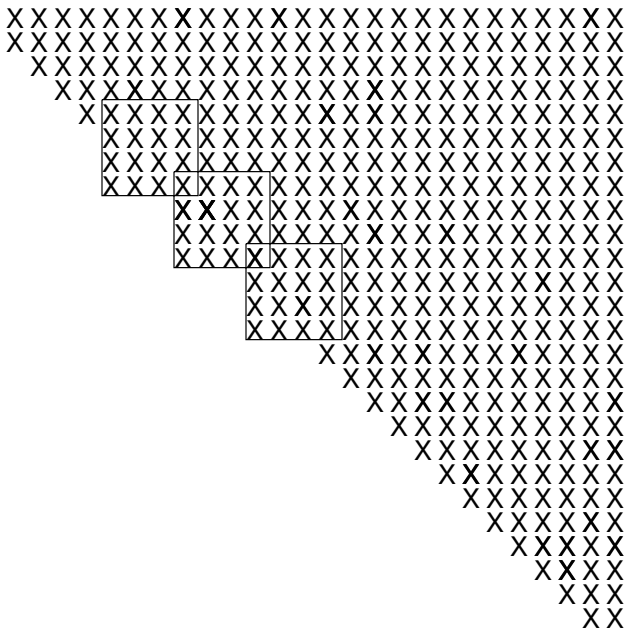


FIG. 2. *A chain of $m = 3$ tightly packed double implicit shift bulges in rows $r = 5$ through $s = r + 3m = 14$.*

that chases many small bulges independently. Lang [36] recently described symmetric QL and QR algorithms that use rotations to simultaneously chase many small bulges. His algorithms also achieve some parallelism and good cache reuse by accumulating the rotations in groups. The algorithm proposed in the next section uses Householder reflections and applies to nonsymmetric matrices.

2.1. Detailed description. This section describes the two-tone small-bulge multishift QR sweep. Let $H \in \mathbf{R}^{n\times n}$ be an unreduced Hessenberg matrix and let $(s_j, t_j) \in \mathbf{C} \times \mathbf{C}$, $j = 1, 2, 3, \ldots, m$, be a set of m pairs of shifts. To avoid complex arithmetic, we require each pair to be closed under complex conjugation, i.e., either $\bar{s}_j = t_j$ or $(s_j, t_j) \in \mathbf{R} \times \mathbf{R}$.

There are three phases in a two-tone QR algorithm. In the first phase, a tightly packed chain of 4-by-4, double implicit shift bulges is introduced into the upper left-hand corner of a Hessenberg matrix. In the second phase, the chain of bulges is chased down the diagonal from the upper left-hand corner to the lower right-hand corner by a sequence of reflections. In the final phase, the matrix is returned to Hessenberg form by chasing the chain off the bottom row.

2.1.1. Phase 2: The central computation. The bulk of the work is in the second phase. Suppose that $H \in \mathbf{R}^{n\times n}$ is an upper Hessenberg matrix with a packet of m double implicit shift bulges in rows r through $s = r + 3m$. See Figure 3. Using a process called "bulge chasing" that has been known at least since the double implicit shift algorithm [35, 25, 26, 27] was introduced in 1961, one may use a sequence of similarity transformations by "3-by-3" Householder reflections to chase the chain of bulges down k rows, one bulge at a time, starting with the lowest bulge. If $\tilde{U} \in \mathbf{R}^{n\times n}$

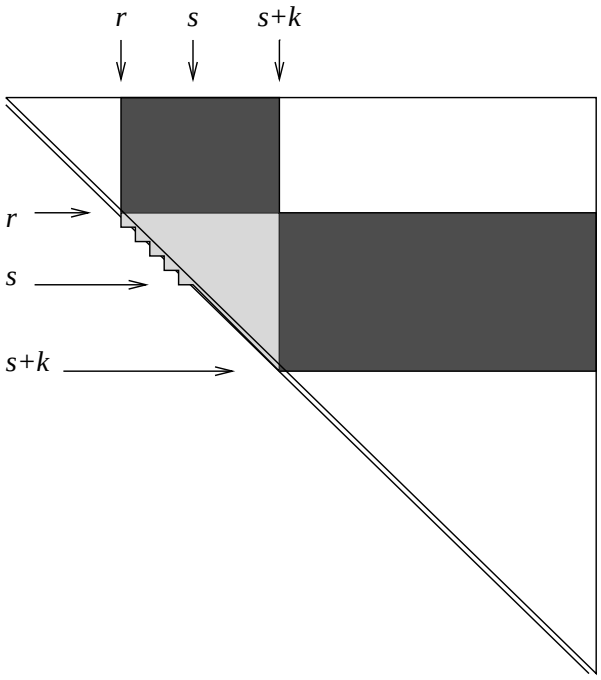


FIG. 3. *Chasing a packet of m 4-by-4 double implicit shift bulges from rows r through $s = r+3m$ down to rows $r+k$ through $s+k$.*

is the product of this sequence of Householder reflections, then $\tilde{U}$ has the form

$$\tilde{U} = \begin{array}{c} \\ r \\ 3m+k-1 \\ n-(s+k)+1 \end{array} \begin{array}{c} \begin{array}{ccc} r & 3m+k-1 & n-(s+k)+1 \end{array} \\ \begin{bmatrix} I & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & I \end{bmatrix} \end{array}, \tag{2.2}$$

where $U = \tilde{U}_{r+1:s+k-1,r+1:s+k-1}$ is itself an orthogonal matrix. For notational convenience, define $\hat{U} \in \mathbf{R}^{(3m+k+1)\times(3m+k+1)}$ by

$$\hat{U} = \begin{array}{c} \\ 1 \\ 3m+k-1 \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3m+k-1 & 1 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & U & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array},$$

i.e., the $r = 1$, $n = s + k$ case of (2.2).

In terms of U and $\hat{U}$, the multibulge chase mentioned above consists of two parts. The first part is a similarity transformation on the the lightly shaded region in Figure 3, i.e.,

$$H_{r:s+k,r:s+k} \leftarrow \hat{U}^T H_{r:s+k,r:s+k} \hat{U}. \tag{2.3}$$

The second part is the matrix multiplication of the dark, horizontal slab by $\hat{U}^T$ from the left, i.e.,

$$H_{r:s+k,s+k+1:n} \leftarrow \hat{U}^T H_{r:s+k,s+k+1:n}, \tag{2.4}$$

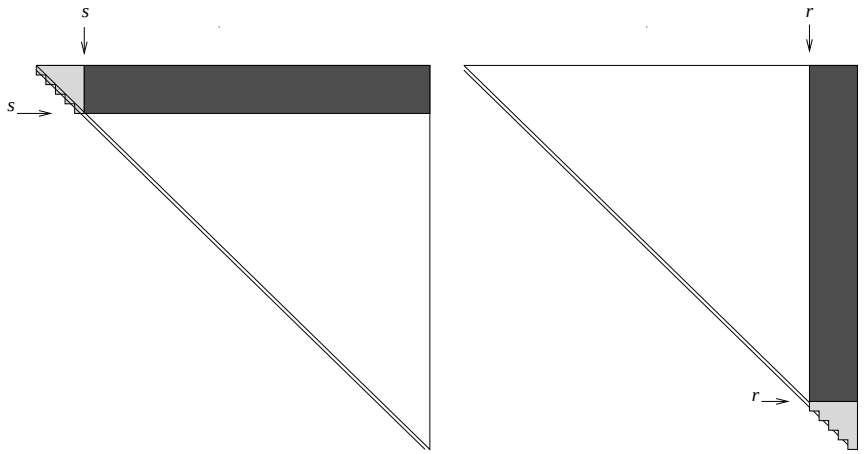


FIG. 4. *Left: introducing m bulges into row 1 through row $s = 1+3m$. Right: chasing m bulges off the bottom from row $r = n-(1+3m)$ through row n.*

and the multiplication of the dark, vertical slab by $\hat{U}$ from the right, i.e.,

$$H_{1:r-1,r:s+k} \leftarrow H_{1:r-1,r:s+k}\hat{U}. \tag{2.5}$$

The similarity transformation (2.3) has the effect of moving the packet of bulges from rows r through s down to rows $r+k$ through $s+k$. Using a sequence of "3-by-3" reflections, this small similarity transformation does not take advantage of the level 3 BLAS, but if $m \ll n$ and $k \ll n$, then the work of updating the lightly shaded region in Figure 3 is small compared to the work of updating the two dark slabs. Updating the dark horizontal and vertical slabs in Figure 3 is a matrix multiplication by the orthogonal matrix U in (2.2). In order to make this into a level 3 BLAS matrix-matrix multiplication, it is necessary to form explicitly the orthogonal submatrix U from the "3-by-3" reflections used to chase the bulges. Accumulating U does not take advantage of the level 3 BLAS either, but this is also a small amount of arithmetic work compared to updating the dark slabs. Using a sequence of reflections to update the lightly shaded region in Figure 3 but using matrix-matrix multiplication to update the dark bands results in a high ratio of level 3 BLAS work to level 1 BLAS work [13].

2.1.2. Phases 1 and 3: Getting started and finishing up. The two-tone structure of the algorithm is slightly simpler during the initial bulge-introduction phase and during the final chasing-bulges-off-the-bottom phase than during most of the computation. See Figure 4. A modified version of the Francis double implicit shift algorithm may be used to introduce the m bulges into the upper left-hand corner of the Hessenberg matrix. One at a time, bulges are introduced and chased down to their proper position in the chain. This is equivalent to doing $3m(m-1)/2$ similarity transformations by "3-by-3" reflections on $H_{1:3m+1,1:3m+1}$, the lightly shaded region on the left of Figure 4. The matrix U has zero structure similar to the $(1,1)$ block of Figure 5.

Chasing the packet of m bulges off the bottom at the end of the QR sweep is similar. It is equivalent to doing a similarity transformation by a sequence of Householder reflections followed by a orthogonal matrix multiplication on the dark vertical slab on the right of Figure 4. This operation also requires roughly the same number of flops as does the operation of introducing a packet of m bulges.

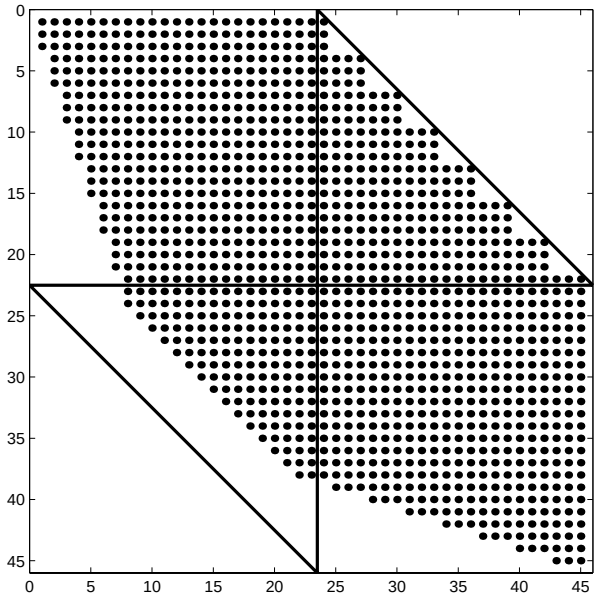


FIG. 5. *The zero structure of the orthogonal submatrix U in* (2.2) *in the special case $m = 8$, $k = 22$.*

2.1.3. Level 3 BLAS and exploiting the zero structure of *U*. The orthogonal matrix U has more structure than a generic, dense orthogonal matrix. The amount of arithmetic work needed to update the dark slabs can, theoretically, be substantially reduced by taking advantage of the zero structure of U. The example in Figure 5 is 38% zeros. However, this adds some complexity to the computation which offsets the reduced flop count. Depending on the computing environment, taking advantage of some or all of the zeros may add more work than it saves. Nevertheless, it is likely to be worth taking advantage of at least some of the zeros either by treating U as a thick band matrix or as a 2-by-2 block matrix.

As illustrated in Figure 5, U has a thick band structure with lower bandwidth $k_l = 2\min(m, k)$ and upper bandwidth $k_u = k + 1$. The band is fairly dense, so, theoretically, the thick, banded matrix-matrix multiply is a level 3 operation. However, except for triangular matrix multiplication, banded matrix-matrix multiplication software is usually implemented as a level 2 (matrix-vector) operation.

In order to take advantage of both band structure and the current set of level 3 BLAS, matrix multiplication by U may be broken up into two dense-by-dense matrix multiplications and two triangular-by-dense matrix multiplications. If $3m - 2 \geq k \geq m + 2$, then one way to do this is to partition U into a 2-by-2 block matrix with two nearly dense, rectangular diagonal blocks and two triangular off-diagonal blocks. See Figure 5. In addition, at least one of the triangular blocks has a thick band of zeros along the diagonal, so the work of multiplying by this triangular matrix can be further reduced.

For a fixed choice of the number of simultaneous shifts $2m$, $k \approx 3m$ approximately minimizes the number of flops in an n-by-n two-tone multibulge QR sweep regardless of the logical zero structure of U. See Table 1. With $k \approx 3m$, the algorithm uses between 1.6 and 2.4 times as many flops as m double implicit shift sweeps, depending

TABLE 1

The ratio of the number of flops performed by two-tone small-bulge multishift QR sweep with m pairs of shifts chased k rows at a time to $10mn^2$ the number of flops needed for m double implicit shift QR sweeps. It is assumed that m is small relative to the order of the Hessenberg matrix n.

	flops $\times 10mn^2$		
Logical U structure	Ratio for $k = 3m$	Approximately optimal k	Ratio for optimal k
Banded	1.57	$k = 2.8m$	1.57
2-by-2 block	1.63	$k \approx 2.54m$	1.62
Dense	2.40	$k = 3m$	2.40

TABLE 2

Flops used by an n-by-n two-tone multibulge QR sweep chasing m bulges, k rows at a time, assuming $k \ll n$ and $m \ll n$. (These estimates do not include the work for accumulating the orthogonal matrix of Schur vectors Q. Accumulating Q approximately doubles the flop count.)

Logical U structure		Flops $+ O(m^2n)$
2-by-2 block	$k = 3m - 2$	$\left(\frac{49m^2-39m+12}{3m-2}\right) n^2$
$m > k$ or	$k = pm$ $(3+p)m$ even	$\left(\frac{(2p^2+6p+13)m^2-pm+2}{pm}\right) n^2$
$3m - 2 \geq k \geq m + 1$	$k = pm$ $(3+p)m$ odd	$\left(\frac{(2p^2+6p+13)m^2+(p-4)m+3}{pm}\right) n^2$
Dense	$k = 3m$	$\left(\frac{2(6m-1)^2}{3m}\right) n^2$
	$k = pm$	$\left(\frac{2((p+3)m-1)^2}{pm}\right) n^2$

on how much of the zero structure of U is exploited. The version of the large-bulge multishift QR algorithm which most closely resembles the two-tone small-bulge multishift QR algorithm aggregates groups of p transformations in order to take advantage of the level 3 BLAS. A $2m$ shift sweep of the aggregated large-bulge multishift QR algorithm needs roughly $1.5 + p/(8m)$ times as many flops as m double implicit shift sweeps [4]. (The version of the large-bulge multishift QR algorithm implemented in `DHSEQR` from LAPACK version 2.0 [1] does not aggregate groups of transformations and takes advantage of only the level 1 and 2 BLAS. It needs roughly the same number of flops to chase a single $2m$ shift bulge as m double implicit shift sweeps [4, section 6].) Detailed algorithms and mathematical flop counts are given in [13] and are summarized here in Tables 2 and 1. In section 3, numerical examples demonstrate that the avoidance of shift blurring by using small bulges and the ability to use level 3 BLAS by using many simultaneous shifts per QR sweep more than overcome the greater flop counts per sweep displayed in Tables 2 and 1.

2.2. Deflation and choice of shifts. It is inexpensive to monitor the subdiagonal entries during the course of a two-tone QR sweep and take advantage of a small-subdiagonal deflation should one occur. Note that this includes taking advantage of any small subdiagonal entries that may appear between the bulges during a two-tone QR sweep. In [55], this is called "vigilant deflation."

Ordinarily, the bulges above a vigilant deflation collapse as they encounter the newly created subdiagonal zero. This may block those shifts so that, for the current QR sweep, the benefit of using many simultaneous shifts may be lost. However, the bulges can be reintroduced in the row in which the new zero subdiagonal appears

using the same methods as are used to introduce bulges at the upper left-hand corner [27, p. 377], [62, p. 530]. In this way, the shift information passes through a zero subdiagonal and the two-tone QR sweep continues with all its shifts.

It is well known that a good choice of shifts in the QR algorithm leads to rapid convergence. Shifts selected to be the eigenvalues of a trailing principal submatrix give local quadratic convergence [62, 60]. In the symmetric case, convergence is cubic [64]. Deferred shifts retard convergence [49]. Following the choice of LAPACK version 2 subroutine `DHSEQR`, our experimental computer programs select the shifts to be the eigenvalues of a trailing principal submatrix.

3. Numerical examples. We compared the execution time of the large- and small-bulge QR algorithm on ad hoc and pseudorandom Hessenberg matrices of order $n = 1{,}000$ to $n = 10{,}000$ and on nonrandom matrices of similar order taken from a variety of applications in science and engineering [3]. (The two algorithms delivered roughly the same accuracy as measured by the relative residual $\|A\tilde{Q} - \tilde{Q}\tilde{T}\|_F/\|A\|$ and the departure from orthogonality $\|\tilde{Q}^T\tilde{Q} - I\|_F/\sqrt{n}$, where $A = \tilde{Q}\tilde{T}\tilde{Q}^T$ is the computed real Schur decomposition of A with computed orthogonal factor $\tilde{Q}$ and computed quasi-triangular factor $\tilde{T}$.)

We call our experimental Fortran implementation of the two-tone small-bulge multishift QR algorithm `TTQR`. When using m pairs of simultaneous shifts, each `TTQR` sweep chases each tightly packed chain of m small bulges $k = 3m - 2$ rows at a time. The implementation exploits the logical 2-by-2 block structure of U including the thick band of zeros along the diagonal of the (1,2) block in Figure 5. Except where noted otherwise, `TTQR` uses 150 simultaneous shifts, but no particular significance should be attached to this choice. `TTQR` uses LAPACK subroutine `DHSEQR` [1], the conventional large-bulge QR algorithm, to reduce diagonal subblocks of order no greater than 1.5 times the number of simultaneous shifts. Following EISPACK [46] and LAPACK [1], a Hessenberg subdiagonal entry $h_{i+1,i}$ is set to zero when $|h_{i+1,i}| \leq \varepsilon\,(|h_{ii}| + |h_{i+1,i+1}|)$ with ε equal to the unit round. (A new deflation procedure which greatly reduces the arithmetic work and execution time that `TTQR` needs is reported in [12, 13].)

For comparison, we used the implementation of the large-bulge multishift QR algorithm in subroutine `DHSEQR` from LAPACK version 2 [1], which is widely recognized to be an excellent implementation. `DHSEQR` uses the double implicit shift QR algorithm to reduce subblocks of size `MAXB`-by-`MAXB` or smaller. In the examples reported here, `MAXB` is chosen to be the greater of 50 and the number of simultaneous shifts. Except where noted otherwise, `DHSEQR` uses only six simultaneous shifts which Figure 1 shows is approximately optimal.

The examples reported here were run on an Origin2000 computer equipped with 400MHz IP27 R12000 processors and 16 gigabytes of memory. Each processor has 32 kilobytes of level 1 instruction cache, 32 kilobytes of level 1 data cache, and 8 megabytes of level 2 combined instruction and data cache. For serial execution, the experimental Fortran implementation of the small bulge multishift QR algorithm was compiled with version 7.30 of the MIPSpro Fortran 77 compiler with options `-64 -TARG:platform=ip27 -Ofast=ip27 -LNO`. For comparison purposes the same options were used to compile `DHSEQR` from LAPACK version 2. (We observe that this compilation of `DHSEQR` is usually slightly faster than the compilation distributed in the SGI/Cray Scientific Library version 1.2.0.0.) For parallel execution the `-mp` and `-pfa` options were added. The programs called optimized BLAS subroutines from the SGI/Cray Scientific Library version 1.2.0.0. In our computational environment, we observed that the measured serial "cpu time" of any particular program with

its particular data might vary by at most a few percent. We were fortunate to get exclusive use of several processors for the purpose of timing parallel benchmark runs.

Except where otherwise mentioned, n-by-n matrices were stored in n-by-n arrays.

We ran the experimental program on both pseudorandomly generated and nonrandom non-Hermitian matrices of order $n = 1{,}000$ to $n = 10{,}000$. We selected the entries on the diagonal and upper triangle of pseudorandomly generated Hessenberg matrices to be normally distributed pseudorandom numbers with mean zero and variance 1. We set the subdiagonal entry $h_{j+1,j} = \sqrt{\chi^2_{n-j}}$, where χ^2_{n-j} is selected from a Chi-squared distribution with $n - j$ degrees of freedom. Hessenberg matrices with the same distribution of random entries can also be generated by applying the reduction to Hessenberg form algorithm [27, Algorithm 7.4.2] to full n-by-n matrices with pseudorandom entries selected from the normal distribution with mean zero variance one.

Our source of nonrandom matrices was the Non-Hermitian Eigenvalue Problems (NEP) collection [3]. This is a collection of eigenvalue problems from a variety of applications in science and engineering. We selected 21 real eigenvalue problems of order 1,000-by-1,000 to 8,192-by-8,192.

In some of the examples reported below, we report an automatic hardware count of the number of floating point instructions executed. Note that a trinary multiply-add instruction counts as just one executed instruction in the hardware count even though it executes two flops.

We also report the floating point execution rate in millions of floating point instructions per second, or "mega-flops" for short. For comparison purposes, we measured the floating point execution rate of the level 3 BLAS full matrix-matrix multiply subroutine `DGEMM` and the triangular matrix-matrix multiply subroutine `DTRMM` from the SGI/Cray Scientific Library version 1.2.0.0 applied to matrix-matrix products similar to updating the dark bands in Figure 3. In serial execution, in the Origin2000 computational environment described above, `DGEMM` computes the product of the transpose of a 200-by-200 matrix times a 200-by-10,000 slab embedded a 10,000-by-10,000 array at roughly 330 mega-flops. It computes the product of a 10,000-by-200 slab times a 200-by-200 matrix at roughly 325 mega-flops. `DTRMM` computes the product of the transpose of a triangular 200-by-200 matrix times a 200-by-10,000 slab at roughly 305 mega-flops if it is an upper triangular matrix and at roughly 260 mega-flops if it is a lower triangular matrix. `DTRMM` computes the product of a 10,000-by-200 slab times a 200-by-200 triangular matrix at roughly 275 mega-flops if it is an upper triangular matrix and at roughly 255 mega-flops if it is a lower triangular matrix.

Example 1. Figure 1 displays the serial execution time of `DHSEQR` and `TTQR` using various numbers of simultaneous shifts to calculate the Schur decomposition of the pseudorandom Hessenberg matrices described above. Both the quasi-triangular and orthogonal Schur factors were computed. The figure demonstrates that the convergence rate of `TTQR` does not suffer from shift blurring even with hundreds of simultaneous shifts. The figure also displays the total number of floating point instructions executed by `DHSEQR` and `TTQR` to calculate the Schur decompositions of pseudorandom Hessenberg matrices. It also demonstrates that the number of floating point operations needed by `TTQR` changes little as the number of simultaneous shifts varies in a range that is small compared to n, the order of the Hessenberg matrix. The number of simultaneous shifts used by `TTQR` may be chosen to fit the cache size and other machine dependent parameters with little effect on the total amount of arithmetic work. However, in this example, `TTQR` executes between 1.5 and 2 times as many floating point instructions as `DHSEQR` with 6 simultaneous shifts. A partial explanation for

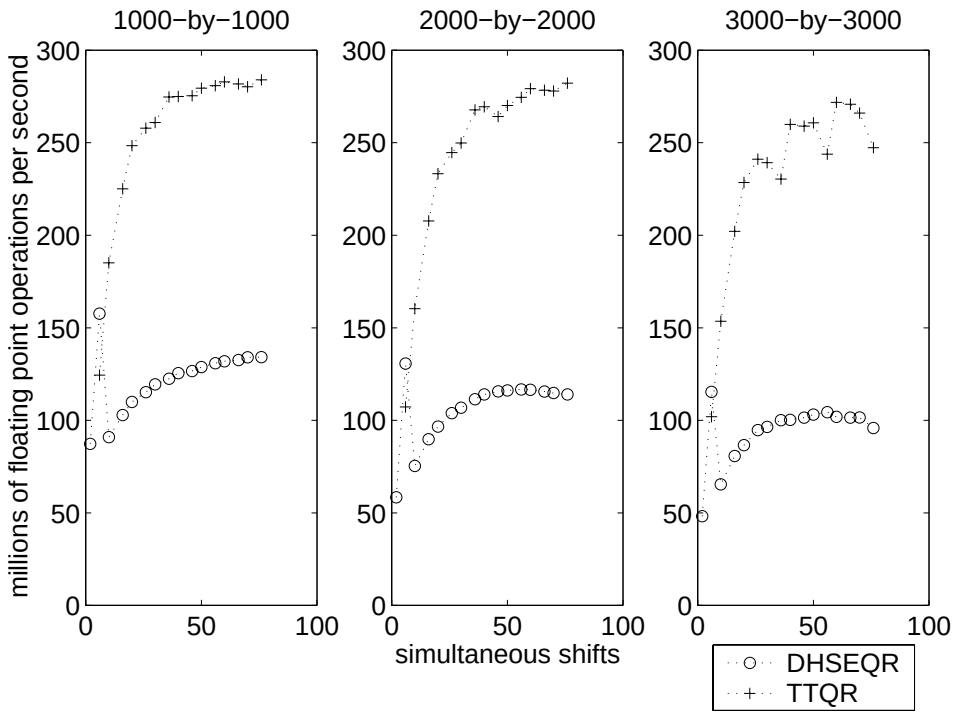


FIG. 6. *Floating point execution rate of* `DHSEQR`*, the large-bulge multishift QR algorithm, and* `TTQR`*, the small-bulge multishift QR algorithm, using varying numbers of simultaneous shifts to calculate the Schur decompositions of the pseudorandom Hessenberg matrices described in section* 3.

this is that `TTQR` executes more floating point instructions per shift than `DHSEQR`. See Table 1. However, in other examples `TTQR` executes fewer floating point instructions than `DHSEQR`. See Figures 7 and 10.

Figure 6 displays the floating point execution rates achieved on the Origin2000 computer described above. It demonstrates that `TTQR` attains substantially higher rates of execution of floating point instructions when using many simultaneous shifts.

On the Origin2000 computer described above, we observe that, on pseudorandom Hessenberg matrices of order roughly $n > 500$, `TTQR` with forty simultaneous shifts calculates the Schur decomposition of pseudorandom Hessenberg faster than `DHSEQR` with six simultaneous shifts. (For matrices of order $n = 100$, `TTQR` with 40 simultaneous shifts uses 170% of the execution time of `DHSEQR` with six simultaneous shifts.)

Example 2. Figure 7 shows the execution time, floating point execution rate, and number of floating instructions executed to calculate the Schur decompositions of ad hoc Hessenberg matrices of the form of

$$S_6 = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 0.001 & 1 & 0 & 0 & 0 & 0 \\ & 0.001 & 2 & 0 & 0 & 0 \\ & & 0.001 & 3 & 0 & 0 \\ & & & 0.001 & 4 & 0 \\ & & & & 0.001 & 5 \end{bmatrix}. \tag{3.1}$$

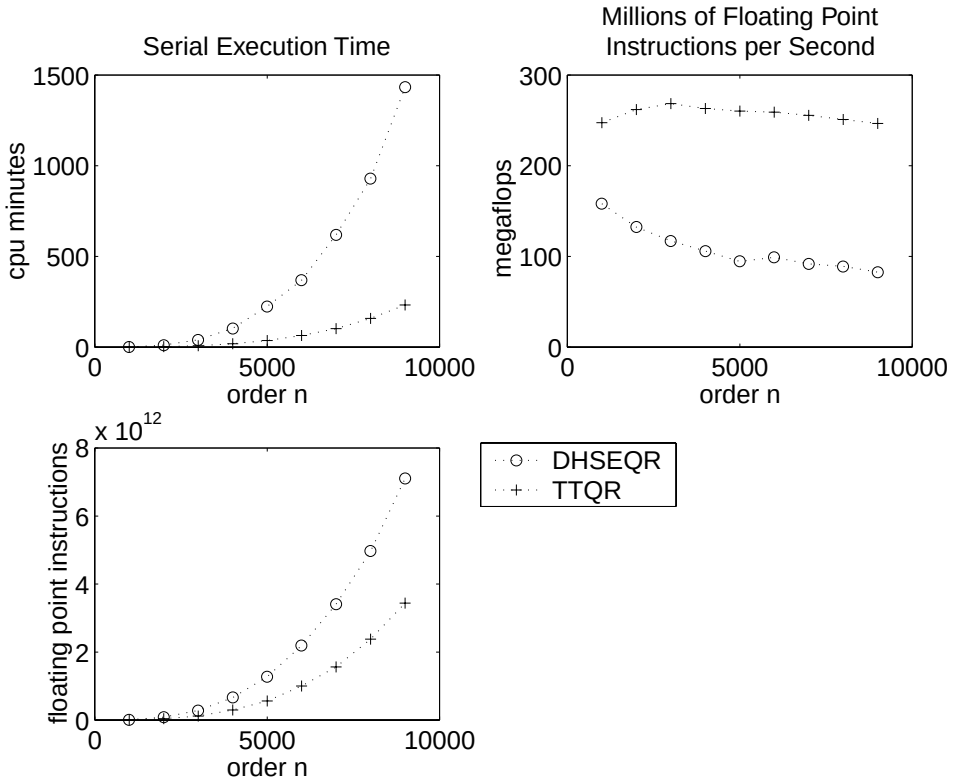


FIG. 7. *Execution time, floating point execution rate, and hardware count of floating point instructions executed to calculate the Schur decomposition (including both orthogonal and quasi-triangular factors) of Hessenberg matrices of the form of* (3.1).

The eigenvalues of trailing principal submatrices of S_n are extraordinarily close approximations to eigenvalues of the whole matrix S_n [13, 12], so, at least initially, the shifts used by both `DHSEQR` and `TTQR` are exceptionally good. Both `DHSEQR` and `TTQR` calculate the Schur decompositions of S_n faster than the Schur decomposition of a pseudorandom Hessenberg matrix of the same size. See Figure 8. However, `TTQR` finishes substantially sooner, maintains a higher floating point execution rate, and executes fewer floating point instructions.

Example 3. Figure 8 compares the serial execution time and floating point instruction execution rate of `TTQR` and `DHSEQR` applied to pseudorandom Hessenberg matrices of order $n = 1{,}000$ to $n = 10{,}000$. In this example, `TTQR` consistently executes more floating point instructions than `DHSEQR` but overcomes this handicap by maintaining a high floating point execution rate. Note that in other examples `TTQR` executes fewer floating point instructions than `DHSEQR`. See Figures 7 and 10.

The advantage of `TTQR` over `DHSEQR` remains qualitatively similar even if only eigenvalues are computed or if only one of the orthogonal or quasi-triangular Schur factors is computed.

Example 4. Figure 9 displays the serial execution times of `DHSEQR` and `TTQR` applied to 20 real non-Hermitian eigenvalue problems from the NEP collection [3].

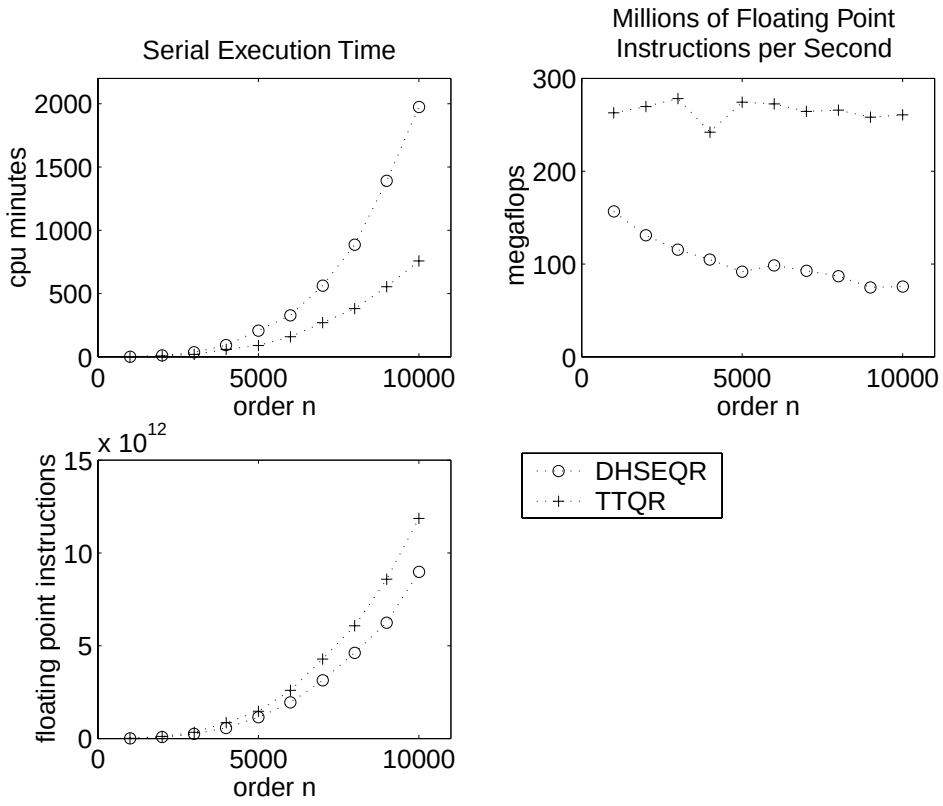


FIG. 8. *Serial execution times, floating point execution rates, and hardware counts of floating point instructions of* `DHSEQR`*, the large-bulge multishift QR algorithm, and* `TTQR`*, the small-bulge multishift QR algorithm. The two subroutines calculated Schur decompositions (including both the orthogonal and quasi-triangular factors) of pseudorandom Hessenberg matrices as described in section 3.*

To avoid cache conflicts, each n-by-n matrix was stored in an $(n + 7)$-by-$(n + 7)$ array. Each matrix is identified by an acronym from [3]. The alphabetic part indicates the application from which the matrix comes. The numerical part gives the order of the matrix. Summaries of the applications, descriptions of the matrices, and references can be found in [3]. `DHSEQR` uses six simultaneous shifts which Figure 1 shows is approximately optimal. `TTQR` uses 60 simultaneous shifts on 1,000-by-1,000 to 1,999-by-1,999 matrices, 116 simultaneous shifts on 2,000-by-2,000 to 2,499-by-2,499 matrices, 150 simultaneous shifts on 2,500-by-2,500 to 3,999-by-3,999 matrices, and 180 simultaneous shifts on 4,000-by-4,000 or larger matrices.

Notice that the superiority of `TTQR` is usually greater for matrices of larger order.

We also mention that DW8192 is not reported in Figure 9 only because the execution times are out of scale with the other reported times. `DHSEQR` calculated the Schur decomposition of the Hessenberg matrix derived from DW8192 in 544 cpu minutes. `TTQR` calculated the Schur decomposition in 301 cpu minutes.

Figure 10 displays hardware counts of the number of floating point instructions executed by `DHSEQR` and `TTQR`. Note that neither `TTQR` nor `DHSEQR` consistently executes more floating point instructions.

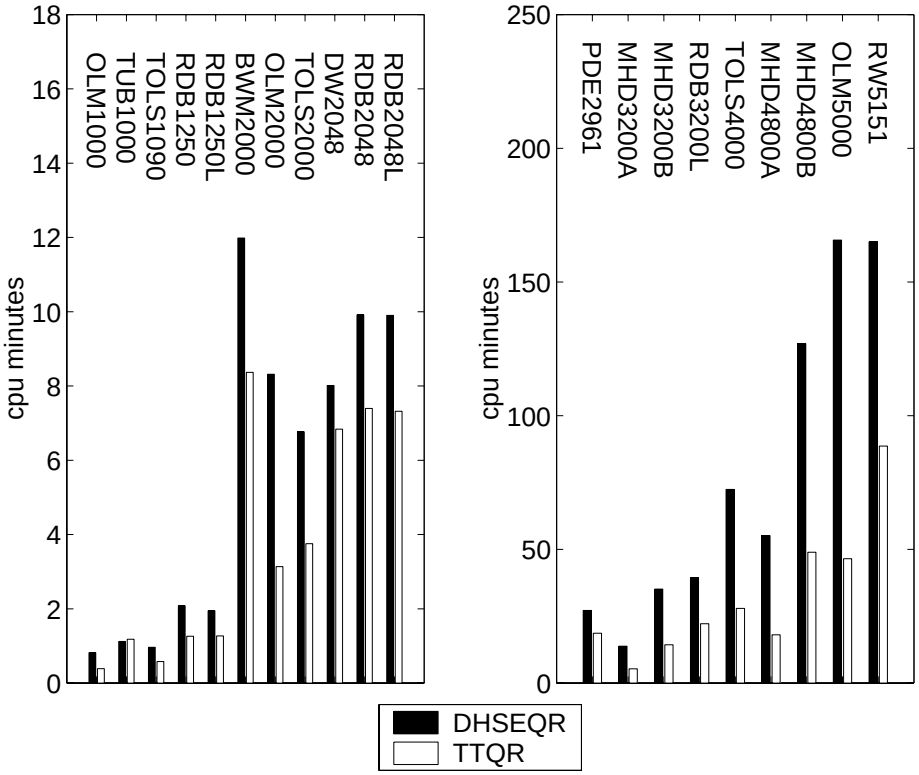


FIG. 9. *Serial execution times of* `DHSEQR`*, the large-bulge multishift QR algorithm, and* `TTQR`*, the small-bulge multishift QR algorithm, applied to* 20 *non-Hermitian eigenvalue problems from* [3]*. The acronyms indicate the corresponding matrix from* [3]*.* `DHSEQR` *uses six simultaneous shifts which Figure* 1 *shows is approximately optimal.* `TTQR` *uses* 60 *simultaneous shifts on* 1,000-*by*-1,000 *to* 1,999-*by*-1,999 *matrices,* 116 *simultaneous shifts on* 2,000-*by*-2,000 *to* 2,499-*by*-2,499 *matrices,* 150 *simultaneous shifts on* 2,500-*by*-2,500 *to* 3,999-*by*-3,999 *matrices, and* 180 *simultaneous shifts on* 4,000-*by*-4,000 *or larger matrices. (The execution time of the reduction to Hessenberg form is not included in the above graphs.)*

Example 5. Our experimental implementation of the small-bulge QR algorithm is not designed for parallel computation. However, `TTQR` makes heavy use of the level 3 BLAS—particularly matrix-matrix multiply. Hence, it is not surprising to observe modest but not insignificant speedups when the experimental version of `TTQR` is compiled for parallel execution and linked with parallel versions of the BLAS.

Figure 11 shows wall clock execution time, parallel speedup, and parallel efficiency of `TTQR` calculating the Schur decomposition (including both the quasi-triangular and orthogonal factors) of pseudorandom Hessenberg matrices on the Origin2000 computer described above. To avoid cache conflicts, some n-by-n matrices were stored in $(n+1)$-by-$(n+1)$ arrays. (Parallel speedup is the ratio T_1/T_p, where T_1 is the 1 processor wall clock execution time and T_p is the p-processor wall clock execution time. Parallel efficiency is $T_1/(pT_p)$.)

A major serial bottle neck can be attributed to computations in the lightly shaded region of Figure 3. There is potential for some small grain parallel execution even in this region, but there is not enough parallel work to overcome the overhead associated with synchronizing the processors. A planned production version of `TTQR` is expected

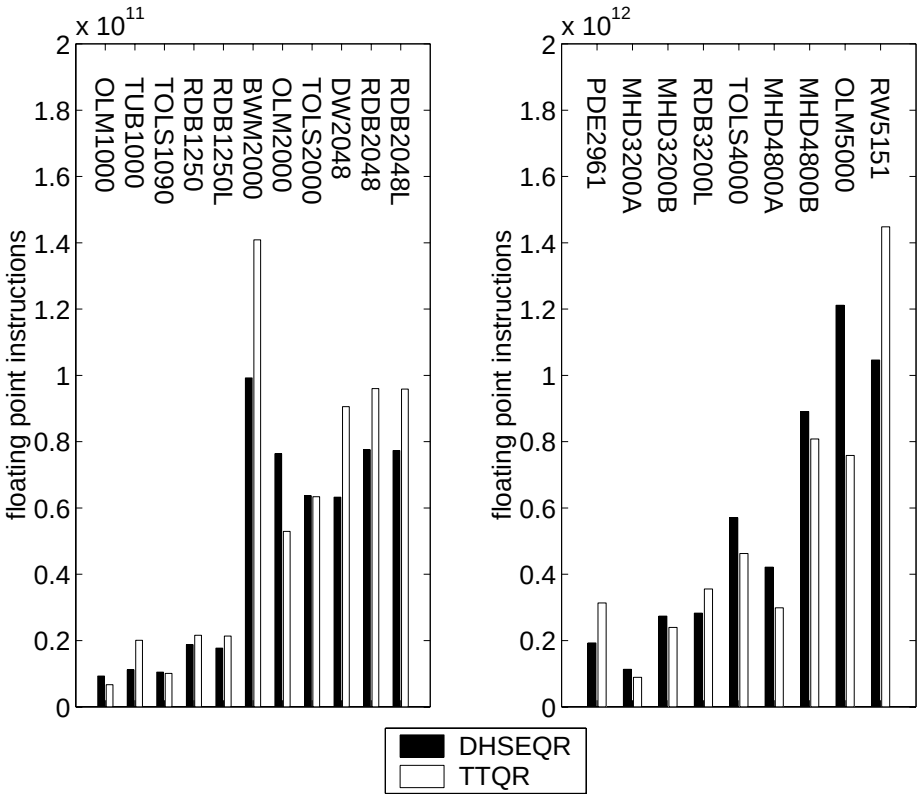


FIG. 10. *Hardware count of floating point instructions executed by* `DHSEQR`*, the large-bulge multishift QR algorithm, and* `TTQR`*, the small-bulge multishift QR algorithm, when applied to* 20 *non-Hermitian eigenvalue problems from* [3]. *The acronyms indicate the corresponding matrix from* [3]. `DHSEQR` *uses six simultaneous shifts which Figure* 1 *shows is approximately optimal.* `TTQR` *uses* 60 *simultaneous shifts on* 1,000-*by*-1,000 *to* 1,999-*by*-1,999 *matrices,* 116 *simultaneous shifts on* 2,000-*by*-2,000 *to* 2,499-*by*-2,499 *matrices,* 150 *simultaneous shifts on* 2,500-*by*-2,500 *to* 3,999-*by*-3,999 *matrices, and* 180 *simultaneous shifts on* 4,000-*by*-4,000 *or larger matrices. (Floating point instructions executed during the reduction to Hessenberg form are not included.)*

to exhibit better parallel speedup by overlapping computation in the lightly shaded region with computations in the dark bands.

4. Conclusions. The small-bulge two-tone multishift QR algorithm avoids shift blurring so completely that it admits a nearly unlimited number of simultaneous shifts—even hundreds—without adverse effects on the convergence rate. With enough simultaneous shifts, level 3 BLAS operations dominate, allowing both high serial floating point execution rates and at least modest parallel speedup.

The possibility of using hundreds of simultaneous shifts makes the choice of how many to use more complicated than for the conventional large-bulge multishift QR algorithm. We observe empirically that the amount of arithmetic work is insensitive to this choice as long as the number of shifts is small compared to the order of the matrix. Hence, the number of shifts may be chosen to fit the cache size and other characteristics of a particular computational environment without compromising the amount of arithmetic work.

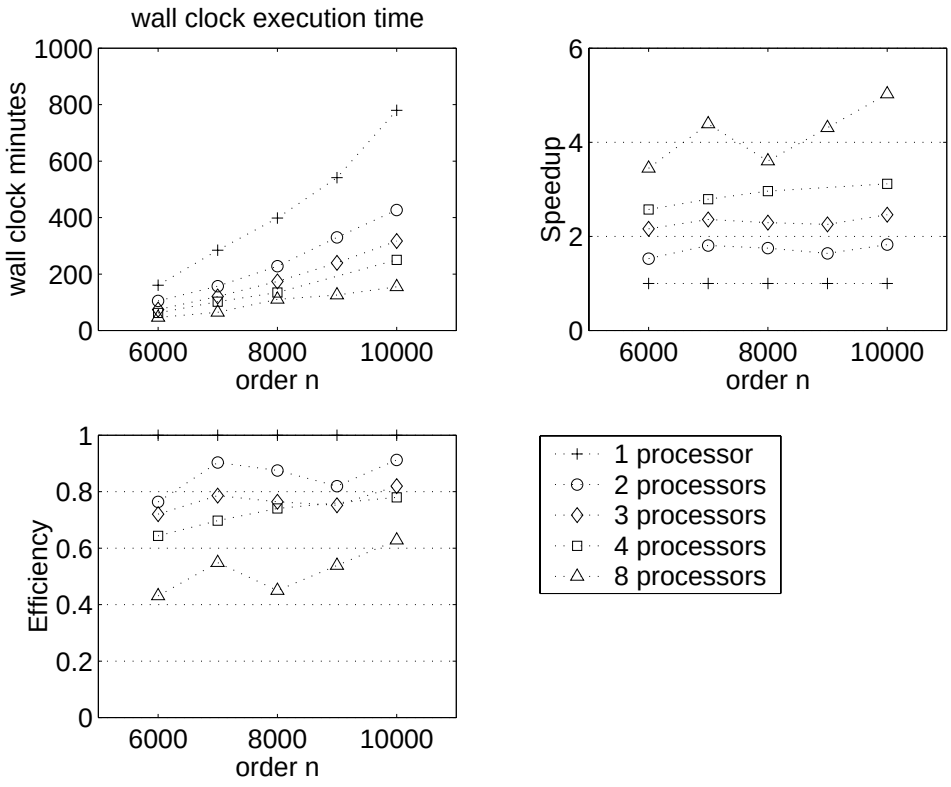


FIG. 11. *Parallel execution time, speedup, and efficiency of* `TTQR`*, the small-bulge multishift QR algorithm, on pseudorandom Hessenberg matrices. The pseudorandom test matrices, computational environment, and program parameters are described in section* 3. `TTQR` *used* 150 *simultaneous shifts. Parallel speedup is the ratio* T_1/T_p*, where* T_1 *is the* 1 *processor wall clock execution time and* T_p *is the p-processor wall clock execution time. Parallel efficiency is* $T_1/(pT_p)$.

Acknowledgment. The authors would like to thank David Watkins for helpful discussions and for bringing [55] to their attention.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, 1992.

[2] L. Auslander and A. Tsao, *On parallelizable eigensolvers*, Adv. in Appl. Math., 13 (1992), pp. 253–261.

[3] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A test matrix collection for non-Hermitian eigenvalue problems*, Department of Mathematics, University of Kentucky, Lexington, KY. Also available online from http://math.nist.gov/MatrixMarket.

[4] Z. Bai and J. Demmel, *On a block implementation of Hessenberg QR iteration*, Intl. J. of High Speed Comput., 1 (1989), pp. 97–112. Also available online as LAPACK Working Note 8 from http://www.netlib.org/lapack/lawns/lawn08.ps and http://www.netlib.org/lapack/lawnspdf/lawn08.pdf.

[5] Z. Bai and J. Demmel, *Design of a parallel nonsymmetric eigenroutine toolbox, part* I, in Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, Vol. 1, SIAM, Philadelphia, 1993, pp. 391–398.

[6] Z. Bai and J. Demmel, *Design of a Parallel Nonsymmetric Eigenroutine Toolbox, Part* II,

Tech. Report 95-11, Department of Mathematics, University of California, Berkeley, 1995.
[7] Z. Bai, J. Demmel, and M. Gu, *An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems*, Numer. Math., 76 (1997), pp. 279–308.
[8] S. Batterson, *Convergence of the shifted QR algorithm on* 3×3 *normal matrices*, Numer. Math., 58 (1990), pp. 341–352.
[9] S. Batterson, *Convergence of the Francis shifted QR algorithm on normal matrices*, Linear Algebra Appl., 207 (1994), pp. 181–195.
[10] S. Batterson and D. Day, *Linear convergence in the shifted QR algorithm*, Math. Comp., 59 (1992), pp. 141–151.
[11] A. Bojanczyk, G. H. Golub, and P. Van Dooren, *The periodic Schur decomposition; algorithms and applications*, Proc. SPIE 1770, Bellingham, WA, 1992, pp. 31–42.
[12] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part* II: *Aggressive early deflation*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 948–973.
[13] K. Braman, R. Byers, and R. Mathias, *The Multi-Shift QR-Algorithm: Aggressive Deflation, Maintaining Well Focused Shifts, and Level* 3 *Performance*, Tech. Report 99-05-01, Department of Mathematics, University of Kansas, Lawrence, KS, 1999. Also available online from http://www.math.ukans.edu/~reports/1999.html.
[14] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *A quaternion QR algorithm*, Numer. Math., 55 (1989), pp. 83–95.
[15] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *A chart of numerical methods for structured eigenvalue problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 419–453.
[16] R. R. Burnside and P. B. Guest, *A simple proof of the transposed QR algorithm*, SIAM Rev., 38 (1996), pp. 306–308.
[17] R. Byers, *A Hamiltonian QR algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[18] K. Dackland and B. Kågström, *Blocked algorithms and software for reduction of a regular matrix pair to generalized Schur form*, ACM Trans. Math. Software, 25 (1999), pp. 425–454.
[19] D. Day, *How the Shifted QR Algorithm Fails to Converge and How to Fix It*, Tech. Report 96-0913, Sandia National Laboratories, Albuquerque, NM, 1996.
[20] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[21] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson, *An extended set of Fortran basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1–17.
[22] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.
[23] J. J. Dongarra and M. Sidani, *A parallel algorithm for the nonsymmetric eigenvalue problem*, SIAM J. Sci. Comput., 14 (1993), pp. 542–569.
[24] A. Dubrulle, *The Multi-Shift QR Algorithm: Is It Worth the Trouble?*, Palo Alto Scientific Center Report G320-3558x, IBM Corp., Palo Alto, CA, 1991.
[25] J. G. F. Francis, *The QR transformation: A unitary analogue to the LR transformation.* I, Comput. J., 4 (1961/1962), pp. 265–271.
[26] J. G. F. Francis, *The QR transformation.* II, Comput. J., 4 (1961/1962), pp. 332–345.
[27] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[28] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.
[29] D. E. Heller and I. C. F. Ipsen, *Systolic networks for orthogonal decompositions*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 261–269.
[30] J. J. Hench and A. J. Laub, *Numerical solution of the discrete-time periodic Riccati equation*, IEEE Trans. Automat. Control, 39 (1994), pp. 1197–1210.
[31] G. Henry, D. S. Watkins, and J. Dongarra, *A Parallel Implementation of the Non-Symmetric QR Algorithm for Distributed Memory Architectures*, Tech. Report CS-97-352, Department of Computer Science, University of Tennessee, 1997. Also available online as LAPACK Working Note 121 from http://www.netlib.org/lapack/lawns/lawn121.ps and http://www.netlib.org/lapack/lawnspdf/lawn121.pdf.
[32] W. Hoffmann and B. N. Parlett, *A new proof of global convergence for the tridiagonal QL algorithm*, SIAM J. Numer. Anal., 15 (1978), pp. 929–937.
[33] E. R. Jessup, *A case against a divide and conquer approach to the nonsymmetric eigenvalue problem*, Appl. Numer. Math., 12 (1993), pp. 403–420.
[34] L. Kaufman, *A parallel QR algorithm for the symmetric tridiagonal eigenvalue problem*, J. Parallel and Distrib. Comput., 3 (1994), pp. 429–434.
[35] V. N. Kublanovskaya, *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.

[36] B. Lang, *Using level 3 BLAS in rotation-based algorithms*, SIAM J. Sci. Comput., 19 (1998), pp. 626–634.
[37] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, *Algorithm 539: Basic linear algebra subprograms for Fortran usage*, ACM Trans. Math. Software, 5 (1979), pp. 324–325.
[38] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, *Basic linear algebra subprograms for Fortran usage*, ACM Trans. Math. Software, 5 (1979), pp. 308–323.
[39] R. Mathias and G. W. Stewart, *A block QR algorithm and the singular value decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.
[40] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.
[41] B. N. Parlett, *Global convergence of the basic QR algorithm on Hessenberg matrices*, Math. Comp., 22 (1968), pp. 803–817.
[42] B. N. Parlett and W. G. Poole, Jr., *A geometric theory for the QR, LU, and power iterations*, SIAM J. Numer. Anal., 10 (1973), pp. 389–412.
[43] R. V. Patel, A. J. Laub, and P. Van Dooren, eds., *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press, New York, 1994.
[44] H. Rutishauser, *Solution of eigenvalue problems with the LR transformation*, in Nat. Bur. Standards Appl. Math. Ser. 49, 1958, pp. 47–81.
[45] T. Schreiber, P. Otto, and F. Hofmann, *A new efficient parallelization strategy for the QR algorithm*, Parallel Comput., 20 (1994), pp. 63–75.
[46] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Comput. Sci. 6, Springer-Verlag, New York, 1976.
[47] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[48] R. A. van de Geijn, *Storage schemes for parallel eigenvalue algorithms*, in Numerical Linear Algebra Digital Signal Processing and Parallel Algorithms, G. Golub and P. Van Dooren, eds., Springer-Verlag, Berlin, 1988, pp. 639–648.
[49] R. A. van de Geijn, *Deferred shifting schemes for parallel QR methods*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 180–194.
[50] R. A. van de Geijn and D. G. Hudson, *An efficient parallel implementation of the nonsymmetric QR algorithm*, in Proceedings of the Fourth Conference on Hypercube Concurrent Computers and Applications, Monterey, CA, 1989, pp. 697–700.
[51] C. F. Van Loan, *A general matrix eigenvalue algorithm*, SIAM J. Numer. Anal., 12 (1975), pp. 819–834.
[52] D. S. Watkins, *Understanding the QR algorithm*, SIAM Rev., 24 (1982), pp. 427–440.
[53] D. S. Watkins, *Fundamentals of Matrix Computations*, John Wiley, New York, 1991.
[54] D. S. Watkins, *Bidirectional chasing algorithms for the eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 166–179.
[55] D. S. Watkins, *Shifting strategies for the parallel QR algorithm*, SIAM J. Sci. Comput., 15 (1994), pp. 953–958.
[56] D. S. Watkins, *Forward stability and transmission of shifts in the QR algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 469–487.
[57] D. S. Watkins, *QR-like algorithms—an overview of covergence theory and practice*, in The Mathematics of Numerical Analysis, Lectures in Appl. Math. 32, J. Renegar, M. Shub, and S. Smale, eds., AMS, Providence, RI, 1996, pp. 879–893.
[58] D. S. Watkins, *The transmission of shifts and shift blurring in the QR algorithm*, Linear Algebra Appl., 241/243 (1996), pp. 877–896.
[59] D. S. Watkins and L. Elsner, *Chasing algorithms for the eigenvalue problem*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 374–384.
[60] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.
[61] R. C. Whaley and J. J. Dongarra, *Automatically tuned linear algebra software*, Tech. report, Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN, 1999. Also available online from www.netlib.org/utk/projects/atlas.
[62] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
[63] J. H. Wilkinson, *Convergence of the LR, QR, and related algorithms*, Comput. J., 8 (1965), pp. 77–84.
[64] J. H. Wilkinson, *Global convergence of tridiagonal QR algorithm with origin shifts*, Linear Algebra Appl., 1 (1968), pp. 409–420.
[65] J. H. Wilkinson and C. Reinsch, eds., *Linear Algebra*, Handbook for Automatic Computation, Vol. 2, Springer-Verlag, New York, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 948–973
© 2002 Society for Industrial and Applied Mathematics

THE MULTISHIFT *QR* ALGORITHM. PART II: AGGRESSIVE EARLY DEFLATION*

KAREN BRAMAN†, RALPH BYERS†, AND ROY MATHIAS‡

Abstract. Aggressive early deflation is a QR algorithm deflation strategy that takes advantage of matrix perturbations outside of the subdiagonal entries of the Hessenberg QR iterate. It identifies and deflates converged eigenvalues long before the classic small-subdiagonal strategy would. The new deflation strategy enhances the performance of conventional large-bulge multishift QR algorithms, but it is particularly effective in combination with the small-bulge multishift QR algorithm. The small-bulge multishift QR sweep with aggressive early deflation maintains a high rate of execution of floating point operations while significantly reducing the number of operations required.

Key words. QR algorithm, deflation, implicit shifts, eigenvalues, eigenvectors

AMS subject classifications. 65F15, 15A18

PII. S0895479801384585

1. Introduction.

An underappreciated consideration in the QR algorithm is the problem of detecting deflation. In conventional Hessenberg QR algorithms convergence is recognized when one of the Hessenberg subdiagonal entries becomes small enough to be safely set to zero. The eigenvalue problem then decouples into two smaller problems. This is the earliest deflation strategy used with the QR algorithm [24, 25, 45], and it has changed little since the early 1960s.

The small-subdiagonal convergence criterion sometimes does not recognize and deflate converged eigenvalues. Consider, for example, the 6-by-6 Hessenberg matrix

$$S_6 = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 0.001 & 1 & 0 & 0 & 0 & 0 \\ & 0.001 & 2 & 0 & 0 & 0 \\ & & 0.001 & 3 & 0 & 0 \\ & & & 0.001 & 4 & 0 \\ & & & & 0.001 & 5 \end{bmatrix}. \tag{1.1}$$

By ordinary standards, the subdiagonal entries of S_6 are not particularly small and certainly not negligible. If shifts are taken to be the eigenvalues of a trailing principal submatrix, then a multishift QR algorithm would use 5, 4, 3, etc., as the shifts for the next QR sweep. Table 1 lists the distances between S_6 and a matrix with eigenvalues equal to these likely choices of shifts. (The distances were estimated by the method described in this paper.) In particular, there is a perturbation $E \in \mathbf{R}^{6\times 6}$,

*Received by the editors February 5, 2001; accepted for publication (in revised form) by D. Boley May 15, 2001; published electronically March 27, 2002. This work was partially supported by the National Computational Science Alliance under award DMS990005N and utilized the NCSA SGI/Cray Origin2000.

http://www.siam.org/journals/simax/23-4/38458.html

†Department of Mathematics, University of Kansas, 405 Snow Hall, Lawrence, KS 66045 (braman@math.ukans.edu, byers@math.ukans.edu). The research of these authors was partially supported by National Science Foundation awards CCR-9732671, DMS-9628626, CCR-9404425, and MRI-9977352 and by the NSF EPSCoR/K*STAR program through the Center for Advanced Scientific Computing.

‡Department of Mathematics, College of William & Mary, Williamsburg, VA 23187 (mathias@math.wm.edu). The research of this author was partially supported by NSF grants DMS-9504795 and DMS-9704534.

TABLE 1
Estimates of the distances between S_6 in (1.1) and a matrix with eigenvalues equal to eigenvalues of a trailing principal submatrix.

A matrix with these eigenvalues...	...is within this spectral norm distance of S_6.
1 2, 3, 4, and 5	1×10^{-3}
2, 3, 4, and 5	1×10^{-6}
3, 4, and 5	5×10^{-10}
4 and 5	2×10^{-13}
5	4×10^{-17}

$\|E\|_2 \approx 4 \times 10^{-17}$ for which 5 is an eigenvalue of $H + E$. In typical finite precision computation, the unit roundoff is roughly 2×10^{-16}. In that context, arguably, the shift 5 is a converged eigenvalue that should be detected and deflated before the next QR sweep. Possibly, both 4 and 5 can be accepted as eigenvalues and be deflated. If the unit roundoff were, say, 1×10^{-7}, then 5, 4, 3, and possibly even 2 might be acceptable eigenvalues.

The undetected but essentially converged shift 5 would ordinarily be used as a shift in the QR algorithm and after a QR sweep or two the corresponding subdiagonal will become small enough to set to zero. The extra sweep or two represents unnecessary arithmetic work, but it is benign. It does not introduce numerical instability. However, the converged eigenvalues do occupy shifts that could be used to start work on other, unconverged eigenvalues. In the multishift QR setting, the small-subdiagonal convergence criterion usually remains unsatisfied until all or nearly all of the simultaneous shifts have converged to eigenvalues to full precision. (The QR algorithm tends to deflate submatrices of roughly the same order as the number of shifts.) However, it is typical for some of the shifts to converge before others. In our experience with the two-tone small-bulge QR algorithm [10] and small-subdiagonal deflation, it is commonly the case that, up to rounding error, *half or more of the shifts are converged eigenvalues!* Unable to recognize and deflate the converged eigenvalues, the algorithm must reuse them in the next sweep. It cannot work on new, unconverged eigenvalues until a small subdiagonal appears when all or nearly all the current shifts have converged. Much of the potential performance of the multishift QR algorithm may be lost.

Readers of this paper need to be familiar with the double implicit shift QR algorithm [24, 25, 32]. (See, for example, any of the textbooks [15, 29, 38, 41, 45].) Familiarity with the large-bulge multishift QR algorithm [3] as implemented in LAPACK [1] and/or the small-bulge multishift QR algorithm [10] is helpful.

1.1. Notation. Throughout this paper we use the following notation and definitions.

1. We will use the "colon notation" to denote submatrices: $H_{i:j,k:l}$ is the submatrix of matrix H in rows i–j and columns k–l inclusively. The notation $H_{:,k:l}$ indicates the submatrix in columns k–l inclusively (and all rows). The notation $H_{i:j,:}$ indicates the submatrix in rows i–j inclusively (and all columns).

2. The notation $\sigma_k = \sigma_k(M)$ denotes the kth largest magnitude singular value of the matrix M. The singular value of smallest magnitude will also be written as $\sigma_{\min} = \sigma_{\min}(M)$.

3. The spectral norm is denoted $\|M\|_2 = \sigma_1(M)$. The Frobenius norm is $\|M\|_F = \sqrt{\text{trace}(M^T M)}$.

4. The jth column of the identity matrix is denoted e_j. The matrix formed from the first k columns of I, $[e_1, e_2, e_3, \ldots, e_k]$, is E_k. The subspace $\mathcal{E}_k \subset \mathbf{C}^n$ is the subspace spanned by the first k columns of I, i.e., $\mathcal{E}_k = \text{span}(e_1, e_2, e_3, \ldots, e_k)$. The orthogonal complement of $\mathcal{E}_k$ is $\text{span}(e_{k+1}, e_{k+2}, e_{k+3}, \ldots, e_n)$ and is denoted $\mathcal{E}_k^{\perp}$.

5. A 1-*unitary* matrix is a unitary matrix $Q \in \mathbf{C}^{n\times n}$ for which $Qe_1 = e_1$ and $e_1^T Q = e_1^T$, i.e., the first row and column of Q is the first row and column of I. A real 1-unitary matrix is called 1-orthogonal. Householder reduction to Hessenberg form [29, Algorithm 7.4.2] produces a 1-unitary matrix in the form of a product of Householder reflections.

6. A *quasi-triangular* matrix is a real, block triangular matrix with 1-by-1 and 2-by-2 blocks along the diagonal.

7. The orthogonal projection of a vector $x \in \mathbf{R}^n$ onto a subspace $\mathcal{Q} \subset \mathbf{C}^n$ is denoted by $\text{Proj}_{\mathcal{Q}}(x)$. If the subspace $\mathcal{Q}$ is spanned by the columns of a matrix Q, we will abbreviate $\text{Proj}_{\text{Range}(Q)}(x)$ by $\text{Proj}_Q(x)$.

8. A matrix $H \in \mathbf{R}^{n\times n}$ is in Hessenberg form if $h_{ij} = 0$ whenever $i > j+1$. The matrix H is said to be unreduced if, in addition, the subdiagonal entries are nonzero, i.e., $h_{ij} \neq 0$ whenever $i = j+1$.

9. Following [29, p. 19], we define a "flop" as a single floating point operation, i.e., either a floating point addition or a floating point multiplication together with its associated subscripting. The Fortran statement

$$\texttt{C(I, J)} = \texttt{C(I, J)} + \texttt{A(I, K)} * \texttt{B(K, J)}$$

involves two flops.

In some of the examples in section 3, we report an automatic hardware count of the number of floating point instructions executed. Note that a trinary multiply-add instruction counts as just one executed instruction in the hardware count even though it executes two flops. Thus, depending on the compiler and optimization level, the above Fortran statement could be executed using either one floating point instruction or two floating point instructions (perhaps along with some integer subscripting calculations) even though it involves two flops.

10. The (i,j)th minor of a matrix $M \in \mathbf{R}^{n\times n}$ is represented by $M(i|j)$. (The (i,j)th minor is the determinant of the matrix obtained by deleting the ith row and jth column.)

11. The classical adjoint matrix or adjugate of a matrix $M \in \mathbf{R}^{n\times n}$ is written $\text{adj}(M)$. Its (i,j)th entry is the (j,i)th cofactor, $(-1)^{i+j}M(i|j)$.

2. Aggressive early deflation. In broad outline, the aggressive early deflation procedure derived and analyzed below works as follows. Partition an unreduced Hessenberg matrix H as

$$H = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \end{array}, \tag{2.1}$$

where k is an integer $1 \le k < n$. Let $H_{33} = VTV^H$ be a Schur decomposition of H_{33}. Consider the similarity transformation to a Hessenberg-plus-spike form

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix}^H \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13}V \\ H_{21} & H_{22} & H_{23}V \\ 0 & s & T \end{bmatrix}. \tag{2.2}$$

We explain below in section 2.8 that it is often the case that the last several components of s are tiny even when no subdiagonal entry of H is particularly small. The right-hand column in Table 1 is the spike s obtained from (1.1) with $k = 5$.

If the trailing m components of s are set to zero, then (2.2) takes the form

$$\begin{array}{c} \\ n-k-1 \\ 1 \\ k-m \\ m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-1 & 1 & k-m & m \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & \tilde{H}_{13} & \tilde{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \tilde{s} & T_{11} & T_{12} \\ 0 & 0 & 0 & T_{22} \end{bmatrix} \end{array}.$$

The eigenvalues of T_{22} are deflated! Ordinarily, many more QR sweeps would be needed to reduce the subdiagonals of H to the point that these eigenvalues deflate.

If $k \ll n$, then the work needed to compute the Schur decomposition, form s, and return to Hessenberg form is small compared to the work of the QR sweeps that are saved. Approximately $4nk^2 + 31k^3$ flops are needed for this. See [10, Appendix B] for a more detailed description of the algorithm and a derivation of the flop count. Of these, if k is big enough, $4nk^2$ will be level 3 BLAS operations.

Incidentally, the eigenvalues of T_{11} make good shifts for the next multishift QR sweep.

2.1. Reducing perturbations. Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix, let $P \in \mathbf{C}^{n\times n}$ be a perturbation matrix, and let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $\hat{H} \equiv Q^H(H+P)Q$ is Hessenberg. We call P a *reducing perturbation* if $\hat{H}$ is a reduced Hessenberg matrix. If P is a reducing perturbation of negligible magnitude, then the problem of finding the eigenvalues of H splits into the smaller problems of finding the eigenvalues of the two or more diagonal blocks of the block triangular matrix $\hat{H} = Q^H(H+P)Q$. At least in principle, aggressive early deflation consists of finding a reducing perturbation P of negligible magnitude, if possible, and using it to deflate the eigenvalue problem into two or more smaller problems.

Reducing perturbations are easily characterized in terms of the zero structure of a left eigenvector.

LEMMA 2.1. *A matrix $P \in \mathbf{C}^{n\times n}$ is a reducing perturbation for a matrix $H \in \mathbf{C}^{n\times n}$ if and only if $H + P$ has a left eigenvector $v \in \mathbf{C}^n$ with zero first component (i.e., $v_1 = 0$).*

Proof. Let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $Q^H(H+P)Q = \hat{H}$ is Hessenberg.

Suppose that P is a reducing perturbation, and, consequently, $\hat{H}$ is a reduced Hessenberg matrix. In particular, $\hat{H}$ is block triangular, so $\hat{H}$ has a left eigenvector $\hat{v}$ with $\hat{v}_1 = 0$. It follows that $v = Q^H\hat{v}$ is a left eigenvector of H and $v_1 = \hat{v}_1 = 0$.

Conversely, suppose that $(H + P)$ has a left eigenvector v with corresponding eigenvalue $\lambda \in \mathbf{C}$ and $v_1 = 0$. So, $\hat{v} = Q^H v$ is a left eigenvector of $\hat{H}$ and $\hat{v}_1 = 0$. Let k be the smallest index for which $\hat{v}_k \neq 0$. The Hessenberg structure of $\hat{H}$ implies that the $(k-1)$st component of $\lambda\hat{v} = \hat{v}^H\hat{H}$ is $0 = \lambda\bar{\hat{v}}_{k-1} = \bar{\hat{v}}_k\hat{h}_{k,k-1}$. Hence, $h_{k,k-1} = 0$ and $\hat{H}$ is a reduced Hessenberg matrix. ∎

Of course, in the context of aggressive deflation, it is not sufficient to find a small norm reducing perturbation. Returning a dense n-by-n matrix $H + P$ to Hessenberg form and accumulating the orthogonal similarity transformations would cost roughly $7n^3 + O(n^2)$ flops [29, Algorithm 7.4.2]—more arithmetic work than needed by many QR sweeps. A useful reducing perturbation P needs to have enough zero structure so that relatively little work is needed to return $H + P$ to Hessenberg form.

If the reducing perturbation P is restricted to be Hessenberg, then no extra work is needed, because $H + P$ is already Hessenberg. It is easy to show that a Hessenberg reducing perturbation of minimal Frobenius norm is zero except for some subdiagonal entry. Thus, restricting P to be Hessenberg leads to the small-subdiagonal deflation strategy.

A more aggressive deflation strategy must consider reducing perturbations that are not necessarily Hessenberg. Consider searching for a deflating perturbation $P \in \mathbf{C}^{n\times n}$ which may be nonzero only its last k rows and $k+1$ columns. With this choice, $H + P$ is Hessenberg in its initial $n - k - 1$ columns. If $k \ll n$, then returning to Hessenberg form would need an acceptably small amount of arithmetic.

The next lemma identifies deflating perturbations of this form that have minimal norm.

LEMMA 2.2. *Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix partitioned as in* (2.1). *Let $\mu_* \in \mathbf{C}$ be a minimizer of $f(\mu) = \sigma_k([H_{32},\, H_{33} - \mu I])$ with corresponding left singular vector $u_* \in \mathbf{C}^k$ and right singular vector $v_* \in \mathbf{C}^{k+1}$. If $P \in \mathbf{C}^{n\times n}$ is given by*

$$P = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & P_{32} & P_{33} \end{bmatrix} \end{array}, \tag{2.3}$$

where $[P_{32},\, P_{33}] = -f(\mu_*)u_*v_*^H \in \mathbf{C}^{k\times(k+1)}$, then $\|P\|_2 = f(\mu_*) = \sigma_k([H_{32},\, H_{33} - \mu_* I])$, and P is a reducing perturbation of minimal spectral and Frobenius norm with the zero structure of (2.3).

Proof. By construction,

$$[0,\, 0,\, u_*^H] \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} + P_{32} & H_{33} + P_{33} \end{bmatrix} = \mu_* [0,\, 0,\, u_*^H].$$

Hence, μ_* is an eigenvalue of $H_{33} + P_{33}$ and $H + P$ with left eigenvectors u_* and $[0,\, 0,\, u_*^H]$, respectively. Hence, by Lemma 2.1, P is a reducing perturbation.

Suppose that $u \in \mathbf{C}^n$ is a left eigenvector of $H + P$ with eigenvalue $\mu \in \mathbf{C}$ and that $u_1 = 0$. Because the first $n-k-1$ columns of $H+P$ are in unreduced Hessenberg form, u must have zeros in its first $n - k$ components. The trailing k components form a left null vector of $[H_{32},\, H_{33} - \mu I] + [P_{32},\, P_{33}]$. It is an application of the singular value decomposition that $[P_{32},\, P_{33}]$ is the perturbation of smallest spectral and Frobenius norm for which $\operatorname{rank}([H_{32},\, H_{33} - \mu I] + [P_{32},\, P_{33}]) < k$. Hence, P is a perturbation of smallest spectral and Frobenius norm for which $H + P$ admits a left eigenvector whose first $n - k$ components are zeros. ∎

If $\|P\|_2$ is tiny enough to be neglected, then μ_* is essentially an eigenvalue of H with left eigenvector $[0,\, 0,\, u_*^H]$.

The problem of minimizing $f(\mu) = \sigma_k([H_{32},\, H_{33} - \mu I])$ has been extensively studied in the form of the controllability radius problem. A pair of matrices (A, B) is said to be controllable if for all $\lambda \in \mathbf{C}$ the matrix $[B,\, A - \lambda I]$ has full row rank [30]. Controllability is a sort of nonsingularity or well-definedness property required in many contexts in control theory [33, 34]. It is also a factor in the numerical condition of computational control problems [11, 16, 14, 34]. The minimum value of $f(\mu)$ is $\nu(H_{33}, H_{32})$, the Frobenius norm distance and spectral norm distance from the pair $(H_{33},\, H_{32})$ to the nearest uncontrollable pair [21, 34, 35].

A remarkably eclectic collection of mathematical tools and techniques have been used to attack the problem of calculating or estimating the controllability radius. See, for example, [6, 8, 7, 12, 13, 16, 21, 22, 23, 26, 27, 44]. By Lemma 2.2, all of them are potentially applicable to shift selection and deflation.

A dissatisfying aspect of Lemma 2.2 is that the optimal deflating perturbation P usually deflates only a single eigenvalue. Possibly, a perturbation of slightly larger magnitude would deflate several eigenvalues. Another dissatisfying aspect of Lemma 2.2 is that even when H is a real matrix, the perturbation matrix P may be complex. The extra work and storage required by complex arithmetic makes its use unattractive, but the staircase form and eigenvalue clustering methods for approximating the controllability radius [5, 16, 18, 17, 31, 35] can be easily and naturally adapted to find a "small norm" real deflating perturbation and so avoid complex arithmetic.

A simple heuristic way to use Lemma 2.2 is to approximate the minimizing μ_* by one of the eigenvalues of H_{33}. The corresponding left eigenvector u_* serves as an approximate left singular vector, and e_1, the first column of the $(k+1)$-by-$(k+1)$ identity, serves as an approximate right singular vector. This leads to a reducing perturbation of the form

$$P = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & P_3 & 0 \end{bmatrix} \end{array}. \tag{2.4}$$

We call perturbation matrices with this sparsity pattern *k-spike perturbations*.

2.2. k-spike reducing perturbations. Spike reducing perturbations can be naturally adapted to avoid complex arithmetic when H is real and to deflate several eigenvalues at a time.

We will say that a k-spike reducing perturbation $P \in \mathbf{C}^{n\times n}$ (2.4) *deflates m eigenvalues and their corresponding left-invariant subspace* from a matrix $H \in \mathbf{C}^{n\times n}$ if there is a 1-unitary matrix Q for which $\hat{H} = Q^H(H+P)Q$ is a reduced Hessenberg matrix with $\hat{h}_{n-m+1,n-m} = 0$. (If several subdiagonal entries of $\hat{H}$ are zero, then there are several different values of m for which a perturbation P may be said to deflate m eigenvalues.) The matrix $\hat{H}$ takes the form

$$\hat{H} = Q^H(H+P)Q = \begin{array}{c} \\ n-m \\ m \end{array} \begin{array}{c} \begin{array}{cc} n-m & m \end{array} \\ \begin{bmatrix} \hat{T}_{11} & \hat{T}_{12} \\ 0 & \hat{T}_{22} \end{bmatrix} \end{array},$$

where $\hat{T}_{11}$ and $\hat{T}_{22}$ are Hessenberg. The last m columns of Q span an m-dimensional left-invariant subspace of $H+P$ corresponding to the eigenvalues of $\hat{T}_{22}$. The 1-unitary matrix Q is not unique, but if $\hat{T}_{11}$ is unreduced Hessenberg, then the implicit Q theorem [29, Theorem 7.4.3] implies that the first $n-m$ columns of Q are unique up to column scaling by numbers of modulus 1. There is enough freedom in the choice of the remaining m columns of Q to make $\hat{T}_{22}$ triangular. (If H and Q are restricted to be real, then there is only enough freedom to make $\hat{T}_{22}$ be quasi-triangular.) In this case m eigenvalues are displayed along the diagonal of $\hat{T}_{22}$. If P is of negligible magnitude, then P deflates the m eigenvalues of $\hat{T}_{22}$ and the left-invariant subspace spanned by the last m columns of Q. The smaller problem of calculating the remaining $n-m$ eigenvalues of $\hat{T}_{11}$ remains.

The following lemma and theorem characterize multieigenvalue deflating k-spike reducing perturbations in terms of the zero structure of the left-invariant subspaces they deflate. In the statement of the lemma, recall that $\mathcal{E}_k$ is the space spanned by the first k columns of I.

LEMMA 2.3. *A matrix $P \in \mathbf{C}^{n\times n}$ is a reducing perturbation for $H \in \mathbf{C}^{n\times n}$ that deflates at least m eigenvalues if and only if $H+P$ has a left-invariant subspace $\mathcal{Q}$ of dimension m or greater for which $\mathcal{Q} \subset \mathcal{E}_1^{\perp}$.*

Proof. Let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $Q^H(H+P)Q = \hat{H}$ is Hessenberg.

Suppose that P is a reducing perturbation that deflates $\hat{m} \geq m$ eigenvalues. The reduced Hessenberg matrix $\hat{H}$ is block triangular with an $\hat{m}$-by-$\hat{m}$ $(2,2)$ block. It follows that the last $\hat{m}$ columns of Q span an $\hat{m}$-dimensional left-invariant subspace, $\mathcal{V}$ of $H+P$. The 1-unitary structure of Q implies that the first entry in each of the last $\hat{m}$ columns of Q is zero. Hence, $\mathcal{V} \subset \mathcal{E}_1^{\perp}$.

Conversely, if $(H+P)$ has a left-invariant subspace $\mathcal{V} \subset \mathcal{E}_1^{\perp}$ of dimension $\hat{m} \geq m$, then $\hat{\mathcal{V}} = Q^H\mathcal{V}$ is an $\hat{m}$-dimensional left-invariant subspace of $\hat{H}$. It follows from the 1-unitary structure of Q that $\hat{\mathcal{V}} \subset \mathcal{E}_1^{\perp}$. Let k be the largest integer for which $\hat{\mathcal{V}} \subset \mathcal{E}_k^{\perp}$. The first k entries of all members of $\hat{\mathcal{V}}$ are zero, but there is a vector $\hat{v} \in \hat{\mathcal{V}}$ for which $\hat{v}_{k+1} \neq 0$. Note that $k \leq n - \hat{m} \leq n - m$, because $\hat{\mathcal{V}}$ has dimension $\hat{m} \geq m$. The matrix $\hat{H}$ is a Hessenberg matrix for which $\hat{v}^H\hat{H} \subset \hat{\mathcal{V}}$. In particular, the kth entry of $\hat{v}^H\hat{H}$ is zero, i.e., $\hat{v}_{k+1}\hat{h}_{k+1,k} = 0$. Hence, $\hat{h}_{k+1,k} = 0$ and P deflates at least m eigenvalues. ∎

The next theorem characterizes k-spike reducing perturbations that deflate several eigenvalues.

THEOREM 2.4. *Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix partitioned as in* (2.1)*, and let P be a k-spike perturbation as in* (2.4)*. The k-spike perturbation P deflates at least $m \leq k$ eigenvalues if and only if for some m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}, $Proj_{\tilde{\mathcal{V}}}(P_3) = -Proj_{\tilde{\mathcal{V}}}(H_{32})$.*

The k-spike reducing perturbation of minimal Frobenius norm corresponds to the choice $P_3 = -Proj_{\tilde{\mathcal{V}}}(H_{32})$ for some m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}.

Proof. Suppose that the columns of $\tilde{V} \in \mathbf{C}^{k\times m}$ form an orthonormal basis of an m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}. Let $\Lambda \in \mathbf{C}^{m\times m}$ satisfy $\tilde{V}^H H_{33} = \Lambda \tilde{V}^H$. Define $V \in \mathbf{C}^{n\times m}$ by

$$V^H = m \begin{array}{c} \\ \left[\right. \end{array} \begin{array}{ccc} n-k-1 & 1 & k \\ 0 & 0 & \tilde{V}^H \end{array} \left.\right]. \tag{2.5}$$

If $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$, then $\tilde{V}^H P_3 = -\tilde{V}^H H_{32}$ and, by direct calculation, $V^H(H+P) = \Lambda V^H$. Hence, $\mathcal{V} \equiv \mathrm{Range}(V)$ is an m-dimensional left-invariant subspace of $H+P$. It follows from (2.5) that $\mathcal{V} \subset \mathcal{E}_1^{\perp}$. Hence, by Lemma 2.3, P is a deflating perturbation that deflates at least m eigenvalues and the left-invariant subspace $\mathcal{V}$.

Conversely, let P be a k-spike perturbation that deflates $\hat{m}$ eigenvalues, with $m \leq \hat{m} \leq k$. Let Q be a 1-unitary matrix chosen so that $\hat{H} = Q^H(H+P)Q$ is Hessenberg. The first k columns of $H+P$ are unreduced Hessenberg, because the first k columns of H are. Without loss of generality, we may choose Q so that $\hat{h}_{n-m+1,n-m} = 0$. The implicit Q theorem [29, Theorem 7.4.3] implies that Q takes

the form

$$Q = \begin{array}{c} \\ n-k-1 \\ 1 \\ k-m \\ m \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & \hat{V} \\ 0 & 0 & \tilde{V} \end{bmatrix} \end{array},$$

where D_1 and D_2 are diagonal and unitary. With this choice, the last m columns of Q span an m-dimensional left-invariant subspace $H+P$. In the notation of (2.1) and (2.4), there exists a matrix $\Lambda \in \mathbf{C}^{m\times m}$ for which

$$\begin{bmatrix} 0 & 0 & \tilde{V} \end{bmatrix}^H \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32}+P_3 & H_{33} \end{bmatrix} = \Lambda \begin{bmatrix} 0 & 0 & \tilde{V} \end{bmatrix}^H.$$

In particular, $\tilde{V}^H(H_{32}+P_3)=0$ and $\tilde{V}^H H_{33} = \Lambda \tilde{V}^H$. So, $\mathcal{V} = \mathrm{Range}(\tilde{V}) \subset \mathbf{C}^{k\times k}$ is an m-dimensional left-invariant subspace of H_{33} and $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$.

An elementary application of linear least squares shows that for any subspace $\mathcal{V} \subset \mathbf{C}^k$, $P_3 = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$ is the minimum Frobenius norm solution to $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$. ∎

2.3. Implementing aggressive early deflation. Theorem 2.4 shows how to calculate (hopefully) small norm k-spike perturbations P that deflate several eigenvalues:

1. Select a "deflation window" consisting of the trailing k-by-k principal submatrix H_{33}.

2. Select an m-dimensional left-invariant subspace $\mathcal{V} \subset \mathbf{C}^k$ of H_{33} such that $\|\mathrm{Proj}_{\mathcal{V}}(H_{32})\|_2$ is "small."

3. Compute $P_3 = -\mathrm{Proj}_{\mathcal{V}}(H_{32})$. (The spike perturbation P is given by (2.4).)

If $\|P\|_F = \|P_3\|_2$ is negligible, then, in principle, it may be used to deflate m eigenvalues by returning $H+P$ to Hessenberg form with a 1-unitary similarity transformation $\hat{H} = Q^H(H+P)Q$.

There are several practical considerations. First, the calculation must be organized so that the computed version of $\hat{H} = Q^H(H+P)Q$ is indeed *reduced* Hessenberg despite rounding errors. Second, some flexible but effective heuristic must be used to select the invariant subspace. (Including the trivial space $\{0\}$, there are typically 2^k invariant subspaces of H_{33}. Computing bases of all of them is impractical.) Finally, if H is real, it would be best to restrict all calculations to real arithmetic.

The following procedure takes these practical considerations into account. Let $H \in \mathbf{R}^{n\times n}$ be a real, unreduced Hessenberg matrix. Make an a priori choice of the deflation window size k. (The success of aggressive early deflation is relatively insensitive to the deflation window size. However, it is best to choose k to be larger than the number of simultaneous shifts in the multishift QR algorithm.) Partition H as in (2.1). Using the QR algorithm (what else?), compute a real Schur decomposition of H_{33}, $V^T H_{33} V = T$, where $V \in \mathbf{R}^{k\times k}$ is orthogonal and $T \in \mathbf{R}^{k\times k}$ is quasi-triangular. (This might be implemented as a recursive subroutine.) The eigenvalues of H_{33} are arranged along the diagonal of T as the 1-by-1 and 2-by-2 diagonal blocks. For each integer $m < k$, for which $t_{k-m+1,k-m} = 0$, $\mathcal{V}_m = \mathrm{Range}(V_{k-m+1:k,:})$ is an m-dimensional left-invariant subspace of H_{33}. If $s = V^T H_{32} \in \mathbf{R}^k$, then $\|s_{k-m+1:k}\|_2 = \|-\mathrm{Proj}_{\mathcal{V}_m}(H_{32})\|_F$. By Theorem 2.4, $\|s_{k-m+1:k}\|_2$ is the magnitude of the minimal

norm k-spike perturbation that deflates the trailing m eigenvalues of T. If $\|s_{k-m+1:k}\|_2$ is so tiny that it may be safely set to zero, then we may use the corresponding k-spike perturbation to deflate m eigenvalues. These m eigenvalues are *deflatable*.

For each possible ordering of the eigenvalues of T (keeping complex conjugate pairs adjacent), there is a real Schur decomposition which achieves that ordering along the diagonal of T. Software for efficiently updating one real Schur decomposition to another by unitary similarity transformation is widely available. (See, for example, [4, 39] or [1, `DTREXC`].) By reordering the eigenvalues along the diagonal of T, a deflation procedure may augment the set of deflatable eigenvalues. Suppose, for example, that the trailing m eigenvalues along the diagonal of T are deflatable, but $\lambda = t_{k-m,k-m}$ is a simple eigenvalue that cannot be classified as deflatable because $|s_{k-m}|$ is not "small enough." Use a unitary similarity transformation to rotate λ up to t_{11} while leaving the m eigenvalues already classified as deflatable in their original positions. The new value $t_{k-m,k-m}$ (or, in case of a complex conjugate pair, the 2-by-2 block $T_{k-m-1:k-m,k-m-1:k-m}$) now may be deflatable. In this way, each eigenvalue may be examined and classified as deflatable or nondeflatable. The deflatable eigenvalues collect in the trailing diagonal entries of T.

Having calculated a suitable, reordered, real Schur decomposition $H_{33} = VTV^T$ and $s = V^T H_{32}$, and having classified the trailing m eigenvalues as deflatable as described above, compute the similarity transformation

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix}^T \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13}V \\ H_{21} & H_{22} & H_{23}V \\ 0 & s & T \end{bmatrix} \tag{2.6}$$

by multiplying H_{13} and H_{23} from the right by V. The trailing m entries of s are tiny—small enough to be set to zero. Set the trailing m entries of s to zero to obtain a perturbed vector $[\tilde{s}^T, 0]^T \equiv [s_{1:k-m}^T, 0]$. The matrix

$$\tilde{H} = \begin{array}{c} \\ n-k-1 \\ 1 \\ k-m \\ m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-1 & 1 & k-m & m \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & \tilde{H}_{13} & \tilde{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \tilde{s} & T_{11} & T_{12} \\ 0 & 0 & 0 & T_{22} \end{bmatrix} \end{array}$$

is block triangular with m-by-m quasi-triangular, trailing principal submatrix T_{22}. The m eigenvalues of T_{22} are eigenvalues of $\tilde{H}$.

It remains to return $\tilde{H}$ to reduced Hessenberg form. The first $n-k-1$ columns of $\tilde{H}$ are still in unreduced Hessenberg form, so the first $n-k-1$ steps of Householder's method [29, Algorithm 7.4.2] do not modify $\tilde{H}$. Hence, when returning $\tilde{H}$ to Hessenberg form, the first $n-k-1$ steps of Householder's method [29, Algorithm 7.4.2] are unnecessary and may be skipped. In addition Householder's method [29, Algorithm 7.4.2] does not modify $T_{22} = H_{n-m:n,n-m:n}$, so the last $m-1$ steps can be skipped as well. If $k^2 \ll n$, then the work required to return $H+P$ back to Hessenberg form is small compared to the cost of a QR sweep.

2.4. When are spike components negligible? When an entry in the spike is "small enough," it may be safely set to zero without significant adverse effect on accuracy. Considerable care is needed to make this decision without unnecessarily sacrificing accuracy. A conservative stopping criterion was all that was needed to greatly increase the accuracy of the Jacobi method [19]. This subsection presents a

short discussion of the question in the context of aggressive early deflation but does not give a definitive answer.

Setting the last m components of the spike to zero is equivalent to adding a k-spike perturbation of norm $\|s_{k-m+1:k}\|_2$. Rounding errors in the reduction to Hessenberg form and during the QR sweep are equivalent to perturbing H by an additive perturbation matrix of norm $p(n)\mu\|H\|_2$, where μ is the unit roundoff and $p(n)$ is a low degree polynomial that depends on the the norm, the details of the algorithm, and the details of the finite precision arithmetic. (See [45, Chap. 3].) It is natural to require that $\|s_{k-m+1:k}\|_2$ be at least roughly as small as the other unavoidable rounding errors before it can be set to zero. This suggests what might be called *norm-stable deflation*.

Norm-stable deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \leq \varepsilon\|H\|_F$.

Requiring spike components to satisfy this criterion before being set to zero is a minimum requirement. If H is a graded matrix, even norm-stable deflation may deflate approximate eigenvalues before they have converged to a limiting accuracy. The more conservative deflation strategies described below are empirically reliable and, in some cases, yield more accurate computed eigenvalues.

In terms of the partitioning (2.1), aggressive early deflation depends only on H_{32} and H_{33}. It is natural to use a deflation criterion that depends only upon them. This suggests what we call window-norm-stable deflation.

Window-norm-stable deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \leq \varepsilon\|[H_{32},\, H_{33}]\|_F$.

Window-norm-stable deflation compromises between norm-stable deflation and the EISPACK and LAPACK compare-to-nearby-diagonal-entries strategy. Note that the deflation window size is likely to be substantial. (In the numerical examples reported in section 3, window sizes ranged from $k = 48$ to $k = 450$.) Hence, window-norm-stable deflation resembles norm-stable deflation and suffers many of the same drawbacks.

EISPACK [28, 37] and LAPACK [1] use small-subdiagonal deflation. Subdiagonal entries are considered small enough to deflate only if they are tiny compared to *nearby* matrix entries. EISPACK [28, 37] subroutines `HQR/HQR2` and LAPACK [1] subroutines `DHSEQR/DLAHQR` ordinarily set a subdiagonal entry $h_{i+1,i}$ to zero only if $|h_{i+1,i}| \leq \mu(|h_{ii}| + |h_{i+1,i+1}|)$, where μ is the unit roundoff. (If both $h_{ii} = 0$ and $h_{i+1,i+1} = 0$, then EISPACK falls back on norm-stable deflation. LAPACK falls back on a modified norm-stable deflation criterion. LAPACK also sets $h_{i+1,i}$ to zero when it is near the underflow threshold.) This more conservative deflation strategy respects the local scale of graded matrices and sometimes significantly improves the accuracy of the computed eigenvalues.

A deflation strategy in the spirit of the EISPACK/LAPACK compare-to-nearby-diagonal-entries strategy might be to compare the spike component s_k to the diagonal block of T in row k. For example, if $t_{k,k}$ is a 1-by-1 diagonal block in T, i.e., a real eigenvalue of T, then s_k might be set to zero if it is tiny compared to t_{kk}. Once s_k has been set to zero, the same deflation strategy may be applied to s_{k-1} and then to s_{k-2} and so on until a nondeflatable eigenvalue is encountered. If $T_{k-1:k,k-1:k}$ is a 2-by-2 diagonal block with complex conjugate eigenvalues λ and $\bar{\lambda}$, then a similar deflation strategy may be applied to the two spike components s_{k-1} and s_k by comparing them

to $\|T_{k-1:k,k-1:k}\|_2$ or $|\lambda| = \sqrt{\det(T_{k-1:k,k-1:k})}$.

Nearby-diagonal deflation:

If t_{jj} is a 1-by-1 diagonal block of T, then s_j may be set to zero if, for some rounding error small number ε, $|s_j| \le \varepsilon\,|t_{jj}|$.

If $T_{j:j+1,j:j+1}$ is a 2-by-2 diagonal block of T, then s_j and s_{j+1} may be set to zero if, for some rounding error small number ε, $\max(s_j, s_{j+1}) \le \varepsilon\sqrt{\det(T_{j:j+1,j:j+1})}$.

Recall that the spike is $s = V^T H_{32} = h_{n-k+1,n-k} V_{1,:}$, where $V^T H_{33} V = T$ is a real Schur decomposition. Setting $s_{k-m+1:k}$ to zero is equivalent to perturbing V to a nearby matrix $\tilde{V}$ by setting $V_{1,k-m+1:k}$ to zero. This corresponds to replacing the Schur decomposition $H_{33} = VTV^T$ by the multiplicatively perturbed factorization $(I+E)^{-1}H_{33}(I+E) = ((I+E)^{-1}V)T((V^T(I+E))) = \tilde{V}^{-T}T\tilde{V}^T$, where E satisfies $V^T(I+E) = \tilde{V}^T$. The matrix E may be chosen to be a 1-spike perturbation with $\|E\|_2 = \|E\|_F = \|V_{1,k-m+1:k}\|_2 = \|s_{k-m+1:k}\|_2/h_{n-k+1,n-k}$.

Finite precision arithmetic nearly always prevents the computed version of V from being exactly orthogonal, but it can be shown that the computed V is of the form $\hat{V}(I+\hat{E})$, where $\hat{V}$ is exactly orthogonal and $\|\hat{E}\|_2$ is rounding error small [45, Chap. 3]. Hence, if $\|E\|_2$ is rounding error small, then replacing V by $\tilde{V} = V(I+E)$ is a perturbation of similar character to and similar or smaller magnitude to the unavoidable rounding errors already contaminating V. Similarly, rounding errors in the computation of the Schur decomposition of H_{33} are equivalent to a perturbation of H_{33} that is typically at least as large as (and probably *less* structured than) $(I+E)^{-1}H_{33}(I+E)$ [29, p. 381]. Therefore, the components of $V_{1,k-m+1:k}$ may not be distinguished from zero within rounding error generated uncertainties. This suggests the next deflation strategy.

Window-Schur deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \le \varepsilon\|H_{32}\|_2$.

The numerical examples reported in this paper use both nearby-diagonal deflation and window-Schur deflation by setting small spike components to zero if either criterion is satisfied.

At this point it is safe to remark that aggressive early deflation equipped with any of the above strategies is a "normwise" backward numerically stable procedure. It uses only orthogonal matrix computations well known to be backward numerically stable in the presence of rounding errors [45], along with tiny perturbations that are equivalent to normwise tiny perturbations of the original matrix A.

2.5. Combining aggressive early deflation with small-subdiagonal deflation. In our experience, aggressive early deflation is more powerful than small-subdiagonal deflation. However, it does not replace small-subdiagonal deflation entirely. Occasionally, a tiny subdiagonal entry may appear outside of the deflation window. Such an opportunity for deflation goes undetected by aggressive early deflation. Even within the deflation window, aggressive early deflation may miss an opportunity to deflate that small-subdiagonal deflation does not. For example, let

$$H = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & \varepsilon & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix},$$

where $0 < \varepsilon \ll 1$. Suppose that ε is negligible (i.e., small enough to be safely set to zero), but $\sqrt{\varepsilon}/2$ is not. The small-subdiagonal strategy would deflate by setting ε to zero.

With a deflation window size of $k = 4$, H_{33} is the trailing principal 4-by-4 submatrix of H, and H_{32} is the first column of the 4-by-4 identity matrix. The eigenvalues of H_{33} are $2 \pm \varepsilon^{1/4}$ and $2 \pm i\varepsilon^{1/4}$ with corresponding almost-normalized right eigenvectors $[\pm\varepsilon^{1/2}, \varepsilon^{3/4}, \pm 1, \varepsilon^{1/4}]$ and $[\pm i\varepsilon^{1/2}, -\varepsilon^{3/4}, \mp i, \varepsilon^{1/4}]$. For any eigenvalue ordering in the Schur decomposition of H_{33}, the tip of the spike has magnitude $|s_4| = \sqrt{\varepsilon}\left(1 + \varepsilon^{1/2} + \varepsilon + \varepsilon^{3/2}\right)^{-1/2} > \sqrt{\varepsilon}/2$, which is not negligible.

It is inexpensive to monitor the subdiagonal entries during the course of a QR sweep and take advantage of a small-subdiagonal deflation should one occur. Note that this includes taking advantage of any small-subdiagonal entries that may appear between the bulges during a two-tone multishift QR sweep [9, 10]. In [42], this is called "vigilant deflation."

Ordinarily, the bulges above a vigilant deflation collapse as they encounter the newly created subdiagonal zero. This may block those shifts so that, for the current QR sweep, the benefit of using many simultaneous shifts may be lost. However, the bulges can be reintroduced, in the row in which the new zero subdiagonal appears, using the same methods that are used to introduce bulges at the upper left-hand corner [20, 24, 25], [29, p. 377], [45, p. 530]. In this way, the shift information passes through a zero subdiagonal and the two-tone QR sweep continues with all its shifts.

2.6. Choice of shifts. It is well known that a good choice of shifts in the QR algorithm leads to rapid convergence. Shifts selected to be the eigenvalues of a trailing principal submatrix give local quadratic convergence [45, 43]. In the symmetric case, convergence is cubic [46]. Deferred shifts retard convergence [40]. The implementation of the large-bulge multishift QR algorithm in LAPACK selects its shifts to be the eigenvalues of a trailing principal submatrix.

With aggressive early deflation, it is natural to select the shifts from among the nondeflatable eigenvalues. This saves an extra small eigenvalue calculation and incorporates more information from the matrix into the shifts. In the numerical examples reported in this paper, we arbitrarily select the shifts to be the nondeflatable eigenvalues that appear lowest along the diagonal of T in (2.6).

2.7. Iterated early deflation. In our experience, aggressive early deflation is so effective that it is often better to skip a multishift QR sweep and immediately apply the aggressive early deflation strategy again to the remaining, undeflated Hessenberg matrix. Aggressive early deflation can be applied over and over again, sometimes deflating a great many eigenvalues without the cost of a QR sweep. We find that it is best to skip the next multishift QR sweep whenever aggressive early deflation isolates more than a few eigenvalues.

2.8. Analysis of early deflation. This subsection gives a partial explanation of the success of aggressive early deflation.

Let $H_{33} = VTV^T$ be the real Schur decomposition in (2.6). Suppose that $t_{nn} = \lambda \in \mathbf{R}$ is a 1-by-1 diagonal block in the quasi-triangular structure of T, i.e., $\lambda = t_{nn}$ is a real eigenvalue of H_{33}. The early deflation strategy finds at least one deflatable eigenvalue if the tip of the spike, $s_k = h_{n-k+1,n-k}v_{1k}$, is small enough. (The tip of the spike is s_k, the last component of s in (2.6).)

Let $QR = (H_{33} - \lambda I)$ be a QR factorization. Because $H_{33} - \lambda I$ is a singular, unreduced Hessenberg matrix, $Q \in \mathbf{R}^{k\times k}$ is also unreduced Hessenberg and the last

row of the triangular factor R is zero. The last column of Q is a normalized left eigenvector of H_{33} corresponding to the eigenvalue λ. The unreduced structure of H_{33} also implies that λ has geometric multiplicity one and that its corresponding real, normalized left eigenvector is unique up to scaling by a complex number of modulus one. The last column of the matrix of Schur vectors V is also a normalized left eigenvector corresponding to λ, and, in particular, $|q_{1k}| = |v_{1k}|$. The tip of the spike may be written as $|s_k| = |h_{n-k+1,n-k}v_{1k}| = |h_{n-k+1,n-k}q_{1k}|$.

The matrix Q is Hessenberg, so the $(1,k)$th cofactor is

$$(-1)^{k+1}Q(1|k) = (-1)^{1+k}\prod_{i=1}^{k-1} q_{i+1,i}.$$

However, Q is also orthogonal, so

$$Q^{-1} = Q^T = \mathrm{adj}(Q)/\det(Q).$$

(Here, $\mathrm{adj}(Q)$ is the classical adjoint matrix or adjugate of Q.) Hence,

$$q_{1k} = \frac{(-1)^{k+1}\prod_{i=1}^{k-1} q_{i+1,i}}{\det(Q)}.$$

If $\bar{q}$ is the geometric mean

$$\bar{q} = \left|\prod_{i=1}^{k-1} q_{i+1,i}\right|^{1/(k-1)},$$

then, because $|\det(Q)| = 1$, the tip of the spike has modulus

$$|s_k| = |h_{n-k+1,n-k}q_{1k}| = |h_{n-k+1,n-k}|\,\bar{q}^{k-1}.$$

The $q_{i+1,i}$'s are entries in an orthogonal matrix, so for each i, $|q_{i+1,i}| \le 1$ and, hence, $\bar{q} \le 1$. Even when $\bar{q}$ is only moderately smaller than one, $\bar{q}^{k-1}$ may be tiny. For example, if $\bar{q} \le 1/2$ and the deflation window size is, say, $k > 50$, then $\bar{q}^{k-1} \le 9\times10^{-16}$ and the tip of the spike, $|s_k| \le 9 \times 10^{-16}\,|h_{n-k+1,n-k}|$, may well be small enough to set to zero. Note that this can occur even when no subdiagonal of Q is particularly small and when many have modulus one or close to one.

The geometric mean of the subdiagonal entries of H_{33}, and $h_{n-k+1,n-k}$,

$$\bar{h} = \left|\prod_{j=n-k}^{n-1} h_{j+1,j}\right|^{1/k},$$

is proportional to $\bar{q}$. To see this, note that the Hessenberg, orthogonal structure of Q, the upper triangular structure of R, $QR = (H_{33} - \lambda I)$, implies that for $i = 1, 2, 3, \ldots, n-1$,

$$q_{i+1,i}r_{ii} = h_{n-k+i+1,n-k+i}$$

and

$$(H_{33} - \lambda I)(k|k) = Q(k|k)R(k|k).$$

Also, because $Q^T = Q^{-1} = \mathrm{adj}(Q)/\det(Q)$, $|q_{kk}| = |Q(k|k)|$. The tip of the spike can then be expressed as

$$\begin{aligned} |s_k| &= |h_{n-k+1,n-k}|\,\bar{q}^{k-1} \\ &= |h_{n-k+1,n-k}| \prod_{i=1}^{k-1} \left| \frac{h_{n-k+i+1,n-k+i}}{r_{ii}} \right| \\ &= \frac{\bar{h}^k}{|R(k|k)|} \qquad (2.7) \\ &= \frac{|q_{kk}|\,\bar{h}^k}{|(H_{33} - \lambda I)(k|k)|}. \qquad (2.8) \end{aligned}$$

This expression can be simplified under the assumption that λ has algebraic multiplicity one. In that case, let $H_{33} = XJX^{-1}$ be the Jordan canonical form of H_{33} ordered so that the 1-by-1 Jordan block corresponding to λ appears in the lower right-hand corner of J. For notational convenience, set $Z = X^{-1}$, and partition $H_{33} - \lambda I = X(J - \lambda I)Z$ as

$$H_{33} - \lambda I = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} (J_{11} - \lambda I) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix},$$

where the $(1,1)$ blocks are $(k-1)$-by-$(k-1)$, the $(1,2)$ blocks are $(k-1)$-by-1, the $(2,1)$ blocks are 1-by-$(k-1)$, and the $(2,2)$ blocks are 1-by-1. It follows that $(H_{33} - \lambda I)(k|k) = \det(X_{11}(J_{11} - \lambda I)Z_{11})$. Now, $Z = X^{-1} = \mathrm{adj}(X)/\det(X)$ and, in particular, $Z_{22} = \det(X_{11})/\det(X)$. Similarly, $X_{22} = \det(Z_{11})\det(X)$. Without loss of generality, we may choose X and $Z = X^{-1}$ such that those rows of Z which are left eigenvectors have 2-norm equal to one. The last row of Z is a normalized left eigenvector of H_{33} corresponding to the eigenvalue λ, so $|q_{kk}| = |Z_{22}| = |\det X_{11}/\det X|$. Hence, (2.8) becomes

$$\begin{aligned} |s_k| &= \frac{|q_{kk}|\,\bar{h}^k}{|(H_{33} - \lambda I)(k|k)|} \\ &= \frac{|q_{kk}|\,\bar{h}^k}{|\det(X_{11}(J_{11} - \lambda I)Z_{11})|} \\ &= \frac{\left|\bar{h}^k\right|}{|\det(J_{11} - \lambda I)\det(X)\det(Z_{11})|} \\ &= \frac{\left|\bar{h}^k\right|}{|\det(J_{11} - \lambda I)|\;|X_{22}|}. \end{aligned}$$

Finally, if $\bar{\mu}$ is the geometric mean of the differences between λ and the other eigenvalues of H_{33},

$$\bar{\mu} = |\det(J_{11} - \lambda I)|^{1/(k-1)},$$

then

$$|s_k| = \frac{\bar{h}^k}{\bar{\mu}^{k-1}\,|X_{22}|},$$

where X_{22} is the last component of a right eigenvector corresponding to eigenvalue λ.

Like $\bar{q}^{k-1}$, $\bar{h}^k$ may be tiny even when no subdiagonal $h_{j+1,j}$ is particularly small. If λ is well separated from the other eigenvalues H_{33} in the sense that $\bar{\mu}^{k-1}\,|X_{22}|$ is not "too small," then even a moderately small geometric mean $\bar{h}$ may make λ a deflatable eigenvalue.

The above observations apply equally well to all normalized left eigenvectors, but, of course, not all of them will have tiny first components (except, perhaps, in the case of a highly ill-conditioned eigenvalue). Suppose H_{33} is diagonalizable and that m of the normalized left eigenvectors of H_{33} have tiny first components. Stack the left eigenvectors as the rows of a matrix Y with the distinguished set of m vectors on the bottom. Now, $H_{33} = Y^{-1}DY$, where D is the diagonal matrix of eigenvalues. If $H_{33} = VTV^H$ is a complex Schur decomposition with eigenvalues ordered along the diagonal of T in the same order as along the diagonal of D, then $R = YV$ is upper triangular and $R^{-1}Y_{:,1} = V_{1,:}^H$. In particular, the trailing m components of the first row of V can be bounded in terms of the corresponding components of Y and the trailing m-by-m principal submatrix of R as

$$\begin{aligned}\|V_{k-m+1:k,1}^H\|_2 &= \|R_{k-m+1:k,k-m+1:k}^{-1} Y_{k-m+1:k,1}\|_2 \\ &\le \|R_{k-m+1:k,k-m+1:k}^{-1}\|_2 \|Y_{k-m+1:k,1}\|_2.\end{aligned}$$

By assumption, $\|Y_{k-m+1:k,1}\|_2$ is tiny. So is $\|V_{1,k-m+1:k}\|_2$, if $\|R_{k-m+1:k,k-m+1:k}^{-1}\|_2$ is not "too big," i.e., if no eigenvalue corresponding to the selected m left eigenvectors is "too ill-conditioned."

3. Numerical examples. We compared the performance of the large-bulge multishift QR algorithm [3] and the small-bulge multishift QR algorithm [9, 10] with and without aggressive early deflation. The test matrices included ad hoc and pseudo-random Hessenberg matrices of order 500-by-500 to 10,000-by-10,000 and nonrandom matrices of similar order taken from a variety of applications in science and engineering [2].

Computational environment. The numerical examples were run on an Origin2000 computer equipped with 400MHz IP27 R12000 processors and 16 gigabytes of memory. Each processor uses 32 kilobytes of level 1 instruction cache, 32 kilobytes of level 1 data cache, and 8 megabytes of level 2 combined instruction and data cache. For serial execution, the experimental Fortran implementation of the small-bulge multishift QR algorithm was compiled with version 7.30 of the MIPSpro Fortran 77 compiler with options `-64 -TARG:platform=ip27 -Ofast=ip27 -LNO`. The same options were used to compile `DHSEQR` from LAPACK version 2. For parallel execution the `-mp` and `-pfa` options were added. The programs called optimized LAPACK and BLAS subroutines from the SGI/Cray Scientific Library version 1.2.0.0.

In our computational environment, we observed that the measured serial "cpu time" of any particular program with its particular data might vary by at most a few percent. We were fortunate to get exclusive use of several processors for the purpose of timing parallel benchmark runs.

Except where otherwise mentioned, n-by-n matrices were stored in n-by-n arrays.

We report the floating point execution rate in millions of floating point instructions per second or "mega-flops" for short. (A trinary multiply-add operation counts as one instruction although it executes two flops.) For comparison purposes, we measured the floating point execution rate of the level 3 BLAS matrix-matrix multiply subroutine `DGEMM` and triangular matrix multiply subroutine `DTRMM` from the SGI/Cray Scientific Library version 1.2.0.0 applied to matrix products similar to those

that dominate the small-bulge multishift QR algorithm [9, 10]. In serial execution, in the Origin2000 computational environment described above, `DGEMM` computes the product of the transpose of a 200-by-200 matrix times a 200-by-10,000 slab embedded a 10,000-by-10,000 array at roughly 330 mega-flops. It computes the product of a 10,000-by-200 slab times a 200-by-200 matrix at roughly 325 mega-flops. `DTRMM` computes the product of the transpose of a triangular 200-by-200 matrix times a 200-by-10,000 slab at roughly 305 mega-flops if it is an upper triangular matrix and at roughly 260 mega-flops if it is a lower triangular matrix. `DTRMM` computes the product of a 10,000-by-200 slab times a 200-by-200 triangular matrix at roughly 275 mega-flops if it is an upper triangular matrix and at roughly 255 mega-flops if it is a lower triangular matrix.

Implementation details. We call our experimental Fortran implementation of the small-bulge multishift QR algorithm *without* aggressive early deflation `TTQR` and *with* aggressive early deflation `TTQRE`. As described in [9, 10], the small-bulge multishift QR algorithm avoids the phenomenon of shift-blurring by chasing a tightly packed chain of m small bulges [9, 10]. Both `TTQR` and `TTQRE` use vigilant small-subdiagonal deflation [42]. Following EISPACK [36] and LAPACK [1], a Hessenberg subdiagonal entry $h_{i+1,j}$ is set to zero when $|h_{i+1,i}| \leq \varepsilon\,(|h_{ii}| + |h_{i+1,i+1}|)$ with ε equal to the unit roundoff.

The experimental implementation of `TTQRE` uses both nearby-diagonal deflation and window-Schur deflation by setting small spike components to zero if either criterion is satisfied. If the early deflation procedure with a k-by-k deflation window isolates $15k/100$ or more eigenvalues, then `TTQRE` skips the next QR sweep and immediately applies the early deflation procedure again to the remaining unreduced Hessenberg submatrix. In this way, `TTQRE` is sometimes able to isolate a great many eigenvalues without the expense of a QR sweep.

Both `TTQR` and `TTQRE` use LAPACK subroutine `DHSEQR` [1], the conventional large-bulge QR algorithm, to reduce diagonal subblocks of order no greater than 1.5 times the number of simultaneous shifts.

For the large-bulge multishift QR algorithm, we use subroutine `DHSEQR` from LAPACK version 2 which is widely recognized to be an excellent implementation. For the large-bulge multishift QR algorithm with aggressive early deflation, we modified `DHSEQR` by inserting aggressive early deflation following the search for small subdiagonals. The modified program performs a large-bulge QR sweep only if no deflations are found. We call the resulting program `DHSEQRE`.

For reference, Table 2 lists names and short descriptions of the four algorithms.

Choosing shift multiplicity and size of the deflation window. As of this writing, it is not well understood how best to choose the number of simultaneous shifts and the size of the deflation window. The relationship between shift multiplicity, deflation window size, and execution time is a complex interaction between the character of the Hessenberg matrix, the hardware architecture of the computational environment, and the not-yet-well-understood convergence behavior of aggressive early deflation. Cache size, cache strategy, and placement of data in machine memory can have a strong effect on execution time.

In Example 1, we did extensive preliminary experiments to determine good choices of these parameters for each algorithm. However, these choices may or may not perform well on a computer with a different architecture or when applied to Hessenberg matrices of different character or different order. In Example 2, we did extensive

TABLE 2
Names and descriptions of the algorithms in the numerical experiments described in section 3.

Name	Description
`DHSEQR`:	The large-bulge multishift QR algorithm [3] using only small-subdiagonal deflation as implemented in LAPACK 2.0 [1].
`DHSEQRE`:	The large-bulge multishift QR algorithm [3] using both small-subdiagonal and aggressive early deflation.
`TTQR`:	The two-tone, small-bulge multishift QR algorithm [9, 10] using only small-subdiagonal deflation.
`TTQRE`:	The two-tone, small-bulge multishift QR algorithm [9, 10] using both small-subdiagonal deflation and aggressive early deflation.

preliminary experiments on the order $n = 5{,}000$ example only. For the other examples, the choices are based upon a few preliminary experiments and ad hoc educated guesses. Consequently, no particular significance should be attached to the shift multiplicities and deflation window sizes used in this paper.

In the examples reported here, we use a fixed number of simultaneous shifts and a fixed deflation window size throughout. However, there may be an advantage in choosing these parameters dynamically as the algorithm progresses.

Example 1. We ran `DHSEQR`, `DHSEQRE`, `TTQR`, and `TTQRE` on pseudorandomly generated Hessenberg matrices of various sizes from 500-by-500 to 1,000-by-1,000. We selected the entries on the diagonal and upper triangle to be normally distributed pseudorandom numbers with mean zero and variance one. We set the subdiagonal entry $h_{j,j+1} = \sqrt{\chi^2_{n-j}}$, where χ^2_{n-j} is selected from a Chi-squared distribution with $n - j$ degrees of freedom. These pseudorandom Hessenberg matrices have essentially the same distribution as if a matrix of normally distributed, mean zero, variance one pseudorandom variables had been reduced to Hessenberg form using [29, Algorithm 7.4.2].

Figure 1 displays the serial execution time, rate of floating point instruction execution, and hardware count of executed floating point instructions for pseudorandom Hessenberg test matrices of orders $n = 500$ to $n = 1{,}000$ using the Origin2000 computer described above.

To choose the number of simultaneous shifts and the deflation window size, we ran preliminary experiments using a wide variety of simultaneous shifts and deflation window sizes. Good choices of the parameters were usually different for each of the different algorithms. Even for individual algorithms no single choice minimizes execution time over the whole range of orders $n = 500$ to $n = 1{,}000$. However, the following choices result in execution times that are no more than 10% longer than the minimum that we observed. In Figure 1, `DHSEQR` uses 6 simultaneous shifts, `DHSEQRE` uses 22 simultaneous shifts, and both `TTQR` and `TTQRE` use 60 simultaneous shifts. `DHSEQRE` uses a 48-by-48 deflation window and `TTQRE` uses a 90-by-90 deflation window. (The execution time of no algorithm would be reduced by changing the number of simultaneous shifts or the deflation window size to the choice used by one of the other algorithms. For example, reducing the number of simultaneous shifts used by `DHSEQRE` to 6 increases its execution time between 10% and 50%. Increasing the number of simultaneous shifts used by `DHSEQR` to 22 increases its execution time

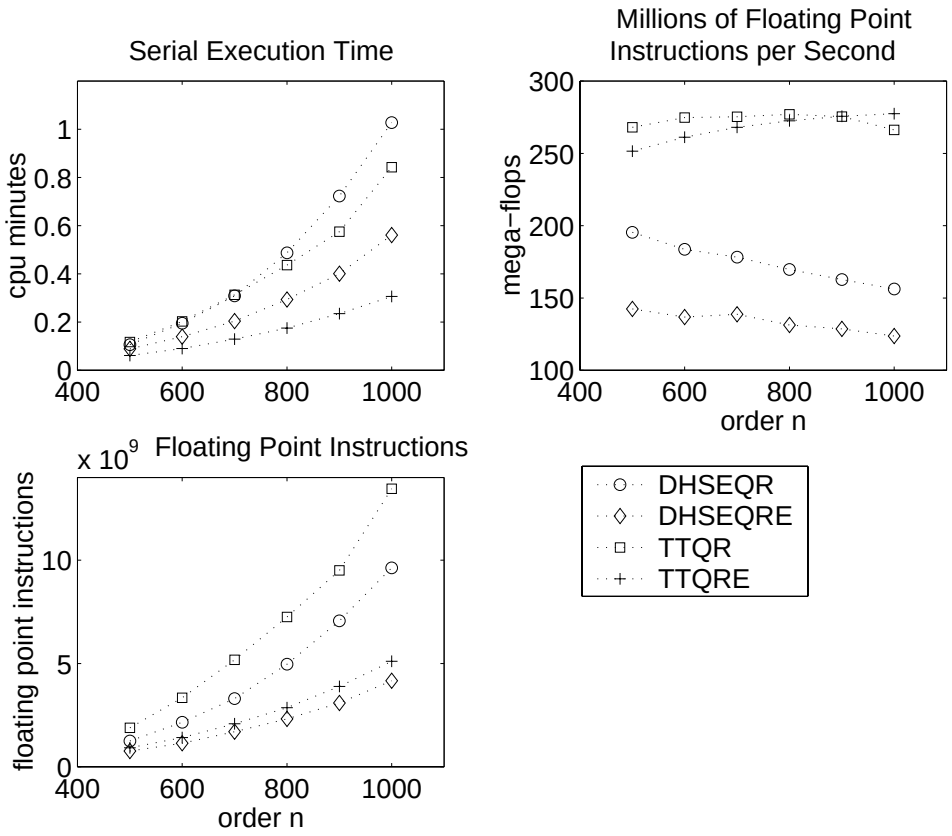


FIG. 1. *Serial cpu execution time, rate of floating point instruction execution, and hardware count of floating point instructions executed by* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing the quasi-triangular and orthogonal Schur factors of pseudorandom Hessenberg matrices. In this figure,* `DHSEQR` *uses* 6 *simultaneous shifts,* `DHSEQRE` *uses* 22 *simultaneous shifts, and both* `TTQR` *and* `TTQRE` *use* 60 *simultaneous shifts.* `DHSEQRE` *uses a* 48*-by-*48 *deflation window and* `TTQRE` *uses a* 90*-by-*90 *deflation window.*

between 35% and 95%.)

To compare the effects of rounding errors in each of the algorithms, we computed the relative residual $\|A\tilde{Q}-\tilde{Q}\tilde{T}\|_F/\|A\|$ and the departure from orthogonality $\|\tilde{Q}^T\tilde{Q}-I\|_F/\sqrt{n}$, where $\tilde{Q}$ is the computed nearly orthogonal factor and $\tilde{T}$ is the computed quasi-triangular factor. (The matrix $\tilde{Q}$ would be orthogonal were it not for rounding errors.) For all tested matrix orders between 500 and 1,000, and for all four algorithms, the relative residuals and departure from orthogonality lie between $\frac{1}{2}\times 10^{-14}$ and 2×10^{-14}. This compares well with the unit roundoff of the finite precision arithmetic of 2.22×10^{-16}. As measured by "normwise backward error," the four algorithms are empirically numerically stable and of roughly equal accuracy.

Figure 1 demonstrates that aggressive early deflation is effective at reducing both the execution time and the number of floating point instructions executed by both the large- and small-bulge multishift QR algorithm. In addition, the level 3 BLAS based small-bulge multishift QR algorithm used by `TTQR` and `TTQRE` [9, 10] maintains a relatively high rate of execution of floating point instructions compared to the level 2 BLAS based large-bulge QR algorithm used by `DHSEQR` and `DHSEQRE`.

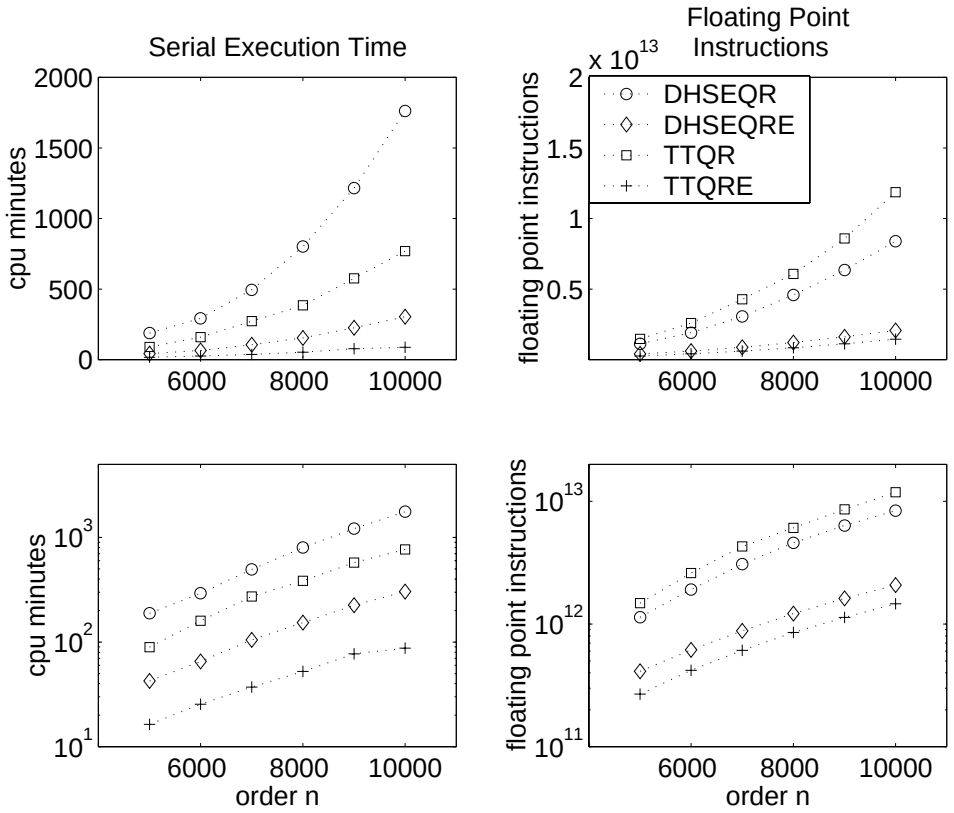


FIG. 2. *Serial cpu execution time of and floating point instructions executed by* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of pseudorandom Hessenberg matrices. The same data is displayed on linear and semilog plots. In this figure, both* `DHSEQR` *and* `DHSEQRE` *use* 8 *simultaneous shifts.* `DHSEQRE` *uses a* 150*-by-*150 *deflation window.* `TTQR` *uses* 150 *simultaneous shifts per QR sweep.* `TTQRE` *uses* 200 *simultaneous shifts with a* 450*-by-*450 *deflation window. Computational costs for the reduction to Hessenberg form are not included.*

Example 2. We repeated the previous numerical experiment using pseudorandom Hessenberg matrices of orders between $n = 5{,}000$ and $n = 10{,}000$.

To choose the number of simultaneous shifts and the deflation window size, we ran preliminary experiments on pseudorandom matrices of order $n = 5{,}000$ using a wide variety of choices of these parameters. The following choices yield the minimum execution times that we observed for the order $n = 5{,}000$ test matrix. In Figure 2, both `DHSEQR` and `DHSEQRE` use 8 simultaneous shifts. `TTQR` uses 150 simultaneous shifts, and `TTQRE` uses 200 simultaneous shifts. `DHSEQRE` used a 150-by-150 deflation window, and `TTQRE` used a 450-by-450 deflation window.

In this example, `TTQR` and `TTQRE` execute between 240 and 280 million floating point instructions per second (counting a trinary multiply-add operation as one instruction) with an average of 260 and 270, respectively. There is no obvious trend in the execution rates. The rate of floating point instruction execution by `DHSEQR` gradually declines from 100 million floating point instructions per second for the 5,000-by-5,000 example down to 80 million floating point instructions per second for the 10,000-by-10,000 example. The rate of floating point instruction execution by

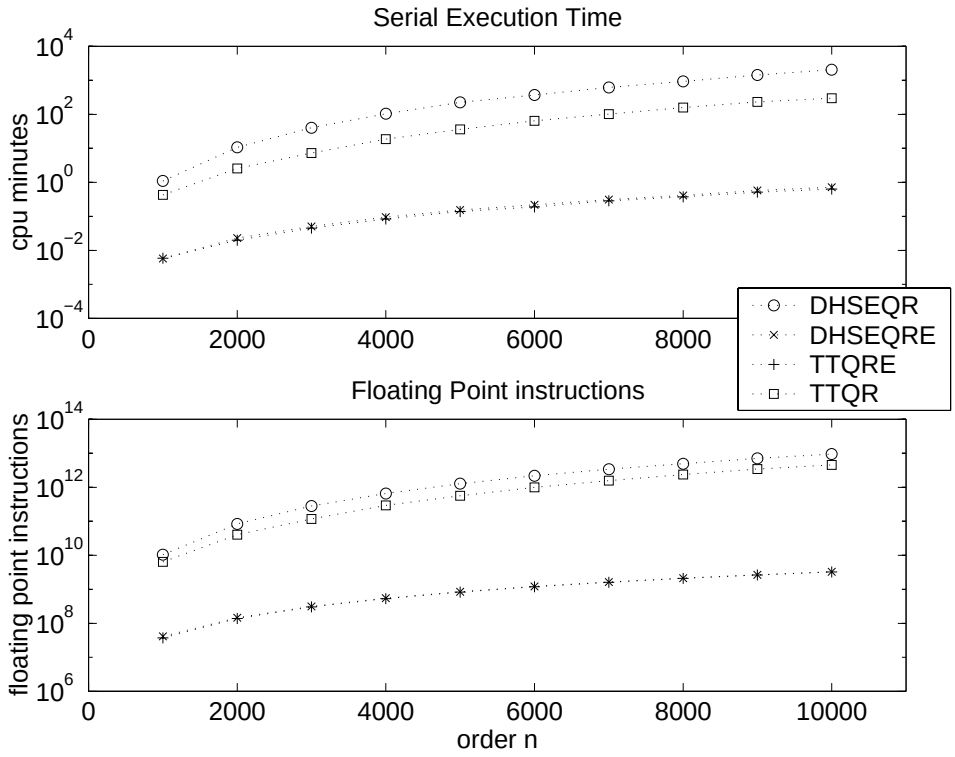


FIG. 3. *Serial execution time, number of floating point instructions executed, and floating point execution rate of* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of matrices of the form of* (1.1)*. The two algorithms* `DHSEQRE` *and* `TTQRE` *perform essentially identical operations, so their graphs coincide.* `DHSEQR` *uses* 6 *simultaneous shifts.* `TTQR` *uses* 150 *simultaneous shifts. Both* `DHSEQRE` *and* `TTQRE` *use a tiny* 10*-by-*10 *deflation window. (Similar results are obtained using larger deflation windows.)*

`DHSEQRE` gradually declines from roughly 160 down to 113 million floating point instructions per second.

As n increases from 5,000 to 10,000, `DHSEQR` took 12 to 20 times longer, `DHSEQRE` took 2.6 to 3.5 times longer, and `TTQR` took 5.5 to 8.8 times longer than `TTQRE`. In addition, `DHSEQR` executed 4.2 to 5.7 times as many, `DHSEQRE` executed 1.4 to 1.5 times as many, and `TTQR` executed 5.5 to 8.1 times as many floating point instructions as `TTQRE`.

Example 3. This example shows aggressive early deflation at its best. Figure 3 displays serial execution time and hardware count of executed floating point instructions for `DHSEQR`, `DHSEQRE`, `TTQR`, and `TTQRE` applied to matrices of the form of (1.1) of orders $n = 1{,}000$ to $n = 10{,}000$. `DHSEQR` uses 6 simultaneous shifts. `TTQR` uses 150 simultaneous shifts. Both `DHSEQRE` and `TTQRE` use a tiny 10-by-10 deflation window. (Similar results are obtained using larger deflation windows.)

In this example, `DHSEQRE` and `TTQRE` complete the Schur decomposition in 0.04% to 0.5% of the execution time used by `DHSEQR` and in 0.25% to 1% of the execution time used by `TTQR`. `TTQR` maintains a floating point execution rate of over 250 million floating point instructions per second throughout. Execution rates for the other three programs drop from roughly 120 million floating point instructions per second for the

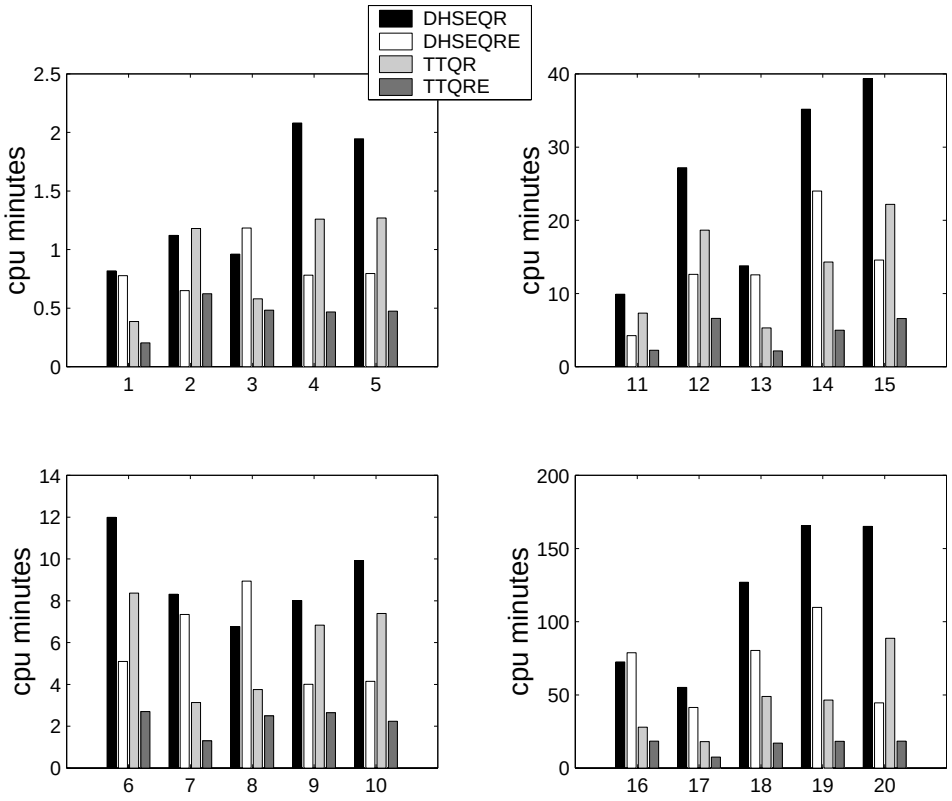


FIG. 4. *Serial execution times of* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing the orthogonal and quasi-triangular factor of the Schur decompositions of* 20 *non-Hermitian eigenvalue selected from* [2]*. The numbers along the bottom edge of the bar graphs correspond to the numbers in the left-hand column of Table* 3*. Both* `DHSEQR` *and* `DHSEQRE` *use* 6 *simultaneous shifts.* `DHSEQRE` *uses a* 50*-by-*50*,* 100*-by-*100*, or* 150*-by-*150 *deflation window when the order* n *of the test matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 4{,}000$*, and* $4{,}000 < n$*, respectively.* `TTQR` *uses* 60*,* 116*,* 150*, or* 180 *simultaneous shifts when the order* n *of the text matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 2{,}500$*,* $2{,}500 \le n < 4{,}000$*, and* $4{,}000 \le n$*, respectively.* `TTQRE` *uses* 90*,* 120*,* 180*,* 240*, or* 270 *simultaneous shifts when the order* n *of the test matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 2{,}500$*,* $2{,}500 \le n < 4{,}500$*,* $4{,}500 \le n < 7{,}000$*, and* $7{,}000 < n$*, respectively. As usual,* `TTQRE` *uses a deflation window of order* 1.5 *times the number of simultaneous shifts. (The execution time of the reduction to Hessenberg form is not included.)*

$n = 1{,}000$ example down to roughly 75 million floating point instructions per second for the $n = 10{,}000$ example.

The remarkable performance of `DHSEQRE` and `TTQRE` is due entirely to aggressive early deflation. In our experimental implementation of aggressive early deflation, if more than a few eigenvalues are isolated, then `TTQRE` and `DHSEQRE` skip the next QR sweep and immediately apply aggressive early deflation to the remaining unreduced Hessenberg submatrix. In this way, `TTQRE` and `DHSEQRE` complete the entire Schur decomposition of (1.1) without once performing a multishift QR sweep outside of a deflation window! This explains why `DHSEQRE` and `TTQRE` show nearly identical execution time and floating point instruction count. It also explains `TTQRE`'s uncharacteristically low floating point execution rate. Most of its level 3 BLAS operations lie in the unexecuted small-bulge multishift QR sweep.

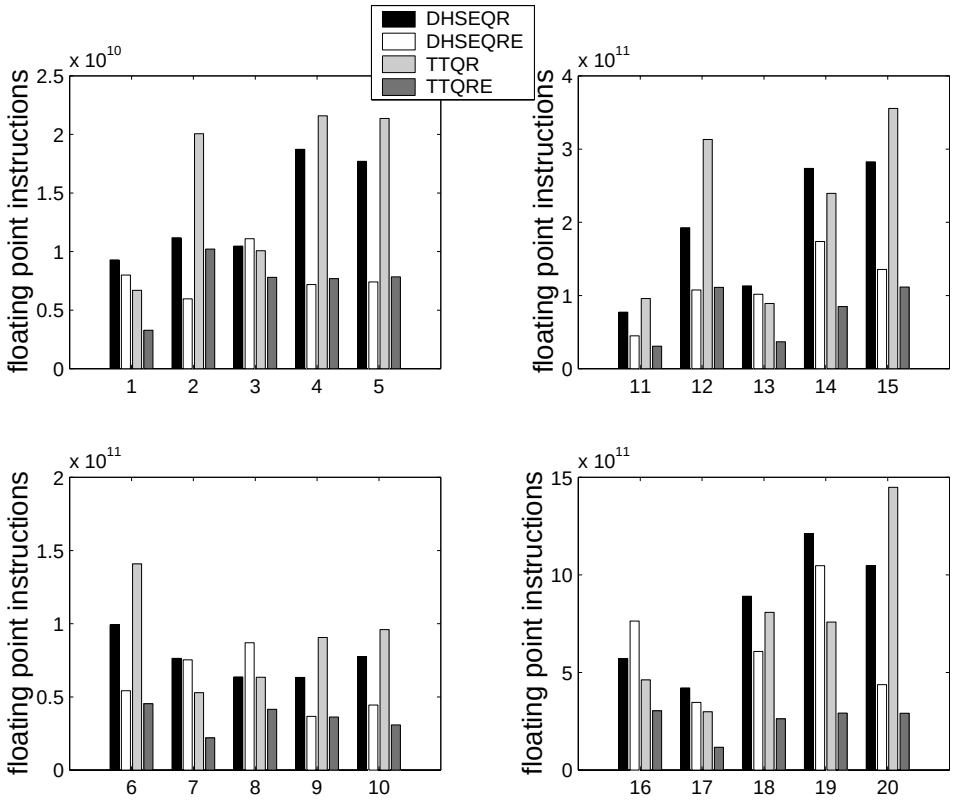


FIG. 5. *Hardware count of floating point instructions executed by* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing the orthogonal and quasi-triangular factor of the Schur decompositions of* 20 *non-Hermitian eigenvalue problems selected from* [2]*. The numbers along the bottom edge of the bar graphs correspond to the numbers in the left-hand column of Table* 3*. Both* `DHSEQR` *and* `DHSEQRE` *use* 6 *simultaneous shifts.* `DHSEQRE` *uses a* 50*-by-*50*,* 100*-by-*100*, or* 150*-by-*150 *deflation window when the order* n *of the test matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 4{,}000$*, and* $4{,}000 < n$*, respectively.* `TTQR` *uses* 60*,* 116*,* 150*, or* 180 *simultaneous shifts when the order* n *of the text matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 2{,}500$*,* $2{,}500 \le n < 4{,}000$*, and* $4{,}000 \le n$*, respectively.* `TTQRE` *uses* 90*,* 120*,* 180*,* 240*, or* 270 *simultaneous shifts when the order* n *of the test matrix falls in the range* $1{,}000 \le n < 2{,}000$*,* $2{,}000 \le n < 2{,}500$*,* $2{,}500 \le n < 4{,}500$*,* $4{,}500 \le n < 7{,}000$*, and* $7{,}000 < n$*, respectively. As usual,* `TTQRE` *uses a deflation window of order* 1.5 *times the number of simultaneous shifts. (Floating point instructions executed during the reduction to Hessenberg form are not included.)*

It is easy to show that in examples like this, if aggressive early deflation eliminates the need for all or nearly all multishift QR sweeps, the amount of arithmetic work needed to compute the n-by-n Hessenberg Schur decomposition grows as $O(n^2)$.

Example 4. Figures 4 and 5 display the serial execution times and hardware count of executed floating point instructions of `DHSEQR`, `DHSEQRE`, `TTQR`, and `TTQRE` applied to the real non-Hermitian eigenvalue problems from the NEP collection [2] that are listed in Table 3. Each set of four bars in Figures 4 and 5 is labeled at the bottom by the number of the corresponding test matrix in Table 3. Summaries of the applications, descriptions of the matrices, and references can be found in [2].

To avoid cache conflicts, each n-by-n matrix is stored in an $(n+7)$-by-$(n+7)$ array. Both `DHSEQR` and `DHSEQRE` use 6 simultaneous shifts. `DHSEQRE` uses a 50-by-50,

TABLE 3

Matrices selected from the collection of non-Hermitian eigenvalue problems [2]. *The numbers along the bottom edge of the bar graphs in Figures* 4 *and* 5 *correspond to the numbers in the left-hand column.*

	Acronym	Order n	Discipline
1	OLM1000	1,000	Hydrodynamics
2	TUB1000	1,000	Computational fluid dynamics
3	TOLS1090	1,090	Aeroelasticity
4	RDB1250	1,250	Chemical engineering
5	RDB1250L	1,250	Chemical engineering
6	BWM2000	2,000	Chemical engineering
7	OLM2000	2,000	Hydrodynamics
8	TOLS2000	2,000	Aeroelasticity
9	DW2048	2,048	Electrical engineering
10	RDB2048	2,048	Chemical engineering
11	RDB2048L	2,048	Chemical engineering
12	PDE2961	2,961	Partial differential equations
13	MHD3200A	3,200	Plasma physics
14	MHD3200B	3,200	Plasma physics
15	RDB3200L	3,200	Chemical engineering
16	TOLS4000	4,000	Aeroelasticity
17	MHD4800A	4,800	Plasma physics
18	MHD4800B	4,800	Plasma physics
19	OLM5000	5,000	Hydrodynamics
20	RW5151	5,151	Probability
21	DW8192	8,192	Electrical engineering

100-by-100, or 150-by-150 deflation window when the order n of the test matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 4{,}000$, and $4{,}000 < n$, respectively. `TTQR` uses 60, 116, 150, or 180 simultaneous shifts when the order n of the text matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 2{,}500$, $2{,}500 \le n < 4{,}000$, and $4{,}000 \le n$, respectively. `TTQRE` uses 90, 120, 180, 240, or 270 simultaneous shifts when the order n of the test matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 2{,}500$, $2{,}500 \le n < 4{,}500$, $4{,}500 \le n < 7{,}000$, and $7{,}000 < n$, respectively. As usual, `TTQRE` uses a deflation window of order 1.5 times the number of simultaneous shifts.

In Figure 4 the median ratio of `DHSEQRE`'s execution time to `DHSEQR`'s execution time is .58. The median ratio of `TTQR`'s execution time to `DHSEQR`'s execution time is .56. The median ratio of `TTQRE`'s execution time to `DHSEQR`'s execution time is .23.

Matrix number 21 in Table 3, DW8192, is not reported in Figures 4 and 5 only because the execution times are out of scale with the other reported times. In the Origin2000 computational environment described above, `DHSEQR` calculates the Schur decomposition (including both quasi-triangular and orthogonal factors) of the Hessenberg matrix derived from DW8192 in 544 cpu minutes; `DHSEQRE` uses 300 minutes; `TTQR` uses 301 cpu minutes; and `TTQRE` uses 120 cpu minutes.

Example 5. Our experimental implementation of `TTQRE` is not well tuned for parallel computation. However, it does make heavy use of the level 3 BLAS (particularly matrix-matrix multiply), so it is not surprising to observe modest but not insignificant speedups when the experimental version of `TTQRE` is compiled for parallel execution and linked with parallel versions of the BLAS.

Figure 6 displays wall clock execution times and parallel speedups for `TTQRE` applied to the pseudorandom Hessenberg matrices described in Example 1. (Parallel speedup is the ratio T_1/T_p, where T_1 is the 1 processor wall clock execution time and T_p is the p-processor wall clock execution time.) We were fortunate to get exclusive use of

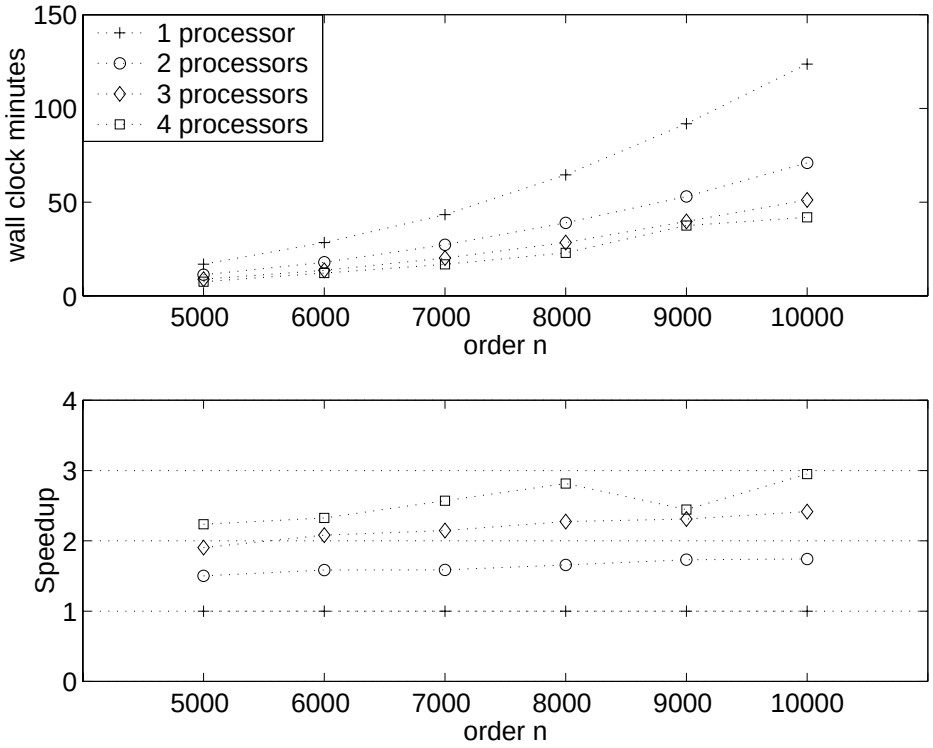


FIG. 6. *Wall clock execution time and parallel speedups of* `TTQRE` *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of pseudorandom Hessenberg matrices. (The execution time of the reduction to Hessenberg form is not included.)*

several processors for the purpose of timing parallel benchmark runs. In this example, `TTQRE` uses 150 simultaneous shifts along with a 226-by-226 deflation window. To avoid cache conflicts, some n-by-n matrices were stored in $(n+1)$-by-$(n+1)$ arrays.

The parallel speedups in Figure 6 are modest. For a 5,000-by-5,000 pseudorandom Hessenberg matrix, `TTQRE` computes the quasi-triangular and orthogonal Schur factors in approximately 9% of the time needed by `DHSEQR`. With four processor parallelism execution time drops to only 4%. For a 10,000-by-10,000 pseudorandom Hessenberg matrix, `TTQRE` computes the quasi-triangular and orthogonal Schur factors in approximately 7% of the time needed by `DHSEQR`. With four processor parallelism execution time drops to only 2%.

Our experimental implementation of `TTQRE` has one or more serial bottle necks. The worst of these can be traced to the "near diagonal" portion of the small-bulge multishift QR algorithm [9, 10]. A tuned production version of `TTQRE` that is designed for parallel computation is in progress.

A similar numerical experiment using `TTQR` with a figure showing parallel execution times and speedups appears in [9].

4. Conclusion. Aggressive early deflation recognizes converged eigenvalues before classical small-subdiagonal deflation would. In experiments with random Hessenberg matrices and with Hessenberg matrices from a variety of engineering and scientific applications [2], it significantly reduces both the number of floating point

instructions and the execution time needed by both the large- and small-bulge multishift QR algorithm. Sometimes, aggressive early deflation reduces execution time from hours down to minutes. In experiments with n-by-n Hessenberg matrices of the form of (1.1), aggressive early deflation computes the Schur decomposition using only $O(n^2)$ flops.

Aggressive early deflation is both theoretically and empirically "normwise" backward numerically stable.

Although aggressive early deflation is effective in combination with conventional QR algorithms, the combination of aggressive early deflation with the two-tone, small-bulge multishift QR algorithm [9, 10] takes advantage of the capabilities of advanced architecture computers to sustain a high floating point instruction execution rate and attain at least modest parallel speedups.

Acknowledgment. The authors would like to thank David Watkins for helpful discussions and for bringing [42] to their attention.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, 1992.

[2] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, Tech. report, Department of Mathematics, University of Kentucky, Lexington, KY. Also available online from http://math.nist.gov/MatrixMarket.

[3] Z. Bai and J. Demmel, *On a block implementation of Hessenberg QR iteration*, Intl. J. of High Speed Comput., 1 (1989), pp. 97–112. Also available online as LAPACK Working Note 8 from http://www.netlib.org/lapack/lawns/lawn08.ps and http://www.netlib.org/lapack/lawnspdf/lawn08.pdf.

[4] Z. Bai and J. Demmel, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95. Also available online as LAPACK Working Note 54 from http://www.netlib.org/lapack/lawns/lawn54.ps and http://www.netlib.org/lapack/lawnspdf/lawn54.pdf.

[5] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[6] D. Boley, *Computing rank-deficiency of rectangular matrix pencils*, Systems Control Lett., 9 (1987), pp. 207–214.

[7] D. Boley, *Estimating the sensitivity of the algebraic structure of pencils with simple eigenvalue estimates*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 632–643.

[8] D. Boley and W. Lu, *Measuring how far a controllable system is from an uncontrollable one*, IEEE Trans. Automat. Control, 31 (1986), pp. 249–251.

[9] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part* I: *Maintaining well-focused shifts and level* 3 *performance*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 929–947.

[10] K. Braman, R. Byers, and R. Mathias, *The Multi-Shift QR-Algorithm: Aggressive Deflation, Maintaining Well Focused Shifts, and Level* 3 *Performance*, Tech. Report 99-05-01, Department of Mathematics, University of Kansas, Lawrence, KS, 1999. Also available online from http://www.math.ukans.edu/~reports/1999.html.

[11] R. Byers, *Numerical condition of the algebraic Riccati equation*, Contemp. Math., 47 (1985), pp. 35–49.

[12] R. Byers, *Detecting nearly uncontrollable pairs*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, Proceedings of the International Symposium MTNS-89, Vol. 3, Amsterdam, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhäuser, Boston, 1990, pp. 447–457.

[13] J. Demmel, *A Lower Bound on the Distance to the Nearest Uncontrollable System*, Tech. report, Courant Institute, Computer Science Dept., New York University, New York, 1987.

[14] J. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251–289.

[15] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[16] J. Demmel and B. Kågstrom, *Accurate solutions of ill-posed problems in control theory*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 126–145.
[17] J. Demmel and B. Kågstrom, *Stably computing the Kronecker structure and reducing subspaces of singular pencils $A - \lambda B$ for uncertain data*, in Large Scale Eigenvalue Problems, J. Cullum and R. A. Willoughby, eds., North-Holland, Amsterdam, 1986.
[18] J. Demmel and B. Kågstrom, *Stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88/89 (1987), pp. 139–186.
[19] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.
[20] A. A. Dubrulle and G. H. Golub, *A multishift QR iteration without computation of the shifts*, Numer. Algorithms, 7 (1994), pp. 173–181.
[21] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4 (1984), pp. 263–264.
[22] R. Eising, *The distance between a system and the set of uncontrollable systems*, in Proceedings of the Mathematical Theory of Networks and Systems, Beer-Sheva, P. A. Fuhrmann, ed., Springer-Verlag, London, 1984, pp. 303–314.
[23] L. Elsner and C. He, *An algorithm for computing the distance to uncontrollability*, Systems Control Lett., 17 (1992), pp. 453–464.
[24] J. G. F. Francis, *The QR transformation: A unitary analogue to the LR transformation.* I, Comput. J., 4 (1961/1962), pp. 265–271.
[25] J. G. F. Francis, *The QR transformation.* II, Comput. J., 4 (1961/1962), pp. 332–345.
[26] P. Gahinet and A. J. Laub, *Algebraic Riccati equations and the distance to the nearest uncontrollable pair*, SIAM J. Control Optim., 30 (1992), pp. 765–786.
[27] M. Gao and M. Neumann, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl., 188/189 (1993), pp. 305–350.
[28] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and C. B. Moler, *Matrix Eigensystem Routines: EISPACK Guide Extension*, Springer-Verlag, New York, 1972.
[29] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
[30] M. L. J. Hautus, *Controllability and observability conditions of linear autonomous systems*, Nederl. Akad. Wetensch. Proc. Ser. A 72, 1969, pp. 443–448.
[31] B. Kågstrom, *RGSVD—an algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.
[32] V. N. Kublanovskaya, *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.
[33] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.
[34] A. Laub, *Numerical linear algebra aspects of control design computations*, IEEE Trans. Automat. Control, 30 (1985), pp. 97–108.
[35] C. C. Paige, *Properties of numerical algorithms related to computing controllability*, IEEE Trans. Automat. Control, 26 (1981), pp. 130–139.
[36] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Comput. Sci. 6, Springer-Verlag, New York, 1976.
[37] B. T. Smith, J. M. Boyle, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines: EISPACK Guide*, 2nd ed., Springer-Verlag, New York, 1970.
[38] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[39] G. W. Stewart, *Algorithm* 406 *HQR*3 *and EXCHNG: Fortran subroutines for calculating and ordering and eigenvalues of a real upper Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.
[40] R. A. van de Geijn, *Deferred shifting schemes for parallel QR methods*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 180–194.
[41] D. S. Watkins, *Fundamentals of matrix computations*, John Wiley, New York, 1991.
[42] D. S. Watkins, *Shifting strategies for the parallel QR algorithm*, SIAM J. Sci. Comput., 15 (1994), pp. 953–958.
[43] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.
[44] M. Wicks and R. DeCarlo, *Computing the Distance to an Uncontrollable System*, IEEE Trans. Automat. Control, 36 (1991), pp. 39–49.
[45] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
[46] J. H. Wilkinson, *Global convergence of tridiagonal QR algorithm with origin shifts*, Linear Algebra Appl., 1 (1968), pp. 409–420.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 974–989
© 2002 Society for Industrial and Applied Mathematics

GEOMETRIC INTEGRATION ON MANIFOLD OF SQUARE OBLIQUE ROTATION MATRICES*

N. DEL BUONO† AND L. LOPEZ†

Abstract. In recent years there has been a growing interest in the dynamics of matrix differential systems on a smooth manifold. Research effort extends to both theory and numerical methods, particularly on the manifolds of orthogonal and symplectic matrices. This paper concerns dynamical systems on the manifold $\mathcal{OB}(n)$ of square oblique rotation matrices, a constraint appearing in some minimization problems and in multivariate data analysis. Background and theoretical results on differential equations on $\mathcal{OB}(n)$ are provided. Moreover, numerical procedures preserving the structure of the solution are found among known quadratic invariant preserving methods. Numerical tests and simulations on the oblique Procrustes problem are also reported.

Key words. manifolds, square oblique rotation matrices, condition number function, Runge–Kutta methods, oblique Procrustes problems

AMS subject classifications. 65F, 65L

PII. S089547980037768X

1. Introduction.

In recent years there has been a growing interest in studying matrix differential systems whose solutions evolve on a smooth manifold such as the manifold of orthogonal or symplectic matrices (see, for instance, [3], [4], [8], [9], [19]). In this paper we consider, both theoretically and numerically, matrix differential systems on the manifold of square oblique rotation matrices

$$\mathcal{OB}(n) = \{Y \in \mathbb{R}^{n\times n} | \mathrm{diag}(Y^TY) = I_n, \text{ and } \det(Y) \neq 0\},$$

i.e., the open set of nonsingular matrices Y satisfying the constraint $\mathrm{diag}(Y^TY) = I_n$, where I_n is the $n \times n$ identity matrix. An example of a matrix differential system on $\mathcal{OB}(n)$ is that associated with the minimization problem

$$\begin{array}{ll} \min & \alpha\|AY - B\|_F + \beta\|XY^{-T} - Z\|_F \\ \text{subject to} & Y \in \mathcal{OB}(n), \end{array} \tag{1.1}$$

where A, B, X, Z are given matrices of dimension n, α and β are known nonnegative weights, and $\|\cdot\|_F$ denotes the Frobenius norm on matrices (see [17], [21]). This problem is known as the oblique Procrustes problem (hereafter ObPP(α, β)). If $\alpha = 1$ and $\beta = 0$, we obtain the so-called *classical* ObPP$(1, 0)$, which is a frequent problem in different areas of multivariate data analysis as, for example, factor analysis for common-factor extraction and multidimensional scaling techniques (see [7], [12], [18]). An advantageous feature of ObPP(1,0) is that it is equivalent to n independent minimization problems on the unit sphere $\mathcal{S}^{n-1}$ in $\mathbb{R}^n$, i.e., it can be transformed into n separate problems for each column of Y. Similarly, ObPP$(0, 1)$ is known as the oblique Procrustes rotation problem to the specified factor-pattern matrix (see [1], [18]). With α and β both different from zero, ObPP(α, β) is a generalization of

*Received by the editors September 4, 2000; accepted for publication (in revised form) by M. Chu October 11, 2001; published electronically March 27, 2002.
http://www.siam.org/journals/simax/23-4/37768.html

†Dipartimento Interuniversitario di Matematica, Via E. Orabona 4, I-70125 Bari, Italy (delbuono@dm.uniba.it, lopezl@dm.uniba.it). This research was supported in part by CNR contract 98.01013.CT01.

ObPP$(1,0)$ and ObPP$(0,1)$ that does not possess an explicit solution. Because of the inverse in the second term of (1.1), separation into n independent minimization problems is not possible. Therefore, as pointed out in [21] and [22], for solving ObPP(α,β) the use of a matrix approach is suggested.

In this paper we shall consider the general case where problems on $\mathcal{OB}(n)$ cannot be separated into a set of n independent problems on the unit sphere. We propose a flow approach to tackle the problems.

The remainder of this paper is organized as follows. In section 2, we provide some background information on differential systems on $\mathcal{OB}(n)$, emphasizing the existence of the solution and its conditioning. Note that it is possible that the solution of a flow on $\mathcal{OB}(n)$ will converge to a singular matrix. We define the conditioning of a problem on $\mathcal{OB}(n)$ and derive an upper bound on the associated condition number as a function of time. In section 3, we focus our attention on the numerical solution of problems on $\mathcal{OB}(n)$. The constraint $\mathrm{diag}(Y^TY) = I_n$ is equivalent to a set of n quadratic conservation laws on the column of Y; hence any numerical method that preserves the obliqueness must first preserve the quadratic. For this reason we look for obliqueness preserving methods among known quadratic integrators. We point out that quadratic preserving methods such as Lie group schemes in [9] and [16] do not correctly solve equations on $\mathcal{OB}(n)$ with respect to the constraint. However, Gauss–Legendre Runge–Kutta methods and projection on the manifold $\mathcal{OB}(n)$ of every explicit one-step or multistep method have good preservation properties. In the last section, we present several numerical tests together with numerical results for problems in ObPP$(1,1)$.

2. Background. Assume $Y \in \mathcal{OB}(n)$ and denote by $\mathcal{T}_Y\mathcal{OB}(n)$ the tangent space at Y. Clearly

$$\mathcal{T}_Y\mathcal{OB}(n) = \{H \in \mathbb{R}^{n\times n} | \mathrm{diag}(Y^TH) = 0\} \subset \mathbb{R}^{n\times n}.$$

Observe that the linear space

$$\mathcal{SK}(n) = \{F \in \mathbb{R}^{n\times n} | \mathrm{diag}(F) = 0\}$$

is the tangent space at the identity matrix.

Let $G : \mathbb{R}\times\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be a continuous and locally Lipschitz matrix function on the set $\mathcal{D} = (\gamma,\omega)\times\mathcal{W}$, where (γ,ω) is an open interval in $\mathbb{R}$ and $\mathcal{W}$ is a subdomain of $\mathcal{OB}(n)$, and let $(t_0, Y_0) \in \mathcal{D}$. Then the differential system

$$Y'(t) = G(t, Y(t)), \qquad Y(t_0) = Y_0 \in \mathcal{OB}(n) \tag{2.1}$$

has a unique solution $Y(t)$ defined in a neighborhood (τ^-,τ^+) of t_0. It is known that $Y(t)$, τ^-, and τ^+ depend on (t_0, Y_0) (see [23]). In the following we assume that (τ^-,τ^+) denotes the maximal interval of existence of the solution $Y(t)$.

THEOREM 2.1. *Let $Y(t)$ be a solution of the system* (2.1). *Suppose that $Y(t) \in \mathcal{OB}(n)$ for all $t \in (\tau^-,\tau^+)$. Then the matrix function $G(t,Y(t))$ may be written as*

$$G(t,Y(t)) = H(t,Y(t)) - Y(t)\mathrm{diag}[Y^T(t)H(t,Y(t))], \tag{2.2}$$

where $H : \mathbb{R}\times\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is a suitable matrix function.

Proof. To prove the relation (2.2) we essentially use a result derived in [5]. The constraint $\mathrm{diag}(Y^TY) = I$ is equivalent to n constraints $y_i^Ty_i = 1$ for $i = 1,\dots,n$ on the columns of Y. This means that, if $Y(t) \in \mathcal{OB}(n)$, then each column $G_i(t,Y(t))$

of the matrix function $G(t,Y(t))$ is necessarily tangent to the unit sphere $\mathcal{S}^{n-1}$, that is, $\langle G_i(t,Y(t)),y_i(t)\rangle = 0$. This feature may be exploited in order to transform a flow on $\mathcal{OB}(n)$ into a sequence of flows on $\mathcal{S}^{n-1}$. In particular, the ith column $y_i(t)$ of the matrix solution $Y(t)$ must satisfy the differential equation

$$y_i'(t) = H_i(t,Y(t)) - \langle H_i(t,Y(t)),y_i(t)\rangle y_i(t), \qquad i=1,\dots,n, \tag{2.3}$$

where the vector function $H_i(t,Y(t))$ is given by

$$H_i(t,Y(t)) = G_i(t,Y(t)) + \alpha_i y_i(t) \tag{2.4}$$

with $\alpha_i \in \mathbb{R}$. Since $\langle G_i(t,Y(t)),y_i(t)\rangle = 0$ and $\langle y_i(t),y_i(t)\rangle = 1$, it follows that $\alpha_i = \langle H_i(t,Y(t)),y_i(t)\rangle$ (see [5]) and this proves the theorem. □

If the ith column of the matrix $H(t,Y(t))$ depends only on the vector $y_i(t)$, i.e., $H_i(t,Y(t)) = H_i(t,y_i(t))$, then (2.3) becomes a set on n independent ODEs on $\mathcal{S}^{n-1}$. When (2.3) is not separable, one can approximate $H_i(t,Y(t))$ with a vector function depending on $y_i(t)$ only, but this approximation provides a differential problem with a larger condition number function.

Observe that since $Y \in \mathcal{OB}(n)$ is a nonsingular matrix, $G \in \mathcal{T}_Y\mathcal{OB}(n)$ if and only if $G = Y^{-T}F$ with $F \in \mathcal{SK}(n)$. Then the following characterization of matrix differential systems on the manifold of square oblique rotation matrices may be derived.

THEOREM 2.2. *Let $Y(t)$ be the solution of (2.1) on the existence interval (τ^-,τ^+). Then $Y(t)$ belongs to $\mathcal{OB}(n)$ for all $t \in (\tau^-,\tau^+)$ if and only if*

$$F(t,Y(t)) = Y^T(t)G(t,Y(t)), \qquad t \in (\tau^-,\tau^+), \tag{2.5}$$

is a continuous and locally Lipschitz matrix function mapping $\mathcal{OB}(n)$ onto $\mathcal{SK}(n)$.

Proof. Let $Y(t)$ be the solution of (2.1); then

$$\frac{d}{dt}[Y^T(t)Y(t)] = \left(\frac{d}{dt}Y^T(t)\right)Y(t) + Y^T(t)\left(\frac{d}{dt}Y(t)\right) = G^T(t,Y(t))Y(t) + Y^T(t)G(t,Y(t)),$$

and, therefore,

$$\frac{d}{dt}\mathrm{diag}(Y^T(t)Y(t)) = \mathrm{diag}[G^T(t,Y(t))Y(t) + Y^T(t)G(t,Y(t))], \qquad t \in (\tau^-,\tau^+).$$

Thus, if $Y(t) \in \mathcal{OB}(n)$, then

$$\mathrm{diag}[G^T(t,Y(t))Y(t) + Y^T(t)G(t,Y(t))] = 0, \qquad t \in (\tau^-,\tau^+).$$

The matrix $G(t,Y(t))$ must belong to $\mathcal{T}_{Y(t)}\mathcal{OB}(n)$, i.e., the matrix function $F(t,Y(t)) = Y^T(t)G(t,Y(t))$ is such that $\mathrm{diag}(F(t,Y(t))) = 0$. It follows that $F(t,Y(t)) \in \mathcal{SK}(n)$ for all $t \in (\tau^-,\tau^+)$.

Conversely, consider the differential system

$$Y'(t) = G(t,Y(t)), \qquad Y(t_0) = Y_0 \in \mathcal{OB}(n) \tag{2.6}$$

and the associated differential system

$$Y'(t) = G(t,Q(Y(t))), \qquad Y(t_0) = Y_0 \in \mathcal{OB}(n), \tag{2.7}$$

where $F(t, Y(t)) = Y^T(t)G(t, Y(t))$ is a continuous, locally Lipschitz matrix function mapping elements of $\mathcal{OB}(n)$ into $\mathcal{SK}(n)$ and $Q : \mathbb{R}^{n\times n} \to \mathcal{OB}(n)$ is a continuous projection of Y on $\mathcal{OB}(n)$. Let $Y(t)$ be the solution of (2.7); then

$$\frac{d}{dt}[Y^T(t)Y(t)] = G^T(t, Q(Y(t)))Y(t) + Y^T(t)G(t, Q(Y(t))).$$

Since $F(t, \cdot)$ maps elements of $\mathcal{OB}(n)$ into $\mathcal{SK}(n)$, the solution of (2.7) belongs to $\mathcal{OB}(n)$ for all t in the existence interval. Therefore $Q(Y(t)) = Y(t)$ and the differential system (2.7) is equivalent to (2.6). ∎

A dynamical system on $\mathcal{OB}(n)$ can therefore be written in the following form:

$$Y'(t) = Y^{-T}(t)F(t, Y(t)), \quad Y(t_0) = Y_0 \in \mathcal{OB}(n), \qquad t \in (\tau^-, \tau^+), \tag{2.8}$$

where F is a continuous and locally Lipschitz matrix function, such that

$$F : \mathbb{R} \times \mathcal{OB}(n) \to \mathcal{SK}(n). \tag{2.9}$$

Remark. We mention that an example of a dynamical system on $\mathcal{OB}(n)$ has arisen from the oblique Procrustes problem. It has been proven in [21], using projected gradient flow theory, that the solution of ObPP(α, β) can be computed as the limit point of a matrix differential system on $\mathcal{OB}(n)$ of the form (2.1) with

$$G(Y) = Y^{-T}\mathrm{off}[\beta(XY^{-T} - Z)^T XY^{-T} - \alpha Y^T A^T(AY - B)], \tag{2.10}$$

where off$(\cdot)$ is the matrix operator defined as $\mathrm{off}(A) = A - \mathrm{diag}(A)$.

Note that when the matrix solution $Y(t)$ of (2.1) evolves on $\mathcal{OB}(n)$, it is a bounded matrix function for all t in the maximal existence interval (τ^-, τ^+). Thus if the escape point τ^+ is a finite value, then $Y(t)$ tends to the *boundary* of the manifold for $t \to \tau^+$, i.e., $Y(t)$ converges to a singular matrix for $t \to \tau^+$.

The value of the escape point τ^+ depends on the matrix function $G(t, Y) = Y^{-T}F(t, Y)$. If $G(t, Y)$ is well defined at all matrices Y of $\mathbb{R}^{n\times n}$, then $\tau^+ = \infty$; otherwise τ^+ is a finite value. For instance, if $F(t, Y)$ is constant for all (t, Y), then G is not well defined at all singular matrices. In this case G will be a continuous and locally Lipschitz matrix function only in neighborhoods of nonsingular initial matrices, so that the solution $Y(t)$ exists only on finite neighborhoods of t_0 and approaches a singular matrix for $t \to \tau^+$.

Example 2.3. The differential system

$$Y' = Y^{-T}\begin{pmatrix} 0 & -\frac{\delta}{2} \\ -\frac{\delta}{2} & 0 \end{pmatrix}, \qquad Y(0) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \qquad \text{with } \delta \neq 0, \tag{2.11}$$

has a solution given by

$$Y(t) = \frac{1}{\sqrt{2}}\begin{pmatrix} \sqrt{1+\delta t} & -\sqrt{1+\delta t} \\ \sqrt{1-\delta t} & \sqrt{1-\delta t} \end{pmatrix},$$

which exists and belongs to $\mathcal{OB}(n)$ in the neighborhood $(-1/\delta, 1/\delta)$ of $t_0 = 0$. In this case the matrix function $G(Y)$ exists and is a continuous and Lipschitz matrix function only in a neighborhood of $Y(0)$.

As shown in Example 2.3 differential problems on $\mathcal{OB}(n)$, the condition number of the solution of a certain differential equation on $\mathcal{OB}(n)$ may become unbounded. For this reason, the following result may be very useful.

THEOREM 2.4. *Let $Y(t)$ be the solution of the differential system* (2.8) *and suppose that*

$$\left\| \int_{t_0}^{t} [F^T(s, Y(s)) + F(s, Y(s))] \mathrm{d}s \right\| \leq g(t), \qquad t \in [t_0, \tau^+), \tag{2.12}$$

where g is a continuous nonnegative function such that $g(t_0) = 0$ and $\|\cdot\|$ is the 2-norm on matrices. Let $\lambda_{\max}(t_0)$ and $\lambda_{\min}(t_0)$ be, respectively, the largest and smallest strictly positive eigenvalues of $Y_0^T Y_0$. Let σ be the largest value of the interval $[t_0, \tau^+)$ such that

$$\lambda_{\min}(t_0) - g(t) > 0, \qquad t \in [t_0, \sigma). \tag{2.13}$$

Then the condition number function $\mu(t)$ in the 2-norm of the solution $Y(t)$ satisfies

$$\mu(t) \leq \sqrt{\frac{\lambda_{\max}(t_0) + g(t)}{\lambda_{\min}(t_0) - g(t)}}, \qquad t \in [t_0, \sigma). \tag{2.14}$$

Proof. The condition number function $\mu(t)$ in the 2-norm of $Y(t)$ is the square root of the ratio between the largest and smallest eigenvalue of the symmetric positive definite matrix $Y^T(t)Y(t)$. This matrix function satisfies the differential equation

$$\frac{d}{dt}[Y^T(t)Y(t)] = [F^T(t, Y(t)) + F(t, Y(t))], \qquad Y^T(t_0)Y(t_0) = Y_0^T Y_0, \tag{2.15}$$

whose solution may be written as

$$Y^T(t)Y(t) = Y_0^T Y_0 + \int_{t_0}^{t} [F^T(s, Y(s)) + F(s, Y(s))] \mathrm{d}s, \qquad t \in (\tau^-, \tau^+). \tag{2.16}$$

From (2.16) and the Bauer–Fike theorem (see [11]), it follows that

$$\min_{1 \leq i \leq n} |\lambda_i(t_0) - \lambda(t)| \leq \left\| \int_{t_0}^{t} [F^T(s, Y(s)) + F(s, Y(s))] \mathrm{d}s \right\|, \qquad t \in [t_0, \tau^+), \tag{2.17}$$

where $\lambda_i(t_0)$ for $i = 1, \ldots, n$ are the eigenvalues of $Y_0^T Y_0$ and $\lambda(t)$ is a generic eigenvalue of $Y^T(t)Y(t)$. Hence, there exists $j \in 1, \ldots, n$ such that

$$|\lambda_j(t_0) - \lambda(t)| = \min_{1 \leq i \leq n} |\lambda_i(t_0) - \lambda(t)|, \qquad t \in [t_0, \tau^+),$$

and consequently

$$\lambda_{\min}(t_0) - \lambda(t) \leq \lambda_j(t_0) - \lambda(t) \leq |\lambda_j(t_0) - \lambda(t)|.$$

Therefore, from (2.12) and (2.17) we have

$$\lambda_{\min}(t_0) - \lambda(t) \leq g(t), \qquad t \in [t_0, \tau^+).$$

In particular, for $\lambda(t) = \lambda_{\min}(t)$ the smallest eigenvalue of $Y^T(t)Y(t)$, we have

$$\lambda_{\min}(t_0) \leq \lambda_{\min}(t) + g(t), \qquad t \in [t_0, \tau^+), \tag{2.18}$$

and hence

$$\lambda_{\min}(t_0) - g(t) \le \lambda_{\min}(t), \qquad t \in [t_0, \tau^+). \tag{2.19}$$

In the same manner, if $\lambda(t)$ is replaced by $\lambda_{\max}(t)$ the largest eigenvalue of $Y(t)^T Y(t)$, we obtain

$$\lambda_{\max}(t) \le \lambda_{\max}(t_0) + g(t), \qquad t \in [t_0, \tau^+). \tag{2.20}$$

Thus, if $t \in [t_0, \sigma)$, then the lower bound on $\lambda_{\min}(t)$ in (2.18) becomes strictly positive. Finally, since $\mu(t) = \sqrt{\lambda_{\max}(t)/\lambda_{\min}(t)}$, by using (2.18) and (2.20), (2.14) follows. ∎

Observe that, when $F(t, Y)$ is a skew-symmetric matrix function, Theorem 2.4 implies that the condition number function $\mu(t)$ is constant on the time interval. The differential problem (2.8) is said to be *ill-conditioned* when $\mu(t)$ is unbounded on the interval $[t_0, \tau^+)$; otherwise, it is said to be *well-conditioned*.

The value σ, given by (2.13), provides a lower bound of the escape point τ^+. The conditioning of a differential problem on $\mathcal{OB}(n)$ depends on the function $g(t)$. In particular, the problem will be ill-conditioned when $g(t)$ approaches $\lambda_{\min}(t_0)$ at some t in $[t_0, \sigma)$.

Example 2.3 has a general implication: Differential systems where F is constant are typically ill-conditioned. In fact, in this case we may choose $g(t) = (t - t_0)\|F + F^T\|$ for all $t \ge t_0$, and so $g(\sigma) = \lambda_{\min}(t_0)$ for $\sigma = (t_0 + \lambda_{\min}(t_0))/\|F + F^T\|$. Thus the escape point τ^+ may be estimated by σ. Instead, examples of well-conditioned problems are those satisfying (2.13) for all t in $[t_0, +\infty)$. These remarks can be summarized in the following result.

PROPOSITION 2.5. *If $F(t, Y)$ is a constant matrix function, then* (2.8) *will be ill-conditioned on $[t_0, \tau^+)$. Furthermore, if the differential system* (2.8) *is well-conditioned on $[t_0, +\infty)$, then $F(t, Y)$ is a skew-symmetric or not constant matrix function.*

3. Numerical methods. For the sake of simplicity, we concentrate our attention on autonomous differential systems, even though the main results derived in this paper may also be applied to the nonautonomous cases. Let $h > 0$ be the time step. Consider a partition of the time interval given by $t_{k+1} = t_k + h$, and denote by Y_k an approximation of $Y(t_k)$ at t_k, for $k \ge 0$.

As pointed out previously, the constraint $\operatorname{diag}(Y^T Y) = I_n$ is equivalent to a set of n quadratic conservation laws on the columns of Y. We thus look for obliqueness preserving schemes among integrators on $\mathcal{S}^{n-1}$. A first negative result is that the quadratic preserving methods based on Lie group theory (see [9] and [16]) do not preserve the constraint $\operatorname{diag}(Y^T Y) = I_n$. Instead, positive results may be obtained by using Gauss–Legendre Runge–Kutta methods and projection techniques. We also show that Gauss–Legendre Runge–Kutta schemes preserve the condition number function of the solution, i.e., they satisfy a relation like (2.16) at each time step t_k. However, they require very small time steps of integration when $Y(t)$ converges to a singularity or $F(t, Y)$ depends on the inverse of $Y(t)$.

3.1. Runge–Kutta methods. Consider the v-stage Runge–Kutta method defined by the Butcher array

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b} \end{array}, \tag{3.1}$$

where $\mathbf{c}^T = (c_1, \ldots, c_v)$, $\mathbf{A} = (a_{ij})$, $\mathbf{b} = (b_1, \ldots, b_v)$ (see [14]). Applying (3.1) to (2.8) we get

$$Y_{k+1} = Y_k + h \sum_{i=1}^{v} b_i Y_{ki}^{-T} F(Y_{ki}), \tag{3.2}$$

with

$$Y_{ki} = Y_k + h \sum_{j=1}^{v} a_{ij} Y_{kj}^{-T} F(Y_{kj}), \qquad i = 1, \ldots, v, \qquad k \geq 0. \tag{3.3}$$

DEFINITION 3.1. *A Runge–Kutta method* (3.1) *applied to* (2.8) *is said to be* diagonal preserving *if and only if it provides a sequence* Y_k *for* $k \geq 0$ *such that*

$$\operatorname{diag}(Y_{k+1}^T Y_{k+1}) = \operatorname{diag}(Y_k^T Y_k), \qquad k \geq 0. \tag{3.4}$$

Furthermore, it is said to be condition number function preserving *if and only if*

$$Y_{k+1}^T Y_{k+1} = Y_k^T Y_k + h \sum_{i=1}^{v} b_i [F(Y_{ki}) + F^T(Y_{ki})], \qquad k \geq 0.$$

Finally, it is said to be obliqueness preserving *when it is both diagonal and condition number function preserving.*

Concerning the diagonal and conditioning preserving feature of Runge–Kutta methods, the following result may be trivially derived.

LEMMA 3.2. *For differential system* (2.8) *where* (2.9) *is replaced by*

$$F : \mathbb{R}^{n \times n} \to \mathcal{SK}(n), \tag{3.5}$$

the Runge–Kutta scheme (3.2)–(3.3) *with coefficient matrix*

$$\mathbf{M} = (b_i b_j - b_i a_{ij} - b_j a_{ji}) \tag{3.6}$$

identically zero preserves both the diagonal and the condition number function along the solution.

Proof. Consider the numerical solution Y_{k+1} given by (3.2)–(3.3). By using techniques similar to those used in quadratic or symplectic methods (see [2], [6], [8], [10], [20]), we obtain

$$\begin{aligned} Y_{k+1}^T Y_{k+1} = \; & Y_k^T Y_k + h \textstyle\sum_{i=1}^{v} b_i [F^T(Y_{ki}) + F(Y_{ki})] \\ & + h^2 \textstyle\sum_{i=1}^{v} \sum_{j=1}^{v} (b_i b_j - b_i a_{ij} - b_j a_{ji}) F^T(Y_{ki}) Y_{ki}^{-1} Y_{kj}^{-T} F(Y_{kj}). \end{aligned} \tag{3.7}$$

If the coefficient matrix $\mathbf{M}$ is equal to zero, then the Runge–Kutta scheme preserves the condition number function. Furthermore, since F maps all matrices of $\mathbb{R}^{n \times n}$ into $\mathcal{SK}(n)$, then $\operatorname{diag}[F^T(Y_{ki}) + F(Y_{ki})] = 0$, and hence

$$\operatorname{diag}(Y_{k+1}^T Y_{k+1}) = \operatorname{diag}(Y_k^T Y_k), \qquad k \geq 0;$$

that is, the Runge–Kutta method is diagonal preserving. ∎

From Lemma 3.2 it follows that no explicit Runge–Kutta method is diagonal preserving, while all v-stage Gauss–Legendre Runge–Kutta methods (denoted by GLv) possess this property (see [14]).

Remark. Note that the intermediate values Y_{ki} do not lie on $\mathcal{OB}(n)$. Thus, if in Lemma 3.2 the condition on the function F is replaced with (2.9), it follows that $F(Y_{ki})$ does not belong to $\mathcal{SK}(n)$ and hence GLv are not diagonal preserving. In this case we can apply the following projected version of GLv (denoted by PGLv):

$$Y_{k+1} = Y_k + h\sum_{i=1}^{v} b_i Y_{ki}^{-T}\mathcal{P}[F(Y_{ki})], \tag{3.8}$$

$$Y_{ki} = Y_k + h\sum_{j=1}^{v} a_{ij} Y_{kj}^{-T}\mathcal{P}[F(Y_{kj})], \qquad i = 1,\dots,v, \tag{3.9}$$

where the projection of F onto $\mathcal{SK}(n)$ is given by

$$\mathcal{P}(F) = F - \operatorname{diag}(F).$$

Obviously the PGLv method is diagonal preserving and it has the same order of accuracy of GLv. In fact, PGLv is equivalent to GLv applied to the differential problem

$$Z' = Z^{-T}\mathcal{P}[F(Z)], \qquad Z(t_0) = Y_0 \in \mathcal{OB}(n),$$

which is equivalent to (2.8). A cheaper procedure may be derived by substituting (3.9) into (3.3). This method will be denoted by WPGLv.

It is known that a cheap way to solve the nonlinear system associated with (3.3) is the functional fixed-point iteration scheme, yielding

$$Y_{kl}^{(m+1)} = Y_k + h\sum_{j=1}^{v} a_{lj}(Y_{kj}^{(m)})^{-T}F(Y_{kj}^{(m)}), \qquad l = 1,\dots,v, \tag{3.10}$$

with initial guess given by $Y_{kl}^{(0)} = Y_k$ for $l = 1,\dots,v$ and stopping criteria

$$\left\| Y_{kl}^{(m+1)} - Y_k - h\sum_{j=1}^{v} a_{lj}[Y_{kj}^{(m+1)}]^{-T}F(Y_{kj}^{(m+1)}) \right\| < tol, \qquad l = 1,\dots,v.$$

We illustrate the scheme by considering the implicit midpoint rule (GL1) where at every step k the nonlinear system to be solved is

$$H(Z) = Z - Y_k - \frac{h}{2}Z^{-T}F(Z) = 0, \tag{3.11}$$

while the functional iteration may be written as

$$Z^{(m+1)} = Y_k + \frac{h}{2}[Z^{(m)}]^{-T}F(Z^{(m)}) \qquad \text{for } m \geq 0, \tag{3.12}$$

with $Z^{(0)} = Y_k$.

THEOREM 3.3. *Let L_1 be the Lipschitz constant of F with respect to Y on a domain D and assume that there exists a constant L_2 such that*

$$\textit{for all } Y \in D: \ \|F(Y)\| \leq L_2. \tag{3.13}$$

Let Z be the solution of the nonlinear system (3.11) *and suppose that the functional iteration* (3.12) *provides approximations $Z^{(m)}$ of Z such that*

$$\|[Z^{(m)}]^{-1}\| \leq L_3\|Z^{-1}\|, \qquad m \geq 0, \tag{3.14}$$

where L_3 is a positive constant independent of m. Then the functional iteration (3.12) *converges if the time step h is such that*

$$h < \frac{2}{L_3\|Z^{-1}\|\{L_1 + L_2\|Z^{-1}\|\}}. \tag{3.15}$$

Proof. From (3.11) and (3.12) we have

$$Z^{(m+1)} - Z = \frac{h}{2}\{[Z^{(m)}]^{-T}F(Z^{(m)}) - Z^{-T}F(Z)\}$$

$$= \frac{h}{2}\{[Z^{(m)}]^{-T}[F(Z^{(m)}) - F(Z)] + ([Z^{(m)}]^{-T} - Z^{-T})F(Z)\}$$

$$= \frac{h}{2}\{[Z^{(m)}]^{-T}[F(Z^{(m)}) - F(Z)] + [Z^{(m)}]^{-T}(Z^T - [Z^{(m)}]^T)Z^{-T}F(Z)\}.$$

Since the 2-norm of a matrix is equal to the 2-norm of its transpose, it follows that

$$\begin{aligned} \|Z^{(m+1)} - Z\| &\leq \tfrac{h}{2}\|Z^{(m)} - Z\|\|[Z^{(m)}]^{-1}\|\{L_1 + L_2\|Z^{-1}\|\} \\ &\leq \tfrac{h}{2}\|Z^{(m)} - Z\|L_3\|Z^{-1}\|\{L_1 + L_2\|Z^{-1}\|\}, \end{aligned}$$

and if (3.15) is satisfied, the convergence follows. ∎

From (3.15), it follows that the functional iteration needs very small time steps of integration near a singularity of the solution Y. Furthermore, even if the matrix solution Y does not approach a singularity, but if $F(Y)$ depends on the inverse of the matrix solution Y, then the constants L_1 and L_2 may grow with $\|Y^{-1}\|$. This means that h may be very small also for problems with $\|Y^{-1}\|$ of moderate size. For instance, in the case of ObPP$(1,1)$, the quotient appearing in (3.15) is $O(\|Z^{-1}\|^4)$.

Remark. A step-size selection strategy may be considered where the value of h is reduced until the functional iteration converges, that is, until the new value h_{new} satisfies (3.15). This reduction may be stopped when h_{new} becomes smaller than a prefixed lower bound.

A more accurate reformulation for the fixed-point iterations in the implicit Runge–Kutta methods may be used in order to reduce the influence of round-off errors and improve convergence (see [15] for details).

3.2. Projected methods. In the same spirit as the work in [8] where projected methods on the orthogonal manifold have been proposed, we can consider numerical procedures based on projecting on $\mathcal{OB}(n)$ of the numerical solution of (2.8) obtained by any explicit Runge–Kutta or multistep method onto $\mathcal{OB}(n)$. A projection Y of a matrix Q on $\mathcal{OB}(n)$ is given by the closest oblique rotation in the least square sense, i.e.,

$$Y = Q\mathrm{diag}(Q^TQ)^{-\frac{1}{2}}, \tag{3.16}$$

with

$$\|Q - Y\| \leq \|Y\|\|I - \mathrm{diag}(Q^T Q)^{\frac{1}{2}}\|.$$

This projection may be used in conjunction with one-step or multistep explicit schemes in order to obtain a semi-implicit procedure where no iteration is required. In particular, given $Y_k \in \mathcal{OB}(n)$, first we compute $\tilde{Y}_{k+1}$ by the explicit Runge–Kutta method

$$\begin{aligned} \tilde{Y}_{k+1} &= Y_k + h\sum_{i=1}^{v} b_i Y_{ki}^{-T} F(Y_{ki}), \\ Y_{ki} &= Y_k + h\sum_{j=1}^{i-1} a_{ij} Y_{kj}^{-T} F(Y_{kj}), \qquad i = 1, \ldots, v. \end{aligned} \tag{3.17}$$

Then we project $\tilde{Y}_{k+1}$ on $\mathcal{OB}(n)$; that is, we compute

$$Y_{k+1} = \tilde{Y}_{k+1}\mathrm{diag}(\tilde{Y}_{k+1}^T \tilde{Y}_{k+1})^{-\frac{1}{2}}. \tag{3.18}$$

If the basic Runge–Kutta method is of order p, we have

$$\mathrm{diag}(\tilde{Y}_{k+1}^T \tilde{Y}_{k+1}) = I + D(h^p), \tag{3.19}$$

where $D(h^p)$ is a diagonal matrix with elements that are $O(h^p)$. Therefore, we have

$$\tilde{Y}_{k+1} - Y_{k+1} = Y_{k+1}[I - \mathrm{diag}(\tilde{Y}_{k+1}^T \tilde{Y}_{k+1})^{\frac{1}{2}}] = Y_{k+1}[I - (I + D(h^p))^{\frac{1}{2}}].$$

Hence, $\|\tilde{Y}_{k+1} - Y_{k+1}\| = O(h^p)$ implies that the projected method is of the same order as that of the basic Runge–Kutta scheme. Furthermore, from (3.18) it follows that

$$Y_{k+1}^T Y_{k+1} = [\mathrm{diag}(\tilde{Y}_{k+1}^T \tilde{Y}_{k+1})^{-\frac{1}{2}}]^T \tilde{Y}_{k+1}^T \tilde{Y}_{k+1} \mathrm{diag}(\tilde{Y}_{k+1}^T \tilde{Y}_{k+1})^{-\frac{1}{2}}.$$

By (3.19), we have

$$Y_{k+1}^T Y_{k+1} = \tilde{Y}_{k+1}^T \tilde{Y}_{k+1} + O(h^p),$$

with

$$\begin{aligned} \tilde{Y}_{k+1}^T \tilde{Y}_{k+1} &= Y_k^T Y_k + h\sum_{i=1}^{v} b_i [F^T(Y_{ki}) + F(Y_{ki})] \\ &\quad + h^2 \sum_{i=1}^{v}\sum_{j=1}^{v} (b_i b_j - b_i a_{ij} - b_j a_{ji}) F^T(Y_{ki}) Y_{ki}^{-1} Y_{kj}^{-T} F(Y_{kj}). \end{aligned}$$

Since the matrix $\mathbf{M}$ of an explicit Runge–Kutta method cannot be the zero matrix, the last term in the previous equality does not vanish. Therefore a projected Runge–Kutta method is *not* condition number function preserving.

Observe that in a step-size selection strategy to control the local truncation error, the new time step h_{new} must satisfy the relation

$$h_{new} \leq \left(\frac{\theta\epsilon}{h\|r(t_k, Y_k)\|}\right)^{\frac{1}{p+1}}, \tag{3.20}$$

where θ is a safety factor, ϵ is the local truncation error bound, and

$$r(t_k, Y_k) = \sum_{i=1}^{\hat{v}} (b_i - \hat{b}_i) Y_{ki}^{-T} F(Y_{ki}),$$

where the $\hat{b}_i$ denote the coefficients of a Runge–Kutta method of order $p+1$. From (3.20) it follows that very small time steps will be required for solving differential problems which are ill-conditioned or where $F(Y)$ depends on Y^{-1}.

4. Numerical tests. All the numerical tests have been obtained by Matlab codes implemented on a scalar computer Alpha 200 5/433 with 512 Mb RAM. We compare obliqueness preserving methods on different problems. Comparisons have been performed in terms of accuracy (measured by $\|Y(t_k) - Y_k\|_\infty$ with $\|\cdot\|_\infty$ the infinity norm on matrices), deviation from the manifold $\mathcal{OB}(n)$ (measured by $\Omega(Y_k) = \|I - \mathrm{diag}(Y_k^T Y_k)\|_F$), and CPU time. The theoretical solution $Y(t_k)$, if unknown, has been estimated applying the numerical method with a half step-size. We denote by PRKv and PABv the projected methods of section 3.2 based on explicit Runge–Kutta and Adams–Bashforth methods of order v, respectively. The starting approximations for PABv have been obtained by a Runge–Kutta method of the same order. GLv have been implemented solving the nonlinear system (3.3) by functional iteration with tolerance $tol = 10^{-15}$.

Example 4.1. First, we consider differential problem (2.8) with constant matrix function

$$F = \begin{pmatrix} 0 & 2/3 & 1 \\ 1 & 0 & 8/5 \\ -3 & 5/4 & 0 \end{pmatrix}$$

and initial condition Y_0 given by the identity matrix. Figure 4.1 (a) plots the behavior of the function $g(t) = \|F^T + F\|t$ in the interval $[0, \sigma)$, where $\sigma \approx 0.2280$ is the intersection point of $g(t)$ and $\lambda_{\min}(0)$.

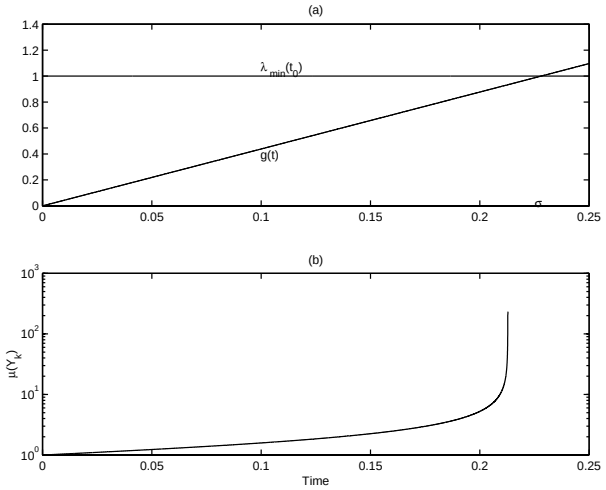


FIG. 4.1. *Conditioning of Example* 4.1.

Figure 4.1 (b) plots the behavior of the condition number function obtained by GL1 applied on $[0, \sigma)$ with the variable step-size strategy described at the end of

TABLE 4.1

Method	CPU time	$\Omega(Y_k)$	Global error	$\mu(t)$
GL1	0.5000	1.1957e-15	0.0016	48.4875
PRK2	0.1167	2.2204e-16	0.0031	50.4054
PAB2	0.0667	2.2402e-16	0.0080	52.4978
GL2	0.5500	1.4603e-15	5.4116e-08	49.1144
PRK4	0.2000	2.2204e-16	1.0151e-07	49.1144
PAB4	0.1060	2.4825e-16	4.1564e-06	49.1161

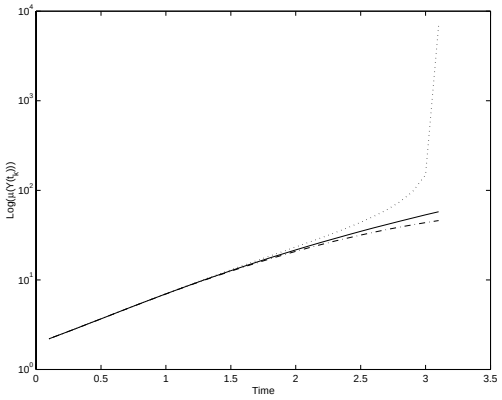


FIG. 4.2. *Condition number function of second order methods.*

subsection 3.1, where the value of h is the largest value for which the functional iteration converges. It can be observed that the condition number function increases near the singularity of the solution.

Example 4.2. We consider the nonautonomous differential system $Y'(t) = Y^{-T}(t)F(t)$ for $t \in [0.1, 3]$, where the matrix function $F(t)$ is given by

$$F(t) = \begin{pmatrix} 0 & \dfrac{t^2 - t\sqrt{t^2+3}}{(t^2+4)^{3/2}\sqrt{t^4+4t^2+3}} \\ \dfrac{t - \sqrt{t^2+3}}{(1+t^2)^{3/2}\sqrt{t^2+4}} & 0 \end{pmatrix}$$

and the theoretical solution is the matrix function

$$Y(t) = \begin{pmatrix} (t^2+1)^{-1/2} & (t^2+4)^{-1/2} \\ t(t^2+1)^{-1/2} & (t^2+3)^{1/2}(t^2+4)^{-1/2} \end{pmatrix}.$$

Table 4.1 reports the performance of the methods at $t = 3$ for $h = 0.25$. The GLv methods seem to be more expensive but more accurate than the projected procedures.

Figure 4.2 plots the logarithm of the condition number function of the theoretical solution (solid line) and of the numerical solution given by GL1 (dash-dotted line) and PRK2 (dotted line) with step-size $h = 0.1$. The GL1 method reproduces the exact behavior of the condition number function while PRK2 needs smaller step-sizes h to correctly integrate the problem.

Example 4.3. We consider the system (2.8), where $F(Y) = A + (I - \mathrm{diag}(Y^T Y))$

maps only matrices of $\mathcal{OB}(n)$ into $\mathcal{SK}(n)$, with A and Y_0 given by

$$A = \begin{pmatrix} 0 & -2.28 & -0.74 \\ 2.1 & 0 & 1.3 \\ 0.8 & -1.5 & 0 \end{pmatrix}, \qquad Y_0 = \begin{pmatrix} 0.9487 & 0.5392 & 0.7071 \\ 0 & 0.5392 & 0 \\ 0.3162 & 0.6470 & 0.7071 \end{pmatrix}.$$

Table 4.2 summarizes the performance at $t = 1$ of the obliqueness preserving methods integrating on $[0, 1]$ with $h = 0.000625$. The GLv methods destroy the diagonal structure of the matrix $Y_k^T Y_k$ because the matrix function F satisfies only (2.9), while WPGLv and PGLv preserve both the diagonal and the condition number function better than GLv.

TABLE 4.2
Performance of the methods at $t = 1$.

Method	CPU time	$\Omega(Y_k)$	Global error	$\mu(t)$
GL1	4.3500	8.9457e-06	5.4614e-05	8.1319
PGL1	4.9833	6.6986e-15	2.7383e-06	8.1314
WPGL1	4.7333	2.5762e-09	2.7508e-06	8.1314
GL2	7.7167	9.8231e-13	7.2228e-12	8.1314
PGL2	8.8333	3.1480e-15	3.9928e-13	8.1314
WPGL2	8.5000	1.4358e-14	3.2199e-13	8.1314

4.1. Numerical tests for ObPP(1, 1). We now approximate the numerical solution of ObPP$(1, 1)$ with problem data sets generated randomly. In particular, we generate random matrices A, X, and Y_{in}, where Y_{in} is a projection on $\mathcal{OB}(n)$ of a random matrix Q. We then define $B = AY_{in}$ and $Z = XY_{in}^{-T}$ so that the underlying problem has a global solution at Y_{in}. The initial condition Y_0 is a projection on $\mathcal{OB}(n)$ of a perturbation of Y_{in} by a random matrix generated by Matlab function `rand`, that is, $Q = Y_{in} + \texttt{rand}(n)$. In our numerical simulations we intend to observe how frequently the numerical methods reconstruct the matrix Y_{in} either with different data sets A, Y_0, Y_{in} or with different initial values Y_0.

The tables show the number of cases in which Y_{in} is reconstructed (called reconstructions); the number of cases in which the objective function is minimized but the numerical solution differs from Y_{in} (called deviations); the number of cases in which it is not minimized (called failures), and, finally, the cases when the functional iteration does not converge (called divergences). The solution Y_{in} is considered to be faithfully reconstructed if the local error $\|Y_k - Y_{in}\|/\|Y_k\|$ is less than 10^{-3} for second order methods and 10^{-6} for fourth order methods. The time step used is $h = 0.001$. We observe that when the functional iteration diverges a smaller step-size should be employed.

Table 4.3 reports the results of 100 simulations with different data, while Table 4.4 gives the results of 100 solutions of the problem with only different starting matrix Y_0.

It seems that GLv reach the global solution of ObPP$(1, 1)$ with different data a greater number of times than the projected procedures.

We analyze in more details the problem with the following data:

$$A = \begin{pmatrix} 0.9688 & 0.7553 & 0.2512 \\ 0.3557 & 0.8948 & 0.9327 \\ 0.0490 & 0.2861 & 0.1310 \end{pmatrix}, \qquad X = \begin{pmatrix} 0.1171 & 0.8234 & 0.9492 \\ 0.7699 & 0.0466 & 0.2888 \\ 0.3751 & 0.5979 & 0.8888 \end{pmatrix}$$

TABLE 4.3

Method	Reconstructions	Deviations	Failures	Divergences
GL1	66	13	12	9
PRK2	67	23	10	-
PAB2	65	21	14	-
GL2	66	13	12	9
PRK4	67	23	10	-
PAB4	65	21	14	-

TABLE 4.4
Solutions of 100 *different ObPP with* $\alpha = 1$ *and* $\beta = 1$.

Method	Reconstructions	Deviations	Failures	Divergences
GL1	63	14	13	10
PRK2	55	23	22	-
PAB2	57	23	20	-
GL2	63	14	13	10
PRK4	55	23	22	-
PAB4	57	23	20	-

and

$$Y_{in} = \begin{pmatrix} 0.6498 & 0.4124 & 0.7964 \\ 0.4848 & 0.8969 & 0.5259 \\ 0.5855 & 0.1598 & 0.2988 \end{pmatrix}, \qquad Y_0 = \begin{pmatrix} 0.5259 & 0.3992 & 0.7934 \\ 0.6942 & 0.9006 & 0.3771 \\ 0.4915 & 0.1719 & 0.4778 \end{pmatrix}.$$

Figures 4.3 (a), (b), and (c) show, respectively, the semilog plot of the deviation from the diagonal structure, the deviation from the expected matrix Y_{in}, and the value of the objective function in (1.1) for the numerical solution given by GL1 with $h = 0.001$.

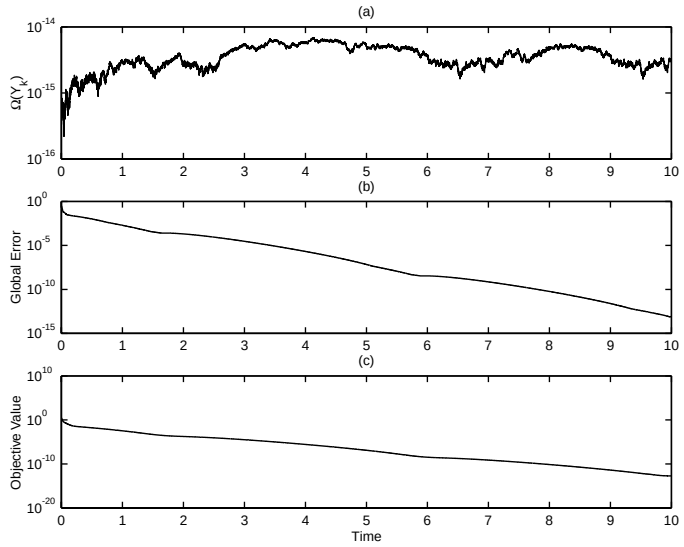


FIG. 4.3. *Performance of GL*1 *method.*

Tables 4.5 and 4.6 summarize the results obtained integrating 50 ObPP$(1, 1)$ problems by means of the variable step-size version of GL1 (denoted by GL1(vs)) and a

TABLE 4.5

Method	Reconstructions	Deviations	Failures
GL1(vs)	28	15	7
PODE23	15	26	9

TABLE 4.6

Method	Reconstructions	Deviations	Failures
GL1(vs)	39	10	1
PODE23	37	12	1

projected version of the Matlab function `ode23` (denoted by PODE23). In particular, Table 4.5 reports the results of 50 numerical simulations of different ObPP$(1,1)$ problems, while Table 4.6 shows the results obtained solving the same problem starting from 50 different initial matrix Y_0. Table 4.5 seems to indicate that GL1(vs) gives better results than the variable step-size PODE23 method.

5. Conclusions. Differential systems on the manifold $\mathcal{OB}(n)$ arise in several important applications. In this paper we have provided some theoretical results on these differential systems and studied the conditioning of the solution. We have also considered different numerical methods for the integration of problems on $\mathcal{OB}(n)$. With the expectation that the integrator should preserve both the oblique structure and the conditioning of the theoretical solution, we have found that Gauss–Legendre Runge–Kutta schemes preserve both these properties. Numerical tests, in particular for the solution of the oblique Procrustes problem, have highlighted the necessity of using very small integration steps.

Acknowledgments. The authors wish to thank the anonymous referees for their many helpful suggestions.

REFERENCES

[1] M.W. BROWNE, *Oblique rotation to a partially specified target*, British J. Math. Statist. Psych., 25 (1972), pp. 207–212.
[2] M.P. CALVO, A. ISERLES, AND A. ZANNA, *Numerical solution of isospectral flows*, Math. Comp., 66 (1997), pp. 1461–1486.
[3] M.T. CHU, *A list of matrix flows with applications*, in Hamiltonian and Gradient Flows, Algorithms and Control, Fields Inst. Commun. 3, AMS, Providence, RI, 1994, pp. 87–97.
[4] M.T. CHU, *Scaled Toda-like flows*, Linear Algebra Appl., 215 (1995), pp. 261–273.
[5] M.T. CHU, *Curves on S^{n-1} that lead to eigenvalues or their means of a matrix*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 425–432.
[6] G. COOPER, *Stability of Runge–Kutta methods for trajectory problems*, IMA J. Numer. Anal., 7 (1987), pp. 1–13.
[7] T.F. COX AND M.A.A. COX, *Multidimensional Scaling*, Chapman & Hall, London, 1995.
[8] L. DIECI, D. RUSSELL, AND E. VAN VLECK, *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM J. Numer. Anal., 31 (1994), pp. 261–281.
[9] F. DIELE, L. LOPEZ, AND R. PELUSO, *The Cayley transform in the numerical solution of unitary differential systems*, Adv. Comput. Math., 8 (1998), pp. 317–334.
[10] T. EIROLA AND J.M. SANZ-SERNA, *Conservation of integrals and symplectic structure in the integration of differential equations by multistep methods*, Numer. Math., 61 (1992), pp. 281–290.
[11] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1983.
[12] J.C. GOWER, *Multivariate analysis: Ordination, multidimensional scaling and allied topics*, in Handbook of Applicable Mathematics, Vol. IV: Statistics, Part B, E. Lloyd, ed., John Wiley & Sons, New York, 1984, pp. 727–781.

[13] G.T. Gruvaeus, *A general approach to Procrustes pattern rotation*, Psychometrika, 35 (1970), pp. 493–505.
[14] E. Hairer, S.P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations*, Vol. I: *Nonstiff Problems*, 2nd ed., Springer-Verlag, Berlin, 1991.
[15] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations*, Vol. II, Springer-Verlag, Berlin, 1991.
[16] A. Iserles, H. Munte-Kaas, S.P. Norsett, and A. Zanna, *Lie group methods*, in Acta Numerica 9, Cambridge University Press, Cambridge, UK, 2000, pp. 215–365.
[17] H.A.L. Kiers, *Joint orthomax rotation of the core and component matrices resulting from a three-mode factor analysis*, J. Classification, 15 (1998), pp. 245–263.
[18] S.A. Mulaik, *The Foundations of Factor Analysis*, McGraw-Hill, New York, 1972.
[19] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems*, Chapman & Hall, London, 1994.
[20] A.M. Stuart and A.R. Humphries, *Dynamical Systems and Numerical Analysis*, Cambridge University Press, Cambridge, UK, 1996.
[21] N.T. Trendafilov, *A continuous-time approach to the oblique Procrustes problem*, Behaviormetrika, 26 (1999), pp. 167–181.
[22] N.T. Trendafilov and R.A. Lippert, *The multimode Procrustes problem*, Linear Algebra Appl., to appear.
[23] H.K. Wilson, *Ordinary Differential Equations*, Addison-Wesley, London, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 990–997
© 2002 Society for Industrial and Applied Mathematics

INVERSE FORMS OF HADAMARD INEQUALITY*

LENG GANGSONG† AND ZHOU GUOBIAO‡

Abstract. In this paper we establish the inverse inequalities of the Hadamard inequality and the Szasz inequality. To prove these results, we give two sharpenings of the Hadamard inequality and the Szasz inequality.

Key words. parallelotope, inverse forms, canonical volume, inequality

AMS subject classification. 52A40

PII. S0895479801387279

1. Introduction.

Let $A = (a_{ij})_{n\times n}$ be an $n \times n$ positive Hermitian matrix. The classical Hadamard inequality states that

$$\det A \le \prod_{i=1}^{n} a_{ii}. \tag{1.1}$$

It is well known that the geometric form of (1.1) is the following: Let $\{x_1, x_2, \ldots, x_n\}$ be a linearly independent set of vectors from $\mathbb{R}^n, V(P_{[x_1,x_2,\ldots,x_n]})$ the volume of the n-parallelotope $P_{[x_1,x_2,\ldots,x_n]}$ which has $\{x_1, x_2, \ldots, x_n\}$ as n edge vectors; then

$$V(P_{[x_1,x_2,\ldots,x_n]}) \le \prod_{i=1}^{n} \|x_i\|. \tag{1.2}$$

Szasz has generalized the Hadamard inequality (1.2). One of the main results of Szasz is the following (see [1, 6]): Let $P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$ be the facet of $P_{[x_1,x_2,\ldots,x_n]}$; then

$$V^{n-1}(P_{[x_1,x_2,\ldots,x_n]}) \le \prod_{i=1}^{n} V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}). \tag{1.3}$$

The equalities in (1.2) and (1.3) occur if and only if $\{x_1, x_2, \ldots, x_n\}$ is a set of orthogonal nonzero vectors in $\mathbb{R}^n$.

The other generalizations of the Hadamard inequality to the block matrices and to other types of matrices are obtained by Fisher, Johnson and Markham, Veljan, and others (see [1, 5, 6, 7, 8, 13, 15, 16]). The estimates for the ratio $\det A/\prod_{i=1}^{n} a_{ii}$ are investigated by Johnson and Newman, Dixon, Reznikov, and others (see [3, 4, 9, 11, 12]). Some eigenvalues estimates of Wolkowitz and Styan could also be interpreted as estimates of $\det A$ [14].

The object of this paper is to investigate the inverse forms of the Hadamard inequality (1.2) and Szasz inequality (1.3).

*Received by the editors April 3, 2001; accepted for publication (in revised form) by R. Bhatia July 17, 2001; published electronically March 27, 2002.
http://www.siam.org/journals/simax/23-4/38727.html

†Department of Mathematics, Shanghai University, Shanghai, 200436, People's Republic of China (lenggangsong@chinaren.com).

‡Department of Applied Mathematics, Shanghai Jiao Tong University, Shanghai, 200030, People's Republic of China (guobiaoc@online.sh.cn).

Denote by $P_{[x_1,x_2,\ldots,x_m]}$ the m-parallelotope which has m linearly independent vectors $x_1, x_2, \ldots, x_m$ as m edge vectors. Let $P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$ be the facet of $P_{[x_1,x_2,\ldots,x_n]}$ which lies in a hyperplane π_i. Let y_i be the orthogonal component of x_i with respect to π_i. Then we call y_i the altitude vector on $P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_m]}$. For a given m-parallelotope, it is easy to see that there exist m linearly independent altitude vectors. Conversely, we have the following lemma.

LEMMA 1.1. *Let $\{x_1, x_2, \ldots, x_m\}$ be a given set of linearly independent vectors from R^n. Then there exists an m-parallelotope $P^*_{[x_1,x_2,\ldots,x_m]}$ which has $x_1, x_2, \ldots, x_m$ as m altitude vectors.*

Proof. Since $x_1, x_2, \ldots, x_m$ are linearly independent, there are m linear functionals $f_1, f_2, \ldots, f_m$ such that

$$f_j(x_i) = \delta_{ij}\|x_i\|^2, \qquad i, j = 1, 2, \ldots, m,$$

where δ_{ij} is the Kronecker delta symbol.

According to Riesz's representation theorem for the linear functional, there are vectors $v_1, v_2, \ldots, v_m$ such that

$$\langle x_i, v_j\rangle = \delta_{ij}\|x_i\|^2,$$

where $\langle , \rangle$ denote the ordinary inner product of $\mathbb{R}^n$.

Suppose that

$$\sum_{j=1}^{m} a_j v_j = 0, \qquad a_j \in \mathbb{R}.$$

Then

$$0 = \left\langle x_i, \sum_{j=1}^{m} a_j v_j \right\rangle = a_i\|x_i\|^2, \qquad i = 1, 2, \ldots, m.$$

Hence $a_i = 0, i = 1, 2, \ldots, m$. It follows that $v_1, v_2, \ldots, v_m$ are linearly independent.

Let $Q_{[v_1,v_2,\ldots,v_m]}$ denote the m-parallelotope which has $v_1, v_2, \ldots, v_m$ as m edge vectors, and let $Q_{[v_1,\ldots,v_{i-1},v_{i+1}\ldots,v_m]}$ denote the facet of $Q_{[v_1,v_2,\ldots,v_m]}$. Since $x_i \perp v_j (j \neq i)$, it follows that

$$x_i \perp Q_{[v_1,\ldots,v_{i-1},v_{i+1},\ldots,v_m]}.$$

Therefore, $Q_{[v_1,v_2,\ldots,v_m]}$ is an m-parallelotope which has $x_1, x_2, \ldots, x_m$ as altitude vectors; this is $P^*_{[x_1,x_2,\ldots,x_m]}$, as desired. ∎

Our main results are the following two theorems.

THEOREM 1.2. *Let $\{x_1, x_2, \ldots, x_n\}$ be a set of linearly independent vectors from $\mathbb{R}^n$ and $P^*_{[x_1,x_2,\ldots,x_n]}$ be the n-parallelotope, which has $x_1, x_2, \ldots, x_n$ as n altitude vectors. Further, let $P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$ be the $(n-1)$-parallelotope, which has $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ as $(n-1)$ altitude vectors. Then*

$$V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]}) \geq \prod_{i=1}^{n} V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}), \tag{1.4}$$

$$V(P^*_{[x_1,x_2,\ldots,x_n]}) \geq \prod_{i=1}^{n} \|x_i\|. \tag{1.5}$$

The equalities in (1.4) *and* (1.5) *occur if and only if* $\{x_1, x_2, \ldots, x_n\}$ *is a set of orthogonal nonzero vectors in* $\mathbb{R}^n$.

In fact, we will establish two inequalities more general than (1.4) and (1.5), as follows.

THEOREM 1.3. *Let* $P^*_{[x_1,x_2,\ldots,x_n]}$ *and* $P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$ *be as in Theorem* 1.2, *let* $\theta_{ij}(1 \le i < j \le n)$ *denote the dihedral angle between* π_i *and* π_j, *and let* $\alpha_{ij}(1 \le i < j \le n)$ *denote the angle between* x_i *and* x_j. *Then*

$$\frac{V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]})}{\prod_{i=1}^n V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})} \ge \frac{1}{\prod_{1\le i<j\le n}\sin\theta_{ij}}, \tag{1.6}$$

$$\frac{V(P^*_{[x_1,x_2,\ldots,x_n]})}{\prod_{i=1}^n \|x_i\|} \ge \frac{1}{\prod_{1\le i<j\le n}\sin\alpha_{ij}}. \tag{1.7}$$

The equalities in (1.6) *and* (1.7) *occur if and only if* $\{x_1, x_2, \ldots, x_n\}$ *is a set of orthogonal nonzero vectors in* R^n.

This paper, except for the introduction, is divided into three sections. In section 2 we make some preparations. We establish the sharpenings of the Hadamard inequality and the Szasz inequality in section 3. By using the results of sections 2 and 3, we will give the proof of Theorem 1.3 in section 4.

2. Some preliminary results. Suppose that Ω is the n-exterior differential form in $\mathbb{R}^n$, namely, $\Omega \in \wedge^n(\mathbb{R}^n)$. The following classical result holds for Ω (see [2]).

LEMMA 2.1. *Let* $\Omega \in \wedge^n(\mathbb{R}^n)$ *and* $\Omega \ne 0$; $\{v_1, v_2, \ldots, v_n\}$ *and* $\{x_1, x_2, \ldots, x_n\}$ *are the two bases of* R^n. *Assume that*

$$v_j = \sum_{i=1}^n a_{ij}x_i, \qquad j = 1, 2, \ldots, n.$$

Then

$$\Omega(v_1, v_2, \ldots, v_n) = \det(a_{ij})_{n\times n} \cdot \Omega(x_1, x_2, \ldots, x_n). \tag{2.1}$$

The main ingredient in the proof is the following well-known formula for the canonical volume forms (see [10]).

LEMMA 2.2. *Let* $\{x_1, x_2, \ldots, x_n\}$ *be a given linearly independent set of vectors from* $\mathbb{R}^n$, *and let* $\Omega \in \wedge^{n-1}(\mathbb{R}^n)$ *be the canonical volume form. Then*

$$\begin{aligned}\Omega(x_2 \wedge x_3 \wedge \cdots \wedge x_n, \quad & x_1 \wedge x_3 \wedge \cdots \wedge x_n, \ldots, x_1 \wedge x_2 \wedge \cdots \wedge x_{n-1}) \\ & = \Omega^{n-1}(x_1, x_2, \ldots, x_n).\end{aligned} \tag{2.2}$$

We need some other notations: Let U be a linear subspace of $\mathbb{R}^n$. For any $x \in \mathbb{R}^n$, it is easy to see that there are $w \perp U$ and $z \in U$ such that $x = w + z$. We call w the orthogonal component of x corresponding to U and call z the orthogonal projection of x corresponding to U. If $x \in \mathbb{R}^n$, we define that $\langle x, U\rangle$ is the angle between x and its orthogonal projection z.

LEMMA 2.3. *Let* T *and* S *be two linear subspaces of* $\mathbb{R}^n$, *and assume* $T \subseteq S$. *Then*

$$\sin\langle x, T\rangle \ge \sin\langle x, S\rangle. \tag{2.3}$$

Proof. Let w_1 and w_2 be orthogonal components of x corresponding to T and S, respectively. Since

$$\|w_1\| = \inf_{v\in T}\|x-v\| \geq \inf_{v\in S}\|x-v\| = \|w_2\|,$$

it follows that

$$\sin\langle x,T\rangle = \frac{\|w_1\|}{\|x\|} \geq \frac{\|w_2\|}{\|x\|} = \sin\langle x,S\rangle. \qquad \square$$

LEMMA 2.4. *Let $\{x_1,x_2,\ldots,x_n\}$ be a set of linearly independent vectors from R^n, and let S_k be the linear subspace spanned by $x_1,x_2,\ldots,x_k (k=1,2,\ldots,n)$. Then*

$$V(P_{[x_1,x_2,\ldots,x_n]}) = \prod_{i=1}^{n}\|x_i\|\prod_{i=2}^{n}\sin\langle x_i,S_{i-1}\rangle. \tag{2.4}$$

Proof. Let $x_i = w_i + z_i, w_i \perp S_{i-1}, z_i \in S_{i-1}$. Then

$$\|w_i\| = \|x_i\|\sin\langle x_i,S_{i-1}\rangle.$$

Hence, according to the recursive definition of the volume of a parallelotope, we have

$$V(P_{[x_1,x_2,\ldots,x_n]}) = \prod_{i=1}^{n}\|w_i\| = \prod_{i=1}^{n}\|x_i\|\prod_{i=2}^{n}\sin\langle x_i,S_{i-1}\rangle.$$

This is the desired equality. $\square$

3. The sharpenings of the Hadamard inequality. To prove Theorem 1.3, we establish the sharpenings of the Hadamard inequality (1.2) and the Szasz inequality (1.3).

The following theorem is a sharpening of the Hadamard inequality.

THEOREM 3.1. *Let $\{x_1,x_2,\ldots,x_n\}$ be a set of linearly independent vectors from $\mathbb{R}^n$, and let α_{ij} be the angle between x_i and x_j $(i,j=1,2,\ldots,n)$. Then*

$$V(P_{[x_1,x_2,\ldots,x_n]}) \leq \prod_{i=1}^{n}\|x_i\|\left(\prod_{1\leq i<j\leq n}\sin\alpha_{ij}\right)^{\frac{2}{n}}. \tag{3.1}$$

The equality in (3.1) *occurs if and only if $\{x_1,x_2,\ldots,x_n\}$ is a set of orthogonal nonzero vectors from $\mathbb{R}^n$.*

Proof. Since $S_1 \subseteq S_{i-1}$ $(i=2,3,\ldots,n)$, it follows from Lemma 2.3 that

$$\sin\langle x_i,S_{i-1}\rangle \leq \sin\alpha_{i1}. \tag{3.2}$$

Combining (2.4) and (3.2), we obtain

$$V(P_{[x_1,x_2,\ldots,x_n]}) \leq \prod_{i=1}^{n}\|x_i\|\prod_{i=2}^{n}\sin\alpha_{i1}. \tag{3.3}$$

By substituting x_1 by x_j in (3.3), we find

$$V(P_{[x_1,x_2,\ldots,x_n]}) \leq \prod_{i=1}^{n}\|x_i\|\prod_{\substack{i=1\\ i\neq j}}^{n}\sin\alpha_{ij}, \qquad j\in\{1,2,\ldots,n\}.$$

By multiplying all n obtained inequalities, we arrive at

$$V^n(P_{[x_1,x_2,\ldots,x_n]}) \le \left(\prod_{i=1}^{n}\|x_i\|\right)^n \prod_{j=1}^{n}\left(\prod_{\substack{i=1\\ i\neq j}}^{n}\sin\alpha_{ij}\right).$$

Rearranging it, inequality (3.1) follows. □

THEOREM 3.2. *Let $\{x_1, x_2, \ldots, x_n\}$ be a set of linearly independent vectors from $\mathbb{R}^n$, and let θ_{ij} be the dihedral angle between π_i and π_j. Then*

$$V^{n-1}(P_{[x_1,x_2,\ldots,x_n]}) \le \prod_{i=1}^{n} V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}) \left(\prod_{1\le i<j\le n}\sin\theta_{ij}\right)^{\frac{2}{n}}. \tag{3.4}$$

The equality in (3.4) *occurs if and only if $\{x_1, x_2, \ldots, x_n\}$ is a set of orthogonal nonzero vectors in $\mathbb{R}^n$.*

Proof. Let $\overline{e_i}$ be the unit normal vector of the facet $P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$. Then

$$x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_n = V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})\overline{e_i}.$$

By Lemma 2.1, we have

$$\begin{aligned}&\Omega(x_2 \wedge x_3 \wedge \cdots \wedge x_n, \quad x_1 \wedge x_3 \wedge \cdots \wedge x_n, \ldots, x_1 \wedge x_2 \wedge \cdots \wedge x_{n-1})\\ &\qquad = \left(\prod_{i=1}^{n} V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})\right)\Omega(\overline{e_1}, \overline{e_2}, \ldots, \overline{e_n}).\end{aligned} \tag{3.5}$$

Applying Lemma 2.2 to the left-hand side of (3.5), we obtain

$$\begin{aligned}V^{n-1}(P_{[x_1,x_2,\ldots,x_n]}) &= \Omega^{n-1}(x_1, x_2, \ldots, x_n)\\ &= \left(\prod_{i=1}^{n} V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})\right)\Omega(\overline{e_1}, \overline{e_2}, \ldots, \overline{e_n}).\end{aligned} \tag{3.6}$$

On the other hand, it is easy to see that the angle between $\overline{e_i}$ and $\overline{e_j}$ is equal to $\pi - \theta_{ij}$. Using Theorem 3.1, we get

$$\begin{aligned}\Omega(\overline{e_1}, \overline{e_2}, \ldots, \overline{e_n}) &= V(P_{[\overline{e_1}.\overline{e_2},\ldots,\overline{e_n}]})\\ &\le \left(\prod_{1\le i<j\le n}\sin(\pi-\theta_{ij})\right)^{\frac{2}{n}} = \left(\prod_{1\le i<j\le n}\sin\theta_{ij}\right)^{\frac{2}{n}}.\end{aligned} \tag{3.7}$$

From (3.6) and (3.7), (3.4) follows, as desired. □

Remark. The inequality (3.4) is a sharpening of the Szasz inequality (1.3).

4. The proof of Theorem 1.3.

THEOREM 4.1. *Let $\{x_1, x_2, \ldots, x_n\}$ be a given set of linearly independent vectors from $\mathbb{R}^n$, and let $P_{[x_1,x_2,\ldots,x_n]}$ and $P^*_{[x_1,x_2,\ldots,x_n]}$ be as in section* 1. *Then*

$$V(P^*_{[x_1,x_2,\ldots,x_n]})V(P_{[x_1,x_2,\ldots,x_n]}) = \left(\prod_{i=1}^{n}\|x_i\|\right)^2. \tag{4.1}$$

Proof. Let $z_1, z_2, \ldots, z_n$ be the n edge vectors of $P^*_{[x_1,x_2,\ldots,x_n]}$, which are linearly independent. Let F^* be the facet of

$$P^*_{[x_1,x_2,\ldots,x_n]} = \operatorname{span}\{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\}.$$

It should be noted that $P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}$ is not the facet of $P^*_{[x_1,x_2,\ldots,x_n]}$. Then

$$\frac{x_1}{\|x_1\|}, \frac{x_2}{\|x_2\|}, \ldots, \frac{x_n}{\|x_n\|}$$

are the unit normal vectors of $F_1^*, F_2^*, \ldots, F_n^*$, respectively.

Since

$$z_1 \wedge \cdots \wedge z_{i-1} \wedge z_{i+1} \wedge \cdots \wedge z_n = V(F_i^*)\frac{x_i}{\|x_i\|}$$

and

$$V(F_i^*)\|x_i\| = V(P^*_{[x_1,x_2,\ldots,x_n]}),$$

it follows that

$$\begin{aligned}\Omega(z_2 \wedge z_3 \wedge \cdots \wedge z_n, \quad & z_1 \wedge z_3 \wedge \cdots \wedge z_n, \ldots, z_1 \wedge z_2 \wedge \cdots \wedge z_{n-1}) \\ &= \prod_{i=1}^{n} \frac{V(F_i^*)}{\|x_i\|}\Omega(x_1, x_2, \ldots, x_n) \\ &= \frac{V^n(P^*_{[x_1,x_2,\ldots,x_n]})V(P_{[x_1,x_2,\ldots,x_n]})}{\left(\prod_{i=1}^{n}\|x_i\|\right)^2}.\end{aligned} \tag{4.2}$$

On the other hand, by Lemma 2.2, we have

$$\begin{aligned}\Omega(z_2 \wedge z_3 \wedge \cdots \wedge z_n, \quad & z_1 \wedge z_3 \wedge \cdots \wedge z_n, \cdots, z_1 \wedge z_2 \wedge \cdots \wedge z_{n-1}) \\ &= \Omega^{n-1}(z_1, z_2, \ldots, z_n) \\ &= V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]}).\end{aligned} \tag{4.3}$$

Therefore, (4.1) follows from (4.2) and (4.3). ∎

Proof of Theorem 1.3. From Theorems 4.1 and 3.1, we obtain

$$\begin{aligned}\left(\prod_{i=1}^{n}\|x_i\|\right)^2 &= V(P^*_{[x_1,x_2,\ldots,x_n]})V(P_{[x_1,x_2,\ldots,x_n]}) \\ &\le V(P^*_{[x_1,x_2,\ldots,x_n]})\prod_{i=1}^{n}\|x_i\|\left(\prod_{1\le i<j\le n}\sin\alpha_{ij}\right)^{\frac{2}{n}}.\end{aligned}$$

Rearranging the above inequality, (1.7) is proved.

Applying Theorem 4.1 to set $\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$, we have

$$V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]}) = \left(\prod_{\substack{j=1\\ j\ne i}}^{n}\|x_j\|\right)^2.$$

It follows that

$$\prod_{i=1}^{n}\left(V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})\right)=\left(\prod_{i=1}^{n}\|x_i\|\right)^{2(n-1)}. \tag{4.4}$$

Again applying Theorem 4.1 to the right-hand side of (4.4), we have

$$\begin{aligned}&\prod_{i=1}^{n}\left(V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})\right)\\&\quad=V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]})V^{n-1}(P_{[x_1,x_2,\ldots,x_n]});\end{aligned}$$

that is,

$$\frac{V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]})}{\prod_{i=1}^{n}V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})}=\frac{\prod_{i=1}^{n}V(P_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})}{V(P_{[x_1,x_2,\ldots,x_n]})}. \tag{4.5}$$

Hence, from Theorem 3.2 and (4.5), we derive inequality (1.6), so Theorem 1.3 is proved. □

By (1.2) and (1.7), we can easily derive the following interesting inequality.

COROLLARY 4.2.

$$\frac{V(P_{[x_1,x_2,\ldots,x_n]})}{V(P^*_{[x_1,x_2,\ldots,x_n]})}\le\left(\prod_{1\le i<j\le n}\sin\alpha_{ij}\right)^{2/n}\le 1. \tag{4.6}$$ □

Let

$$\theta=\frac{\sum_{1\le i<j\le n}\theta_{ij}}{\binom{n}{2}},\qquad \alpha=\frac{\sum_{1\le i<j\le n}\alpha_{ij}}{\binom{n}{2}}.$$

Since $y=\sin x$ is a convex function in $[0,\pi]$, by applying the arithmetricogeometric mean inequality and the Jensen inequality, we get

$$\begin{aligned}\left(\prod_{1\le i<j\le n}\sin\theta_{ij}\right)^{\frac{2}{n}}&\le\left[\frac{\sum_{1\le i<j\le n}\sin\theta_{ij}}{\binom{n}{2}}\right]^{n-1}\\&\le\sin^{n-1}\left[\frac{\sum_{1\le i<j\le n}\theta_{ij}}{\binom{n}{2}}\right]\\&\le(\sin\theta)^{n-1}.\end{aligned}$$

Similarly,

$$\left(\prod_{1\le i<j\le n}\sin\alpha_{ij}\right)^{\frac{2}{n}}\le(\sin\alpha)^{n-1}.$$

Therefore, from Theorem 1.3 *and Corollary* 4.2*, we immediately get the following corollary.*

COROLLARY 4.3.

$$\frac{V^{n-1}(P^*_{[x_1,x_2,\ldots,x_n]})}{\prod_{i=1}^n V(P^*_{[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]})} \geq \frac{1}{(\sin\theta)^{n-1}},$$

$$\frac{V(P^*_{[x_1,x_2,\ldots,x_n]})}{\prod_{i=1}^n \|x_i\|} \geq \frac{1}{(\sin\alpha)^{n-1}},$$

$$\frac{V(P_{[x_1,x_2,\ldots,x_n]})}{V(P^*_{[x_1,x_2,\ldots,x_n]})} \leq (\sin\alpha)^{n-1}.$$

REFERENCES

[1] E. F. BECKENBACH AND R. BELLMAN, *Inequalities*, Springer, Berlin, 1961.

[2] W. M. BOOTHBY, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, New York, 1975.

[3] J. D. DIXON, *How good is Hadamard's inequality for determinants?*, Canad. Math. Bull., 27 (1984), pp. 260–264.

[4] J. D. DIXON, *Estimating extremal eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal., 20 (1983), pp. 812–814.

[5] G. M. ENGLE AND H. SCHNEIDER, *The Hadamard-Fischer inequality for a class of matrices defined by eigenvalue monotonicity*, Linear and Multilinear Algebra, 4 (1976), pp. 155–176.

[6] R. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[7] C. R. JOHNSON, *Optimization, matrix inequalities and matrix completions*, in Operator Theory, Analytic Function, Matrices and Electrical Engineering, CBMS Reg. Conf. Ser. Math. 68, J. W. Melton, ed., AMS, Providence, RI, 1987.

[8] C. R. JOHNSON AND T. MARKHAM, *Compression and Hadamard power inequalities for M-matrices*, Linear and Multilinear Algebra, 18 (1985), pp. 23–34.

[9] C. R. JOHNSON AND M. NEWMAN, *How bad is the Hadamard determinantal bound?*, J. Res. Nat. Bureau Stand., 78 (1974), pp. 167–169.

[10] A. G. REZNIKOV, *A strengthened isoperimetric inequality for simplices*, in Geometric Aspects of Functional Analysis (1989-90), Lecture Notes in Math. 1469, J. Lindenstrauss and V. D. Milmar, eds., Springer, New York, 1991, pp. 90–93.

[11] A. G. REZNIKOV, *Determinant inequalities with applications to isoperimetrical inequalities*, in Geometric Aspects of Functional Analysis (Israel, 1992-94), Oper. Theory Adv. Appl. 77, Birkäuser, Basel, 1995, pp. 239–244.

[12] J. S. SHIUE, *On a generalization of a theorem of Johnson and Newman*, Soochow J. Math. Natur. Sci., 2 (1976), pp. 57–61.

[13] D. VELJAN, *The sine theorem and inequalities for volume of simplices and determinants*, Linear Algebra Appl., 219 (1995), pp. 79–91.

[14] H. WOLKOWITZ AND G. STYAN, *Bounds for eigenvalues using traces*, Linear Algebra Appl., 29 (1980), pp. 471–506.

[15] ZHANG XIAODONG AND YANG SHANGJUN, *An improvement of Hadamard's inequality for totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 705–711.

[16] ZHANG XIAODONG AND YANG SHANGJUN, *A note on Hadamard's inequality*, Acta Math. Appl. Sinica, 20 (1997), pp. 269–274.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 998–1012
© 2002 Society for Industrial and Applied Mathematics

NESTED-DISSECTION ORDERINGS FOR SPARSE LU WITH PARTIAL PIVOTING*

IGOR BRAINMAN† AND SIVAN TOLEDO†

Abstract. We describe the implementation and performance of a novel fill-minimization ordering technique for sparse LU factorization with partial pivoting. The technique was proposed by Gilbert and Schreiber in 1980 but never implemented and tested. Like other techniques for ordering sparse matrices for LU with partial pivoting, our new method preorders the columns of the matrix (the row permutation is chosen by the pivoting sequence during the numerical factorization). Also like other methods, the column permutation Q that we select is a permutation that attempts to reduce the fill in the Cholesky factor of $Q^T A^T A Q$. Unlike existing column-ordering techniques, which all rely on minimum-degree heuristics, our new method is based on a nested-dissection ordering of $A^T A$. Our algorithm, however, never computes a representation of $A^T A$, which can be expensive. We only work with a representation of A itself. Our experiments demonstrate that the method is efficient and that it can reduce fill significantly relative to the best existing methods. The method reduces the LU running time on some very large matrices (tens of millions of nonzeros in the factors) by more than a factor of 2.

Key words. nested-dissection, vertex separators, wide separators, LU factorization, Gaussian elimination, partial pivoting

AMS subject classifications. 05C50, 15A23, 65F05, 65F50

PII. S0895479801385037

1. Introduction.

Reordering the columns of sparse nonsymmetric matrices can significantly reduce fill in sparse LU factorizations with partial pivoting. Reducing fill in a factorization reduces the amount of memory required to store the factors, the amount of work in the factorization, and the amount of work in subsequent triangular solves. Symmetric positive-definite matrices, which can be factored without pivoting, are normally reordered to reduce fill by applying the same permutation to both the rows and columns of the matrix. Applying the same permutation to the rows and columns preserves the symmetry of the matrix. When partial pivoting is required for maintaining numerical stability, however, prepermuting the rows is meaningless, since the rows are exchanged again during the factorization. Therefore, we often preorder the columns and let numerical considerations dictate the row ordering. Since columns are reordered before the row permutation is known, we need to order the columns such that fill is reduced no matter how rows are exchanged. (Some nonsymmetric factorization codes that employ pivoting, such as MA38 [5, 6], determine the column permutation during the numerical factorization; such codes do not preorder columns, so the technique in this paper does not apply to them.)

A result by George and Ng [10] suggests one effective way to preorder the columns to reduce fill. They have shown that the fill of the LU factors of PA is essentially contained in the fill of the Cholesky factor of $A^T A$ for every row permutation P. (P is a permutation matrix that permutes the rows of A and represents the actions of partial pivoting.) Gilbert and Ng [13] later showed that this upper bound on the fill

*Received by the editors February 14, 2001; accepted for publication (in revised form) by E. Ng October 24, 2001; published electronically April 10, 2002. This research was supported by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities (grant number 572/00 and grant number 9060/99) and by the University Research Fund of Tel-Aviv University.
http://www.siam.org/journals/simax/23-4/38503.html

†School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (stoledo@tau.ac.il, http://www.tau.ac.il/~stoledo).

of the U factor is not too loose, in the sense that for a large class of matrices, for every fill element in the Cholesky factor of A^TA there is a pivoting sequence P that causes the element to fill in the U factor of A. Thus, nonsymmetric direct sparse solvers often preorder the columns of A using a permutation Q that attempts to reduce fill in the Cholesky factor of Q^TA^TAQ.

The main challenge in column-ordering algorithms is to find a fill-minimizing permutation without computing A^TA or even its nonzero structure. While computing the nonzero structure of A^TA allows us to use existing symmetric ordering algorithms and codes, it may be grossly inefficient. For example, when an n-by-n matrix A has nonzeros only in the first row and along the main diagonal, computing A^TA takes $\Omega(n^2)$ work, but factoring it takes only $O(n)$ work. Consider an n-by-n matrix in which all the nonzeros are in the first row and along the main diagonal, such as ($\times$'s represent nonzeros)

$$A = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ & \times & & & & \\ & & \times & & & \\ & & & \times & & \\ & & & & \times & \\ & & & & & \times \end{bmatrix}.$$

The matrix A^TA is full, so computing its structure requires at least $\Theta(n^2)$ work. But since A^TA is full, all its orderings are equivalent in terms of fill. Thus, we perform $\Omega(n^2)$ work and get no useful information. If we factor this matrix without reordering its columns, no pivoting takes place and no fill is produced, so the factorization requires only $\Theta(n)$ work. To summarize, computing A^TA may require significantly more memory and work than the partial-pivoting numerical factorization requires. (This example is somewhat weak, in that A is already triangular. The sum of a tridiagonal matrix and a zero matrix with one dense row provides a stronger example. Such a matrix A cannot be permuted to a nontrivial block triangular form, and forming A^TA requires $\Theta(n^2)$ work, whereas factoring it may take as little as $\Theta(n)$ if the dense row is used as the last pivot row.)

This challenge has been met for the class of reordering algorithms based on the minimum-degree heuristic. Modern implementations of minimum-degree heuristics use a *clique-cover* to represent the graph G_A of the matrix[1] A (see [9]). A clique-cover represents the edges of the graph (the nonzeros in the matrix) as a union of cliques, or complete subgraphs. The clique-cover representation allows us to simulate the elimination process with a data structure that only shrinks and never grows. There are two ways to initialize the clique-cover representation of G_{A^TA} directly from the structure of A. Both ways create a data structure whose size is proportional to the number of nonzeros in A, not the number of nonzeros in A^TA. From then on, the data structure only shrinks, so it remains small even if A^TA is relatively dense. In other words, finding a minimum-degree column ordering for A requires about the same amount of work and memory as finding a symmetric ordering for $A^T + A$, the symmetric completion of A.

Nested-dissection ordering methods were proposed by George in the early 1970s [11]. Berman and Schnitger [2] showed that they are theoretically superior to minimum-degree methods for important classes of sparse symmetric definite matrices. Only

[1] The graph $G_A = (V, E)$ of an n-by-n matrix A has a vertex set $V = \{1, 2, \ldots, n\}$ and an edge set $E = \{(i, j) | a_{ij} \neq 0\}$. We ignore numerical cancellations in this paper.

in the last few years, however, have nested-dissection methods been shown experimentally to be more effective than minimum-degree methods. Today's state-of-the-art methods use fast multilevel algorithms for finding separators and fuse nested-dissection and minimum-degree to reduce fill below the level that either method alone produces.

In 1980 Gilbert and Schreiber proposed a method for ordering G_{A^TA} when A is symmetrically structured using nested-dissection heuristics, without ever forming A^TA [12, 14]. Their method uses *wide separators*, a term that they coined. They have never implemented or tested their proposed method.

The main contribution of this paper is an implementation and an experimental evaluation of the wide-separator ordering method, along with a new presentation of the theory of wide separators. Our code dissects G_{A^TA} without forming it. The code then uses existing techniques for minimum-degree column ordering to reduce fill in LU with partial pivoting to below that of any existing technique.

Modern symmetric ordering methods generally work as follows:

1. The methods find a small vertex separator that separates the graph G into two subgraphs with roughly the same size.

2. Each subgraph is dissected recursively until each subgraph is fairly small (typically several hundred vertices).

3. The separators are used to impose a coarse ordering. In *nested-dissection* codes, the vertices in the top-level separator are ordered last, the vertices in the second-to-top level come before them, and so on. The vertices in the small subgraphs that are not dissected any further appear first in the ordering. The ordering within each separator and the ordering within each subgraph have not yet been determined. In *multisector* codes the vertices in all the separators are ordered last and the other vertices ordered first. The ordering within the multisector and the ordering of nonmultisector vertices has not yet been determined.

4. A minimum-degree algorithm computes the final ordering, subject to the coarse ordering constraints.

While there are many variants, most codes use this overall framework.

Our methods apply the same framework to the graph of A^TA, but without computing it. We find separators in A^TA by finding *wide separators* in $A^T + A$. We find a wide separator by finding a conventional vertex separator and widening it by adding to it all the vertices that are adjacent to the separator in one of the subgraphs. Such a wide separator corresponds to a vertex separator in A^TA. Just like symmetric methods, our methods recursively dissect the graph, but using wide separators. When the remaining subgraphs are sufficiently small, we compute the final ordering using a constrained column-minimum-degree algorithm. We use nested-dissection-type constraints, as opposed to multisector constraints. We use existing techniques to produce a minimum-degree ordering of A^TA without computing G_{A^TA} (either the row-clique method or the augmented-matrix method).

The (conventional) vertex-separator code that we use is part of a library called SPOOLES [1]. Our code can use SPOOLES's minimum-degree code, as well as a version of COLAMD [7, 15] that we modified to respect the coarse ordering.

Experimental results show that our method can reduce the work in the LU factorization by up to a factor of 3 compared to state-of-the-art column-ordering codes. The running times of our method are higher than the running times of strict minimum-degree codes, such as COLAMD, but they are low enough to easily justify using the new method. On many matrices, including large ones, our method significantly reduces

the work compared to all the existing column-ordering methods. On some matrices, however, constraining the ordering using wide separators increases fill rather than reducing it.

The rest of the paper is organized as follows. Section 2 presents the theory of wide separators and algorithms for finding them. Our experimental results are presented in section 3. We discuss our conclusions from this research in section 4.

2. Wide separators: Theory and algorithms. Our column-ordering methods find separators in G_{A^TA} by finding a so-called *wide separator* in G_{A^T+A}. We work with the graph of $A^T + A$ and not with G_A for two reasons. First, this simplifies the definitions and proofs. Second, to the best of our knowledge all existing vertex-separator codes work with undirected graphs, so there is no point in developing the theory for the directed graph G_A.

A vertex subset $S \subseteq V$ of an undirected graph $G = (V, E)$ is a *separator* if the removal of S and its incident edges breaks the graph into two components $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that any path between $i \in V_1$ and $j \in V_2$ passes through at least one vertex in S. A vertex set is a *wide separator* if every path between $i \in V_1$ and $j \in V_2$ passes through a sequence of two vertices in S (one after the other along the path).

Our first task is to show that every wide separator in G_{A^T+A} is a separator in G_{A^TA}. The next theorem proves this result. Figures 1 and 2 illustrate the same result using matrices rather than graphs.

THEOREM 2.1. *A wide separator in G_{A^T+A} is a separator in G_{A^TA}.*

Proof. Let S be a wide separator in G_{A^T+A}. Suppose for contradiction that S is not a separator in G_{A^TA}: there exists a path in G_{A^TA} between $i \in V_1$ and $j \in V_2$ that does not pass though a vertex s in S. There must be a pair of vertices i' and j' along the path such that $i' \in V_1$ and $j' \in V_2$. Thus, i' and j' are neighbors in G_{A^TA}, so the (i', j') element in A^TA is nonzero. Since $(A^TA)_{i',j'} = \sum_k A_{k,i'}A_{k,j'} \neq 0$, there must be some k such that $A_{k,i'} \neq 0$ and $A_{k,j'} \neq 0$. Hence, there is a path $i' \leftrightarrow k \leftrightarrow j'$ in G_{A^T+A} between i' and j' that passes through only one vertex in S, a contradiction. ∎

The converse is not always true. There are matrices with separators in G_{A^TA} that do not correspond to wide separators in $A^T + A$. Consider

$$A = \begin{bmatrix} & \times \\ \times & \end{bmatrix}$$

(the $\times$'s represent nonzeros). The empty set is a separator in the graph of

$$A^TA = \begin{bmatrix} & \times \\ \times & \end{bmatrix} \begin{bmatrix} & \times \\ \times & \end{bmatrix} = \begin{bmatrix} \times & \\ & \times \end{bmatrix},$$

but it is not a wide separator in the graph of $A^T + A$ (it is not even a separator). The converse of the Theorem 2.1 is true, however, for symmetrically structured matrices with no zeros on the main diagonal.

THEOREM 2.2. *If A is symmetrically structured with no zeros on the diagonal, then a separator in G_{A^TA} is a wide separator in G_{A^T+A}.*

Proof. Let S be a separator in G_{A^TA}. Suppose for contradiction that S is not a wide separator in G_{A^T+A}. There exists a path in G_{A^T+A} between some $i \in V_1$ and $j \in V_2$ that does not pass though a sequence of two vertices in S. This can happen in two ways: (1) there are some $i' \in V_1$ and $j' \in V_2$ that are adjacent in G_{A^T+A}(that is,

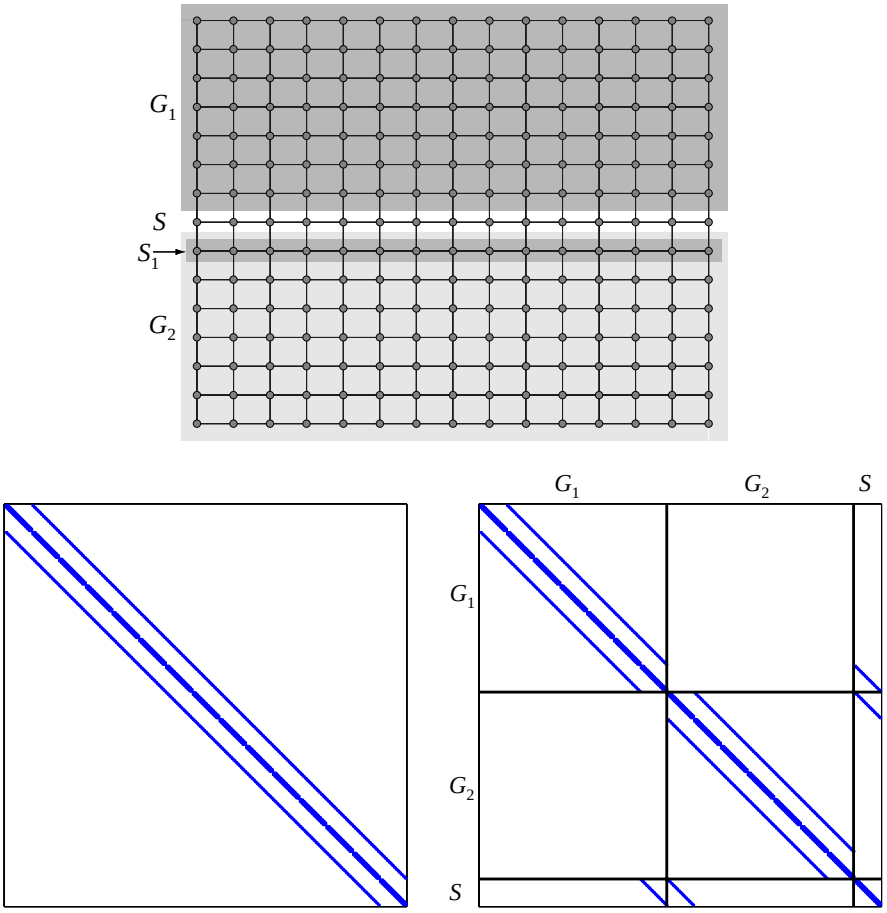


FIG. 1. *A separator and a wide separator. The top figure shows a* 15*-by-*15 *mesh* G *which has been partitioned into* G_1 *and* G_2 *using a separator* S*. The vertex set* $S_1 \subseteq G_2$ *widens the separator:* $S \cup S_1$ *widely separates* G_1 *from* $G_2 \setminus S_1$*. When the vertices are ordered row by row, top to bottom, the nonzero pattern of the matrix of the graph is pentadiagonal (bottom left). Permuting the rows and the columns of the matrix so that* S*'s vertices appear last results in a* 3*-by-*3 *block matrix with large zero* 21 *and* 12 *blocks (bottom right). (Continued in Figure* 2*.)*

S is not a separator at all in G_{A^T+A}), or (2) there are some $i' \in V_1$ and $j' \in V_2$ that are separated in G_{A^T+A} but not widely; there is a path $i' \leftrightarrow s \leftrightarrow j'$ in G_{A^T+A}.

In both cases the edge (i', j') will be in G_{A^TA}, as illustrated in Figure 3. In case (1), (i', j') will be in G_{A^TA} because $A^T_{i',i'} = A_{i',i'}$ and $A_{i',j'}$ are nonzero and because $A^T_{i',j'} = A_{j',i'}$ and $A_{j',j'}$ are nonzero. In case (2), (i', j') will be in G_{A^TA} because $A^T_{i',s} = A_{s,i'}$ and $A_{s,j'}$ are nonzero. □

The theorem does not hold for matrices without a symmetric structure, even if they have a nonzero diagonal. Consider

$$A = \begin{bmatrix} \times & & \times \\ & \times & \times \\ & & \times \end{bmatrix}.$$

Vertex 3 is a separator in G_{A^TA}, but not a wide separator in G_{A^T+A}.

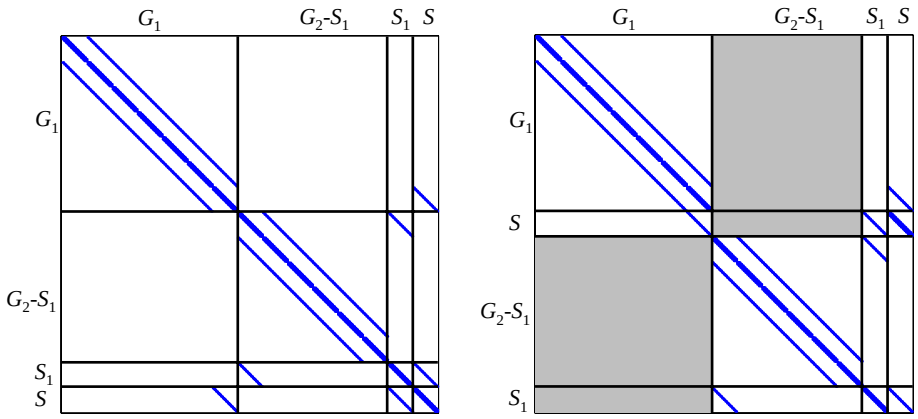


FIG. 2. *(Continued from Figure* 1*.) The permuted matrix in Figure* 1 *has many zero columns in its* 32 *block, which can be blocked together by permuting the rows and columns of* S_1 *to the end of the* G_2 *(left). By permuting only the rows so that the rows of* $G_1 \cup S$ *appear first and the rows of* G_2 *appear last, we observe that two large rectangular zero blocks have formed (in gray). The zeros in this block are preserved by LU with partial pivoting; the columns of* $S \cup S_1$ *form a column separator. In practice there is no need to reorder the rows: we have reordered them in the figure only to reveal the zero blocks.*

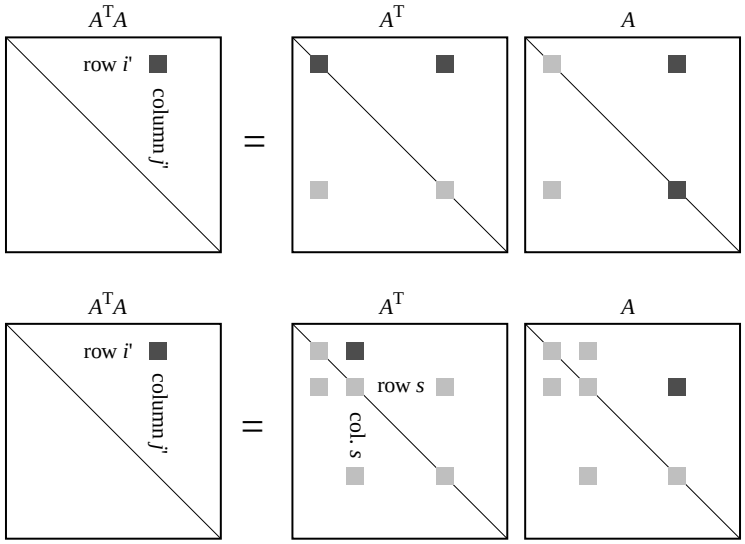


FIG. 3. *An illustration of the two cases in Theorem* 2.2. *Case* (1) *is illustrated on top, and case* (2) *on the bottom. The nonzeros, represented by the dark squares, are the ones that cause element* i', j' *in* $A^T A$ *to fill.*

Given a code that finds conventional separators in an undirected graph, finding wide separators is easy. The separator and its neighbors in either G_1 or G_2 form a wide separator, as stated by the following lemma.

LEMMA 2.3. *Let* S *be a separator in an undirected graph* G*. The sets* $S_1 = S \cup \{i | i \in V_1, (i,j) \in E \text{ for some } j \in S\}$ *and* $S_2 = S \cup \{i | i \in V_2, (i,j) \in E \text{ for some } j \in S\}$ *are wide separators in* G*.*

The proof is trivial. The sizes of S_1 and S_2 are bounded by $d|S|$, where d is the maximum degree of vertices in S. Given S, it is easy to enumerate S_1 and S_2 in time $O(d|S|)$. This running time is typically insignificant compared to the time it takes to find S.

Which one of the two candidate wide separators should we choose? A wide separator that is small and that dissects the graph evenly reduces fill in the Cholesky factor of A^TA, and hence in the LU factors of A. The two criteria are usually contradictory. Over the years it has been determined that the best strategy is to choose a separator that is as small as possible, as long as the ratio of the number of vertices in G_1 and G_2 does not exceed 2 or so.

The following method is, therefore, a reasonable way to find a wide separator: Select the smallest of S_1 and S_2, unless the smaller wide separator unbalances the separated subgraphs (so that one is more than twice as large as the other) but the larger does not. Our code, however, is currently more naive and always chooses the smaller wide separator.

3. Experimental results. This section summarizes our experimental results. We begin by describing our code, our collection of test matrices, and the computer that was used to carry out the experiments. We then describe and analyze the results of our experiments. The analyses focus on the effectiveness of various ordering methods and on their performance. By effectiveness we mean the number of nonzeros in the factors, the number of floating-point operations (flops) required to compute them, and the factorization time. By performance we mean the cost, mostly in terms of time, of the ordering algorithm itself.

3.1. Experimental setup. The experiments that this section describes test the effectiveness and performance of several column-ordering codes. We have tested our new codes, which implement nested-dissection-based orderings, as well as two existing ordering codes.

Our codes build a hierarchy of wide separators and use the separators to constrain a minimum-degree algorithm. We obtain the wide separators by widening separators in G_{A^T+A} that SPOOLES [1] finds. SPOOLES is a library of sparse ordering and factorization codes written by Ashcraft and others. Our codes then invoke a column-minimum-degree code to produce the final ordering. One minimum-degree code that we use is SPOOLES's multistage-minimum-degree (MSMD) algorithm, which we run on the augmented matrix

$$\widetilde{A} = \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} .$$

We constrain the minimum-degree code to eliminate the first n rows/columns first. This elimination constructs a clique-cover representation of G_{A^TA} on the remaining vertices, which MSMD eliminates next under the wide-separator constraints.

The other minimum-degree code that we used is a version of COLAMD [7, 15] that we modified to respect the constraints imposed by the separators.

We use the following acronyms to refer to the ordering methods: MSMD refers to SPOOLES's minimum-degree code operating on the augmented matrix without constraints, WSMSMD refers to the same minimum-degree code but constrained to respect wide separators, and similarly for COLAMD and WSCOLAMD.

In the experiments reported here, we always reduce the input matrices to *block triangular form* (see [17]) and factor only the diagonal blocks in the reduced form.

Many of the matrices in our test suite have numerous tiny diagonal blocks (most of them 1-by-1); we report the performance of factoring all the diagonal blocks with dimension at least 250.

We factor the reordered matrix using SuperLU [8, 16] version 2.0, a state-of-the-art sparse partial-pivoting LU code. SuperLU uses the Basic Linear Algebra Subroutines (BLAS). We used ATLAS,[2] a high-performance implementation of the BLAS.

We conducted the experiments on a 600MHz dual Pentium III computer with 2 GBytes of main memory running Linux. The machine was configured without swap space so no paging occurred during the experiments. This machine has two processors, but our code uses only one processor. The compiler that we used is GCC version egcs-2.91.66. We used the recommended optimization level for each package: `-O` for SPOOLES and `-O3` for SuperLU.

3.2. Matrices. We tested the ordering methods on a set of nonsymmetric sparse matrices from Davis's sparse matrix collection.[3] We used all the nonsymmetric matrices in Davis's collection that are not too small (factorization time with the best ordering method at least 1/1000 of a second). Two of the matrices in Davis's collection were too large to factor on our machine (`appu` and `pre2`) and SPOOLES broke down on two more (`av41092` and `twotone`; we are unsure whether the breakdown was due to a bug in our code or due to a problem in SPOOLES).

The matrices are listed in Tables 1 and 2. We split the matrices into small ones and large ones based on the number of flops in the factorization. We refer to matrices whose factorization with the best ordering requires more than 100 Mflops (millions of flops) as *large*, the rest are referred to as small. We always sort matrices by this factorization-flops metric. The tables show the matrix's name, dimension (N), number of nonzeros (NNZ), number of blocks in the block triangular form, number of big blocks (dimension at least 250) in the block triangular form, and the best flop count in millions.

Note that three of the matrices, `psmigr_1`, `psmigr_2`, and `psmigr_3`, are essentially dense. Although they are fairly sparse to begin with, as shown by the number of nonzeros, they fill tremendously with all the ordering codes. The number of flops to factor them with even the best ordering code is about 75% of the flop count required to factor a dense matrix of the same order, so their sparsity is insignificant.

We also run experiments on matrices whose graphs are regular 2- and 3-dimensional meshes and whose values are random numbers in the range $[0, 1]$.

3.3. Results and analysis. Table 3 and Figures 4, 5, and 6 summarize the results of our experiments. These results supersede the preliminary results that we reported in [3, 4].

Table 3 shows that wide-separator (WS) orderings are both effective and efficient. On the largest 2- and 3-dimensional meshes, WS orderings lead to the fastest factorization times and to the fastest overall solution time (including ordering time). Beyond performance, WS orderings enable us to solve problems that we could simply not solve with minimum-degree orderings with this amount of main memory (2GB).

Wide-separator orderings are not effective on small matrices. Of the 45 small matrices in our test suite, WS orderings reduce flop counts significantly (by more than 25%) over COLAMD on only 2 matrices (`ex8` and `ex9`). We note, however, that even

[2] http://www.netlib.org/atlas

[3] http://www.cise.ufl.edu/research/sparse/matrices

TABLE 1
General information for the small matrices.

NO	NAME	N	NNZ	BTF BLOCKS	BIG BTF BLOCKS	BEST MFLOPS
1	raefsky6	3402	137845	3402	0	0
2	raefsky5	6316	168658	6316	0	0
3	poli_large	15575	33074	15466	0	0
4	bwm2000	2000	7996	1	1	0
5	epb0	1794	7764	1	1	0
6	cavity04	317	7327	82	0	0
7	lhr01	1477	18592	298	1	2
8	rdist2	3198	56934	199	1	4
9	bayer02	13935	63679	2151	1	7
10	bayer10	13436	94926	1541	1	8
11	rdist3a	2398	61896	99	1	8
12	rdist1	4134	94408	199	1	8
13	orani678	2529	90158	700	1	17
14	lhr04c	4101	82682	439	1	17
15	lhr04	4101	82682	439	1	18
16	bayer04	20545	159082	6378	1	18
17	ex9	3363	99471	1	1	23
18	lhr07c	7337	156508	672	2	27
19	lhr07	7337	156508	672	2	27
20	ex31	3909	115357	1	1	28
21	rw5151	5151	20199	6	1	30
22	bayer01	57735	277774	8861	1	30
23	ex28	2603	77781	1	1	35
24	lhr11	10964	233741	1192	3	35
25	lhr11c	10964	233741	1192	3	35
26	lhr10c	10672	232633	908	3	36
27	lhr10	10672	232633	908	3	36
28	ex19	12005	259879	305	3	36
29	memplus	17758	126150	1	1	39
30	lhr14	14270	307858	1556	5	42
31	lhr14c	14270	307858	1556	5	42
32	lhr17	17576	381975	1798	6	55
33	lhr17c	17576	381975	1798	6	56
34	ex8	3096	106344	1	1	68
35	ex35	19716	228208	173	4	73
36	cavity26	4562	138187	322	1	91
37	cavity24	4562	138187	322	1	91
38	onetone2	36057	227628	3843	1	92
39	cavity25	4562	138187	322	1	92
40	cavity23	4562	138187	322	1	92
41	cavity22	4562	138187	322	1	92
42	cavity21	4562	138187	322	1	94
43	cavity20	4562	138187	322	1	94
44	cavity19	4562	138187	322	1	95
45	cavity18	4562	138187	322	1	98

though wide separators do not reduce work in the factorization of small matrices, they rarely increase work by a factor of 2 or more. Since wide-separator orderings do not appear to be effective on small matrices, the rest of this section refers only to large matrices.

Figures 4, 5, and 6 summarize the results with large matrices from a test-matrix collection. A comparison of the best WS method to the best non-WS method, shown in Figure 6, shows that WS orderings are effective. WS and non-WS orderings produced similar flop counts (within 25%) on 14 of the 33 matrices. WS orderings reduced flop

TABLE 2
General information for the large matrices.

NO	NAME	N	NNZ	# OF BLOCKS	# OF BIG BLOCKS	BEST MFLOPS
46	cavity17	4562	138187	322	1	101
47	epb1	14734	95053	1	1	123
48	utm5940	5940	83842	147	1	126
49	lhr34	35152	764014	3533	10	128
50	lhr71	70304	1528092	7066	20	259
51	lhr71c	70304	1528092	7066	20	262
52	shyy161	76480	329762	25761	1	479
53	epb2	25228	175027	1	1	517
54	goodwin	7320	324784	2	1	524
55	epb3	84617	463625	1	1	809
56	raefsky2	3242	294276	1	1	921
57	raefsky1	3242	294276	1	1	921
58	graham1	9035	335504	478	1	989
59	garon2	13535	390607	1	1	1061
60	ex40	7740	458012	1	1	1075
61	rim	22560	1014951	2	1	1877
62	onetone1	36057	341088	3843	1	2371
63	olafu	16146	1015156	1	1	2584
64	venkat01	62424	1717792	1	1	4299
65	venkat50	62424	1717792	1	1	4299
66	venkat25	62424	1717792	1	1	4299
67	rma10	46835	2374001	1	1	4386
68	af23560	23560	484256	1	1	4515
69	raefsky3	21200	1488768	1	1	5243
70	raefsky4	19779	1328611	1	1	7800
71	ex11	16614	1096948	1	1	11194
72	psmigr_2	3140	540022	1	1	13412
73	psmigr_3	3140	543162	1	1	14649
74	psmigr_1	3140	543162	1	1	14776
75	wang3	26064	177168	1	1	15515
76	wang4	26068	177196	1	1	24484
77	bbmat	38744	1771722	1	1	44553
78	li	22695	1350309	2	2	84241

TABLE 3
A comparison of wide-separator and minimum-degree orderings on regular 2*- and* 3*-dimensional meshes. All the matrix entries are random. The first three columns show the dimensions of the meshes, the next two the best factorization time and the ordering method that led to the best time. The last four columns show the combined ordering and factorization times. All times are reported in seconds.*

N_x	N_y	N_z	Best Fact. Time	Best Method	Ordering+Factorization Times WSCOLAMD	COLAMD	WSCOLMSMD	COLMSMD
500	500		113	WSCOLAMD	150	202	150	—
750	750		496	WSCOLAMD	601	—	684	—
30	30	30	352	COLAMD	399	352	1210	404
40	40	40	786	WSCOLAMD	792	2340	958	—

counts by more than 25% on 12 matrices including 5 of the 10 largest. On the other hand, WS orderings increased flop counts on only 7 matrices, none of them in the top 10. The results with COLMSMD and WSCOLMSMD, shown in Figure 5, are even better: the overall numbers are the same, but WS orderings reduce work significantly on 7 matrices in the top 10. The results with COLAMD and WSCOLAMD are a bit less favorable to WS orderings: they reduce work on 9 matrices but increase work on 8

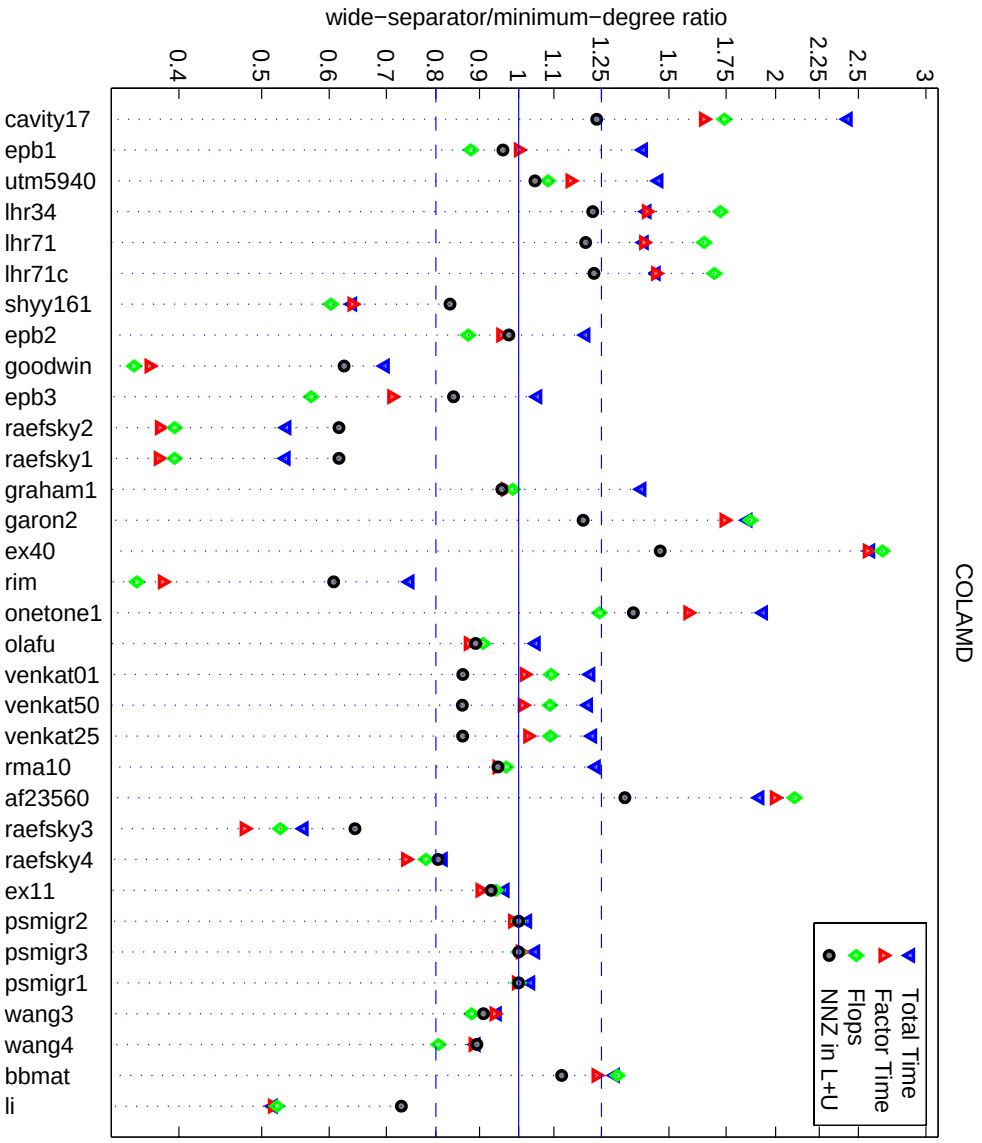


FIG. 4. *The ratios of* WSCOLAMD*'s performance to that of* COLAMDS. *Performance is reported in terms of flops, number of nonzeros in* L *and* U*, factorization times, and total times (including ordering). Data points below* 1 *indicate that* WSCOLAMD *is better than* COLAMD. *Matrices are sorted by best factorization flops. The y-axis is logarithmic.*

(discounting relative differences of less than 25%).

Nonzero counts in the LU factors and factorization times are generally correlated with flop counts; smaller flop counts usually imply fewer nonzeros in the LU factors and shorter factorization times.

The improvement due to wide separators is often large. On the largest matrix in our test suite, `li`, wide separators reduce flop counts and factorization time by about a factor of 2. The reduction in terms of flop counts compared to non-WS methods is also highly significant on `wang3`, `raefsky1/2/3`, `rim`, and especially `epb3`.

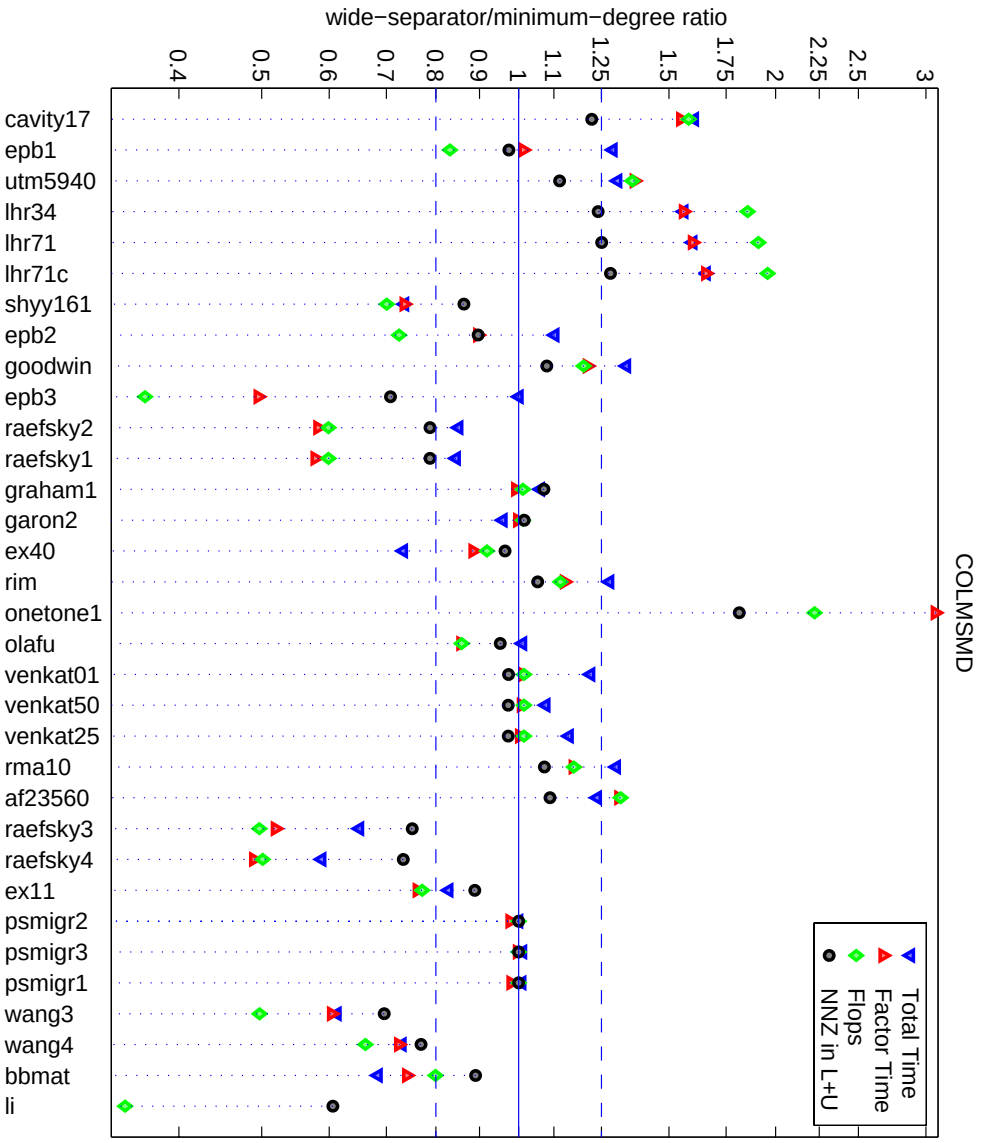


FIG. 5. WSCOLMSMD *to* COLMSMD *ratios.*

When WS orderings do poorly compared to non-WS methods, however, they sometimes do significantly poorer. On `ex40`, for example, using wide separators slows down the factorization by a factor of about 2.5. Figure 7 shows that without reduction to block triangular form, the slowdowns are even more dramatic. The figure shows that on some of the matrices, especially the `lhr` ones, reduction to block triangular form saves significant amounts of time, and that the savings are larger for WS orderings than for non-WS orderings.

WS orderings are somewhat more expensive to compute than strict minimum-degree orderings. Figure 6 shows that when the ordering times are taken into account, WS orderings speed up the total solution time by more than 25% in only 6 out of the

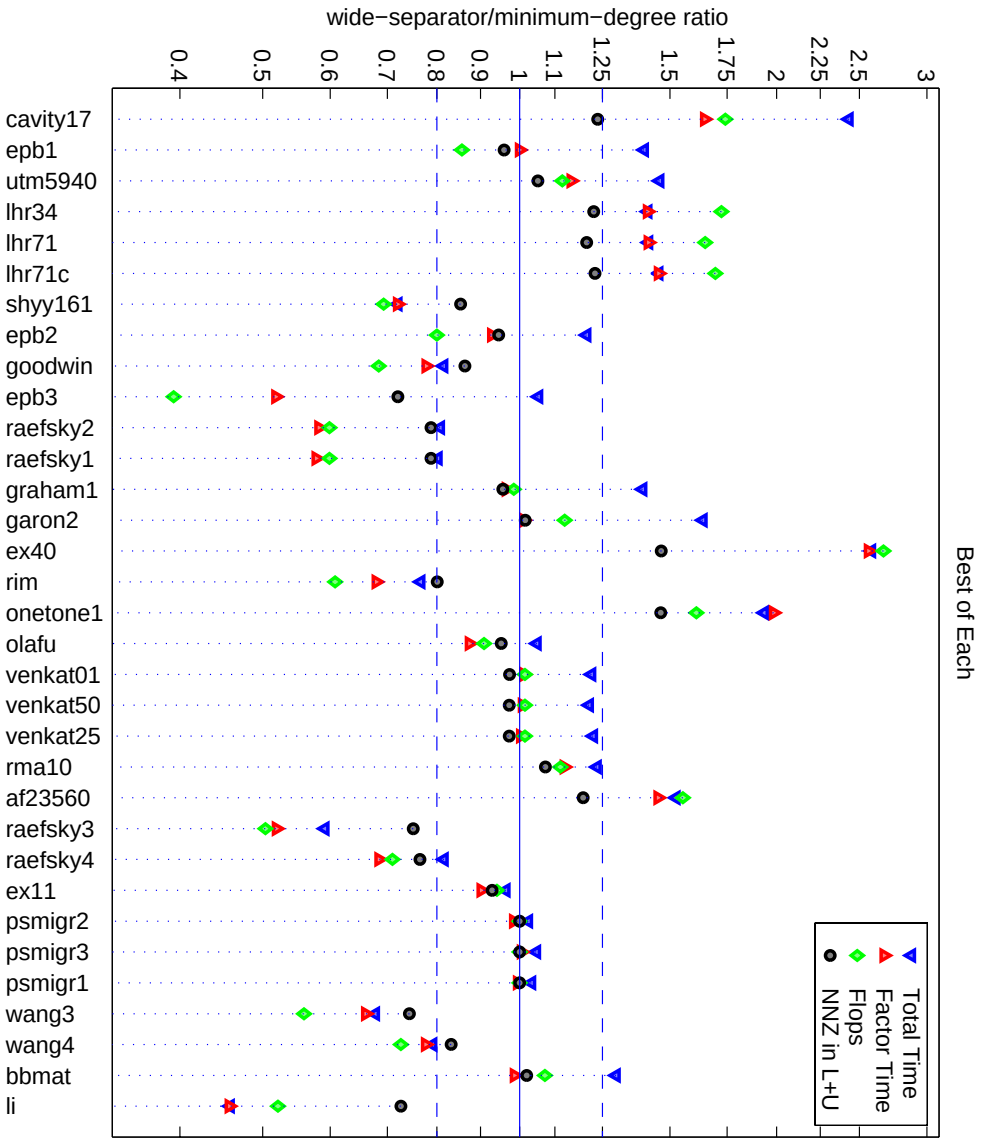


FIG. 6. *Best* WS *to best non-*WS *methods.*

33 matrices (but including 4 in the top 10). But there are no cases where WS orderings significantly reduce the factorization time but significantly increase the total times. Hence, ordering is a significant cost in WS-based factorization, but not a dominant one. We also note that even when a WS ordering reduces the factorization time but not the total time, it typically also reduces the size of the factors, which is often highly important (since it saves memory, reduces the occurrence of paging, and speeds up subsequent triangular solves).

4. Conclusions and discussion. Our main conclusion from this research is that hybrid wide-separator minimum-degree column orderings are effective. Wide-separator orderings are clearly superior to minimum-degree orderings alone on large

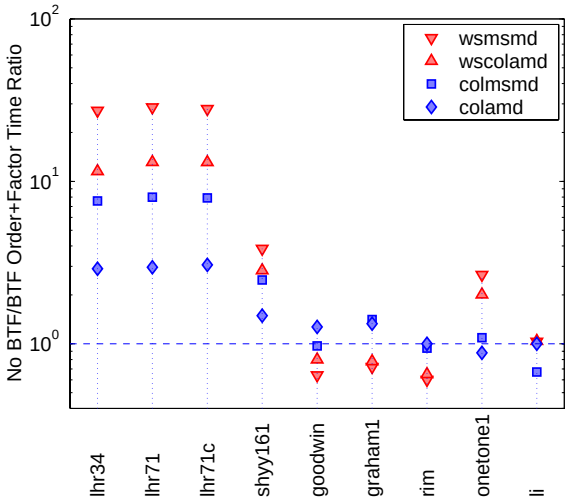


FIG. 7. *The effect of reduction to block triangular form on the factorization and ordering time. The graph shows, on a logarithmic scale, the ratio of the ordering+factorization times without reduction to block triangular form to the times with reduction. Data points above* 1 *indicate that the reduction saves time. The graph shows the ratios for four ordering methods. Only large matrices with nontrivial block triangular form are shown.*

2- and 3-dimensional meshes that require pivoting. On matrices obtained from a matrix collection, wide-separator orderings often substantially reduce the amount of time and storage required to factor a sparse matrix with partial pivoting, compared to column-minimum-degree orderings. They are more expensive to compute than minimum-degree orderings, but the expense is often more than paid for by reductions in time and storage during the factorization stage.

Wide-separator orderings, like other column orderings based on fill in the factors of A^TA, are robust but pessimistic. They are robust in the sense that they reduce worst-case fill. Optimistic column orderings that attempt to reduce the fill in the factors of $A^T + A$ tend to reduce fill better than pessimistic orderings when little or no pivoting occurs, but can lead to catastrophic fill when pivoting does occur. Further discussion of pessimistic versus optimistic orderings is beyond the scope of this paper.

The combined results of this paper and of an earlier paper [3] show that first permuting the matrix to block triangular form reduces the wide-separator ordering times and improves the quality of the ordering on some matrices.

This work can be extended in several directions. First, improving the performance of the ordering phase itself would be significant. This can be done by tuning the parameters of the ordering code (stopping the recursive bisection on fairly large subgraphs) or by improving the wide-separator algorithm itself. Second, one can try to improve the orderings by trying to derive smaller wide separators from a given conventional separator. Third, one can interleave the ordering and factorization in a way that widens separators only when necessary. That is, we would find a conventional separator S in G, recursively order G_1, and factor the columns corresponding to G_1. Once this phase is completed, we can widen the separator by adding to S the neighbors of vertices that were used as pivots. We now recursively order and factor the (shrunken) G_2.

Acknowledgments. Thanks to John Gilbert for telling us about wide-separator orderings. Thanks to John Gilbert and Bruce Hendrickson for helpful comments on an early draft of the paper. Thanks to Cleve Ashcraft for his encouragement, for numerous discussions concerning this research, and for his prompt response to our questions concerning SPOOLES. Thanks to the two anonymous referees, who pointed out a serious error in Theorem 2.2, suggested the counterexample that follows it, the tridiagonal-with-dense-row example in the introduction, and the inclusion of Figures 1 and 2.

REFERENCES

[1] C. Ashcraft and R. 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.

[2] P. Berman and G. Schnitger, *On the performance of the minimum degree ordering for Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 83–88.

[3] I. Brainman and S. Toledo, *Nested-dissection orderings for sparse LU with partial pivoting*, in Proceedings of the 2nd Conference on Numerical Analysis and Applications, Rousse, Bulgaria, 2000, Lecture Notes in Comput. Sci. 1988, L. Vulkuv, J. Waśniewski, and P. Yalamov, eds., Springer, Berlin, 2001, pp. 125–132.

[4] I. Brainman and S. Toledo, *Nested-dissection orderings for sparse LU with partial pivoting*, in Proceedings of the Tenth SIAM Conference on Parallel Processing for Scientific Computing, Norfolk, VA, 2001, CD-ROM, SIAM, Philadelphia, 2001.

[5] T. A. Davis and I. S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 140–158.

[6] T. A. Davis and I. S. Duff, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Trans. Math. Software, 25 (1999), pp. 1–19.

[7] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G. Ng, *A column approximate minimum degree ordering algorithm*, ACM Trans. Math. Software, submitted.

[8] 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.

[9] A. George and J. W. H. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Rev., 31 (1989), pp. 1–19.

[10] A. George and E. Ng, *On the complexity of sparse QR and LU factorization on finite-element matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 849–861.

[11] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.

[12] J. R. Gilbert, *Graph Separator Theorems and Sparse Gaussian Elimination*, Ph.D. thesis, Stanford University, 1980.

[13] J. R. Gilbert and E. Ng, *Predicting structure in nonsymmetric sparse matrix factorizations*, in Graph Theory and Sparse Matrix Computation, IMA Vol. Math. Appl. 56, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, New York, 1993.

[14] J. R. Gilbert and R. Schreiber, *Nested dissection with partial pivoting*, in Sparse Matrix Symposium 1982: Program and Abstracts, Fairfield Glade, TN, 1982, p. 61.

[15] S. I. Larimore, *An Approximate Minimum Degree Column Ordering Algorithm*, master's thesis, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 1998; also available as CISE Tech Report TR-98-016 at ftp://ftp.cise.ufl.edu/cis/tech-reports/tr98/tr98-016.ps.

[16] X. S. Li, *Sparse Gaussian Elimination on High Performance Computers*, Ph.D. thesis, Department of Computer Science, University of California, Berkeley, CA, 1996.

[17] A. Pothen and C.-J. Fan, *Computing the block triangular form of a sparse matrix*, ACM Trans. Math. Software, 16 (1990), pp. 303–324.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1013–1024
© 2002 Society for Industrial and Applied Mathematics

ERROR ESTIMATES FOR LINEAR COMPARTMENTAL SYSTEMS*

RUPERT LASSER† AND SEBASTIAN WALCHER‡

Abstract. We present and discuss error estimates for nonautonomous linear compartmental systems. The estimates are based on Kamke's comparison theorem for cooperative differential equations.

Key words. compartmental system, M-matrix, monotone dynamical system, ecotoxicology

AMS subject classifications. 34A30, 34A40, 34C12, 15A48

PII. S0895479800374522

Introduction. Compartmental models are abundant in medicine, physiology, ecology, and other disciplines of science. Frequently they represent simplified versions of real-life scenarios, but models of this type are very useful as first approximations. Moreover, if the underlying mechanisms are not well understood, or there are insufficient data, compartmental models may even be seen as the most adequate approach. Many examples and applications can be found in the classical monograph by Jacquez [5], the book by Anderson [1], and the recent book by Walter and Contreras [10]. Applications to fate modeling of chemicals and ecotoxicology can be found in Mackey et al. [8] and Hutzinger et al. [4]. It may be said that linear compartmental systems form the most important class in view of applications. Even though this may be considered a well-understood class, there is little knowledge about the properties of the solutions in the nonautonomous case. The purpose of this note is to contribute a better understanding of such systems.

A frequent problem with models of biological or ecological systems is rooted in the fact that parameters are not easy to determine, and often only very rough estimates are known. (For instance, estimates for partition coefficients in ecotoxicology models frequently differ by several orders of magnitude.) In such situations, standard sensitivity analysis—which, by design, tests a system's response to small parameter changes—is not an appropriate tool to investigate the effects of parameter variations. The purpose of this note is to show that linear compartmental systems are amenable to quite efficient error estimates. The underlying reason is that linear compartmental systems are cooperative systems in the sense of Kamke [7] and Walter [11], and for these systems Kamke's comparison theorem holds true. The main technical problem is that a direct application of Kamke's theorem may yield unsatisfactory results (e.g., the resulting estimates may grow exponentially although the solutions are bounded); therefore a different approach is necessary; see Theorem 2.1. The results are stated and proved for nonautonomous systems, but they address a problem that also occurs in the autonomous case, and the method is of course also applicable in this context. Several examples illustrate the method and show that it is capable of producing satisfactory results.

*Received by the editors June 26, 2000; accepted for publication (in revised form) by L. El Ghaoui November 28, 2001; published electronically April 10, 2002.
http://www.siam.org/journals/simax/23-4/37452.html

†Zentrum Mathematik, TU München, 80290 München, Germany and Institut für Biomathematik und Biometrie, GSF-Forschungszentrum für Umwelt und Gesundheit, Ingolstädter Landstr. 1, 85764 Neuherberg, Germany (lasser@gsf.de).

‡Lehrstuhl A für Mathematik, RWTH Aachen, 52056 Aachen, Germany (walcher@mathA.rwth-aachen.de).

1. Basic facts. In this section we introduce some notation and review a number of facts about compartmental systems. Proofs are included only if they are not directly accessible from other sources.

A linear compartmental system is, by definition, a differential equation,

$$\dot{x} = \begin{pmatrix} -r_1 - \sum_{j\neq 1} d_{j1} & d_{12} & \cdots & \cdots & d_{1n} \\ d_{21} & -r_2 - \sum_{j\neq 2} d_{j2} & d_{23} & \cdots & \vdots \\ \vdots & d_{32} & \cdots & \cdots & \vdots \\ \vdots & \vdots & & & d_{1,n-1} \\ d_{n1} & d_{n2} & \cdots & \cdots & -r_n - \sum_{j\neq n} d_{jn} \end{pmatrix} \cdot x + \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ \vdots \\ b_n \end{pmatrix}.$$

Here the r_i, d_{ij}, and b_i are nonnegative continuous functions on some interval (usually $[0, \infty)$). We abbreviate this system as

$$\dot{x} = A \cdot x + b,$$

and we call A a compartmental matrix, following the terminology of Jacquez and Simon [6].

The situation we want to investigate is as follows. We assume that there are estimates

$$\begin{aligned} 0 \leq \underline{d}_{ij} \leq d_{ij} \leq \overline{d}_{ij}\,, \\ 0 \leq \underline{r}_i \leq r_i \leq \overline{r}_i\,, \\ 0 \leq \underline{b}_i \leq b_i \leq \overline{b}_i \end{aligned}$$

for all i and j. The underlined and overlined quantities may themselves be (continuous) functions; in this case the inequalities are to be understood pointwise. We will use the notation introduced above throughout this paper. In applications, the available estimates are not necessarily "good," and the problem we want to address is determining what estimates can still be deduced for the solutions of the differential equation.

The theory of compartmental matrices with constant coefficients may be seen as part of the theory of nonnegative matrices, as was noted and elaborated by Jacquez and Simon [6]. (Negatives of compartmental matrices are known as M-matrices.) We recall some essential features. As usual, the notion $P \geq 0$, resp., $q \geq 0$, for real matrices or vectors is to be understood entrywise, and we speak of nonnegative matrices, resp., vectors. The notion of irreducibility carries over verbatim to compartmental matrices. A reducible compartmental matrix may, after a permutation of indices, be written in block triangular form, with irreducible blocks in the diagonal. It is immediate from the definitions that these blocks are themselves compartmental matrices.

PROPOSITION 1.1. *Let A be a nonzero compartmental matrix with constant coefficients. Then the following hold.*

(a) *There is a nonnegative matrix B, with spectral radius $\rho(B)$, and a real $s > 0$, $s \geq \rho(B)$, such that $A = B - sE$.*
(b) *If A is irreducible and some $r_j > 0$, then A is invertible.*
(c) *Let $A = B - sE$ as in part (a). Then*
  (c1) *$\rho(B) - s$ is an eigenvalue of A, and all eigenvalues of A have real parts $\leq \rho(B) - s$;*

(c2) *if A is irreducible, then $\rho(B) - s$ has multiplicity 1, and all other eigenvalues of A have real parts $< \rho(B) - s$.*

(d) *If A is invertible, and $b \geq 0$, then the unique solution of $Ax + b = 0$ is nonnegative.*

(e) *If A is invertible, then there exists v with positive entries such that all entries of Av are negative.*

Proof. All proofs follow from results in Berman and Plemmons [2, Chap. 6]; see also Jacquez and Simon [6]. Therefore we elaborate only the proofs of parts (a) and (b), which are not immediate from these sources. (Jacquez and Simon discuss only invertible compartmental matrices.)

(a) We show that $-A$ is an M-matrix in the sense of [2, Chap. 6]; by using criterion A_6 from Theorem 4.6 in [2, Chap. 6] for the transpose of A. Accordingly, we have to show that for each signature matrix $S = \operatorname{diag}(\epsilon_1, \ldots, \epsilon_n)$, with all $\epsilon_i \in \{1, -1\}$, there is a nonzero $x \geq 0$ such that $-SA^tSx \geq 0$. We may assume $S = \operatorname{diag}(1, \ldots, 1, -1, \ldots, -1)$, with the first $q \geq 1$ entries equal to 1. Then

$$-SA^tS = \begin{pmatrix} r_1 + \sum d_{j1} & -d_{21} & \cdots & -d_{q1} & \\ -d_{12} & & \vdots & & \\ \vdots & & & \vdots & * \\ \vdots & & & -d_{q-1,q} & \\ -d_{1,q} & \cdots & & r_q + \sum d_{jq} & \\ d_{1,q+1} & \cdots & & d_{q,q+1} & \\ \vdots & & & \vdots & * \\ d_{1n} & \cdots & & d_{qn} & \end{pmatrix},$$

and $x := (1, \ldots, 1, 0, \ldots, 0)^t$ satisfies

$$-SA^tSx = \left(r_1 + \sum_{j>q} d_{j1}, \ldots, r_q + \sum_{j>q} d_{jq}, \sum_{j\leq q} d_{j,q+1}, \ldots, \sum_{j\leq q} d_{jn} \right) \geq 0.$$

Now the assertion $-A = sE - B$ follows from the definition of an M-matrix in [2, Chap. 6].

(b) Take $v = (v_1, \ldots, v_n)^t$, with all $v_i > 0$. Since

$$(1, \ldots, 1) \cdot (-A) = (r_1, \ldots, r_n)$$

and some $r_j > 0$, one gets

$$(1, \ldots, 1) \cdot (-A) \cdot v \geq r_j v_j > 0.$$

Therefore $A \cdot v \neq 0$, and the assertion follows from [2, Chap. 6, Thm. 4.16(2)].

Part (c) follows from Perron–Frobenius.

(d) The inverse of $-A$ is nonnegative according to [2, Chap. 6, Thm. 2.3].

(e) This is criterion M_{34} in [2, Chap. 6, Thm. 2.3]. ∎

Remark 1.2. Suppose that A is an invertible compartmental matrix with constant entries. Using Proposition 1.1(e) for the transpose of A, we see that there is a row vector $(v_1, \ldots, v_n)$ such that all $v_i > 0$ and all entries of vA are negative. There are some cases when such a vector can be determined easily, as can be seen below:

(i) If all $r_i > 0$, then one may choose $v = (1, \ldots, 1)$.

(ii) If A is irreducible, then "the" positive left eigenvector of A (guaranteed by Perron–Frobenius) has the desired property.

Some of this may be saved for the nonautonomous case. For instance, if there is a constant $\rho > 0$ such that $r_i(t) \geq \rho$ for all t and i, one still has

$$(1, \ldots, 1) \cdot A \leq -\rho \cdot (1, \ldots, 1).$$

Likewise, it may be possible to find estimates in cases such as (ii) by employing positive left eigenvectors of suitable constant matrices.

Let us now turn to the linear compartmental system

$$(*) \qquad \dot{x} = A(t) \cdot x + b(t).$$

The nonnegative orthant of $\mathbf{R}^n$ is forward invariant for this differential equation, and we are exclusively interested in nonnegative solutions. (Boundedness of solutions is not automatic, even if b is bounded.)

Differential inequalities yield the following result, which is sometimes useful for finding upper estimates.

PROPOSITION 1.3. *Assume that there are $\alpha_1, \ldots, \alpha_n > 0$ and $\alpha_1^*, \ldots, \alpha_n^* > 0$ such that*

$$(\alpha_1, \ldots, \alpha_n) \cdot A \leq -(\alpha_1^*, \ldots, \alpha_n^*)$$

and define $\mu := \min\{\alpha_1^*/\alpha_1, \ldots, \alpha_n^*/\alpha_n\}$. Let $z(t)$ be a nonnegative solution of $(*)$. Then

$$z_i(t) \leq v(t)/\alpha_i \quad \text{for all } t \geq t_0,\ 1 \leq i \leq n,$$

with the solution $v(t)$ of

$$\dot{y} = -\mu \cdot y + (\alpha_1 b_1 + \cdots + \alpha_n b_n), \quad y(t_0) = \alpha_1 z_1(t_0) + \cdots + \alpha_n z_n(t_0).$$

In particular, if $b(t)$ is bounded, then every solution of $(*)$ is bounded.

Proof. The hypothesis implies

$$\frac{d}{dt}(\alpha_1 z_1 + \cdots + \alpha_n z_n) \leq -\mu \cdot (\alpha_1 z_1 + \cdots + \alpha_n z_n) + (\alpha_1 b_1 + \cdots + \alpha_n b_n);$$

thus $(\alpha_1 z_1 + \cdots + \alpha_n z_n) \leq v(t)$ from standard properties of scalar differential inequalities; see Walter [11]. □

The hypothesis of (1.3) is always satisfied for invertible compartmental matrices with constant entries, as Proposition 1.1(e) shows. Moreover, we then have a (possibly crude) upper estimate for the solutions. A variant of this strategy may be used to provide lower estimates.

Now we recall Kamke's comparison theorem for cooperative systems in the special context of linear cooperative systems. Generally, a differential equation $\dot{x}_i = f_i(t, x)$ $(1 \leq i \leq n)$, which is defined on a convex subset U of $\mathbf{R}^n$ with nonempty interior and C^1 right-hand side in x, is called cooperative on U if $\partial f_i/\partial x_j \geq 0$ whenever $i \neq j$. (There is an extensive qualitative theory of autonomous cooperative systems which was initiated by Hirsch [3]; see the monograph by Smith [9].) A linear differential equation $\dot{x} = P \cdot x + q$ is cooperative if and only if all off-diagonal entries of P are nonnegative. In this paper we will refer to such matrices as cooperative matrices. The specialization of Kamke's comparison theorem [7] to linear systems is as follows.

PROPOSITION 1.4. *Let $U \subseteq \mathbf{R}^n$ be convex, with nonempty interior, and suppose that the linear differential equations*

$$\dot{x} = P(t) \cdot x + q(t) \quad \textit{and} \quad \dot{x} = P^*(t) \cdot x + q^*(t)$$

are given on U, with t in some interval J. Assume that

(i) *$P(t)$ is cooperative for all t or $P^*(t)$ is cooperative for all t;*

(ii) *$P(t) \cdot x + q(t) \geq P^*(t) \cdot x + q^*(t)$ for all $t \in J$ and all $x \in U$.*

Let $t_0 \in J$, and let $u(t)$, respectively, $u^*(t)$, be a solution of the first, respectively, the second, equation in U such that $u(t_0) \geq u^*(t_0)$. Then $u(t) \geq u^*(t)$ for all $t \geq t_0$.

This result has an obvious application to linear compartmental systems on the nonnegative orthant. (Clearly compartmental systems are cooperative.) Given A, b as above, define

$$\underline{A} = \begin{pmatrix} -\overline{r}_1 - \sum_{j\neq 1} \overline{d}_{j1} & \underline{d}_{12} & \cdots & \cdots & \underline{d}_{1n} \\ \underline{d}_{21} & -\overline{r}_2 - \sum_{j\neq 2} \overline{d}_{j2} & \underline{d}_{23} & \cdots & \vdots \\ \vdots & \underline{d}_{32} & \cdots & \cdots & \vdots \\ \vdots & \vdots & & & \underline{d}_{1,n-1} \\ \underline{d}_{n1} & \underline{d}_{n2} & \cdots & \cdots & -\overline{r}_n - \sum_{j\neq n} \overline{d}_{jn} \end{pmatrix}, \ \underline{b} = \begin{pmatrix} \underline{b}_1 \\ \vdots \\ \vdots \\ \vdots \\ \underline{b}_n \end{pmatrix},$$

$$\overline{A} = \begin{pmatrix} -\underline{r}_1 - \sum_{j\neq 1} \underline{d}_{j1} & \overline{d}_{12} & \cdots & \cdots & \overline{d}_{1n} \\ \overline{d}_{21} & -\underline{r}_2 - \sum_{j\neq 2} \underline{d}_{j2} & \overline{d}_{23} & \cdots & \vdots \\ \vdots & \overline{d}_{32} & \cdots & \cdots & \vdots \\ \vdots & \vdots & & & \overline{d}_{1,n-1} \\ \overline{d}_{n1} & \overline{d}_{n2} & \cdots & \cdots & -\underline{r}_n - \sum_{j\neq n} \underline{d}_{jn} \end{pmatrix}, \ \overline{b} = \begin{pmatrix} \overline{b}_1 \\ \vdots \\ \vdots \\ \vdots \\ \overline{b}_n \end{pmatrix}.$$

Then, by construction,

$$\underline{A} \cdot x + \underline{b} \leq A \cdot x + b \leq \overline{A} \cdot x + \overline{b}$$

for all nonnegative x. Thus we have the following.

COROLLARY 1.5. *If $\underline{u}$, u, and $\overline{u}$, respectively, are nonnegative solutions of $\dot{x} = \underline{A} \cdot x + \underline{b}$, $\dot{x} = A \cdot x + b$, and $\dot{x} = \overline{A} \cdot x + \overline{b}$, with $\underline{u}(t_0) \leq u(t_0) \leq \overline{u}(t_0)$, then $\underline{u}(t) \leq u(t) \leq \overline{u}(t)$ for all $t \geq t_0$.*

Note that $\dot{x} = \underline{A} \cdot x + \underline{b}$ is again a compartmental system. On the other hand, $\dot{x} = \overline{A} \cdot x + \overline{b}$ is in general no longer a compartmental system and will in general have unbounded solutions in the nonnegative orthant. (For instance, there may be an index j such that $r_j = 0$ and $\underline{d}_{ij} < \overline{d}_{ij}$ for some i. Then $\overline{A}$ is not a compartmental matrix.) If all solutions of $\dot{x} = A \cdot x + b$ are bounded, then $\dot{x} = \overline{A} \cdot x + \overline{b}$ will not provide sensible estimates for large times. Therefore, more work is needed here to obtain upper estimates. For arbitrary cooperative systems there are improvements of Kamke's theorem by Walter [11], which generally yield sharper estimates but do not resolve the problem here. For linear systems there is another, direct, approach, which we will introduce in the next section.

2. Estimates for linear cooperative systems.

THEOREM 2.1. *Let P and P^* be cooperative matrices with $P \geq P^*$, let $\overline{P}$ be such that $\overline{P} \geq P$, and furthermore let q, $q^* \geq 0$. Let u and u^* be nonnegative solutions of $\dot{x} = P \cdot x + q$ and $\dot{x} = P^* \cdot x + q^*$, respectively, for $t \geq 0$. Moreover, let $t_0 \geq 0$ and let z, z^* be functions such that $u(t) \leq z(t)$ and $u^*(t) \leq z^*(t)$ for all $t \geq t_0$. Then the following hold:*

(a) *For $t \geq t_0$ one has $u(t) - u^*(t) \geq v(t)$, where $v(t)$ solves any of the initial value problems*

$$\begin{aligned} \dot{x} &= P \cdot x + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0); \\ \dot{x} &= P^* \cdot x + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0). \end{aligned}$$

(b) *For $t \geq t_0$ one has $u(t) - u^*(t) \leq w(t)$, where $w(t)$ solves any of the initial value problems*

$$\begin{aligned} \dot{x} &= P \cdot x + (P - P^*) \cdot z^* + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0); \\ \dot{x} &= P^* \cdot x + (P - P^*) \cdot z + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0); \\ \dot{x} &= P \cdot x + (\overline{P} - P^*) \cdot z^* + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0); \\ \dot{x} &= P^* \cdot x + (\overline{P} - P^*) \cdot z + (q - q^*), \quad x(t_0) = u(t_0) - u^*(t_0). \end{aligned}$$

Proof. We have

$$\frac{d}{dt}(u - u^*) = P \cdot u - P^* \cdot u^* + q - q^* = \left\{ \begin{array}{l} P \cdot (u - u^*) + (P - P^*) \cdot u^* + q - q^* \\ P^* \cdot (u - u^*) + (P - P^*) \cdot u + q - q^* \end{array} \right\}$$

and both P and P^* are cooperative. Since $P \geq P^*$, we get $0 \leq (P - P^*) \cdot u^* \leq (P - P^*) \cdot z^*$, and therefore

$$P \cdot (u - u^*) + q - q^* \leq \frac{d}{dt}(u - u^*) \leq P \cdot (u - u^*) + (P - P^*) \cdot z^* + q - q^*.$$

The first assertions of parts (a) and (b) now follow directly from (1.4). The remaining claims are proven in a similar manner. ∎

Note that all the systems given in (2.1) are themselves cooperative, and therefore Kamke's theorem may be used once more.

Let us briefly discuss the quality of the estimates thus obtained. Concerning lower estimates, in case $q \geq q^*$ part (a) will yield a result which is no worse than direct application of Kamke's theorem. Concerning part (b), the crucial question is whether an initial estimate $u \leq z$ will actually be improved. As it turns out, much depends on the proper choice of q^*. We illustrate this in a very elementary situation.

EXAMPLE 2.2. *Consider one-dimensional equations, with $P = -1$, $P^* = -2$, $q = 3$, and q^* to be determined later. Let u, u^* with $u(0) = u^*(0) = 0$; thus $u(t) = 3 - 3\exp(-t)$, and $u^*(t) = q^*/2 - q^*/2 \cdot \exp(-2t)$.*

Take some constant upper estimate z for u (any $z \geq 3$ works), and use the second equation in Theorem 2.1(b) *to compare solutions. Thus $\dot{v} = -2v + (3 + z - q^*)$, $v(0) = 0$, and choosing $q^* = 3 + z$ yields $v = 0$. Hence, if $u^*(t)$ solves $\dot{x} = -2x + (3 + z)$, $x(0) = 0$, then $u \leq u^*$. The computation yields $u(t) \leq (3 + z)/2 \cdot (1 - \exp(-2t)) \leq (3 + z)/2$, hence an improved upper estimate for u. Note that the procedure can be iterated.*

The strategy used in this simple example can be adapted to more interesting scenarios.

Remark 2.3. Given a linear compartmental system $\dot{x} = A \cdot x + b$, one may choose $P = A$, $P^* = \underline{A}$ and compare solutions with solutions of a system

$$\dot{x} = \underline{A} \cdot x + b^*,$$

with b^* to be determined. If an initial (bounded) upper estimate z for a solution u of $\dot{x} = A \cdot x + b$ is known (for instance, with the help of Proposition 1.3), then Theorem 2.1(b), together with Kamke's theorem, yields $u(t) - u^*(t) \le w(t)$, with $w(t)$ solving

$$\dot{x} = \underline{A} \cdot x + (\overline{A} - \underline{A}) \cdot z + (\overline{b} - b^*); \quad w(t_0) = u(t_0) - u^*(t_0).$$

Since $\underline{A}$ is a compartmental matrix, one will obtain bounded estimates by choosing b^* such that

$$(\overline{A} - \underline{A}) \cdot z + \overline{b} - b^* \ge 0.$$

The special choice

$$b^* = (\overline{A} - \underline{A}) \cdot z + \overline{b}$$

yields the equation $\dot{x} = \underline{A} \cdot x$ for w, and thus $\lim_{t\to\infty} w(t) = 0$ under mild additional conditions. (For instance, this holds whenever $\underline{A}$ has constant entries and is invertible.) In particular, $w = 0$ if $u(t_0) = u^*(t_0)$, and then $u \le u^*$ follows.

3. Examples. We first discuss a nontrivial example which is not directly related to an application, but it will illustrate how the results of the previous section can be applied.

3.1. A two-compartment model. Consider the compartmental system with

$$A = \begin{pmatrix} -r - d_{21} & d_{12} \\ d_{21} & -d_{12} \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 + \beta \end{pmatrix},$$

where $r(t) = 3 + \rho(t)$, $d_{21}(t) = 4 + \theta_{21}(t)$, $d_{12}(t) = 3 + \theta_{12}(t)$. We assume $0 \le \theta_{12} \le 2$, $0 \le \theta_{21} \le 1$, $0 \le \rho \le 1$, and $0 \le \beta \le 9$.

In this model, inflow goes exclusively to compartment 2, and loss or degradation of material occurs only in compartment 1. The rates are given only within certain error bounds, which have the same order of magnitude as the constant approximation. Therefore, the errors are not "small." In particular, the inflow rate is permitted to vary over a wide range. Lower and upper estimates are

$$\underline{A} = \begin{pmatrix} -9 & 3 \\ 4 & -5 \end{pmatrix}, \quad \overline{A} = \begin{pmatrix} -7 & 5 \\ 5 & -3 \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \overline{b} = \begin{pmatrix} 0 \\ 10 \end{pmatrix}.$$

We will discuss the solution $u(t)$ with $u(0) = 0$, and in particular its long-term behavior. Since $\overline{A}$ has a positive eigenvalue, a direct application of Proposition 1.4 will not produce bounded estimates.

(a) Use Proposition 1.3 to find an initial upper estimate:

$$\begin{aligned} \frac{d}{dt}(2x_1 + 3x_2) &= -(2x_1 + 3x_2) - 2\rho x_1 + \theta_{21} x_1 - \theta_{12} x_2 + 3(1 + \beta) \\ &\le -(2x_1 + 3x_2) + \theta_{21}/2 \cdot (2x_1 + 3x_2) + 30 \\ &\le -1/2 \cdot (2x_1 + 3x_2) + 30, \end{aligned}$$

and thus

$$2u_1 + 3u_2 \leq 60(1 - \exp(-t/2)) \leq 60.$$

This yields

$$u_1(t) \leq z_1 := 30 \quad \text{and} \quad u_2(t) \leq z_2 := 20.$$

(The choice of linear form is motivated by the observation that $(2, 3)$ is a left eigenvector of A in case $\rho = \theta_{21} = \theta_{12} = 0$. As noted in section 1, finding such an initial upper estimate may be problematic; here it works because of the bounds on ρ and the θ_{ij}.)

(b) Kamke's theorem (Proposition 1.4) provides as a lower estimate the solution of $\dot{x} = \underline{A} \cdot x + \underline{b}$, $x(0) = 0$. The solution of this equation approaches the stationary point $\frac{1}{11}(1, 3)^t$.

(c) Now use the initial upper estimate z and Theorem 2.1(b) to obtain improved upper estimates.

(i) If $w(t)$ solves

$$\dot{x} = \underline{A} \cdot x + (\overline{A} - \underline{A}) \cdot z + (\overline{b} - b^*), \quad x(0) = 0,$$

and $u^*(t)$ solves

$$\dot{x} = \underline{A} \cdot x + b^*, \quad x(0) = 0,$$

then $u - u^* \leq w$ for all $t \geq 0$. Following Remark 2.3, choose

$$\begin{aligned} b^* &= (\overline{A} - \underline{A}) \cdot z + \overline{b} \\ &= \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 30 \\ 20 \end{pmatrix} + \begin{pmatrix} 0 \\ 10 \end{pmatrix} = \begin{pmatrix} 100 \\ 80 \end{pmatrix} \end{aligned}$$

to ensure $u \leq u^*$.

Here we focus on the behavior for large times. For $t \to \infty$, u^* approaches the stationary point

$$\underline{A}^{-1} \cdot b^* = \frac{20}{33} \cdot \begin{pmatrix} 37 \\ 56 \end{pmatrix} < \begin{pmatrix} 22.5 \\ 34 \end{pmatrix},$$

and hence $u(t) \leq \binom{22.5}{34}$ for all sufficiently large t. Combining estimates, one has

$$u(t) \leq \begin{pmatrix} 22.5 \\ 20 \end{pmatrix}$$

for sufficiently large t.

(ii) Repeat the procedure with $z = \binom{22.5}{20}$. Then

$$b^* = \begin{pmatrix} 85 \\ 72.5 \end{pmatrix} \quad \text{and} \quad u(t) < \begin{pmatrix} 19.5 \\ 20 \end{pmatrix}$$

for sufficiently large t. A few more repetitions yield

$$u(t) \leq \begin{pmatrix} 17.8 \\ 20 \end{pmatrix}$$

for sufficiently large t.

(d) Obviously, the iteration carried out above to determine estimates in the limiting case $t \to \infty$ is related to the linear difference equation with matrix

$$Q := \underline{A}^{-1} \cdot (\overline{A} - \underline{A})$$

and constant term $c := \underline{A}^{-1} \cdot \overline{b}$. (A similar observation holds generally.) In the given situation, the eigenvalues of Q have modulus < 1 and the iteration converges, albeit slowly. But it should be noted that pursuing this "pure" strategy is less effective than combining estimates, as above. (Naturally, the combined strategy itself leads to a difference equation.)

(e) The error in these estimates is of the same order of magnitude as the errors in the right-hand side of the equation, and this is quite satisfactory concerning estimates for large times. An elementary computation shows that among the autonomous systems in the class under consideration, the limit is always bounded above by $\frac{1}{3}\binom{10}{30}$, and thus the estimates are reasonable, if not perfect.

(f) Finally, consider the "degradation problem"

$$\dot{x} = Ax, \quad x(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

as an illustration of the method with nonconstant estimates. Just as in part (a) the solution u satisfies

$$\frac{d}{dt}(2u_1 + 3u_2) \leq -1/2 \cdot (2u_1 + 3u_2),$$

and with $(2u_1 + 3u_2)(0) = 5$ this yields the initial upper estimates

$$u_1(t) \leq z_1(t) := 5/2 \cdot \exp(-t/2), \quad u_2(t) \leq z_2(t) := 5/3 \cdot \exp(-t/2).$$

Now we use the fourth equation in Theorem 2.1(b) with $q^* = 0$ and $u^*(0) = 0$ (thus u^* is identically zero) to see that u is majorized by the solution of

$$\dot{x} = \underline{A}x + \left(\overline{A} - \underline{A}\right) z, \quad x(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Solving the latter equation shows that the new estimates are indeed better than the initial ones and that, for instance,

$$u_1(1) < 0.382, \quad u_1(2) < 0.229, \quad u_1(3) < 0.139;$$

$$u_2(1) < 0.571, \quad u_2(2) < 0.340, \quad u_2(3) < 0.206.$$

Compare this, for example, with the solution w of $\dot{x} = \tilde{A}x$ with the same initial value, where $\tilde{A}$ is characterized by $\rho = 0$, $\theta_{21} = 1$, $\theta_{12} = 2$. Then

$$w_1(1) = 0.230, \quad w_1(2) = 0.085, \quad w_1(3) = 0.031;$$

$$w_2(1) = 0.460, \quad w_2(2) = 0.170, \quad w_2(3) = 0.062.$$

Once again, the estimates are not perfect but reasonably good. Repeating the procedure in this case, with z replaced by the improved estimates found in the first step, turns out to yield upper estimates that are actually worse. (A look at the structure of the closed form solution shows that such a phenomenon may occur.) Finally, it should be remarked that making direct use of Kamke's theorem here does not lead to meaningful estimates.

Next comes an example from "real life."

3.2. An example from ecotoxicology. The system to be discussed here is derived from a model describing the fate of a chemical substance (nonylphenol) in a small pond, as part of an interdisciplinary ecotoxicology project. The model is used for simulations to determine the amount of chemical in the different compartments of the pond system. Its principal purpose is to help the experimenters in understanding the distribution of the substance in the system, and therefore the task includes detecting inadequacies in the model itself, with subsequent corrections. (For instance, there may be a priori unknown effects which contribute to degradation of the substance.) After calibration, a simulation should produce reliable estimates of the substance's concentration in various compartments. Since this is a typical situation where certain parameters (like partition coefficients) are known only within large margins of error, and others (such as temperature) may vary strongly with time, error estimates are crucial to assessing the quality of the simulation results.

(a) The full model will be discussed elsewhere. Here we present a simplified version, but it should be emphasized that this version reflects a real system: The "pond" in question is a vessel which contains about 200 liters of lake water above 40 liters of lake sediment. The sediment layer has a thickness of 7.5 centimeters. The vessel itself is immersed in a water pool. The chemical is added to the water at a constant rate. In the model to be discussed here, there are only four compartments (water and three sediment compartments) for the sake of simplicity. Moreover, we discuss an idealized situation where the chemical leaves the system only by evaporation from water to air, and there is no degradation in any compartment. The upper sediment compartment has an organic carbon content (TOC) of 3%, while the water and the other sediment compartments contain no organic carbon at all. This idealized situation has been chosen in part to make error estimates harder to obtain, but it is also a reasonable strategy to initially work with a model that allows no degradation within the compartments: determining a priori the rates of such processes may be a serious problem.

The numbers x_i, $i = 1, \dots, 4$, represent the mass of the chemical in the compartments, in milligrams. The unit of time is one day. Compartment 1 is water, compartments 2 through 4 are sediment layers of increasing depth. The differential equation is

$$\dot{x} = A \cdot x + b,$$

with

$$A = \begin{pmatrix} -(5r_1 + 6) \cdot 10^{-3} & 2K^{-1} & 0 & 0 \\ -6 \cdot 10^{-3} & -4K^{-1} & 0.33 & 0 \\ 0 & 2K^{-1} & -0.53 & 0.2 \\ 0 & 0 & 0.2 & -0.2 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Estimates for the organic carbon-water partition coefficient K in the literature vary between 10^4 and 10^6, and the rate r_1, which describes evaporation, varies between 0.5 and 1.6, corresponding to a variation in temperature between 10 and 30 degrees centigrade. For reasons of convenience, we have included some idealizations in the model; for instance, we assume that all the other quantities are known with negligibly small error. But these idealizations do not significantly alter the picture. The experimenters are interested in the distribution of the chemical over the duration of a few months. In this example we choose a time span of one hundred days, and we assume that initially there is no chemical substance in the system. Thus we investigate the

solution $u(t)$ of

$$\dot{x} = A \cdot x + b, \quad x(0) = 0.$$

(b) Using the terminology introduced in section 2, we have

$$\underline{A} = \begin{pmatrix} -1.4 \cdot 10^{-2} & 2 \cdot 10^{-6} & 0 & 0 \\ 0.6 \cdot 10^{-2} & -4 \cdot 10^{-4} & 0.33 & 0 \\ 0 & 2 \cdot 10^{-6} & -0.53 & 0.2 \\ 0 & 0 & 0.2 & -0.2 \end{pmatrix},$$

$$\overline{A} = \begin{pmatrix} -0.9 \cdot 10^{-2} & 2 \cdot 10^{-4} & 0 & 0 \\ 0.6 \cdot 10^{-2} & -4 \cdot 10^{-6} & 0.33 & 0 \\ 0 & 2 \cdot 10^{-4} & -0.53 & 0.2 \\ 0 & 0 & 0.2 & -0.2 \end{pmatrix},$$

and $\underline{b} = \overline{b} = b$.

To obtain a lower estimate, we solve the equation

$$\dot{x} = \underline{A} \cdot x + b, \quad x(0) = 0.$$

At time $t = 100$ (days) this yields

$$u_1(100) > 538, \quad u_2(100) > 195, \quad u_3(100) > 10^{-3}, \quad u_4(100) > 10^{-3}.$$

(c) In the given scenario, and for the given duration of time, it turns out that a direct application of Kamke's theorem, i.e., solving

$$\dot{x} = \overline{A} \cdot x + b, \quad x(0) = 0,$$

yields sharper upper bounds than using Theorem 2.1(b), mostly since no good initial upper estimate is available. (There is a natural candidate for such an estimate: Take $z = 10^3 \cdot (1, 1, 1, 1)^t$; this reflects the fact that each compartment contains at most the mass that was brought into the whole system during the given time span. But starting with this and using Theorem 2.1(b) yields weaker estimates than proceeding directly.) The estimates we obtain are

$$u_1(100) < 661, \quad u_2(100) < 229, \quad u_3(100) < 0.13, \quad u_4(100) < 0.12.$$

Thus we have reasonable upper and lower estimates in this situation.

(d) The strategy from Theorem 2.1(b) is also useful here, since the matrix $\overline{A}$ has a positive eigenvalue (approximately equal to $3 \cdot 10^{-4}$), and therefore solutions grow exponentially with time. Let us consider the equation $\dot{x} = Ax$, with initial value $(600, 200, 0.1, 0.1)$ at $t = 0$. This may describe the case when the substance has been added at a constant rate 10 for 100 days (compare the data from parts (b) and (c)), and then the system is left to itself. We will proceed as in (e) of section 3.1.

The problematic part is to find an initial upper estimate. If we take v as a left eigenvector of A with parameters $K = 10^6$, and $r = 0.5$, then applying Proposition 1.3 yields $\mu = 6.67 \cdot 10^{-7}$ and the initial upper estimate

$$z(t) = 300 \exp(-\mu t) \begin{pmatrix} 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}.$$

Since this is the best possible μ for one particular case, one cannot expect better general estimates. Incidentally, v is close (but not equal) to $(2, 3, 3, 3)$.

Solving $\dot{x} = \underline{A}x + (\overline{A} - \underline{A})z$ with the given initial values produces improved upper estimates. For instance, the upper estimate for u_1 decreases fairly quickly, with $u_1(50) < 484$ and $u_1(100) < 396$. In contrast, the estimate for u_2 is increasing for t between 0 and 100, with $u_2(100) < 506$. (Such a phenomenon also occurs in the pond system: Even if there is no more addition of chemical substance to the water compartment, the concentration in the sediment layers will still increase for some time.) The mass in the third and fourth compartments also increases for t between 0 and 100 and seems to approach a plateau near 0.37. Of course, in the long range all the masses (and estimates) will approach zero. Comparing the estimates with simulations shows, once again, that they are quite reasonable. The principal value of such estimates is that they provide verifiable (or refutable) predictions that can be tested against experimental data, and thus they are useful in the model validation process.

REFERENCES

[1] D.H. Anderson, *Compartmental Modeling and Tracer Kinetics*, Lecture Notes in Biomath. 50, Springer-Verlag, New York, 1983.

[2] A. Berman and R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.

[3] M.W. Hirsch, *Systems of differential equations which are competitive or cooperative.* I. *Limit sets*, SIAM J. Math. Anal., 13 (1982), pp. 167–179.

[4] O. Hutzinger, K.-W. Schramm, A. Fischer, K.-U. Goss, and J. Klasmeier, *Verteilung von Umweltchemikalien in einem standardisierten ökologischen System—Vergleich und Optimierung von Rechenmodellen anhand von Ergebnissen aus Experimenten*, Forschungsbericht Umweltchemikalien/Schadstoffwirkung 10602067, Universität Bayreuth, Bayreuth, Germany, 1991.

[5] J.A. Jacquez, *Compartmental Analysis in Biology and Medicine*, 3rd ed., Biomedware, Ann Arbor, MI, 1996.

[6] J.A. Jacquez and C.P. Simon, *Qualitative theory of compartmental systems*, SIAM Rev., 35 (1993), pp. 43–79.

[7] E. Kamke, *Zur Theorie der Systeme gewöhnlicher Differentialgleichungen* II, Acta Math., 58 (1932), pp. 57–85.

[8] D. Mackay, S. Paterson, B. Cheung, and W. Nealy, *Evaluation of the environmental behavior of chemicals with a level* III *fugacity model*, Chemosphere, 14 (1985), pp. 335–375.

[9] H.L. Smith, *Monotone Dynamical Systems*, AMS, Providence, RI, 1995.

[10] G.G. Walter and M. Contreras, *Compartmental Modeling with Networks*, Birkhäuser Boston, Boston, 1999.

[11] W. Walter, *Differential and Integral Inequalities*, Springer-Verlag, Berlin, 1970.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1025–1044
© 2002 Society for Industrial and Applied Mathematics

THE CENTROID DECOMPOSITION: RELATIONSHIPS BETWEEN DISCRETE VARIATIONAL DECOMPOSITIONS AND SVDs*

MOODY T. CHU† AND ROBERT E. FUNDERLIC‡

Abstract. The centroid decomposition, an approximation for the singular value decomposition (SVD), has a long history among the statistics/psychometrics community for factor analysis research. We revisit the centroid method in its original context of factor analysis and then adapt it to other than a covariance matrix. The centroid method can be cast as an $\mathcal{O}(n)$-step ascent method on a hypercube. It is shown empirically that the centroid decomposition provides a measurement of second order statistical information of the original data in the direction of the corresponding left centroid vectors. One major purpose of this work is to show fundamental relationships between the singular value, centroid, and semidiscrete decompositions. This unifies an entire class of truncated SVD approximations. Applications include semantic indexing in information retrieval.

Key words. data matrix, loading matrix, scoring matrix, indexing matrix, factor analysis, centroid method, singular value decomposition, low rank approximation, semidiscrete decomposition, centroid decomposition, low rank decompositions, integer programming

AMS subject classifications. 15A21, 65F30, 62H25, 15A23, 68Q25

PII. S0895479800382555

1. Introduction.

We first review factor analysis [5, 9, 7] in the terms used by the applied statistics/psychometrics (AS/P) community with the notation of numerical linear algebra. This provides a setting to show how the centroid method developed as an approximate singular value decomposition (SVD). Our work was motivated by a recent article [11] and correspondence from Lawrence Hubert drawing our attention to the application of SVDs in the AS/P context. In particular we have found Horst's [9] description of the centroid decomposition proposed in the AS/P literature quite illuminating. The use of the SVD or ideas associated with it has a rich history [7] in the AS/P community dating back at least to Pearson [15] in 1901. Stewart's scholarly historical treatise [16] has traced the early history of the SVD back to Beltrami in 1873 and Jordan in 1874. Within the numerical linear algebra (NLA) community, besides Hotelling's work [10] and that of Eckert and Young [6], there seems little awareness of the AS/P work. The AS/P community generally considers Thurston's 1931 paper [17] as being the most complete description of the centroid method. In point of fact the centroid method was used in 1917 by Burt [2]. So what turned out to be an approximation for the SVD had its beginnings before there was widespread knowledge of the SVD itself.

We begin in section 2 with the factor analysis setting, providing a brief but practical background for further understanding of the underlying matrix decompositions. This should unify the differences in vocabulary and notation used by the AS/P and NLA communities. The classical Wedderburn rank reduction formula has been used by the AS/P community at least since the early 1940s. In section 3 we show how they

*Received by the editors December 14, 2000; accepted for publication (in revised form) by L. Eldén October 18, 2001; published electronically April 10, 2002.
http://www.siam.org/journals/simax/23-4/38255.html
†Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (chu@math.ncsu.edu). This author's research was supported in part by the National Science Foundation under grants DMS-9803759 and DMS-0073056.
‡Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 (ref@csc.ncsu.edu).

connect the rank reduction formula with the centroid method, which provides insight into the reduced matrix. What they call centroid factors are indeed the centroids of a sequence of the orthogonally reduced loading matrices. Section 4 further develops the centroid method with the necessary modifications for the reduction of the covariance matrix. Section 5 gives the details of the centroid algorithm along with an ascent hypercube description, proof of convergence, and computational complexity. Section 6 provides a general stochastic treatment of the truncated SVD and thereby the statistical soundness of the centroid decomposition. Section 7 compares the similar yet different setups between the centroid method and some latent semantic indexing techniques used in data mining. Section 8 discusses a modified centroid algorithm that does not require the explicit formation of the product moment or covariance matrix. Finally, in section 9 we use the SVD triad of variational formulations to unify a class of approximations to the SVD, including the data retrieval semidiscrete decomposition (SDD) and the centroid decomposition.

2. The factor analysis setting. An indispensable task in almost every discipline of science is the analysis of data in search of relationships between sets of externally caused and internal variables. Such a task has become especially important in this era of information and digital technologies, when massive amounts of data are being generated at almost all levels of applications. In many situations, the digitized information is gathered and stored as a data matrix. Quite often the data observed from complex phenomena represent the integrated result of several interrelated variables acting together. When these variables are less precisely defined, it becomes important to distinguish which variable is related to which and how the variables are related before deductive sciences can further be applied. Toward that end, factor analysis is a class of procedures that can help identify and test what constructs might be used to explain the interrelationships among the variables.

Let $Y = [y_{ij}] \in \mathbb{R}^{n\times\ell}$ denote the matrix of observed data. One of the main applications of factor analysis is to analyze relationships between questions on tests. Thus we will use here, as is done for almost any application, y_{ij} to represent, in a broad sense, the *standard score* of variable i on entity j. By a standard score we mean that a raw score has been normalized to have mean 0 and standard deviation 1. The matrix

$$R := \frac{1}{\ell} YY^T, \tag{2.1}$$

therefore, represents the correlation matrix of all n variables. Note that $r_{ii} = 1$ and $|r_{ij}| \le 1$ for all $i, j = 1, \ldots, n$.

In a linear model, it is assumed that the score y_{ij} is a linearly weighted score of entity j on several factors. That is, we assume

$$Y = AF, \tag{2.2}$$

where $A = [a_{ik}] \in \mathbb{R}^{n\times m}$ is a matrix with a_{ik} denoting loadings of variable i on factor k, and $F = [f_{kj}] \in \mathbb{R}^{m\times\ell}$ with f_{kj} denoting the score of factor k on entity j. To better grasp the notion of linear modeling in (2.2), readers might want to think, for example, that each of the ℓ columns of the observed matrix Y represents the transcript of a college student (an entity) at his/her freshman year on n fixed subjects (the variables), e.g., calculus, English, chemistry, and so on. It is generally believed that a college freshman's academic performance depends on a number of factors, including, for instance, family social status, finances, high school GPA, cultural background, and

so on. Upon entering the college, each student could be asked to fill out a questionnaire inquiring about these factors of his/her background. In turn, individual responses to those factors are translated into scores and placed in the corresponding column of the scoring matrix F. What is not clear to the educators/administrators is how to choose the factors to compose the questionnaire or how each of the chosen factors would be weighted (the loadings) to reflect the effect on each particular subject. In practice, we usually do not have a priori knowledge about the number and character of underlying factors in A. Sometimes we do not even know the factor scores in F. Only the data matrix Y is observable. Explaining the complex phenomena observed in Y with the help of a minimal number of factors extracted from the data matrix is the primary and most important goal of factor analysis.

It is customary to assume that all sets of factors being considered are uncorrelated with each other. If we further assume, similar to Y, that the scores in F for each factor are normalized, then it is true that

$$\frac{1}{\ell} F F^T = I_m, \tag{2.3}$$

where I_m stands for the identity matrix in $\mathbb{R}^{m\times m}$. It follows that the correlation matrix R can be expressed directly in terms of the loading matrix A, i.e.,

$$R = AA^T. \tag{2.4}$$

Factor extraction now becomes a problem of decomposing the correlation matrix R into the product AA^T using as few factors as possible.

As a whole, the ith row of A may be interpreted as how the data variable i is weighted across the list of current factors. If the sum of squares of this row, called the *communality* of variable i, is small, it suggests that this specific variable is of little consequence to the current factors. On the other hand, the kth column of A may be interpreted as correlations of the data variables with that particular kth factor. Those data variables with high factor loadings are considered to be more like the factor in some sense, and those with zero or near-zero loadings are treated as being unlike the factor. The quality of this likelihood, which we call the *significance* of the corresponding factor, is measured by the norm of the kth column of A. One basic idea in factor analysis is to rewrite the loadings of variables over some newly selected factors so as to manifest more clearly the correlation between variables and factors. Suppose the newly selected factors are expressed in terms of columns of the orthogonal matrix

$$V := [\mathbf{v}_1, \ldots, \mathbf{v}_m] \in \mathbb{R}^{m\times m}. \tag{2.5}$$

Then this rewriting of factor loadings with respect to V is mathematically equivalent to a change of basis, i.e., A is now written as $B := AV$. One of the fundamental problems in the practice of factor analysis is to determine some appropriate new basis for V. Note that because $VV^T = I_m$, the very same observed data now is decomposed as $Y = AF = (AV)(V^T F) = BG$ with B and $G = V^T F$ representing, respectively, the factor loadings and uncorrelated standard factor scores corresponding to the factors in V. From this we also see that the correlation matrix $R = AA^T = BB^T \in \mathbb{R}^{n\times n}$ is independent of factors selected. This is another reason that in the process of defining new factors it is often desirable to retrieve information directly from the correlation matrix R rather than from any particular loading matrix A. The centroid method,

the main topic of this paper, has been used for retrieving such factors. The new factors in the centroid method were defined via successive rank reduction applied to the correlation matrix R.

3. Centroid factor. We shall denote $A_1 := A$, $R_1 := R$, and call the relationship $R_1 = A_1 A_1^T$ the *product moment* of A_1. Temporarily assuming that a loading matrix A_1 is given, the coordinate axes in $\mathbb{R}^m$ represent a set of m abstractly defined factors. The centroid method amounts to a procedure of defining a new coordinate system representing what are called the *centroid factors*. The most important feature of the centroid method is that loadings with respect to the centroid factors can be calculated without the knowledge of A_1 or even of the centroid factors. The assumption of knowing A_1 a priori, therefore, is not needed. But in the following we continue to use A_1 to gain insight into the meaning of the extraction steps.

The ith row in the matrix A_1 denotes the loadings of variable i across the spectrum of the current set of factors. Denoting each row of A_1 as a point in the factor space $\mathbb{R}^m$, the arithmetic mean of these points *might* be used to indicate a collective trend of the variables. That direction could be a substantial factor to be weighted in and thus constitutes the essential idea of a centroid factor. Before we move into details, we should comment that generally variables that tend to vary together form clusters. If all the variables are truly independent, there should be no clusters at all. On the other extreme, if all the variables are dependent on the same factor, then a single cluster should be formed. In between the two extremes, we do not know a priori how many clusters are to be expected. There are many cluster detection techniques. See, for example, the book [1]. Among these, the so called *k-means method* is perhaps the most commonly used in practice. The centroid method we are about to describe is the simplest special variation of the k-means method. The centroid method iteratively searches for one mean a time. Since the goal of this paper is to compare the relationships of various discrete variational decompositions with the SVD, our present discussion will be concentrating on the (1-mean) centroid method only. The generalization of comparison to a k-means method should be another interesting research topic in the future.

Given $A_1 \in \mathbb{R}^{n\times m}$, the centroid of these n variables is given by the column vector

$$\mathbf{c}_1 := \frac{A_1^T \mathbf{1}_n}{n} = \left[\frac{\sum_{i=1}^n a_{i1}}{n}, \ldots, \frac{\sum_{i=1}^n a_{im}}{n}\right]^T, \tag{3.1}$$

where $\mathbf{1}_n$ denotes the column vector $\mathbf{1}_n := [1, \ldots, 1]^T \in \mathbb{R}^n$. The first *centroid factor* is defined to be the normalized vector

$$\mathbf{v}_1 := \frac{\mathbf{c}_1}{\|\mathbf{c}_1\|}. \tag{3.2}$$

The new loadings of variables with respect to this new factor $\mathbf{v}_1$, i.e., the first column $\mathbf{b}_1 = [b_{11}, \ldots, b_{n1}]^T$ of the new loading matrix B (which is yet to be found), can be calculated without referring to A_1 as follows: Each component b_{j1} is precisely the projection component of variable j along the unit vector $\mathbf{v}_1$, i.e., $\mathbf{b}_1 = A_1\mathbf{v}_1$. This can be rewritten as

$$\mathbf{b}_1 = A_1 \frac{A_1^T \mathbf{1}_n}{\|A_1^T \mathbf{1}_n\|} = \frac{R_1 \mathbf{1}_n}{\sqrt{\mathbf{1}_n^T R_1 \mathbf{1}_n}}. \tag{3.3}$$

In this way, we note that the first loading vector $\mathbf{b}_1$ is extracted directly from R_1. No reference to A_1 or $\mathbf{v}_1$ is needed.

Once the loadings $\mathbf{b}_1$ for the centroid factor $\mathbf{v}_1$ are found, the product moment R_1 is conventionally updated to a new matrix R_2 defined by

$$R_2 := R_1 - \frac{R_1 \mathbf{1}_n \mathbf{1}_n^T R_1}{\mathbf{1}_n^T R_1 \mathbf{1}_n}. \tag{3.4}$$

It is important to understand the meaning of R_2. Define

$$A_2 := A_1 - A_1 \mathbf{v}_1 \mathbf{v}_1^T. \tag{3.5}$$

Observe that each row in the matrix A_2 represents the component of the original loadings A_1 in the direction orthogonal to $\mathbf{v}_1$. We shall call A_2 the *orthogonally reduced loading matrix* of A_1 with respect to $\mathbf{v}_1$. Note that A_2 inherits most of the loading information of the original A_1 except for the loadings along the direction $\mathbf{v}_1$. The information along $\mathbf{v}_1$ is subtracted from A_1 to form A_2. The following statement provides an interesting interpretation of R_2.

THEOREM 3.1 (see [3]). *The traditional rank-one update* (3.4) *from* R_1 *is simply another way to calculate the product moment of the orthogonally reduced loading matrix* A_2 *without directly referring to* A_2.

Proof. The product moment of A_2 can be computed as follows:

$$\begin{aligned} A_2 A_2^T &= \left(A_1 - A_1 \mathbf{v}_1 \mathbf{v}_1^T\right)\left(A_1^T - \mathbf{v}_1 \mathbf{v}_1^T A_1^T\right) \\ &= A_1 A_1^T - A_1 \mathbf{v}_1 \mathbf{v}_1^T A_1^T \\ &= R_1 - \frac{R_1 \mathbf{1}_n \mathbf{1}_n^T R_1}{\mathbf{1}_n^T R_1 \mathbf{1}_n}, \end{aligned}$$

where the last equality follows from (3.1). ∎

With A_2 or R_2 in hand, it seems that the above procedure can be repeated to extract the next centroid factor for A_2, to introduce the next reduced loading matrix, and so on. Unfortunately, this is not the case. The procedure cannot be repeated because $A_2^T \mathbf{1}_n = \mathbf{0}_m$. In other words, because the centroid of A_2 is residing squarely at the origin of $\mathbb{R}^m$, the second centroid factor is null. We have to modify the notion of centroid somewhat to circumvent this situation.

It is worth mentioning that the update (3.4) is simply one special case of the well-known Wedderburn rank reduction formula [4]. The rank of R_2 is precisely one less than that of R_1.

4. Modified centroid factor. In factor analysis, one major task is to ascribe the loadings in A_1 to as few *essential factors* as possible. We consider that a factor is essential if loadings with respect to that particular factor are relatively weighty. Being the average of all variables, the centroid factor $\mathbf{v}_1$ would delineate an essential factor under the following circumstances:

1. When all points in $\mathbb{R}^m$ representing rows of A_1 stay near the line determined by $\mathbf{v}_1$: In this case, each variable is approximately a scalar multiple of $\mathbf{v}_1$. The scalar can be positive or negative, indicating a positive or negative linear correlation between the variable and the factor $\mathbf{v}_1$. In either case, it is clear that a substantial portion of loadings in A_1 should be attributed to the factor $\mathbf{v}_1$.
2. When the centroid $\mathbf{c}_1$ is far away from the origin: In this case, the variables are asymmetrically distributed in the factor space $\mathbb{R}^m$. The quantity $\|\mathbf{c}_1\|$

measures, in some sense, the *eccentricity* of the system of variables with respect to the origin. That is, the farther $\mathbf{c}_1$ is away from the origin, the more variables are qualitatively scattered in a general area surrounding $\mathbf{c}_1$. Thus the larger $\|\mathbf{c}_1\|$ is, the better an essential factor $\mathbf{v}_1$ represents.

It is worth noting again, as we have already pointed out in the first remark above, that replacing one particular variable by its negative does not cause trouble in the identification of an essential factor. We therefore should change the sign of certain rows if that helps to bring out other properties such as that described in the second remark above. On the other hand, the scalar

$$\mathbf{1}_n^T R \mathbf{1}_n = \|A_1^T \mathbf{1}_n\|^2 = n^2 \|\mathbf{c}_1\|^2 \tag{4.1}$$

is a fixed multiple of $\|\mathbf{c}_1\|$. Combining these observations, we are motivated to consider the integer programming problem

$$\max_{|\mathbf{z}|=1} \mathbf{z}^T R_1 \mathbf{z}, \tag{4.2}$$

where $|\mathbf{z}| = 1$ means the components of the column vector $\mathbf{z}$ are either 1 or -1. We call $\mathbf{z}$ a *sign vector*. There are only 2^n sign vectors for a fixed n. Without causing any ambiguity, we shall use the same notation to represent the vectors

$$\mathbf{c}_1 := \frac{A_1^T \mathbf{z}_1}{n}, \tag{4.3}$$

$$\mathbf{v}_1 := \frac{A_1^T \mathbf{z}_1}{\|A_1^T \mathbf{z}_1\|}, \tag{4.4}$$

where $\mathbf{z}_1$ is the optimizer of (4.2), and call them the *modified centroid* and the *modified centroid factor*, respectively. For later reference, we shall call

$$\mu_1 := \frac{1}{n} \max_{|\mathbf{z}|=1} \mathbf{z}^T R_1 \mathbf{z} \tag{4.5}$$

the first *centroid value* of A_1. The following results are generalizations of (3.3) and Theorem 3.1.

THEOREM 4.1. *The loading* $\mathbf{b}_1$ *with respect to the modified centroid factor* $\mathbf{v}_1$ *defined by* (4.4) *is given by the projection* $\mathbf{b}_1 = A_1 \mathbf{v}_1$ *and can be computed by*

$$\mathbf{b}_1 = \frac{R_1 \mathbf{z}_1}{\sqrt{\mathbf{z}_1^T R_1 \mathbf{z}_1}}. \tag{4.6}$$

The product moment R_2 *of the orthogonally reduced loading matrix* $A_2 = A_1 - A_1 \mathbf{v}_1 \mathbf{v}_1^T$ *can be computed by*

$$R_2 = R_1 - \frac{R_1 \mathbf{z}_1 \mathbf{z}_1^T R_1}{\mathbf{z}_1^T R_1 \mathbf{z}_1}. \tag{4.7}$$

We remark again that in the above expression both $\mathbf{b}_1$ and R_2 can be calculated without making explicit reference to A_1. By now, it should be clear that the notion of modified centroid factor can be applied to R_2 to induce the next R_3, and so on. With this generalization, we should also point out that henceforth the matrix R no longer denotes a correlation matrix but rather a general symmetric and positive semidefinite

matrix. Each application of this centroid factor retrieval will reduce the rank of the loading matrix by one. The procedure therefore has to come to a stop in finitely many steps. In this way, with the recurrence

$$A_i = A_{i-1} - A_{i-1}\mathbf{v}_{i-1}\mathbf{v}_{i-1}^T, \quad i = 2, \ldots, \gamma, \tag{4.8}$$

where $\mathbf{v}_i$ is the modified centroid factor of A_i, γ is the rank of A_1, and with the loadings $b_i = A_i\mathbf{v}_i$, we may write

$$A = A_1 = \mathbf{b}_\gamma \mathbf{v}_\gamma^T + \cdots + \mathbf{b}_1\mathbf{v}_1^T, \tag{4.9}$$

which we will call a *centroid decomposition* of A.

Let $\mathbf{z}_2$ be the sign vector that maximizes $z^T R_2 z$. Note that $\mathbf{z}_2 \neq \mathbf{z}_1$ because $R_2\mathbf{z}_1 = 0$. The modified centroid $\mathbf{c}_2$ for A_2, according to (4.3), should be $\mathbf{c}_2 = \frac{A_2^T\mathbf{z}_2}{n}$. It is interesting to note that

$$\mathbf{c}_1^T\mathbf{c}_2 = \frac{1}{n^2}\left(\mathbf{z}_1^T A_1\right)\left(A_2^T\mathbf{z}_2\right) = \frac{1}{n^2}\left(\mathbf{z}_1^T A_1\right)\left[A_1^T\left(\mathbf{z}_2 - \frac{\mathbf{z}_1^T R_1\mathbf{z}_2}{\mathbf{z}_1^T R_1\mathbf{z}_1}\mathbf{z}_1\right)\right] = 0, \tag{4.10}$$

i.e., the modified centroids (and factors) are mutually orthogonal even though they are not explicitly calculated.

5. Centroid method. To perform the centroid decomposition, a sequence of integer programming problems such as (4.2) must be solved. The feasible set consists of 2^n sign vectors. An exhaustive search would be expensive. Fortunately, an interesting quick iterative approach, called the *centroid method*, has been developed in the AS/P literature for solving the underlying maximization problem. We shall briefly review the centroid method in this section. In particular, we want to provide a geometric interpretation of the centroid method.

Upon identifying -1 as 0 and keeping 1 as 1, we can associate a unique binary tag to each sign vector. Each binary tag, in turn, is translated into a unique integer between 0 and $2^n - 1$ that provides a natural ordering of the sign vectors. For example, sign vectors $[-1,-1,-1,-1]^T$ and $[-1,1,-1,1]^T$ have binary tags 0000 and 0101 and are the 0th and the 5th in the order, respectively. If we consider each sign vector as one node connected to all other sign vectors whose binary tags differ from its own by exactly one bit, then topologically the set of 2^n sign vectors can be identified as an n-dimensional *hypercube*. A 4-dimensional hypercube layout together with the ordering of its vertices is depicted in Figure 5.1. Note (see Figure 5.1) that each n-dimensional hypercube consists of two $(n-1)$-dimensional subhypercubes where one subhypercube is simply a *bit reversal* of the other. The objective values $z^T Rz$ therefore always appear in pairs.

The integer programming problem over sign vectors now becomes the maximization of $\mathbf{z}^T R\mathbf{z}$ over vertices on the hypercube. Without causing any ambiguity, let R stand for any of the product moments R_i involved in the process. Write $R = [r_{ij}] = P + \mathrm{diag}(\mathrm{diag}(R))$. Since $\mathbf{z}^T R\mathbf{z} = \mathbf{z}^T P\mathbf{z} + \sum_{i=1}^n r_{ii}$, it suffices to consider the problem of maximizing

$$f(\mathbf{z}) := \mathbf{z}^T P\mathbf{z}$$

with $|\mathbf{z}| = 1$. The classical centroid method is described next.

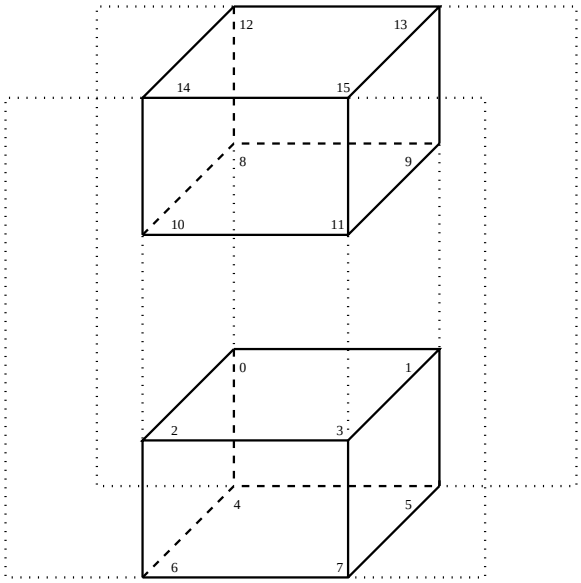


FIG. 5.1. *Topology of a 4-dimensional hypercube.*

ALGORITHM 5.1. *Given any sign vector* $\mathbf{z}^{(0)}$ *and machine zero threshold* ϵ*, define* $\mathbf{w}^{(0)} := P\mathbf{z}^{(0)}$*. Repeat the following steps for* $i = 0, 1, \ldots$:

1. *If* $sgn(\mathbf{w}_k^{(i)}) = sgn(\mathbf{z}_k^{(i)})$ *for all* $k = 1, \ldots, n$*, then stop; otherwise, choose* k *so that* $|\mathbf{w}_k^{(i)}| > \epsilon$ *and is the largest among all* $|\mathbf{w}_j^{(i)}|$*'s where* $sgn(\mathbf{w}_j^{(i)}) \neq sgn(\mathbf{z}_j^{(i)})$.
2. *Define* $\mathbf{z}^{(i+1)}$ *by simply changing the sign of* $\mathbf{z}_k^{(i)}$.
3. *Define* $\mathbf{w}^{(i+1)} := \mathbf{w}^{(i)} + 2sgn(\mathbf{z}_k^{(i+1)})P(:, k)$.

Since at most one bit is changed in each cycle, it is seen from the above that the centroid method involves advancing from one node to one of its neighboring nodes on the hypercube. The convergence behavior of this algorithm can be seen from the following result.

THEOREM 5.1. *The sequence* $\{f(\mathbf{z}^{(i)})\}$ *where* $\mathbf{z}^{(i)}$ *is generated by the centroid method from any starting value* $\mathbf{z}^{(0)}$ *is finite and increasing.*

Proof. We can rewrite the definition of $\mathbf{z}^{(i+1)}$ as

$$\mathbf{z}^{(i+1)} := \mathbf{z}^{(i)} - 2\text{sgn}(\mathbf{z}_k^{(i)})\mathbf{e}_k,$$

where $\mathbf{e}_k$ is the standard kth unit vector. Observe

$$\begin{aligned} f(\mathbf{z}^{(i+1)}) &= \left(\mathbf{z}^{(i)} - 2\text{sgn}(\mathbf{z}_k^{(i)})\mathbf{e}_k\right)^T P \left(\mathbf{z}^{(i)} - 2\text{sgn}(\mathbf{z}_k^{(i)})\mathbf{e}_k\right) \\ &= f(\mathbf{z}^{(i)}) - 4\text{sgn}(\mathbf{z}_k^{(i)})(\mathbf{e}_k^T P\mathbf{z}^{(i)}) \\ &= f(\mathbf{z}^{(i)}) - 4\text{sgn}(\mathbf{z}_k^{(i)})\mathbf{w}_k^{(i)}. \end{aligned}$$

Note that, by the definition of k, the second term in the last equality is negative, showing that $f(\mathbf{z}^{(i+1)})$ is strictly larger than $f(\mathbf{z}^{(i)})$ by $4|\mathbf{w}_k^{(i)}|$. The centroid method

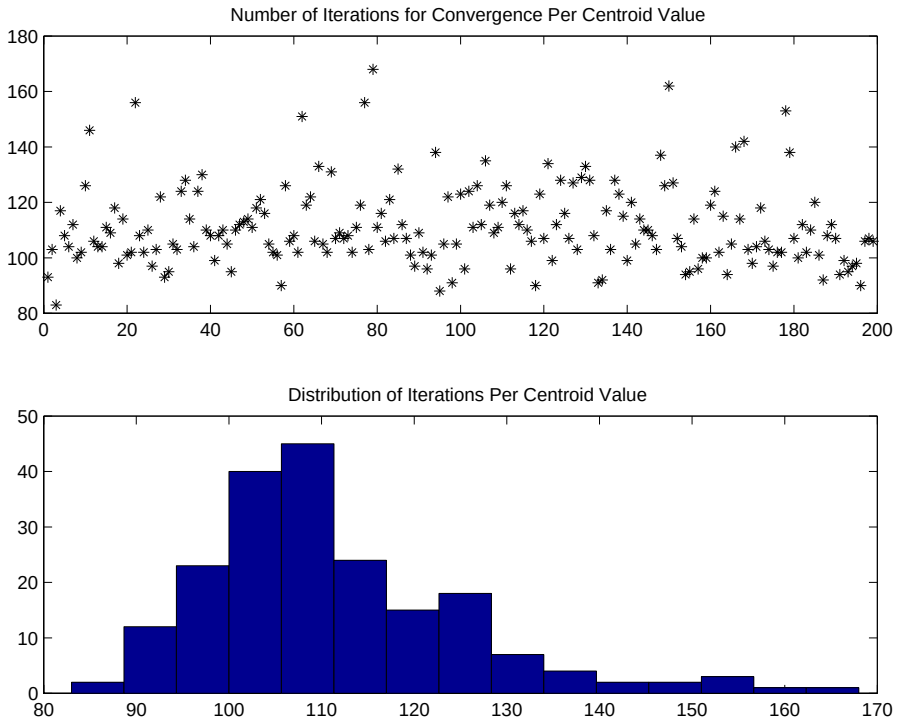


FIG. 5.2. *Number of steps per centroid value in the centroid method for a matrix of size* 200.

can be regarded as a steepest ascent method along the nodes of the hypercube. There are only finitely many nodes; the sequence therefore has to converge in finitely many steps. □

Although there are 2^n nodes on an n-dimensional hypercube, to go from one node to another node in order to maximize $\mathbf{z}^T R\mathbf{z}$ is not an NP problem. Indeed, recall that each node is identified by a binary tag of length n. Recall also that the centroid method (Algorithm 5.1) has the unique feature of changing only one bit at a time and never descends. The worst scenario is that the iteration moves from one binary tag, say 1010 in the case $n = 4$, to its bit reversal tag, say 0101. In other words, it takes at most n iterations to locate a maximum. We can prove by induction that the expected number of iterations required for convergence to a centroid value is in fact $\frac{n}{2}$. To illustrate this point, we report in Figure 5.2 just one of the many numerical simulations we have conducted on the number of iterations needed to generate each centroid value. The lower graph in Figure 5.2, depicting the histogram of these number of iterations, suggests that the mean is about 100.

We conclude this section by cautioning that the centroid method only finds a *local* maximum. Even after excluding the parity resulting from bit reversal mentioned before, the local maximum may not be unique for a given P. For example, with

$$P = \begin{bmatrix} 0 & 3.5 & 3 & 1 \\ 3.5 & 0 & -4 & -3 \\ 3 & -4 & 0 & -3.5 \\ 1 & -3 & -3.5 & 0 \end{bmatrix},$$

the objective value $z^T Pz$ has local maxima at the 1st, 2nd, and 4th sign vectors. The mechanism built in the first step of the centroid method dictates that the algorithm converges to the 2nd (or its bit reversal, 14th) sign vector $[-1,-1,-1,1]^T$ unless the starting value happens to be the other two local maximizers, in which case the algorithm stalls right there.

6. Relationship to truncated matrices. In the practice of information retrieval, quite often the original data matrix Y is not exact due to noise. It is often sufficient to replace Y by a simpler approximation. This approximation matrix is obtained by *truncating* the original matrix in some sense. In this section we shall provide a statistical meaning of truncation. At the end of this section we establish the statistical soundness of the centroid method and compare its decomposition with the SVD. In contrast,we shall see in the next section that the SDD [12], though approximating a SVD decomposition, does not readily imply the same desirable stochastic meaning.

We first consider a general random variable $\mathcal{X}$ in $\mathbb{R}^n$. Let $\mathcal{E}[\mathcal{X}]$ denote the expected value of $\mathcal{X}$. Typically, $\text{cov}(\mathcal{X}) := \mathcal{E}[(\mathcal{X}-\mathcal{E}[\mathcal{X}])(\mathcal{X}-\mathcal{E}[\mathcal{X}])^T] \in \mathbb{R}^{n\times n}$ is defined as the *covariance matrix* of $\mathcal{X}$. Let

$$\text{cov}(\mathcal{X}) = \sum_{j=1}^{n} \lambda_j \mathbf{u}_j \mathbf{u}_j^T \tag{6.1}$$

denote the spectral decomposition of $\text{cov}(\mathcal{X})$ with eigenvalues arranged in the descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Note that $\mathbf{u}_1, \ldots, \mathbf{u}_n$ form an orthonormal basis for $\mathbb{R}^n$. Express the random column variable $\mathcal{X}$ as

$$\mathcal{X} = \sum_{j=1}^{n} (\mathbf{u}_j^T \mathcal{X}) \mathbf{u}_j. \tag{6.2}$$

Note that the columns in the matrix $U := [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ are deterministic vectors themselves. The randomness of $\mathcal{X}$ therefore must come solely from the randomness of each coefficient in (6.2). The following observation in [3] sheds important insight on the portion of randomness of $\mathcal{X}$ in each eigenvector direction of $\text{cov}\mathcal{X}$.

THEOREM 6.1. *Let $\boldsymbol{\alpha} := U^T \mathcal{X}$. Then $\boldsymbol{\alpha}$ is a random variable whose components are mutually stochastically independent. Indeed,*

$$\mathcal{E}[\boldsymbol{\alpha}] = U^T \mathcal{E}[\mathcal{X}], \tag{6.3}$$

$$cov(\boldsymbol{\alpha}) = \text{diag}\{\lambda_1, \ldots, \lambda_n\}. \tag{6.4}$$

In other words, the larger the eigenvalue λ_j of $\text{cov}(\mathcal{X})$ is, the larger the variance of α_j is, i.e., the more stochastic properties such as randomness the vector $\alpha_j \mathbf{u}_j$ contributes to $\mathcal{X}$. From (6.2) it appears intuitive that those coefficients α_j with larger variance represent a more integral part of $\mathcal{X}$. We therefore can *rank* the importance of corresponding eigenvectors u_j as essential components for the variable $\mathcal{X}$ according to the magnitude of λ_j.

If it becomes desirable to approximate the random variable $\mathcal{X}$ by another unbiased yet *simpler* variable $\hat{\mathcal{X}}$, we see from Theorem 6.1 that $\hat{\mathcal{X}}$ had better capture those components corresponding to larger λ_j in the expression (6.2). Indeed, it is entirely sensible to require that $\text{cov}(\hat{\mathcal{X}})$ be reasonably close to $\text{cov}(\mathcal{X})$. We quantify this notion with the following theorem, which provides the basic idea of truncation. The proof can be found in [3], which uses results from [13, 14].

THEOREM 6.2. *Suppose that $\mathcal{X}$ is a random variable in $\mathbb{R}^n$ with mean zero and that its covariance matrix has a spectral decomposition given by* (6.1). *Then among all unbiased variables restricted to any r-dimensional subspaces in $\mathbb{R}^n$, the random variable*

$$\hat{\mathcal{X}} := \sum_{j=1}^{r} (\mathbf{u}_j^T \mathcal{X}) \mathbf{u}_j \tag{6.5}$$

is the best approximation to $\mathcal{X}$ in the sense that $\|cov(\hat{\mathcal{X}}) - cov(\mathcal{X})\|$ is minimized. In addition, $\hat{\mathcal{X}}$ is the best linear minimum-variance estimate of $\mathcal{X}$ in the sense that $\mathcal{E}[\|\hat{\mathcal{X}} - \mathcal{X}\|^2]$ is minimized.

It is important to note that in the above linear minimum-variance estimation, the variable $\mathcal{X}$ is *centered* at zero. If $\mathcal{X}$ is not centered at zero, the expression for truncation would be much more complicated. Without the centering, the mere truncated data in the form of a low rank approximation would suffer from the loss of some significant statistical meanings.

The above observation is based on the fact that the random variable $\mathcal{X}$ is completely known. Such an assumption is not practical in reality since often the probability distribution function of $\mathcal{X}$ is not known a priori. One common practice in applications then is to simulate the random variable $\mathcal{X}$ by a collection of ℓ random samples. These samples are recorded in an $n \times \ell$ matrix. Our data matrix $Y = [y_{ij}]$ is precisely such an example where each column of Y represents one random sample of (standard) score for a certain random variable $\mathcal{X} \in \mathbb{R}^n$ which, in this case, has mean zero. It is known that when ℓ is large enough, many of the stochastic properties of $\mathcal{X}$ can be recouped from Y.

The question now is how to retrieve a sample data matrix from Y to represent the truncated variable $\hat{\mathcal{X}}$. The connection lies in the observations that the matrix R is close to $\mathrm{cov}(\mathcal{X})$ by the law of large numbers. Note that the eigenvalues of R are precisely the squares of the singular values of $Y/\sqrt{\ell}$ and that the singular values, by Theorem 6.1, measure the degree of randomness (of $\mathcal{X}$) in the direction of the left singular vectors (of Y). In the spirit of truncation described in Theorem 6.2 above, the data matrix $\hat{Y}$ for $\hat{\mathcal{X}}$ should be such that both $\|Y - \hat{Y}\|$ and $\|YY^T - \hat{Y}\hat{Y}^T\|$ are minimized. It turns out that the truncated SVD of Y by throwing away its smaller singular values satisfies precisely these requirements. See [3] for a more detailed discussion. In this way, we understand now that the truncated SVD $\hat{Y}$ not only is the best approximation to Y in the sense of norm but more importantly is the closest approximation to Y in the sense of statistics. It maintains the most significant stochastic portion of the original data matrix Y. Generally speaking, any lower rank approximation to an empirical data matrix Y should carry properties similar to the truncated SVD, i.e., should contain substantial stochastic information about the original random variable $\mathcal{X}$.

Coming back to the factor retrieval problem (2.4), we should note that while the product moment AA^T gives rise to the same (covariance) matrix R as Y does, the loading matrix A itself does not represent a sample data matrix of any random variable. Indeed, the number m of factors (or columns) in A could be far shorter than ℓ to represent meaningful samples. However, as far as approximating R by the product moment of some lower rank matrices is concerned, the idea of truncation can still be carried over. That is, we would like that a significant portion of those components in R corresponding to larger eigenvalues be captured by its low rank approximation $\hat{R}$ regardless of whether $\hat{R}$ is calculated via truncated random samples $\hat{Y}$ or reduced

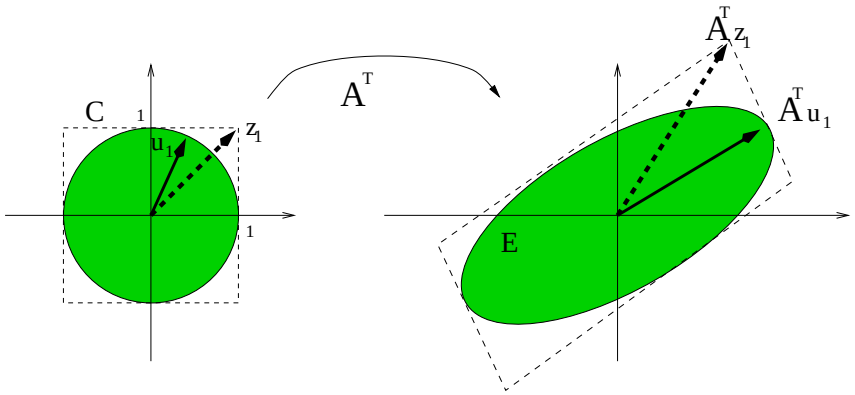


FIG. 6.1. *Comparison of geometric meanings of* $\mathbf{z}_1$ *and* $\mathbf{u}_1(R_1)$ *when* $n = 2$.

factors $\hat{A}$. The centroid method is an alternative way to accomplish that goal, as we shall now explain below.

For convenience, let $\lambda_1(M) \geq \lambda_2(M) \geq \cdots \geq \lambda_n(M)$ denote the eigenvalues of any given real-valued symmetric M. Let the corresponding unit eigenvectors be denoted as $\mathbf{u}_1(M), \ldots, \mathbf{u}_n(M)$. Recall the Rayleigh–Ritz theorem asserting that [8]

$$\lambda_1(M) = \max_{\|\mathbf{x}\|=1} \mathbf{x}^T M \mathbf{x}, \tag{6.6}$$

$$\lambda_k(M) = \max_{\substack{\|\mathbf{x}\|=1 \\ \mathbf{x}\perp \mathbf{u}_1(M),\ldots,\mathbf{u}_{k-1}(M)}} \mathbf{x}^T M \mathbf{x} \quad \text{for } k = 2, \ldots, n. \tag{6.7}$$

This variational characterization suggests a scheme for eigenvalue computation. Although the scheme is of little practical value in itself, its comparison with the centroid method is worth mentioning. First, observe that

$$\begin{aligned} \lambda_1(R_1) = (\mathbf{u}_1(R_1))^T R_1 \mathbf{u}_1(R_1) &= \max_{\|\mathbf{u}\|=1} \mathbf{u}^T R_1 \mathbf{u} = \max_{\|\mathbf{u}\|=1} \|A_1^T \mathbf{u}\|^2 \\ &\geq \mu_1 = \frac{1}{n} \mathbf{z}_1^T R_1 \mathbf{z}_1 = \frac{1}{n} \max_{|\mathbf{z}|=1} \mathbf{z}^T R_1 \mathbf{z} = \frac{1}{n} \max_{|\mathbf{z}|=1} \|A_1^T \mathbf{z}\|^2, \end{aligned} \tag{6.8}$$

where $\mathbf{z}_1$ is used to define the first modified centroid (see (4.2)). This relationship suggests that the sign vector $\mathbf{z}_1$ and the centroid value μ_1 are *mimicking* the left singular vector $\mathbf{u}_1$ and the square of the singular value λ_1 of A_1, respectively. Recall that the singular values of A_1^T are precisely the lengths of the semiaxes of the hyperellipsoid E defined by

$$E := \{A_1^T \mathbf{x} \in \mathbb{R}^m | \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\},$$

whence the first left singular vector $\mathbf{u}_1(R_1)$ of A_1 is mapped via A_1^T to the first major semiaxis of E. On the other hand, the unit cube

$$C := \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_\infty = 1\}$$

is mapped under A_1^T to a hyperparallelepiped that circumscribes E. The geometric meanings of $\mathbf{z}_1$ and $\mathbf{u}_1(R_1)$ can be compared via Figure 6.1, where we draw C and E for the case $n = 2$.

Recall that once a factor represented by a unit vector $\mathbf{v}$ is determined, the components in the product $\mathbf{b} = A_1\mathbf{v}$ represent the loadings of all variables in that factor. The size of $\|\mathbf{b}\|$, called the significance earlier, can be used to indicate how essential that factor is to the variables. While the modified centroid is obtained by weighting loadings of all variables uniformly in the factor space, i.e., by constant weight $\frac{1}{n}$ except signs, the SVD amounts to weighting these loadings unevenly so as to maximize the significance of the resulting *biased centroid* (giving rise to the left singular vector). The latter is a fairly expensive and difficult task to accomplish, while the former is relatively easy to do via the centroid method. In this sense, we say that $\mathbf{z}_1$ gives a foretaste of the location of $\mathbf{u}_1$.

Recall in the centroid method that once the first centroid factor is found, the matrix R_1 is reduced to R_2 according to (4.7) and the search for the next centroid factor continues. In exactly the same way, once the first eigenvector $\mathbf{u}_1(R_1)$ is found, the matrix R_1 can be reduced, by the same Wedderburn rank reduction formula, to

$$\overline{R}_2 := R_1 - \lambda_1 \mathbf{u}_1(R_1)\,(\mathbf{u}_1(R_1))^T\,. \tag{6.9}$$

It can easily be proved from (6.7) that

$$(\mathbf{u}_2(R_1))^T\, R_1 \mathbf{u}_2(R_1) = \lambda_2(R_1) = \lambda_1(\overline{R}_2) = \left(\mathbf{u}_1(\overline{R}_2)\right)^T \overline{R}_2 \mathbf{u}_1(\overline{R}_2). \tag{6.10}$$

The relationship described above between $\mathbf{z}_1$ and $\mathbf{u}_1(R_1)$ in principle can be carried over to a similar relationship between $\mathbf{z}_2$ and $\mathbf{u}_2(R_1)$. The only problem is that

$$R_2 \neq \overline{R}_2 \tag{6.11}$$

because the two matrices are reduced from R_1 using $\mathbf{z}_1$ and $\mathbf{u}_1(R_1)$, respectively. However, we have pointed out earlier that at least in the initial stage $\mathbf{z}_1$ mimics the role of $\mathbf{u}_1(R_1)$ reasonably, so most of the stochastic information in R_2 should remain close to that in $\overline{R}_2$. As the iteration continues, of course, the closeness between R_i and $\overline{R}_i$ begins to depart. Consequently, the resemblance between $\mathbf{z}_i$ and $u_i(R_i)$ is expected to deteriorate progressively. Regardless, if we are interested in only the first few essential factors, i.e., in capturing the qualitative behavior of the (truncated) SVD of A_1, the centroid decomposition seems to be a reasonable and quick alternative. We summarize the comparison of the centroid decomposition and SVD in Table 6.1. We indicate only the first step in both decompositions. The successive steps are done similarly.

Furthermore, we plot in Figure 6.2 the centroid values and the singular values of the correlation matrix of a randomly generated 200×200 matrix A_1. Recall from Theorem 6.1 that the singular values indicate the degree of contribution to the randomness by the left singular vectors. Figure 6.2 is a typical representation of our many random tests. From the figure we see that the centroid values seem to mimic the behavior of singular values reasonably well and, hence, should provide a reasonable measurement of the original stochastic nature.

On the other hand, we point out that the reduced matrices R_i are no longer the same as $\overline{R}_i$ after the first step. We therefore plot in Figure 6.3 the logarithmic values of $|\cos(\theta_i)|$ where θ_i is the angle between $\mathbf{z}_i$ and $\mathbf{u}_i$ for $i = 1, \ldots, 200$. These values allow us to examine the degree of alignment of the sign vector $\mathbf{z}_i$ for the matrix R_i with the ith left singular vector of R_1. This diagram seems to suggest that the loss of alignment is not bad. In fact, toward the end of the calculation, it seems that the alignment is remarkably good. Further research is needed to understand this alignment issue.

TABLE 6.1
Comparison of centroid decomposition and SVD.

Centroid decomposition	SVD
$\mu_1 = \frac{1}{n}\max_{\|\mathbf{z}\|=1} \mathbf{z}^T R_1 \mathbf{z}$ (centroid value)	$\lambda_1 = \max_{\|\mathbf{x}\|=1} \mathbf{x}^T R_1 \mathbf{x}$ (eigenvalue)
$\mathbf{z}_1 = \arg\max_{\|\mathbf{z}\|=1} \mathbf{z}^T R_1 \mathbf{z}$ (sign vector for modified centroid)	$\mathbf{u}_1 = \arg\max_{\|\mathbf{x}\|=1} \mathbf{x}^T R_1 \mathbf{x}$ (left singular vector)
easy to obtain $\mathbf{z}_1$ in $O(n)$ steps (transverse hypercube)	not easy to obtain $\mathbf{u}_1$ via iterations (nonlinear iteration)
$\mathbf{v}_1 = \frac{A_1^T \mathbf{z}_1}{\sqrt{n\mu_1}}$ (centroid factor)	$\hat{\mathbf{v}}_1 = \frac{A_1^T \mathbf{u}_1}{\sqrt{\lambda_1}}$ (right singular vector)
$\gamma_1 = \|A_1 \mathbf{v}_1\|$ (significance)	$\sigma_1 = \sqrt{\lambda_1} = \|A_1 \hat{\mathbf{v}}_1\|$ (largest singular value)
$b_1 = A_1 \mathbf{v}_1$ (loading vector)	$\sigma_1 \mathbf{u}_1 = A_1 \hat{\mathbf{v}}_1$ (internal relation)
$A_1 = \sum b_i \mathbf{v}_i^T$ (centroid decomposition)	$A_1 = \sum \sigma_i \mathbf{u}_i \hat{\mathbf{v}}_i^T$ (SVD)
$R = \sum b_i b_i^T = \sum \gamma_i^2 \frac{b_i}{\|b_i\|}\left(\frac{b_i}{\|b_i\|}\right)^T$ (factor decomposition)	$R = \sum \lambda_i \mathbf{u}_i \mathbf{u}_i^T = \sum \sigma_i^2 \mathbf{u}_i \mathbf{u}_i^T$ (spectral decomposition)
$R_2 = R_1 - \frac{R_1 \mathbf{z}_1 \mathbf{z}_1^T R_1}{\mathbf{z}_1^T R_1 \mathbf{z}_1} = R_1 - \gamma_1^2 \frac{b_1}{\|b_1\|}\left(\frac{b_1}{\|b_1\|}\right)^T$ (rank reduction)	$\overline{R}_2 = R_1 - \frac{R_1 \mathbf{u}_1 \mathbf{u}_1^T R_1}{\mathbf{u}_1^T R_1 \mathbf{u}_1} = R_1 - \lambda_1 \mathbf{u}_1 \mathbf{u}_1^T$ (rank reduction)

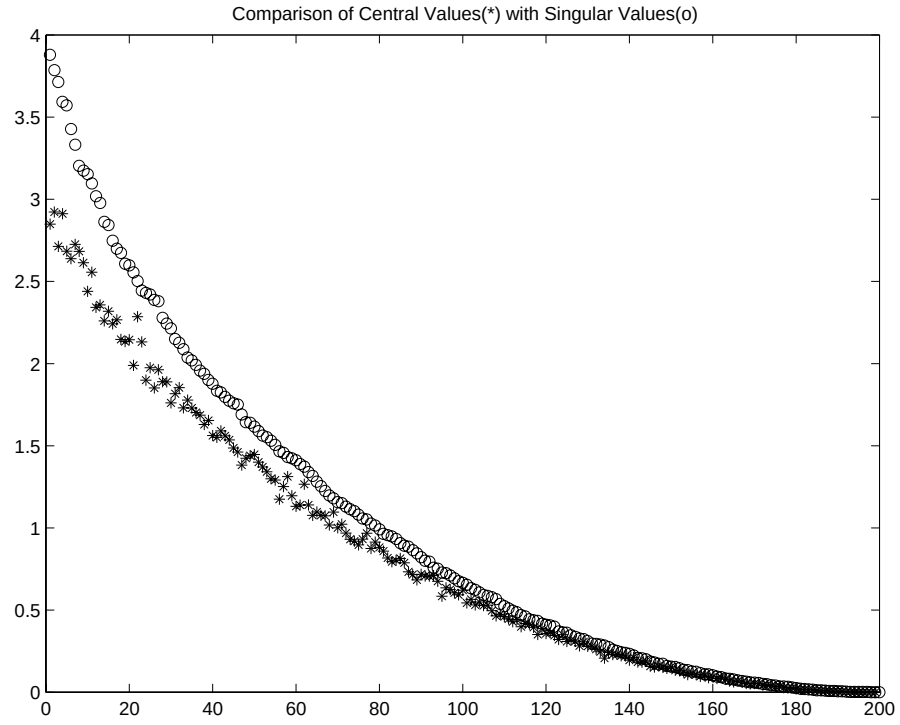


FIG. 6.2. *Comparison of centroid values and singular values for correlation matrix of* $n = 200$.

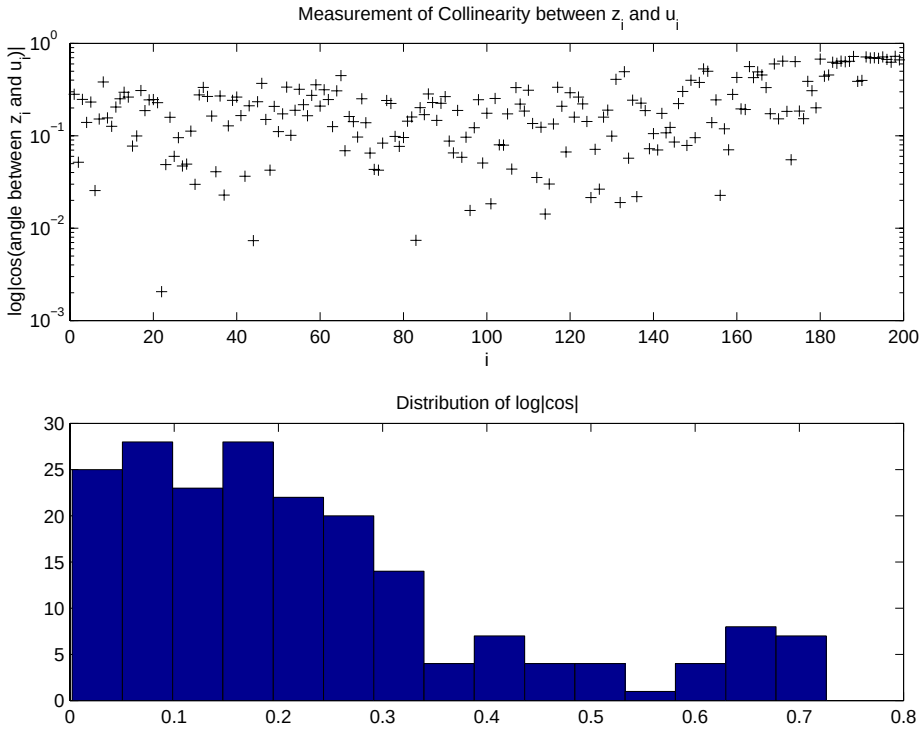


FIG. 6.3. *Degree of alignment between* $\mathbf{z}_i$ *and* u_i.

7. Relationship to data retrieval. The research and notions of factor analysis have been used in many disciplines, notably in educational, social, psychological, and behavioral measurements [5]. In this section, we connect and illustrate factor analysis development of the centroid method to that of information retrieval and data mining. This illustration leads to the unification of a class of approximations to the SVD.

We shall limit our goal in information retrieval to the task of finding documents relevant to given queries [12]. The idea in the so-called *latent semantic indexing* (LSI) is as follows: The textual documents are usually collected in an *indexing matrix* $H = [h_{kj}]$ in $\mathbb{R}^{m\times\ell}$. Each document is represented by one column in H. The entry h_{kj} in H represents the *weight* of one particular *term* k in document j whereas each term could be defined by just one single word or a string of phrases. A natural choice of the weight h_{kj} is obtained by counting the number of times that the term k occurs in document j. More elaborate weighting schemes can be found in the literature (see, for example, [12]) and are observed to yield better performance. Each query is represented as a row vector $\mathbf{q}_i^T = [q_{i1}, \ldots, q_{im}]$ in $\mathbb{R}^m$ where q_{ik} represents the weight of term k in the query i. Again, the weighting for terms in a query can also use more elaborate schemes. To measure how the query $\mathbf{q}_i^T$ matches the documents, we calculate the row vector

$$\mathbf{s}_i^T = \mathbf{q}_i^T H \tag{7.1}$$

and rank the relevance of documents to $\mathbf{q}$ according to the *scores* in $\mathbf{s}$. To put the

notation in the context of our discussion in the preceding sections, we observe the following analogies:

$$\begin{aligned}
\text{indexing matrix } H &\longleftrightarrow \text{scoring matrix } F,\\
\text{document } j &\longleftrightarrow \text{entity } j,\\
\text{term } k &\longleftrightarrow \text{factor } k,\\
\text{weight } h_{kj} &\longleftrightarrow \text{score of factor } k \text{ on entity } j,\\
\text{one query } \mathbf{q}_i^T &\longleftrightarrow \text{one row in loading matrix } A,\\
\text{weights } q_{ik} &\longleftrightarrow \text{loadings of query } i \text{ on factor } k,\\
\text{scores in } \mathbf{s}_i^T &\longleftrightarrow \text{scores in } i \text{ row of data matrix } Y.
\end{aligned}$$

Nevertheless, in contrast to the factor retrieval described above, the calculations involved in an LSI application place emphasis not so much on computing the factors based on the scores in $\mathbf{s}_i^T$, $i = 1, \ldots, n$, but rather on the vector-matrix multiplication (7.1). Indeed, in a search engine application, quite often there is only one query, i.e., $n = 1$, per the user's request. The factors are already specified by predetermined terms. The focus in LSI has been on representing the indexing matrix and the queries in a more compact form so as to facilitate the computation of the scores. Toward that end, one way to do LSI is to use the truncated SVD of H.

From our discussion in section 6, we now understand well why using the truncated SVD to represent H makes sense, provided data in H have been centered. It is not just the best approximation to H in norm, but more importantly it also contains a substantial portion of stochastic nature of the original H. On the other hand, since the original indexing matrix H is never exact, truncation also has the benefit of cutting away noise when the signal-to-noise ratio (SNR) is too small. Suppose that

$$H = \sum_{i=1}^{\gamma} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{7.2}$$

denotes the SVD of H of rank γ. A general practice in LSI is to replace H by

$$\hat{H}_k := \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{7.3}$$

with $k \ll \gamma$ and compute $\mathbf{s} \approx \mathbf{q}^T \hat{H}_k$. The problem is that the low rank $\hat{H}_k$ could require more storage than the original H that often is sparse. One of the suggestions for saving storage has been to approximate H by the SDD [12] that also resembles the SVD, i.e.,

$$\tilde{H}_k := \sum_{i=1}^{k} \delta_i \mathbf{x}_i \mathbf{y}_i^T, \tag{7.4}$$

where each $\mathbf{x}_i$ and $\mathbf{y}_i$ is constrained to have integer entries $-1, 0$, or 1, and the d_i are positive real numbers.

One purpose of this paper is to suggest using the truncated centroid decomposition as an alternative low rank approximation to an indexing matrix H, even though the objective of LSI is not to retrieve factors from observed data of scores. There is considerable similarity between the centroid decomposition and the SDD, but there is also a significant difference, as we shall address later in section 9.

8. General centroid algorithm. In the context of factor retrieval, we would like to have as few factors as possible. It is often the case that $m \ll \ell$ and $n \ll \ell$. The centroid method is applied directly to the product moment $R = AA^T$. In the context of data retrieval, what is given is the index matrix H. If $m \ll \ell$, we could simply apply Algorithm 5.1 to the product moment HH^T. This would correspond to a subject area where the number of terms is relatively limited, while the number of documents is large. However, in the context of LSI, it is common that each document contains many more terms (or keywords) and that the contents of the documents should sparsely overlap each other. It is reasonable to assume that $m \gg \ell$. In this case, it probably is not economical to form the product moment HH^T, as is the practice with factor analysis (2.1). We modify the centroid method to work directly with H.

ALGORITHM 8.1 (general centroid algorithm). *Assume that initial values* $|\mathbf{z}^{(0)}| = \mathbf{1}, \mathbf{w}^{(0)} := H^T\mathbf{z}^{(0)}$ *as well the vector* $\mathbf{d} = \operatorname{diag}(HH^T)$ *are available. Repeat the following steps for* $i = 0, 1, \ldots$:

1. *Compute* $\mathbf{g}^{(i)} := \mathbf{d} - sgn(\mathbf{z^{(i)}}) \circ (H\mathbf{w}^{(i)})$.
2. *Choose* k *so that* $\mathbf{g}_k^{(i)}$ *is maximal and greater than* ϵ*; otherwise, stop.*
3. *Define* $\mathbf{z}^{(i+1)}$ *by simply changing the sign of* $\mathbf{z}_k^{(i)}$.
4. *Update* $\mathbf{w}^{(i+1)} := \mathbf{w}^{(i)} + 2sgn(\mathbf{z}_k^{(i+1)})H(k,:)^T$.

As with Algorithm 5.1 this is an ascent method (on an m-dimensional hypercube) because

$$\begin{aligned}\|H^T\mathbf{z}^{(i+1)}\|^2 &= \left(\mathbf{z}^{(i)} - 2sgn(\mathbf{z}_k^{(i)})\mathbf{e}_k\right)^T HH^T\left(\mathbf{z}^{(i)} - 2sgn(\mathbf{z}_k^{(i)})\mathbf{e}_k\right)\\ &= \|H^T\mathbf{z}^{(i)}\|^2 + 4\mathbf{e}_k^T HH^T\mathbf{e}_k - 4sgn(\mathbf{z}_k^{(i)})(\mathbf{e}_k^T H\mathbf{w}^{(i)})\\ &= \|H^T\mathbf{z}^{(i)})\|^2 + 4\mathbf{g}_k^{(i)}\end{aligned}$$

and $\mathbf{g}_k^{(i)} > \epsilon$. Once the optimal $\mathbf{z_1}$ is found, the matrix $\mathbf{H_1} = \mathbf{H}$ is reduced to

$$H_2 := H_1 - H_1\mathbf{v}_1\mathbf{v}_1^T, \tag{8.1}$$

where $\mathbf{v}_1$ is the normalized unit vector of $H_1^T\mathbf{z}_1$, and then the algorithm is applied to H_2 and so on. A common practice in factor analysis is to terminate the algorithm whenever the resulting significance $\|\mathbf{b}_k\|$ drops below a specific threshold level. If the algorithm is carried to the end, we obtain the centroid decomposition

$$H = H_1 = \mathbf{b}_\gamma\mathbf{v}_\gamma^T + \cdots + \mathbf{b}_1\mathbf{v}_1^T, \tag{8.2}$$

where $\mathbf{b}_k := H_k\mathbf{v}_k$, $k = 1, \ldots, \gamma$, is the *loading* whose significance is analogous to the singular value of H_k. (See the comparison in Table 6.1.) Furthermore, from the discussion in section 6, we see that the first few loadings carry most of the stochastic information in H. That is, a truncated centroid decomposition may often be as effective as the truncated SVD and can be expected to be much cheaper computationally, as we will now see.

Since a single centroid iteration on a correlation has an order n sorting (step 1 in Algorithm 5.1) and an n-dimensional vector addition (step 3 in Algorithm 5.1), the complexity is $\mathcal{O}(kn^2)$ for a rank-k approximation. When the centroid algorithm is executed on H rather than HH^T, then the number of expected iteration steps is $\frac{m}{2}$ for each centroid value. Note that the first step in Algorithm 8.1 involves an m-dimensional vector subtraction and an $\mathcal{O}(m\ell)$ matrix to vector multiply, the next

step an m-dimensional sorting, and an ℓ-dimensional vector addition in the last step. A rank-k centroid decomposition approximation to the rank-k truncated SVD of the scoring matrix H would involve $\mathcal{O}(km^2\ell)$ complexity. Obviously, the complexity may be further reduced if sparsity can be exploited.

9. Conclusions. We have recast the centroid method as an $\mathcal{O}(n)$-step optimization problem on a hypercube. This interpretation enables us to view the centroid method as a matrix approximation with many similarities to the truncated SVD.

Furthermore, we offer the insight that given any data matrix (with mean zero) whose columns represent random samples from a certain unknown distribution, its singular values then provide a measurement of the second order statistical information of the original data in the direction of the corresponding left singular vectors. This insight explains why, how, and when a low rank approximation can be used as a reasonable approximation to the original matrix. Although low rank approximation has been a common practice used in many important applications, we have not seen a satisfactory stochastic justification. There seems to be much misuse and misunderstanding of low rank approximation techniques.

We justify the truncated SVD as not only the nearest distancewise approximation, but also as the minimum-variance approximation to the original data. It seems fitting that any low rank approximation should carry stochastic properties similar to the truncated SVD. We have shown this to be true of the centroid decomposition empirically. Furthermore, we have shown that the centroid method can be generalized so that it might be used for many applications, e.g., the LSI problem.

Figure 9.1 can be viewed as a fundamental triad which includes the three equivalent variational formulations for the largest singular value of a matrix A.

The SDD method is analogous to the top vertex of the triad using only vectors $\mathbf{u}$ and $\mathbf{v}$, whose components are restricted to the set $\{0, 1, -1\}$. The centroid method

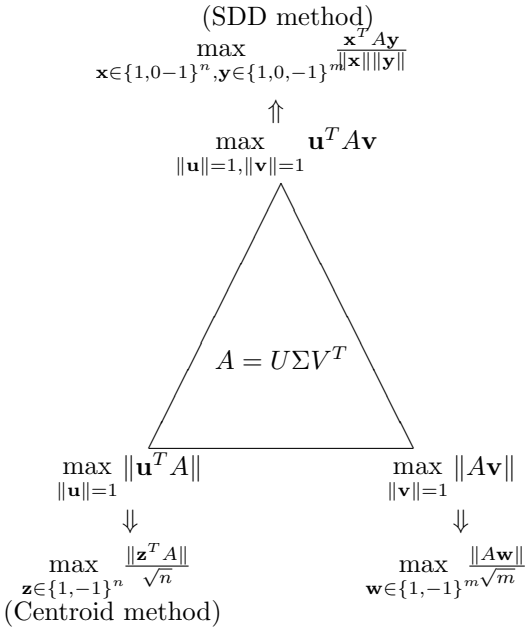


FIG. 9.1. *Fundamental SVD triad.*

is analogous to the left vertex of the triad, with the restriction that the vector $\mathbf{u}$ is allowed to have components only from the set $\{0, 1, -1\}$; note, however, that the inclusion of 0 does not give any additional discrete approximation because the corresponding objective function is convex and the maximum must occur at a sign vector (vertex). We call these sets *restricted*, whereas the vectors that determine the singular values are completely unrestricted. The form in the lower right corner stands ready to be analyzed in future work.

Finally, the important unification idea is that we can consider the centroid and semidiscrete methods as producing approximations to the truncated SVD using just two of the classes of many restricted discrete sets. We summarize relationships of centroid decomposition, SDD, and SVD in Table 9.1.

TABLE 9.1
Comparison of centroid decomposition, SVD, and SDD.

Decomposition		
Centroid	Singular value	Semidiscrete
$\mu = \frac{1}{n}\max_{\lvert\mathbf{z}\rvert=\mathbf{1}} \lVert\mathbf{z}^T A\rVert^2$ (centroid value)		
	$\sigma = \max_{\lVert\mathbf{u}\rVert=1} \lVert\mathbf{u}^T A\rVert$ (singular value)	
$\mathbf{v} = \frac{A^T\mathbf{z}}{\lVert A^T\mathbf{z}\rVert}$ (centroid factor)	$\mathbf{v} = \frac{A^T\mathbf{u}}{\lVert A^T\mathbf{u}\rVert}$ (right singular vector)	
	$\sigma = \max_{\lVert\mathbf{u}\rVert=\lVert\mathbf{v}\rVert=1} \lvert\mathbf{u}^T A\mathbf{v}\rvert$	$\delta = \max_{\lvert\mathbf{x}\rvert=\lvert\mathbf{y}\rvert\in\{1,0\}} \frac{\lvert\mathbf{x}^T A\mathbf{y}\rvert}{\lVert\mathbf{x}\rVert\lVert\mathbf{y}\rVert}$
	$\sigma = \max_{\lVert\mathbf{v}\rVert=1} \lVert A\mathbf{v}\rVert$	
$\mathbf{b} = A\mathbf{v}$ (loading vector)	$\sigma\mathbf{u} = A\mathbf{v}$ (internal relation)	
$\gamma = \lVert\mathbf{b}\rVert$ (significance)		
$A \approx (A\mathbf{v})\mathbf{v}^T$	$A \approx (A\mathbf{v})\mathbf{v}^T$	$A \approx (\delta\mathbf{x})\mathbf{y}^T$
$A_1 = \sum \mathbf{b}_i\mathbf{v}_i^T$ (CD)	$A_1 = \sum \sigma_i\mathbf{u}_i\mathbf{v}_i^T$ (SVD)	$A_1 = \sum \delta_i\mathbf{x}_i\mathbf{y}_i^T$ (SDD)
$\operatorname{rank}(A - \gamma\frac{\mathbf{b}}{\lVert\mathbf{b}\rVert}\mathbf{v}^T) =$ $\operatorname{rank}(A) - 1$	$\operatorname{rank}(A - \sigma\mathbf{u}\mathbf{v}^T) =$ $\operatorname{rank}(A) - 1$	(no rank subtraction)

Acknowledgment. The authors wish to give special thanks to the referees for important insights, suggestions, criticisms, and corrections.

REFERENCES

[1] M. ANDERBERG, *Cluster Analysis for Applications*, Academic Press, New York, 1973.
[2] C. BURT, *The Distribution and Relations of Educational Abilities*, P. S. King and Son, London, 1917.
[3] M. T. CHU, *On the Statistical Meaning of the Truncated Singular Decomposition*, manuscript.

[4] M. T. Chu, R. E. Funderlic, and G. H. Golub, *A rank-one reduction formula and its applications to matrix factorizations*, SIAM Rev., 37 (1995), pp. 512–530.
[5] A. L. Comrey and H. B. Lee, *A First Course in Factor Analysis*, Erlbaum Associates, Hillsdale, NJ, 1992.
[6] C. Eckert and G. Young, *The approximation of one matrix by another of lower rank*, Psych., 1 (1936), pp. 211–218.
[7] H. H. Harman, *Modern Factor Analysis*, University of Chicago Press, Chicago, 1967.
[8] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[9] P. Horst, *Factor Analysis of Data Matrices*, Holt, Rinehart and Winston, New York, 1965.
[10] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Ed. Psych., 24 (1933), pp. 417–441, 498–502.
[11] L. Hubert, J. Meulman, and W. Heiser, *Two purposes for matrix factorization: A historical appraisal*, SIAM Rev., 42 (2000), pp. 68–82.
[12] T. G. Kolda and D. P. O'Leary, *A semi-discrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Trans. Inform. Systems, 16 (1998), pp. 322–346.
[13] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1968.
[14] J. L. Melsa and D. L. Cohn, *Decision and Estimation Theory*, McGraw-Hill, New York, 1978.
[15] K. Pearson, *On lines and planes of closest fit to systems in space*, Phil. Mag., 6 (1901), pp. 559–572.
[16] G. W. Stewart, *On the early history of singular value decomposition*, SIAM Rev., 35 (1993), pp. 551–566.
[17] L. L. Thurston, *Multiple factor analysis*, Psych. Rev., 38 (1931), pp. 406–427.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1045–1069
© 2002 Society for Industrial and Applied Mathematics

EXISTENCE, UNIQUENESS, AND PARAMETRIZATION OF LAGRANGIAN INVARIANT SUBSPACES*

GERHARD FREILING†, VOLKER MEHRMANN‡, AND HONGGUO XU§

Abstract. The existence, uniqueness, and parametrization of Lagrangian invariant subspaces for Hamiltonian matrices is studied. Necessary and sufficient conditions and a complete parametrization are given.

Some necessary and sufficient conditions for the existence of Hermitian solutions of algebraic Riccati equations follow as simple corollaries.

Key words. eigenvalue problem, Hamiltonian matrix, symplectic matrix, Lagrangian invariant subspace, algebraic Riccati equation

AMS subject classifications. 65F15, 93B40, 93B36, 93C60

PII. S0895479800377228

1. Introduction.

The computation of invariant subspaces of Hamiltonian matrices is an important task in many applications in linear quadratic optimal and H_∞ control, Kalman filtering, or spectral factorization; see [13, 15, 20, 28] and the references therein.

DEFINITION 1.1. *A matrix $\mathcal{H} \in \mathbf{C}^{2n,2n}$ is called Hamiltonian if $J_n\mathcal{H} = (J_n\mathcal{H})^H$ is Hermitian, where $J_n = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$, I_n is the $n \times n$ identity matrix, and the superscript H denotes the conjugate transpose.*

Every Hamiltonian matrix $\mathcal{H}$ has the block form

$$\mathcal{H} = \begin{bmatrix} A & M \\ G & -A^H \end{bmatrix},$$

with $M = M^H$, $G = G^H$. Hamiltonian matrices are closely related to algebraic Riccati equations of the form

$$A^H X + XA - XMX + G = 0. \tag{1.1}$$

It is well known [15] that if $X = X^H$ solves (1.1), then

$$\mathcal{H}\begin{bmatrix} I_n & 0 \\ -X & I_n \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ -X & I_n \end{bmatrix}\begin{bmatrix} (A - MX) & M \\ 0 & -(A - MX)^H \end{bmatrix}. \tag{1.2}$$

This implies that the columns of $\left[\begin{smallmatrix} I_n \\ -X \end{smallmatrix}\right]$ span an invariant subspace of $\mathcal{H}$ associated with the eigenvalues of $A - MX$. Invariant subspaces of this form are called *graph subspaces* [15]. The graph subspaces of Hamiltonian matrices are special *Lagrangian subspaces*.

*Received by the editors August 29, 2000; accepted for publication by A.C.M. Ran May 3, 2001; published electronically April 10, 2002. The research of the second and third authors was supported by *Deutsche Forschungsgemeinschaft*, research grant Me 790/7-2.

http://www.siam.org/journals/simax/23-4/37722.html

†Fachbereich Mathematik, Universität Duisburg, D-47048 Duisburg, Germany (freiling@math.uni-duisburg.de).

‡Institut für Mathematik, MA 4-5, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, Germany (mehrmann@math.tu-berlin.de).

§Department of Mathematics, University of Kansas, Lawrence, KS 66045 (xu@math.ukans.edu).

DEFINITION 1.2. *A subspace* $\mathbf{L}$ *of* $\mathbf{C}^{2n}$ *is called a Lagrangian subspace if it has dimension* n *and*

$$x^H J_n y = 0 \quad \forall x, y \in \mathbf{L}.$$

Clearly a subspace $\mathbf{L}$ is Lagrangian if and only if every matrix L whose columns span $\mathbf{L}$ satisfies $\operatorname{rank} L = n$ and $L^H J_n L = 0$.

Despite the fact that Hamiltonian matrices, algebraic Riccati equations, and their properties have been a very active area of research for the last 40 years, there are still many open problems. These problems are mainly concerned with Hamiltonian matrices that have eigenvalues with zero real part and in particular with numerical methods for such problems.

In this paper we summarize and extend the known conditions for existence of Lagrangian invariant subspaces of a Hamiltonian matrix. Based on these results we then give a complete parametrization of all possible Lagrangian invariant subspaces and also discuss necessary and sufficient conditions for the uniqueness of Lagrangian invariant subspaces.

Most of the literature on this topic is stated in terms of Hermitian solutions for algebraic Riccati equations; see [15]. For several reasons we will, however, be mainly concerned with the characterization of Lagrangian invariant subspaces. First of all, the concept of Lagrangian invariant subspaces is a more general concept than that of Hermitian solutions of the Riccati equation, since only graph subspaces are associated with Riccati solutions. A second and more important reason is that in most applications the solution of the Riccati equation is not the primary goal, but rather a dangerous detour; see [21]. Finally, even most numerical solution methods for the solution of the algebraic Riccati equations (with the exception of Newton's method) proceed via the computation of Lagrangian invariant subspaces to determine the solution of the Riccati equation; see [3, 5, 6, 7, 8, 16, 17, 20, 27]. These methods employ transformations with symplectic matrices.

DEFINITION 1.3. *A matrix* $\mathcal{S} \in \mathbf{C}^{2n,2n}$ *is called symplectic if* $\mathcal{S}^H J_n \mathcal{S} = J_n$.

If $\mathcal{S}$ is symplectic, then by definition its first n columns span a Lagrangian subspace. Conversely, if the columns of S_1 span a Lagrangian subspace, then it generates a symplectic matrix, given, for example, by $\mathcal{S} = [S_1, J_n S_1 (S_1^H S_1)^{-1}]$. Hence the relation between Lagrangian subspaces and symplectic matrices can be summarized as follows.

PROPOSITION 1.4. *If* $\mathcal{S} \in \mathbf{C}^{2n,2n}$ *is symplectic, then the columns of* $\mathcal{S}\left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right]$ *span a Lagrangian subspace. If the columns of* $S_1 \in \mathbf{C}^{2n,n}$ *span a Lagrangian subspace, then there exists a symplectic* $\mathcal{S}$ *such that* $\operatorname{range} \mathcal{S}\left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right] = \operatorname{range} S_1$.

Considering Lagrangian invariant subspaces $\mathbf{L}$ of a Hamiltonian matrix $\mathcal{H}$, we immediately have the following important equivalence.

PROPOSITION 1.5. *Let* $\mathcal{H} \in \mathbf{C}^{2n,2n}$ *be a Hamiltonian matrix. There exists a Lagrangian invariant subspace* $\mathbf{L}$ *of* $\mathcal{H}$ *if and only if there exists a symplectic matrix* $\mathcal{S}$ *such that* $\operatorname{range} \mathcal{S}\left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right] = \mathbf{L}$ *and*

$$\mathcal{S}^{-1} \mathcal{H} \mathcal{S} = \begin{bmatrix} R & D \\ 0 & -R^H \end{bmatrix}. \tag{1.3}$$

The form (1.3) is called *Hamiltonian block triangular form*, and if furthermore R is upper triangular (or quasi-upper triangular in the real case), it is called *Hamiltonian triangular form* or *Hamiltonian Schur form*. Note that for the existence of Lagrangian

invariant subspaces it is not necessary that R in (1.3) is triangular if one is not interested in displaying actual eigenvalues. Most numerical methods, however, will return a Hamiltonian triangular or quasi-triangular form.

Necessary and sufficient conditions for the existence of such transformations were given in [18, 22] and in full generality in [19], and we will briefly summarize these conditions in the next section. Numerically backward stable methods to compute such forms have been developed in [1, 2, 3, 4].

The contents of this paper are summarized as follows. In section 2, after recalling some of the results on Hamiltonian triangular forms, we discuss the existence of Lagrangian invariant subspaces corresponding to all possible eigenvalue selections. In section 3 we give complete parametrizations of all possible Lagrangian subspaces of a Hamiltonian matrix associated with a particular set of eigenvalues. Based on these results we summarize necessary and sufficient conditions for the existence and uniqueness of Lagrangian invariant subspaces in section 4. Finally we apply these results to give some simple proofs of (mostly known) theorems on existence and uniqueness of Hermitian solutions to algebraic Riccati equations in section 5.

2. Hamiltonian block triangular forms and existence of Lagrangian invariant subspaces. To study an invariant subspace problem we first need to discuss the possible selection of associated eigenvalues.

We denote by $\Lambda(A)$ the spectrum of a square matrix A, counting multiplicities. For a Hamiltonian matrix, if $\lambda \in \Lambda(\mathcal{H})$ and $\operatorname{Re}\lambda \neq 0$, then it is easy to see that also $-\bar{\lambda} \in \Lambda(\mathcal{H})$; see [15, 20]. Furthermore, if $\mathcal{H}$ has the block triangular form (1.3) and if $i\alpha$ is a purely imaginary eigenvalue (including zero), then it must have even algebraic multiplicity. It follows that the spectrum of a Hamiltonian matrix $\mathcal{H}$ in the form (1.3) can be partitioned into two disjoint subsets,

$$\begin{aligned}\Lambda_1(\mathcal{H}) &= \{\underbrace{\lambda_1,\dots,\lambda_1}_{n_1},\underbrace{-\bar{\lambda}_1,\dots,-\bar{\lambda}_1}_{n_1},\dots,\underbrace{\lambda_\mu,\dots,\lambda_\mu}_{n_\mu},\underbrace{-\bar{\lambda}_\mu,\dots,-\bar{\lambda}_\mu}_{n_\mu}\},\\ \Lambda_2(\mathcal{H}) &= \{\underbrace{i\alpha_1,\dots,i\alpha_1}_{2m_1},\dots,\underbrace{i\alpha_\nu,\dots,i\alpha_\nu}_{2m_\nu}\},\end{aligned} \tag{2.1}$$

where $\lambda_1,\dots,\lambda_\mu$ are pairwise disjoint eigenvalues with positive real part and $i\alpha_1,\dots,i\alpha_\nu$ are pairwise disjoint purely imaginary eigenvalues (including zero).

If a matrix is transformed as in (1.3), then the spectrum associated with the Lagrangian invariant subspace spanned by the first n columns of $\mathcal{S}$ is $\Lambda(R)$. Since $\Lambda(\mathcal{H}) = \Lambda(R) \cup \Lambda(-R^H)$, it follows that $\Lambda(R)$ must be associated to a characteristic polynomial

$$\prod_{j=1}^{\mu}(\lambda-\lambda_j)^{t_j}(\lambda+\bar{\lambda}_j)^{n_j-t_j}\prod_{j=1}^{\nu}(\lambda-i\alpha_j)^{m_j},$$

where t_j are integers with $0 \le t_j \le n_j$ for $j = 1,\dots,\mu$. We denote the set of all possible such selections of eigenvalues by $\Omega(\mathcal{H})$. Note that $\Omega(\mathcal{H})$ contains $\prod_{j=1}^{\mu}(n_j+1)$ different selections.

In most applications it is desirable to determine Lagrangian invariant subspaces associated with eigenvalue selections for which only one of the eigenvalues of the pair $\lambda_j, -\bar{\lambda}_j$ (which are not purely imaginary) can be chosen in $\Lambda(\mathcal{R})$. In another words, t_j must be either 0 or n_j. Such subspaces all called *unmixed*, and the associated Riccati solution, if it exists, is called the *unmixed solution* of the Riccati equation; see

[26]. We denote the subset of all possible such selections by $\tilde{\Omega}(\mathcal{H})$. Obviously $\tilde{\Omega}(\mathcal{H})$ contains 2^μ different elements.

Note that all selections in $\Omega(\mathcal{H})$ contain the same purely imaginary eigenvalues. Note further that if $\mathcal{H}$ cannot be transformed to the Hamiltonian block triangular form (1.3), then the set $\Omega(\mathcal{H})$ may be empty. A simple example for this is the matrix J_1.

We now recall some results on the existence of Hamiltonian triangular forms. In the following we denote a single Jordan block associated with an eigenvalue λ by $N_r(\lambda) = \lambda I_r + N_r$, where N_r is a nilpotent Jordan block of size r. We also frequently use the antidiagonal matrices

$$P_r = \begin{bmatrix} & & & -1 \\ & & (-1)^2 & \\ & \iddots & & \\ (-1)^r & & & \end{bmatrix} \tag{2.2}$$

and denote by e_j the jth unit vector of appropriate size.

LEMMA 2.1 (see [19]). *Suppose that $i\alpha$ is a purely imaginary eigenvalue of a Hamiltonian matrix $\mathcal{H}$ and that the Jordan block structure associated with this eigenvalue is $N(i\alpha) := i\alpha I + N$, where*

$$N = \operatorname{diag}(N_{r_1}, \ldots, N_{r_s}).$$

Then there exists a full column rank matrix U such that $\mathcal{H}U = UN(i\alpha)$ and

$$U^H J_n U = \operatorname{diag}(\pi_1 P_{r_1}, \ldots, \pi_s P_{r_s}),$$

where $\pi_k \in \{1, -1\}$ if r_k is even and $\pi_k \in \{i, -i\}$ if r_k is odd.

Using the indices and matrices introduced in Lemma 2.1, the *structure inertia index* associated with the eigenvalue $i\alpha$ is defined as

$$\operatorname{Ind}_S(i\alpha) = \{\beta_1, \ldots, \beta_s\},$$

where $\beta_k = (-1)^{\frac{r_k}{2}}\pi_k$ if r_k is even, and $\beta_k = (-1)^{\frac{r_k-1}{2}} i\pi_k$ if r_k is odd. Note that the β_i are all ± 1 and there is one index associated with every Jordan block. The structure inertia index is closely related to the well-known sign characteristic for Hermitian pencils (see [15]), since every Hamiltonian matrix $\mathcal{H}$ can be associated with the Hermitian pencil $\lambda iJ - J\mathcal{H}$. Although the sign characteristic is a more general concept since it also applies to general Hermitian pencils, we prefer to use the structure inertia index, because it is better suited for the analysis of Hamiltonian triangular forms; see [19].

For the following analysis the tuple $\operatorname{Ind}_S(i\alpha)$ is partitioned into three parts, $\operatorname{Ind}_S^e(i\alpha)$, $\operatorname{Ind}_S^c(i\alpha)$, $\operatorname{Ind}_S^d(i\alpha)$, where $\operatorname{Ind}_S^e(i\alpha)$ contains all the structure inertia indices corresponding to even r_k, $\operatorname{Ind}_S^c(i\alpha)$ contains the maximal number of structure inertia indices corresponding to odd r_k in ± 1 pairs, and $\operatorname{Ind}_S^d(i\alpha)$ contains the remaining indices; i.e., all indices in $\operatorname{Ind}_S^d(i\alpha)$ have the same sign; see [19].

Necessary and sufficient conditions for the existence of a symplectic similarity transformation to a Hamiltonian triangular Jordan-like form (1.3) are given in the following theorem.

THEOREM 2.2. *Let $\mathcal{H}$ be a Hamiltonian matrix, let $i\alpha_1, \ldots, i\alpha_\nu$ be its pairwise distinct purely imaginary eigenvalues, and let the columns of U_k, $k = 1, \ldots, \nu$, span the associated invariant subspaces of dimension m_k. Then the following are equivalent:*

(i) *There exists a symplectic matrix* $\mathcal{S}$ *such that* $\mathcal{S}^{-1}\mathcal{H}\mathcal{S}$ *is Hamiltonian block triangular.*
(ii) *There exists a unitary symplectic matrix* $\mathcal{U}$ *such that* $\mathcal{U}^H\mathcal{H}\mathcal{U}$ *is Hamiltonian block triangular.*
(iii) $U_k^H J U_k$ *is congruent to* J_{m_k} *for all* $k = 1, \ldots, \nu$.
(iv) $\mathrm{Ind}_S^d(i\alpha_k)$ *is void for all* $k = 1, \ldots, \nu$.

Moreover, if any of the equivalent conditions holds, then the symplectic matrix $\mathcal{S}$ *can be chosen such that* $\mathcal{S}^{-1}\mathcal{H}\mathcal{S}$ *is in Hamiltonian triangular Jordan form*

$$\begin{bmatrix} R_r & 0 & 0 & 0 & 0 & 0 \\ 0 & R_e & 0 & 0 & D_e & 0 \\ 0 & 0 & R_c & 0 & 0 & D_c \\ 0 & 0 & 0 & -R_r^H & 0 & 0 \\ 0 & 0 & 0 & 0 & -R_e^H & 0 \\ 0 & 0 & 0 & 0 & 0 & -R_c^H \end{bmatrix}, \tag{2.3}$$

where the blocks with subscript r *are associated with eigenvalues of nonzero real part and have the substructure*

$$R_r = \mathrm{diag}(R_1^r, \ldots, R_\mu^r), \quad R_k^r = \mathrm{diag}(N_{d_{k,1}}(\lambda_k), \ldots, N_{d_{k,p_k}}(\lambda_k)), \quad k = 1, \ldots, \mu.$$

The blocks with subscript e *are associated with the structure inertia indices of even* r_k *for all purely imaginary eigenvalues and have the substructure*

$$\begin{aligned} R_e &= \mathrm{diag}(R_1^e, \ldots, R_\nu^e), \quad R_k^e = \mathrm{diag}(N_{l_{k,1}}(i\alpha_k), \ldots, N_{l_{k,q_k}}(i\alpha_k)), \\ D_e &= \mathrm{diag}(D_1^e, \ldots, D_\nu^e), \quad D_k^e = \mathrm{diag}(\beta_{k,1}^e e_{l_{k,1}} e_{l_{k,1}}^H, \ldots, \beta_{k,q_k}^e e_{l_{k,q_k}} e_{l_{k,q_k}}^H). \end{aligned}$$

The blocks with subscript c *are associated with pairs of blocks of inertia indices associated with odd-sized blocks for purely imaginary eigenvalues and have the substructure*

$$\begin{aligned} R_c &= \mathrm{diag}(R_1^c, \ldots, R_\nu^c), \quad R_k^c = \mathrm{diag}(B_{k,1}, \ldots, B_{k,r_k}), \\ D_c &= \mathrm{diag}(D_1^c, \ldots, D_\nu^c), \quad D_k^c = \mathrm{diag}(C_{k,1}, \ldots, C_{k,r_k}), \end{aligned}$$

where

$$B_{k,j} = \begin{bmatrix} N_{m_{k,j}}(i\alpha_k) & 0 & -\frac{\sqrt{2}}{2} e_{m_{k,j}} \\ 0 & N_{n_{k,j}}(i\alpha_k) & -\frac{\sqrt{2}}{2} e_{n_{k,j}} \\ 0 & 0 & i\alpha_k \end{bmatrix},$$

$$C_{k,j} = \frac{\sqrt{2}}{2} i\beta_{k,j}^c \begin{bmatrix} 0 & 0 & e_{m_{k,j}} \\ 0 & 0 & -e_{n_{k,j}} \\ -e_{m_{k,j}}^H & e_{n_{k,j}}^H & 0 \end{bmatrix}.$$

Proof. The proof of equivalence for (i) and (iv) is given in Theorem 1.3 in [25]. The equivalence of the other conditions and the structured Hamiltonian triangular Jordan form (2.3) was derived in [19]. ∎

Remark 1. For real Hamiltonian matrices a real quasi-triangular Jordan form analogous to (2.3) and a similar set of equivalent conditions as in Theorem 2.2 can be given. We refer the reader to [25] and Theorem 24 in [19] for details.

The necessary and sufficient conditions in Theorem 2.2 guarantee the existence of only *one* Lagrangian invariant subspace associated to *one* selection in $\Omega(\mathcal{H})$. But the

following theorem shows they also guarantee the existence of a Lagrangian invariant subspace associated to *every* selection in $\Omega(\mathcal{H})$.

THEOREM 2.3. *Let $\mathcal{H}$ be a Hamiltonian matrix. If any of the conditions in Theorem 2.2 holds, then for every eigenvalue selection $\omega \in \Omega(\mathcal{H})$ there exists at least one corresponding Lagrangian invariant subspace.*

Proof. A proof for this result based on condition (iv) was given in [23, 25], but a simple proof follows directly from (2.3). Note that any ω contains half the number of eigenvalues for every purely imaginary eigenvalue. So a basis for a corresponding invariant subspace is easily determined from (2.3). For an eigenvalue pair $\lambda_k, -\bar{\lambda}_k$ we need to consider only the small Hamiltonian block $\left[\begin{smallmatrix} R_k^r & 0 \\ 0 & -(R_k^r)^H \end{smallmatrix}\right]$. Note that R_k^r is upper triangular. Suppose that the selection ω contains t_k copies of λ_k and s_k copies of $-\bar{\lambda}_k$. A corresponding basis of the invariant subspace can then be chosen based on a symplectic permutation which exchanges trailing $s_k \times s_k$ blocks in R_k^r and $-(R_k^r)^H$. ∎

In this section we have reviewed some results on the existence of (unitary) symplectic transformations to Hamiltonian block triangular form and the existence of Lagrangian invariant subspaces. In the next section we use these results to give a full parametrization of all possible Lagrangian subspaces and therefore also a parametrization of all symplectic similarity transformations to Hamiltonian block triangular form.

3. Parametrization of all Lagrangian invariant subspaces. In the previous section we have shown that if $\mathcal{H}$ has a Hamiltonian block triangular form, then for every eigenvalue selection $\omega \in \Omega(\mathcal{H})$ there exists at least one corresponding invariant subspace. In this section we will parametrize all possible Lagrangian invariant subspaces associated to a given selection ω.

For this we will need some technical lemmas.

LEMMA 3.1. *Consider pairs of matrices $(\pi_k P_{r_k}, N_{r_k})$, $k = 1, 2$, where r_1, r_2 are either both even or both odd. Let $\pi_1, \pi_2 \in \{1, -1\}$ if both r_k are even and $\pi_1, \pi_2 \in \{i, -i\}$ if both r_k are odd; let*

$$(\mathcal{P}_c, \mathcal{N}_c) := \left(\begin{bmatrix} \pi_1 P_{r_1} & 0 \\ 0 & \pi_2 P_{r_2} \end{bmatrix}, \begin{bmatrix} N_{r_1} & 0 \\ 0 & N_{r_2} \end{bmatrix} \right);$$

and let $d := \frac{|r_1 - r_2|}{2}$. If $\pi_1 = (-1)^{d+1}\pi_2$, i.e., $\beta_1 = -\beta_2$ for the corresponding β_1 and β_2, then we have the following transformations to Hamiltonian triangular form:

1. *If $r_1 \geq r_2$, then with*

$$Z_1 := \begin{bmatrix} I_d & 0 & 0 & 0 \\ 0 & I_{r_2} & 0 & -\frac{1}{2}\bar{\pi}_2 P_{r_2}^{-1} \\ 0 & 0 & \bar{\pi}_1 P_d^{-1} & 0 \\ 0 & -I_{r_2} & 0 & -\frac{1}{2}\bar{\pi}_2 P_{r_2}^{-1} \end{bmatrix}$$

we obtain $Z_1^H \mathcal{P}_c Z_1 = J_{\frac{r_1+r_2}{2}}$ and

$$Z_1^{-1} \mathcal{N}_c Z_1 = \begin{bmatrix} N_{\frac{r_1+r_2}{2}} & D \\ 0 & -N_{\frac{r_1+r_2}{2}}^H \end{bmatrix},$$

where $D = \tau e_d e_{\frac{r_1+r_2}{2}}^H + \bar{\tau} e_{\frac{r_1+r_2}{2}} e_d^H$, $\tau = -\frac{1}{2}\pi_2$.

2. *If $r_1 < r_2$, then with*

$$Z_2 = \begin{bmatrix} \pi_1 P_{r_1} & 0 & \frac{1}{2} I_{r_1} & 0 \\ 0 & \pi_2 P_d & 0 & 0 \\ -\pi_1 P_{r_1} & 0 & \frac{1}{2} I_{r_1} & 0 \\ 0 & 0 & 0 & I_d \end{bmatrix}$$

we obtain that $Z_2^H \mathcal{P}_c Z_2 = J_{\frac{r_1+r_2}{2}}$ and

$$Z_2^{-1} \mathcal{N}_c Z_2 = \begin{bmatrix} -N^H_{\frac{r_1+r_2}{2}} & D \\ 0 & N^H_{\frac{r_1+r_2}{2}} \end{bmatrix},$$

where $D = \tau e_1 e^H_{r_1+1} + \bar{\tau}_1 e_{r_1+1} e_1^H$, $\tau = -\frac{1}{2}\pi_1$.

Proof. The proof is a simple modification of the proof of Lemma 18 in [19]. ∎

LEMMA 3.2. *Consider a nilpotent matrix in Jordan form $N = \operatorname{diag}(N_{r_1}, \ldots, N_{r_p})$.*

(i) *If the columns of the full column rank matrix X form an invariant subspace of N, i.e., $NX = XA$ for some matrix A, then $X = UZ$, where Z is nonsingular and*

$$U = \begin{bmatrix} I_{t_1} & 0 & \ldots & 0 & 0 \\ 0 & V_{1,2} & \ldots & V_{1,p-1} & V_{1,p} \\ 0 & I_{t_2} & \ldots & 0 & 0 \\ 0 & 0 & \ldots & V_{2,p-1} & V_{2p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & I_{t_{p-1}} & 0 \\ 0 & 0 & \ldots & 0 & V_{p-1,p} \\ 0 & 0 & \ldots & 0 & I_{t_p} \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix}. \tag{3.1}$$

Here for $k = 1, \ldots, p$, $0 \le t_k \le r_k$, and for $i = 1, \ldots, p-1$ and $j = i+1, \ldots, p$, we have $V_{i,j} \in \mathbf{C}^{s_i, t_j}$ with $s_i = r_i - t_i$. Moreover, if $M_s = \operatorname{diag}(N_{s_1}, \ldots, N_{s_p})$, $M_t = \operatorname{diag}(N_{t_1}, \ldots, N_{t_p})$, and $E = \operatorname{diag}(e_{t_1} e_1^H, \ldots, e_{t_p} e_1^H)$, then

$$V = \begin{bmatrix} 0 & V_{12} & \ldots & V_{1p} \\ & \ddots & \ddots & \vdots \\ & & \ddots & V_{p-1,p} \\ & & & 0 \end{bmatrix}$$

satisfies the algebraic Riccati equation

$$M_s V - V M_t - V E V = 0.$$

(ii) *If the columns of the full column rank matrix X form an invariant subspace of N^H, i.e., $N^H X = -XA$ for some matrix A, then $X = \hat{U} Z$, where Z is*

nonsingular and

$$\hat{U} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ -I_{\hat{t}_1} & 0 & \dots & & 0 \\ \hat{V}_{2,1} & 0 & \dots & 0 & 0 \\ 0 & -I_{\hat{t}_2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{V}_{p-1,1} & \hat{V}_{p-1,2} & \dots & 0 & 0 \\ 0 & 0 & \dots & -I_{\hat{t}_{p-1}} & 0 \\ \hat{V}_{p,1} & \hat{V}_{p,2} & \dots & \hat{V}_{p,p-1} & 0 \\ 0 & 0 & \dots & 0 & -I_{\hat{t}_p} \end{bmatrix}. \tag{3.2}$$

Here for $k = 1, \dots, p$, $0 \le \hat{t}_k \le r_k$, and for $i = 2, \dots, p$ and $j = 1, \dots, i-1$, we have $\hat{V}_{i,j} \in \mathbf{C}^{\hat{s}_i, \hat{t}_j}$ with $\hat{s}_i = r_i - \hat{t}_i$. Moreover, if $M_{\hat{s}} = \operatorname{diag}(N_{\hat{s}_1}, \dots, N_{\hat{s}_p})$, $M_{\hat{t}} = \operatorname{diag}(N_{\hat{t}_1}, \dots, N_{\hat{t}_p})$, and $E = \operatorname{diag}(e_1 e_{\hat{s}_1}^H, \dots, e_1 e_{\hat{s}_p}^H)$, then

$$\hat{V} = \begin{bmatrix} 0 & & & \\ \hat{V}_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \hat{V}_{p1} & \dots & \hat{V}_{p,p-1} & 0 \end{bmatrix}$$

satisfies the algebraic Riccati equation

$$M_{\hat{s}}^H \hat{V} - \hat{V} M_{\hat{t}}^H - \hat{V} E \hat{V} = 0.$$

Proof. We will derive the structure of X by multiplying nonsingular matrices to X from the right. Let us first prove part (i). Partition $X = \left[\begin{smallmatrix} X_1 \\ X_2 \end{smallmatrix}\right]$ so that X_2 has r_p rows. Then using the QR or singular value decomposition [12], there exists a nonsingular (actually unitary) matrix Y_1 such that $X_2 = [0, X_{22}]Y_1$, where $X_{22} \in \mathbf{C}^{r_p, t_p}$ and $\operatorname{rank} X_{22} = t_p$. (This implies that $0 \le t_p \le r_p$.) Then we have the partition $\hat{X} = XY_1^{-1} = \left[\begin{smallmatrix} X_{11} & X_{12} \\ 0 & X_{22} \end{smallmatrix}\right]$. Since $\operatorname{range} X$ is an invariant subspace of N, so is $\operatorname{range} \hat{X}$. Hence, there exists a matrix $\hat{A}$ such that

$$N\hat{X} = \hat{X}\hat{A}. \tag{3.3}$$

If we partition $\hat{A} = \left[\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right]$ conformally with $\hat{X}$, then (3.3) implies that $A_{21} = 0$ and $N_{r_p} X_{22} = X_{22} A_{22}$. Because X_{22} has full column rank and N_{r_p} is a single Jordan block, it is clear that A_{22} is similar to N_{t_p}, i.e., there exists a nonsingular matrix Y_{22} such that $Y_{22}^{-1} A_{22} Y_{22} = N_{t_p}$, and hence $N_{r_p}(X_{22}Y_{22}) = (X_{22}Y_{22})N_{t_p}$. By Lemma 4.4.11 in [14], $X_{22}Y_{22} = \left[\begin{smallmatrix} T \\ 0 \end{smallmatrix}\right]$, where T is an upper triangular Toeplitz matrix and T must be nonsingular, since X_{22} has full column rank. Therefore, by setting $\tilde{X} = \hat{X}Y_2$ with $Y_2 = \operatorname{diag}(I, Y_{22})T^{-1}$, it follows that

$$\tilde{X} = \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 \\ 0 & I_{t_p} \\ 0 & 0 \end{bmatrix},$$

and (3.3) becomes $N\tilde{X} = \tilde{X}\left[\begin{smallmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & N_{t_p} \end{smallmatrix}\right]$. Setting $\tilde{N} = \operatorname{diag}(N_{r_1}, \dots, N_{r_{p-1}})$ it follows that $\tilde{N}\tilde{X}_1 = \tilde{X}_1 \tilde{A}_{11}$, and since X has full column rank, $\tilde{X}_1$ also has full column rank.

By inductively applying the construction that leads from X to $\tilde{X}$, we determine a nonsingular matrix Z_1 such that $XZ_1^{-1} = \breve{X}$, where $\breve{X}$ has the block structure

$$\breve{X} = \begin{bmatrix} I_{t_1} & W_{1,2} & \dots & W_{1,p-1} & W_{1,p} \\ 0 & V_{1,2} & \dots & V_{1,p-1} & V_{1,p} \\ 0 & I_{t_2} & \dots & W_{2,p-1} & W_{2,p} \\ 0 & 0 & \dots & V_{2,p-1} & V_{2,p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_{t_{p-1}} & W_{p-1,p} \\ 0 & 0 & \dots & 0 & V_{p-1,p} \\ 0 & 0 & \dots & 0 & I_{t_p} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

with $0 \le t_i \le r_i$. The blocks $W_{i,j}$ in $\breve{X}$ can be eliminated by performing a sequence of block Gaussian type eliminations from the right. Hence, there exists a nonsingular matrix Z_2 such that $\breve{X}Z_2^{-1} = U$, where U is in (3.1). Therefore, by setting $Z := Z_2Z_1$ we have $X = UZ$.

From the block form of U we can determine a block permutation matrix Q such that $QU = \left[\begin{smallmatrix} I \\ V \end{smallmatrix}\right]$ and $QNQ^{-1} = \left[\begin{smallmatrix} M_t & E \\ 0 & M_s \end{smallmatrix}\right]$. Since $\left[\begin{smallmatrix} I \\ V \end{smallmatrix}\right]$ is invariant to QNQ^{-1}, we have $M_sV - VM_t - VEV = 0$.

Part (ii) is proved analogously by beginning the reduction from the top and compressing in each step to the left. ∎

Using these lemmas we are able to parametrize the set of all Lagrangian invariant subspaces of a Hamiltonian matrix $\mathcal{H}$ associated with a fixed eigenvalue selection in $\omega \in \Omega(\mathcal{H})$. Let $\mathcal{H}$ be in Hamiltonian block triangular form (1.3) and let the spectrum of $\mathcal{H}$ be as in (2.1). Then (see [19]) there exists a symplectic matrix $\mathcal{S}$ such that $\mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \left[\begin{smallmatrix} R & D \\ 0 & -R^H \end{smallmatrix}\right]$, where $R = \operatorname{diag}(R_1, \dots, R_{\mu+\nu})$ and $D = \operatorname{diag}(D_1, \dots, D_{\mu+\nu})$. Furthermore, the blocks are reordered such that $H_k := \left[\begin{smallmatrix} R_k & D_k \\ 0 & -R_k^H \end{smallmatrix}\right]$ is Hamiltonian block triangular and associated with an eigenvalue pair $\lambda_k, -\bar{\lambda}_k$ with nonzero real part for $k = 1, \dots, \mu$ and purely imaginary eigenvalues $i\alpha_k$ for $k = \mu+1, \dots, \mu+\nu$. Furthermore, $\Lambda(R) = \omega$ and $\operatorname{range} \mathcal{S}\left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right] = \mathbf{L}$.

For this block diagonal form there exists a block permutation matrix $\mathcal{P}$ such that

$$\begin{aligned} \mathcal{P}^H J \mathcal{P} &= \operatorname{diag}(J_{n_1}, \dots, J_{n_\mu}; J_{m_1}, \dots, J_{m_\nu}) =: \tilde{J}, \\ \mathcal{P}^{-1}\mathcal{S}^{-1}\mathcal{H}\mathcal{S}\mathcal{P} &= \operatorname{diag}(H_1, \dots, H_\mu; H_{\mu+1}, \dots, H_{\mu+\nu}). \end{aligned} \tag{3.4}$$

Suppose that there exists another Lagrangian invariant subspace $\tilde{\mathbf{L}}$ corresponding to ω. Using the same argument, there exists a symplectic matrix $\tilde{\mathcal{S}}$ such that for the same block permutation matrix $\mathcal{P}$ we have

$$\mathcal{P}^{-1}\tilde{\mathcal{S}}^{-1}\mathcal{H}\tilde{\mathcal{S}}\mathcal{P} = \operatorname{diag}(\tilde{H}_1, \dots, \tilde{H}_\mu; \tilde{H}_{\mu+1}, \dots, \tilde{H}_{\mu+\nu}),$$

where again all $\tilde{H}_k$ are Hamiltonian block triangular and $\Lambda(\tilde{H}_k) = \Lambda(H_k)$ for all $k = 1, \dots, \mu+\nu$. Therefore, we have $\tilde{\mathcal{S}}\mathcal{P} = \mathcal{S}\mathcal{P}\mathcal{E}$ for some block diagonal matrix $\mathcal{E} = \operatorname{diag}(E_1, \dots, E_{\mu+\nu})$ satisfying $H_kE_k = E_k\tilde{H}_k$. Since $\mathcal{P}^H J \mathcal{P} = \tilde{J}$ and since $\mathcal{S}$ and $\tilde{\mathcal{S}}$ are symplectic, it follows that $\mathcal{E} = \mathcal{P}^{-1}\mathcal{S}^{-1}\tilde{\mathcal{S}}\mathcal{P}$ satisfies $\mathcal{E}^H\tilde{J}\mathcal{E} = \tilde{J}$, which implies that all blocks E_k are symplectic. Since $\tilde{\mathcal{S}} = \mathcal{S}\mathcal{P}\mathcal{E}\mathcal{P}^{-1}$, the difference between $\tilde{\mathcal{S}}$ and $\mathcal{S}$ (and therefore $\tilde{\mathbf{L}}$ and $\mathbf{L}$) is completely described by the first half of the columns of the symplectic matrices E_k, i.e., the Lagrangian invariant subspaces of the small

Hamiltonian matrices H_k (note that all $\tilde{H}$ are Hamiltonian block triangular). Following this argument, it is sufficient to parametrize all possible Lagrangian invariant subspaces of a Hamiltonian matrix with either a single purely imaginary eigenvalue $i\alpha$ or a single eigenvalue pair $\lambda, -\bar{\lambda}$ with $\operatorname{Re}\lambda \neq 0$.

Consider first the case of a single purely imaginary eigenvalue. In this case $\Omega(\mathcal{H})$ has only one element. So all Lagrangian invariant subspaces are associated to the same eigenvalue.

To simplify our analysis we need the following Hamiltonian Jordan form.

LEMMA 3.3. *Let $\mathcal{H}$ be a Hamiltonian matrix that has only one eigenvalue $i\alpha$. Then there exists a symplectic matrix $\mathcal{S}$ such that*

$$\mathcal{R} := \mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \begin{bmatrix} N(i\alpha) & D \\ 0 & -N(i\alpha)^H \end{bmatrix}, \tag{3.5}$$

where $N = \operatorname{diag}(N_{r_1}, \ldots, N_{r_p})$, $D = \operatorname{diag}(D_1, \ldots, D_p)$. Here either $D_j = \beta_j^e e_{r_j} e_{r_j}^H$, so that $\mathcal{H}$ has a Jordan block N_{2r_j} with structure inertia index $\beta_j^e \in \{1, -1\}$, or $D_j = \tau_j e_{d_j} e_{r_j}^H + \bar{\tau}_j e_{r_j} e_{d_j}^H$ with $\tau_j = \frac{1}{2}(-1)^{\frac{r_j+d_j+1}{2}} i\beta_j$ if $r_j + d_j$ is odd, and $\tau_j = \frac{1}{2}(-1)^{\frac{r_j+d_j}{2}}\beta_j$ if $r_j + d_j$ is even for some $\beta_j \in \{-1, 1\}$, so that $\mathcal{H}$ has two Jordan blocks $N_{r_j+d_j}$, $N_{r_j-d_j}$ with structure inertia indices β_j, $-\beta_j$, respectively.

Proof. Since $\mathcal{H} - i\alpha I$ is Hamiltonian, we may without loss of generality (w.l.o.g.) consider the problem with $\alpha = 0$, i.e., $\mathcal{H}$ that has only the eigenvalue zero. Since $\mathcal{H}$ has only one multiple eigenvalue, the columns of every nonsingular matrix span a corresponding invariant subspace so that condition (iii) of Theorem 2.2 holds. The canonical form (3.5) then is obtained in a similar way as for (2.3); see [19]. The only difference is that here we match all possible pairs of Jordan block with opposite structure inertia indices in such a way that even blocks are matched with even blocks, and odd blocks with odd blocks, and furthermore the blocks are ordered in decreasing size. Finally we use the technique given in Lemma 3.1. ∎

The complete parametrization is then as follows.

THEOREM 3.4. *Let $\mathcal{H}$ be a Hamiltonian matrix that has only one purely imaginary eigenvalue. Let $\mathcal{S}$ be symplectic such that $\mathcal{S}^{-1}\mathcal{H}\mathcal{S}$ is in Hamiltonian canonical form* (3.5). *Then all possible Lagrangian subspaces can be parametrized by* $\operatorname{range}\mathcal{S}U$, *where*

$$U = \begin{bmatrix}
I_{t_1} & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & V_{12} & \ldots & V_{1,p-1} & V_{1p} & W_{11} & \ldots & W_{1,p-1} & W_{1p} \\
0 & I_{t_2} & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & V_{2,p-1} & V_{2p} & W_{12}^H & \ldots & W_{2,p-1} & W_{2p} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & I_{t_{p-1}} & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & V_{p-1,p} & W_{1,p-1}^H & \ldots & W_{p-1,p-1} & W_{p-1,p} \\
0 & 0 & \ldots & 0 & I_{t_p} & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 & W_{1p}^H & \ldots & W_{p-1,p}^H & W_{pp} \\
0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 & -I_{s_1} & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 & 0 & V_{12}^H & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 0 & V_{1,p-1}^H & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 & 0 & \ldots & -I_{s_{p-1}} & 0 \\
0 & 0 & \ldots & 0 & 0 & V_{1p}^H & \ldots & V_{p-1,p}^H & 0 \\
0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & -I_{s_p}
\end{bmatrix}, \tag{3.6}$$

with block sizes $0 \le s_j, t_j \le r_j$ *and* $s_j + t_k = r_j$. *Then, setting*

$$\begin{aligned} M_t &= \operatorname{diag}(N_{t_1}, \ldots, N_{t_p}), \\ M_s &= \operatorname{diag}(N_{s_1}, \ldots, N_{s_p}), \\ E &= \operatorname{diag}(e_{t_1} e_1^H, \ldots, e_{t_p} e_1^H), \end{aligned}$$

partitioning the Hermitian blocks

$$D_j = \begin{bmatrix} G_j & F_j \\ F_j^H & K_j \end{bmatrix},$$

and setting

$$\begin{aligned} K &= \operatorname{diag}(K_1, \ldots, K_s), \\ F &= \operatorname{diag}(F_1, \ldots, F_s), \\ G &= \operatorname{diag}(G_1, \ldots, G_s), \end{aligned}$$

it follows that the block matrices

$$V := \begin{bmatrix} 0 & V_{1,2} & \cdots & V_{1,p} \\ & \ddots & \ddots & \vdots \\ & & \ddots & V_{p-1,p} \\ & & & 0 \end{bmatrix}, \quad W := \begin{bmatrix} W_{1,1} & \cdots & W_{1,p} \\ \vdots & \ddots & \vdots \\ W_{1,p}^H & \cdots & W_{p,p} \end{bmatrix} = W^H$$

satisfy

$$\begin{aligned} &\begin{bmatrix} M_s & F^H \\ 0 & -M_t^H \end{bmatrix} \begin{bmatrix} W & V \\ V^H & 0 \end{bmatrix} + \begin{bmatrix} W & V \\ V^H & 0 \end{bmatrix} \begin{bmatrix} M_s^H & 0 \\ F & -M_t \end{bmatrix} \\ &\quad - \begin{bmatrix} W & V \\ V^H & 0 \end{bmatrix} \begin{bmatrix} 0 & E^H \\ E & G \end{bmatrix} \begin{bmatrix} W & V \\ V^H & 0 \end{bmatrix} - \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} = 0, \end{aligned} \tag{3.7}$$

or equivalently V, W *satisfy*

$$0 = M_s V - V M_t - V E V, \tag{3.8}$$

$$\begin{aligned} 0 = {} & (M_s - VE)W + W(M_s - VE)^H \\ & + (VF)^H + VF - VGV^H - K. \end{aligned} \tag{3.9}$$

Every Lagrangian invariant subspace is uniquely determined by a set of parameters $t_1, \ldots, t_p$ *with* $0 \le t_j \le r_j$ *and a set of matrices* $V_{i,j}$, $i = 1, \ldots, p-1$, $j = i+1, \ldots, p$, *and* $W_{i,j}$, $i = 1, \ldots, p$, $j = i, \ldots, p$, *satisfying* (3.8) *and* (3.9).

Moreover, all symplectic matrices that transform $\mathcal{H}$ *to Hamiltonian block triangular form can be parametrized as* $\mathcal{SUY}$, *where* $\mathcal{Y}$ *is a symplectic block triangular matrix,*

$\mathcal{U} = [U, \tilde{U}]$, *with* U *as in* (3.6), *and*

$$\tilde{U} = \begin{bmatrix}
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & I_{t_1} & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & I_{t_2} & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & I_{t_{p-1}} & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & I_{t_p} \\
I_{s_1} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & I_{s_2} & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & I_{s_{p-1}} & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & I_{s_p} & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0
\end{bmatrix}. \tag{3.10}$$

Proof. As in Lemma 3.3, we assume that the only eigenvalue of $\mathcal{H}$ is zero. Considering the form (3.5), it is sufficient to prove that every basis X of a Lagrangian invariant subspace of $\mathcal{R}$ can be expressed as $X = UY$. To prove this, we first compress the bottom square block of X, i.e., we determine a matrix Y_1 such that $XY_1 = \left[\begin{smallmatrix} X_{11} & X_{12} \\ 0 & X_{22} \end{smallmatrix}\right]$ where X_{22} has full column rank. Obviously X_{11} also has full column rank. Then, since XY_1 is still a basis of an invariant subspace of $\mathcal{R}$, the block triangular form of $\mathcal{R}$ implies that the columns of X_{11} and X_{22} form bases of the invariant subspace of N and $-N^H$, respectively. Applying Lemma 3.2, there exist matrices Z_1 and Z_2 such that $U_{11} := X_{11}Z_1$ and $U_{22} := X_{22}Z_2$ have structures as the matrices in (3.1) and (3.2) associated with the integer parameters $t_1, \dots, t_p$ and $\hat{t}_1, \dots, \hat{t}_p$, respectively. Now let $U := XY_1 \left[\begin{smallmatrix} Z_1 & 0 \\ 0 & Z_2 \end{smallmatrix}\right] Y_2 = \left[\begin{smallmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{smallmatrix}\right]$, where Y_2 is used to eliminate the blocks in $X_{12}Z_2$ using the identity blocks in $U_{1,1}$. Since X, and hence also U, is Lagrangian, we have that $U_{11}^H U_{22} = 0$. Thus, we have $\hat{t}_j = m_j - t_j =: s_j$ for all $j = 1, \dots, p$ and $\hat{V}_{i,j} = V_{j,i}^H$ for all $i = 2, \dots, p$, $j = 1, \dots, p-1$. Since $U_{12}^H U_{22}$ is Hermitian, it follows that U_{12} has the desired form. To prove (3.7), as in the proof of Lemma 3.2, there exists a block permutation matrix P such that $P[U_{11}, U_{12}] = \left[\begin{smallmatrix} I & 0 \\ V & W \end{smallmatrix}\right]$. Let $\tilde{\mathcal{P}} = \operatorname{diag}(P, P)$, which is symplectic. Then

$$\tilde{\mathcal{P}}U = \begin{bmatrix} I & 0 \\ V & W \\ 0 & V^H \\ 0 & -I \end{bmatrix}, \quad \tilde{\mathcal{P}}^{-1}\mathcal{R}\tilde{\mathcal{P}} = \begin{bmatrix} M_t & E & G & F \\ 0 & M_s & F^H & K \\ 0 & 0 & -N_t^H & 0 \\ 0 & 0 & -E^H & -N_s^H \end{bmatrix}.$$

Since the columns of $\tilde{\mathcal{P}}U$ form an invariant subspace for $\tilde{\mathcal{P}}^{-1}\mathcal{R}\tilde{\mathcal{P}}$, it follows that the matrices V, W satisfy (3.7). Conditions (3.8) and (3.9) follow directly from (3.7). To show the uniqueness of a particular Lagrangian invariant subspace, suppose that there are two matrices U_1, U_2 of the same form as U such that $\operatorname{range} \mathcal{S}U_1 = \operatorname{range} \mathcal{S}U_2$. Then $U_2^H J U_1 = 0$, and from this it follows first that the associated integer parameters $t_1, \dots, t_p$ must be the same, and thus all the blocks $V_{i,j}$, $W_{i,j}$ must be the same.

To prove the second part, let $\mathcal{X}$ be a symplectic matrix which triangularizes $\mathcal{H}$. Since the first n columns of $\mathcal{X}$ form a Lagrangian invariant subspace, there exists a matrix U of the form (3.6) such that $\operatorname{range} \mathcal{X} \left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right] = \operatorname{range} \mathcal{S}U$. Then the matrix

$\mathcal{U} = [U, \tilde{U}]$ with $\tilde{U}$ as in (3.10) is symplectic. Since both $\mathcal{X}$ and $\mathcal{SU}$ are symplectic and their first n columns span the same subspace, there exists a symplectic block triangular matrix $\mathcal{Y}$ such that $\mathcal{X} = \mathcal{SU}\mathcal{Y}$. □

These results show that the parameters that characterize a Lagrangian invariant subspace are integers t_j with $0 \le t_j \le m_j$, and the matrices $V_{i,j}$, $W_{i,j}$ satisfying the Riccati equations (3.7) or, equivalently, (3.8) and (3.9). Note that the equation for W is a singular Lyapunov equation. The equation for V is quadratic. But if we consider it blockwise, it is equivalent to a sequence of singular Sylvester equations,

$$N_{s_i} V_{i,j} - V_{i,j} N_{t_j} - \sum_{k=i+1}^{j-1} V_{i,k} E_k V_{k,j} = 0 \tag{3.11}$$

for $i = p-1, \ldots, 1$, $j = i+1, \ldots, p$. For results on nonsymmetric Riccati equations, see [10].

In general not much more can be said about this parametrization. In the special case of a Hamiltonian matrix $\mathcal{H}$ that has only two Jordan blocks, we have the following result.

COROLLARY 3.5. *Consider a Hamiltonian matrix $\mathcal{H}$ that has exactly two Jordan blocks $N_{r_1}(i\alpha), N_{r_2}(i\alpha)$ with $0 < r_2 \le r_1$ and the corresponding structure inertia indices $\beta_1 = -\beta_2$. Then there exists a symplectic matrix $\mathcal{S}$ such that*

$$\mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \begin{bmatrix} N_m(i\alpha) & D \\ 0 & -N_m(i\alpha)^H \end{bmatrix},$$

where $m = (r_1 + r_2)/2$, $d = (r_1 - r_2)/2$, and $D = \tau e_d e_m^H + \bar{\tau} e_m e_d^H$, and $\tau = \pm i/2$ if r_1 is odd and $\tau = \pm 1/2$ if r_2 is even. All Lagrangian invariant subspaces of $\mathcal{H}$ can be parametrized by

$$\operatorname{range} \mathcal{S} \begin{bmatrix} I_t & 0 \\ 0 & W \\ 0 & 0 \\ 0 & -I_s \end{bmatrix},$$

and all symplectic matrices that transform $\mathcal{H}$ to Hamiltonian block triangular form can be parametrized as

$$\mathcal{S} \begin{bmatrix} I_p & 0 & 0 & 0 \\ 0 & W & 0 & I_q \\ 0 & 0 & I_p & 0 \\ 0 & -I_q & 0 & 0 \end{bmatrix} \mathcal{Y},$$

where $\mathcal{Y}$ is symplectic block triangular, $d \le t \le m$, $t + s = m$, $W = W^H$ satisfying

$$N_s W + W N_s^H = 0, \tag{3.12}$$

which has infinitely many solutions for every $s > 0$.

Proof. Note that $r_1 + r_2$ is the size of the Hamiltonian matrix $\mathcal{H}$, which must be even. So r_1 and r_2 must be both even or odd. The canonical form and the form of the parametrization follow directly from Theorem 3.4 by setting there $p = 1$. So we need to prove only that $d \le t \le m$ and that (3.12) holds. For $p = 1$, (3.8) reduces to

$$N_s W + W N_s^H = K,$$

where K is the trailing $s \times s$ block of D. Then $K = 0$ if $t \geq d$ ($s \leq r_2$) and $K = \tau e_{d-t} e_s^H + \bar{\tau} e_s e_{d-t}^H$ if $t < d$ ($s > r_2$). If $t \geq d$, then the singular Lyapunov equation has infinitely many Hermitian solutions W; see [11, 14]. If $t < d$ and r_1, r_2 are both even, then it follows that $\tau \neq 0$ is real. By comparing the elements, it follows that the Lyapunov equation has no solution. The same conclusion follows for the case that r_1, r_2 are both odd. Consequently W exists if and only if $d \leq t \leq m$. ∎

In this simple case the parameters are completely given. But more importantly this result also gives a sufficient condition that a Hamiltonian matrix has infinitely many Lagrangian invariant subspaces.

COROLLARY 3.6. *If a Hamiltonian matrix $\mathcal{H}$ has exactly one eigenvalue $i\alpha$ and has at least two even-sized or two odd-sized Jordan blocks with opposite structure inertia indices, then $\mathcal{H}$ has infinitely many Lagrangian invariant subspaces.*

Proof. We may assume w.l.o.g. that the two (even or odd) Jordan blocks are arranged in trailing position of $\mathcal{R}$ in the canonical form (3.5). Choosing $t_j = r_j$ for all $j = 1, \ldots, p-1$ implies that all $V_{i,j}$ are void, $W = W_{p,p}$, and

$$U = \begin{bmatrix} I & 0 & 0 \\ 0 & I_{t_p} & 0 \\ 0 & 0 & W_{pp} \\ 0 & 0 & 0 \\ 0 & 0 & -I_{s_p} \end{bmatrix}.$$

By Corollary 3.5 there are infinitely many Lagrangian invariant subspaces (that are parametrized by $W_{p,p}$) for the small Hamiltonian matrix

$$H_p := \begin{bmatrix} N_{m_p}(i\alpha) & D_p \\ 0 & -N_{m_p}(i\alpha)^H \end{bmatrix},$$

and hence there are also infinitely many Lagrangian invariant subspaces for $\mathcal{H}$. ∎

This corollary shows that to obtain a unique Lagrangian invariant subspace, all structure inertia indices of $\mathcal{H}$ must have the same sign. Moreover, by Theorem 2.2 this also implies that $\mathcal{H}$ has only even-sized Jordan blocks. In the next section we will prove that this is also sufficient.

In order to complete the analysis we need to study Hamiltonian matrices $\mathcal{H}$ that have only two eigenvalues λ, $-\bar{\lambda}$ that are not purely imaginary. If $\mathcal{H} \in \mathbf{C}^{2n,2n}$, then the algebraic multiplicities of λ, $-\bar{\lambda}$ are both n, and hence $\Omega(\mathcal{H})$ consists of $n+1$ selections $\omega(m)$, $m = 0, \ldots, n$, where $\omega(m)$ contains m copies of λ and $n-m$ copies of $-\bar{\lambda}$.

It follows from Theorem 2.2 that in this case there exists a symplectic matrix $\mathcal{S}$ such that

$$\mathcal{R} := \mathcal{S}^{-1} \mathcal{H} \mathcal{S} = \begin{bmatrix} N(\lambda) & 0 \\ 0 & -N(\lambda)^H \end{bmatrix}, \tag{3.13}$$

where $N(\lambda) = \lambda I + N$, $N = \operatorname{diag}(N_{r_1}, \ldots, N_{r_p})$.

For every $\omega(m)$, $0 \leq m \leq n$, the parametrization of all possible Lagrangian invariant subspaces can be derived in a similar way as in the case of purely imaginary eigenvalues.

THEOREM 3.7. *Let $\mathcal{H} \in \mathbf{C}^{2n \times 2n}$ be a Hamiltonian matrix that has only eigenvalues $\lambda, -\bar{\lambda}$ which are not purely imaginary. Let $\mathcal{S}$ be a symplectic matrix that transforms $\mathcal{H}$ to the form* (3.13). *For every selection $\omega(m) \in \Omega(\mathcal{H})$ all the corresponding*

invariant subspaces can be parametrized by $\operatorname{range} \mathcal{S}U$*, where* U *has the form*

$$
\left[\begin{array}{cccccccccc}
I_{t_1} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & V_{12} & \dots & V_{1,p-1} & V_{1,p} & 0 & 0 & \dots & 0 & 0 \\
0 & I_{t_2} & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & V_{2,p-1} & V_{2,p} & 0 & 0 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & I_{t_{p-1}} & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & V_{p-1,p} & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & I_{t_p} & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & -I_{s_1} & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & V_{1,2}^H & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & -I_{s_2} & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & V_{1,p-1}^H & V_{2,p-1}^H & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & -I_{s_{p-1}} & 0 \\
0 & 0 & \dots & 0 & 0 & V_{1p}^H & V_{2,p}^H & \dots & V_{p-1,p}^H & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & -I_{s_p}
\end{array}\right], \tag{3.14}
$$

with $0 \le s_j, t_j \le r_j$*,* $s_j + t_j = r_j$*, and* $\sum_{j=1}^{p} t_j = m$*. If we set*

$$
\begin{aligned}
M_t &= \operatorname{diag}(N_{t_1}, \dots, N_{t_p}), \\
M_s &= \operatorname{diag}(N_{s_1}, \dots, N_{s_p}), \\
E &= \operatorname{diag}(e_{t_1} e_1^H, \dots, e_{t_p} e_1^H),
\end{aligned}
$$

then the matrix

$$
V := \left[\begin{array}{cccc}
0 & V_{12} & \dots & V_{1p} \\
 & \ddots & \ddots & \vdots \\
 & & \ddots & V_{p-1,p} \\
 & & & 0
\end{array}\right]
$$

must satisfy the Riccati equation

$$
0 = M_s V - V M_t - V E V. \tag{3.15}
$$

Every Lagrangian invariant subspace associated with $\omega(m)$ *is uniquely determined by a set of parameters* $\{t_1, \dots, t_p\}$ *with* $0 \le t_j \le r_j$ *and* $\sum_{j=1}^{p} t_j = m$ *and a set of matrices* $V_{i,j}$*,* $i = 1, \dots, p-1$*,* $j = i+1, \dots, p$*, satisfying* (3.15)*.*

Moreover, all symplectic matrices that transform $\mathcal{H}$ *to Hamiltonian block triangular form can be parametrized by* $\mathcal{SUY}$*, where* $\mathcal{Y}$ *is symplectic block triangular,*

$\mathcal{U} = [U, \tilde{U}]$ *with* U *as in* (3.10)*, and*

$$\tilde{U} = \begin{bmatrix}
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & I_{t_1} & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & I_{t_2} & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & I_{t_{p-1}} & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & I_{t_p} \\
I_{s_1} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & I_{s_2} & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & I_{s_{p-1}} & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & I_{s_p} & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0
\end{bmatrix}.$$

Proof. It is sufficient to consider the Lagrangian invariant subspaces of $\mathcal{R}$ in (3.13). Let the columns of X span a Lagrangian invariant subspace of $\mathcal{R}$ associated with $\omega(m)$. Then $\mathcal{R}X = XA$ and $\Lambda(A) = \omega(m)$. Since $\lambda \neq -\bar{\lambda}$, there exists a matrix Y such that $Y^{-1}AY = \left[\begin{smallmatrix} A_1 & 0 \\ 0 & A_2 \end{smallmatrix}\right]$, where A_1 is $m \times m$ and has only the eigenvalue λ and A_2 is $(n-m) \times (n-m)$ and has only the eigenvalue $-\bar{\lambda}$. If we partition $XY = \left[\begin{smallmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{smallmatrix}\right]$ conformally with $Y^{-1}AY$, then from the block diagonal form of $\mathcal{R}$ we obtain $X_{12} = 0$, $X_{21} = 0$ and $N(\lambda)X_{11} = X_{11}A_1$, $-N(\lambda)^H X_{22} = X_{22}A_2$, since X_{11}, X_{22} must have full column rank. We apply Lemma 3.2, and then the result follows as in the case of purely imaginary eigenvalues. ∎

The parametrization in this case is essentially the same as in the case of purely imaginary eigenvalues except that here W is void and $\sum_{j=1}^{p} t_j$ is fixed for a given $\omega(m)$. In both cases the blocks $V_{i,j}$ still satisfy a sequence of Sylvester equations (3.11).

Again we have a corollary.

COROLLARY 3.8. *Let* $\mathcal{H} \in \mathbf{C}^{2n \times 2n}$ *be a Hamiltonian matrix that has only the eigenvalues* $\lambda, -\bar{\lambda}$ *which are not purely imaginary. If* $\mathcal{H}$ *has exactly two Jordan blocks with respect to* λ*, then for every fixed* $\omega(m) \in \Omega(\mathcal{H})$ *the corresponding Lagrangian invariant subspaces can be parametrized as*

$$\mathcal{S} \begin{bmatrix}
I_{t_1} & 0 & 0 & 0 \\
0 & V & 0 & 0 \\
0 & I_{t_2} & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -I_{s_1} & 0 \\
0 & 0 & V^H & 0 \\
0 & 0 & 0 & -I_{s_2}
\end{bmatrix},$$

where $t_1 + t_2 = m$*,* $t_j + s_j = r_j$*, and* $0 \leq t_j, s_j \leq r_j$ *for* $j = 1, 2$*.*

Furthermore, $V = [0, T]$ *if* $s_1 < t_2$ *and* $V = \left[\begin{smallmatrix} T \\ 0 \end{smallmatrix}\right]$ *if* $s_1 \geq t_2$*, where* T *is an arbitrary square upper triangular Toeplitz matrix. So for every* $\omega(m)$ *with* $0 < m < n$ *there are infinitely many Lagrangian invariant subspaces.*

Proof. Applying Theorem 3.7 for $p = 2$ we obtain the parametrization and the restrictions for t_1, t_2. The expression for V follows from the fact that V satisfies the Sylvester equation $N_{s_1} V - V N_{t_2} = 0$. ∎

In this special case we have the following uniqueness result.

COROLLARY 3.9. *Let* $\mathcal{H} \in \mathbf{C}^{2n \times 2n}$ *be a Hamiltonian matrix that has only the eigenvalues* $\lambda, -\bar{\lambda}$ *which are not purely imaginary. Then we have the following:*

(i) *For* $\omega(0)$ *or* $\omega(n)$ *the corresponding Lagrangian subspace is unique.*

(ii) *If* $\mathcal{H}$ *has only a single Jordan block with respect to* λ*, then for every fixed* $\omega(m) \in \Omega(\mathcal{H})$ *with* $0 \leq m \leq n$ *the corresponding Lagrangian invariant subspace is unique. In this case there exists a symplectic matrix* $\hat{\mathcal{S}}$ *such that*

$$\hat{\mathcal{S}}^{-1} \mathcal{H} \hat{\mathcal{S}} = \begin{bmatrix} R & D \\ 0 & -R^H \end{bmatrix}, \tag{3.16}$$

with $R = \operatorname{diag}(N_m(\lambda), -N_{n-m}(\lambda)^H)$*,* $D = e_m e_{m+1}^H + e_{m+1} e_m^H$*.*

(iii) *If* $\mathcal{H}$ *has at least two Jordan blocks with respect to* λ*, then for every fixed* $\omega(m) \in \Omega(\mathcal{H})$ *with* $0 < m < n$ *there are infinitely many corresponding Lagrangian invariant subspaces.*

Proof. (i) For $\omega(0)$ all t_j must be zero, so $U = \left[\begin{smallmatrix} 0 \\ -I_n \end{smallmatrix}\right]$ is unique. Analogously, for $\omega(n)$ the unique Lagrangian invariant subspace is $U = \left[\begin{smallmatrix} I_n \\ 0 \end{smallmatrix}\right]$.

(ii) By assumption $p = 1$, so for a fixed $\omega(m)$, U is unique as

$$\begin{bmatrix} I_m \\ 0 \\ 0 \\ -I_{n-m} \end{bmatrix}.$$

Then (3.16) follows from (3.13) and the special form $\mathcal{U}$ for $p = 1$.

(iii) In this case we can choose the integers t_j such that $t_1 < r_1$ and $t_p > 0$. We set $V_{i,j} = 0$ except for $V_{1,p}$, which is chosen to satisfy $N_{s_1} V_{1,p} - V_{1,p} N_{t_p} = 0$. Since $s_1, t_p > 0$, there are infinitely many solutions $V_{1,p}$ and, hence, infinitely many U. ∎

In the next section we will use the parametrizations to characterize the existence and uniqueness of Lagrangian invariant subspaces.

4. Existence and uniqueness of Lagrangian invariant subspaces. In this section we summarize all results given in the previous sections and give a complete characterization of the existence and the uniqueness of Lagrangian invariant subspaces for a Hamiltonian matrix. This complete result includes previous results based on the structure inertia indices of [23, 25].

THEOREM 4.1 (existence). *Let* $\mathcal{H} \in \mathbf{C}^{2n \times 2n}$ *be a Hamiltonian matrix, let* $i\alpha_1, \ldots, i\alpha_\nu$ *be its pairwise distinct purely imaginary eigenvalues, and let* $\lambda_1, -\bar{\lambda}_1, \ldots, \lambda_\mu, -\bar{\lambda}_\mu$ *be its pairwise distinct nonimaginary eigenvalues. The following are equivalent:*

(i) $\mathcal{H}$ *has a Lagrangian invariant subspace for one* $\omega \in \Omega(\mathcal{H})$*.*

(ii) $\mathcal{H}$ *has a Lagrangian invariant subspace for all* $\omega \in \Omega(\mathcal{H})$*.*

(iii) *There exists a symplectic matrix* $\mathcal{S}$ *such that* $\mathcal{S}^{-1} \mathcal{H} \mathcal{S}$ *is Hamiltonian block triangular.*

(iv) *There exists a unitary symplectic matrix* $\mathcal{U}$ *such that* $\mathcal{U}^H \mathcal{H} \mathcal{U}$ *is Hamiltonian block triangular.*

(v) *For all $k = 1, \ldots, \nu$, if U_k span the invariant subspace associated with $i\alpha_k$, then $U_k^H J U_k$ is congruent to J_{m_k}.*

(vi) $\mathrm{Ind}_S^d(i\alpha_k)$ *is void for all $k = 1, \ldots, \nu$.*

Proof. This result in different notation is known; see [19, 23, 24, 25]. ∎

THEOREM 4.2 (uniqueness for $\tilde{\Omega}(\mathcal{H})$). *Let $\mathcal{H} \in \mathbf{C}^{2n\times 2n}$ be a Hamiltonian matrix. Let $i\alpha_1, \ldots, i\alpha_\nu$ be its pairwise distinct purely imaginary eigenvalues and let $\lambda_1, -\bar{\lambda}_1, \ldots, \lambda_\mu, -\bar{\lambda}_\mu$ be its pairwise distinct nonimaginary eigenvalues. Suppose that any of the equivalent conditions of Theorem 4.1 for the existence of Lagrangian invariant subspaces holds. Then the following are equivalent:*

(i) *For every $\omega \in \tilde{\Omega}(\mathcal{H})$ there exists a unique associated Lagrangian invariant subspace.*

(ii) *If $\omega \in \tilde{\Omega}(\mathcal{H})$ and if $\mathcal{S}_1$ and $\mathcal{S}_2$ are symplectic matrices such that $\mathcal{S}_1^{-1}\mathcal{H}\mathcal{S}_1 = \left[\begin{smallmatrix} R_1 & D_1 \\ 0 & -R_1^H \end{smallmatrix}\right]$, $\mathcal{S}_2^{-1}\mathcal{H}\mathcal{S}_2 = \left[\begin{smallmatrix} R_2 & D_2 \\ 0 & -R_2^H \end{smallmatrix}\right]$, and $\Lambda(R_1) = \Lambda(R_2) = \omega$, then $\mathcal{S}_1^{-1}\mathcal{S}_2$ is symplectic block triangular.*

(iii) *There exists an $\omega \in \tilde{\Omega}(\mathcal{H})$ such that $\mathcal{H}$ has a unique associated Lagrangian invariant subspace.*

(iv) *There exists an $\omega \in \tilde{\Omega}(\mathcal{H})$ such that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are symplectic matrices satisfying $\mathcal{S}_1^{-1}\mathcal{H}\mathcal{S}_1 = \left[\begin{smallmatrix} R_1 & D_1 \\ 0 & -R_1^H \end{smallmatrix}\right]$, $\mathcal{S}_2^{-1}\mathcal{H}\mathcal{S}_2 = \left[\begin{smallmatrix} R_2 & D_2 \\ 0 & -R_2^H \end{smallmatrix}\right]$, and $\Lambda(R_1) = \Lambda(R_2) = \omega$, then $\mathcal{S}_1^{-1}\mathcal{S}_2$ is symplectic block triangular.*

(v) *Let $\left[\begin{smallmatrix} A & B \\ 0 & -A^H \end{smallmatrix}\right]$ be an arbitrary Hamiltonian block triangular form of $\mathcal{H}$. If for a purely imaginary eigenvalue $i\alpha_k$ the columns of Φ_k form a basis of the left eigenvector subspace of A, i.e., $\Phi_k^H A = i\alpha_k \Phi_k^H$, then $\Phi_k^H B \Phi_k$ is positive definite or negative definite.*

(vi) *For every purely imaginary eigenvalue $i\alpha_k$ there are only even-sized Jordan blocks which, furthermore, have all structure inertia indices of the same sign.*

If the uniqueness conditions do not hold, then for every $\omega \in \tilde{\Omega}(\mathcal{H})$ there are infinitely many Lagrangian invariant subspaces. They can be parametrized by applying Theorem 3.4 for every $i\alpha_k$.

Proof. The proof of the equivalence of (i) and (vi) has been given (in different notation) in Theorem 1.3 of [25]. For completeness we give the whole proof in our terminology. By the argument in section 3 it suffices to consider a Hamiltonian matrix $\mathcal{H}$ that has either a single purely imaginary eigenvalue $i\alpha$ or an eigenvalue pair λ and $-\bar{\lambda}$. In the first case we again take $i\alpha = 0$.

Since by Corollary 3.9 for nonimaginary eigenvalues the corresponding invariant subspaces are unique, we need to consider only the case of a purely imaginary eigenvalue.

The proofs of (i) ⇔ (ii) and (iii) ⇔ (iv) are obvious. Corollary 3.6 implies that (ii) ⇒ (vi). If (vi) holds, then by Theorem 2.2 there exists a symplectic matrix $\mathcal{S}$ such that

$$\mathcal{R} := \mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \begin{bmatrix} R & D \\ 0 & -R^H \end{bmatrix}, \tag{4.1}$$

where $R = \mathrm{diag}(N_{l_1}, \ldots, N_{l_q})$ and $D = \beta\,\mathrm{diag}(e_{l_1}e_{l_1}^H, \ldots, e_{l_q}e_{l_q}^H)$. (Recall that $i\alpha = 0$.) We need to prove only that for every symplectic $\mathcal{Z}$ satisfying $\mathcal{Z}^{-1}\mathcal{R}\mathcal{Z} = \left[\begin{smallmatrix} \tilde{R} & \tilde{D} \\ 0 & -\tilde{R}^H \end{smallmatrix}\right]$, $\mathcal{Z}$ is block triangular. Partitioning $\mathcal{Z} = \left[\begin{smallmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{smallmatrix}\right]$, it follows that

$$RZ_{11} + DZ_{21} = Z_{11}\tilde{R} \tag{4.2}$$

and

$$-R^H Z_{21} = Z_{21}\tilde{R}. \tag{4.3}$$

Suppose that $Z_{21} \neq 0$; then by (4.3) it follows that range Z_{21} is an invariant subspace of $-R^H$. Hence, there exists a vector x such that $Z_{21}x \neq 0$ and

$$R^H Z_{21} x = 0, \tag{4.4}$$

i.e., $Z_{21}x$ is a left eigenvector of R. Multiplying $(Z_{21}x)^H$ and x on both sides of (4.2) and using (4.4), we get

$$(Z_{21}x)^H D(Z_{21}x) = -x^H Z_{21}^H Z_{11}\tilde{R}x.$$

Since $\mathcal{Z}$ is symplectic, we have $Z_{21}^H Z_{11} = Z_{11}^H Z_{21}$. Combining (4.3) and (4.4), we get

$$x^H Z_{21}^H Z_{11}\tilde{R}x = x^H Z_{11}^H Z_{21}\tilde{A}x = -x^H Z_{11}^H R^H Z_{21}x = 0$$

and, therefore,

$$(Z_{21}x)^H D(Z_{21}x) = 0.$$

On the other hand, since $Z_{21}x$ is a left eigenvector of R, by the structure of R there must exist a nonzero vector y such that $Z_{21}x = Ey$, where

$$E := [e_{p_1}, \ldots, e_{p_q}], \tag{4.5}$$

with $p_k = \sum_{j=1}^{k} l_j$ for $k = 1, \ldots, q$. But $E^H DE = \beta I_q$ and hence

$$0 = (Z_{21}x)^H D(Z_{21}x) = y^H E^H DEy = \beta y^H y \neq 0,$$

which is a contradiction.

(i) $\Rightarrow$ (iii) is obvious and (iii) $\Rightarrow$ (i) follows from (iii) $\Rightarrow$ (vi) by Corollary 3.6 and (vi) $\Leftrightarrow$ (i).

To prove (vi) $\Rightarrow$ (v) let $\hat{\mathcal{R}} = \left[\begin{smallmatrix} A & B \\ 0 & -A^H \end{smallmatrix}\right]$ be an arbitrary Hamiltonian triangular form of $\mathcal{H}$ and let $\mathcal{R}$ be as in (4.1). Since (vi) holds and (vi)$\Leftrightarrow$ (ii), there exists a symplectic block triangular matrix $\mathcal{S} = \left[\begin{smallmatrix} S_1 & S_2 \\ 0 & S_1^{-H} \end{smallmatrix}\right]$ (see [5]) such that $\hat{\mathcal{R}} = \mathcal{S}^{-1}\mathcal{R}\mathcal{S}$. Hence $S_1^{-1}RS_1 = A$ and $B = S_1^{-1}RS_2 + S_1^{-1}DS_1^{-H} + S_2^H R^H S_1^{-H}$. Since A is similar to R, a left eigenvector subspace of A can be chosen as $\Phi = S_1^H E$, where E is as in (4.5). Then a simple calculation yields $\Phi^H B\Phi = \beta I_q$.

For (v) $\Rightarrow$ (vi) suppose that $\hat{\mathcal{R}} = \left[\begin{smallmatrix} A & B \\ 0 & -A^H \end{smallmatrix}\right]$ satisfies (v). Using the same argument as for (vi) $\Rightarrow$ (ii) and replacing $\mathcal{R}$ by $\hat{\mathcal{R}}$ we obtain that (v) $\Rightarrow$ (ii). Since (ii) $\Leftrightarrow$ (vi), it follows that (v) also implies (vi). $\square$

THEOREM 4.3 (uniqueness for $\Omega(\mathcal{H})$). *Let $\mathcal{H} \in \mathbf{C}^{2n\times 2n}$ be a Hamiltonian matrix. Let $i\alpha_1, \ldots, i\alpha_\nu$ be its pairwise distinct purely imaginary eigenvalues and let $\lambda_1, -\bar{\lambda}_1, \ldots, \lambda_\mu, -\bar{\lambda}_\mu$ be its pairwise distinct nonimaginary eigenvalues. Suppose that any of the equivalent conditions of Theorem 4.1 for the existence of Lagrangian invariant subspaces holds. Then the following are equivalent:*

(i) *For every $\omega \in \Omega(\mathcal{H})$ there exists a unique associated Lagrangian invariant subspace.*

(ii) *Let* $\omega \in \Omega(\mathcal{H})$. *If* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are symplectic matrices such that* $\mathcal{S}_1^{-1}\mathcal{H}\mathcal{S}_1 = \left[\begin{smallmatrix} R_1 & D_1 \\ 0 & -R_1^H \end{smallmatrix}\right]$, $\mathcal{S}_2^{-1}\mathcal{H}\mathcal{S}_2 = \left[\begin{smallmatrix} R_2 & D_2 \\ 0 & -R_2^H \end{smallmatrix}\right]$, *and* $\Lambda(R_1) = \Lambda(R_2) = \omega$, *then* $\mathcal{S}_1^{-1}\mathcal{S}_2$ *is symplectic block triangular.*

(iii) *There exists an* $\omega \in \Omega(\mathcal{H})$, *but* $\omega \notin \tilde{\Omega}(\mathcal{H})$, *such that* $\mathcal{H}$ *has a unique associated Lagrangian invariant subspace.*

(iv) *There exists an* $\omega \in \Omega(\mathcal{H})$, *but* $\omega \notin \tilde{\Omega}(\mathcal{H})$, *such that if* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are symplectic matrices satisfying* $\mathcal{S}_1^{-1}\mathcal{H}\mathcal{S}_1 = \left[\begin{smallmatrix} R_1 & D_1 \\ 0 & -R_1^H \end{smallmatrix}\right]$, $\mathcal{S}_2^{-1}\mathcal{H}\mathcal{S}_2 = \left[\begin{smallmatrix} R_2 & D_2 \\ 0 & -R_2^H \end{smallmatrix}\right]$, *and* $\Lambda(R_1) = \Lambda(R_2) = \omega$, *then* $\mathcal{S}_1^{-1}\mathcal{S}_2$ *is symplectic block triangular.*

(v) *Let* $\left[\begin{smallmatrix} A & B \\ 0 & -A^H \end{smallmatrix}\right]$ *be an arbitrary Hamiltonian block triangular form of* $\mathcal{H}$. *Then either* A *has one of* $\lambda_k, -\bar{\lambda}_k$ *as its eigenvalue and has a unique corresponding left eigenvector, or* A *has both* $\lambda_k, -\bar{\lambda}_k$ *as eigenvalues and has unique corresponding left eigenvectors* x_k *and* y_k *such that* $x_k^H B y_k \neq 0$. *Furthermore, for every* $i\alpha_k$ *if the columns of* Φ_k *form a basis of the left eigenvector subspace of* A, *i.e.,* $\Phi_k^H A = i\alpha_k \Phi_k^H$, *then* $\Phi_k^H B \Phi_k$ *is positive definite or negative definite.*

(vi) *For every nonimaginary eigenvalue,* $\mathcal{H}$ *has only one corresponding Jordan block, and for every purely imaginary eigenvalue* $i\alpha_k$, $\mathcal{H}$ *has only even-sized Jordan blocks with all structure inertia indices of the same sign.*

If the uniqueness conditions do not hold, then for every $\omega \in \Omega(\mathcal{H})$ *there are infinitely many Lagrangian invariant subspaces. They can be parametrized by applying Theorem* 3.4 *for every* $i\alpha_k$ *and Theorem* 3.7 *for every pair* $\lambda_k, -\bar{\lambda}_k$.

Proof. The proof of the equivalence of (i) and (vi) has again been given (in different notation) in Theorem 1.3 of [25]. For completeness we again give the whole proof in our terminology.

By the argument in section 3 it suffices to consider that the Hamiltonian matrix $\mathcal{H}$ has only either a single purely imaginary eigenvalue $i\alpha$ or an eigenvalue pair λ and $-\bar{\lambda}$, and in the first case we will assume $i\alpha = 0$. For the purely imaginary eigenvalue the proof is as that of Theorem 4.2. Hence, consider $\mathcal{H}$ with an eigenvalue pair $\lambda, -\lambda$. The parts (i) $\Leftrightarrow$ (ii) and (iii) $\Leftrightarrow$ (iv) are obvious. (i) $\Leftrightarrow$ (vi) follows from Corollary 3.9. (i) $\Leftrightarrow$ (iii) follows, since (iii) $\Rightarrow$ (vi) and (vi) $\Leftrightarrow$ (i), and since $\omega \notin \tilde{\Omega}(\mathcal{H})$ implies that both λ and $-\bar{\lambda}$ have been chosen in ω. It remains to prove (v) $\Leftrightarrow$ (vi). We may assume that both λ, $-\bar{\lambda}$ are in $\Lambda(A)$, since otherwise $\omega \in \tilde{\Omega}(\mathcal{H})$.

For (vi) $\Rightarrow$ (v) let $\hat{\mathcal{R}} = \left[\begin{smallmatrix} A & B \\ 0 & -A^H \end{smallmatrix}\right]$ be an arbitrary Hamiltonian triangular form of $\mathcal{H}$. Since (vi) holds, by (3.16) in Corollary 3.9 the Hamiltonian canonical form is $\mathcal{R} = \left[\begin{smallmatrix} R & D \\ 0 & -R^H \end{smallmatrix}\right]$, where $R = \operatorname{diag}(N_t(\lambda), -N_s(\lambda)^H)$, $D = e_t e_{t+1}^H + e_{t+1} e_t^H$, and t is the multiplicity of λ in $\Lambda(A)$. By (ii) there exists a symplectic matrix $\mathcal{S} = \left[\begin{smallmatrix} S_1 & S_2 \\ 0 & S_1^{-H} \end{smallmatrix}\right]$ such that $\hat{\mathcal{R}} = \mathcal{S}^{-1}\mathcal{R}\mathcal{S}$. Hence $S_1^{-1} R S_1 = A$ and $B = S_1^{-1} R S_2 + S_1^{-1} D S_1^{-H} + S_2^H R^H S_1^{-H}$. If only one of $\lambda, -\bar{\lambda}$ is in $\Lambda(A)$, then, since A is similar to R, it is also in $\Lambda(R)$. Hence, either $t = 0$ or $s = 0$ and R (and thus also A) has only one corresponding left eigenvector. If both $\lambda, -\bar{\lambda}$ are in $\Lambda(A)$, then $s, t > 0$. In this case R has only left eigenvectors e_t and e_{t+1} with respect to λ and $-\bar{\lambda}$, respectively. Therefore, A also has only left eigenvectors $S_1^H e_t$ and $S_1^H e_{t+1}$ for λ and $-\bar{\lambda}$, respectively. Then it is easy to see that $e_t^H S_1 B S_1 e_{t+1} = e_t^H D e_{t+1} = 1$.

For (v) $\Rightarrow$ (vi), if A has only one of $\lambda, -\bar{\lambda}$ as its eigenvalue and has a unique left eigenvector, then A also has only one right eigenvector. Since $\Lambda(A) \cap \Lambda(-A^H) = \emptyset$, in this case $\hat{\mathcal{R}}$ has also a unique corresponding right eigenvector. Therefore, there is only one corresponding Jordan block. By the canonical form (3.13) for the other eigenvalue there is also only one Jordan block. If A has both $\lambda, -\bar{\lambda}$, then we first show that for left eigenvectors x, y of A such that $x^H B y \neq 0$, condition (ii) holds. Then we

show that x, y must be unique. As in the proof of Theorem 4.2 we need only prove that for every symplectic matrix $\mathcal{Z}$ satisfying $\mathcal{Z}^{-1}\hat{\mathcal{R}}\mathcal{Z} = \left[\begin{smallmatrix} \tilde{R} & \tilde{D} \\ 0 & -\tilde{R}^H \end{smallmatrix}\right]$, $\Lambda(\tilde{R}) = \Lambda(A)$ it follows that $\mathcal{Z}$ is block triangular. Partitioning $\mathcal{Z} = \left[\begin{smallmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{smallmatrix}\right]$, it follows that

$$AZ_{11} + BZ_{21} = Z_{11}\tilde{R} \tag{4.6}$$

and

$$-A^H Z_{21} = Z_{21}\tilde{R}.$$

Suppose that $Z_{21} \neq 0$; then range Z_{21} is an invariant subspace of $-A^H$. Hence there exists z_1, with either $\tilde{R}z_1 = -\bar{\lambda}z_1$ or $\tilde{R}z_1 = \lambda z_1$ such that $Z_{21}z_1 \neq 0$, which implies that $Z_{21}z_1$ is the left eigenvectors of A corresponding to λ or $-\bar{\lambda}$. W.l.o.g., assume that z_1 satisfies $\tilde{R}z_1 = -\bar{\lambda}z_1$. Let $z_2 \neq 0$ satisfy $z_2^H A = -\bar{\lambda}z_2^H$. Multiplying z_2^H and z_1 on both sides of (4.6), a simple calculation yields $z_1^H B(Z_{21}z_2) = 0$, which is a contradiction.

Suppose that x, y are not unique. Then let X form the left eigenvector subspace of A with respect to λ. Since $X^H By$ has more than one row, there always exists a vector z such that $z^H X^H By = 0$, which is a contradiction. So x and y must be unique. □

Remark 2. For real Hamiltonian matrices it is reasonable to consider real Lagrangian invariant subspaces. For this problem we have to give a natural additional restriction on the eigenvalue selections. Note that in this case if λ is a nonreal eigenvalue of $\mathcal{H}$, then $\bar{\lambda}, -\bar{\lambda}$, and $-\lambda$ are also eigenvalues of $\mathcal{H}$. To obtain real invariant subspaces it is necessary to keep the associated eigenvalues in conjugate pairs. So if we choose a nonreal λ we must choose $\bar{\lambda}$ with same multiplicity. But essentially we can use the same construction as for the complex case to solve this problem (see [19]), since if $\mathcal{H}$ is real, then for real eigenvalues the corresponding invariant subspaces can be chosen real. So for these eigenvalues we can still use Theorems 3.7 and 3.4 by choosing V and W real.

In this section we have given necessary conditions for the existence and uniqueness of Lagrangian invariant subspaces. In the following section we obtain as corollaries several results on the existence and uniqueness of Hermitian solutions of the algebraic Riccati equation.

5. Hermitian solutions of Riccati equations. In this section we apply the existence and uniqueness results for Lagrangian invariant subspaces to analyze the existence and uniqueness of Hermitian solutions of the algebraic Riccati equation

$$A^H X + XA - XMX + G = 0, \tag{5.1}$$

with $M = M^H$ and $G = G^H$. The related Hamiltonian matrix is $\mathcal{H} = \left[\begin{smallmatrix} A & M \\ G & -A^H \end{smallmatrix}\right]$. The following result is well known; see, e.g., [15].

PROPOSITION 5.1. *The algebraic Riccati equation* (5.1) *has a Hermitian solution if and only if there exists a* $2n \times n$ *matrix* $L = \left[\begin{smallmatrix} L_1 \\ L_2 \end{smallmatrix}\right]$, *with* $L_1, L_2 \in \mathbf{C}^{n\times n}$ *and* L_1 *invertible, such that the columns of* L *span a Lagrangian invariant subspace of the related Hamiltonian matrix* $\mathcal{H}$ *associated to* $\omega \in \Omega(\mathcal{H})$. *In this case* $X = -L_2 L_1^{-1}$ *is Hermitian and solves* (5.1) *and* $\Lambda(A - MX) = \omega$.

It follows that we can study the existence and uniqueness of solutions of algebraic Riccati equations via the analysis of Lagrangian invariant subspaces of the associated Hamiltonian matrices.

Unlike the Lagrangian invariant subspace problem, which only depends on the Jordan structure, Hermitian solutions of the Riccati equation depend further on the top block of the basis of the Lagrangian invariant subspace and the choice of the associated eigenvalues. In other words, for a given Hamiltonian block triangular form $\mathcal{R}$, all Hamiltonian matrices which are similar to $\mathcal{R}$ have Lagrangian invariant subspaces, while for Riccati equation solutions these Hamiltonian matrices may be partitioned into three groups which (i) have Hermitian solutions for all selections $\Omega(\mathcal{R})$, (ii) have Hermitian solutions for some $\omega \in \Omega(\mathcal{R})$, (iii) have no Hermitian solution for any $\omega \in \Omega(\mathcal{R})$.

Example 1. Consider three Riccati equations with matrices

$$\text{(a)}\ A = \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & -1-i \\ -1+i & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\text{(b)}\ A = \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & -1-i \\ -1+i & -2 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\text{(c)}\ A = \begin{bmatrix} i & 0 \\ 0 & -1 \end{bmatrix}, \quad M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} -1 & -1+i \\ -1-i & 0 \end{bmatrix}.$$

In all three cases the related Hamiltonian matrices have the same Hamiltonian Jordan canonical form

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

and $\Omega(\mathcal{H}) = \{\omega_1, \omega_2\}$ with $\omega_1 = \{i, 1\}$, $\omega_2 = \{i, -1\}$. Certainly for both ω_1, ω_2 all Hamiltonian matrices have a unique Lagrangian invariant subspace. But the Hermitian solutions of the Riccati equation are different. In case (a) for ω_1 the solution is $X = 0$ and for ω_2 there is no solution. In case (b) for ω_1 the solution is 0 and for ω_2 the solution is $X = \left[\begin{smallmatrix} 0 & 0 \\ 0 & -1 \end{smallmatrix}\right]$. In case (c) for both ω_1 and ω_2 there is no solution at all. It is also possible that the Riccati equation has no Hermitian solution, while the related Hamiltonian matrix has infinitely many Lagrangian invariant subspaces.

Example 2. For

$$A = M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

the Riccati equation (5.1) has no solution. But for the associated Hamiltonian matrix the bases of the Lagrangian invariant subspace can be parametrized as

$$\begin{bmatrix} -i\beta & 0 \\ -1 & 0 \\ 0 & 1 \\ \alpha & i\beta \end{bmatrix}, \quad \begin{bmatrix} 0 & \gamma \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$

where α, β, γ are real.

By using the parametrizations in section 3 we can give a necessary and sufficient condition for the existence of the Hermitian solutions of the Riccati equation (5.1). Note that for the solvability of the Riccati equation it is necessary that the Hamiltonian matrix $\mathcal{H}$ associated to (5.1) has a Hamiltonian block triangular form. So there

exists a symplectic matrix $\mathcal{S}$ such that

$$\mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \begin{bmatrix} R & D \\ 0 & -R^H \end{bmatrix} \tag{5.2}$$

with $R = \operatorname{diag}(R_1, \ldots, R_\mu; R_{\mu+1}, \ldots, R_{\mu+\nu})$, $D = \operatorname{diag}(0, \ldots, 0; D_{\mu+1}, \ldots, D_{\mu+\nu})$. A submatrix $H_k := \left[\begin{smallmatrix} R_k & 0 \\ 0 & -R_k^H \end{smallmatrix}\right]$ has the Jordan form (3.13) with respect to the eigenvalues $\lambda_k, -\bar{\lambda}_k$ for $k = 1, \ldots, \mu$, and a submatrix $H_k := \left[\begin{smallmatrix} R_k & D_k \\ 0 & -R_k^H \end{smallmatrix}\right]$ has the Jordan form (3.5) with respect to $i\alpha_{k-\mu}$ for $k = \mu+1, \ldots, \mu+\nu$.

THEOREM 5.2. *Let $\mathcal{H}$ be the Hamiltonian matrix associated with the algebraic Riccati equation* (5.1) *and assume that $\mathcal{H}$ has a Hamiltonian block triangular form. Let $\mathcal{S}$ be a symplectic matrix satisfying* (5.2) *and let $\mathcal{P}$ be a permutation matrix such that*

$$\begin{aligned} \mathcal{P}^{-1}\mathcal{S}^{-1}\mathcal{H}\mathcal{S}\mathcal{P} &= \operatorname{diag}(H_1, \ldots, H_\mu; H_{\mu+1}, \ldots, H_{\mu+\nu}), \\ \mathcal{P}^H J \mathcal{P} &= \operatorname{diag}(J_{n_1}, \ldots, J_{n_\mu}; J_{m_1}, \ldots, J_{m_\nu}), \end{aligned} \tag{5.3}$$

where $H_k = \left[\begin{smallmatrix} R_k & 0 \\ 0 & -R_k^H \end{smallmatrix}\right]$. Then for an eigenvalue selection $\omega \in \Omega(\mathcal{H})$, the Riccati equation (5.1) *has a Hermitian solution X with $\Lambda(A - MX) = \omega$ if and only if there exist matrices $U_1, \ldots, U_\mu$ and $Q_1, \ldots, Q_\nu$ with the following properties. The matrices U_k are $2n_k \times n_k$ and have the block form* (3.10) *with blocks satisfying* (3.15) *and the matrices Q_k are $2m_k \times m_k$ and have the block form* (3.6) *with blocks satisfying* (3.8) *and* (3.9) *such that*

$$L_1 := [I_n, 0]\mathcal{S}\mathcal{P} \operatorname{diag}(U_1, \ldots, U_\mu; Q_1, \ldots, Q_\nu) \tag{5.4}$$

is nonsingular.

Moreover, $X = -[0, I_n]\mathcal{S}\mathcal{P} \operatorname{diag}(U_1, \ldots, U_\mu; Q_1, \ldots, Q_\nu) L_1^{-1}$.

Proof. Since $\mathcal{H}$ has a Hamiltonian block triangular form, we have (5.2) and $\mathcal{P}$ can easily be determined to obtain (5.3). A given ω specifies the number elements λ_k, $-\bar{\lambda}_k$, and hence by Theorems 3.7 and 3.4 we obtain the parametrizations for the bases of the associated Lagrangian invariant subspaces of $\mathcal{H}$. Thus by Proposition 5.1 we have the conclusion. ∎

Remark 3. If in the Hamiltonian matrix $\mathcal{H} = \left[\begin{smallmatrix} A & M \\ G & -A^H \end{smallmatrix}\right]$ the matrix M is positive or negative semidefinite, then the invertibility of L_1 in (5.4) is ensured by a controllability assumption; see Theorem 3.1 and Remark 3.2 in [9] or [15] for details. If (5.1) has a Hermitian solution with respect to a selection ω, then the uniqueness follows directly from the uniqueness results for Lagrangian invariant subspaces.

THEOREM 5.3. *Let $X = X^H$ be a Hermitian solution of* (5.1) *with $\Lambda(A - MX) = \omega$. Then X associated to ω is unique if and only if the related Hamiltonian matrix $\mathcal{H}$ has a unique Lagrangian invariant subspace associated to ω. Moreover, in this case if $\omega \in \tilde{\Omega}(\mathcal{H})$, then for every selection in $\tilde{\Omega}(\mathcal{H})$ for which the associated Hermitian solutions exists, it is unique.*

If the uniqueness condition for the Lagrangian invariant subspaces of $\mathcal{H}$ does not hold and if (5.1) *has at least one Hermitian solution associated with a selection ω, then* (5.1) *has infinitely many Hermitian solutions associated to ω.*

Proof. The uniqueness conditions for the Hermitian solutions follows from the equivalence of (i) and (iii) in Theorems 4.2 and 4.3.

If (5.1) has a solution X associated to an ω, following Theorem 5.2, there must be two sets of matrices $U_1, \ldots, U_\mu$ and $Q_1, \ldots, Q_\nu$ such that for

$$L = \mathcal{SP}\,\mathrm{diag}(U_1, \ldots, U_\mu; Q_1, \ldots, Q_\nu) =: \begin{bmatrix} L_1 \\ L_2 \end{bmatrix},$$

L_1 is nonsingular and $X = -L_2 L_1^{-1}$. If the uniqueness condition for $\mathcal{H}$ does not hold, then for at least one pair λ_k, $-\bar{\lambda}_k$ or one purely imaginary eigenvalue $i\alpha_k$ the uniqueness condition does not hold. In the case of a pair $\lambda, -\bar{\lambda}$, by Theorem 3.7 the parameters $s_1, \ldots, s_p$ cannot be all zero. So the matrix V cannot be void and satisfies (3.15) or equivalently (3.11). For every $V_{i,j}$ the associated equation is a singular Sylvester equation. So at least for the last $V_{i,j}$, say $V_{1,p}$, there are infinitely many solutions. This means that we can choose infinitely many bases which are near to a certain U_k. For the case of an eigenvalue $i\alpha$ from Theorem 3.4 again $s_1, \ldots, s_p$ cannot be all zero. So W cannot be void. Since W must satisfy the singular Lyapunov equation (3.8), there are infinitely many solutions. So we can also choose infinitely many bases which are near to a certain Q_k. Consequently if the uniqueness condition of $\mathcal{H}$ does not hold, then there are infinitely many bases $\tilde{L}$ of the Lagrangian invariant subspaces associated to ω such that $||\tilde{L} - L|| < ||L_1^{-1}||$, which implies that there are infinitely many Hermitian solutions corresponding to such $\tilde{L}$. □

If a Hermitian solution X is known, then we can use it to verify the uniqueness.

COROLLARY 5.4. *Let X be a Hermitian solution of* (5.1) *with $\Lambda(A - MX) = \omega$. Let the columns of Φ_k, $k = 1, \ldots, \nu$, span the left eigenspaces of $A - MX$ corresponding to $i\alpha_k$. If $\Phi_k^H M \Phi_k$ is either positive definite or negative definite for all $k = 1, \ldots, \nu$, then $\omega \in \tilde{\Omega}(\mathcal{H})$ implies that X is unique. If $\omega \notin \tilde{\Omega}(\mathcal{H})$, then X is unique if we have the additional condition that for every eigenvalue pair λ_k and $-\bar{\lambda}_k$ the matrix $A - MX$ either has one of them as its eigenvalue and has a unique corresponding left eigenvector, or has both of them as eigenvalues and the corresponding left eigenvectors x_k, y_k satisfy $x_k^H M y_k \neq 0$ for $k = 1, \ldots, \mu$.*

Proof. The proof follows directly from the fact that

$$\mathcal{S}^{-1}\mathcal{H}\mathcal{S} = \begin{bmatrix} A - MX & M \\ 0 & -(A - MX)^H \end{bmatrix} =: \mathcal{R},$$

where $\mathcal{S} = \left[\begin{smallmatrix} I & 0 \\ -X & I \end{smallmatrix}\right]$ is symplectic, and from (v) in Theorems 4.2 and 4.3. □

6. Conclusion. Based on Hamiltonian block triangular forms for Hamiltonian matrices under symplectic similarity transformations we have given necessary and sufficient conditions for the existence and uniqueness of Lagrangian invariant subspaces. If the subspace is not unique, then we have given a complete parametrization of all possible Lagrangian invariant subspaces. We have then applied these results to derive existence and uniqueness results for Hermitian solutions of algebraic Riccati equations as corollaries.

REFERENCES

[1] G. AMMAR, P. BENNER, AND V. MEHRMANN, *A multishift algorithm for the numerical solution of algebraic Riccati equations*, Electron. Trans. Numer. Anal., 1 (1993), pp. 33–48.

[2] P. BENNER, V. MEHRMANN, AND H. XU, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, J. Comput. Appl. Math., 86 (1997), pp. 17–43.

[3] P. BENNER, V. MEHRMANN, AND H. XU, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., 78 (1998), pp. 329–358.

[4] P. Benner, V. Mehrmann, and H. Xu, *A note on the numerical solution of complex Hamiltonian and skew-Hamiltonian eigenvalue problems*, Electron. Trans. Numer. Anal., 8 (1999), pp. 115–126.
[5] A. Bunse-Gerstner, *Matrix factorization for symplectic QR-like methods*, Linear Algebra Appl., 83 (1986), pp. 49–77.
[6] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *Numerical methods for algebraic Riccati equations*, in Proceedings of the Workshop on the Riccati Equation in Control, Systems, and Signals, S. Bittanti, ed., Como, Italy, 1989, pp. 107–116.
[7] R. Byers, *Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation*, Ph.D. thesis, Cornell University, Ithaca, NY, 1983.
[8] R. Byers, *A Hamiltonian QR-algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[9] G. Freiling, *On the existence of Hermitian solutions of general algebraic Riccati equations*, in Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems, Perpignan, France, 2000, A. El Jai and M. Fliess, eds., to appear.
[10] G. Freiling and G. Jank, *Non-symmetric matrix Riccati equations*, Z. Anal. Anwendungen, 14 (1995), pp. 259–284.
[11] F. Gantmacher, *Theory of Matrices*, Vol. 1, Chelsea, New York, 1959.
[12] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[13] M. Green and D. Limebeer, *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[14] R. Horn and C. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[15] P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*, Oxford University Press, Oxford, 1995.
[16] A. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 913–921.
[17] 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, 1991, pp. 163–196.
[18] W.-W. Lin and T.-C. Ho, *On Schur Type Decompositions for Hamiltonian and Symplectic Pencils*, Tech. report, Institute of Applied Mathematics, National Tsing Hua University, Taiwan, 1990.
[19] W.-W. Lin, V. Mehrmann, and H. Xu, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 302/303 (1999), pp. 469–533.
[20] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, Lecture Notes in Control and Information Sciences 163, Springer-Verlag, Heidelberg, 1991.
[21] V. Mehrmann and H. Xu, *Numerical methods in control*, J. Comput. Appl. Math., 123 (2000), pp. 371–394.
[22] C. Paige and C. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 14 (1981), pp. 11–32.
[23] A. Ran and L. Rodman, *Stability of invariant maximal semidefinite subspaces*, Linear Algebra Appl., 62 (1984), pp. 51–86.
[24] A. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces* I, in Topics in Operator Theory, Oper. Theory Adv. Appl. 32, I. Gohberg, ed., Birkhäuser, Basel, 1988, pp. 181–218.
[25] A. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces* II, in The Gohberg Anniversary Collection, Oper. Theory Adv. Appl. 40, H. Dym, S. Goldberg, M.A. Kaashoek, and P. Lancaster, eds., Birkhäuser, Basel, 1989, pp. 391–425.
[26] M. A. Shayman, *Homogeneous indices, feedback invariants and control structure theorem for generalized linear systems*, SIAM J. Control Optim., 26 (1988), pp. 387–400.
[27] V. Sima, *Algorithms for Linear-Quadratic Optimization*, Monogr. Textbooks Pure Appl. Math. 200, Marcel Dekker, New York, 1996.
[28] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1995.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1070–1091
© 2002 Society for Industrial and Applied Mathematics

EXCLUSION AND INCLUSION REGIONS FOR THE EIGENVALUES OF A NORMAL MATRIX*

MARKO HUHTANEN† AND RASMUS MUNK LARSEN‡

Abstract. Assume $N \in \mathbb{C}^{n\times n}$ is a square matrix with the characteristic polynomial $p(z) = f(x,y) + ig(x,y)$. Viewing the spectrum $\sigma(N)$ of N as an algebraic subvariety of $\mathbb{R}^2$, by Bézout's theorem, the degrees of f and g seem to be unnecessarily high for locating $\sigma(N)$. Starting from this observation, we employ real analytic techniques to find the spectrum of a normal matrix N. At most 1-dimensional information is obtained with polyanalytic polynomials of degree not exceeding $\sqrt{2n}$. This is achieved by performing only matrix-vector products with an algorithm relying on a recurrence with a very slowly growing length. For large problems three practical alternatives are proposed via computing Ritz values, eigenvalue exclusion, and eigenvalue inclusion regions.

Key words. Bézout's theorem, algebraic subvariety of $\mathbb{R}^2$, eigenvalue localization, normal matrix, polyanalytic polynomial, slowly growing length of the recurrence, Ritz value

AMS subject classifications. 65F15, 42C05

PII. S0895479801384561

1. Introduction.

Assume $N \in \mathbb{C}^{n\times n}$ is a square matrix with the characteristic polynomial p. Rewriting p as $p(z) = f(x,y) + ig(x,y)$ with bivariate polynomials f and g, the spectrum of N equals the algebraic subvariety of $\mathbb{R}^2$ defined by

$$\begin{cases} f(x,y) = 0, \\ g(x,y) = 0. \end{cases} \tag{1.1}$$

It is clear that $\deg(f) = \deg(g) = \deg(p) = n$. A naive application of Bézout's theorem then tells us that this system could have as many as n^2 isolated solutions. So using complex analytic techniques on (1.1) to find the spectrum of N would appear fruitless. In this paper we demonstrate that if N is normal, then real analytic techniques yield us bivariate polynomials such that generically

$$\deg(f)\deg(g) \le 4n \tag{1.2}$$

holds pairwise. As there are n eigenvalues, counting multiplicities, (1.2) is of correct order in light of Bézout's theorem. We show that the algebraic subvariety of $\mathbb{R}^2$ defined by this altered set of bivariate polynomials remains unchanged so that the spectrum of N can actually be found with our techniques. For large problems, aside from computing Ritz values, we derive a new approach to finding exclusion and inclusion regions for the eigenvalues.

From now on we assume that N belongs to the set of normal matrices $\mathcal{N}$. Identifying multiplications by N and N^* with z and $\overline{z}$, respectively, we can associate

*Received by the editors February 5, 2001; accepted for publication (in revised form) by A. Edelman November 20, 2001; published electronically April 10, 2002.
http://www.siam.org/journals/simax/23-4/38456.html

†SCCM program, Computer Science Department, Stanford University, Stanford, CA 94305. Current address: Department of Mathematics, Room 2-335, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 01239 (Marko.Huhtanen@hut.fi). The research of this author was supported by the Academy of Finland and the Alfred Kordelin Foundation.

‡SCCM program, Computer Science Department, Stanford University, Stanford, CA 94305 and SOI-MDI, W.W. Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA 94305 (rmunk@solen.stanford.edu).

polyanalytic polynomials [3] with the elements of $\mathcal{N}$ in a natural way. Denoting by $\mathcal{P}_k$ the set of polynomials of degree at most k, polyanalytic polynomials are of the form

$$p(z) = \sum_{j=0}^{k} h_j(z)\overline{z}^j, \tag{1.3}$$

with $h_j \in \mathcal{P}_{k-j}$ and $k \in \mathbb{N}_0$. We use the notation $\mathcal{PP}_k$ for polyanalytic polynomials of degree at most k and set $\mathcal{PP} = \bigcup_{k\in\mathbb{N}_0} \mathcal{PP}_k$. Since N commutes with its adjoint, $p(N)$ is well defined for $p \in \mathcal{PP}$ by identifying z and $\overline{z}$ with N and N^*, respectively.

Polyanalytic polynomials of the form $z^j\overline{z}^l$ are called polyanalytic monomials and an order $>$ among them is set as follows. Let $z^{j_1}\overline{z}^{l_1}$ and $z^{j_2}\overline{z}^{l_2}$ be two polyanalytic monomials. If $j_1 + l_1 > j_2 + l_2$, then $z^{j_1}\overline{z}^{l_1} > z^{j_2}\overline{z}^{l_2}$. If $j_1 + l_1 = j_2 + l_2$ and $j_1 > j_2$, then $z^{j_1}\overline{z}^{l_1} > z^{j_2}\overline{z}^{l_2}$. With an order among the polyanalytic monomials we define the minimal polyanalytic polynomial $p_{j,l} \in \mathcal{PP}$ of $N \in \mathcal{N}$ to be the monic polyanalytic polynomial of least degree $j+l$ annihilating N. The minimal polyanalytic polynomial has a number of desired properties, i.e., it is unique, its degree does not exceed $\sqrt{2\deg(N)}$, and it can be computed in a stable manner. Moreover, its zero set is at most 1-dimensional containing the spectrum of N. Its intersection with Gershgorin's disks yields very accurate information about the location of the eigenvalues. However, as opposed to the minimal polynomial, the zero set of $p_{j,l}$ can be strictly larger than the spectrum so that for equality further annihilating polyanalytic polynomials may need to be computed.

Finding further annihilating polyanalytic polynomials for N is not obvious. Since there are infinitely many algebraic curves in $\mathbb{R}^2$ containing the eigenvalues of N, a natural criterion for choosing a preferred polyanalytic polynomial is computability. The minimal polyanalytic polynomial can be found by introducing an Arnoldi [2] type of minimization problem

$$\left|\left|N^j N^{l*}\hat{q}_0 - \hat{p}_{j,l}(N)\hat{q}_0\right|\right| = \min_{\hat{p}<z^j\overline{z}^l} \left|\left|N^j N^{l*}\hat{q}_0 - \hat{p}(N)\hat{q}_0\right|\right| \tag{1.4}$$

for a vector $\hat{q}_0 \in \mathbb{C}^n$ [18]. The corresponding monic polyanalytic polynomial of interest equals $P_{j,l}(z) = z^j\overline{z}^l - \hat{p}_{j,l}(z)$, yielding a sequence $P_{j,l} \in \mathcal{PP}_k$ with $j + l = k$ for $k \in \mathbb{N}$. The process for computing these polyanalytic polynomials is iterative, relying on a construction of an orthonormal basis of $\mathbb{C}^n$. This is achieved by multiplying an already computed vector with either N or N^*. Then a new vector is obtained by orthogonalizing this against the vectors computed so far. The length of the recurrence for this purpose neither is fixed 3 nor does it grow linearly with the iteration number, as is the case with the Hermitian Lanczos and Arnoldi methods, respectively. Denoting by d the number of the orthogonal polyanalytic polynomials computed so far, the length of the recurrence is at most $\sqrt{8d}$.

There appears zero vectors among the vectors generated and to each of them corresponds an element of $\mathcal{PP}$ annihilating N. The first zero vector yields the minimal polyanalytic polynomial of N. The remaining ones can be divided into two classes: irreducible and reducible. The irreducible polyanalytic polynomials yield independent information, and we show that the intersection of their zero sets yields the spectrum of N exactly. We find this quite remarkable as the characteristic polynomial is not used to this end. Generically this involves polyanalytic polynomials of degree at most $\sqrt{2\deg(N)}$, i.e., pairwise the degree condition (1.2) is satisfied.

For large problems the computation of all the annihilating polyanalytic polynomials of N as just described is not realistic. So we propose three practical alternative tools for isolating the eigenvalues: Ritz values, eigenvalue exclusion regions, and eigenvalue inclusion regions. The computation of the Ritz values is fairly straightforward by projecting N to the subspaces computed. Since this is realized with a slowly growing recurrence, our scheme is significantly more efficient than the Arnoldi method. Regarding the second alternative, we have a method for generating potentially very "sharp" exclusion regions for the eigenvalues of N by computing lemniscates

$$\{z \in \mathbb{C} : ||p(N)|| < |p(z)|\,,\ p \in \mathcal{PP}\} \tag{1.5}$$

due to the fact that (1.4) gives rise to polyanalytic polynomials $P_{j,l}$ approximately annihilating N. The complexity of this scheme is $O(n^2)$. In particular, using these techniques with the Bauer–Fike bound yields a method for generating exclusion regions for the eigenvalues of any matrix $A \in \mathbb{C}^{n\times n}$. Then the problem reduces, in essence, to finding normal approximants to A. For eigenvalue inclusion regions we invoke the ideas of Householder [14, 15] by employing polyanalytic polynomials to compute separating loci.

The paper is organized as follows. In section 2 we consider computable algebraic subvarieties of $\mathbb{R}^2$ for finding the spectrum of a normal matrix. In section 3 we introduce an Arnoldi type of iterative method, i.e., a method relying on matrix-vector multiplications for generating polyanalytic polynomials in practice. The algorithmic derivation in section 3 overlaps with [18]. In section 4 we consider computing Ritz values with the method proposed. In section 5 we consider generating eigenvalue exclusion regions, and in section 6 we show how to compute eigenvalue inclusion regions. A brief discussion of the implementation of Algorithm 1 from section 3 in finite precision and numerical examples are given in section 7.

2. Algebraic subvarieties of $\mathbb{R}^2$ for the spectrum of a normal matrix. Let $p(z) = f(x,y) + ig(x,y)$ be the characteristic polynomial of a square matrix $N \in \mathbb{C}^{n\times n}$. By considering the real and imaginary parts of p separately, the spectrum of N is an algebraic subvariety of $\mathbb{R}^2$ defined by

$$\begin{cases} f(x,y) = 0, \\ g(x,y) = 0 \end{cases} \tag{2.1}$$

with $\deg(f) = \deg(g) = \deg(p) = n$. Since p is complex analytic, this 2-by-2 system has at most n solutions in $\mathbb{R}^2$. However, by Bézout's theorem there are $\deg(f)\deg(g) = n^2$ solutions in the projective plane $\mathbf{P}^2(\mathbb{C})$. Thus, is it possible to come up with another set of bivariate polynomials, without altering the subvariety, such that $\deg(f)\deg(g) \le cn$ would hold for some constant $c \ll n$ instead? In this section we show that for a normal N there is an algorithm to this end with $c = 4$ generically.

2.1. The minimal polyanalytic polynomial of a normal matrix. Assume N is a normal matrix and consider executing iterative methods based on matrix-vector products with N and its adjoint N^*. Since N commutes with N^*, the set of polynomials is not the largest class of functions that can be used to analyze such algorithms. Identifying multiplications by N and N^* with z and $\overline{z}$, respectively, renders the usage of the so-called polyanalytic polynomials [3] more natural. Let $\mathcal{P}_j$ denote the set of polynomials of degree at most j and set $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$.

DEFINITION 2.1. *Polyanalytic polynomials are functions of the form*

$$p(z) = \sum_{j=0}^{k} h_j(z)\overline{z}^j, \tag{2.2}$$

with $h_j \in \mathcal{P}_{k-j}$ *and* $k \in \mathbb{N}_0$.

If, for some j, there holds $\deg(h_j) = k - j$ for p in (2.2), then the degree of p is k. The set of polyanalytic polynomials of degree at most k will be denoted by $\mathcal{PP}_k$ and $\mathcal{PP} = \bigcup_{k\in\mathbb{N}_0} \mathcal{PP}_k$. In particular, $\mathcal{P}_k \subset \mathcal{PP}_k$. Polyanalytic polynomials of the form $z^j\overline{z}^l$, with $j, l \in \mathbb{N}_0$, will be called polyanalytic monomials.

As opposed to complex analytic polynomials, equations involving polyanalytic polynomials can have no solution, a discrete set of solutions, or an infinite number of solutions. To give an example, consider the polyanalytic polynomial $q(z) = z\overline{z} + 1$. Obviously, the equation $q(z) = 0$ has no solutions. On the other hand, if $w(z) = z\overline{z}-1$, then the solution set of $w(z) = 0$ is the unit circle, i.e., a continuum.

On $\mathcal{PP}$ we employ the order set in section 1. Then the leading term $\mathrm{LT}(p)$ of a polyanalytic polynomial p is well defined.

Example 1. For $p(z) = i\overline{z}^3 + z\overline{z}^2 + (1-i)z^2 - 3\overline{z}$ the leading term is $\mathrm{LT}(p) = z\overline{z}^2$.

With an order on $\mathcal{PP}$ we define $p \in \mathcal{PP}$ to be monic if $\mathrm{LT}(p) = z^j\overline{z}^l$, that is, if the leading term of p is a polyanalytic monomial. For instance, p in Example 1 is a monic polyanalytic polynomial. Recall that the minimal polynomial of $A \in \mathbb{C}^{n\times n}$ is the monic polynomial of least degree annihilating A. The degree of the minimal polynomial of A is denoted by $\deg(A)$

DEFINITION 2.2. *A minimal polyanalytic polynomial of* $N \in \mathcal{N}$ *is a monic* $p \in \mathcal{PP}$ *of least possible degree annihilating* N.

Example 2. Assume $N \in \mathbb{C}^{n\times n}$ is unitary with at least 4 distinct eigenvalues. Then a minimal polyanalytic polynomial of N is $p(z) = z\overline{z} - 1$. On the other hand, if N is Hermitian, then $p(z) = z - \overline{z}$ is a minimal polyanalytic polynomial of N.

By the same argument as for the minimal polynomial, it is easy to see that a minimal polyanalytic polynomial is unique as well as unitarily invariant [18].

PROPOSITION 2.3. *A minimal polyanalytic polynomial of* $N \in \mathcal{N}$ *is unique, and unitarily similar* $N_1, N_2 \in \mathcal{N}$ *have the same minimal polyanalytic polynomial.*

An algorithm for finding the minimal polyanalytic polynomial of N will be presented later.

The subindices of the minimal polyanalytic polynomial $p_{j,l}$ of $N \in \mathcal{N}$ refer to $\mathrm{LT}(p_{j,l}) = z^j\overline{z}^l$. The following is obvious.

PROPOSITION 2.4. *Let* $p_{j,l}$ *be the minimal polyanalytic polynomial of* $N \in \mathcal{N}$. *Then* $q_{j,l}(z) := \overline{p_{j,l}(\overline{z})}$ *is the minimal polyanalytic polynomial of* N^*.

The degree of $p_{j,l}$ can be bounded as follows.

THEOREM 2.5 (see [18]). *Let* $p_{j,l}$ *be the minimal polyanalytic polynomial of* $N \in \mathcal{N}$. *Then* $\deg(p_{j,l}) \le \sqrt{2\deg(N)}$.

Proof. With $d = \deg(N)$ form a matrix Krylov subspace

$$\mathcal{K}_d(N; I) := \mathrm{span}\{I, N, \dots, N^{d-1}\} \tag{2.3}$$

which contains all the polynomials in N. Since N is normal, $N^* = q(N)$ for a polynomial q; see, e.g., [11, Condition 17]. Thus,

$$\mathcal{K}_d(N; I) = \mathrm{span}_{j+l=k\in\mathbf{N}_0}\{N^jN^{l*}\} = \mathrm{span}\{I, N^*, N, N^{2*}, NN^*, N^2, \dots\}. \tag{2.4}$$

Forming a basis of $\mathcal{K}_d(N;I)$ from the sequence (2.4) instead must yield a linearly dependent $N^j N^{l*}$ for $k = l + j$ with $\frac{(k+1)(k+2)}{2} \geq \deg(N)$. The first one corresponds to the minimal polyanalytic polynomial of N and thus $k \leq \sqrt{2\deg(N)}$. ∎

By $\sigma(A)$ we denote the spectrum of a square matrix A and by $||A||$ its spectral norm. The zero set of the minimal polynomial equals the spectrum, whereas with the minimal polyanalytic polynomial we have an inclusion.

THEOREM 2.6. *Assume $N \in \mathcal{N}$ and $p \in \mathcal{PP}$. Then $\sigma(N) \subset \{z \in \mathbb{C} : |p(z)| \leq ||p(N)||\}$.*

Proof. A simple way to see this is to recall that since N is normal, $N^* = q(N)$ for a polynomial q. Therefore, inserting this into $p(N) \equiv p(N, N^*)$ in place of N^* we have $||p(N, N^*)|| = \max_{\lambda \in \sigma(N)} |p(\lambda, q(\lambda))|$, which yields the claim. ∎

See also [24, section 2.10] for this concept when p is polynomial.

COROLLARY 2.7. *Assume $N \in \mathcal{N}$ and $p \in \mathcal{PP}$. Then the boundary of $\{z \in \mathbb{C} : |p(z)| \leq ||p(N)||\}$ contains an eigenvalue of N.*

Proof. Clearly $p(N)$ is normal. The claim follows then from the fact that the spectral radius of a normal matrix equals the spectral norm of the matrix. ∎

COROLLARY 2.8. *Let $p_{j,l}$ be the minimal polyanalytic polynomial of $N \in \mathcal{N}$. Then the eigenvalues of N are contained in the zero set of $p_{j,l}$.*

Consequently, consider those $N \in \mathcal{N}$ whose eigenvalues lie on a given algebraic curve. Then the corresponding polyanalytic polynomial annihilates N regardless of the number of distinct eigenvalues of N. Conversely, recall that $\deg(N)$ of a normal matrix N equals the cardinality of $\sigma(N)$.

Example 3. Assume the spectrum of a normal N is contained in the set defined by the limacon $r = 1 - \cos(\theta)$. Then the minimal polyanalytic polynomial of N is

$$p_{2,2}(z) = z^2\overline{z}^2 - z^2\overline{z} - z\overline{z}^2 + \frac{1}{4}z^2 - \frac{1}{2}z\overline{z} - \frac{1}{4}\overline{z}^2$$

as long as $\sigma(N)$ has at least 12 distinct eigenvalues. (If there were fewer, then the minimal polyanalytic polynomial would change. For instance, with only 11 distinct eigenvalues the leading term of the minimal polyanalytic polynomial would be $z\overline{z}^3$.)

Thus, finding the minimal polyanalytic polynomial $p_{j,l}$ of $N \in \mathcal{N}$ yields a spectral exclusion set for the eigenvalues after finding the zero set of $p_{j,l}$. As we have seen in Example 3, this can be a continuum so that the difference is genuine then. This does not mean that Corollary 2.8 cannot yield accurate information.

Example 4. Let $N = \mathrm{diag}(1, 0, i) \in \mathbb{C}^{3\times 3}$. The minimal polyanalytic polynomial of N is $p_{0,1}(z) = \overline{z}^2 - (\frac{1}{2} + \frac{i}{2})z + (\frac{-1}{2} + \frac{i}{2})\overline{z} = x^2 - y^2 - x + y - 2xyi = (x - y)(x + y - 1) - 2xyi$. The zero set of the real part is the union of the lines $y = x$ and $y = -x + 1$. The zero set of the imaginary part is the union of the real and imaginary axes. Intersecting these zero sets yields the spectrum exactly.

Let $\Gamma(N)$ denote the zero set of the minimal polyanalytic polynomial of a normal matrix N. First, $\Gamma(N)$ cannot have interior points in $\mathbb{C}$ so that its planar measure is zero. Second, although the length of $\Gamma(N)$ may be infinite, a relevant portion of it can be bounded as follows.

THEOREM 2.9. *Assume $N \in \mathcal{N}$ and a disk D of radius R contains $\sigma(N)$. If $p_{j,l} = p^{\Re}_{j,l} + ip^{\Im}_{j,l}$ is the minimal polyanalytic polynomial of N and d is the smallest strictly positive value of $\deg(p^{\Re}_{j,l})$ and $\deg(p^{\Im}_{j,l})$, then the length of $\Gamma(N) \cap D$ is at most $2\pi Rd$.*

Proof. For the proof, modify [4] to our setting. ∎

Note the significance of the square root of Theorem 2.5 giving $d \leq \sqrt{2n}$, which is a very modest growth (as a function of the dimension n) for the length of $\Gamma(N) \cap D$.

There are several inexpensive alternatives for finding disks D containing $\sigma(N)$, such as the field of values or Gershgorin's disks. For finding an inclusion region for the eigenvalues, the following is useful when combined with Theorem 2.9.

THEOREM 2.10 (see [26]). *For $N = (n_{ij}) \in \mathcal{N}$ each of the disks*

$$\left\{ \lambda \in \mathbf{C} : |n_{ii} - \lambda| \leq \left(\sum_{j \neq i} |n_{ij}|^2 \right)^{\frac{1}{2}} \right\}$$

contains an eigenvalue of N.

2.2. Further computable annihilating polyanalytic polynomials for a normal matrix. The spectrum of N is a zero-dimensional algebraic subvariety of $\mathbb{R}^2$ so that there exists an infinite number of algebraic curves containing it as a subset. Being given by the characteristic polynomial, only algebraic curves defined by $p \in \mathcal{PP}_k$, with $k \leq n$, are of interest for finding the spectrum in practice. According to Theorem 2.5, the minimal polyanalytic polynomial belongs to this category. However, since its zero set can be 1-dimensional, additional annihilating elements of $\mathcal{PP}$ may need to be generated for locating the eigenvalues exactly. To illustrate this consider the following example.

Example 5. Assume $N \in \mathbb{C}^{8\times 8}$ is unitary with the characteristic polynomial $p(z) = z^8 - 1$. Then the minimal polyanalytic polynomial of N is $p_{1,1}(z) = z\overline{z} - 1$. Besides these, $p_{4,0}(z) = z^4 - \overline{z}^4 = 8xy(x^2 - y^2)i$ annihilates N. Its zero set is the union of the real and imaginary axis and the lines $y = x$ and $y = -x$. The intersection of the zero sets of $p_{1,1}$ and $p_{4,0}$ equals the spectrum of N such that $\deg(p_{1,1})\deg(p_{4,0}) = 8$.

More generally, let $N \in \mathbb{C}^{n\times n}$ be unitary of even degree at least 4 so that $p_{1,1}(z) = z\overline{z} - 1$ is the minimal polyanalytic polynomial of N. Form a union of $\frac{1}{2}\deg(N)$ distinct lines passing through two different eigenvalues of N each. The corresponding polyanalytic polynomial q annihilating N is of degree $\frac{1}{2}\deg(N)$. The intersection of the zero sets of $p_{1,1}$ and q equals the spectrum of N such that $\deg(p_{1,1})\deg(q) = \deg(N) \leq n$. Clearly q is not unique (as opposed to the minimal polyanalytic polynomial, which is unique) for finding the spectrum of N with $p_{1,1}$.

In addition to the degree restriction, only those annihilating polyanalytic polynomials can be regarded as useful which are computable with an algorithm whose complexity does not exceed the complexity of finding the characteristic polynomial. The minimal polynomial of N belongs to this category as it can be found, for instance, by generating the matrix Krylov subspace

$$\mathcal{K}_d(N; I) = \operatorname{span}\{I, N, \ldots, N^{d-1}\}, \tag{2.5}$$

with $d = \deg(N)$. By using the standard inner product

$$(A, B) := \operatorname{tr}(AB^*) \tag{2.6}$$

on $\mathbb{C}^{n\times n}$, the Arnoldi method yields an orthonormal basis $\{Q_t\}_{t=0}^{d-1}$ of (2.5). Then $Q_t = p_t(N)$ for a polynomial p_t of degree t for $0 \leq t \leq d-1$. After computing the Fourier coefficients α_t, we have $NQ_{d-1} - \sum_{t=0}^{d-1} \alpha_t p_t(N) = 0$ so that rearranging the terms gives the minimal polynomial of N. Of course, to get a less complex algorithm, one should employ Krylov subspaces instead [15].

We mimic this approach to compute other annihilating polyanalytic polynomials for N. Namely, with the Arnoldi method only the minimal polynomial is obtained annihilating the spectrum. To get others, we generate an orthonormal basis of (2.5) by employing both N and its adjoint N^* after noticing that

$$\mathcal{K}_d(N;I) = \operatorname{span}_{j+l=k\in\mathbf{N}_0}\{N^jN^{l*}\} = \operatorname{span}\{I, N^*, N, N^{2*}, NN^*, N^2, \dots\}, \tag{2.7}$$

indicating the order of the multiplications. By performing the orthogonalizations, this yields a sequence of matrices $\{Q_t\}_{t\geq 0}$ according to the rule

$$\begin{array}{ccccccc} I & \overbrace{N^*\ N} & \overbrace{N^*\ N\ \ N} & \overbrace{N^*\ N\ \ N\ \ N} & \overbrace{N^*\ N\ \ N\ \ N\ \ N} & \cdots \\ I & Q_0\ Q_0 & Q_1\ Q_1\ Q_2 & Q_3\ Q_3\ Q_4\ Q_5 & Q_6\ Q_6\ Q_7\ Q_8\ Q_9 & \cdots \\ Q_0 & Q_1\ Q_2 & Q_3\ Q_4\ Q_5 & Q_6\ Q_7\ Q_8\ Q_9 & Q_{10}Q_{11}Q_{12}Q_{13}Q_{14} & \cdots, \end{array} \tag{2.8}$$

where the first row shows the multiplying matrix, i.e., either N or N^*. The second row indicates the numbering of the matrix which is multiplied by the matrix in the first row above. And the third row indicates, after orthogonalizing against all the previous matrices in the third row and scaling by its length, the numbering of the resulting new matrix. We call, for $k = 1, 2, \dots$, the orthogonalization sweeps "cycles" corresponding to the overbraces. Thus, the kth cycle consists of $k+1$ mappings

$$Q_{\frac{(k-1)k}{2}} \to Q_{\frac{k(k+1)}{2}} \text{ and } Q_{\frac{(k-1)k}{2}-1+s} \to Q_{\frac{k(k+1)}{2}+s} \tag{2.9}$$

for $1 \leq s \leq k$. The first one corresponds to the multiplication by N^* and the remaining k to the multiplications by N.

Analogously to the Arnoldi method, behind this process there are polyanalytic polynomials $p_{j,l}$ yielding matrices Q_t when evaluated at N. More precisely, due to the identification used, the orthogonalization process (2.8) gives rise to polyanalytic polynomials $p_{j,l}$ with the leading terms $z^j\bar{z}^l$ as illustrated in Figure 1. Thus, each cycle corresponds to climbing the points along the respective diagonal.

LEMMA 2.11. *For $N \in \mathcal{N}$ the orthogonalization process* (2.8) *yields* $\deg(N)$ *orthonormal matrices.*

Proof. Since N^* is a polynomial of N, there will be at most $\deg(N)$ orthonormal matrices. So it remains to show that there will be exactly $\deg(N)$ orthonormal matrices.

Assume Q_{t_T} is the last nonzero matrix obtained with the process (2.8). Collect all the computed nonzero Q_{t_l} for $l = 1, \dots, T$. Clearly Q_{t_1} is a multiple of the unit matrix and each Q_{t_l} is a polynomial of N since N^* is a polynomial of N. Now

$$N[Q_{t_1}\cdots Q_{t_T}] = \left[\sum_{l=1}^{T}\alpha_l^1 Q_{t_l}\cdots\sum_{l=1}^{T}\alpha_l^T Q_{t_l}\right] \tag{2.10}$$

holds for some constants $\alpha_l^k \in \mathbb{C}$ for $1 \leq l, k \leq T$. Take any vector $b \in \mathbb{C}^n$. Then (2.10) yields

$$N[Q_{t_1}b\cdots Q_{t_T}b] = \left[\sum_{l=1}^{T}\alpha_l^1 Q_{t_l}b\cdots\sum_{l=1}^{T}\alpha_l^J Q_{t_l}b\right], \tag{2.11}$$

that is, $\operatorname{span}\{Q_{t_l}b\}_{l=1}^{T}$ is an invariant subspace of N containing the vector b. Choose b such that it has components from every spectral subspace of N. Because of the

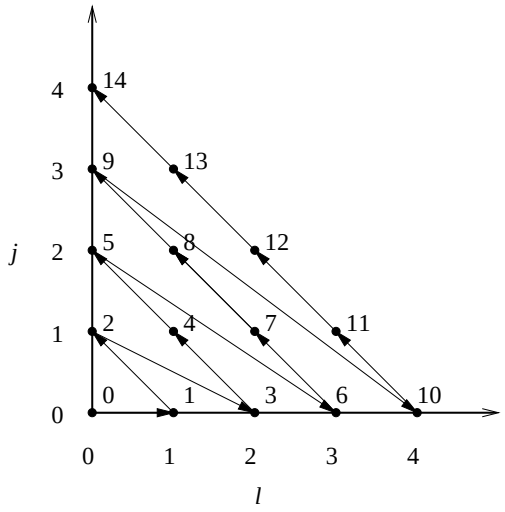


FIG. 1. *The order in which polyanalytic polynomials $p_{j,l}$ are generated by the process* (2.8). *The nodes are numbered as $t = 0, 1, 2, \ldots$ corresponding to the matrices Q_t.*

invariance, the dimension of $\operatorname{span}\{Q_{t_l} b\}_{l=1}^{T}$ must equal $\deg(N)$. Thus, $T = \deg(N)$, and this completes the proof. □

Note that there can be zero matrices among $\{Q_t\}_{t\geq 0}$. To give an example, if N is Hermitian, then at every cycle all but one Q_t generated equals zero. We are interested in those matrices $Q_t \neq 0$ which are mapped to the zero matrix in the process (2.9), since to each of them there corresponds an annihilating polyanalytic polynomial of N. However, a portion of these matrices can yield redundant information in the same sense that the characteristic polynomial yields redundant information compared with the minimal polynomial.

Example 6. Assume $N \in \mathbb{C}^{n\times n}$ is a nonderogatory Hermitian matrix. Clearly $p_{1,0}(z) = z - \overline{z}$ is its minimal polyanalytic polynomial and only one nonzero Q_t is produced at every cycle. At the 2nd cycle $0 \neq Q_1 \to Q_4 = 0$ to which corresponds $q(z) = z^2 - \overline{z}^2$. Since $q(z) = p_{1,0}(z)(z + \overline{z})$, the zero set of q yields no new information regarding the location of $\sigma(N)$. The next element of $\mathcal{PP}$ annihilating N which is not divisible by $p_{1,0}$ is obtained at the nth cycle. It is the characteristic polynomial of N.

A generalization of this also tells us why the redundant annihilating polyanalytic polynomials are easy to detect.

PROPOSITION 2.12. *Assume $p \in \mathcal{PP}$ with $\mathrm{LT}(p) = z^j \overline{z}^l$ annihilates $N \in \mathcal{N}$. Then, for $k \geq j + l$, we have $Q_{\frac{k(k+1)}{2}+s} = 0$ for $j \leq s \leq k - l$.*

Obviously it is only the following set of annihilating polyanalytic polynomials that is of interest for our purposes.

DEFINITION 2.13. *Assume a monic $q \in \mathcal{PP}$ annihilates $N \in \mathcal{N}$. If no monic $p \in \mathcal{PP}$ annihilating N divides q, then q is an irreducible annihilating polyanalytic polynomial for N.*

In particular, if the minimal and characteristic polynomials of N differ, then the former is irreducible whereas the latter is not. Obviously the minimal polyanalytic polynomial of N is irreducible.

Example 7. Let $N = \operatorname{diag}(1, -1, i, -i, e^{i\frac{\pi}{4}}, e^{-i\frac{\pi}{4}}, e^{i\frac{3\pi}{4}}, e^{-i\frac{3\pi}{4}}) \in \mathbb{C}^{8\times 8}$, so that

$p_{1,1}(z) = z\overline{z} - 1$. With this matrix we have $Q_7 = Q_8 = 0$ since we already had $Q_4 = 0$. However, $0 \neq Q_9 \to Q_{14} = 0$ and the corresponding polyanalytic polynomial is $q(z) = z^4 - \overline{z}^4 = ixy(x^2 - y^2)$, which is clearly not divisible by $p_{1,1}$. In particular, q is an irreducible annihilating polyanalytic polynomial for N.

Let $\{p_k\}_{k=1}^s$ denote the irreducible annihilating polyanalytic polynomials of $N \in \mathcal{N}$ generated with the process (2.8). It is obvious that s is finite, as the process (2.8) will stop after a finite number of steps. We denote by

$$\mathrm{V}(\{p_k\}) := \{z \in \mathbb{C} : p_k(z) = 0 \text{ for all } 1 \leq k \leq s\} \tag{2.12}$$

the algebraic subvariety of $\mathbb{R}^2$ defined by $\{p_k\}_{k=1}^s$. Now we can state the main result of this section.

THEOREM 2.14. *Let $\{p_k\}_{k=1}^s$ be the irreducible annihilating polyanalytic polynomials of $N \in \mathcal{N}$ generated with the process* (2.8). *Then* $\mathrm{V}(\{p_k\}) = \sigma(N)$.

Proof. By Lemma 2.11 there will be $d = \deg(N)$ orthonormal matrices Q_t after the process (2.8) has been completed. Let Q_{t_d} be the one that was computed last. Thus $Q_{t_d+s} = 0$ for $s \geq 1$, with the process (2.8). Let k_d be the cycle corresponding to Q_{t_d} and finish the computation of the (k_d+1)st cycle. Then all the arising Q_t's will be zero. Take the corresponding polyanalytic polynomials $\{\hat{p}_k\}$. They are obviously all annihilating for N.

Since $\sigma(N) \subset \mathrm{V}(\{p_k\})$, we need to demonstrate that the inclusion is actually an equality. Assume that this is not the case. For that purpose, let S be a finite subset of $\mathrm{V}(\{p_k\})$ that contains $\sigma(N)$ genuinely, i.e., $S \neq \sigma(N)$. Assign to this set a normal matrix N_S with $\sigma(N_S) = S$. Then run the orthogonalization process (2.8) for N_S. Since there are annihilating polyanalytic polynomials $\{\hat{p}_k\}$ with the properties just described, this orthogonalization process will stop before it has generated $\mathrm{card}(\sigma(N_S))$ orthogonal matrices, as all Q_t will be zero for $t > t_d$. This is in contradiction with Lemma 2.11. Therefore $\mathrm{V}(\{p_k\}) = \sigma(N)$ must hold. ∎

With a Hermitian N we actually do get the minimal polynomial at the last step with the process (2.8). Then the minimal polyanalytic polynomial is obtained at the 3rd step which is very early. This is also very exceptional since for a generic normal matrix N only zero matrices occur after its minimal polyanalytic polynomial has been generated. This can be seen by considering the determinant of an n-by-n matrix with the jth row consisting of the n first polyanalytic monomials evaluated at the jth point of $\sigma(N)$ (assume fixing some order). Generically the determinant is nonzero. In this case each p_k in (2.12) satisfies $\deg(p_k) \leq \sqrt{2\deg(N)}$. Thereby (1.2) also holds. Of course, then the minimal polynomial cannot be among $\{p_k\}_{k=1}^s$.

3. An Arnoldi type of iterative method for generating polyanalytic polynomials in practice. Being based on the inner product (2.6), the algorithm suggested in the previous section is too complex for generating polyanalytic polynomials in practice. Consequently, we will use a method relying only on matrix-vector products instead. The algorithm and its description follows [18] closely, where it was used in least squares approximation and bivariate interpolation.

To this end, consider an Arnoldi [2] type of minimization problem as follows. For $N \in \mathcal{N}$ and for $j + l = k$ set

$$\left|\left|N^j N^{*l} \hat{q}_0 - \hat{p}_{j,l}(N)\hat{q}_0\right|\right| = \min_{\hat{p} < z^j \overline{z}^l} \left|\left|N^j N^{*l} \hat{q}_0 - \hat{p}(N)\hat{q}_0\right|\right| \tag{3.1}$$

for a vector $\hat{q}_0 \in \mathbb{C}^n$. The resulting monic polyanalytic polynomial of interest equals $P_{j,l}(z) = z^j \overline{z}^l - \hat{p}_{j,l}(z)$. To compute these and the value of (3.1), the multiplications

by N and N^* need to be ordered appropriately to take all the vectors of the form $N^j N^{*l} \hat{q}_0$, for $j, l \in \mathbb{N}_0$, into account. Also, the corresponding orthogonalization process must be realized in a numerically stable way while simultaneously keeping track of the arising polyanalytic polynomials. For that purpose, take a vector $\hat{q}_0 \in \mathbb{C}^n$ of unit length and form a table

$$
\begin{array}{cccccccccccc}
I & \overbrace{N^*\ N} & \overbrace{N^*\ N\ N} & \overbrace{N^*\ N\ N\ N} & \overbrace{N^*\ N\ N\ N\ N} & \cdots \\
\hat{q}_0 & \hat{q}_0\ \hat{q}_0 & \hat{q}_1\ \hat{q}_1\ \hat{q}_2 & \hat{q}_3\ \hat{q}_3\ \hat{q}_4\ \hat{q}_5 & \hat{q}_6\ \hat{q}_6\ \hat{q}_7\ \hat{q}_8\ \hat{q}_9 & \cdots \\
\hat{q}_0 & \hat{q}_1\ \hat{q}_2 & \hat{q}_3\ \hat{q}_4\ \hat{q}_5 & \hat{q}_6\ \hat{q}_7\ \hat{q}_8\ \hat{q}_9 & \hat{q}_{10}\ \hat{q}_{11}\ \hat{q}_{12}\ \hat{q}_{13}\ \hat{q}_{14} & \cdots,
\end{array}
\tag{3.2}
$$

which is read from left to right. As in (2.8), the first row indicates the multiplying matrix. The second row indicates the numbering of the vector which is multiplied by the matrix in the first row above. And the third row indicates, after orthogonalizing against all the previous vectors in the third row and scaling by its length, the numbering of the resulting new vector. The purpose of the overbraces is to make the proposed rule more clear, as there are several ways to arrange the orthogonalizations. For $k = 1, 2, \ldots$, each cycle indicated by the overbraces gives rise to $k+1$ vectors. The rule is such that, to obtain the kth cycle, the first vector produced in the $(k-1)$st cycle is multiplied once by N^*. Then all the vectors of the $(k-1)$st cycle are multiplied once by N starting from the first vector. Consequently, after k cycles have been completed there are $\frac{(k+1)(k+2)}{2}$ vectors in all.

With the ordering (3.2) all the powers $N^j N^{*l} \hat{q}_0$ are catered to in a numerically stable way, and we have obtained an Arnoldi type of iterative method for normal matrices. We stress that we have not ruled out the possibility of having zero vectors among $\{\hat{q}_j\}_{j \geq 0}$. For instance, if N is Hermitian, then most of these vectors equal zero. In doing so we can with relative ease keep track of the arising polyanalytic polynomials. Obviously, if each cycle produces $k+1$ new linearly independent vectors, then the number of cycles is bounded according to $\frac{(k+1)(k+2)}{2} \leq n$.

A pseudocode for the process (3.2) for computing the orthogonalizations as well as all the relevant quantities is as follows.

ALGORITHM 1. "For computing vectors $\{\hat{q}_j\}_{j \geq 0}$."
Assume $N \in \mathcal{N}$ and $\hat{q}_0$ is of unit length.

for $k = 1 : K$
  Define $k_0 \equiv (k-2)(k-1)/2$, $k_1 \equiv (k-1)k/2$, $k_2 \equiv k(k+1)/2$
  $q_{k_2} = N^* \hat{q}_{k_1}$
  for $s = k_0 : k_2 - 1$
    $\alpha_s^{k_1} = (q_{k_2}, \hat{q}_s), \quad q_{k_2} = q_{k_2} - \alpha_s^{k_1} \hat{q}_s$
  end
  $\alpha_{k_2}^{k_1} = ||q_{k_2}||$
  if $\alpha_{k_2}^{k_1} > 0$ **then**
    $\hat{q}_{k_2} = (1/\alpha_{k_2}^{k_1})\, q_{k_2}$
  else
    $\hat{q}_{k_2} = 0$
  for $l = k_1 : k_2 - 1$
    $q_{k+l+1} = N\, \hat{q}_l$
    for $s = l - k + 1 : k + l$
      $\beta_s^l = (q_{k+l+1}, \hat{q}_s), \quad q_{k+l+1} = q_{k+l+1} - \beta_s^l \hat{q}_s$
    end
    $\beta_{k+l+1}^l = ||q_{k+l+1}||$

if $\beta^l_{k+l+1} > 0$ **then**
$\hat{q}_{k+l+1} = (1/\beta^l_{k+l+1})\, q_{k+l+1}$
else
$\hat{q}_{k+l+1} = 0$
end
end

The corresponding polyanalytic polynomial recurrence obtained by computing the coefficients with Algorithm 1 can then be written as

$$p_{0,0}(z) = 1, \tag{3.3}$$

$$\alpha^0_1 p_{0,1}(z) = \overline{z} p_{0,0}(z) - \alpha^0_0 p_{0,0}(z), \tag{3.4}$$

$$\beta^0_2 p_{1,0}(z) = z p_{0,0}(z) - \beta^0_1 p_{0,1}(z) - \beta^0_0 p_{0,0}(z), \tag{3.5}$$

$$\alpha^1_3 p_{0,2}(z) = \overline{z} p_{0,1}(z) - \alpha^1_2 p_{1,0}(z) - \alpha^1_1 p_{0,1}(z) - \alpha^1_0 p_{0,0}(z), \tag{3.6}$$

$$\beta^1_4 p_{1,1}(z) = z p_{0,1}(z) - \beta^1_3 p_{0,2}(z) - \beta^1_2 p_{1,0}(z) - \beta^1_1 p_{0,1}(z) - \beta^1_0 p_{0,0}(z), \tag{3.7}$$

and so on.

We state a number of basic properties of the orthogonalization process (3.2) and the corresponding Algorithm 1.

PROPOSITION 3.1 (see [18]). *Assume $N \in \mathcal{N}$ and $\hat{q}_0 \in \mathbb{C}^n$. Then*

$$\operatorname{span}\{\hat{q}_0, \ldots, \hat{q}_{\frac{(k+1)(k+2)}{2}-1}\} = \operatorname{span}_{p \in \mathcal{PP}_k}\{p(N)\hat{q}_0\}.$$

By a generic $\hat{q}_0 \in \mathbb{C}^n$ for $N \in \mathcal{N}$ we mean a vector that is supported by every spectral subspace of N.

COROLLARY 3.2. *Let $\hat{q}_0 \in \mathbb{C}^n$ be generic for $N \in \mathcal{N}$ and assume no element of $\mathcal{PP}_k \backslash \{0\}$ vanishes on $\sigma(N)$. Then $\hat{q}_0, \ldots, \hat{q}_{\frac{(k+1)(k+2)}{2}-1}$ are linearly independent.*

As usual, for $1 \leq k \leq n$, we denote by

$$\mathcal{K}_k(N; \hat{q}_0) = \operatorname{span}\{\hat{q}_0, N\hat{q}_0, \ldots, N^{k-1}\hat{q}_0\} \tag{3.8}$$

the Krylov subspaces of $N \in \mathbb{C}^{n\times n}$ at $\hat{q}_0 \in \mathbb{C}^n$.

PROPOSITION 3.3 (see [18]). *Assume $N \in \mathcal{N}$ and $\hat{q}_0 \in \mathbb{C}^n$. Then the number of nonzero vectors generated by the process* (3.2) *equals* $\dim(\mathcal{K}_n(N; \hat{q}_0))$.

The reason for arranging the orthogonalizations according to (3.2) is to be able to control the leading terms of the polyanalytic polynomials yielding vectors $\hat{q}_j$.

THEOREM 3.4 (see [18]). *Assume $N \in \mathcal{N}$ and $\hat{q}_0 \in \mathbb{C}^n$. Then the process* (3.2) *yields $\hat{q}_{\frac{k(k+1)}{2}+s} = p_{s,k-s}(N)\hat{q}_0$ with $\mathrm{LT}(p_{s,k-s}) = c_{\frac{k(k+1)}{2}+s} z^s \overline{z}^{k-s}$ and $c_{\frac{k(k+1)}{2}+s} \in \mathbb{C}$ for $k \in \mathbb{N}$ and $0 \leq s \leq k$.*

Note that we allow $c_{k(k+1)/2+s} = 0$.

COROLLARY 3.5. *Polyanalytic polynomials $P_{s,k-s}(z) = \frac{1}{c_{k(k+1)/2+s}} p_{s,k-s}(z)$ realize* (3.1). *The first $c_{k(k+1)/2+s} = 0$ corresponds to zero in* (3.1).

Proposition 2.12 has an analogue for Algorithm 1 revealing redundant annihilating polyanalytic polynomials that are divisible by the minimal polyanalytic polynomial.

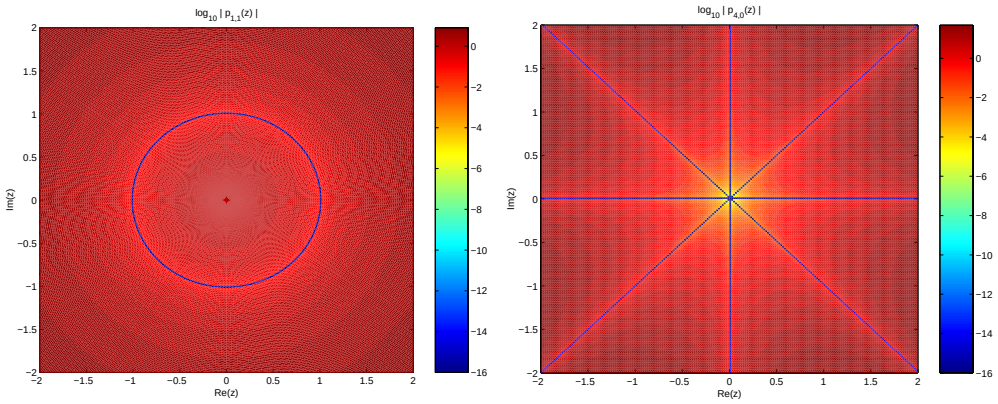


FIG. 2. *Illustration of the annihilating polyanalytic polynomials* $p_{1,1}(z)$ *and* $p_{4,0}(z)$ *from Example* 7 *and* 8 *computed with Algorithm* 1.

COROLLARY 3.6. *Let* $\hat{q}_0 \in \mathbb{C}^n$ *be generic for* $N \in \mathcal{N}$ *with the minimal polyanalytic polynomial* $p_{j,l}$. *Then, for* $k \geq j + l$, *we have* $\hat{q}_{\frac{k(k+1)}{2}+s} = 0$ *for* $j \leq s \leq k - l$.

In practice Algorithm 1 proceeds as follows.

Example 8. Let $N = \mathrm{diag}(1, -1, i, -i, e^{i\frac{\pi}{4}}, e^{-i\frac{\pi}{4}}, e^{i\frac{3\pi}{4}}, e^{-i\frac{3\pi}{4}}) \in \mathbb{C}^{8\times 8}$ and take $\hat{q}_0 = \frac{1}{2\sqrt{2}}[1, 1, \ldots, 1]^T$. The minimal polyanalytic polynomial of N is obtained by multiplying $\hat{q}_1 \neq 0$ by N. The next nonzero vector yielding a zero vector that is not forecast by Corollary 3.6 is $\hat{q}_9$ when multiplied with N, since then $N\hat{q}_9 - \hat{q}_{10} = 0$. The corresponding polyanalytic polynomial equals $p_{4,0}(z) = z^4 - \overline{z}^4 = 8xy(x^2 - y^2)i$. This annihilates N, and its zero set is the union of the lines $x = 0$, $y = 0$, $y = x$, and $y = -x$. This set is very different from the one yielded by the minimal polyanalytic polynomial of N. In fact, their intersection equals the spectrum of N; see Figure 2.

By Proposition 3.1 the vectors $\hat{q}_j$ have the property that, after having completed $k - 1$ cycles, gives us

$$\mathrm{span}\{\hat{q}_0, \ldots, \hat{q}_{\frac{k(k+1)}{2}-1}\} = \mathrm{span}_{p\in\mathcal{PP}_{k-1}}\{p(N)\hat{q}_0\}. \tag{3.9}$$

Then the kth cycle is obtained by multiplying, for $\frac{(k-1)k}{2} \leq s \leq \frac{k(k+1)}{2} - 1$, vectors $\hat{q}_s$ by either N or N^*. However, the inner products

$$(N^*\hat{q}_s, \hat{q}_j) = (\hat{q}_s, N\hat{q}_j,) \quad \text{and} \quad (N\hat{q}_s, \hat{q}_j) = (\hat{q}_s, N^*\hat{q}_j) \tag{3.10}$$

are zero for $\hat{q}_j \in \mathrm{span}_{p\in\mathcal{PP}_{k-3}}\{p(N)\hat{q}_0\}$. This property that the new iterates spanning the kth cycle are orthogonal against the previous span, modulo a small portion of vectors, has been noticed by Elsner and Ikramov in [10], where they used it for deriving condensed forms for normal matrices.

A more careful inspection of the ordering (3.2) of the orthogonalizations reveals that to finish the kth cycle, we need to save at most $2k + 1$ vectors. In other words, the length of recurrence is at most $2k$ during the kth cycle.

THEOREM 3.7. *Let* $\hat{q}_0 \in \mathbb{C}^n$ *be generic for* $N \in \mathcal{N}$ *and assume no element of* $\mathcal{PP}_k\backslash\{0\}$ *vanishes on* $\sigma(N)$. *Then, to finish the* k*th cycle, the length of the recurrence for computing* $\hat{q}_{\frac{k(k+1)}{2}+s}$ *is* $2k$ *for* $s = 0, \ldots, k$.

Proof. In the beginning of the kth cycle there are $(k-1)+k = 2k-1$ stored vectors. Then, to generate $\hat{q}_{\frac{k(k+1)}{2}+2}$, orthogonalize $N\hat{q}_{\frac{(k-1)k}{2}+1}$ against the vectors generated at the $(k-2)$nd and $(k-1)$st cycle as well as against $\hat{q}_{\frac{k(k+1)}{2}}$ and $\hat{q}_{\frac{k(k+1)}{2}+1}$, i.e., in all against $2k+1$ vectors. However, the first vector of the $(k-2)$nd cycle is already orthogonal against $N\hat{q}_{\frac{(k-1)k}{2}+1}$ since

$$(N\hat{q}_{\frac{(k-1)k}{2}+1}, \hat{q}_{\frac{(k-2)(k-1)}{2}}) = (\hat{q}_{\frac{(k-1)k}{2}+1}, N^*\hat{q}_{\frac{(k-2)(k-1)}{2}}) \tag{3.11}$$

and $N^*\hat{q}_{\frac{(k-2)(k-1)}{2}}$ is a linear combination of the vectors $\hat{q}_j$, with $0 \le j \le \frac{(k-1)k}{2}$. Thus, orthogonalization needs to be made only against $2k$ vectors. Similarly, to compute $\hat{q}_{\frac{k(k+1)}{2}+3}$ the first and second vectors of the $(k-2)$nd cycle are already orthogonal against $N\hat{q}_{\frac{(k-1)k}{2}+2}$ since

$$(N\hat{q}_{\frac{(k-1)k}{2}+2}, \hat{q}_{\frac{(k-2)(k-1)}{2}}) = (\hat{q}_{\frac{(k-1)k}{2}+2}, N^*\hat{q}_{\frac{(k-2)(k-1)}{2}}) \tag{3.12}$$

and

$$(N\hat{q}_{\frac{(k-1)k}{2}+2}, \hat{q}_{\frac{(k-2)(k-1)}{2}+1}) = (\hat{q}_{\frac{(k-1)k}{2}+2}, N^*\hat{q}_{\frac{(k-2)(k-1)}{2}+1}). \tag{3.13}$$

Now (3.11) is zero by the same argument as (3.12) and (3.13) is zero by the following arguments. By Theorem 3.4, the leading term of $\hat{q}_{\frac{(k-2)(k-1)}{2}+1}$ is $z\overline{z}^{(k-2)-1}$ multiplied by a constant. Therefore the leading term of $N^*\hat{q}_{\frac{(k-2)(k-1)}{2}+1}$ is $z\overline{z}^{(k-1)-1}$ multiplied by a constant. Consequently, $N^*\hat{q}_{\frac{(k-2)(k-1)}{2}+1}$ is a linear combination of the vectors $\hat{q}_j$, with $0 \le j \le \frac{(k-1)k}{2}+1$, and thus (3.12) is zero. This reasoning extends throughout the kth cycle in the sense that when $\hat{q}_{\frac{k(k+1)}{2}+s}$ is computed with $2 \le s \le k$, then orthogonalizations against $\hat{q}_j$ with $0 \le j \le \frac{(k-2)(k-1)}{2}+s-2$ are redundant. ∎

We actually could squeeze this length by one, i.e., to $2k-1$ by a minor modification.

Clearly $d = \frac{(k+1)(k+2)}{2}$ yields an upper bound on the dimension of the subspace $\operatorname{span}_{p\in\mathcal{PP}_k}\{p(N)\hat{q}_0\}$. Conversely, $k^2 \le 2d$ so that the number of saved vectors is bounded by $2\sqrt{2d}+1 = \sqrt{8d}+1$ when the subspace generated has dimension d. This is very modest growth.

If the minimal polyanalytic polynomial $p_{j,l}$ of $N \in \mathcal{N}$ is of low degree, then zero vectors will appear among $\{\hat{q}_j\}_{j\ge 0}$ according to Corollary 3.6. As their number increases at every cycle, these zero vectors are not saved in practice to save storage. Also, Algorithm 1 is tuned in such a way that zero vectors are discarded from the recurrence. With these modifications, the length of the recurrence does not grow but remains of fixed length from that point on.

4. Computing Ritz values. The matrix representation for the action of N in (3.2) can be used to compute Ritz values; for a wealth of information regarding Ritz values, see [29, 25, 9]. More precisely, with the ordering (3.2) of the orthogonalization

process, the canonical form $\hat{Q}^* N \hat{Q}$ of Elsner and Ikramov has the nonzero structure

$$
\begin{bmatrix}
\boldsymbol{\beta_0^0} & \beta_0^1 & & & & & & \\
\beta_1^0 & \boldsymbol{\beta_1^1} & \beta_1^2 & \beta_1^3 & & & & \\
\beta_2^0 & \beta_2^1 & \boldsymbol{\beta_2^2} & \beta_2^3 & \beta_2^4 & & & \\
 & \beta_3^1 & \beta_3^2 & \boldsymbol{\beta_3^3} & \beta_3^4 & \beta_3^5 & \beta_3^6 & \\
 & \beta_4^1 & \beta_4^2 & \beta_4^3 & \boldsymbol{\beta_4^4} & \beta_4^5 & \beta_4^6 & \\
 & & \beta_5^2 & \beta_5^3 & \beta_5^4 & \boldsymbol{\beta_5^5} & \beta_5^6 & \\
 & & & \beta_6^3 & \beta_6^4 & \beta_6^5 & \boldsymbol{\beta_6^6} & \cdots \\
 & & & \beta_7^3 & \beta_7^4 & \beta_7^5 & \beta_7^6 & \\
 & & & & \beta_8^4 & \beta_8^5 & \beta_8^6 & \\
 & & & & & \beta_9^5 & \beta_9^6 & \\
 & & & & & & \beta_{10}^6 & \\
 & & & & & & \beta_{11}^6 &
\end{bmatrix}
\tag{4.1}
$$

under the assumptions that the starting vector is generic and no $p \in \mathcal{PP}_k \backslash \{0\}$, with $\frac{(k+1)(k+2)}{2} < n$, vanishes on the spectrum of N. For clarity, the diagonal is boldfaced. Note that the nonzero structure is "nearly" symmetric, i.e., on the jth column there is at most one more nonzero element than on the jth row for every $j \geq 1$.

During the iteration we have

$$
N[\hat{q}_0\ \hat{q}_1 \cdots \hat{q}_{\frac{k(k-1)}{2}-1}] = [\hat{q}_0\ \hat{q}_1 \cdots \hat{q}_{\frac{(k+1)(k+2)}{2}-1}] N_k \tag{4.2}
$$

after k cycles have been completed. Here $N_k \in \mathbb{C}^{\frac{(k+1)(k+2)}{2} \times \frac{k(k+1)}{2}}$ is obtained from Algorithm 1 by collecting the values of the inner products as in the matrix representation (4.1). Removing from N_k the last $k+1$ rows gives a square matrix which is then used to compute Ritz values. The resulting matrices are not normal in general. However, because of the sparsity structure of the canonical form (4.1), they are nearly normal in the sense introduced in [19], that is, their self-commutators are of small rank.

If the minimal polyanalytic polynomial of N is of low degree, then there appear zero vectors among $\{\hat{q}_j\}_{j \geq 0}$. In the corresponding matrix representation (4.1) these vectors are obviously discarded as the columns of $\hat{Q}$ are orthonormal. For example, if N is Hermitian, then the arising matrix is exactly the one yielded by the Hermitian Lanczos method. Or, conversely, Algorithm 1 extends the Hermitian Lanczos method to normal matrices.

Behind the Ritz values computed, there is the characteristic polynomial p_k of the square matrix obtained from N_k. We do not know if there is any way to relate p_k to the polyanalytic polynomial $P_{j,l}$ realizing the minimum (3.1) at the corresponding step.

5. Generating eigenvalue exclusion regions. Besides Ritz values, the prescribed scheme yields other tools to locate the eigenvalues of large matrices. In Bauer's terminology, an exclusion region for the eigenvalues of $A \in \mathbb{C}^{n \times n}$ is a subset of $\mathbb{C}$ that contains no eigenvalues, and hence its complement contains all the eigenvalues of A; see [14, Chapter 19] and [15]. In particular, as opposed to Ritz values, exclusion regions can be regarded as yielding reliable information.

5.1. Normal matrices. The construction of eigenvalue exclusion regions for a given square matrix is a classical problem of matrix analysis; see, e.g., [13, Chapter 6]. Using lemniscates to locate the eigenvalues has been suggested at least in [24, section

2.10] and in [30, p. 262]. The difference between these two considerations is that in the former the proposed sets are guaranteed to contain the eigenvalues, whereas in the latter the regions are more heuristic. Both consider just using polynomials.

Employing Theorem 2.6, i.e., the fact that for any $p \in \mathcal{PP}$ the set

$$\{z \in \mathbb{C} : |p(z)| \leq ||p(N)||\} \tag{5.1}$$

contains the eigenvalues of $N \in \mathcal{N}$, can be used to generate eigenvalue exclusion regions. Because of the minimization property (3.1), we also have reasonable candidates from $\mathcal{PP}$ before any annihilating polyanalytic polynomials of N have been found. For a computed $p \in \mathcal{PP}$ we then need to consider ways to find, with sufficient accuracy,

$$\{z \in \mathbb{C} : |p(z)| = C\} \tag{5.2}$$

for those parts of the complex plane that are of interest. Crude approximations for finding (5.2) are available by using contour plotting with the existing mathematical software. For more accurate estimates one can proceed as follows. To find a point on the lemniscate, there are several efficient algorithms [5, 7] implemented on most available software packages. These can then be combined, for instance, with path following techniques to locate the further points on the lemniscate. A good general reference to this end is [1].

Regarding numerical stability of this process, the polyanalytic polynomials computed with Algorithm 1 have generically very low degree. In particular, according to Theorem 2.5, we have at most 1-dimensional information with a polyanalytic polynomial of degree not exceeding $\sqrt{2\deg(N)}$. This is in strong contrast with the Arnoldi method, which yields only analytic polynomials. Their degree grows linearly with the iteration number.

ALGORITHM 2. "Eigenvalue exclusion regions for a normal matrix $N \in \mathbb{C}^{n\times n}$."

Step 1. With Algorithm 1 and for $\hat{q}_0 \in \mathbb{C}^n$, generate a polyanalytic polynomial p.

Step 2. Estimate $C = ||p(N)||$.

Step 3. Find (5.2).

For estimating the spectral norm at Step 2 there are methods with $O(n^2)$ complexity [12]. Step 3 is computed only for those parts of the complex plane that are of interest, for example, by employing Gershgorin's disks first. Alternatively, Step 3 can be implemented by computing an inclusion of the set (5.2) by means of interval arithmetic (see, e.g., [23]), which has become readily available in several software implementations [20, 28].

5.2. Nonnormal matrices. For a nonnormal $A \in \mathbb{C}^{n\times n}$, exclusion regions for the eigenvalues are typically computed by using normal matrices as a tool. The most famous examples in this respect are Gershgorin's disks and the ovals of Cassini; see, e.g., [6] and the references therein. Exclusion regions with these methods are generated by employing very simple normal matrices; i.e., first the diagonal of A is extracted, and thereafter its eigenvalues are surrounded with sets in the well-known manner. Obviously, these estimates are not unitarily invariant.

For unitarily invariant estimates, assume $A \in \mathbb{C}^{n\times n}$ is nonnormal and let N be any normal approximant of A. The computation of an eigenvalue exclusion region for the eigenvalues of N with Algorithm 2 combined with the Bauer–Fike bound yields in a straightforward manner an exclusion region for the eigenvalues of A. Recall that the Bauer–Fike bound guarantees that

$$\operatorname{dist}(\lambda, \sigma(N)) \leq ||A - N|| \tag{5.3}$$

holds for any $\lambda \in \sigma(A)$, where $||\cdot||$ denotes the spectral norm. This provides a stable way to circumvent conceivably a very ill-conditioned nonnormal eigenproblem with a recurrence having a slowly growing length. Moreover, assume $S \subset \mathbb{C}$ containing the spectrum of N is so generated. Then it is elementary to verify that

$$\left|\left|(\lambda I - A)^{-1}\right|\right| \leq \frac{1}{\operatorname{dist}(\lambda, S) - ||A - N||} \tag{5.4}$$

holds as long as $\lambda \in \mathbb{C}$ is chosen so that $||A - N|| < \operatorname{dist}(\lambda, S)$. Thus, finding eigenvalue exclusion regions for normal matrices yields resolvent estimates for near nonnormal matrices. Obviously the key for the proposed scheme is to have sparse methods for finding normal matrices close to a given square matrix. An algorithm to this end was suggested in [16, 17].

6. Generating eigenvalue inclusion regions. In Bauer's terminology, an inclusion region for the eigenvalues of $A \in \mathbb{C}^{n\times n}$ is a subset of $\mathbb{C}$ that contains at least one eigenvalue of A; see [14, Chapter 19] and [15]. Corollary 2.7 as well as Theorem 2.10 yield an inclusion region for the eigenvalues of a normal matrix N. Orthogonal polyanalytic polynomials provide a further tool to this end.

6.1. Orthogonal polyanalytic polynomials. The process described in section 3 yields orthogonal polyanalytic polynomials on discrete subsets of $\mathbb{C}$ with respect to the following measure. Let $N = U\Lambda U^*$ be a diagonalization of a normal $N \in \mathbb{C}^{n\times n}$ by a unitary matrix U. Denote by $q_1, \ldots, q_n$ the columns of U and by $\lambda_1, \ldots, \lambda_n$ the corresponding eigenvalues. Without loss of generality, assume that the eigenvalues of N are distinct. We set an inner product in $\mathcal{PP}$ via

$$\langle p, q\rangle = (p(N)\hat{q}_0, q(N)\hat{q}_0) = \sum_{j=1}^{n} |(\hat{q}_0, q_j)|^2 \, p(\lambda_j)\overline{q(\lambda_j)} \tag{6.1}$$

for polyanalytic polynomials p and q. In particular, attaching to λ_j the mass $m_j = |(\hat{q}_0, q_j)|^2$ for $j = 1, \ldots, n$ yields a discrete measure on the spectrum of N. With this inner product we obtain orthogonal polyanalytic polynomials with Algorithm 1 as initialized in (3.3)–(3.7) after dividing by the multiplicative constants on the left-hand side. According to Theorem 3.7, the length of the recurrence for computing these orthogonal functions has a very modest growth.

THEOREM 6.1 (see [18]). *Assume $N \in \mathbb{C}^{n\times n}$ is normal with n distinct eigenvalues and $\hat{q}_0 \in \mathbb{C}^n$ is generic for N. Then Algorithm* 1 *produces n polyanalytic polynomials orthogonal with respect to the inner product* (6.1).

Example 9. This is Example 8 continued. Clearly N is unitary, so that the unit circle is an algebraic curve of degree 2 containing the spectrum. For this N and $\hat{q}_0$ Algorithm 1 produces zero vectors $\hat{q}_4 = \hat{q}_7 = \hat{q}_8 = 0$. Therefore we have $\mathbb{C}^8 = \operatorname{span}\{\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_5, \hat{q}_6, \hat{q}_9, \hat{q}_{10}\}$, and the corresponding orthogonal polyanalytic polynomials are $p_{0,0}(z) = 1$, $p_{0,1}(z) = \overline{z}$, $p_{1,0}(z) = z$, $p_{0,2}(z) = \overline{z}^2$, $p_{2,0}(z) = z^2$, $p_{0,3}(z) = \overline{z}^3$, $p_{3,0}(z) = z^3$, and $p_{0,4}(z) = \overline{z}^4$.

6.2. The Householder method for eigenvalue inclusion regions. The following approach for generating inclusion regions for the eigenvalues is, in essence, from [14, Chapter 20]. There, only the Arnoldi method is considered with analytic polynomials, whereas our purpose is to employ orthogonal polyanalytic polynomials obtained with Algorithm 1 without any additional cost.

THEOREM 6.2. *Assume $N \in \mathbb{C}^{n\times n}$ is normal, $\hat{q}_0 \in \mathbb{C}^n$, and $p_1, p_2 \in \mathcal{PP}$. If $p_2(N)\hat{q}_0 \neq 0$, then*

$$\left\{\lambda \in \mathbb{C} : \frac{||p_1(N)\hat{q}_0||}{||p_2(N)\hat{q}_0||} \leq \left|\frac{p_1(\lambda)}{p_2(\lambda)}\right|\right\} \tag{6.2}$$

is an inclusion region for the eigenvalues of N.

Proof. The reasoning follows [14, Chapter 20], except that now we consider polyanalytic polynomials. Assuming for the moment that $p_2(N)$ is invertible, we have

$$||p_1(N)\hat{q}_0|| = \left|\left|p_1(N)p_2(N)^{-1}p_2(N)\hat{q}_0\right|\right| \leq \left|\left|p_1(N)p_2(N)^{-1}\right|\right| ||p_2(N)\hat{q}_0||,$$

so that since $\left|\left|p_1(N)p_2(N)^{-1}\right|\right| = \max_{\lambda\in\sigma(N)}\left|\frac{p_1(\lambda)}{p_2(\lambda)}\right|$, we obtain

$$\frac{||p_1(N)\hat{q}_0||}{||p_2(N)\hat{q}_0||} \leq \max_{\lambda\in\sigma(N)}\left|\frac{p_1(\lambda)}{p_2(\lambda)}\right|. \tag{6.3}$$

Thus the claim follows, since at least one eigenvalue must be contained in the set (6.2).

If $p_2(N)$ is not invertible, then we proceed as follows. Since $N^* = q(N)$ for a polynomial q, we can assume without loss of generality that p_1 and p_2 are polynomials. Let $p_1(N)\hat{q}_0 = \tilde{p}_1(N)\tilde{q}_0$ and $p_2(N)\hat{q}_0 = \tilde{p}_2(N)\tilde{q}_0$ be such that $p_1(\lambda) = \tilde{p}_1(\lambda)p(\lambda)$ and $p_2(\lambda) = \tilde{p}_2(\lambda)p(\lambda)$, where $\tilde{p}_1, \tilde{p}_2 \in \mathcal{P}$ do not have common factors. Then if $\tilde{p}_2(N)$ is invertible, we have

$$\frac{||p_1(N)\hat{q}_0||}{||p_2(N)\hat{q}_0||} = \frac{||\tilde{p}_1(N)\tilde{q}_0||}{||\tilde{p}_2(N)\tilde{q}_0||} \leq \max_{\lambda\in\sigma(N)}\left|\frac{\tilde{p}_1(\lambda)}{\tilde{p}_2(\lambda)}\right| = \max_{\lambda\in\sigma(N)}\left|\frac{p_1(\lambda)}{p_2(\lambda)}\right|. \tag{6.4}$$

If $\tilde{p}_2(N)$ is not invertible, then the right-hand side of (6.4) is infinity and the claim is trivially true. ∎

COROLLARY 6.3. *Assume $p_1(N)\hat{q}_0$ and $p_2(N)\hat{q}_0$ are orthonormal and $\tau \in \mathbb{C}$ is such that $|\tau| = 1$. Then*

$$\left\{\lambda \in \mathbb{C} : 1 \leq \left|\frac{\tau p_1(\lambda) + p_2(\lambda)}{p_1(\lambda) - \overline{\tau}p_2(\lambda)}\right|\right\} \tag{6.5}$$

is an inclusion region for the eigenvalues of N.

Proof. This follows from

$$||(\tau p_1(N) + p_2(N))\hat{q}_0||^2 = ||(p_1(N) - \overline{\tau}p_2(N))\hat{q}_0||^2 = 2,$$

which with (6.2) proves the claim. ∎

We illustrate this with an example.

Example 10. This is Example 9 continued. After five iterations we know that the eigenvalues are on the unit circle. Take $p_1(z) = p_{0,2}(z) = \overline{z}^2$ and one more iterate to get $p_2(z) = p_{2,0}(z) = z^2$. Then with $\tau = -1$ the set (6.5) is $\{\lambda \in \mathbb{C} : 1 \leq |\frac{2xy}{x^2-y^2}|\}$. Thus, we obtain a sector in each of the 4 quadrants located symmetrically on the unit circle such that the one in the first quadrant is $\{e^{i\theta} : \frac{\pi}{8} \leq \theta \leq \frac{3\pi}{8}\}$. By Corollary 6.3 we can deduce that there is at least one eigenvalue in one of these sectors.

In practice these regions can be computed with a modification of Algorithm 2 by simply discarding Step 2, that is, the evaluation of the spectral norm.

7. Implementation and numerical examples. Next we consider issues related to implementation. Then finally we illustrate with numerical examples how Algorithms 1 and 2 perform. All the computations were done with `Matlab` version 5.3 [22] on a PC with IEEE double precision arithmetic running the Linux operating system.

7.1. Numerical implementation of Algorithm 1. The implementation of Algorithm 1 is straightforward except for the detection of zero vectors among $\hat{q}_0$, $\hat{q}_1$, $\ldots$, $\hat{q}_{\frac{k(k+1)}{2}}$ that correspond to annihilating polyanalytic polynomials of N. In finite precision the orthogonalization will not, due to round-off errors, yield exact zero vectors, and a more robust numerical criterion has to be used. From Algorithm 1 we notice that this also corresponds to deciding whether the vector being orthogonalized lies numerically in the span of the vectors it is being orthogonalized against. Several authors [8, 27], [25, page 113] present numerical algorithms for solving this problem in the context of updating a QR factorization after adding a column to the original matrix. Here we use a variant of the implementation from [27], based on modified Gram–Schmidt orthogonalization with one step of reorthogonalization.

Given k orthonormal vectors $Q \equiv [q_1, \ldots, q_k]$ and the vector w, the function ORTH computes the Fourier coefficients $s_i = (q_i, w)$, $i = 1, \ldots, k$, and $q_{k+1} = (I - QQ^*)\, w$, the orthogonal projection of w onto the complement of $\operatorname{span}\{q_1, \ldots, q_k\}$. If w is numerically in $\operatorname{span}\{q_1, \ldots, q_k\}$, then $q_{k+1} = 0$ is returned. On exit $s_{k+1} = \|q_{k+1}\|$.

THE FUNCTION ORTH.

function $\left[\, [s_1, \ldots, s_k, s_{k+1}],\, q_{k+1} \,\right] \;=\; \mathsf{ORTH}([q_1, \ldots, q_k],\, w)$
$\nu_0 = \|w\|$
for $i = 1 : k$
    $s_i = (q_i,\, w), \quad w = w - s_i q_i$
end
if $\|w\| > 0.707\nu_0$ **then**
    $q_{k+1} = w/\|w\|, \quad s_{k+1} = \|w\|$
    return
end
$\nu_1 = \|w\|$
for $i = 1 : k$
    $s'_i = (q_i,\, w), \quad w = w - s'_i q_i, \quad s_i = s_i + s'_i$
end
if $\|w\| > \max(0.707\nu_1,\, \epsilon_{tol}\, \nu_0)$ **then**
    $q_{k+1} = w/\|w\|, \quad s_{k+1} = \|w\|$
else
    /* w lies numerically in $\operatorname{span}\{q_1, \ldots, q_k\}$ */
    $q_{k+1} = 0, \quad s_{k+1} = 0$
end

In our implementation ϵ_{tol} was chosen as $2l\mathbf{u}$, where l is the length of w and $\mathbf{u}$ is the unit round-off in the floating point arithmetic used. When ORTH is used, Algorithm 1 can be written in the following compact form.

ALGORITHM 1 USING ORTH.

Assume $N \in \mathcal{N}$ and $\hat{q}_0$ is of unit length.
for $k = 1 : K$
    Define $k_0 \equiv (k-2)(k-1)/2$, $k_1 \equiv (k-1)k/2$, $k_2 \equiv k(k+1)/2$
    $q_{k_2} = N^*\, \hat{q}_{k_1}$

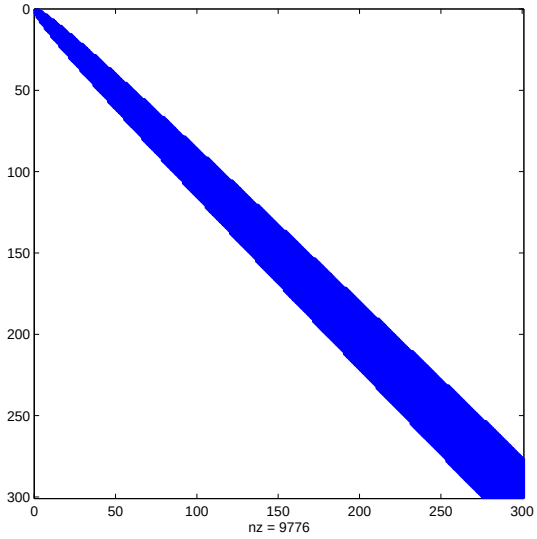


FIG. 3. *The sparsity structure of the canonical form of Elsner and Ikramov for the random normal matrix* $N \in \mathbb{C}^{300\times 300}$ *of Example* 11.

$$\left[[\alpha_{k_0}^{k_1}, \ldots, \alpha_{k_2-1}^{k_1}, \alpha_{k_2}^{k_1}],\ \hat{q}_{k_2} \right] \ = \ \mathsf{ORTH}([\hat{q}_{k_0}, \ldots, \hat{q}_{k_2-1}],\ q_{k_2})$$

for $l = k_1 : k_2 - 1$

$q_{k+l+1} = N\,\hat{q}_l$

$$\left[[\beta_{l-k+1}^{l}, \ldots, \beta_{l+k}^{l}, \beta_{k+l+1}^{l}],\ \hat{q}_{k+l+1} \right] \ = \ \mathsf{ORTH}([\hat{q}_{l-k+1}, \ldots, \hat{q}_{l+k}],\ q_{k+l+1})$$

end

end

The behavior of Algorithm 1 in finite precision seems to be similar to that of the Lanczos algorithm in the sense that some loss of orthogonality occurs among vectors from different cycles after many steps have been performed. As in the Hermitian Lanczos algorithm, this phenomenon appears to be closely connected to the convergence of Ritz values associated with (4.2). None of the examples below were affected by this effect.

7.2. Numerical examples. We demonstrate Algorithms 1 and 2 with four examples. The first one illustrates how far it is possible to iterate with these algorithms, due to the mild growth of the length of the recurrence, as opposed to the Arnoldi method. In the two examples that then follow we compare Ritz values computed with Algorithm 1 and the Arnoldi method. The final example deals with generating exclusion regions for the eigenvalues.

Example 11. In this example we spy the sparsity structure of the canonical form of Elsner and Ikramov (4.1) by taking a random diagonal normal matrix $N \in \mathbb{C}^{300\times 300}$. Already this small example illustrates well the significance of the factor $\sqrt{8d}$ of Theorem 3.7 for the length of recurrence, since we only need to store $\lfloor \sqrt{8 * 300} \rfloor \leq 50$ vectors to compute (4.1), as plotted in Figure 3. For comparison, with the Arnoldi method only a tiny Hessenberg matrix—$\frac{1}{6} \times \frac{1}{6}$ of the size of the computed canonical form in Figure 3—could be generated with the same amount of storage.

Example 12. In this example we compute Ritz values with the aid of the canonical form (4.1) and the relation (4.2). We consider a "larger" version of [21, Example 1];

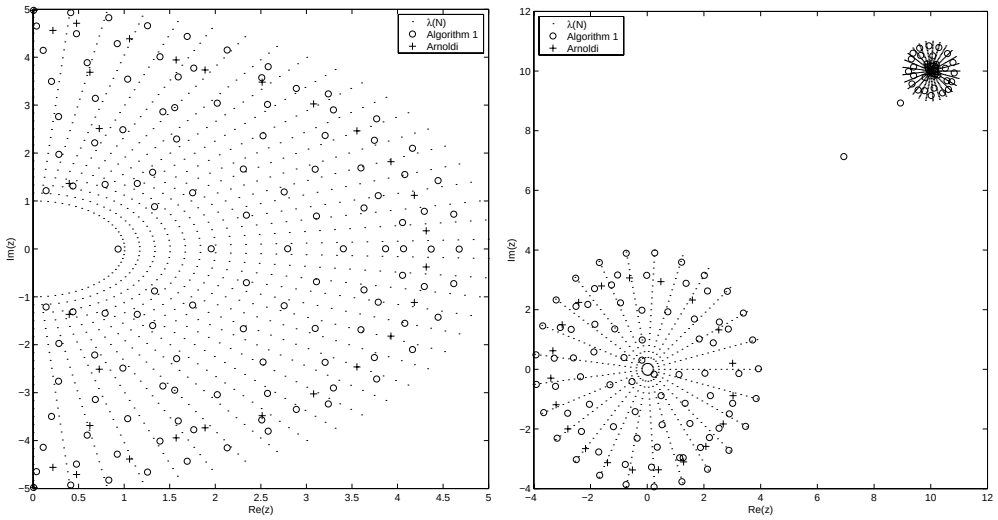


FIG. 4. *Ritz values computed by Algorithm* 1 *(designated by "∘") and the Arnoldi algorithm (designated by "+") for the matrices in Example* 12 *(left panel) and Example* 13 *(right panel).*

that is, we assume N to be of size 1000 with eigenvalues uniformly distributed in the half-annulus $\{z \in \mathbb{C} : 1 \leq |z| \leq 5,\ \mathrm{Re}(z) \leq 0\}$. We assume that the storage is the bottleneck, so that at most 30 vectors can be saved. This means that we can compute a matrix of size 30 and 120 with the Arnoldi method and Algorithm 1, respectively, that is, $\lfloor \sqrt{8 * 120} \rfloor \leq 30$. In the left panel of Figure 4 we have plotted the corresponding approximations obtained with the starting vector $\hat{q}_0 = \frac{1}{\sqrt{1000}}[1, 1, \ldots, 1]^T$. Note how uniformly Algorithm 1 generates Ritz values with respect to $\sigma(N)$ compared with the Arnoldi method.

Example 13. In this example let N be of size 1000, and again we compute Ritz values. We assume that the eigenvalues of N are contained uniformly inside two "disks" as follows. One is centered at the origin of radius 4 and the other is centered at the point $10 + 10i$ of radius 1. Again we assume that the storage is the bottleneck, so that at most 30 vectors can be saved. Thus we can compute a matrix of size 30 and 120 with the Arnoldi method and Algorithm 1, respectively. In the right panel of Figure 4 we have plotted the corresponding approximations obtained with the starting vector $\hat{q}_0 = \frac{1}{\sqrt{1000}}[1, 1, \ldots, 1]^T$. Again, note how uniformly Algorithm 1 yields Ritz values with respect to $\sigma(N)$ compared with the Arnoldi method. The two Ritz values outside are "on their way" to the disk centered at $10 + 10i$.

Example 14. In this example we illustrate Algorithm 2. Let $N \in \mathbb{C}^{200\times 200}$ be the unitary Hessenberg matrix where the elements on the first subdiagonal and the $(1, 200)$-element are 1 while the other elements are zero. Let $N = U\Lambda U^*$ be a diagonalization of N by a unitary U, where Λ is diagonal with the numbers $\{z \in \mathbb{C} : z^{200} = 1\}$ on the diagonal. We perturb Λ slightly with a random diagonal matrix Δ with $\|\Delta\| \leq 0.2$ to have $\tilde{N} = U(\Lambda + \Delta)U^*$. Obviously $\tilde{N} \in \mathcal{N}$ although $\tilde{N}$ is not unitary anymore. Also, Gershgorin's disks are not very accurate as $\tilde{N}$ is close to N, for which Gershgorin's disks are just disks of radius 1 centered at the origin. In Figure 5 we have plotted examples of eigenvalue exclusion regions for $\tilde{N}$ computed by Algorithm 2 with the starting vector $\hat{q}_0 = \frac{1}{\sqrt{1000}}[1, 1, \ldots, 1]^T$. As can be seen from

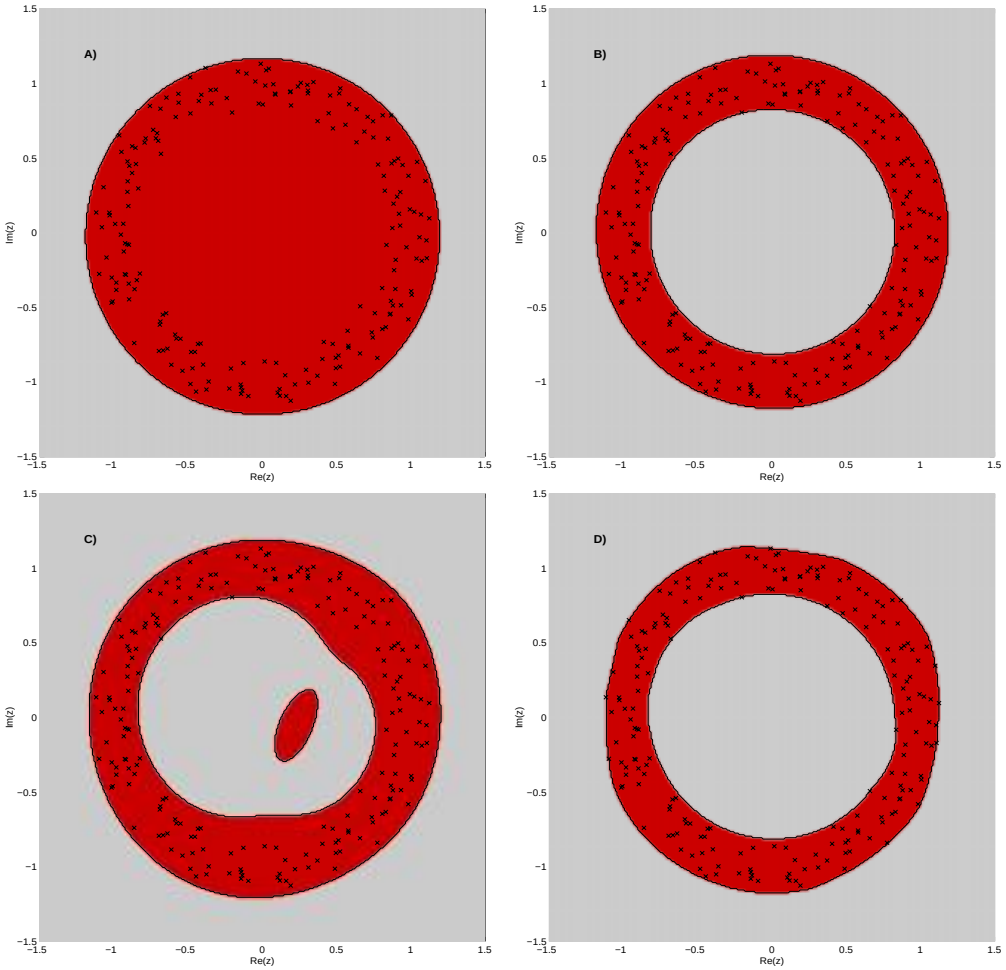


FIG. 5. *Examples of eigenvalue exclusion regions computed by Algorithm* 2 *for the matrix from Example* 14. *Panels A, B, and C show the sets* $\{z \in \mathbb{C} : |q(z)| \leq \|q(\tilde{N})\|\}$ *for* $q \in \{p_{1,0}, p_{1,1}, p_{4,0}\}$. *Panel D shows the union of the eigenvalue exclusion regions corresponding to the first* 15 *polyanalytic polynomials generated by Algorithm* 1. *Crosses indicate the position of the eigenvalues of* $\tilde{N}$.

panel B, very accurate information about the location of the spectrum is obtained with the polynomial $p_{1,1}$, which is generated after performing only five steps with Algorithm 1. For an illustration, two other exclusion regions corresponding to $p_{1,0}$ and $p_{4,0}$ are plotted in panels A and C. Panel C was obtained after 15 iterations. In panel D we have taken the union of all 15 exclusions regions computed so far.

As a final comment regarding Example 14, recall that by Corollary 2.7 the boundaries of these exclusion regions always contain at least one eigenvalue of N. So in panels A, B, and C, there is an eigenvalue on each of the boundaries.

8. Conclusions. Considering the 2-by-2 nonlinear system of real polynomials given by the real and imaginary parts of the characteristic polynomial naturally leads one to consider real analytic techniques for finding the eigenvalues. Starting from this observation, we have shown that the spectrum of a normal matrix can be found exactly without resorting to the characteristic polynomial. Besides Ritz values, the

algorithm introduced generates low degree polyanalytic polynomials with a slowly growing length of the recurrence. These can be used to generate exclusion regions for the eigenvalues. Orthogonal polyanalytic polynomials can be used in finding eigenvalue inclusion regions. Thus, with the iterative method suggested, these 3 different tools can be employed simultaneously for eigenapproximation.

Acknowledgments. We are very grateful to the anonymous referees for their careful reading and extremely useful comments on an earlier version of the paper. As a result, the paper improved significantly.

REFERENCES

[1] E. Allgower and K. Georg, *Continuation and path following*, Acta Numer., 1993, pp. 1–64.
[2] W.E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
[3] M.B. Balk, *Polyanalytic Functions*, Akademie-Verlag, Berlin, 1991.
[4] P. Borwein, *The arc length of the lemniscate* $\{|p(z)| = 1\}$, Proc. Amer. Math. Soc., 123 (1995), pp. 797–799.
[5] R. Brent, *An algorithm with guaranteed convergence for finding a zero of a function*, Comput. J., 14 (1971), pp. 422–425.
[6] R. Brualdi and S. Mellendorf, *Regions in the complex plane containing the eigenvalues of a matrix*, Amer. Math. Monthly, 101 (1994), pp. 975–985.
[7] J. Bus and T. Dekker, *Two efficient algorithms with guaranteed convergence for finding a zero of a function*, ACM Trans. Math. Software, 1 (1975), pp. 330–345.
[8] J.W. Daniel, W.B. Gragg, L. Kaufman, and G.W. Stewart, *Reorthogonalization and stable algorithms updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.
[9] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[10] L. Elsner and Kh.D. Ikramov, *On a condensed form for normal matrices under finite sequence of elementary similarities*, Linear Algebra Appl., 254 (1997), pp. 79–98.
[11] R. Grone, C.R. Johnson, E.M. Sa, and H. Wolkowicz, *Normal matrices*, Linear Algebra Appl., 87 (1987), pp. 213–225.
[12] N.J. Higham, *Estimating the matrix p-norm*, Numer. Math., 62 (1992), pp. 511–538.
[13] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1987.
[14] A.S. Householder, *Lectures on Numerical Algebra*, Mathematical Association of America, Buffalo, NY, 1972.
[15] A.S. Householder, *Principles of Numerical Analysis*, Dover, New York, 1974.
[16] M. Huhtanen, *A stratification of the set of normal matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 349–367.
[17] M. Huhtanen, *Splitting Matrices into a Sum of Two Normal Matrices*, manuscript.
[18] M. Huhtanen, *Orthogonal polyanalytic polynomials and normal matrices*, Math. Comp., to appear.
[19] M. Huhtanen and O. Nevanlinna, *Minimal decompositions and iterative methods*, Numer. Math., 86 (2000), pp. 257–281.
[20] O. Knüppel, *PROFIL/BIAS—A Fast Interval Library*, Computing, 53 (1994), pp. 277–287.
[21] T. Manteuffel and G. Starke *On hybrid iterative methods for nonsymmetric systems of linear equations*, Numer. Math, 73 (1996), pp. 489–506.
[22] Mathworks, *Matlab*, available online from www.mathworks.com/products/matlab.
[23] R.E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[24] O. Nevanlinna, *Convergence of Iterations for Linear Equations*, Lectures in Mathematics ETH Zürich, Birkhäuser-Verlag, Basel, 1993.
[25] B.N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.
[26] V. Pták and J. Zemánek, *Continuité lipschitzienne du spectre comme fonction d'un opérateur normal*, Comment. Math. Univ. Carolin., 17 (1976), pp. 507–512.
[27] L. Reichel and W.B. Gragg, *Algorithm* 686*: FORTRAN subroutines for updating the QR decomposition*, ACM Trans. Math. Software, 16 (1990), pp. 369–377.
[28] S.M. Rump, *INTLAB—INTerval LABoratory*, in Developments in Reliable Computing, T. Csendes, ed., Kluwer Academic, Dordrecht, The Netherlands, 2000, pp. 77–105.
[29] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York, 1992.
[30] L.N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1092–1108
© 2002 Society for Industrial and Applied Mathematics

A HERMITIAN LANCZOS METHOD FOR NORMAL MATRICES*

MARKO HUHTANEN†

Abstract. We present an algorithm for iteratively solving a linear system $Nx = b$, with a normal $N \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$, with an optimal 3-term recurrence by extending the Hermitian Lanczos method to normal matrices. This is achieved by considering the Toeplitz decomposition $N = H + iK$ of N with Hermitian H and K. Generically, the inverse of a normal matrix is a polynomial in its Hermitian part. Using this and the fact that N and H commute, we obtain a minimization problem

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(H)b - b|| = \min_{p_{j-1}\in\mathcal{P}_{j-1}} ||p_{j-1}(H)Nb - b||\,,$$

where $\mathcal{P}_{j-1}$ denotes the set of polynomials of degree $j-1$ at most. Thus, at the jth step, the best approximation to b needs to be found from the Krylov subspace $\mathcal{K}_j(H; Nb)$. Since this involves the Hermitian matrix H, this is realizable with a 3-term recurrence.

Key words. normal matrix, GMRES, Toeplitz decomposition, Hermitian Lanczos method

AMS subject classification. 65F10

PII. S0895479800398468

1. Introduction.

Assume $N \in \mathbb{C}^{n\times n}$ is invertible and $b \in \mathbb{C}^n$ and consider solving the linear system

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

using an optimal k-term recurrence. By an optimal k-term recurrence we mean a method that produces approximate solutions to (1.1) characterized by a minimization property over a Krylov subspace and which can be obtained recursively from the most recently generated k vectors. It is well known that an optimal 3-term recurrence is obtained by considering the normal equations instead. Then the original linear system is converted into a completely different one by multiplying (1.1) with N^* from the left. It is equally well known that this may not be a good idea as the speed of convergence of iterations can be very slow for this new system. This can be explained, at least partially, by the squared conditioning of the altered coefficient matrix. Consequently, a major problem of numerical linear algebra, posed by Golub [31], was that of devising an optimal k-term recurrence, with $k \ll n$, for solving (1.1) that avoids using the normal equations or a proof that there cannot be such a method.

In [7] Faber and Manteuffel showed that there does not exist an optimal k-term recurrence (in any of the norms that are typically of interest) for solving (1.1) relying on matrix-vector multiplications with N only, i.e., by generating Krylov subspaces with N, unless N is normal. See also [32, 8]. Their result is very negative in the sense that even then, except for a few anomalies, the eigenvalues of N must be contained in a line. If this is the case, then $k = 3$. Thus, it seems that without employing the normal equations there does not exist an optimal short term recurrence for solving (1.1) for any readily available class of matrices aside from translations and rotations

*Received by the editors August 26, 2000; accepted for publication (in revised form) by G. H. Golub November 12, 2001; published electronically April 10, 2002.
http://www.siam.org/journals/simax/23-4/39846.html

†Institute of Mathematics, Helsinki University of Technology, Espoo, FIN-02150, Finland. Current address: Department of Mathematics, Room 2-335, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 01239 (huhtanen@math.mit.edu).

of Hermitian matrices. For a scheme that is in a less strict sense a Krylov subspace method, see [30].

In this paper we show that if N is normal, then there exists an optimal 3-term recurrence for solving the linear system (1.1) without resorting to the normal equations by extending the Hermitian Lanczos method to the set of normal matrices $\mathcal{N}$. To this end we employ the Toeplitz decomposition of N defined via

$$N = H + iK, \tag{1.2}$$

with Hermitian $H = \frac{1}{2}(N + N^*)$ and $K = \frac{1}{2i}(N - N^*)$. The key is to notice that a generic normal matrix N is a polynomial in its Hermitian part H obtained from the Toeplitz decomposition [18]. Since N is assumed to be invertible, by elementary linear algebra, its inverse is a polynomial in N. Consequently, we can deduce that N^{-1} is generically a polynomial in H.

Although seldom stated explicitly, the simple fact that the inverse of N is a polynomial in N is the reason behind the success of many iterative methods. Denoting by $\mathcal{P}_{j-1}$ the set of polynomials of degree $j-1$ at most, this is most concretely seen from the GMRES [29] minimization problem

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(N)b - b|| \tag{1.3}$$

at the jth step. In practice the inverse can be approximated with very low degree polynomials in N only because, unless N is a translation and a rotation of a Hermitian matrix, the length of the recurrence as well as the need for storage grow linearly with the iteration number. To circumvent this we do not construct Krylov subspaces with the coefficient matrix. Instead, since for a normal N the inverse is generically a polynomial in H, we replace the GMRES optimality condition (1.3) with

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(H)b - b|| \,. \tag{1.4}$$

The reasoning behind the minimization problems (1.3) and (1.4) is based on exactly the same argument, that is, N^{-1} is a polynomial in N and H, respectively.

The key to solving the minimization problem (1.4) with a 3-term recurrence is to employ the fact that N and H commute. Consequently, (1.4) equals

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||p_{j-1}(H)Nb - b|| \,, \tag{1.5}$$

which can be readily solved by computing the best approximation to the vector b from the Krylov subspace $\mathcal{K}_j(H; Nb) = \text{span}\{Nb, HNb, \ldots, H^{j-1}Nb\}$. Using the Hermitian Lanczos method with H and the starting vector $\hat{q}_0 = \frac{Nb}{\|Nb\|}$ yields the value of (1.5) and thereby that of (1.4) with an optimal 3-term recurrence. This does not, however, yield an approximation to the solution of the linear system (1.1) unless we multiply once with the inverse of N. This is obviously not what we suggest, as the inverse is not assumed to be available. Instead, we use the Hermitian Lanczos recurrence to generate the standard Lanczos vectors, but premultiplied by N^{-1}. However, we are able to do without ever applying N^{-1} because of our starting vector for the iteration. This gives rise to a 3-term recurrence for solving (1.1) with the same storage requirements as in the classical Hermitian Lanczos method.

At each step j the approximate solution generated satisfies the optimality condition (1.4). For a Hermitian N this condition is the GMRES minimization problem.

Thus, we extend MINRES for Hermitian matrices to normal matrices without losing the optimality or increasing the length of the recurrence. This demonstrates, combined with the results of [18], that the set of Hermitian matrices does not enjoy any particular algorithmic advantages over its complement in $\mathcal{N}$. Of course, the qualitative behavior of the algorithm can be very varied for different matrices from $\mathcal{N}$.

In addition to the basic algorithm, we consider its restarted and rotated implementation aimed at nongeneric or nearly nongeneric normal matrices. These include, in particular, those normal N which are not polynomials in their Hermitian part. However, generically, a rotation of N is a polynomial in its Hermitian part. Thus, a solution is to restart the basic algorithm with a rotated $e^{i\theta}N$ for $\theta \in [0, 2\pi)$. Values for θ can be chosen based on some a priori information or they can be randomly generated periodically. This can speed up the convergence dramatically and, in particular, it is simple to code to increase adaptivity in the basic algorithm.

The scheme proposed cannot be extended beyond normal matrices as such. In particular, it seems that the barrier between normal matrices and nonnormal matrices is impenetrable. However, an optimal method for solving linear systems involving normal matrices can be utilized when solving nonnormal problems. An obvious alternative is to use the method introduced in preconditioning. Aside from this, we discuss how the algorithm is related to solving linear systems in general.

The paper is organized as follows. In section 2 we consider properties of the Toeplitz decomposition for normal matrices. These properties are then used in deriving a minimization problem for the elements of $\mathcal{N}$ that is analogous to the ideal GMRES problem [13]. In section 3 we introduce a local version of this which gives rise to a Hermitian Lanczos method for solving linear systems involving normal matrices. In section 4 we consider how the methods obtained relate to solving nonnormal linear systems. Numerical experiments are presented in section 5.

2. The Toeplitz decomposition for normal matrices. The set of normal matrices is a large class of matrices containing, for instance, the set of Hermitian, skew-Hermitian, circulant, and unitary matrices. For numerical manipulations normal matrices are particularly well suited, as a matrix is normal if and only if it is unitarily similar to a diagonal matrix. Although this is how normality is most often used in practice, the original definition is purely algebraic in the sense that $N \in \mathbb{C}^{n\times n}$ is defined to be normal if

$$NN^* - N^*N = 0 \tag{2.1}$$

holds. Besides these two characterizations, there are many other ways to define normality. So far about 90 equivalent conditions for a matrix to be normal have been collected by Grone et al. [15] and by Elsner and Ikramov [5]. The characterization (2.1) is taken to be the first in these listings although it is numbered as the condition zero.

Given the abundance of characterizations of normality, it is not surprising that several of them deal with the canonical decompositions. Being unitarily diagonalizable is, for instance, one such. Another is based on the Toeplitz decomposition

$$N = H + iK \tag{2.2}$$

of N, with $H = \frac{1}{2}(N + N^*)$ and $K = \frac{1}{2i}(N - N^*)$. The condition 21 in [15] states that N is normal if and only if H and K commute. In what follows, we consider the Toeplitz decomposition for normal matrices in more detail.

Commutativity is both a powerful tool and a stringent condition. In particular, it is well known that for a nonderogatory matrix $A \in \mathbb{C}^{n\times n}$ the set of matrices commuting with A equals the set of polynomials in A [16]. With normal matrices this can be used as follows. We denote by $\mathcal{N}_n \subset \mathbb{C}^{n\times n}$ the set of normal matrices N whose Hermitian part $H = \frac{1}{2}(N + N^*)$ is nonderogatory. In our claims regarding normal matrices we use the induced topology of the standard metric topology of $\mathbb{C}^{n\times n}$.

THEOREM 2.1 (see [18]). *$\mathcal{N}_n$ is an open dense subset of $\mathcal{N}$.*

This set is readily parametrizable. Namely, if we denote by $\mathcal{H}_n \in \mathbb{C}^{n\times n}$ the set of nonderogatory Hermitian matrices, then the mapping

$$(H, \alpha_0, \dots, \alpha_{n-1}) \to H + i\sum_{j=0}^{n-1} \alpha_j H^j \tag{2.3}$$

from $\mathcal{H}_n \times \mathbb{R}^n$ onto $\mathcal{N}_n$ is injective. In particular, combining this with Theorem 2.1, we can deduce that a generic normal matrix N is a polynomial in its Hermitian part H. This is also true for certain normal matrices not belonging to $\mathcal{N}_n$. By a generic property in a set S we mean that it holds for an open dense subset of S. We denote by $\sigma(A)$ the spectrum and by $\#\sigma(A)$ the number of *distinct* eigenvalues of $A \in \mathbb{C}^{n\times n}$.

THEOREM 2.2. *Assume $N = H + iK$ is normal such that $\#\sigma(N) = \#\sigma(H)$. Then $N = H + ip(H)$ for a polynomial p of degree $\#\sigma(N) - 1$ at most.*

Proof. Let N be unitarily diagonalizable by a unitary matrix U so that its Toeplitz decomposition is

$$N = U \begin{bmatrix} \Re(\lambda_1) & & \\ & \ddots & \\ & & \Re(\lambda_n) \end{bmatrix} U^* + iU \begin{bmatrix} \Im(\lambda_1) & & \\ & \ddots & \\ & & \Im(\lambda_n) \end{bmatrix} U^* = H + iK, \tag{2.4}$$

where $\lambda_1, \dots, \lambda_n$ denote the eigenvalues of N, counting multiplicities, arranged in nondecreasing order of modulus. Since $\#\sigma(N) = \#\sigma(H)$, we can construct the Lagrange interpolation polynomial p attaining the values $\Im(\lambda_1), \dots, \Im(\lambda_n)$ at points $\Re(\lambda_1), \dots, \Re(\lambda_n)$. The degree of p is $\#\sigma(N) - 1$ and clearly $N = H + ip(H)$. ∎

The property $\#\sigma(N) = \#\sigma(H)$ is generic not only in $\mathcal{N}$ but also in subsets of $\mathcal{N}$ that are relevant for our purposes.

PROPOSITION 2.3. *The property $\#\sigma(N) = \#\sigma(H)$ is generic in*

$$\{N = H + iK \in \mathcal{N} | \#\sigma(N) = k\} \tag{2.5}$$

for every $1 \le k \le n$.

Proof. Assume $N \in \{N = H + iK \in \mathcal{N} | \#\sigma(N) = k\}$ and let $\hat{\lambda}_1, \dots, \hat{\lambda}_k$ denote the distinct eigenvalues of N sorted by real part. Define a mapping

$$N \to \prod_{j=1}^{k-1} (\Re(\hat{\lambda}_{j+1}) - \Re(\hat{\lambda}_j)) \tag{2.6}$$

from $\{N = H + iK \in \mathcal{N} | \#\sigma(N) = k\}$ into $\mathbb{R}$. This is clearly continuous. The inverse image of 0 for (2.6) equals those $N \in \{N = H + iK \in \mathcal{N} | \#\sigma(N) = k\}$ for which $\#\sigma(N) > \#\sigma(H)$, which thereby is a closed set. To see that its complement is dense, rotations $e^{i\theta}N$ of N remain in the set (2.5). Choosing an arbitrarily small positive θ results in $\#\sigma(e^{i\theta}N) = \#\sigma(\frac{1}{2}(e^{i\theta}N + e^{-i\theta}N^*))$. ∎

With these preliminaries, assume one has an invertible, not necessarily normal matrix $A \in \mathbb{C}^{n\times n}$. By elementary linear algebra, $A^{-1} = p(A)$ for a polynomial p. Based on this, in the ideal GMRES approximation problem [13] one considers, for $1 \le j \le n$,

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Ap_{j-1}(A) - I||\,, \tag{2.7}$$

where $\mathcal{P}_{j-1}$ denotes the set of polynomials of degree $j-1$ at most. Typically the behavior of these quantities is of interest for only small values of j compared with the dimension n. The reason for this is that (2.7) can be related to solving a linear system $Ax = b$, with a vector $b \in \mathbb{C}^n$, by using GMRES via the inequality

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Ap_{j-1}(A)b - b|| \le \min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Ap_{j-1}(A) - I||\,||b||\,. \tag{2.8}$$

Since in general both the length of the GMRES recurrence and the required storage grow linearly with the iteration number, one typically must accept $j \ll n$ in practice. Therefore the problems of understanding the convergence behavior of ideal GMRES as well as GMRES for solving $Ax = b$ have received a lot attention in the 1990s; see [29, 11, 24, 14] and references therein as well as the more recent [21, 19, 17, 20]. For normal matrices both of these problems can be considered to be understood [12].

There is an algebraic property of A bounding the ultimate number of steps needed for solving $Ax = b$ exactly (in exact arithmetic) with GMRES. To this end, recall that the minimal polynomial of A is the monic polynomial of least degree annihilating A. Let $\deg(A)$ denote the degree of the minimal polynomial of $A \in \mathbb{C}^{n\times n}$.

PROPOSITION 2.4. *Assume $A \in \mathbb{C}^{n\times n}$ is invertible. Then $A^{-1} = p(A)$ for a polynomial of degree* $\deg(A) - 1$.

This is well known and can be found, e.g., in [24, 1]. In particular, due to the inequality (2.8), after at most $\deg(A)$ steps GMRES yields the solution to the corresponding linear system.

For normal matrices Theorem 2.2 gives rise to a problem analogous to the ideal GMRES problem in the following manner. Namely, according to Theorem 2.2, a generic normal matrix N is a polynomial in its Hermitian part $H = \frac{1}{2}(N + N^*)$. Assuming N to be invertible, it follows that N^{-1} is a polynomial in N. Consequently, the inverse of a generic normal invertible matrix N is a polynomial in its Hermitian part. Regarding Proposition 2.4, we have the following analogy.

THEOREM 2.5. *Assume $N \in \mathcal{N}$ is invertible and generic in the sense of Proposition* 2.3. *Then $N^{-1} = p(H)$ for a polynomial p of degree* $\deg(N) - 1$ *at most.*

Proof. Let $N = U\Lambda U^*$ be a diagonalization of N by a unitary matrix U so that its Toeplitz decomposition is as in (2.4). Since N is normal, there holds $\#\sigma(N) = \deg(N)$ and being generic in the sense of Proposition 2.3 implies that $\deg(H) = \deg(N)$.

The Toeplitz decomposition $N^{-1} = \hat{H} + i\hat{K}$ of N^{-1} is

$$N^{-1} = U \begin{bmatrix} \Re(1/\lambda_1) & & \\ & \ddots & \\ & & \Re(1/\lambda_n) \end{bmatrix} U^* + iU \begin{bmatrix} \Im(1/\lambda_1) & & \\ & \ddots & \\ & & \Im(1/\lambda_n) \end{bmatrix} U^*. \tag{2.9}$$

By using Lagrange interpolation, find a real polynomial p_1 that attains the values $\Re(1/\lambda_1), \dots, \Re(1/\lambda_n)$ at points $\Re(\lambda_1), \dots, \Re(\lambda_n)$. Then, analogously, find a real polynomial p_2 attaining the values $\Im(1/\lambda_1), \dots, \Im(1/\lambda_n)$ at the points $\Re(\lambda_1), \dots, \Re(\lambda_n)$.

The degrees of both p_1 and p_2 are at most $\deg(N)-1$. By this construction we have $p(H) = p_1(H) + ip_2(H) = N^{-1}$ and the claim follows. ∎

Thus, since the inverse of $N \in \mathcal{N}$ is generically a polynomial in its Hermitian part, we are naturally led to consider, aside from the ideal GMRES problem,

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(H) - I|| \tag{2.10}$$

for $1 \le j \le n$. According to Theorem 2.5, zero is attained in (2.10) no later than in the ideal GMRES problem whenever N is generic. As with (2.7), for practical purposes the behavior of the quantities (2.10) is of most interest for values of j far smaller than the underlying dimension n. However, as opposed to solving complex polynomial approximation problems on the spectrum of N, the corresponding approximation problems are solved on the spectrum of H, that is, on a subset of $\mathbb{R}$. From the proof of Theorem 2.5 we obtain the following obvious bound.

COROLLARY 2.6. *Let p_r and p_i solve $\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||\Re(\Lambda^{-1}) - p_{j-1}(\Lambda_H)||$ and $\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||\Im(\Lambda^{-1}) - p_{j-1}(\Lambda_H)||$, respectively. Then*

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} \left|\left|p_{j-1}(H) - N^{-1}\right|\right| \le \left|\left|p_r(H) + ip_i(H) - N^{-1}\right|\right|.$$

Whenever N is Hermitian, then (2.10) reduces to the standard ideal GMRES problem for N. In other words, the minimization problem (2.10) is a continuous extension of the ideal GMRES problem for Hermitian matrices to the set of normal matrices. This extension is clearly not the same as (2.7).

A given normal matrix can be nongeneric in the sense that it is not a polynomial in its Hermitian part. This is the case if N is skew-Hermitian, because then the Hermitian part equals zero. Or more generally, if the matrix has more than one eigenvalue located on a vertical line, then the Lagrange interpolation polynomial of Theorem 2.2 is not computable. There is a simple trick to overcome this problem. Namely, instead of N, consider its rotation $e^{i\theta}N$ with $\theta \in [0, 2\pi)$. It is obvious that a rotation of a normal matrix remains normal.

THEOREM 2.7 (see [18]). *Assume $N \in \mathcal{N}$. Then, for θ belonging to an open dense subset of $[0, 2\pi)$, there holds*

$$e^{i\theta}N = H_\theta + ip_\theta(H_\theta)$$

with $H_\theta = \frac{1}{2}(e^{i\theta}N + e^{-i\theta}N^*)$ and for a polynomial p_θ with real coefficients.

By the same reasoning as before, we have, for an invertible $N \in \mathcal{N}$ and for θ belonging to an open dense subset of $[0, 2\pi)$, a representation

$$N^{-1} = p_\theta(H_\theta) \tag{2.11}$$

for the inverse with a polynomial p_θ. It is readily seen that the claim of Theorem 2.5 holds as well; i.e., the degree of p_θ in (2.11) is at most $\deg(N)-1$. However, it is not obvious how to pick a rotation θ yielding an optimal decay in the corresponding minimization problem (2.10). Still, since these rotations are important for our purposes, we introduce the following definition.

DEFINITION 2.8. *For $N \in \mathbb{C}^{n\times n}$ and $\theta \in [0, 2\pi)$ let H_θ and K_θ denote the Hermitian and skew-Hermitian part of $e^{i\theta}N$, respectively. Then $N = e^{-i\theta}H_\theta + ie^{-i\theta}K_\theta$ is the rotated Toeplitz decomposition of N by the angle θ.*

In this decomposition the parts are typically not Hermitian matrices. This is obviously irrelevant as all the computational aspects are analogous for H_θ and $e^{-i\theta}H_\theta$.

3. A Hermitian Lanczos method for normal matrices. Assume $N \in \mathbb{C}^{n\times n}$ is normal and invertible and consider iteratively solving the linear system

$$Nx = b \tag{3.1}$$

for $b \in \mathbb{C}^n$. As described in the introduction, to achieve this with an optimal k-term recurrence there are two well-known alternatives. One is to consider the CGN method [11], that is, (3.1) is multiplied with N^* from the left. Another is to perform matrix vector multiplications with N only. In the latter case the length of the recurrence is determined by the theorem of Faber and Manteuffel [7]. See also [32]. Their result is extremely negative in the sense that the value of k is typically not only unacceptably large but also very unstable in the following sense.

Example 1. Assume $H \in \mathbb{C}^{n\times n}$ is Hermitian and invertible and consider iteratively solving the linear system $Hx = b$ with $b \in \mathbb{C}^n$. This can be done by using, for example, MINRES [11]. However, if H is perturbed slightly in a very simple manner, for instance, by setting $N = H + i\alpha H^2$, where $\alpha > 0$ is a small parameter, then N is not quite Hermitian anymore. Furthermore, the spectrum of N does not lie on a straight line but is slightly concave up. As a result, the 3-term recurrence of MINRES is no longer optimal.

In what follows we will derive a new optimal 3-term recurrence for solving (3.1). The arising method does not rely on using the normal equations and thereby does not lead to a squared condition number. Also, since the length of recurrence is constant for the elements of $\mathcal{N}$, the unstable behavior of the optimal recurrence length of Example 1 is not possible. To this end, in order to obtain an iterative method, we replace (2.10) with a minimization problem involving the vector b. Thus, analogously to the GMRES minimization problem, we consider

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(H)b - b|| \tag{3.2}$$

with $H = \frac{1}{2}(N + N^*)$. By the results of [12], the connection between this and the problem (2.10) is well understood.

Since N commutes with H, (3.2) is equal to

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||p_{j-1}(H)Nb - b||\,, \tag{3.3}$$

the value of which is readily obtained with Krylov subspace methods. More precisely, finding (3.3) is equivalent to approximating b from the Krylov subspace

$$\mathcal{K}_j(H; Nb) = \text{span}\{Nb, HNb, \dots, H^{j-1}Nb\}. \tag{3.4}$$

An inexpensive way to realize this is to execute the Hermitian Lanczos method with H and the starting vector $\hat{q}_0 = \frac{Nb}{\|Nb\|}$. The Hermitian Lanczos method is a classical algorithm similarity transforming a Hermitian matrix to a tridiagonal matrix. This is achieved by computing

$$T_j := \hat{Q}_j^* H \hat{Q}_j = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ 0 & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{j-1} & \beta_{j-1} \\ & & & \beta_{j-1} & \alpha_j \end{bmatrix}, \tag{3.5}$$

where $\hat{Q}_j \in \mathbb{C}^{n\times j}$ has orthonormal columns spanning the Krylov subspace (3.4). The elements of the matrix (3.5) can be computed with the 3-term recurrence

$$\beta_k \hat{q}_k = H\hat{q}_{k-1} - (H\hat{q}_{k-1}, \hat{q}_{k-1})\hat{q}_{k-1} - (H\hat{q}_{k-1}, \hat{q}_{k-2})\hat{q}_{k-2}, \tag{3.6}$$

where β_k equals the norm of the right-hand side of (3.6) and $\alpha_k = (H\hat{q}_{k-1}, \hat{q}_{k-1})$. For further details of the Hermitian Lanczos method, see, e.g., [25, 4].

To solve (3.3), we formally start the iteration with $\hat{q}_0 = \frac{Nb}{\|Nb\|} \in \mathbb{C}^n$ since

$$||p(H)Nb - b|| = ||Nb|| \left|\left| p(H)\frac{Nb}{\|Nb\|} - \frac{b}{\|Nb\|} \right|\right| \tag{3.7}$$

for every polynomial p. The Hermitian Lanczos iteration (3.6) would then proceed as

$$\hat{q}_1 = \frac{1}{\beta_1}\big(H\hat{q}_0 - (H\hat{q}_0, \hat{q}_0)\hat{q}_0\big), \tag{3.8}$$

$$\hat{q}_2 = \frac{1}{\beta_2}\big(H\hat{q}_1 - (H\hat{q}_1, \hat{q}_1)\hat{q}_1 - (H\hat{q}_1, \hat{q}_0)\hat{q}_0\big), \tag{3.9}$$

$$\vdots$$

so that at the jth step one computes

$$\hat{q}_{j-1} = \frac{1}{\beta_{j-1}}\big(H\hat{q}_{j-2} - (H\hat{q}_{j-2}, \hat{q}_{j-2})\hat{q}_{j-2} - (H\hat{q}_{j-2}, \hat{q}_{j-3})\hat{q}_{j-3}\big), \tag{3.10}$$

where the constants β_k are chosen such that $\|\hat{q}_k\| = 1$ for $0 \le k \le j-1$. Then the minimum (3.3) is realized with the polynomial p_{j-1} satisfying

$$p_{j-1}(H)Nb = \sum_{k=0}^{j-1} (b, \hat{q}_k)\hat{q}_k \tag{3.11}$$

as the vectors $\{\hat{q}_k\}_{k=0}^{j-1}$ are orthonormal. The problem is that this scheme does not yield a solution candidate x_{j-1} for solving the linear system (3.1). Instead, it yields the minimum value (3.3) so that the vector (3.11) should be multiplied by N^{-1} to get the approximation x_{j-1}. To avoid the inversion, the trick is that, since

$$x_{j-1} = \sum_{k=0}^{j-1} (b, \hat{q}_k)N^{-1}\hat{q}_k, \tag{3.12}$$

we do not actually compute (3.8), (3.9), and (3.10). Instead, we set $q_0 = \frac{b}{\|Nb\|}$ and

$$q_1 := N^{-1}\hat{q}_1 = \frac{1}{\beta_1}\big(Hq_0 - (H\hat{q}_0, \hat{q}_0)q_0\big), \tag{3.13}$$

$$q_2 := N^{-1}\hat{q}_2 = \frac{1}{\beta_2}\big(Hq_1 - (H\hat{q}_1, \hat{q}_1)q_1 - (H\hat{q}_1, \hat{q}_0)q_0\big), \tag{3.14}$$

$$\vdots$$

so that at the jth step we have

$$q_{j-1} := N^{-1}\hat{q}_{j-1} = \frac{1}{\beta_{j-1}}\big(Hq_{j-2} - (H\hat{q}_{j-2}, \hat{q}_{j-2})q_{j-2} - (H\hat{q}_{j-2}, \hat{q}_{j-3})q_{j-3}\big). \tag{3.15}$$

At each step the inner products only formally involve $\hat{q}_k$ as they can be computed by using the relation $Nq_k = \hat{q}_k$. Consequently, only vectors q_k need to be stored and we have a 3-term recurrence since (3.12) yields an updated approximation

$$x_{j-1} = \sum_{k=0}^{j-1}(b, Nq_k)q_k = x_{j-2} + (b, Nq_{j-1})q_{j-1} \tag{3.16}$$

at the jth step. Thus we have obtained the following basic algorithm.

ALGORITHM 1. "For solving the linear system $Nx = b$."
For $N = H + iK \in \mathcal{N}$ *and* $b \in \mathbb{C}^n$, *set* $H = \frac{1}{2}(N + N^*)$, $q_{-1} = 0$, $q_0 = \frac{b}{\|Nb\|}$, $x_0 = (b, Nq_0)q_0$, *and* $r_0 = b - Nx_0$.
for $k = 1$ *to* $j - 1$ *compute*
$q_k = Hq_{k-1} - (HNq_{k-1}, Nq_{k-1})q_{k-1} - (HNq_{k-1}, Nq_{k-2})q_{k-2}$
$q_k = \frac{q_k}{\|Nq_k\|}$
$\alpha_k = (r_{k-1}, Nq_k)$
$x_k = x_{k-1} + \alpha_k q_k$
$r_k = r_{k-1} - \alpha_k Nq_k$
end for

Remark 1. As opposed to the CGN method, we are solving the original linear system and not at any point the normal equations. To illustrate this, assume $N = p(H)$ for a polynomial p. Then the residual generated with Algorithm 1 can be bounded by considering the approximation problem

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Np_{j-1}(H)b - b|| \le \min_{p_{j-1}\in\mathcal{P}_{j-1}} \max_{\lambda\in\sigma(H)} |p(\lambda)p_{j-1}(\lambda) - 1|\,||b|| \tag{3.17}$$

on the spectrum of H (and not on the spectrum of N^*N). In particular, if N was Hermitian, i.e., $N = H$, then we would have the standard GMRES bound

$$\min_{p_{j-1}\in\mathcal{P}_{j-1}} ||Hp_{j-1}(H)b - b|| \le \min_{p_{j-1}\in\mathcal{P}_{j-1}} \max_{\lambda\in\sigma(H)} |\lambda p_{j-1}(\lambda) - 1|\,||b||\,. \tag{3.18}$$

Remark 2. In addition to the complexity of the classical Hermitian Lanczos iteration step (3.10), three additional matrix vector products Nq_{k-2}, Nq_{k-1}, and Nq_k are needed for computing q_k with Algorithm 1. The actual complexity will obviously depend on the way the algorithm is eventually implemented.

Remark 3. Due to the formal multiplication by the inverse of N, the vectors $\{q_k\}_{k=0}^{j-1}$ computed are not orthonormal, whereas the vectors $\{\hat{q}_k\}_{k=0}^{j-1} = \{Nq_k\}_{k=0}^{j-1}$ are. Numerical stability properties for the so-called A^*A-variant of GMRES have been considered in [27] by Rozložnik and Strakoš. This implementation of GMRES corresponds to Algorithm 1 and their analysis can be repeated with the method proposed. Of course, now one also needs to take into account the fact that Algorithm 1 relies on a 3-term recurrence.

For a Hermitian matrix, Algorithm 1 is thus equivalent to GMRES. The same is true for a rotation and translation of a Hermitian matrix as long as two angles are excluded.

PROPOSITION 3.1. *Assume $H \in \mathbb{C}^{n\times n}$ is Hermitian and $b \in \mathbb{C}^n$. Then, for $\theta \in [0, 2\pi)\backslash\{\frac{\pi}{2}, \frac{3\pi}{2}\}$ and $\lambda \in \mathbb{C}$, Algorithm 1 for solving $Nx = b$, with $N = e^{i\theta}H + \lambda I$, is equivalent to GMRES.*

Proof. With $\lambda = x + iy$ the Hermitian part of $e^{i\theta}H + \lambda I$ equals $\cos(\theta)H + xI$. Therefore, if $\theta \in [0, 2\pi)\backslash\{\frac{\pi}{2}, \frac{3\pi}{2}\}$, then for the Krylov subspaces we have

$$\mathcal{K}_j(\cos(\theta)H + xI; (e^{i\theta}H + \lambda I)b) = \mathcal{K}_j(e^{i\theta}H + \lambda I; (e^{i\theta}H + \lambda I)b)$$

for every $j \geq 1$. Thus, the claim follows from (3.4). ∎

It is straightforward to modify Algorithm 1 so that an initial guess x_{-1} can be used instead. Then b is replaced with the residual $r = b - Nx_{-1}$ and $q_0 = \frac{r}{\|Nr\|}$ is the starting vector for the iteration. The corresponding approximate solution is $x_{-1} + x_{j-1}$ at the jth step.

Because of the generic representation (2.11), a rotated Toeplitz decomposition of N can be employed by replacing H with H_θ in Algorithm 1. Combining this with using initial guesses gives rise to a scheme in which the rotation parameter is modified during the iteration. More precisely, the iteration is started by introducing a parameter $\theta \in [0, 2\pi)$ and an initial guess x_{-1}. Then Algorithm 1 is executed with $r = b - Nx_{-1}$ and H_θ in place of b and H, respectively. After, let us say, j steps, Algorithm 1 has produced x_{j-1} so that an approximate solution $x_{-1} + x_{j-1}$ is obtained. Then another rotation parameter θ is chosen and Algorithm 1 is restarted by using the vector $x_{-1} + x_{j-1}$ as an initial guess. An algorithm for this purpose is shown below.

ALGORITHM 2. "For solving the linear system $Nx = b$."
For $N = e^{-i\theta}H_\theta + ie^{-i\theta}K_\theta \in \mathcal{N}$ and an initial guess x_{-1}, set $r = b - Nx_{-1}$, $q_{-1} = 0$, $q_0 = \frac{r}{\|Nr\|}$, $x_0 = (r, Nq_0)q_0$, and $r_0 = r - Nx_0$.
for $k = 1$ to $j - 1$ compute
$q_k = H_\theta q_{k-1} - (H_\theta Nq_{k-1}, Nq_{k-1})q_{k-1} - (H_\theta Nq_{k-1}, Nq_{k-2})q_{k-2}$
$q_k = \frac{q_k}{\|Nq_k\|}$
$\alpha_k = (r_{k-1}, Nq_k)$
$x_k = x_{k-1} + \alpha_k q_k$
$r_k = r_{k-1} - \alpha_k Nq_k$
end for
Replace x_{-1} with $x_{-1} + x_{j-1}$
Restart with a new θ.

Remark 4. The nongeneric case, that is, N is not a polynomial in its Hermitian part, can be dealt with by employing restarts and rotations as suggested. To this end the norm of the updates $\alpha_k q_k$ can be monitored. If they remain under a threshold for a number of steps and the approximate solution is not sufficiently accurate, then a new rotation is introduced and the approximation computed so far is used as an initial guess.

Remark 5. In a nearly nongeneric case the convergence can slow down unless rotations and restarts are used. This is readily explained by the bound of Corollary 2.6. Namely, then the Hermitian part of N has clustered eigenvalues and the polynomials p_r and p_i of Corollary 2.6 may need to oscillate wildly over a short interval to approximate $\Re(\Lambda^{-1})$ and $\Im(\Lambda^{-1})$ well. A potential remedy to this is to restart Algorithm 2 frequently by using random rotations. By frequent restarting we mean that j is very small in Algorithm 2. According to our numerical experiments of section 5, this seems to average out the angle dependence of the method resulting in a performance comparable with GMRES. Recall that GMRES is a rotation invariant algorithm.

4. Dealing with nonnormal problems. A purpose of this paper, combined with the results of [18], is to demonstrate that the set of normal matrices is a homogeneous set in the sense that Hermitian matrices do not enjoy any particular algorithmic advantages among $\mathcal{N}$. Both solving linear systems and computing eigenvalue approximations can be accomplished with a 3-term recurrence for any of the elements of $\mathcal{N}$. Still, as a stratified manifold the set of normal matrices is not significantly larger than the set of Hermitian matrices $\mathcal{H}$. The (real) dimensions of $\mathcal{N} \subset \mathbb{C}^{n\times n}$ and $\mathcal{H} \subset \mathbb{C}^{n\times n}$ are $n^2 + n$ and n^2, respectively [18]. Thus, the question arises as to whether the extended 3-term recurrence relation can really be useful. To our mind $\mathcal{N}$ is larger than $\mathcal{H}$ in a critical way so that it has potential to be.

The first property that needs to be emphasized is that every square matrix can be represented as a product of two normal matrices. The most famous such factorization is, of course, the polar decomposition. Thus, in principle, *every* linear system can be solved via solving two consecutive linear systems involving normal matrices. We find this quite remarkable since only matrices with a real determinant are representable as a product of Hermitian matrices. Then the number of Hermitian matrices needed for this purpose is four at most [26].

A further distinction between $\mathcal{H}$ and $\mathcal{N}$ arises when splitting matrices. More precisely, assume one has a decomposition $A = N + F$ of $A \in \mathbb{C}^{n\times n}$ with a normal N and a small rank matrix F. If this is achievable with a sufficiently small rank F, then, by using inner-outer iterations, the system is solvable with modest storage requirements [21, 20]. In [20] it was demonstrated that every $A \in \mathbb{C}^{n\times n}$ possesses a representation $A = N+F$ with $N \in \mathcal{N}$ and F of rank at most $\lfloor \frac{n}{2} \rfloor$. If N is constrained to be Hermitian instead, then F with rank less than n cannot be found in general.

These two above-mentioned properties can be exploited directly or indirectly in, for instance, preconditioning. Using Hermitian matrices in preconditioning of nonnormal problems was initiated by Concus and Golub [3] and Widlund [34]. For more recent references see also [10]. The approach starts from splitting (a non-Hermitian) $A \in \mathbb{C}^{n\times n}$ as $A = H + iK$, where H and K are the Hermitian and skew-Hermitian parts of A, respectively, i.e., A is Toeplitz decomposed. The scheme proposed is then based on the assumption that the associated Hermitian (or skew-Hermitian) linear systems are readily solvable, i.e., not ill-conditioned. However, with an optimal 3-term recurrence for linear systems involving normal matrices, this approach can be generalized, as any normal matrix can then be used in preconditioning. Simple choices are polynomials in the Hermitian part of a rotated Toeplitz decomposition of A. In particular, there are sparse methods for computing these polynomials [18].

5. Numerical experiments. Next we consider numerical experiments for solving a linear system $Nx = b$ with a normal coefficient matrix. Problems considered are relatively small, as our main purpose is to illustrate how Algorithms 1 and 2 extend GMRES and how to use restarts with Algorithm 2. In particular, the examples are constructed so that the spectra differ from one another and therefore the chosen problems are somewhat artificial. However, clear variation in the spectra illustrates best the convergence properties of the method. The computations are performed with `matlab` [22] and we use its syntax to explain the numerical experiments. In all of the examples $N \in \mathbb{C}^{n\times n}$ is normal and, unless otherwise stated, $b \in \mathbb{C}^n$ is a random complex vector, that is, $b = \mathrm{rand}(n, 1) + i\mathrm{rand}(n, 1)$.

Example 2. We start with a very well understood example illustrating how Algorithm 1 extends GMRES for Hermitian matrices to normal matrices in a continuous way. We compare the convergence for a Hermitian positive definite

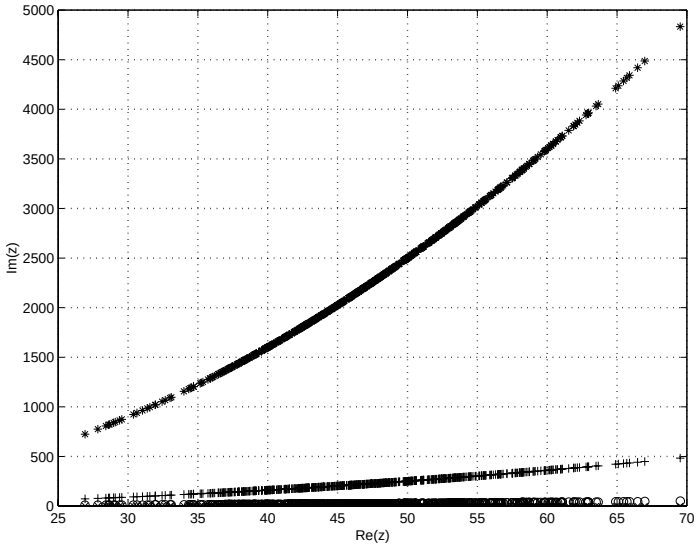


FIG. 1. *The eigenvalues of the matrices N_0, N_1, N_2, and N_3 of Example* 2 *are denoted by x, o, +, and ∗, respectively.*

$N_0 = H = \mathrm{diag}(8(6 + \mathrm{randn}(600, 1))) \in \mathbb{C}^{600\times 600}$ and for "slightly" bent N_0, that is, for $N_j = H + i\alpha_j H^2$, with small positive values for α_j. Besides $\alpha_0 = 0$, we set $\alpha_1 = 0.01$, $\alpha_2 = 0.1$, and $\alpha_3 = 1$. Note that for N_0 Algorithm 1 is MINRES. In Figure 1 we have depicted the eigenvalues of N_0, N_1, N_2, and N_3. The convergence of Algorithm 1 is nearly similar for these matrices even though they differ considerably in norm; see Figure 2, where we have plotted the relative residuals $\frac{\|r_k\|}{\|r_0\|} = \frac{\|b-Nx_k\|}{\|b\|}$.

For an illustration, we also compared Algorithm 1 with BiCGSTAB [33] and QMR [9] for solving $N_3 x = b$ using the implementations of `matlab`. The convergence behavior is plotted in Figure 3.

Example 3. This example illustrates the effect of rotations combined with using initial guesses as described in Remarks 4 and 5. We set $N = \mathrm{diag}([n1; n2; n3; n4])$, with $n1 = 3(5+\mathrm{randn}(100, 1))$, $n2 = 5(-7+\mathrm{randn}(100, 1))$, $n3 = 4(6+\mathrm{randn}(100, 1))i$, and $n4 = 2(-10 + \mathrm{randn}(100, 1))i$. Thus, $N \in \mathbb{C}^{400\times 400}$ and its spectrum lies on the union of the real and imaginary axes like a "cross"; see Figure 4. In particular, N is nongeneric as half of the eigenvalues are on a vertical line, that is, on the imaginary axis. This is a difficult problem for GMRES since the origin is almost symmetrically in the middle of the spectrum. We use the following strategies for choosing the rotations with Algorithm 2:

- **20-20** is such that we assume we have a rough idea of the location of the eigenvalues; that is, we know that the spectrum belongs to the union of the real and imaginary axes. This information is used while choosing rotations as follows. We take 20 steps with H and then rotate by $\frac{\pi}{2}$ and take 20 steps with $H_{\frac{\pi}{2}}$. This is then repeated.
- **8-8** is such that we take 8 steps with H and then rotate by $\frac{\pi}{2}$ and take 8 steps with $H_{\frac{\pi}{2}}$. This is then repeated.
- **Random-5** is such that we assume knowing nothing about the spectrum so that there is no reason to prefer any particular angle. We use Algorithm 2 by taking a random rotation and then perform 5 steps. This is then repeated.

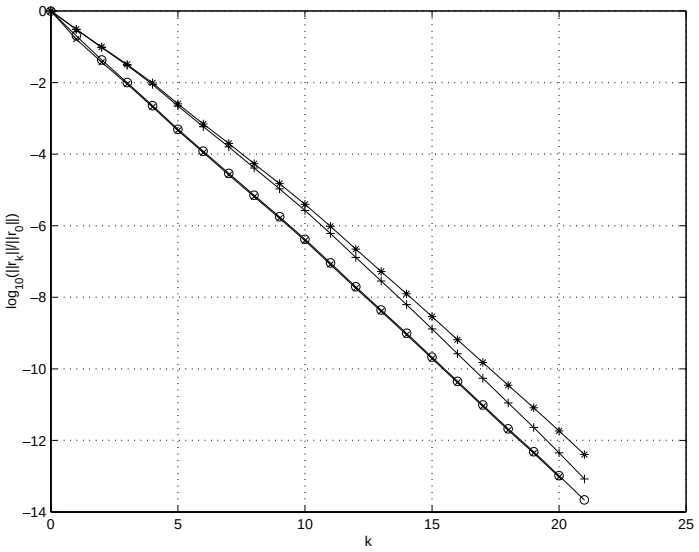


FIG. 2. *The convergence of the relative residuals on a* log10*-scale for Algorithm* 1 *for Example* 2. *The convergence is denoted by* $-x$, $-o$, $-+$, *and* $-*$ *for* N_0, N_1, N_2, *and* N_3, *respectively.*

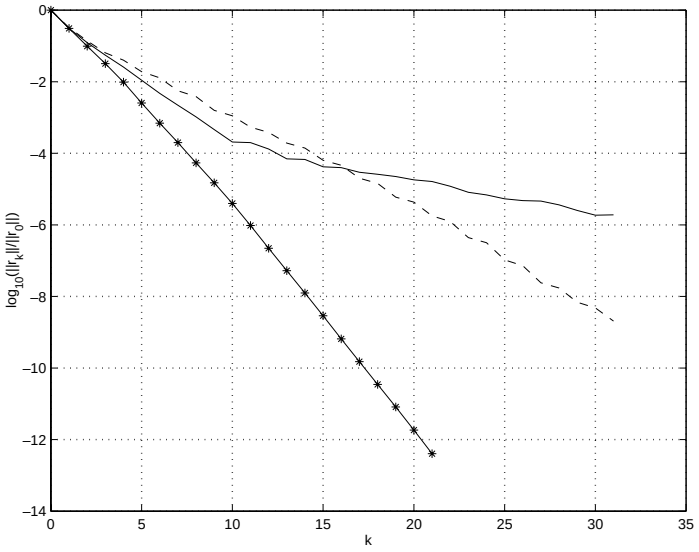


FIG. 3. *The convergence of the relative residuals for Algorithm* 1, *BiCGSTAB, and QMR on a* log10*-scale for solving* $N_3x = b$ *in Example* 2. *The convergence is denoted by* $-*$, $--$, *and the solid line, respectively.*

In Figure 5 we have plotted the relative residuals $\frac{\|r_k\|}{\|r_0\|} = \frac{\|b-Nx_k\|}{\|b\|}$ for GMRES and each of the rotation strategies described.

Let us try to explain the dependence of the convergence on the rotations for **20-20** and **8-8**. Starting with $\theta_0 = 0$ means that H_0 does not possess any information regarding the eigenvalues on the imaginary axis. Thus, then the method behaves almost like MINRES for the Hermitian matrix $\mathrm{diag}([n1; n2])$ and decreases the residual in the corresponding subspace of dimension 200. After a stagnation, the corresponding

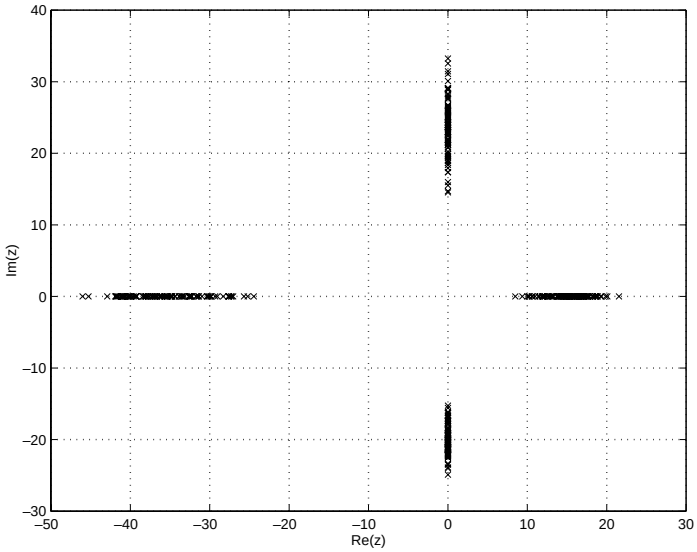


FIG. 4. *The eigenvalues of the matrix N in Example* 3.

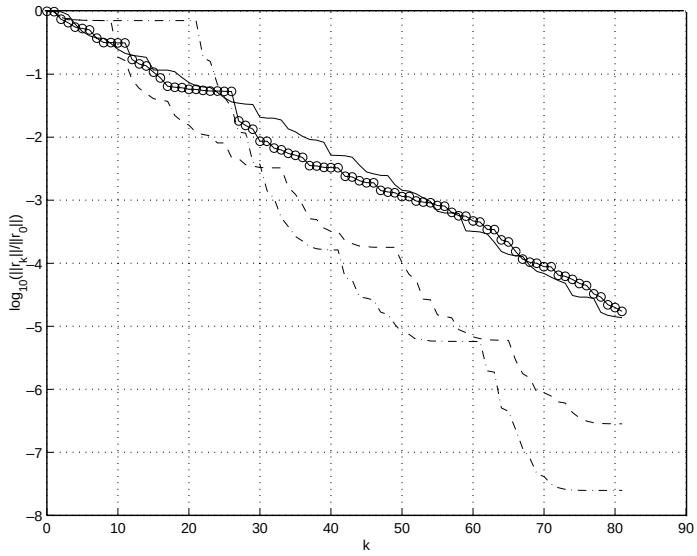


FIG. 5. *The convergence of the relative residuals on a* log10*-scale for Example* 3. *The solid line is GMRES and* $-\cdot$, $--$, *and* $-o$ *are* **20-20**, **8-8**, *and* **Random-5**, *respectively.*

projected problem onto that subspace is approximately solved. Performing a rotation with $\theta = \frac{\pi}{2}$ using the approximate solution generated as an initial guess, the method behaves then almost like MINRES for the Hermitian matrix $\mathrm{diag}([n3/i; n4/i])$ and decreases the residual in the corresponding subspace of dimension 200.

Note that the strategies **20-20** and **8-8** win and only **Random-5** decreases the residual like GMRES.

Example 4. We let $N \in \mathbb{C}^{500\times 500}$ be the notorious unitary shift and choose $b \in \mathbb{C}^{500}$ to be a standard unit basis vector. Thus, the origin is surrounded,

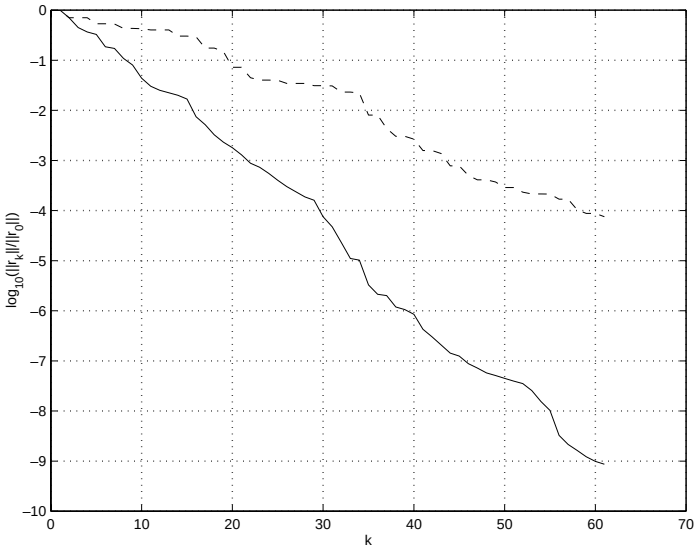


FIG. 6. *The convergence of the relative residuals on a* log10*-scale for Example* 4. *The solid line is* **Random-1** *and* −− *is* **Random-3**.

uniformly on the unit circle, by the eigenvalues of N. Although being an easy problem for CGN [2, 23], this is a very difficult problem for GMRES (of course GMRES would be a ridiculous choice for solving this problem as opposed to employing the normal equations). More precisely, the convergence of GMRES for this problem is catastrophic as no progress is made before the 500th step. Note that N is again nongeneric so that we execute Algorithm 2. As there are no angles to prefer, we use very frequent restarting with random rotations. That is, we use the **Random-1** and **Random-3** strategies explained in Example 3. The convergence of the relative residuals $\frac{\|r_k\|}{\|r_0\|} = \frac{\|b-Nx_k\|}{\|b\|}$ is depicted in Figure 6. Very frequent restarting seems to be a good choice for this problem.

Example 5. We set $N = I + 0.1\text{diag}(\text{randn}(100,1) + i\text{randn}(100,1)) \in \mathbb{C}^{100\times 100}$. As opposed to Example 4, this matrix is constructed to be particularly favorable to GMRES since the spectrum of N is concentrated around 1. The spectrum is otherwise very unstructured so that there are no angles to prefer. In particular, the polynomial approximation problem on the right-hand side of the bound (3.17) is difficult as relatively high degree polynomials are needed for a close approximation so that Algorithm 1 actually performs poorly. Therefore we execute Algorithm 2 with frequent restarting by using again the **Random-1** and **Random-3** strategies as explained in Example 3. For the behavior of the residuals, see Figure 7.

Remark 6. Although these experiments are rather preliminary, we find the following questions particularly interesting. For a given linear system, is there a strategy for choosing the rotations such that Algorithm 2 would perform at least as well as GMRES. And if so, is this achievable by using random rotations? What should be the restarting frequency then?

6. Conclusions. We have derived a Hermitian Lanczos method for normal matrices. The algorithm is realizable with an optimal 3-term recurrence without resorting to the normal equations. The algorithm reduces to MINRES whenever N is Hermitian while otherwise the speed of convergence is bounded by a polynomial approximation

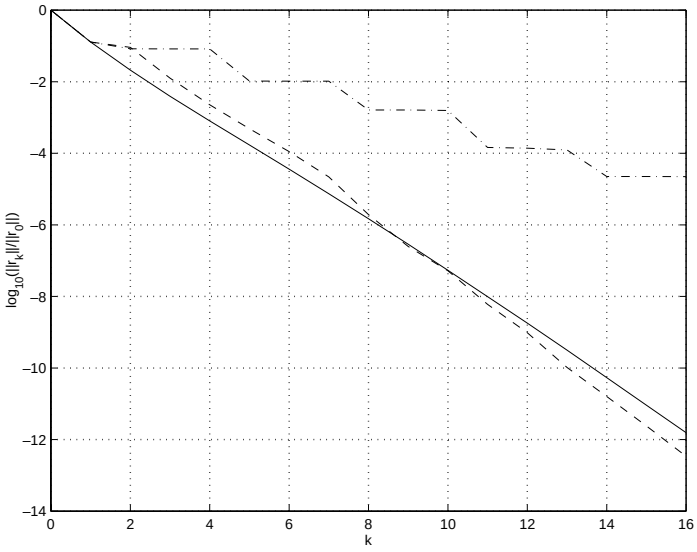


FIG. 7. *The convergence of the relative residuals on a* log10*-scale for Example* 5. *The solid line is GMRES,* −− *is* **Random-1**, *and* −· *is* **Random-3**.

problem on the spectrum of the Hermitian part of N. Rotations and restarts can be used to improve the convergence in case the approximation problem cannot be solved sufficiently accurately with low degree polynomials. If there is information about the spectrum, then rotations should be chosen so that various parts of the spectrum are well approximated by low degree polynomials. In general we do not know how to choose the rotations and restarting frequency optimally. However, very frequent restarting combined with random rotations seems to be a good choice, averaging out the effect of the rotation angles. This intuitive interpretation partly explains why the method seems to perform at least as well as GMRES which is a rotation invariant algorithm.

Acknowledgments. We would like to thank the referees for their exceptionally detailed comments that considerably improved earlier versions of the paper. We are grateful to Dr. Rasmus Munk Larsen for enlightening discussions about how to implement Algorithm 1 stably. His suggestions are taken into account in the present versions of the algorithms.

REFERENCES

[1] M. ARIOLI, V. PTÁK, AND Z. STRAKOŠ, *Krylov sequences of maximal length and convergence of GMRES*, BIT, 38 (1998), pp. 636–643.

[2] P. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

[3] P. CONCUS AND G.H. GOLUB, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Sciences and Engineering, Lecture Notes in Econ. and Math. Systems 134, R. Glowinski and J.L. Lions, eds., Springer-Verlag, Berlin, New York, 1976, pp. 56–65.

[4] J. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[5] L. ELSNER AND KH.D. IKRAMOV, *Normal matrices: An update*, Linear Algebra Appl., 285 (1998), pp. 291–303.

[6] L. Elsner and M.H.C. Paardekooper, *On measures of nonnormality of matrices*, Linear Algebra Appl., 92 (1987), pp. 107–124.
[7] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[8] V. Faber and T. Manteuffel, *Orthogonal error methods,* SIAM J. Numer. Anal., 24 (1987), pp. 170–187.
[9] R.W. Freund and N.M. Nachtigal, *QMR: A quasi-minimal residual method for solving non-Hermitian linear systems,* Numer. Math., 60 (1991), pp. 315–339.
[10] G.H. Golub and D. Vanderstraeten, *On the preconditioning of matrices with dominant skew-symmetric component*, Numer. Algorithms, 25 (2000), pp. 223–239.
[11] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
[12] A. Greenbaum and L. Gurvits, *Max-min properties of matrix factor norms*, SIAM J. Sci. Comp., 15 (1994), pp. 348–358.
[13] A. Greenbaum and L.N. Trefethen, *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems*, SIAM J. Sci. Comput., 15 (1994), pp. 359–368.
[14] A. Greenbaum and Z. Strakoš, *Matrices that generate the same Krylov residual spaces*, in Recent Advances in Iterative Methods, IMA Vol. Math. Appl. 60, G. Golub, A. Greenbaum, and M. Luskin, eds., Springer-Verlag, New York, 1994, pp. 95–118.
[15] R. Grone, C.R. Johnson, E.M. Sa, and H. Wolkowicz, *Normal matrices,* Linear Algebra Appl., 87 (1987), pp. 213–225.
[16] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[17] M. Huhtanen, *Ideal GMRES Can Be Bounded from Below by Three Factors*, Math. Report A412, Helsinki University of Technology, Espoo, Finland, 1999.
[18] M. Huhtanen, *A stratification of the set of normal matrices*, SIAM J. Matrix. Anal. Appl., 23 (2001), pp. 349–367.
[19] M. Huhtanen, *Pole Assignment Problems for Error Bounds for GMRES*, Math. Report A418, Helsinki University of Technology, Espoo, Finland, 1999.
[20] M. Huhtanen, *A matrix nearness problem related to iterative methods*, SIAM J. Numer. Anal., 39 (2001), pp. 407–422.
[21] M. Huhtanen and O. Nevanlinna, *Minimal decompositions and iterative methods*, Numer. Math., 86 (2000), pp. 257–281.
[22] Mathworks, Inc., *Matlab*, www.mathworks.com/products/matlab.
[23] N.M. Nachtigal, S. Reddy, and L.N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.
[24] O. Nevanlinna, *Convergence of Iterations for Linear Equations*, Lectures in Mathematics ETH Zürich, Birkhäuser-Verlag, Basel, 1993.
[25] B.N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[26] H. Radjavi, *Products of Hermitian matrices and symmetries*, Proc. Amer. Math. Soc., 21 (1969), pp. 369–372; *Errata*, Proc. Amer. Math. Soc., 26 (1970), p. 701.
[27] M. Rozložnik and Z. Strakoš, *Variants of the residual minimizing Krylov space methods,* in Proceedings of the XIth Summer School on Software and Algorithms of Numerical Mathematics, I. Marek, ed., Zelezna Ruda, Western Bohemia, 1995, pp. 208–225. Available online at http://deimos.zcu.cz/konference/SANM/SANM.html#SANM'95.
[28] A. Ruhe, *Closest normal matrix finally found!,* BIT, 27 (1987), pp. 585–598.
[29] Y. Saad and M.H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comp., 7 (1986), pp. 856–869.
[30] M. Saunders, H. Simon, and E. Yip, *Two conjugate-gradient-type methods for unsymmetric linear equations*, SIAM J. Numer. Anal., 25 (1988), pp. 927–940.
[31] Signum Newsletter, 16 (1981), p. 7.
[32] V.V. Voevodin, *The question of non-self-adjoint extensions of the conjugate gradients methods is closed*, USSR Comput. Maths. Phys., 23 (1983), pp. 143–144.
[33] H. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Comput., 13 (1992), pp. 631–644.
[34] O. Widlund, *A Lanczos method for a class of nonsymmetric linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–812.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1109–1119
© 2002 Society for Industrial and Applied Mathematics

ON A QUESTION CONCERNING CONDITION NUMBERS FOR MARKOV CHAINS*

S. KIRKLAND†

Abstract. Let S be an irreducible stochastic matrix of order n with left stationary vector π^T, and let $S_{(i)}$ denote the principal submatrix of S formed by deleting the ith row and column. We prove that $\max_{1\leq i\leq n}\pi_i||(I-S_{(i)})^{-1}||_\infty \leq \min_{1\leq j\leq n}||(I-S_{(j)})^{-1}||_\infty$, thus answering a question posed by Cho and Meyer. We provide an attainable lower bound on $\max_{1\leq i\leq n}\pi_i||(I-S_{(i)})^{-1}||_\infty$, and discuss the case that equality holds in that bound.

Key words. stochastic matrix, Markov chain, stationary vector, condition number

AMS subject classifications. 15A51, 15A18, 65F35, 60J10

PII. S0895479801390947

1. Introduction.

Suppose that S is an irreducible $n\times n$ stochastic matrix, so that it can be thought of as the transition matrix for a Markov chain. One of the principal quantities of interest associated with the Markov chain is the stationary vector for S, i.e., the entrywise positive vector π^T such that $\pi^T S=\pi^T$ and $\pi^T 1=1$, where 1 denotes the all-ones vector.

A number of papers consider stability problems of the following type: Perturb S to produce another irreducible stochastic matrix $\tilde{S}\equiv S-E$ with corresponding stationary vector $\tilde{\pi}^T$, and find a *condition number* κ so that for some suitable norm, $||\pi^T-\tilde{\pi}^T||\leq\kappa||E||$. A recent paper by Cho and Meyer [5] surveys several of these condition numbers and provides comparisons between them.

One of the key condition numbers discussed in [5] is given (adopting the notation of [5]) by

$$\kappa_3\equiv\frac{1}{2}\max_{1\leq i\leq n}\pi_i||(I-S_{(i)})^{-1}||_\infty,$$

where $S_{(i)}$ denotes the principal submatrix of S formed by deleting its ith row and column and where $||\bullet||_\infty$ is the maximum absolute row sum norm. Of the eight condition numbers considered in [5], κ_3 is shown to be the minimum among seven of them, while no comparison is made between κ_3 and the condition number

$$\kappa_7\equiv\frac{1}{2}\min_{1\leq j\leq n}||(I-S_{(j)})^{-1}||_\infty.$$

Cho and Meyer then go on to pose the question of whether $\kappa_3\leq\kappa_7$ for each irreducible stochastic S; in the case that S is rank 1, an affirmative answer is provided in [5]. In this paper, we prove that $\kappa_3\leq\kappa_7\leq n\kappa_3$ for any irreducible stochastic matrix S of order n. We also provide lower bounds on $\pi_i||(I-S_{(i)})^{-1}||_\infty$ for any i, and on $\max_{1\leq i\leq n}\pi_i||(I-S_{(i)})^{-1}||_\infty$; finally, we discuss the cases of equality in both bounds.

*Received by the editors June 18, 2001; accepted for publication (in revised form) by I. C. F. Ipsen October 23, 2001; published electronically April 10, 2002. This research was supported in part by an NSERC Research grant.

http://www.siam.org/journals/simax/23-4/39094.html

†Department of Mathematics and Statistics, University of Regina, Regina, SK, S4S 0A2, Canada (kirkland@math.uregina.ca).

In establishing our results, we will assume basic knowledge of Markov chains and the theory of nonnegative matrices. We refer the reader to [2], [8], and [9] for the necessary background.

2. Main results. Throughout the rest of the paper, S will denote an irreducible stochastic matrix of order n, and π^T will denote its left stationary vector. We begin with an expression for a single entry in π^T.

LEMMA 2.1. *Suppose that we have an irreducible stochastic matrix S of order n, partitioned as*

$$S = \left[\begin{array}{c|c} A & y \\ \hline x^T & 1 - x^T 1 \end{array}\right],$$

with left stationary vector π^T partitioned conformally as $\pi^T = [\hat{\pi}^T | \pi_n]$. Then we have $\pi_n = 1/(1 + x^T (I - A)^{-1} 1)$.

Proof. Since $\hat{\pi}^T A + \pi_n x^T = \hat{\pi}^T$, we find that $\hat{\pi}^T = \pi_n x^T (I - A)^{-1}$. The result now follows from the observation that $1 - \pi_n = \hat{\pi}^T 1 = \pi_n x^T (I - A)^{-1} 1$. ∎

The following result will allow us to derive an expression for $||(I - S_{(i)})^{-1}||_\infty$ for an irreducible stochastic matrix S. We note that the expression for $(I - S_{(n)})^{-1}$ below can also be deduced from standard results on the Schur complement and on the inverse of a rank 1 perturbation (see [6]).

LEMMA 2.2. *Suppose that we have an $n \times n$ irreducible stochastic matrix S, partitioned as*

$$S = \left[\begin{array}{c|cc} T & y_1 & y_2 \\ \hline x_1^T & a_1 & b_1 \\ x_2^T & b_2 & a_2 \end{array}\right].$$

Let $\gamma_1 = 1/(1 - a_1 - x_1^T (I - T)^{-1} y_1)$. Then

$$(I - S_{(n)})^{-1} = \left[\begin{array}{c|c} (I - T)^{-1} + \gamma_1 (I - T)^{-1} y_1 x_1^T (I - T)^{-1} & \gamma_1 (I - T)^{-1} y_1 \\ \hline \gamma_1 x_1^T (I - T)^{-1} & \gamma_1 \end{array}\right].$$

In particular,

$$(I - S_{(n)})^{-1} 1 = \left[\begin{array}{c} (I - T)^{-1} 1 + \gamma_1 (x_1^T (I - T)^{-1} 1 + 1)(I - T)^{-1} y_1 \\ \hline \gamma_1 (x_1^T (I - T)^{-1} 1 + 1) \end{array}\right].$$

Proof. We have

$$I - S_{(n)} = \left[\begin{array}{c|c} I - T & -y_1 \\ \hline -x_1^T & 1 - a_1 \end{array}\right],$$

and the results follow from direct computations. ∎

Remark 2.3. Using the notation of Lemma 2.2, suppose that S has left stationary vector π^T, and let $\gamma_2 = 1/(1 - a_2 - x_2^T (I - T)^{-1} y_2)$. By Lemma 2.1, we have

$$\pi_n = \frac{1}{1 + [x_2^T | b_2](I - S_{(n)})^{-1} 1}.$$

Since $T1 + y_1 + y_2 = 1$, we have $(I - T)^{-1} y_2 = 1 - (I - T)^{-1} y_1$. Using this fact and

the fact that $x_2^T 1 + b_2 = 1 - a_2$, we find from Lemma 2.2 that

$$\pi_n = \frac{1}{\gamma_1(x_1^T(I-T)^{-1}1+1)(1-a_2-x_2^T(I-T)^{-1}y_2)+x_2^T(I-T)^{-1}1+1}$$
$$= \frac{\gamma_2}{\gamma_1(x_1^T(I-T)^{-1}1+1)+\gamma_2(x_2^T(I-T)^{-1}1+1)}.$$

Here is the key result.

THEOREM 2.4. *Suppose that S is an $n \times n$ irreducible stochastic matrix, partitioned as in Lemma 2.2 and having left stationary vector π^T. Then $\pi_n||(I-S_{(n)})^{-1}||_\infty \le ||(I-S_{(n-1)})^{-1}||_\infty$. Further, equality holds if and only if $x_2 = 0$, $||(I-T)^{-1}||_\infty = 1/(1-a_2)$, and $(I-T)^{-1}y_2 \le 1 - (1-a_2)(I-T)^{-1}1$.*

Proof. Let $c_i = x_i^T(I-T)^{-1}1 + 1$ for $i = 1, 2$, so that $\pi_n = \frac{\gamma_2}{\gamma_1 c_1 + \gamma_2 c_2}$. Note that since $I - T$ is a nonsingular M-matrix, $(I-T)^{-1}$ is entrywise nonnegative, so that in particular, $c_1, c_2 \ge 1$. By Lemma 2.2, we have

$$(I - S_{(n)})^{-1}1 = \left[\frac{(I-T)^{-1}1 + \gamma_1 c_1 (I-T)^{-1}y_1}{\gamma_1 c_1}\right]$$

and

$$(I - S_{(n-1)})^{-1}1 = \left[\frac{(I-T)^{-1}1 + \gamma_2 c_2 (I-T)^{-1}y_2}{\gamma_2 c_2}\right].$$

If $||(I - S_{(n)})^{-1}||_\infty = \gamma_1 c_1$, then note that

$$\pi_n||(I-S_{(n)})^{-1}||_\infty = \frac{\gamma_1\gamma_2 c_1}{\gamma_1 c_1 + \gamma_2 c_2} = \gamma_2\left(\frac{\gamma_1 c_1}{\gamma_1 c_1 + \gamma_2 c_2}\right) < \gamma_2 \le \gamma_2 c_2 \le ||(I-S_{(n-1)})^{-1}||_\infty.$$

Henceforth, we will assume that $||(I - S_{(n)})^{-1}||_\infty > \gamma_1 c_1$, so that in fact

$$||(I - S_{(n)})^{-1}||_\infty = \max_{1\le i\le n-2} e_i^T((I-T)^{-1}1 + \gamma_1 c_1 (I-T)^{-1}y_1).$$

Suppose first that $||(I-T)^{-1}||_\infty > \gamma_2$. Then necessarily we have

$$\gamma_1\gamma_2 c_1 + \gamma_2||(I-T)^{-1}||_\infty < \gamma_1 c_1||(I-T)^{-1}||_\infty + \gamma_2 c_2||(I-T)^{-1}||_\infty,$$

so that

$$\frac{\gamma_2(||(I-T)^{-1}||_\infty + \gamma_1 c_1)}{\gamma_1 c_1 + \gamma_2 c_2} < ||(I-T)^{-1}||_\infty.$$

Since $(I-T)^{-1}y_1 + (I-T)^{-1}y_2 = 1$, we find that

$$\pi_n||(I - S_{(n)})^{-1}||_\infty = \frac{\gamma_2}{\gamma_1 c_1 + \gamma_2 c_2} \max_{1\le i\le n-2} e_i^T((I-T)^{-1}1 + \gamma_1 c_1 (I-T)^{-1}y_1)$$

$$= \frac{\gamma_2}{\gamma_1 c_1 + \gamma_2 c_2} \max_{1\le i\le n-2} e_i^T((I-T)^{-1}1 + \gamma_1 c_1 1 - \gamma_1 c_1 (I-T)^{-1}y_2)$$

$$\le \frac{\gamma_2}{\gamma_1 c_1 + \gamma_2 c_2}(||(I-T)^{-1}||_\infty + \gamma_1 c_1).$$

Further, $||(I-T)^{-1}||_\infty \leq ||(I-\ S_{(n-1)})^{-1}||_\infty$, so we find that

$$\pi_n||(I-\ S_{(n)})^{-1}||_\infty \leq \frac{\gamma_2}{\gamma_1 c_1+\gamma_2 c_2}(||(I-T)^{-1}||_\infty+\gamma_1 c_1) < ||(I-T)^{-1}||_\infty$$

$$\leq ||(I-\ S_{(n-1)})^{-1}||_\infty.$$

Now we suppose that $||(I-T)^{-1}||_\infty \leq \gamma_2$. Then certainly we have $(I-T)^{-1}\mathbf{1} \leq \gamma_2\mathbf{1}$. Since $c_2 \geq 1$, we thus find that

$$(I-T)^{-1}\mathbf{1} \leq \gamma_2\mathbf{1} \leq (\gamma_1 c_1(c_2-1)+\gamma_2 c_2^2)\mathbf{1}+\gamma_1 c_1(I-T)^{-1}y_2,$$

which is equivalent to

$$(I-T)^{-1}\mathbf{1}+\gamma_1 c_1\mathbf{1}-\gamma_1 c_1(I-T)^{-1}y_2 \leq c_2(\gamma_1 c_1+\gamma_2 c_2)\mathbf{1}.$$

But this last yields

$$\pi_n((I-T)^{-1}\mathbf{1}+\gamma_1 c_1(I-T)^{-1}y_1) = \pi_n((I-T)^{-1}\mathbf{1}+\gamma_1 c_1\mathbf{1}-\gamma_1 c_1(I-T)^{-1}y_2) \leq \gamma_2 c_2\mathbf{1},$$

so that $\pi_n||(I-\ S_{(n)})^{-1}||_\infty \leq \gamma_2 c_2 \leq ||(I-\ S_{(n-1)})^{-1}||_\infty$. Consequently, in all cases, we find that $\pi_n||(I-\ S_{(n)})^{-1}||_\infty \leq ||(I-\ S_{(n-1)})^{-1}||_\infty$.

Next we consider the case of equality. If $\pi_n||(I-S_{(n)})^{-1}||_\infty = ||(I-S_{(n-1)})^{-1}||_\infty$, then we find from the argument above that necessarily

$$||(I-\ S_{(n)})^{-1}||_\infty = \max_{1\leq i\leq n-2} e_i^T((I\ -\ T)^{-1}\mathbf{1}\ +\ \gamma_1 c_1(I\ -\ T)^{-1}\ y_1)$$

and $||(I-T)^{-1}||_\infty \leq \gamma_2$. Note that if $||(I-T)^{-1}||_\infty < \gamma_2$, then as above,

$$(I-T)^{-1}\mathbf{1} < \gamma_2\mathbf{1} \leq (\gamma_1 c_1(c_2-1)+\gamma_2 c_2^2)\mathbf{1}+\gamma_1 c_1(I-T)^{-1}y_2;$$

similarly, if $c_2 > 1$, then

$$(I-T)^{-1}\mathbf{1} \leq \gamma_2\mathbf{1} < (\gamma_1 c_1(c_2-1)+\gamma_2 c_2^2)\mathbf{1}+\gamma_1 c_1(I-T)^{-1}y_2.$$

In either case, we find that

$$(I-T)^{-1}\mathbf{1}+\gamma_1 c_1\mathbf{1}-\gamma_1 c_1(I-T)^{-1}y_2 < c_2(\gamma_1 c_1+\gamma_2 c_2)\mathbf{1},$$

so that

$$\pi_n((I-T)^{-1}\mathbf{1}+\gamma_1 c_1(I-T)^{-1}y_1) = \pi_n((I-T)^{-1}\mathbf{1}+\gamma_1 c_1\mathbf{1}-\gamma_1 c_1(I-T)^{-1}y_2) < \gamma_2 c_2\mathbf{1}.$$

But then $\pi_n||(I-\ S_{(n)})^{-1}||_\infty < \gamma_2 c_2 \leq ||(I-\ S_{(n-1)})^{-1}||_\infty$.

Consequently, if $\pi_n||(I-\ S_{(n)})^{-1}||_\infty = ||(I-\ S_{(n-1)})^{-1}||_\infty$, we must have $||(I-\ T)^{-1}||_\infty\ = \gamma_2, c_2 = 1$, and $||(I-\ S_{(n-1)})^{-1}||_\infty = \gamma_2$. The condition $c_2 = 1$ is equivalent to $x_2 = 0$, from which it follows that $\gamma_2 = 1/(1-a_2)$. Since $x_2 = 0$, we have

$$(I-S_{(n-1)})^{-1} = \left[\begin{array}{c|c} (I-T)^{-1} & \gamma_2(I-T)^{-1}y_2 \\ \hline 0^T & 1/(1-a_2) \end{array}\right],$$

so that the condition $||(I-T)^{-1}||_\infty = 1/(1-a_2)$ implies that $(I-T)^{-1}y_2 \leq \mathbf{1}\ -\ (1\ -\ a_2)(I-T)^{-1}\mathbf{1}$.

Finally, suppose that $x_2 = 0$, $||(I-T)^{-1}||_\infty = 1/(1-a_2)$, and $(I-T)^{-1}y_2 \leq \mathit{1} - (1-a_2)(I-T)^{-1}\mathit{1}$. Then from Lemma 2.2 it follows that $||(I - S_{(n-1)})^{-1}||_\infty = \gamma_2 = 1/(1-a_2)$. Further, since $(I-T)^{-1}y_2 \leq \mathit{1} - (1-a_2)(I-T)^{-1}\mathit{1}$, we see that for each entry of $(I-T)^{-1}\mathit{1}$ of maximum size (i.e., $1/(1-a_2)$), the corresponding entry of $(I-T)^{-1}y_2$ must be 0. Consequently, the maximum entry of $(I-T)^{-1}\mathit{1} + \gamma_1 c_1[\mathit{1} - (I-T)^{-1}y_2]$ is at least $\gamma_2 + \gamma_1 c_1$, so that $||(I - S_{(n)})^{-1}||_\infty \geq \gamma_2 + \gamma_1 c_1$. It now follows readily that $\pi_n||(I - S_{(n)})^{-1}||_\infty \geq \gamma_2 = ||(I - S_{(n-1)})^{-1}||_\infty$ and hence that $\pi_n||(I - S_{(n)})^{-1}||_\infty = ||(I - S_{(n-1)})^{-1}||_\infty$. ∎

Remark 2.5. We note that if equality is to hold in Theorem 2.4, then in the case that T is irreducible, we must have $y_2 = 0$, otherwise $(I-T)^{-1}y_2 > 0$. A similar argument applies if T is reducible, in which case we find that if $e_i^T(I-T)^{-1}\mathit{1} = 1/(1-a_2)$, and if indices i and j are in the same irreducible component of T, then the jth entry of y_2 must be 0.

Applying Theorem 2.4, we now answer the question posed in [5].

COROLLARY 2.6. *Suppose that S is an $n \times n$ irreducible stochastic matrix with left stationary vector π^T. Then*

$$\max_{1\leq i\leq n} \pi_i||(I - S_{(i)})^{-1}||_\infty \leq \min_{1\leq j\leq n} ||(I - S_{(j)})^{-1}||_\infty.$$

In particular, $\kappa_3 \leq \kappa_7$.

Proof. Evidently we have $\pi_i||(I - S_{(i)})^{-1}||_\infty < ||(I - S_{(i)})^{-1}||_\infty$ for $1 \leq i \leq n$. Suppose we have indices i, j with $1 \leq i \neq j \leq n$, and if necessary, apply a permutation similarity to S which sends i to n and j to $n-1$. From Theorem 2.4 we find that $\pi_i||(I - S_{(i)})^{-1}||_\infty \leq ||(I - S_{(j)})^{-1}||_\infty$, and the result follows. ∎

The following example shows that equality can hold in Corollary 2.6.

Example 2.7. Let x be a positive vector such that $x^T\mathit{1} = 1$, and let

$$S = \left[\begin{array}{c|c} 0 & \mathit{1} \\ \hline x^T & 0 \end{array}\right].$$

Then $\pi^T = [(1/2)x^T|1/2]$ and $||(I - S_{(n)})^{-1}||_\infty = 1$. For each $1 \leq i \leq n-1$, we have

$$(I - S_{(i)})^{-1} = \left[\begin{array}{c|c} I + \frac{1}{x_i}\mathit{1}\hat{x}^T & \frac{1}{x_i}\mathit{1} \\ \hline \frac{1}{x_i}\hat{x}^T & \frac{1}{x_i} \end{array}\right],$$

where $\hat{x}$ is formed from x by deleting the ith entry. It follows that $||(I - S_{(i)})^{-1}||_\infty = 2/x_i$, so that for each $1 \leq i \leq n-1$,

$$\pi_i||(I - S_{(i)})^{-1}||_\infty = \frac{x_i}{2}\frac{2}{x_i} = 1 = ||(I - S_{(n)})^{-1}||_\infty.$$

Having shown that $\kappa_3 \leq \kappa_7$, the next result discusses how far apart these quantities can be.

THEOREM 2.8. *Suppose that S is an $n \times n$ irreducible stochastic matrix with left stationary vector π^T. Then*

$$n \max_{1\leq i\leq n} \pi_i||(I - S_{(i)})^{-1}||_\infty \geq \min_{1\leq j\leq n} ||(I - S_{(j)})^{-1}||_\infty.$$

In particular, $n\kappa_3 \geq \kappa_7$. Equality holds if and only if S is a doubly stochastic matrix such that $||(I - S_{(i)})^{-1}||_\infty = ||(I - S_{(j)})^{-1}||_\infty$ for $i, j = 1, \ldots, n$.

Proof. We have

$$\max_{1\le i\le n}\pi_i||(I-\ S_{(i)})^{-1}||_\infty \ge\ \max_{1\le i\le n}\left\{\pi_i\left(\min_{1\le j\le n}||(I-\ S_{(j)})^{-1}||_\infty\right)\right\}$$

$$=\left(\max_{1\le i\le n}\pi_i\right)\left(\min_{1\le j\le n}||(I-\ S_{(j)})^{-1}||_\infty\right)\ge\frac{1}{n}\left(\min_{1\le j\le n}||(I-\ S_{(j)})^{-1}||_\infty\right),$$

yielding the desired inequality. Note that if equality holds, then necessarily $\max_{1\le i\le n}\pi_i = \frac{1}{n}$, from which it follows that each π_j is $\frac{1}{n}$, so that S is doubly stochastic. It is now readily deduced that if equality is to hold, we must also have $||(I-\ S_{(i)})^{-1}||_\infty = ||(I-\ S_{(j)})^{-1}||_\infty$ for $i,j = 1,\dots,n$. Conversely, if S is doubly stochastic and $||(I-S_{(i)})^{-1}||_\infty = ||(I-S_{(j)})^{-1}||_\infty$ for $i,j = 1,\dots,n$, then certainly $n\kappa_3 = \kappa_7$. □

Remark 2.9. Theorem 2.8 shows that κ_7 can be significantly larger than κ_3. For example, if S is an $n\times n$ circulant matrix, then it is readily seen that $\kappa_7 = n\kappa_3$, so that for large n, κ_3 is a much tighter bound. For a general transition matrix S, the derivation of the inequality in Theorem 2.8 indicates that if the entries of the stationary vector do not vary too much, and if the quantities $||(I-\ S_{(i)})^{-1}||_\infty, i = 1,\dots,n$, are reasonably close together, then for large values of n, κ_7 will be significantly larger than κ_3.

Next, we give a lower bound on $\pi_j||(I-S_{(j)})^{-1}||_\infty$.

THEOREM 2.10. *Let S be an $n\times n$ stochastic matrix with left stationary vector π^T. For each $j = 1,\dots,n$ we have $\pi_j||(I-S_{(j)})^{-1}||_\infty \ge 1/2$. Equality holds for some j_0 if and only if S is permutationally similar to*

$$R=\left[\begin{array}{c|c} 0 & 1 \\ \hline x^T & 0 \end{array}\right]$$

(where the index j_0 corresponds to n in R), for some positive vector x such that $x^T 1 = 1$.

Proof. By applying a permutation similarity if necessary, we assume that $j = n$. From Lemma 2.1, we have

$$\pi_n||(I-S_{(n)})^{-1}||_\infty = \frac{||(I-S_{(n)})^{-1}||_\infty}{1+x^T(I-S_{(n)})^{-1}1},$$

where x^T is the vector consisting of the first $n-1$ entries of the last row of S. Now $x^T 1\le 1$, so we see that $||(I-S_{(n)})^{-1}||_\infty \ge x^T(I-S_{(n)})^{-1}1$, so that

$$\frac{||(I-S_{(n)})^{-1}||_\infty}{1+x^T(I-S_{(n)})^{-1}1}\ge\frac{||(I-S_{(n)})^{-1}||_\infty}{1+||(I-S_{(n)})^{-1}||_\infty}.$$

Since $||(I-S_{(n)})^{-1}||_\infty\ge 1$, we see that

$$\frac{||(I-S_{(n)})^{-1}||_\infty}{1+||(I-S_{(n)})^{-1}||_\infty}\ge\frac{1}{2},$$

yielding the desired inequality.

From the argument above, we see that $\pi_n||(I-S_{(n)})^{-1}||_\infty = 1/2$ if and only if $x^T 1 = 1$ and $||(I-S_{(n)})^{-1}||_\infty = 1$; since $(I-S_{(n)})^{-1} = \sum_{k\ge 0} S_{(n)}^k$, the latter condition is easily seen to be equivalent to $S_{(n)} = 0$. □

We remark that $\max_{1\le j\le n}\pi_j||(I-S_{(j)})^{-1}||_\infty$ is unbounded as S ranges over the class of irreducible $n\times n$ stochastic matrices; to see this, note that for

$$S=\left[\begin{array}{c|c}(1-\epsilon)I & \epsilon 1\\ \hline (\epsilon/(n-1))1^T & 1-\epsilon\end{array}\right],$$

we have $\pi_n||(I-S_{(n)})^{-1}||_\infty = 1/(2\epsilon)$, which is unbounded as $\epsilon\to 0^+$. In contrast, the next result provides an attainable lower bound on $\max_{1\le j\le n}\pi_j||(I-S_{(j)})^{-1}||_\infty$.

THEOREM 2.11. *Let S be an $n\times n$ stochastic matrix with left stationary vector π^T. Then*

$$\max_{1\le j\le n}\pi_j||(I-S_{(j)})^{-1}||_\infty \ge (n-1)/n. \tag{1}$$

Equality holds if and only if

(i) *S is a doubly stochastic matrix with zero diagonal;*

(ii) *$||(I-S_{(j)})^{-1}||_\infty = n-1$ for each $j=1,\dots,n$; and*

(iii) *if i and j are indices such that $s_{j,i}>0$, then the entry of $(I-S_{(j)})^{-1}1$ corresponding to index i is equal to $n-1$.*

Proof. For each $j=1,\dots,n$, let r_j^T denote the vector formed from the jth row of S by deleting the jth entry. By Lemma 2.1, we have $\pi_j=\frac{1}{1+r_j^T(I-S_{(j)})^{-1}1}$. Thus we have

$$\begin{aligned}1&=\sum_{j=1}^n\pi_j=\sum_{j=1}^n\frac{1}{1+r_j^T(I-S_{(j)})^{-1}1}\\ &\ge\sum_{j=1}^n\frac{1}{1+||(I-S_{(j)})^{-1}||_\infty}\ge\frac{n}{1+\max_{1\le j\le n}||(I-S_{(j)})^{-1}||_\infty}.\end{aligned}$$

Consequently, we find that $\max_{1\le j\le n}||(I-S_{(j)})^{-1}||_\infty\ge n-1$. Hence, applying the fact that the function $t/(t+1)$ is increasing for $t>0$, we find that

$$\max_{1\le j\le n}\pi_j||(I-S_{(j)})^{-1}||_\infty\ge\max_{1\le j\le n}\frac{||(I-S_{(j)})^{-1}||_\infty}{1+||(I-S_{(j)})^{-1}||_\infty}$$

$$=\frac{\max_{1\le j\le n}||(I-S_{(j)})^{-1}||_\infty}{1+\max_{1\le j\le n}||(I-S_{(j)})^{-1}||_\infty}\ge\frac{n-1}{n},$$

yielding the desired inequality.

Note that if (i), (ii), and (iii) hold, then for each $j=1,\dots,n$, $\pi_j||(I-S_{(j)})^{-1}||_\infty = (n-1)/n$ so that equality holds in (1). Now suppose that $\max_{1\le j\le n}\pi_j||(I-S_{(j)})^{-1}||_\infty = (n-1)/n$. From the argument above, we see that this implies that for each $j=1,\dots,n$, we have $||(I-S_{(j)})^{-1}||_\infty = n-1$, $r_j^T(I-S_{(j)})^{-1}1 = n-1$, and $\pi_j=1/n$. In particular, since each $r_j^T(I-S_{(j)})^{-1}1=n-1=||(I-S_{(j)})^{-1}||_\infty$, we have $r_j^T1=1$, so that S has zero diagonal. Further, it must also be the case that r_j has positive entries only in positions corresponding to rows of $(I-S_{(j)})^{-1}$ attaining the maximum sum of $n-1$. Conditions (i), (ii), and (iii) now follow readily. ∎

Remark 2.12. It is well known (see [8]) that the mean first passage time from state i to state j in the Markov chain corresponding to S is given by $m_{i,j}=e_i^T(I-S_{(j)})^{-1}1$. Thus condition (iii) in Theorem 2.11 can be rephrased in terms of the chain as follows:

If distinct states i and j have the property that state i can be reached from state j in one step, then $m_{i,j} = n - 1$.

Remark 2.13. We note that if S satisfies conditions (i), (ii), and (iii) above, then by Theorem 2.8 we also have $\kappa_7 = n\kappa_3$. Thus each $n \times n$ stochastic matrix for which κ_3 is a minimum also has the property that κ_7/κ_3 is a maximum.

Motivated in part by Remark 2.13, the last part of this paper pursues the case of equality in (1). Our next two examples illustrate that situation.

Example 2.14. Let $S = \frac{1}{n-1}(J - I)$, where J denotes the $n \times n$ all-ones matrix. For each $j = 1, \ldots, n$, we have $(I - S_{(j)})\mathbf{1} = \frac{1}{n-1}\mathbf{1}$, so that $(I - S_{(j)})^{-1}\mathbf{1} = (n-1)\mathbf{1}$. Evidently S is doubly stochastic with zero diagonal, and it follows that conditions (i), (ii), and (iii) of Theorem 2.11 are satisfied, so that equality holds in (1).

Example 2.15. Consider a transition matrix whose directed graph is an n-cycle, say

$$S = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Then $(I - S_{(n)})^{-1}$ is the upper triangular matrix with ones on and above the diagonal, so that

$$(I - S_{(n)})^{-1}\mathbf{1} = \begin{bmatrix} n-1 \\ n-2 \\ \vdots \\ 2 \\ 1 \end{bmatrix}.$$

Evidently $||(I - S_{(n)})^{-1}||_\infty = n - 1$ and $s_{n,i} > 0$ only for $i = 1$, in which case, $e_i^T(I - S_{(n)})^{-1}\mathbf{1} = n - 1$. Since all states in the Markov chain corresponding to S are isomorphic, it follows that conditions (i), (ii), and (iii) of Theorem 2.11 are satisfied, so that equality holds in (1).

Based on the preceding examples, we formulate the following.

Conjecture 1. *The only irreducible stochastic matrices of order n satisfying* (i), (ii), *and* (iii) *of Theorem* 2.11 *are the permutation matrices corresponding to n-cycles, and* $\frac{1}{n-1}(J - I)$.

Next, we confirm Conjecture 1 in a special case.

Proposition 2.16. *Suppose that S is an irreducible stochastic matrix of order n satisfying* (i), (ii), *and* (iii) *of Theorem* 2.11. *If S has no off-diagonal zero entries, then* $S = \frac{1}{n-1}(J - I)$.

Proof. Since $s_{j,i} > 0$ whenever $i \neq j$, we see from (ii) that for each j, $(I - S_{(j)})^{-1}\mathbf{1} = (n-1)\mathbf{1}$, or equivalently, $S_{(j)}\mathbf{1} = \frac{n-2}{n-1}\mathbf{1}$. But since $S\mathbf{1} = \mathbf{1}$, it follows that each off-diagonal entry in the jth column of S is $1/(n-1)$. The result now follows. ∎

We close with a verification of Conjecture 1 for matrices of low order.

Proposition 2.17. *Conjecture* 1 *holds for* $n = 2, 3, 4$.

Proof. The only 2×2 matrix satisfying (i), (ii), and (iii) of Theorem 2.11 is $\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, which evidently has the form described in Conjecture 1. Each 3×3 doubly stochastic

matrix with zero diagonal has the form

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

for some $0 \leq a \leq 1$. If a is either 0 or 1, then we have a permutation matrix for a 3-cycle. If $0 < a < 1$, then by Proposition 2.16, a must be $1/2$. This confirms Conjecture 1 when $n = 3$.

Now suppose that S is a 4×4 matrix satisfying (i), (ii), and (iii) of Theorem 2.11. If S has no off-diagonal zeros, then $S = (1/3)(J - I)$ by Proposition 2.16. So suppose that S has an off-diagonal zero; by applying a permutation similarity if necessary, we take $s_{1,4} = 0$. By Birkhoff's theorem (see [3], for example), S is a convex combination of permutation matrices. Since S has zero diagonal and we are taking $s_{1,4}$ to be 0, it follows that S is a convex combination of the following six matrices:

$$P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad P_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad P_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$P_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad P_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad P_6 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Thus we have $S = \sum_{i=1}^{6} a_i P_i$ for some $a_1, \ldots, a_6 \geq 0$ such that $\sum_{i=1}^{6} a_i = 1$.

For $1 \leq i \leq 6$, let δ_i be 0 if $a_i = 0$, and let δ_i be 1 if $a_i > 0$. Let M be the mean first passage matrix for the Markov chain corresponding to S (see [4]). Recall that the diagonal entries of M are the reciprocals of the corresponding entries in the left stationary vector, and that $M = SM - S\mathrm{diag}(M) + J$, where $\mathrm{diag}(M)$ denotes the diagonal matrix arising from the diagonal entries of M. Since S is a 4×4 doubly stochastic matrix, each diagonal entry of M is 4 so that $M = SM - 4S + J$. From Remark 2.12 we find that if $s_{i,j} > 0$, then $m_{j,i} = 3$. Consequently, it follows that $M \geq 3 \sum_{i=1}^{6} \delta_i P_i^T$. Thus we find that

$$M = SM - 4S + J \geq 3 \left(\sum_{i=1}^{6} a_i P_i \right) \left(\sum_{i=1}^{6} \delta_i P_i^T \right) - 4S + J$$

$$= 3 \sum_{i=1}^{6} a_i \delta_i P_i P_i^T + 3 \left(\sum_{1 \leq i,j \leq 6, i \neq j} a_i a_j \delta_i \delta_j P_i P_j^T \right) - 4S + J.$$

Since $\sum_{i=1}^{6} a_i \delta_i = 1$, we find that

$$M \geq 3I + J - 4S + 3 \left(\sum_{1 \leq i,j \leq 6, i \neq j} a_i a_j \delta_i \delta_j P_i P_j^T \right).$$

In particular, if $a_i, a_j > 0$ for some $i \neq j$, then necessarily the diagonal entries of $P_i P_j^T$ must be 0, otherwise we find that some diagonal entry of M exceeds 4. We thus conclude that if $a_i, a_j > 0$ for some $i \neq j$, then there is no position such that both P_i and P_j have 1's in that position.

Applying that restriction to our list of admissible permutation matrices, we find that for some $0 \leq a \leq 1, S$ is one of the following matrices: $T(a) \equiv aP_1 + (1-a)P_3, U(a) \equiv aP_2 + (1-a)P_3, V(a) \equiv aP_4 + (1-a)P_6$, and $W(a) = aP_5 + (1-a)P_2$. If a is either 0 or 1, then each of T, U, V, W either is a permutation matrix corresponding to a 4-cycle or is reducible, confirming the conjecture. Thus it remains only to consider the case that $0 < a < 1$.

First we deal with $T(a)$. Computing the first row of $(I-(T(a))_{(4)})^{-1}$ by cofactors yields

$$\frac{1}{a(a^2-2a+2)}[1 \quad a \quad (a^2+1-a)],$$

which sums to $(2+a^2)/(a(a^2-2a+2))$; it is readily seen that this expression is greater than 3, so that T does not satisfy (ii). Noting that $W(a)$ is permutationally similar to $T(a)$, we find that W does not satisfy (ii).

Next, we consider $U(a)$. A direct computation shows that

$$(I-(U(a))_{(4)})^{-1} = \begin{bmatrix} \frac{1}{2a(1-a)} & \frac{1}{2(1-a)} & \frac{1}{2a} \\ \frac{1}{2(1-a)} & 1+\frac{a}{2(1-a)} & \frac{1}{2} \\ \frac{1}{2a} & \frac{1}{2} & 1+\frac{1-a}{2a} \end{bmatrix}.$$

As a result,

$$(I-(U(a))_{(4)})^{-1}\mathbf{1} = \begin{bmatrix} \frac{1}{a(1-a)} \\ \frac{2-a}{1-a} \\ \frac{a+1}{a} \end{bmatrix}.$$

It is now straightforward to show that $||(I-(U(a))_{(4)})^{-1}||_\infty > 3$, so that U does not satisfy (ii).

Finally, we consider $V(a)$. A computation reveals that

$$(I-(V(a))_{(4)})^{-1} = \frac{1}{a^2+(1-a)^2}\begin{bmatrix} 1 & a & 1-a \\ 1-a & 1-a+a^2 & (1-a)^2 \\ a & a^2 & 1-a+a^2 \end{bmatrix}.$$

If V satisfies (iii), then from the fact that $v_{4,2}, v_{4,3} > 0$, it follows that the last two rows of $(I-(V(a))_{(4)})^{-1}$ sum to 3. We find that $a = 1/2$, which implies that the first row of $(I-(V(a))_{(4)})^{-1}$ sums to 4, so that V does not satisfy (ii).

Consequently, if $0 < a < 1$, then none of T, U, V, and W satisfies (i)–(iii). Thus we find that Conjecture 1 holds for $n = 4$. □

3. Related remarks. It is natural to consider some of the issues surrounding the implementation of the bounds κ_3 and κ_7 and in this section we do so.

We begin by discussing the number of operations necessary to compute κ_3. As is observed in [5], for an irreducible stochastic $n \times n$ matrix S with left stationary vector π^T,

$$\max_{1 \leq j \leq n} \pi_j ||(I-S_{(j)})^{-1}||_\infty = \max_{1 \leq j \leq n} \left(q_{jj}^{\#} - \min_{1 \leq i \leq n} q_{ij}^{\#} \right),$$

where $Q \equiv I - S$ and where $Q^{\#}$ denotes the group generalized inverse of Q (see [4] for background on this generalized inverse). According to the results in [1], applying the so-called shuffle algorithm to Q yields $Q^{\#}$ in roughly $3n^3$ arithmetic operations (of the multiply-and-add type); once $Q^{\#}$ has been found, κ_3 can then be computed at negligible additional expense.

The computation of κ_7 turns out to be fairly expensive, as is noted in [7]. For example, using Gaussian elimination to compute $(I-S_{(i)})^{-1}\mathbf{1}$, from which we can find $||(I-S_{(i)})^{-1}||_\infty$, requires roughly $n^3/3$ arithmetic operations. With this approach, we see that κ_7 will incur a cost of about $n^4/3$ arithmetic operations, which is significantly worse than the shuffle algorithm approach to κ_3 described above.

In order to save on the number of operations involved in finding κ_7, one might want to simply provide an upper estimate on that quantity. One approach would be to find $||(I - S_{(i)})^{-1}||_\infty$ for some i. As noted above this is already an order n^3 task, as is the computation of κ_3. Moreover, it is conceivable that an unlucky choice of i could lead to a poor estimate. For instance, consider the matrix of Example 2.7, with x equal to $\frac{1}{n-1}\mathbf{1}$. Then for each $i = 1, \ldots, n-1$, we have $||(I - S_{(i)})^{-1}||_\infty = 2(n-1)$, while $||(I - S_{(n)})^{-1}||_\infty = 1$. Thus in that example, for large values of n there is exactly one choice of i (namely $i = n$) for which $||(I - S_{(i)})^{-1}||_\infty$ is of the right order of magnitude. Another strategy might be to use the easily computed quantity $1/(1 - ||S_{(i)}||_\infty) = 1/(\min_{j\neq i} s_{ji})$ as a further upper bound on $||(I - S_{(i)})^{-1}||_\infty$. Evidently, this is feasible only in the case that the ith column of S has no off-diagonal zeros. In particular, if S has an off-diagonal zero in each column, then this approach is not helpful.

Given that κ_3 is a tighter bound than κ_7, that κ_7 can be greater than κ_3 by a factor of n, and that κ_7 may be difficult to estimate efficiently, it seems that κ_3 offers some advantage over κ_7 as a condition number. However, as the discussion of κ_7 in [7] makes clear, it provides valuable qualitative information concerning the nature of a Markov chain.

Acknowledgment. The author is grateful to the referees, whose comments resulted in improvements to this paper.

REFERENCES

[1] K. Anstreicher and U. Rothblum, *Using Gauss–Jordan elimination to compute the index, generalized null space, and Drazin inverse*, Linear Algebra Appl., 85 (1987), pp. 221–239.
[2] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
[3] R. Brualdi and H. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, UK, 1991.
[4] S. Campbell and C. Meyer, *Generalized Inverses of Linear Transformations*, Dover, New York, 1991.
[5] G. Cho and C. Meyer, *Comparison of perturbation bounds for the stationary distribution of a Markov chain*, Linear Algebra Appl., 335 (2001), pp. 137–150.
[6] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[7] I. Ipsen and C. Meyer, *Uniform stability of Markov chains*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1061–1074.
[8] J. Kemeny and J. Snell, *Finite Markov Chains*, Springer-Verlag, New York, 1976.
[9] E. Seneta, *Non-negative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1120–1142
© 2002 Society for Industrial and Applied Mathematics

A REGULARIZED ROBUST DESIGN CRITERION FOR UNCERTAIN DATA*

A. H. SAYED†, V. H. NASCIMENTO‡, AND F. A. M. CIPPARRONE‡

Abstract. This paper formulates and solves a robust criterion for least-squares designs in the presence of uncertain data. Compared with earlier studies, the proposed criterion incorporates simultaneously both regularization and weighting and applies to a large class of uncertainties. The solution method is based on reducing a vector optimization problem to an equivalent scalar minimization problem of a provably unimodal cost function, thus achieving considerable reduction in computational complexity.

Key words. least-squares, regularization, robustness, min-max, uncertainty, game problem

AMS subject classifications. 15A06, 15A63, 65K10, 90C47, 91A40

PII. S0895479800380799

1. Introduction.

As is well known, many estimation and control problems rely on solving regularized least-squares problems of the form

$$\min_x \left[x^T Q x + (Ax-b)^T W (Ax-b)\right], \tag{1.1}$$

where $x^T Q x$ is a regularization term, $Q > 0$ and $W \geq 0$ are Hermitian weighting matrices, x is an unknown n-dimensional column vector, A is a known $N \times n$ data matrix, and b is a known $N \times 1$ measurement vector. The solution of (1.1) is

$$\hat{x} = \left[Q + A^T W A\right]^{-1} A^T W b, \tag{1.2}$$

where the invertibility of $(Q + A^T W A)$ is guaranteed by the positive-definiteness of Q.

When the nominal data $\{A, b\}$ are subject to disturbances and/or uncertainties, the performance of the optimal estimator (1.1) can degrade appreciably. For example, if the actual data matrix were $(A + \delta A)$ for some unknown perturbation δA, then the estimator (1.1) that is designed based on A alone, and without accounting for the existence of δA, can perform poorly. This fact has motivated numerous works in the literature that attempt to make the solution of least-squares designs robust in the presence of data uncertainties. Some notable methods are the total-least-squares and the $\mathcal{H}_\infty$ formalisms (see, e.g., [1, 2] and the many references therein). These methods are known to lead to solutions that perform data deregularization and which, at times, may be conservative.

In this work, we propose a robust alternative to the regularized and weighted least-squares problem (1.1), which is shown to lead to a regularized solution, as opposed to a

*Received by the editors November 9, 2000; accepted for publication (in revised form) by S. Van Huffel September 6, 2001; published electronically April 10, 2002. This work was supported in part by a grant from the National Science Foundation under award ECS-9820765. An early version of some of the results in this article appeared as reference [10] in the Proceedings of the Workshop on Robust Identification and Control, Siena, Italy, 1998.

http://www.siam.org/journals/simax/23-4/38079.html

†Department of Electrical Engineering, University of California, Los Angeles, CA 90095 (sayed@ee.ucla.edu).

‡Department of Electronic Systems Engineering (PSI-EPUSP), Universidade de São Paulo, Av. Prof. L. Gualberto, trav. 3, nº 158, CEP 05508-900, SP, Brazil (cippafla@voz1.lps.usp.br).

deregularized solution. This property is useful, especially for on-line implementations, since the regularized solution does not require existence conditions. The special case of $Q = 0$ and $W = I$ (which corresponds to unweighted least-squares problems without regularization) was studied in [3, 4] by different methods; one relies on linear matrix inequality (LMI) techniques while the other relies on SVD techniques. It turns out that nontrivial choices for $\{Q, W\}$ require special care, and a technique is developed here that leads to the following contributions. First, the technique can handle general choices for $\{Q, W\}$. Second, the problem formulation applies to a large class of data uncertainties, as will be explained below. And third, we show how to replace a vector optimization problem by a scalar minimization problem of a cost function that is provably unimodal. This step leads to significant simplifications in complexity, and a justification for its validity is provided in the appendices at the end of the paper.

Applications of the proposed methodology to recursive estimation, control, and data fusion problems appear in [5, 6, 7, 8]; we refer the reader to these articles for motivation, examples, simulations, and comparisons with other related techniques. As a brief motivation, one application in the context of state-space estimation is succinctly described in section 2.3, with full details provided in [6]. In most of the paper, however, we opt to focus on studying the properties and technical aspects of the robust least-squares problem that is formulated below in (2.1).

As mentioned above, the formulation in this article is useful for at least two reasons. First, it leads to a robust solution that involves regularization rather than deregularization. In this way, existence conditions do not arise, which could be a burden for on-line solutions (see, e.g., [6, 7]). Second, the framework incorporates both regularization and weighting into the cost function. Such extensions are needed in order to handle, for example, quadratic control and estimation problems where regularization and weighting are prevalent (see, e.g., [5, 6, 7, 10]).

2. Problem formulation. A generalization of the cost function in (1.1) that accounts for uncertainties in the data $\{A, b\}$ can be obtained as follows. Introduce the two-variable cost function

$$J(x, y) \triangleq x^T Q x + R(x, y),$$

where $R(x, y)$ is a modified residual term that is defined by

$$R(x, y) \triangleq \Big(Ax - b + Hy\Big)^T W \Big(Ax - b + Hy\Big).$$

Here, H is an $N \times m$ known matrix, and y denotes an $m \times 1$ unknown perturbation vector. When $H = 0$, $J(x, y)$ reduces to the standard regularized cost function in (1.1). The presence of H and y in the expression for $R(x, y)$ allows us to account for uncertainties in the data, as will become more evident from the discussions in what follows.

To guarantee optimal performance in a worst-case scenario, we consider a min-max optimization problem of the form

$$\hat{x} = \arg\min_{x} \max_{\|y\| \le \phi(x)} J(x, y), \tag{2.1}$$

where the notation $\|\cdot\|$ stands for the Euclidean norm of its vector argument or the maximum singular value of its matrix argument. The nonnegative function $\phi(x)$ is assumed to be a known bound on the perturbation y and is a function of x only.

Problem (2.1) can be interpreted as a constrained two-player game problem, with the designer trying to pick an $\hat{x}$ that minimizes the cost, while the opponent $\{y\}$ tries to maximize the cost (e.g., [9]). The game problem is constrained since it imposes a limit (through $\phi(x)$) on how large (or how damaging) the opponent $\{y\}$ can be. We assume in what follows that H and $\phi(x)$ are not identically zero, i.e.,

$$H \neq 0 \quad \text{and} \quad \phi(x) \not\equiv 0,$$

since if either is zero, the game problem (2.1) trivializes to the standard regularized least-squares problem (1.1). The choice of H allows us to handle situations in which the uncertainties are known to be restricted to a certain subspace.

An initial study of problem (2.1) appears in [10] without the details and some of the properties that are offered in this article and, in particular, without the arguments and proofs for general functions $\phi(x)$ that appear in the appendices of this paper.

Two useful special cases of the formulation (2.1) are described below. They correspond to special choices of the function $\phi(x)$. These examples are meant to show how the freedom in selecting $\phi(x)$ allows us to handle different uncertainty models.

2.1. A special case: Bounded uncertainties. Consider uncertainties $\{\delta A, \delta b\}$ that are only known to lie within certain balls of radii $\{\eta, \eta_b\}$, i.e., they are known to be bounded and satisfy

$$\|\delta A\| \leq \eta, \quad \|\delta b\| \leq \eta_b.$$

Now consider an optimization problem of the form

$$\min_{x} \max_{\substack{\|\delta A\| \leq \eta \\ \|\delta b\| \leq \eta_b}} \left[x^T Q x + \Big((A+\delta A)x - (b+\delta b)\Big)^T W \Big((A+\delta A)x - (b+\delta b)\Big) \right]. \tag{2.2}$$

It can be verified that this problem is a special case of (2.1) since it can be shown to be equivalent to a problem of the form

$$\min_{x} \max_{\|y\| \leq \eta\|x\| + \eta_b} \left[x^T Q x + \Big(Ax - b + y\Big)^T W \Big(Ax - b + y\Big) \right], \tag{2.3}$$

which corresponds to the special choices $H = I$ and $\phi(x) = \eta\|x\| + \eta_b$.

To verify that problems (2.1) and (2.3) are indeed equivalent, we proceed as in [5] and show that the two variables $\{\delta A, \delta b\}$ in (2.1) can be replaced by a single variable y, which would therefore allow us to replace the maximization in (2.1) over *two* constrained variables by a maximization over a *single* constrained variable as in (2.3).

Indeed, for any fixed value of x, let $\mathcal{Z}_x$ denote the set of all vectors z that are generated as follows:

$$\mathcal{Z}_x = \{z : \ \ z = \delta A x - \delta b, \ \ \|\delta A\| \leq \eta, \ \ \|\delta b\| \leq \eta_b\}$$

for all possible $\{\delta A, \delta b\}$ within the prescribed bounds. Also let $\mathcal{Y}_x$ denote the set of all vectors y that are generated as follows:

$$\mathcal{Y}_x = \{y : \ \ \|y\| \leq \eta\|x\| + \eta_b\}.$$

Then $\mathcal{Z}_x = \mathcal{Y}_x$. That is, if $z \in \mathcal{Z}_x$, then $z \in \mathcal{Y}_x$. (This direction is immediate and follows from the triangle inequality of norms.) Conversely, if $y \in \mathcal{Y}_x$, then $y \in \mathcal{Z}_x$. To establish the result for $x \neq 0$, define for a given y the perturbations

$$\delta A(y) = \frac{\eta}{\eta\|x\| + \eta_b} \frac{yx^T}{\|x\|}, \qquad \delta b(y) = -\frac{\eta_b y}{\eta\|x\| + \eta_b}. \tag{2.4}$$

Then $\{\delta A(y), \delta b(y)\}$ are valid perturbations and $y = \delta A(y)x - \delta b(y)$ so that $y \in \mathcal{Z}_x$, which justifies our claim. (When $x = 0$, we select $\delta b = -y$ and δA arbitrary.)

As mentioned before, the special case $Q = 0$ and $W = I$ was treated in [3, 4] by different methods; one uses SVD techniques while the other uses LMI techniques. For this special case, a geometric framework that is similar in nature to the geometry of least-square problems was also developed in [11, 12].

2.2. A special case: Uncertainties in factored form. Consider now a problem of the form

$$\min_x \max_{\delta A, \delta b} \left[x^T Q x + \Big((A + \delta A)x - (b + \delta b)\Big)^T W \Big((A + \delta A)x - (b + \delta b)\Big) \right], \tag{2.5}$$

where the perturbations $\{\delta A, \delta b\}$ are assumed to satisfy a model of the form

$$\begin{bmatrix} \delta A & \delta b \end{bmatrix} = HS \begin{bmatrix} E_a & E_b \end{bmatrix}, \tag{2.6}$$

where S is an arbitrary contraction, $\|S\| \leq 1$, and $\{H, E_a, E_b\}$ are *known* quantities of appropriate dimensions. Perturbation models of this form are common in robust filtering and control and can arise from tolerance specifications on physical parameters (see [13]). The quantity H allows the designer to restrict the range of allowable uncertainties $\{\delta A, \delta b\}$ to a certain column span. Assume, for example, that one wishes to model only uncertainties in the $(0, 0)$ entry of A. Then one could choose

$$H = \mathrm{col}\{1, 0, \ldots, 0\}, \quad E_a = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}, \quad E_b = 0,$$

and S would denote in this case an arbitrary scalar that is less than unity in magnitude. Other choices for $\{H, E_a, E_b\}$ would correspond to different assumptions on the uncertainties.

In order to see how (2.5) is related to (2.1), we rewrite the cost in (2.5) as

$$\left[x^T Q x + \Big(Ax - b + (\delta A x - \delta b)\Big)^T W \Big(Ax - b + (\delta A x - \delta b)\Big) \right]$$

so that with y defined as $y = S(E_a x - E_b)$ and Hy defined as

$$Hy \triangleq \delta A x - \delta b = HS(E_a x - E_b)$$

problem (2.5) can be verified to be equivalent to the following problem:

$$\min_x \max_{\|y\| \leq \|E_a x - E_b\|} \left[x^T Q x + \Big(Ax - b + Hy\Big)^T W \Big(Ax - b + Hy\Big) \right],$$

which is again a special case of (2.1) for the particular choice $\phi(x) = \|E_a x - E_b\|$.

2.3. An application: State estimation. Before proceeding to a discussion of the solution and properties of the general problem (2.1), we motivate this optimization problem by considering an application in the context of state-space estimation.

Thus consider a state-space model of the form

$$x_{i+1} = F_i x_i + G_i u_i, \quad i \geq 0, \tag{2.7}$$

$$y_i = H_i x_i + v_i, \tag{2.8}$$

where $\{x_0, u_i, v_i\}$ are uncorrelated zero-mean random variables with variances

$$E\left(\begin{bmatrix} x_0 \\ u_i \\ v_i \end{bmatrix}\begin{bmatrix} x_0 \\ u_j \\ v_j \end{bmatrix}^T\right) = \begin{bmatrix} \Pi_0 & 0 & 0 \\ 0 & Q_i\delta_{ij} & 0 \\ 0 & 0 & R_i\delta_{ij} \end{bmatrix} \tag{2.9}$$

that satisfy $\Pi_0 > 0$, $R_i > 0$, and $Q_i > 0$. Here, δ_{ij} is the Kronecker delta function that is equal to unity when $i = j$ and zero otherwise. The well-known Kalman filter [14] provides the optimal linear least-mean-squares (l.l.m.s., for short) estimate of the state variable given prior observations. It admits the following deterministic interpretation [15].

Fix a time instant i and assume that a so-called filtered estimate $\hat{x}_{i|i}$ of x_i has already been computed with the corresponding error variance matrix $P_{i|i}$. Given a new measurement y_{i+1}, one can seek to improve the estimate of x_i, along with estimating u_i, by solving

$$\min_{x_i, u_i} \left[\|x_i - \hat{x}_{i|i}\|^2_{P_{i|i}^{-1}} + \|u_i\|^2_{Q_i^{-1}} + \|y_{i+1} - H_{i+1}x_{i+1}\|^2_{R_{i+1}^{-1}} \right]. \tag{2.10}$$

Substituting x_{i+1} by the state-equation $x_{i+1} = F_i x_i + G_i u_i$, the above cost function becomes one of the regularized and weighted least-squares form (1.1), and its solution leads to the Kalman filter recursions.

Now, assume that the state-space model includes parametric uncertainties, say of the form

$$x_{i+1} = (F_i + \delta F_i)x_i + (G_i + \delta G_i)u_i, \quad i \geq 0, \tag{2.11}$$

$$y_i = H_i x_i + v_i, \tag{2.12}$$

where the uncertainties $\{\delta F_i, \delta G_i\}$ lie in a certain domain, say of the form

$$\begin{bmatrix} \delta F_i & \delta G_i \end{bmatrix} = M_i S \begin{bmatrix} E_{f,i} & E_{g,i} \end{bmatrix}$$

for some known matrices $\{M_i, E_{f,i}, E_{g,i}\}$ and an arbitrary contraction S. We can then consider replacing (2.10) by

$$\min_{\{x_i, u_i\}} \max_{\delta F_i, \delta G_i} \left[\|x_i - \hat{x}_{i|i}\|^2_{P_{i|i}^{-1}} + \|u_i\|^2_{Q_i^{-1}} + \|y_{i+1} - H_{i+1}x_{i+1}\|^2_{R_{i+1}^{-1}} \right]. \tag{2.13}$$

If we substitute x_{i+1} by its state-equation $x_{i+1} = (F_i + \delta F_i)x_i + (G_i + \delta G_i)u_i$, the above min-max problem becomes again a special case of the robust cost function (2.1), actually one of the form (2.5)–(2.6); see [6] for the details, including numerical examples and comparison with several other classes of state-space estimation algorithms such as Kalman filters, $\mathcal{H}_\infty$ filters, guaranteed-cost filters, and set-valued estimation filters.

3. Solution of the optimization problem. We now proceed to the solution of problem (2.1). In particular, we shall show that the solution has a regularized form, albeit one that operates on corrected data; i.e., it replaces $\{Q, W\}$ by corrections $\{\widehat{Q}, \widehat{W}\}$. In addition, and significantly, we shall show that the corrected parameters are determined in terms of the *unique* minimizing scalar argument, λ^o, of a *unimodal* cost function. In this way, we end up with a technique that enforces robustness via regularization, rather than deregularization as is common in many robust procedures in the literature, and whose optimal solution involves determining the minimizing argument of a scalar unimodal function, a step that simplifies the complexity of the solution to great extent.

3.1. Uniqueness of solution. We start by noting that the condition $Q > 0$ implies that (2.1) has a unique, finite solution. Indeed, for any given y, the residual cost $R(x, y)$ is convex in x. Therefore, the maximum

$$C(x) \stackrel{\Delta}{=} \max_{\|y\| \leq \phi(x)} R(x, y) \tag{3.1}$$

is a convex function in x. In addition, the first term in $J(x, y)$, $x^T Q x$, is strictly convex in x and radially unbounded (i.e., $|x^T Q x|$ goes to infinity as $\|x\| \to \infty$) when $Q > 0$. We conclude that $x^T Q x + C(x)$ is also strictly convex in x and radially unbounded, which implies that problem (2.1) has a unique global minimum $\hat{x}$. To determine $\hat{x}$, we proceed in steps.

3.2. The maximization problem. We first solve (3.1) for any fixed x. Note that for fixed x, both the cost $R(x, y)$ and the constraint $\|y\| \leq \phi(x)$ are convex in y, so that the maximum

$$\max_{\|y\| \leq \phi(x)} R(x, y)$$

is achieved at the boundary, $\|y\| = \phi(x)$. We can therefore replace the inequality constraint in (3.1) by an equality. Introducing a Lagrange multiplier λ, the solution to (3.1) can then be found from the unconstrained problem

$$\max_{y, \lambda} \Big[\big(Ax - b + Hy\big)^T W \big(Ax - b + Hy\big) - \lambda\big(\|y\|^2 - \phi^2(x)\big) \Big]. \tag{3.2}$$

Differentiating (3.2) with respect to y and λ, and denoting the optimal solutions by $\{y^o, \lambda^o\}$, we obtain the equations

$$(\lambda^o I - H^T W H) y^o = H^T W (Ax - b), \qquad \|y^o\| = \phi(x). \tag{3.3}$$

It turns out that the solution λ^o should satisfy $\lambda^o \geq \|H^T W H\|$. This is because the Hessian of the cost in (3.2) with respect to y, which is equal to

$$H^T W H - \lambda I,$$

must be nonpositive-definite at $\lambda = \lambda^o$ [16].[1] We should further stress that the solutions $\{y^o, \lambda^o\}$ depend on x, and we shall therefore sometimes denote this dependence explicitly by writing $\{y^o(x), \lambda^o(x)\}$.[2]

[1] We refer to the case $\lambda^o = \|H^T W H\|$ as the *singular* case, while $\lambda^o > \|H^T W H\|$ is called the *regular* case. Both cases are handled simultaneously in our framework through the use of the pseudo-inverse notation.

[2] In fact, we show in Appendix A that the solution $\lambda^o(x)$ is always a continuous function of x, while there might exist several y^o when $\lambda^o(x) = \|H^T W H\|$.

At this stage, we do not need to solve the equations (3.3) for $\{y^o, \lambda^o\}$. It is enough to know that the optimal $\{y^o, \lambda^o\}$ satisfy (3.3). Using this fact, we can verify that the maximum cost in (3.2) is equal to

$$C(x) = (Ax-b)^T \left[W + WH\big(\lambda^o(x)I - H^TWH\big)^\dagger H^TW\right](Ax-b) + \lambda^o(x)\phi^2(x), \tag{3.4}$$

where the notation $X^\dagger$ denotes the pseudo-inverse of X.

3.3. The minimization problem. The original problem (2.1) is therefore equivalent to

$$\min_x \ \left[x^TQx \ + \ C(x)\right]. \tag{3.5}$$

However, rather than minimizing the above cost over n variables, which are the entries of the vector x, we can instead reduce the problem to one of minimizing a certain cost function over a single scalar variable (see (3.9) further ahead). For this purpose, we introduce the following function of two independent variables x and λ,

$$C(x,\lambda) = (Ax-b)^T \left[W + WH\big(\lambda I - H^TWH\big)^\dagger H^TW\right](Ax-b) + \lambda\phi^2(x),$$

where λ is an independent variable. Then it can be verified, by direct differentiation with respect to λ and by using the expression for $\lambda^o(x)$ from (3.3), that

$$\lambda^o(x) \ = \ \arg\min_{\lambda \geq \|H^TWH\|} C(x,\lambda)\,. \tag{3.6}$$

In other words, the optimal $\lambda^o(x)$ from (3.3) coincides with the argument that minimizes $C(x,\lambda)$ over λ (with λ restricted to the interval $\big[\ \|H^TWH\|, \infty\big)$).

In this way, problem (2.1) becomes equivalent to

$$\min_x \min_{\lambda \geq \|H^TWH\|} \left[x^TQx \ + \ C(x,\lambda)\right] = \min_{\lambda \geq \|H^TWH\|} \min_x \left[x^TQx \ + \ C(x,\lambda)\right]. \tag{3.7}$$

The cost function in the above expression, viz., $J(x,\lambda) = x^TQx + C(x,\lambda)$, is now a function of two independent variables $\{x,\lambda\}$. This should be contrasted with the cost function in (3.5). Now, for compactness of notation, we introduce the quantities

$$\begin{aligned} W(\lambda) &\triangleq W + WH\big(\lambda I - H^TWH\big)^\dagger H^TW, \\ M(\lambda) &\triangleq Q + A^TW(\lambda)A, \\ D(\lambda) &\triangleq A^TW(\lambda)b. \end{aligned}$$

To solve problem (3.7), we first search for the minimum over x for every fixed value of λ, which can be done (if $\phi^2(x)$ is differentiable) by setting the derivative of $J(x,\lambda)$ with respect to x equal to zero. This shows that any minimum x must satisfy the equation

$$M(\lambda)x + \frac{1}{2}\lambda\nabla\phi^2(x) = D(\lambda), \tag{3.8}$$

where $\nabla\phi^2(x)$ is the gradient of $\phi^2(x)$ with respect to x. Any x satisfying this equation will of course be a function of λ, and we shall denote it by $x^o(\lambda)$. Now let $G(\lambda)$ denote the minimum value of the cost over x, i.e.,

$$G(\lambda) \triangleq \min_x [x^TQx + C(x,\lambda)] \ = \ x^{oT}(\lambda)Qx^o(\lambda) + C(x^o(\lambda),\lambda).$$

Then problem (3.7) becomes equivalent to

$$\min_{\lambda \geq \|H^T W H\|} G(\lambda).$$

Thus we see that the solution of (2.1) simply requires that we determine an optimal scalar parameter λ^o. The scalar minimization problem that defines λ^o is well behaved since (as we show in Theorem 3.1 below) $G(\lambda)$ is unimodal (i.e., there is a unique global minimum λ^o and no other local minima).

3.4. Statement of solution in the general case. Whenever $\phi(x)$ is a convex function, the cost $J(x, \lambda)$ will be strictly convex in x, and the minimization over x on the right-hand side of (3.7) will have a unique solution $x^o(\lambda)$ (as was the case with the above two special cases). We thus have a procedure that allows us to determine the minimizing x^o for every λ. This in turn allows us to re-express the resulting cost $J(x^o(\lambda), \lambda)$ as a function of λ alone, say $G(\lambda) = J(x^o(\lambda), \lambda)$. In this way, we concluded above that the solution x^o of the original optimization problem (2.1) can be solved by determining the λ^o that solves

$$\min_{\lambda \geq \|H^T W H\|} G(\lambda) \tag{3.9}$$

and by taking the corresponding $x^o(\lambda^o)$ as x^o. We summarize the solution in the following statement.

THEOREM 3.1 (solution). *Consider a regularized and weighted robust least-squares problem of the form*

$$\hat{x} = \arg\min_x \max_{\|y\| \leq \phi(x)} \left[x^T Q x + \Big(Ax - b + Hy \Big)^T W \Big(Ax - b + Hy \Big) \right], \tag{3.10}$$

where $\{A, b, H\}$ are known quantities of appropriate dimensions, $W \geq 0$ and $Q > 0$ are known weighting matrices, and $\phi(x)$ is a given convex function. It is further assumed that H and $\phi(x)$ are not identically zero. Then problem (3.10) *has a unique global minimum $\hat{x}$ that can be determined as follows:*

1. *Introduce the modified matrices*

$$\begin{aligned} W(\lambda) &\triangleq W + WH\big(\lambda I - H^T W H\big)^{\dagger} H^T W, \\ M(\lambda) &\triangleq Q + A^T W(\lambda) A, \\ D(\lambda) &\triangleq A^T W(\lambda) b. \end{aligned}$$

2. *Let $x^o(\lambda)$ denote the unique solution of the minimization problem*

$$\min_x \left[x^T Q x \; + \; (Ax - b)^T W(\lambda)(Ax - b) + \lambda \phi^2(x) \right].$$

When $\phi^2(x)$ is differentiable, $x^o(\lambda)$ can also be found as the unique solution of the equation

$$M(\lambda) x + \frac{1}{2} \lambda \nabla \phi^2(x) = D(\lambda),$$

where the notation $\nabla \phi^2(x)$ denotes the gradient of $\phi^2(x)$ with respect to x.

3. *Introduce the cost function*

$$G(\lambda) \;=\; x^{oT}(\lambda)Qx^o(\lambda) + C[x^o(\lambda), \lambda].$$

4. *Let λ^o denote the solution of the scalar-valued minimization problem*

$$\lambda^o = \arg \min_{\lambda \geq \|H^T W H\|} G(\lambda).$$

5. *Then the optimum solution of* (3.10) *is $\hat{x} = x^o(\lambda^o)$. In addition, it holds that the cost function $G(\lambda)$ is unimodal, i.e., it has a unique global minimum and no local minima.*

Proof. The only point not yet proven is the fact that $G(\lambda)$ is unimodal. This follows from Lemma C.2 in Appendix C and from the continuity of $\lambda^o(x)$ in (3.6), which is established in Appendix A.2. □

We now illustrate the solution method by reconsidering the two special cases we introduced before. In both examples, $\phi(x)$ is convex, so the minimization problem over x in (3.7) has a unique solution and is easily computable. In one of the examples, $\phi^2(x)$ is not differentiable at $x = 0$.

3.5. Uncertainties in factored form. Consider first the special case of section 2.2 with

$$\phi(x) = \|E_a x - E_b\|.$$

For this choice of $\phi(x)$, we obtain

$$\nabla \phi^2(x) = 2E_a^T \left(E_a x - E_b\right)$$

so that the solution of (3.8), which is dependent on λ, becomes

$$x^o(\lambda) = \left[M(\lambda) + \lambda E_a^T E_a\right]^{-1} \left(D(\lambda) + \lambda E_a^T E_b\right). \tag{3.11}$$

Using this expression for $x^o(\lambda)$ we find that the corresponding function $G(\lambda)$ is given by

$$G(\lambda) = \lambda E_b^T E_b + b^T W(\lambda) b - B^T(\lambda) E^{-1}(\lambda) B(\lambda),$$

where $W(\lambda)$ is as before, and the functions $\{B(\lambda), E(\lambda)\}$ are given by

$$\begin{aligned} B(\lambda) &= A^T W(\lambda) b + \lambda E_a^T E_b, \\ E(\lambda) &= Q + \lambda E_a^T E_a + A^T W(\lambda) A. \end{aligned}$$

We are thus led to the following statement.

THEOREM 3.2 (uncertainties in factored form). *Consider a regularized and weighted robust least-squares problem of the form*

$$\min_x \max_{\delta A, \delta b} \left[x^T Q x + \Big((A + \delta A)x - (b + \delta b)\Big)^T W \Big((A + \delta A)x - (b + \delta b)\Big) \right], \tag{3.12}$$

where $\{A, b\}$ are known quantities of appropriate dimensions, $W \geq 0$ and $Q > 0$ are known weighting matrices, and the perturbations $\{\delta A, \delta b\}$ are assumed to satisfy a model of the form

$$\begin{bmatrix} \delta A & \delta b \end{bmatrix} \;=\; H S \begin{bmatrix} E_a & E_b \end{bmatrix}$$

for some known quantities $\{H, E_a, E_b\}$ *and where* S *denotes an arbitrary contraction. Then problem* (3.12) *has a unique global minimum* $\hat{x}$ *that is given by (compare with* (1.2)*)*

$$\hat{x} = \left[\widehat{Q} + A^T \widehat{W} A\right]^{-1} \left[A^T \widehat{W} b + \lambda^o E_a^T E_b\right], \tag{3.13}$$

where the modified weighting matrices $\{\widehat{Q}, \widehat{W}\}$ *are obtained from* $\{Q, W\}$ *via*

$$\begin{aligned} \widehat{Q} &\triangleq Q + \lambda^o E_a^T E_a, \\ \widehat{W} &\triangleq W + WH(\lambda^o I - H^T W H)^{\dagger} H^T W, \end{aligned}$$

and the nonnegative scalar parameter λ^o *is determined from the scalar-valued optimization*

$$\lambda^o \;=\; \arg \min_{\lambda \geq \|H^T W H\|} G(\lambda),$$

where the function $G(\lambda)$ *is defined as*

$$G(\lambda) = \|x^o(\lambda)\|_Q^2 \;+\; \lambda \|E_a x^o(\lambda) - E_b\|^2 \;+\; \|Ax^o(\lambda) - b\|_{W(\lambda)}^2.$$

Here

$$\begin{aligned} W(\lambda) &\triangleq W + WH\big(\lambda I - H^T W H\big)^{\dagger} H^T W, \\ Q(\lambda) &\triangleq Q + \lambda E_a^T E_a, \end{aligned}$$

and

$$x^o(\lambda) \triangleq \left[Q(\lambda) + A^T W(\lambda) A\right]^{-1} \left[A^T W(\lambda) b + \lambda E_a^T E_b\right].$$

We thus see that the solution of (3.12) requires that we first determine an optimal nonnegative scalar parameter, λ^o, which corresponds to the minimizing argument of the function $G(\lambda)$ over the semiopen interval $[\|H^T W H\|, \infty)$. Compared with the solution (1.2) of the standard regularized least-squares problem (1.1), the expression for $\hat{x}$ in (3.13) is distinct in two important ways:

(a) First, the weighting matrices $\{Q, W\}$ need to be replaced by corrected versions $\{\widehat{Q}, \widehat{W}\}$. These corrections are defined in terms of the optimal parameter λ^o and are also dependent on the uncertainty model.

(b) Second, the right-hand side of (3.13) contains an additional term that is equal to $\lambda^o E_a^T E_b$. This means that, with λ^o given, the $\hat{x}$ in (3.13) can be interpreted as the solution to a regularized least-squares problem of the form

$$\min_x \left(\begin{bmatrix} 1 & x^T \end{bmatrix} \begin{bmatrix} \hat{\lambda}\|E_b\|^2 & -\hat{\lambda} E_b^T E_a \\ -\hat{\lambda} E_a^T E_b & \widehat{Q} \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix} + (Ax - b)^T \widehat{W} (Ax - b) \right)$$

with a *cross-coupling* term between x and unity.

The complexity of the solution in the factored uncertainty case is therefore comparable to that of a standard regularized least-squares problem with the additional task of determining the optimal scalar parameter λ^o by minimizing the cost function $G(\lambda)$ over the open interval $[\|H^T W H\|, \infty)$. As is clear from the statement of Theorem 3.1 in the general case, this function is unimodal and has a unique global minimum over the interval of interest. Therefore, the determination of λ^o can be pursued by employing standard search procedures without worries about convergence to undesired local minima.

3.6. Bounded uncertainties. Consider next the special case of section 2.1 with

$$\phi(x) = \eta\|x\| + \eta_b.$$

In this case, solving for x^o is not so immediate since (3.8) now becomes, for any nonzero x,

$$x = \left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\|x\|} I\right)\right]^{-1} D(\lambda)\ . \tag{3.14}$$

Note that x appears on both sides of the equality (except when $\eta_b = 0$, in which case the expression for x is complete in terms of $\{M(\lambda), \lambda, \eta, D(\lambda)\}$). To solve for x in the general case we let $\alpha = \|x\|$ and square the above equation to obtain the scalar equation in α:

$$\alpha^2 - D^T(\lambda)\left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\alpha}\right) I\right]^{-2} D(\lambda)\ = 0\ . \tag{3.15}$$

It is shown in Appendix B that a unique solution $\alpha^o(\lambda) > 0$ exists for this equation if and only if $\lambda\eta\eta_b\ < \|D(\lambda)\|$. Otherwise, $\alpha^o(\lambda) = 0$. In the former case, the expression for x^o, which is a function of λ, becomes

$$x^o(\lambda) = \left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\alpha^o(\lambda)}\right) I\right]^{-1} D(\lambda)\ . \tag{3.16}$$

In the latter case we clearly have $x^o(\lambda) = 0$.

Substituting the expression for $x^o(\lambda)$ into (3.16) we get

$$G(\lambda) = x^o(\lambda)^T Q x^o(\lambda) + \big(Ax^o(\lambda) - b\big)^T W(\lambda)\big(Ax^o(\lambda) - b\big) + \lambda\phi^2\big(x^o(\lambda)\big).$$

We are thus led to the following statement.

THEOREM 3.3 (bounded uncertainties). *Consider a regularized and weighted robust least-squares problem of the form*

$$\min_x \max_{\substack{\|\delta A\| \le \eta \\ \|\delta b\| \le \eta_b}} \left[x^T Q x + \Big((A+\delta A)x - (b+\delta b)\Big)^T W \Big((A+\delta A)x - (b+\delta b)\Big)\right], \tag{3.17}$$

where $\{A, b\}$ are known quantities of appropriate dimensions, $W \ge 0$ and $Q > 0$ are known weighting matrices, and the perturbations $\{\delta A, \delta b\}$ are assumed to be bounded by $\{\eta, \eta_b\}$. Then problem (3.3) *has a unique global minimum $\hat{x}$ that can be determined as follows:*

1. *Introduce the modified matrices*

$$\begin{aligned} W(\lambda) &\triangleq W + W\big(\lambda I - W\big)^{\dagger} W, \\ M(\lambda) &\triangleq Q + A^T W(\lambda) A, \\ D(\lambda) &\triangleq A^T W(\lambda) b. \end{aligned}$$

2. *For every λ, define*

$$x^o(\lambda) = \begin{cases} 0 & \text{if } \lambda\eta\eta_b < \|D(\lambda)\|, \\ \left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\alpha^o(\lambda)}\right) I\right]^{-1} D(\lambda) & \text{otherwise}, \end{cases}$$

where in the second case, $\alpha^o(\lambda)$ is the unique positive solution of the equation

$$\alpha^2 - D^T(\lambda)\left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\alpha}\right) I\right]^{-2} D(\lambda) \ = 0.$$

3. *Introduce the cost function*

$$G(\lambda) = x^o(\lambda)^T Q x^o(\lambda) + \big(Ax^o(\lambda) - b\big)^T W(\lambda)\big(Ax^o(\lambda) - b\big) + \lambda\phi^2\big(x^o(\lambda)\big),$$

where $\phi(x) = \eta\|x\| + \eta_b$.

4. *Let λ^o denote the solution of the scalar-valued minimization problem*

$$\lambda^o = \arg\min_{\lambda \geq \|W\|} \ G(\lambda).$$

5. *Then the optimum solution of* (3.3) *is*

$$\hat{x} = \begin{cases} 0 & \text{if } \lambda^o\eta\eta_b < \|D(\lambda^o)\|, \\ x^o(\lambda^o) & \text{otherwise}, \end{cases}$$

where in the second case, the solution $\hat{x}$ admits the form

$$\hat{x} = \left[\widehat{Q} + A^T\widehat{W}A\right]^{-1} A^T\widehat{W}b,$$

where the modified weighting matrices $\{\widehat{Q}, \widehat{W}\}$ are obtained from $\{Q, W\}$ via

$$\widehat{Q} \triangleq Q + \lambda^o\eta\left(\eta + \frac{\eta_b}{\alpha^o(\lambda^o)}\right) I, \quad \widehat{W} \triangleq W + W(\lambda^o I - W)^{\dagger} W.$$

Here again we find that the solution requires that we first determine an optimal nonnegative scalar parameter, λ^o, which corresponds to the minimizing argument of the corresponding function $G(\lambda)$ over the semiopen interval $[\|W\|, \infty)$. In the special case $\eta_b = 0$, we do not need to worry about determining $\alpha^o(\cdot)$ anymore since the expression for the solution $\hat{x}$ simplifies to

$$\hat{x} = \left[\widehat{Q} + A^T\widehat{W}A\right]^{-1} A^T\widehat{W}b$$

with

$$\widehat{Q} = Q + \lambda^o\eta^2 I, \quad \widehat{W} = W + W(\lambda^o I - W)^{\dagger} W,$$

and $G(\lambda)$ is now defined in terms of

$$x^o(\lambda) = \big[M(\lambda) + \lambda\eta^2 I\big]^{-1} D(\lambda).$$

4. Concluding remarks. In this paper we formulated and solved a robust optimization problem that involves a least-squares criterion with both regularization and weighting. The solution turns out to be in regularized form, albeit one that involves corrected weighting matrices. Compared with other robust solutions, the technique does not perform deregularization and, consequently, does not require existence conditions. This fact is useful for applications that involve real-time operations. In such applications, existence conditions can be a burden, since when they fail, the solution breaks down. Applications of the proposed methodology to recursive Kalman estimation, quadratic control, and data fusion problems in wireless communications appear in [5, 6, 7, 8].

Appendix A. Properties of $\lambda^o(x)$. In this appendix, we prove that a solution λ^o of (3.3) exists and is unique for every $x \in \mathbb{R}^n$. We also prove that the function $\lambda^o(x)$ is continuous, a fact that was used in section 3.4.

Before we proceed, however, we remark that the arguments are made simpler if we assume that H^TWH is a diagonal matrix. This can be done without any loss of generality by a change of variables in y. Indeed, define

$$\bar{y} = Uy, \quad \bar{H} = HU^T,$$

where U is an orthogonal matrix ($U^TU = I$) such that $U(H^TWH)U^T = \Omega = \mathrm{diag}(\omega_i)$; and note that the two sets below are equal:

$$\{\bar{y} : \ \|\bar{y}\| \le \phi(x)\} = \{y : \ \|y\| \le \phi(x)\},$$

since $\|\bar{y}\| = \|Uy\| = \|y\|$ by the orthogonality of U. In addition, $\bar{H}\bar{y} = Hy$ and $\bar{H}^TW\bar{H} = \Omega$. In the following appendices we shall therefore assume that $H^TWH = \mathrm{diag}(\omega_i)$.

A.1. Solution of (3.3). The entries of the diagonal matrix (see above) $H^TWH = \mathrm{diag}(\omega_i)$ can be ordered such that

$$\|H^TWH\| = \omega_1 = \omega_2 = \cdots = \omega_p > \omega_{p+1} \ge \cdots \ge \omega_m \ge 0, \tag{A.1}$$

where p is the multiplicity of the largest eigenvalue of H^TWH, $\omega_1 = \|H^TWH\|$.

Partition H^TWH as follows:

$$H^TWH = \Omega = \begin{bmatrix} \omega_1 I_p & 0 \\ 0 & \Omega_2 \end{bmatrix},$$

where $\Omega_2 = \mathrm{diag}(\omega_{p+1}, \ldots, \omega_m)$. Define also the vector

$$z(x) = \begin{bmatrix} z_1(x) \\ z_2(x) \end{bmatrix} = H^TW(Ax - b),$$

where $z_1(x) \in \mathbb{R}^p$ and $z_2(x) \in \mathbb{R}^{m-p}$. For every $\lambda > \omega_1$, the matrix $(\lambda I - H^TWH)$ is invertible, and we can define

$$y(\lambda, x) = \begin{bmatrix} y_1(\lambda, x) \\ y_2(\lambda, x) \end{bmatrix} = \begin{bmatrix} \frac{1}{\lambda - \omega_1} z_1(x) \\ (\lambda I_{m-p} - \Omega_2)^{-1} z_2(x) \end{bmatrix} = (\lambda I - H^TWH)^{-1} z(x)$$

with $y_1 \in \mathbb{R}^p$ and $y_2 \in \mathbb{R}^{m-p}$. We found previously that the worst-case disturbance y^o and the Lagrange multiplier λ^o must satisfy (3.3), repeated below:

$$(\lambda^o I - H^TWH)y^o = H^TW(Ax - b), \qquad \|y^o\|^2 = \phi^2(x).$$

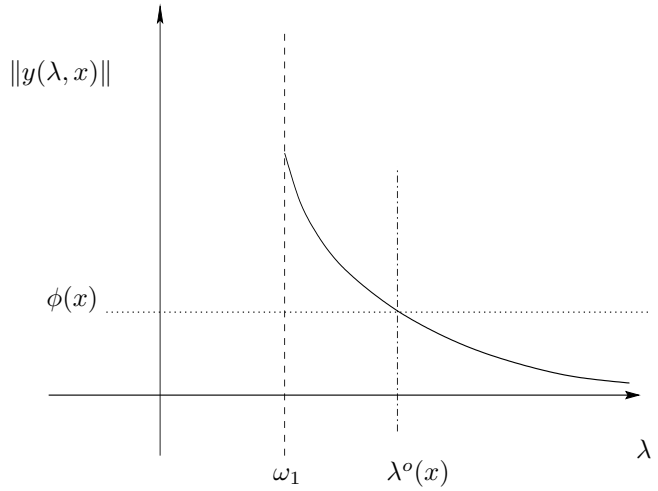


FIG. A.1. *Solution of* (3.3).

If $\{y^o, \lambda^o\}$ are such that $\lambda^o > \omega_1$, these conditions reduce to

$$\|y(\lambda^o, x)\| = \phi(x). \tag{A.2}$$

We now study the behavior of $\|y(\lambda, x)\|^2$ to find when there is a $\lambda^o > \omega_1$ satisfying the above condition. Note that, for fixed x, $\|y(\lambda, x)\|^2$ is a differentiable function of λ, with

$$\begin{aligned}\frac{d\|y(\lambda,x)\|^2}{d\lambda} &= z(x)^T \left(\frac{d}{d\lambda} \begin{bmatrix} \frac{1}{(\lambda-\omega_1)^2} & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \frac{1}{(\lambda-\omega_m)^2} \end{bmatrix} \right) z(x) \\ &= z(x)^T \begin{bmatrix} -\frac{2}{(\lambda-\omega_1)^3} I_p & 0 \\ 0 & -2(\lambda I_{m-p} - \Omega_2)^{-3} \end{bmatrix} z(x).\end{aligned}$$

The derivative is therefore negative for $z(x) \neq 0$, since the above matrix is negative-definite when $\lambda > \omega_1$. Note that when $z(x) = 0$, $y(\lambda, x) = 0$ for all $\lambda > \omega_1$. We later show that in this case the solution will be $\lambda^o = \omega_1$.

FACT 1. *We conclude that for $\lambda > \omega_1$, $\|y(\lambda, x)\|^2$ is a strictly decreasing, continuous function of λ (except when $z(x) = 0$). Therefore, the solution to* (A.2)*, if it exists, is unique (see Figure* A.1*).*

Consider now the following cases:

1. $z_1(x) \neq 0$ (in this case, $\lim_{\lambda \downarrow \omega_1} \|y(\lambda, x)\| = \infty$);
2. $z_1(x) = 0$, but $\|y_2(\omega_1, x)\| > \phi(x)$ (in this case, $\lim_{\lambda \downarrow \omega_1} \|y(\lambda, x)\| > \phi(x)$);
3. $z_1(x) = 0$ and $\|y_2(\omega_1, x)\| \leq \phi(x)$.

FACT 2. *In all cases, the limit of $\|y(\lambda, x)\|$ as λ goes to infinity is zero. This observation and Fact* 1 *imply that* (A.2) *will have a solution $\lambda^o > \omega_1$ if and only if*

$$\lim_{\lambda \downarrow \omega_1} \|y(\lambda, x)\| > \phi(x),$$

which is the situation in cases 1 *and* 2*. We refer to a point $x \in \mathbb{R}^n$ for which $\lambda^o(x) > \omega_1$ as a* regular *point. A point x satisfying the conditions in case* 3 *will be called a* singular *point.*

We argue now that if x is a singular point (i.e., if case 3 happens), the corresponding Lagrange multiplier must be $\lambda^o(x) = \omega_1$. In case 3, $\|y_2(\omega_1, x)\| \leq \phi(x)$, and condition (A.2) will not be satisfied even in the limit as $\lambda \to \omega_1$. The *original* condition (3.3) can still be satisfied, however, as we show next.

Assume that the conditions in case 3 hold, and choose $\lambda = \omega_1$. Then condition (3.3) reads

$$\begin{bmatrix} 0 & 0 \\ 0 & \omega_1 I_{m-p} - \Omega_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ z_2(x) \end{bmatrix}, \qquad \|y\|^2 = \|y_1\|^2 + \|y_2\|^2 = \phi^2(x).$$

The first condition is satisfied for

$$y_2 = (\omega_1 I_{m-p} - \Omega_2)^{-1} z_2(x),$$

where the inverse exists since by definition $\omega_1 > \omega_{p+1} = \|\Omega_2\|$. The second condition in case 3 is $\|y_2\| \leq \phi(x)$. Therefore, to satisfy the norm condition in (3.3), we just choose any $y_1 \in \mathbb{R}^p$ whose norm satisfies

$$\|y_1\|^2 = \phi^2(x) - \left\| (\omega_1 I_{m-p} - \Omega_2)^{-1} z_2(x) \right\|^2.$$

LEMMA A.1 (solution to the maximization problem). *If a point $x \in \mathbb{R}^n$ is regular, viz.,*

$$z_1(x) \neq 0 \qquad \text{and} \qquad \lim_{\lambda \downarrow \omega_1} \|y(\lambda, x)\| > \phi(x), \tag{A.3}$$

then the Lagrange multiplier at the maximum $\lambda^o(x) > \omega_1$ is the unique solution to (A.2). *(Although the first condition implies the second, we want to state it explicitly here for further reference.) In this case, the worst-case disturbance*

$$y^o = y(\lambda^o, x) = (\lambda^o I - H^T W H)^{-1} H^T W (Ax - b), \qquad \|y^o\| = \phi(x)$$

is also unique.

On the other hand, if x is singular, viz.,

$$z_1(x) = 0 \qquad \text{and} \qquad \lim_{\lambda \downarrow \omega_1} \|y(\lambda, x)\| \leq \phi(x), \tag{A.4}$$

then the Lagrange multiplier will be $\lambda^o(x) = \omega_1$. Now the worst-case disturbance is no longer unique—any disturbance of the form below will achieve the maximum:

$$y^o = \begin{bmatrix} y_1^o \\ y_2^o \end{bmatrix},$$

with $y_1^o \in \mathbb{R}^p$, and

$$y_2^o = (\omega_1 I_{m-p} - \Omega_2)^{-1} z_2(x), \qquad \|y_1^o\|^2 = \phi^2(x) - \|y_2^o\|^2.$$

In addition, using the pseudo-inverse notation, we can write

$$y^o = (\omega_1 I - H^T W H)^{\dagger} H^T W (Ax - b) + \begin{bmatrix} y_1^o \\ 0 \end{bmatrix},$$

where

$$\|y_1^o\|^2 = \phi^2(x) - \left\| (\omega_1 I - H^T W H)^{\dagger} H^T W (Ax - b) \right\|^2.$$

A.2. Continuity of $\lambda^o(x)$. This property of $\lambda^o(x)$ was invoked in section 3.4 to argue that $G(\lambda)$ is unimodal (see Theorem 3.1). We again treat regular and singular points separately.

Regular points. By definition, at a regular point $\tilde{x}$, $\lambda^o(\tilde{x}) > \omega_1$ and

$$f(\lambda^o, \tilde{x}) \triangleq \phi^2(\tilde{x}) - (A\tilde{x} - b)^T W H(\lambda^o(\tilde{x})I - H^T W H)^{-2} H^T W (A\tilde{x} - b) = 0.$$

Now from the implicit function theorem [17], the function $\lambda^o(x)$ defined by the above condition is continuous at a given point x if the gradient $\nabla_\lambda f(\lambda, x)$ is nonzero at $\lambda = \lambda^o$. To check this condition, compute the partial derivative

$$\frac{\partial f(\lambda, x)}{\partial \lambda} = 2(Ax - b)^T W H(\lambda I - H^T W H)^{-3} H^T W (Ax - b).$$

At a regular point $\tilde{x}$, recall that we must have either $z_1(\tilde{x}) \neq 0$ or $\|y_2(\omega_1, x)\| > \phi(\tilde{x})$ (see (A.3)). Both these conditions would be violated if $A\tilde{x} - b = 0$, so our assumption that $\tilde{x}$ is regular implies that $A\tilde{x} - b \neq 0$. With this fact, and noting that $(\lambda^o(\tilde{x})I - H^T W H)^{-3} > 0$ (from the regularity of $\tilde{x}$), we conclude that

$$\frac{\partial f(\lambda^o(x), x)}{\partial \lambda} > 0,$$

satisfying the condition of the implicit function theorem. We have thus proved that $\lambda^o(x)$ is continuous at any regular point $\tilde{x}$.

Singular points. Now let $\bar{x} \in \mathbb{R}^n$ be a singular point. We prove the continuity of $\lambda^o(\cdot)$ at $x = \bar{x}$ from the definition. Given an $\epsilon > 0$, we shall find $\delta(\epsilon) > 0$ such that $(\lambda^o(\bar{x}) = \omega_1)$

$$\|x - \bar{x}\| < \delta \Rightarrow |\lambda^o(x) - \omega_1| < \epsilon.$$

To find such a δ, we shall need some properties of singular points and of $\phi^2(x)$ and $z(x)$—if $\bar{x}$ is a singular point, then from the previous sections we have the following:

1. $\phi(\bar{x}) \geq \|(\omega_1 I - H^T W H)^\dagger H^T W (A\bar{x} - b)\|$;
2. $z_1(\bar{x}) = 0$;
3. $\|y_2(\lambda, x)\|^2$ is continuous in (λ, x) on $(\omega_1, \infty) \times \mathbb{R}^n$, and continuous and strictly decreasing in $\lambda > \omega_1$ for fixed x.

Recalling that $z(x) = H^T W(Ax - b)$, we also have the following:

4. $\|z_1(x)\|^2$ and $\|z_2(x)\|^2$ are continuous functions for all $x \in \mathbb{R}^n$.

Finally, we must make an assumption on the uncertainty bound, namely, we assume that

5. $\phi^2(x)$ is continuous for all $x \in \mathbb{R}^n$ (in fact, this follows from our assumption in Theorem 3.1 that $\phi^2(x)$ is convex).

Two situations may occur:

A. There exists a neighborhood $N(\bar{x})$ whose points are all singular, i.e., $x \in N(\bar{x}) \Rightarrow \lambda^o(x) = \omega_1$. In this situation, the continuity of $\lambda^o(\cdot)$ at $\bar{x}$ is trivial.

B. Every neighborhood of $\bar{x}$ contains a regular point x^*.

Let us consider the second case. We now find a ball $B_\delta(\bar{x}) = \{x : \|x - \bar{x}\| < \delta\}$ for which

$$\sup_{x \in B_\delta(\bar{x})} \lambda^o(x) < \lambda^o(\bar{x}) + \epsilon = \omega_1 + \epsilon.$$

The above properties and assumptions imply that for any K_1, K_2, and K_3, it is possible to find δ_1, δ_2, and $\delta_3 > 0$ such that

$$\begin{aligned}
&\|x-\bar{x}\| < \delta_1 \Rightarrow \left| \|z_1(x)\|^2 - \underbrace{\|z_1(\bar{x})\|^2}_{=0} \right| < \frac{\epsilon}{K_1}, \\
&\|x-\bar{x}\| < \delta_2 \Rightarrow \|z_2(x) - z_2(\bar{x})\|^2 < \frac{\epsilon}{K_2}, \\
&\|x-\bar{x}\| < \delta_3 \Rightarrow \left|\phi^2(x) - \phi^2(\bar{x})\right| < \frac{\epsilon}{K_3}.
\end{aligned} \tag{A.5}$$

Choose the K_i such that

$$K_1 = \frac{\bar{K}_1}{\epsilon^2}, \qquad \frac{1}{\bar{K}_1} + \frac{1}{(\omega_1 - \omega_{p+1})^2 K_2} + \frac{1}{K_3} < \frac{\|y_2(\omega_1, \bar{x})\|^2}{2(\omega_1 - \omega_{p+1})} \tag{A.6}$$

and let $\delta = \min\{\delta_1, \delta_2, \delta_3\}$. Remark that the K_i cannot be chosen to satisfy (A.6) only if $y_2(\omega_1, \bar{x}) = 0$. We shall assume for now that $y_2(\omega_1, \bar{x}) \neq 0$ and treat the other case later.

Since we are studying case B, let x^* be a regular point in $B_\delta(\bar{x})$. As a regular point, x^* satisfies $\lambda^o(x^*) > \omega_1$ and

$$\|y(\lambda^o(x^*), x^*)\| = \phi(x^*) \qquad \text{and} \qquad \lim_{\lambda \downarrow \omega_1} \|y(\lambda, x^*)\| > \phi(x^*)$$

(the limit may be infinity).

We now show that for $\lambda^* = \omega_1 + \epsilon$, it necessarily holds that

$$\|y(\lambda^*, x^*)\| < \phi(x^*),$$

which means that $\omega_1 < \lambda^o(x^*) < \omega_1 + \epsilon$, which is our desired result. Let us then evaluate $\|y(\lambda, x^*)\|^2$:

$$\begin{aligned}
\|y(\lambda, x^*)\|^2 &= \left\| (\lambda I - H^T W H)^{-1} \begin{bmatrix} z_1(x^*) \\ z_2(x^*) \end{bmatrix} \right\|^2 \\
&= (\lambda - \omega_1)^{-2} \|z_1(x^*)\|^2 + \left\| \begin{bmatrix} \lambda - \omega_{p+1} & \dots & 0 \\ & \ddots & \\ 0 & \dots & \lambda - \omega_m \end{bmatrix}^{-1} z_2(x^*) \right\|^2.
\end{aligned}$$

We use (A.5) to bound these norms,

$$\|y(\lambda, x^*)\|^2 < \frac{\epsilon}{(\lambda - \omega_1)^2 K_1} + \|y_2(\lambda, \bar{x})\|^2 + \frac{\epsilon}{(\lambda - \omega_{p+1})^2 K_2}, \tag{A.7}$$

where we used $\| \operatorname{diag}\big((\lambda - \omega_j)^{-1}\big) \| = (\lambda - \omega_{p+1})^{-1}$ and $z_2(x^*) = z_2(\bar{x}) + \big(z_2(x^*) - z_2(\bar{x})\big)$.

To bound the second term, we write

$$y_2(\lambda,\bar{x}) = \begin{bmatrix} \frac{\omega_1-\omega_{p+1}}{\lambda-\omega_{p+1}} & & \\ & \ddots & \\ & & \frac{\omega_1-\omega_m}{\lambda-\omega_{p+1}} \end{bmatrix} \begin{bmatrix} \omega_1-\omega_{p+1} & & \\ & \ddots & \\ & & \omega_1-\omega_m \end{bmatrix}^{-1} z_2(\bar{x})$$
$$= \begin{bmatrix} \frac{\omega_1-\omega_{p+1}}{\lambda-\omega_{p+1}} & & \\ & \ddots & \\ & & \frac{\omega_1-\omega_m}{\lambda-\omega_{p+1}} \end{bmatrix} y_2(\omega_1,\bar{x}) \stackrel{\Delta}{=} P(\lambda) y_2(\omega_1,\bar{x}).$$

Let $\lambda = \omega_1 + \epsilon$; then the largest element of $P(\lambda)$ will be (if ϵ is small enough)

$$\frac{\omega_1-\omega_{p+1}}{\omega_1+\epsilon-\omega_{p+1}} = 1 - \frac{\epsilon}{\omega_1-\omega_{p+1}} + \frac{\epsilon^2}{(\omega_1-\omega_{p+1})^2} - \underbrace{\left(\frac{\epsilon^3}{(\omega_1-\omega_{p+1})^3}+\cdots\right)}_{\geq 0 \text{ if } \epsilon/(\omega_1-\omega_{p+1})<1/2}$$
$$< 1 - \frac{\epsilon}{\omega_1-\omega_{p+1}} + \frac{\epsilon^2}{(\omega_1-\omega_{p+1})^2} < 1 - \frac{\epsilon}{2(\omega_1-\omega_{p+1})}$$

and thus,

$$\|y_2(\lambda,\bar{x})\|^2 < \left(1 - \frac{\epsilon}{2(\omega_1-\omega_{p+1})}\right)\|y_2(\omega_1,\bar{x})\|^2$$
$$< \phi^2(\bar{x}) - \frac{\epsilon}{2(\omega_1-\omega_{p+1})}\|y_2(\omega_1,\bar{x})\|^2$$
$$< \phi^2(x^*) + \frac{\epsilon}{K_3} - \frac{\epsilon}{2(\omega_1-\omega_{p+1})}\|y_2(\omega_1,\bar{x})\|^2.$$

Using this bound in (A.7), we obtain

$$\|y(\omega_1+\epsilon,x^*)\|^2 < \frac{\epsilon}{\epsilon^2 K_1} + \frac{\epsilon}{(\omega_1+\epsilon-\omega_{p+1})^2 K_2} + \phi^2(x^*)$$
$$+ \frac{\epsilon}{K_3} - \frac{\epsilon}{2(\omega_1-\omega_{p+1})}\|y_2(\omega_1,\bar{x})\|^2 < \phi^2(x^*),$$

where the last inequality follows from our choice of the K_i in (A.6). The inequality shows that $\lambda^o(x^*) < \omega_1 + \epsilon$. Since the above argument holds for any regular point in $B_\delta(\bar{x})$, we have

$$x \in B_\delta(\bar{x}) \Rightarrow \omega_1 \leq \lambda^o(x) < \omega_1 + \epsilon,$$

which proves the continuity of $\lambda^o(\cdot)$ at singular points $\bar{x}$ for which $y_2(\omega_1,\bar{x}) \neq 0$.

Finally we consider singular points $\bar{x}$ for which $y_2(\omega_1,\bar{x})$ is zero. In this case, $A\bar{x} - b$ is necessarily zero (since $z_1(\bar{x}) = 0$ for singular points). Again, two situations may happen:

(i) $\phi(\bar{x}) = 0$. In this situation, the solution of the maximization problem is trivial, as the uncertainty will be identically zero.

(ii) $\phi(\bar{x}) > 0$. Now, from the continuity of $\|y_2(\lambda,\bar{x})\|^2$ and of $\phi^2(x)$, there exists a ball $B_{\delta_4}(\bar{x})$ such that

$$x \in B_{\delta_4}(\bar{x}) \Rightarrow \|y_2(\omega_1,x)\|^2 < \frac{\phi^2(\bar{x})}{2}.$$

With this inequality, if we choose K_1 and K_3 such that

$$K_1 = \frac{\bar{K}_1}{\epsilon^2}, \qquad \frac{1}{\bar{K}_1} + \frac{1}{2K_3} < \frac{\phi^2(\bar{x})}{2} - \frac{1}{2K_3},$$

then if $\delta = \min\{\delta_1, \delta_3, \delta_4\}$, for all regular points $x^* \in B_\delta(\bar{x})$, we can replace (A.7) by the simpler expression

$$\|y(\lambda, x^*)\|^2 < \frac{\epsilon}{(\lambda - \omega_1)^2 K_1} + \|y_2(\lambda, x^*)\|^2 < \frac{\epsilon}{(\lambda - \omega_1)^2 K_1} + \frac{\phi^2(\bar{x})}{2},$$

where we used the fact that $\|y_2(\lambda, x^*)\|^2$ is decreasing with λ. With our choice of K_1, for $\lambda = \omega_1 + \epsilon$ we obtain

$$\begin{aligned}\|y(\omega_1 + \epsilon, x^*)\|^2 &< \frac{\epsilon}{\bar{K}_1} + \frac{\phi^2(\bar{x})}{2} < \frac{\epsilon}{\bar{K}_1} + \frac{\epsilon}{2K_3} + \frac{\phi^2(x^*)}{2} \\ &< \frac{\phi^2(\bar{x})}{2} - \frac{\epsilon}{2K_3} + \frac{\phi^2(x^*)}{2} < \phi^2(x^*),\end{aligned}$$

which implies that $\lambda^o(x^*) < \omega_1 + \epsilon$.

Appendix B. Computation of $x^o(\lambda)$ in the bounded uncertainty case. With $\phi(x) = \eta\|x\| + \eta_b$, the vector x that achieves the minimum on the right-hand side of (3.7) is the solution to the equation

$$x \triangleq x^o(\lambda) = \left(Q + A^T W(\lambda) A + \lambda\eta^2 I + \frac{\lambda\eta\eta_b}{\|x\|} I\right)^{-1} A^T W(\lambda) b. \tag{B.1}$$

The value of x is clearly a function of λ. Observe, however, that this equation defines x implicitly since x appears on both sides of the equality. To proceed, we consider two cases.

Case 1: $\eta_b = 0$. In this case, the expression for $x^o(\lambda)$ collapses to

$$x^o(\lambda) = \left[Q + \lambda\eta^2 I + A^T W(\lambda) A\right]^{-1} A^T W(\lambda) b.$$

That is, the term $\|x\|$ disappears from the right-hand side of (B.1). Consequently, this expression defines $x^o(\lambda)$ explicitly.

Case 2: $\eta_b \neq 0$. In this case, the term $\|x\|$ does not disappear from the right-hand side of (B.1). In order to solve for x we proceed as follows. First, we introduce the scalar $\alpha = \|x\|$ and square both sides of (B.1). This leads to the following nonlinear equation in α:

$$\alpha^2 - D^T(\lambda)\left[M(\lambda) + \lambda\eta\left(\eta + \frac{\eta_b}{\alpha}\right) I\right]^{-2} D(\lambda) \ = 0, \tag{B.2}$$

where

$$M(\lambda) = Q + A^T W(\lambda) A, \quad D(\lambda) = A^T W(\lambda) b.$$

The value of α is again dependent on λ. The following result indicates that the solution to the above nonlinear equation for α is either at $\alpha = 0$ or at a *unique* positive value.

LEMMA B.1. *Let $x^o(\lambda)$ minimize the inner cost on the right-hand side of* (3.7)*. If and only if $\lambda\eta\eta_b < \|D(\lambda)\|$, the norm $\|x^o(\lambda)\|$ is equal to the unique positive solution*

of equation (B.2), $\alpha^o(\lambda)$. *Otherwise, the solution to the minimization problem is* $x^o(\lambda) = 0$, *i.e.,* $\alpha^o(\lambda) = 0$.

Proof. We shall first find the solutions of (B.2) when $\lambda\eta\eta_b < \|D(\lambda)\|$; afterwards we relate these conditions to the solutions of the inner minimization problem on the right-hand side of (3.7).

Introduce the SVD of the symmetric positive-definite matrix $M(\lambda)$, say $M(\lambda) = U\Sigma U^T$, where $\{U, \Sigma\}$ are also dependent on λ. We denote the entries of Σ by $\{\sigma_i\}$. Substituting this decomposition into the left side of (B.2), it reduces to the function

$$f(\alpha) \triangleq \alpha^2 - \sum_{i=1}^{n} \frac{\bar{d}_i^2}{[\sigma_i + \lambda\eta(\eta + \frac{\eta_b}{\alpha})]^2},$$

where $\{\bar{d}_i\}$ denotes the entries of the transformed vector $U^T D(\lambda)$. We are seeking the roots of $f(\alpha)$.

Note that $\alpha = 0$ is always a solution if $\eta_b > 0$. Let us search for a solution $\alpha > 0$. Assuming $\alpha > 0$, we can write

$$f(\alpha) = \alpha^2 \left[1 - \sum_{i=1}^{n} \frac{\bar{d}_i^2}{(\sigma_i\alpha + \lambda\eta^2\alpha + \lambda\eta\eta_b)^2}\right] \triangleq \alpha^2 g(\alpha),$$

where we introduced the function $g(\alpha)$. Taking the limits as $\alpha \to 0$ and $\alpha \to \infty$ we find that

$$\lim_{\alpha\to 0} g(\alpha) = 1 - \sum_{i=1}^{n} \frac{\bar{d}_i^2}{(\lambda\eta\eta_b)^2},$$

$$\lim_{\alpha\to\infty} g(\alpha) = 1 > 0.$$

Therefore, $g(\alpha)$ will have a zero for $\alpha > 0$ if and only if the first limit above is negative, i.e., if $\{\lambda, \eta, \eta_b\}$ satisfy

$$\lambda\eta\eta_b < \|D(\lambda)\|.$$

In addition, since the derivative of $g(\alpha)$ with respect to α is

$$\frac{dg(\alpha)}{d\alpha} = 2\sum_{i=1}^{n} \frac{\bar{d}_i^2(\lambda\eta^2 + \sigma_i)}{[\alpha\sigma_i + \lambda\eta(\alpha\eta + \eta_b)]^3} > 0,$$

we conclude that the root is necessarily unique.

Let us verify now that this root really corresponds to the solution of our minimization problem. The point that may cause trouble is $x = 0$, where the cost function is not differentiable. The cost at this point is

$$C(0, \lambda) = b^T W(\lambda) b + \lambda\eta_b^2.$$

If we move a little away from $x = 0$, say to $x = \delta x$, then the cost becomes

$$\begin{aligned} C(\delta x, \lambda) &= (A\delta x - b)^T W(\lambda)(A\delta x - b) + \lambda(\eta\|\delta x\| + \eta_b)^2 \\ &= b^T W(\lambda) b - 2\delta x^T A^T W(\lambda) b + \delta x^T A^T W(\lambda) A\delta x + \lambda\eta^2\|\delta x\|^2 \\ &\quad + 2\lambda\eta\eta_b\|\delta x\| + \lambda\eta_b^2 \end{aligned}$$

and thus, for small δx,

$$\begin{aligned} C(\delta x, \lambda) &= C(0,\lambda) - 2\delta x^T D(\lambda) + 2\lambda\eta\eta_b\|\delta x\| + O(\|\delta x\|^2) \\ &\geq C(0,\lambda) - 2\|\delta x\|\Big(\|D(\lambda)\| - 2\lambda\eta\eta_b\Big) + O(\|\delta x\|^2). \end{aligned}$$

We conclude that, for small δx, $C(\delta x, \lambda)$ is smaller than $C(0,\lambda)$ if and only if $\lambda\eta\eta_b < \|D(\lambda)\|$. In this situation, $x = 0$ cannot be a minimum of $C(x,\lambda)$, and the optimum $x^o(\lambda)$ must be such that its norm solves (B.2) with $\alpha^o(\lambda) > 0$.

On the other hand, if $\lambda\eta\eta_b \geq \|D(\lambda)\|$, the cost for small δx satisfies $C(\delta x, \lambda) > C(0,\lambda)$. (We can include the case when equality holds, since the terms in $O(\|\delta x\|^2)$ above are all positive.) The point $x = 0$ must thus be a local minimum to $C(x,\lambda)$. Since we know that this cost is strictly convex in x for fixed λ, $x = 0$ must be the global minimum. ∎

Appendix C. A result on convex optimization problems. In this appendix we establish a result that was used to show that $G(\lambda)$ is unimodal. Let $f(x,y)$ be a real function of variables $x \in X,\ y \in Y$. We shall study the problem

$$\min_{x\in X, y\in Y} f(x,y).$$

Define the functions

$$\begin{aligned} g:\ & X \to \mathbb{R}, \\ g(x) &= \min_{y\in Y} f(x,y) \end{aligned}$$

and

$$\begin{aligned} h:\ & Y \to \mathbb{R}, \\ h(y) &= \min_{x\in X} f(x,y). \end{aligned}$$

We denote by (x_{op}, y_{op}) one of the (possibly many) global minimum points of $f(x,y)$ in $X \times Y$, by x_g one of the global minima of $g(x)$ in X, and by y_h one of the global minima of $h(y)$ in Y. With these definitions, we prove the following results.

LEMMA C.1. *If any of the minima below is attainable, then it holds that*

$$\min_{(x,y)\in X\times Y} f(x,y) = \min_{x\in X} g(x) = \min_{y\in Y} h(y) \quad \textit{and} \quad (x_{op}, y_{op}) = (x_g, y_{x_g}) = (x_{y_h}, y_h).$$

Proof. This is a classic result. To prove it, simply notice that all points (x,y) are compared in the minimization of all three functions above. If the minima are not attainable, the result is still true if we substitute the min by inf. ∎

LEMMA C.2. *Let X, Y be subsets of a metric space and assume that the functions*

$$\begin{aligned} f(\bar{x}, y):\ & Y \to \mathbb{R} \qquad \textit{for all } \bar{x} \in X \textit{ fixed}, \\ f(x, \bar{y}):\ & X \to \mathbb{R} \qquad \textit{for all } \bar{y} \in Y \textit{ fixed}, \\ g(x):\ & X \to \mathbb{R} \end{aligned}$$

have unique global minima and are unimodal in their respective domains; i.e., assume that each function does not admit local minima different than their global minima.

We now define the functions

$$y_m : \ X \to Y,$$
$$y_m(x) = \arg\min_{y \in Y} f(x, y)$$

and

$$x_m : \ Y \to X,$$
$$x_m(y) = \arg\min_{x \in X} f(x, y).$$

Under these conditions, if $y_m(x)$ is continuous in X, then $h(y)$ is also unimodal.

Proof. $y_m(x)$ is a function, since, by hypothesis, $f(\bar{x}, y)$, $\bar{x}$ fixed, is unimodal in Y. A similar argument implies that $x_m(y)$ is a function. Now assume (by contradiction) that $h(y)$ is not unimodal, i.e., it has a local minimum at $y_l \neq y_{op}$. This means that there is an open ball $B_\delta(y_l) \in Y$ such that y_l is the global minimum of $h(y)$ *inside the ball.*

From the previous lemma, we find that

$$\min_{(x,y) \in X \times B_\delta(y_l)} f(x, y) = \min_{y \in B_\delta(y_l)} h(y)$$

and $(x_l, y_l) = (x_m(y_l), y_l)$. This implies that (x_l, y_l) is a local minimum of $f(x, y)$ in $X \times Y$ different from the global minimum (x_{op}, y_{op}). In particular, this means that, fixing x_l, $f(x_l, y)$ has a (local) minimum at $y = y_l$.

Since we assumed that $f(x_l, y)$ is unimodal, it must hold that $y_m(x_l) = y_l$. Function $y_m(\cdot)$ is continuous on X by hypothesis, thus there exists a ball $B_\gamma(x_l)$ whose points satisfy

$$x \in B_\gamma(x_l) \Rightarrow y_m(x) \in B_\delta(y_l).$$

Since (x_l, y_l) is the global minimum of f in $X \times B_\delta(y_l)$, x_l must be the global minimum of $g(x) = f(x, y_m(x))$ in $B_\gamma(x_l)$, that is, x_l is a local minimum of $g(x)$.

Finally, note that we assumed that $y_l \neq y_{op}$. Since $y_m(x_l) = y_l$ and $y_m(x_{op}) = y_{op}$, we must have $x_l \neq x_{op}$—that is, we found a local minimum of $g(x)$ different than x_{op}, contradicting our initial assumption that $g(x)$ is unimodal. ∎

REFERENCES

[1] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.

[2] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control: A Unified Approach to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Theories*, SIAM, Philadelphia, 1999.

[3] L. E. Ghaoui and H. Lebret, *Robust solutions to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064

[4] 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.

[5] V. H. Nascimento and A. H. Sayed, *Optimal state regulation for uncertain state-space models*, in Proceedings of the American Control Conference, Vol. 1, San Diego, CA, 1999, pp. 419–424.

[6] A. H. Sayed, *A framework for state-space estimation with uncertain models*, IEEE Trans. Automat. Control, 46 (2001), pp. 998–1013.

[7] A. H. Sayed and A. Subramanian, *State-space estimation with uncertain models*, in Total Least Squares and Errors-in-Variables Modeling, III: Analysis, Algorithms and Applications, S. Van Huffel and P. Lemmerling, eds., Kluwer Academic, Dordrecht, The Netherlands, 2002.

[8] A. H. Sayed, T. Y. Al-Naffouri, and T. Kailath, *Robust estimation for uncertain models in a data fusion scenario*, in System Identification, Vol. 3, R. Smith, ed., Pergamon, Oxford, 2001, pp. 899–904.

[9] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Academic Press, London, 1982.

[10] A. H. Sayed and V. H. Nascimento, *Design criteria for uncertain models with structured and unstructured uncertainties*, in Robustness in Identification and Control, A. Garulli, A. Tesi, and A. Vicino, eds., Proceedings of the Workshop on Robust Identification and Control, Siena, Italy, 1998, Vol. 245, Springer-Verlag, London, 1999, pp. 159–173.

[11] A. H. Sayed, V. H. Nascimento, and S. Chandrasekaran, *Estimation and control with bounded data uncertainties*, Linear Algebra Appl., 284 (1998), pp. 259–306.

[12] A. H. Sayed and S. Chandrasekaran, *Parameter estimation with multiple sources and levels of uncertainties*, IEEE Trans. Signal Process., 48 (2000), pp. 680–692.

[13] Y. Cheng and B. L. de Moor, *Robustness analysis and control system design for hydraulic servo system*, IEEE Trans. Control Sys. Technol., 2 (1994), pp. 183–198.

[14] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice-Hall, Englewood Cliffs, NJ, 2000.

[15] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, Blaisdell, Waltham, MA, 1969.

[16] R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, UK, 1987.

[17] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., McGraw-Hill, New York, 1976.

[18] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1143–1170
© 2002 Society for Industrial and Applied Mathematics

NUMERICALLY RELIABLE COMPUTING FOR THE ROW BY ROW DECOUPLING PROBLEM WITH STABILITY*

DELIN CHU† AND ROGER C. E. TAN†

Abstract. This is the first of two papers on the row by row decoupling problem and the triangular decoupling problem. In this paper we study the row by row decoupling problem with stability in control theory. We first prove a nice reduction property for the row by row decoupling problem with stability and then develop a numerically reliable method for solving it. The basis of our main results is some condensed forms under orthogonal transformations, which can be implemented in numerically stable ways. Hence our results lead to numerically reliable methods for solving the studied problem using existing numerical linear algebra software such as MATLAB.

In the sequel [*SIAM J. Matrix Anal. Appl.*, 23 (2002), pp. 1171–1182], we will consider a related problem—the triangular decoupling problem—and parameterize all solutions for it.

Key words. row by row decoupling, stability, orthogonal transformation, reliable computing

AMS subject classifications. 93B05, 93B40, 93B52, 65F35

PII. S0895479801362546

1. Introduction.

In this paper we study the numerical computation of the row by row decoupling problem with stability for a system of the form

$$E\dot{x} = Ax + Bu, \quad y = Cx, \tag{1.1}$$

where $E, A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{m\times n}$, E is nonsingular, $x \in \mathbf{R}^n$ is the state, $u \in \mathbf{R}^m$ is the control input, and $y \in \mathbf{R}^m$ is the output. Unless otherwise stated, we shall denote the ith row of C by c_i. We shall also assume, without loss of generality, that $(E^{-1}A, E^{-1}B)$ is controllable, i.e.,

$$\operatorname{rank}\begin{bmatrix} sE - A & B \end{bmatrix} = \operatorname{rank}\begin{bmatrix} sI - E^{-1}A & E^{-1}B \end{bmatrix} = n \qquad \forall s \in \mathbf{C}.$$

If this is not the case, then we can always take only its controllable part, which can be obtained by numerically stable algorithms, for example, the "controllability algorithm" in [15], since the noncontrollable part does not contribute to the transfer matrix of the closed-loop system (1.4) below for any feedback $u = Fx + Hv$. Hence, if the noncontrollable part is not stable (the system is not stabilizable), the row by row decoupling problem and the triangular decoupling problem with stability, defined below, are both unsolvable; otherwise, there is no necessity to consider it.

Theoretically, (1.1) is equivalent to

$$\dot{x} = E^{-1}Ax + E^{-1}Bu, \quad y = Cx, \tag{1.2}$$

because E is nonsingular. However, E may be highly ill-conditioned and the explicit computation of E^{-1} may be numerically unstable, which is well known [11]. Thus, from a numerical computation point of view, the transformation from system (1.1)

*Received by the editors April 2, 2001; accepted for publication (in revised form) by D. Boley December 14, 2001; published electronically May 10, 2002. This research was supported by research grants from NUS under R-146-000-016-112 and the CIM-IHPC joint research project.

http://www.siam.org/journals/simax/23-4/36254.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@math.nus.edu.sg, mattance@math.nus.edu.sg).

to system (1.2) should be avoided [11] and we would have to work directly with the system (1.1).

In this paper we will focus our attention on systems of the form (1.1) with E nonsingular. The reasons are that (i) if $E = I$, then system (1.1) is the standard one studied in most of the existing literature; (ii) we have shown in [4] that the row by row decoupling problem for descriptor systems can be reduced by orthogonal transformations into the same one for a lower-dimensional system of the form (1.1) with E nonsingular.

If we apply the state feedback of the form

$$u = Fx + Hv \tag{1.3}$$

to (1.1), then the closed-loop system becomes

$$E\dot{x} = (A + BF)x + BHv, \quad y = Cx. \tag{1.4}$$

The transfer matrix from output y to input v in (1.4) is $C(sE-A-BF)^{-1}BH$. Hence, the row by row decoupling problem with stability can be formulated mathematically as follows.

DEFINITION 1.1. *Given a system of the form* (1.1)*, the row by row decoupling problem with stability (RRDPS) for system* (1.1) *is solvable if there exist matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *with* H *nonsingular such that*

$$C(sE - A - BF)^{-1}BH \text{ is nonsingular and diagonal}, \tag{1.5}$$

and $E^{-1}(A + BF)$ *is stable; i.e., all eigenvalues of* $E^{-1}(A + BF)$ *are in the open left-half complex plane.*

The row by row decoupling problem plays a central role in classical as well as modern control theory. It has been investigated extensively in the last three decades by the geometric approach and the structural approach (see [1, 2, 7, 10, 12, 13, 14, 20, 21]). The geometric solutions and structural solutions given in these papers are expressed in terms of some invariant subspaces and the structure at infinity of system (1.1), respectively.

In the following we shall introduce some existing results for the RRDPS to motivate as well as provide the necessary background for the present paper.

Let $\mathbf{C}^-$ and $\sigma(M)$ denote the open left-half complex plane and the spectrum of the square matrix M, respectively. Given matrices $A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}$, a subspace $\mathcal{V} \subset \mathbf{R}^n$ is said to be (A, B)-invariant if there exists an $F \in \mathbf{R}^{m\times n}$ such that $(A + BF)\mathcal{V} \subset \mathcal{V}$. If $\mathcal{V}$ is an (A, B)-invariant subspace, $\mathcal{F}(A, B, \mathcal{V})$ denotes the class of all $F \in \mathbf{R}^{m\times n}$ satisfying $(A + BF)\mathcal{V} \subset \mathcal{V}$. In that case, $A + BF|\mathcal{V}$ denotes the double restriction of $A + BF$ to $\mathcal{V}$. An (A, B)-invariant subspace $\mathcal{V}_g$ is said to be an (A, B)-stabilizable subspace if there exists an $F \in \mathcal{F}(A, B, \mathcal{V}_g)$ such that $\sigma(A + BF|\mathcal{V}_g) \subset \mathbf{C}^-$. An (A, B)-invariant subspace $\mathcal{R}$ of dimension ρ is said to be an (A, B)-controllability subspace if for any conjugate set of ρ complex numbers, say Λ, there exists an $F \in \mathcal{F}(A, B, \mathcal{R})$ such that $\sigma(A + BF|\mathcal{R}) = \Lambda$. Let $\underline{\mathcal{V}}$, $\underline{\mathcal{V}_g}$, and $\underline{\mathcal{R}}$ be the sets of all (A, B)-invariant subspaces, (A, B)-stabilizable subspaces, and (A, B)-controllability subspaces, respectively. It is well known that $\underline{\mathcal{V}}$, $\underline{\mathcal{V}_g}$, and $\underline{\mathcal{R}}$ are all closed under subspace addition, and thus there are supremal subspaces of all elements of $\underline{\mathcal{V}}$, $\underline{\mathcal{V}_g}$, and $\underline{\mathcal{R}}$ contained in any given subspace $\mathcal{L} \subset \mathbf{R}^n$, respectively. These three supremal subspaces will be denoted by $\mathcal{V}^\star(A, B, \mathcal{L})$, $\mathcal{V}^\star_{stab}(A, B, \mathcal{L})$, and $\mathcal{R}^\star(A, B, \mathcal{L})$, respectively.

The structure at infinity of the matrix triplet (A, B, C) with appropriate dimensions is described by the multiplicity of the zeros at infinity of the rational function matrix $C(sI - A)^{-1}B$. The content at infinity of (A, B, C), denoted as $C_\infty(A, B, C)$, is the total sum of the orders (counting multiplicities) of the zeros of $C(sI-A)^{-1}B$ at infinity. The unstable invariant content of (A, B, C), denoted as $C^+(A, B, C)$, is the total sum of the multiplicity orders of the unstable invariant zeros of $C(sI - A)^{-1}B$.

The geometric and structural solutions of the RRDPS are stated in the next two theorems.

THEOREM 1.2 (geometric solution [13]). *Given system* (1.1) *with* $E = I$*, assume that* (A, B) *is controllable. Then the following statements are equivalent:*

(i) *The RRDPS is solvable.*
(ii) $\mathcal{V}^\star(A, B, \mathrm{Ker}(C)) = \bigcap_{i=1}^m \mathcal{V}^\star(A, B, \mathrm{Ker}(c_i))$,
$\mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(C)) \supset \bigcap_{i=1}^m \mathcal{R}^\star(A, B, \mathrm{Ker}(c_i))$.
(iii) $\mathcal{V}^\star(A, B, \mathrm{Ker}(C)) = \bigcap_{i=1}^m \mathcal{V}^\star(A, B, \mathrm{Ker}(c_i))$,
$\mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(C)) = \bigcap_{i=1}^m \mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(c_i))$.

THEOREM 1.3 (structural solution [7, 13]). *Given system* (1.1) *with* $E = I$*, assume that* (A, B) *is controllable. Then the RRDPS is solvable if and only if*

$$C_\infty(A, B, C) = \sum_{i=1}^m C_\infty(A, B, c_i), \quad C^+(A, B, C) = \sum_{i=1}^m C^+(A, B, c_i). \tag{1.6}$$

In principle, all invariant subspaces needed in Theorem 1.2 can be computed by the computer routines found in [1], and we can verify the solvability of the RRDPS using Theorem 1.2 and Theorem 1.3 by computing either $2m + 4$ subspaces $\mathcal{V}^\star(A, B, \mathrm{Ker}(C))$, $\mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(C))$, $\mathcal{V}^\star(A, B, \mathrm{Ker}(c_i))$, and $\mathcal{R}^\star(A, B, \mathrm{Ker}(c_i))$ (or $\mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(c_i))$), $i = 1, \ldots, m$, $\bigcap_{i=1}^m \mathcal{V}^\star(A, B, \mathrm{Ker}(c_i))$ and $\bigcap_{i=1}^m \mathcal{R}^\star(A, B, \mathrm{Ker}(c_i))$ (or $\bigcap_{i=1}^m \mathcal{V}^\star_{stab}(A, B, \mathrm{Ker}(c_i))$), or the structures at infinity and the unstable invariant zero structures of $m + 1$ matrix triplets (A, B, C), (A, B, c_i), $i = 1, \ldots, m$. However, Theorems 1.2 and 1.3 do not provide numerically reliable procedures for the construction of the desired feedback matrices F and H. To our knowledge, there are at present no numerically implementable and reliable methods developed for solving the RRDPS based on Theorems 1.2 and 1.3.

If the stability requirement is not imposed, then an explicit solution for the row by row decoupling problem can be found in [3, 10].

THEOREM 1.4 (see [3, 10]). *Given system* (1.1) *with* E *nonsingular, denote* c_i *to be the* i*th row of* C *if* $c_i(E^{-1}A)^j(E^{-1}B) \neq 0$ *for some nonnegative integer* j*, then set*

$$l_i = \min\{j \geq 0 : j \text{ is integer satisfying } c_i(E^{-1}A)^j(E^{-1}B) \neq 0\};$$

otherwise, set $l_i = n - 1$*. Define*

$$L = \begin{bmatrix} c_1(E^{-1}A)^{l_1} \\ c_2(E^{-1}A)^{l_2} \\ \vdots \\ c_m(E^{-1}A)^{l_m} \end{bmatrix} (E^{-1}B).$$

Then there exist matrices $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{n\times n}$ *with* H *nonsingular such that* (1.5) *holds if and only if the matrix* L *is nonsingular. In this case, the desired matrices*

F and H can be chosen to be

$$F = -L^{-1} \begin{bmatrix} c_1(E^{-1}A)^{l_1+1} \\ c_2(E^{-1}A)^{l_2+1} \\ \vdots \\ c_m(E^{-1}A)^{l_m+1} \end{bmatrix}$$

and $H = L^{-1}$.

Theorem 1.4 provides a numerical procedure for solving the row by row decoupling problem without stability requirement. However, this procedure is numerically unreliable, because if E is ill-conditioned, then the explicit computation of E^{-1} will lead to a numerically unstable procedure [11]. Furthermore, even in the case $E = I$, it is very dangerous to compute the powers of matrix A. This can be illustrated by the following example.

EXAMPLE 1 (see [17]). *Let the third power of*

$$A = \begin{bmatrix} -20.8571 & -9.06467 & 12.9955 \\ -43.1924 & -18.7528 & 26.8994 \\ -63.5860 & -27.6219 & 39.6100 \end{bmatrix}$$

be obtained from $\bar{A}^2 = \mathrm{fl}(A \times A)$ and $\bar{A}^3 = \mathrm{fl}(\bar{A}^2 \times A)$ using floating-point arithmetic with accuracy 2^{-15}. The final result is

$$\bar{A}^3 = \begin{bmatrix} -0.057004 & -0.024739 & 0.035445 \\ -0.785385 & -0.341064 & 0.489075 \\ -0.154087 & -0.067112 & 0.096242 \end{bmatrix},$$

while the exact answer (up to six significant digits) is

$$A^3 = \begin{bmatrix} 0.0090618 & 0.0039361 & -0.0056432 \\ 0.0187545 & 0.0081469 & -0.0011680 \\ 0.0276109 & 0.0119938 & -0.0171949 \end{bmatrix}.$$

Here, we see that even the signs of the elements of the computed result are wrong.

This implies that Theorem 1.4 cannot be used to solve the row by row decoupling problem even without stability requirement. Up to now, Theorem 1.4 has not been generalized to the RRDPS, and therefore there does not yet exist a similar explicit solution for the RRDPS. Hence, Theorem 1.4 cannot lead to a numerically reliable algorithm for the RRDPS.

Finally, we note that in [1] the row by row decoupling problem with stability by measurement feedback is studied, and necessary and sufficient conditions are presented with a constructive proof. But the idea in this constructive proof does not work for the RRDPS. Moreover, in the diskette provided with [1] there are no routines available for computing the desired feedback matrices of the RRDPS.

Based on the above, our observation is that there is still a lack of effective numerical methods for solving the RRDPS. The stability of the closed-loop system (1.4) is one of most important properties and one of the fundamental requirements in systems design. Hence, it is necessary to develop numerically reliable methods for solving the RRDPS. This motivates us to consider the numerical computation of the RRDPS. We will develop a numerically reliable algorithm for the RRDPS using Theorem 1.2 and the numerical linear algebra technique.

The basis of our main results is some condensed forms under orthogonal transformations, which can be implemented in numerically stable ways. Hence, our results lead to numerically reliable methods for solving the RRDPS using standardized numerical linear algebra software such as MATLAB.

Throughout this paper, $\mathrm{rank}_g[\cdot](s)$ denotes the generic rank of a rational matrix $[\cdot](s)$. For convenience we do not distinguish between a matrix with orthogonal columns and the space spanned by its columns.

2. Preliminary. In this section we will provide three condensed forms, which will then be used in the next section to establish a nice reduction property for the RRDPS. The first condensed form is presented in Theorem 2.1 to reveal the $(E^{-1}A, E^{-1}B)$-stabilizable subspace $\mathcal{V}^{\star}_{stab}(E^{-1}A, E^{-1}B, \mathrm{Ker}(C))$.

THEOREM 2.1. *Given a system of the form* (1.1) *with* $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{m\times n}$, *assume that* $(E^{-1}A, E^{-1}B)$ *is controllable. There exist orthogonal matrices* $P, Q \in \mathbf{R}^{n\times n}$ *such that*

$$P(sE-A)Q = \begin{array}{c} \\ n_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} n_1 & n_2 \end{array} \\ \begin{bmatrix} sE_{11}-A_{11} & sE_{12}-A_{12} \\ -A_{21} & sE_{22}-A_{22} \\ 0 & sE_{32}-A_{32} \end{bmatrix} \end{array}, \quad PB = \begin{array}{c} \\ n_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \end{array}\!\!\begin{bmatrix} B_1 \\ B_2 \\ 0 \end{bmatrix}, \quad CQ = \begin{array}{c} \begin{array}{cc} n_1 & n_2 \end{array} \\ \begin{bmatrix} 0 & C_2 \end{bmatrix} \end{array}, \tag{2.1}$$

where $n_1 + n_2 = n_1 + \tilde{n}_2 + \tilde{n}_3 = n$, E_{11} *is nonsingular,* B_2 *is of full row rank, and*

$$\mathrm{rank}\begin{bmatrix} sE_{11}-A_{11} & B_1 \\ -A_{21} & B_2 \end{bmatrix} = n_1 + \tilde{n}_2 \qquad \forall s \in \mathbf{C}/\mathbf{C}^{-}, \tag{2.2}$$

$$\mathrm{rank}\begin{bmatrix} sE_{32}-A_{32} \\ C_2 \end{bmatrix} = n_2 \qquad \forall s \in \mathbf{C}^{-}. \tag{2.3}$$

Consequently,

$$\mathcal{V}^{\star}_{stab}(E^{-1}A, E^{-1}B, \mathrm{Ker}(C)) = Q\begin{bmatrix} I_{n_1} \\ 0 \end{bmatrix}. \tag{2.4}$$

Proof. The condensed form (2.1) is constructed by Algorithm 2 in the appendix, and (2.4) is proved in the necessity part of Theorem 3.2 (see (3.31)). ∎

After deflating the subspace $\mathcal{V}^{\star}_{stab}(E^{-1}A, E^{-1}B, \mathrm{Ker}(C))$ by the condensed form (2.1), we get a reduced matrix quadruplet $(\left[\begin{smallmatrix} E_{22} \\ E_{32} \end{smallmatrix}\right], \left[\begin{smallmatrix} A_{22} \\ A_{32} \end{smallmatrix}\right], \left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right], C_2)$. Related to this reduced matrix quadruplet is the following theorem.

THEOREM 2.2. *Given a system of the form* (1.1) *with* $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{m\times n}$, *assume that the condensed form* (2.1) *has been determined.*

(i) *There exist orthogonal matrices* U, V, *and* W *such that*

$$U\begin{bmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{bmatrix}V = \begin{array}{c} \\ \mu_1 \\ \tilde{\mu}_2 \\ \tilde{\mu}_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} \mu_1 & \mu_2 \end{array} \\ \begin{bmatrix} s\Theta_{11}-\Phi_{11} & s\Theta_{12}-\Phi_{12} \\ -\Phi_{21} & s\Theta_{22}-\Phi_{22} \\ 0 & s\Theta_{32}-\Phi_{32} \end{bmatrix} \end{array},$$

$$U\begin{bmatrix} B_2 \\ 0 \end{bmatrix}W = \begin{array}{c} \\ \mu_1 \\ \tilde{\mu}_2 \\ \tilde{\mu}_3 \end{array}\!\!\begin{array}{c} \begin{array}{cc} m-\tilde{\mu}_2 & \tilde{\mu}_2 \end{array} \\ \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ 0 & \Psi_{22} \\ 0 & 0 \end{bmatrix} \end{array}, \quad C_2V = \begin{array}{c} \\ m-1 \\ 1 \end{array}\!\!\begin{array}{c} \begin{array}{cc} \mu_1 & \mu_2 \end{array} \\ \begin{bmatrix} \Xi_{11} & \Xi_{12} \\ 0 & \xi_{22} \end{bmatrix} \end{array}, \tag{2.5}$$

where Θ_{11} *and* Ψ_{22} *are nonsingular,* $\mu_1+\mu_2=\mu_1+\tilde{\mu}_2+\tilde{\mu}_3=n_2$, *and*

$$\operatorname{rank}\begin{bmatrix} s\Theta_{11}-\Phi_{11} & \Psi_{11} & \Psi_{12} \\ -\Phi_{21} & 0 & \Psi_{22} \end{bmatrix} = \mu_1+\tilde{\mu}_2 \qquad \forall s\in\mathbf{C}, \tag{2.6}$$

$$\operatorname{rank}_g\begin{bmatrix} s\Theta_{32}-\Phi_{32} \\ \xi_{22} \end{bmatrix} = \mu_2. \tag{2.7}$$

(ii) *There exist orthogonal matrices* $\mathcal{U}$ *and* $\mathcal{V}$ *such that*

$$\mathcal{U}U\begin{bmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{bmatrix}V\mathcal{V} = \begin{matrix} \\ \nu_1 \\ \tilde{\nu}_2 \\ \tilde{\nu}_3 \end{matrix}\begin{matrix} \begin{matrix} \nu_1 & \nu_2 \end{matrix} \\ \begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} & s\mathcal{E}_{12}-\mathcal{A}_{12} \\ -\mathcal{A}_{21} & s\mathcal{E}_{22}-\mathcal{A}_{22} \\ 0 & s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix} \end{matrix},$$

$$\mathcal{U}U\begin{bmatrix} B_2 \\ 0 \end{bmatrix}W = \begin{matrix} \\ \nu_1 \\ \tilde{\nu}_2 \\ \tilde{\nu}_3 \end{matrix}\begin{matrix} \begin{matrix} m-\tilde{\mu}_2 & \tilde{\mu}_2 \end{matrix} \\ \begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \\ \mathcal{B}_{21} & \mathcal{B}_{22} \\ 0 & 0 \end{bmatrix} \end{matrix}, \quad C_2V\mathcal{V} = \begin{matrix} m-1 \\ 1 \end{matrix}\begin{matrix} \begin{matrix} \nu_1 & \nu_2 \end{matrix} \\ \begin{bmatrix} 0 & \mathcal{C}_{12} \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \end{matrix}, \tag{2.8}$$

where $\mathcal{E}_{11}$ *is nonsingular,* $\begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}$ *is of full row rank,* $\nu_1+\nu_2=\nu_1+\tilde{\nu}_2+\tilde{\nu}_3=n_2$, *and*

$$\operatorname{rank}\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} & \mathcal{B}_{11} & \mathcal{B}_{12} \\ -\mathcal{A}_{21} & \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix} = \nu_1+\tilde{\nu}_2 \qquad \forall s\in\mathbf{C}, \tag{2.9}$$

$$\operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{32}-\mathcal{A}_{32} \\ \mathcal{C}_{12} \end{bmatrix} = \nu_2. \tag{2.10}$$

Proof. The condensed form (2.5) is constructed by Algorithm 3 in the appendix. The condensed form (2.8) can be constructed similarly. ∎

The following four lemmas will be used frequently in the next two sections.

LEMMA 2.3 (see [5]). *Given* $\mathcal{E}$, $\mathcal{A}\in\mathbf{R}^{n\times n}$, $\mathcal{B}\in\mathbf{R}^{n\times m}$, $\mathcal{C}\in\mathbf{R}^{p\times n}$, *and* $\mathcal{D}\in\mathbf{R}^{p\times m}$ *with* $\mathcal{E}$ *nonsingular, then* $\mathcal{C}(s\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}+\mathcal{D}=0$ *if and only if* $\mathcal{D}=0$ *and* $\operatorname{rank}_g\begin{bmatrix} s\mathcal{E}-\mathcal{A} & \mathcal{B} \\ \mathcal{C} & 0 \end{bmatrix}=n$.

Proof. Since $\mathcal{E}$ is nonsingular, the proof of Lemma 2.3 follows directly from the fact that

$$n+\operatorname{rank}(\mathcal{D}) \le n+\operatorname{rank}_g(\mathcal{C}(s\mathcal{E}-\mathcal{A})^{-1}\mathcal{B}+\mathcal{D}) = \operatorname{rank}_g\begin{bmatrix} s\mathcal{E}-\mathcal{A} & \mathcal{B} \\ \mathcal{C} & -\mathcal{D} \end{bmatrix}. \tag{2.11}$$ ∎

LEMMA 2.4. *Given* $\mathcal{E},\mathcal{A}\in\mathbf{R}^{n\times n},\mathcal{B}\in\mathbf{R}^{n\times m},\mathcal{C}\in\mathbf{R}^{p\times n},\mathcal{D}\in\mathbf{R}^{p\times m}$ *with* $\mathcal{E}$ *nonsingular, and* $(\mathcal{E},\mathcal{A};\mathcal{B})$ *is controllable, then*

$$\operatorname{rank}_g\begin{bmatrix} s\mathcal{E}-\mathcal{A} & \mathcal{B} \\ \mathcal{C} & -\mathcal{D} \end{bmatrix} = n \tag{2.12}$$

if and only if $\mathcal{C}=0$ *and* $\mathcal{D}=0$.

Proof. We only need to prove the "necessity." That $\mathcal{D}=0$ follows directly from (2.11) and (2.12). Now we should only show that $\mathcal{C}=0$. In fact, if $\mathcal{C}\neq 0$, then there exist nonsingular matrices $\mathcal{X}$ and $\mathcal{Y}$ [20] such that

$$\mathcal{X}(s\mathcal{E}-\mathcal{A})\mathcal{Y} = \begin{matrix} n_1 \\ n_2 \end{matrix}\begin{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} & 0 \\ s\mathcal{E}_{21}-\mathcal{A}_{21} & s\mathcal{E}_{22}-\mathcal{A}_{22} \end{bmatrix} \end{matrix}, \quad \mathcal{X}B = \begin{matrix} n_1 \\ n_2 \end{matrix}\begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \end{bmatrix}, \ \mathcal{C}\mathcal{Y} = \begin{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{bmatrix} \mathcal{C}_1 & 0 \end{bmatrix} \end{matrix},$$

where $\mathcal{C}_1 \neq 0$ and $(\mathcal{E}_{11}^T, \mathcal{A}_{11}^T; \mathcal{C}_1^T)$ is controllable. The controllability of $(\mathcal{E}, \mathcal{A}; \mathcal{B})$ implies that $(\mathcal{E}_{11}, \mathcal{A}_{11}; \mathcal{B}_1)$ is controllable. Hence, $\mathcal{C}(s\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} = \mathcal{C}_1(s\mathcal{E}_{11} - \mathcal{A}_{11})^{-1}\mathcal{B}_1 \neq 0$, which, along with $\mathcal{D} = 0$, gives

$$\operatorname{rank}_g \begin{bmatrix} s\mathcal{E} - \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} = n + \operatorname{rank}_g(\mathcal{C}_1(s\mathcal{E}_{11} - \mathcal{A}_{11})^{-1}\mathcal{B}_1) > n.$$

This contradicts condition (2.12). Therefore, we must have $\mathcal{C}_1 = 0$, i.e., $\mathcal{C} = 0$. ∎

In general, for matrices $\mathcal{A}_{ij}(i, j = 1, 2)$ of appropriate dimensions, 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, we have Lemma 2.5, as follows.

LEMMA 2.5. *Given matrices $\mathcal{A}_{11} \in \mathbf{R}^{n_1 \times n_1}$, $\mathcal{A}_{12} \in \mathbf{R}^{n_1 \times n_2}$, $\mathcal{A}_{21} \in \mathbf{R}^{\tilde{n}_2 \times n_1}$, $\mathcal{A}_{22} \in \mathbf{R}^{\tilde{n}_2 \times n_2}$ with $\mathcal{A}_{11}$ nonsingular, let orthogonal matrix $\mathcal{P}$ satisfy*

$$\mathcal{P} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} = \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \begin{bmatrix} \tilde{\mathcal{A}}_{11} \\ 0 \end{bmatrix}, \quad \mathcal{P} = \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \overset{\begin{matrix} n_1 & \tilde{n}_2 \end{matrix}}{\begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix}}. \tag{2.13}$$

Denote

$$\mathcal{P} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} = \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \begin{bmatrix} \tilde{\mathcal{A}}_{12} \\ \tilde{\mathcal{A}}_{22} \end{bmatrix}.$$

Then $\mathcal{P}_{11}$ and $\mathcal{P}_{22}$ are nonsingular,

$$\begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11} \\ \mathcal{A}_{21} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{12} \\ \mathcal{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{12} \\ \tilde{\mathcal{A}}_{22} \end{bmatrix}, \tag{2.14}$$

and, furthermore,

$$\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} = \mathcal{P}_{22}^{-1}\tilde{\mathcal{A}}_{22}. \tag{2.15}$$

Proof. Equation (2.14) is obvious. Since

$$\mathcal{A}_{11} = \mathcal{P}_{11}^T\tilde{\mathcal{A}}_{11},$$

then $\mathcal{P}_{11}$ 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, along with the nonsingularity of $\mathcal{P}_{11}$ and $\mathcal{P}_{22}$, gives

$$\mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T = \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T - \mathcal{P}_{22}\mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T) = \mathcal{P}_{22}^{-1}(\mathcal{P}_{22}\mathcal{P}_{22}^T + \mathcal{P}_{21}\mathcal{P}_{21}^T) = \mathcal{P}_{22}^{-1}.$$

On the other hand,

$$\mathcal{A}_{21} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{11}, \quad \mathcal{A}_{22} = \mathcal{P}_{12}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{22}^T\tilde{\mathcal{A}}_{22}, \quad \mathcal{A}_{12} = \mathcal{P}_{11}^T\tilde{\mathcal{A}}_{12} + \mathcal{P}_{21}^T\tilde{\mathcal{A}}_{22}.$$

Hence, we have

$$\mathcal{A}_{22} - \mathcal{A}_{21}\mathcal{A}_{11}^{-1}\mathcal{A}_{12} = (\mathcal{P}_{22}^T - \mathcal{P}_{12}^T\mathcal{P}_{11}^{-T}\mathcal{P}_{21}^T)\tilde{\mathcal{A}}_{22} = \mathcal{P}_{22}^{-1}\tilde{\mathcal{A}}_{22}.$$ ∎

Lemma 2.5 implies 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 $\left[\begin{smallmatrix} I & 0 \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{smallmatrix}\right]$ is not orthogonal, (2.14) is computed

by only orthogonal transformations which are numerically stable. This feature will play an important role in the next two sections.

The following result is well known in control theory.

LEMMA 2.6 (see [20]). *Given $E, A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}$ with E nonsingular, there exists a matrix $F \in \mathbf{R}^{m\times n}$ such that $E^{-1}(A+BF)$ is stable, i.e.,*

$$\operatorname{rank}(sE - A - BF) = n \qquad \forall s \in \mathbf{C}/\mathbf{C}^-$$

if and only if

$$\operatorname{rank}\begin{bmatrix} sE - A & B \end{bmatrix} = n \qquad \forall s \in \mathbf{C}/\mathbf{C}^-.$$

The numerically reliable methods for the construction of F in Lemma 2.6 can be found in [15, 19].

3. Reduction property. The purpose of this section is to derive a reduction property for the RRDPS based on the condensed forms (2.1), (2.5), and (2.8). This reduction property will be the basis of the numerically reliable algorithm for solving the RRDPS in section 4.

First we consider the row by row decoupling problem without stability requirement for the reduced matrix quadruplet $(\left[\begin{smallmatrix} E_{22} \\ E_{32} \end{smallmatrix}\right], \left[\begin{smallmatrix} A_{22} \\ A_{32} \end{smallmatrix}\right], \left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right], C_2)$ in the condensed form (2.1).

THEOREM 3.1. *Given system* (1.1), *assume that the condensed forms* (2.1), (2.5), *and* (2.8) *have been determined. Then there exist matrices $F_2 \in \mathbf{R}^{m\times n_2}$ and $H \in \mathbf{R}^{m\times m}$ with H nonsingular such that*

$$C_2 \begin{bmatrix} sE_{22} - A_{22} - B_2F_2 \\ sE_{32} - A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \text{ is nonsingular and diagonal} \tag{3.1}$$

if and only if

$$\tilde{\mu}_2 = 1, \quad \tilde{\nu}_2 = m - 1, \tag{3.2}$$

and, furthermore, there exist matrices $\mathcal{F} \in \mathbf{R}^{(m-1)\times\nu_2}$ and $\mathcal{H} \in \mathbf{R}^{(m-1)\times(m-1)}$ with $\mathcal{H}$ nonsingular such that

$$\mathcal{C}_{12} \begin{bmatrix} s\mathcal{E}_{22} - \mathcal{A}_{22} - \mathcal{B}_{21}\mathcal{F} \\ s\mathcal{E}_{32} - \mathcal{A}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{B}_{21} \\ 0 \end{bmatrix} \mathcal{H} \text{ is nonsingular and diagonal.} \tag{3.3}$$

Moreover, in the case that conditions (3.2) *and* (3.3) *hold, we have that $\mathcal{B}_{21}$ and $\left[\begin{smallmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{smallmatrix}\right]$ are nonsingular, and the desired feedback matrices F_2 and H can be parameterized by*

$$H = W \begin{bmatrix} \mathcal{H} & -\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{H}_{22} \\ 0 & \mathcal{H}_{22} \end{bmatrix}, \quad F_2 = W \begin{bmatrix} \mathcal{F}_{11} & \mathcal{F} - \mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{F}_{22} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} \mathcal{V}^T V^T, \tag{3.4}$$

where $\mathcal{H}_{22} \in \mathbf{R}$, $\mathcal{H}_{22} \neq 0$, $(\mathcal{F}, \mathcal{H})$ satisfy (3.3),

$$\begin{bmatrix} \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} = \begin{bmatrix} -\Phi_{21}/\Psi_{22} & X \end{bmatrix} \mathcal{V}, \ X \in \mathbf{R}^{1\times\mu_2} \text{ is arbitrary}, \tag{3.5}$$

$$\mathcal{F}_{11} = -\mathcal{B}_{21}^{-1}(\mathcal{A}_{21} + \mathcal{B}_{22}\mathcal{F}_{21}). \tag{3.6}$$

Proof. First we prove the "necessity" and then the "sufficiency."

Necessity. Assume that matrices $F_2 \in \mathbf{R}^{m\times n_2}$ and $H \in \mathbf{R}^{m\times m}$ with H nonsingular satisfy (3.1). Partition F_2, H into

$$W^T H = \begin{matrix} \\ m-\tilde{\mu}_2 \\ \tilde{\mu}_2 \end{matrix} \begin{matrix} \begin{matrix} m-1 & 1 \end{matrix} \\ \begin{bmatrix} \mathcal{H} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{bmatrix} \end{matrix}, \qquad W^T F_2 V\mathcal{V} = \begin{matrix} \\ m-\tilde{\mu}_2 \\ \tilde{\mu}_2 \end{matrix} \begin{matrix} \begin{matrix} \nu_1 & \nu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{F}_{11} & \tilde{\mathcal{F}} \\ \mathcal{F}_{21} & \mathcal{F}_{22} \end{bmatrix} \end{matrix},$$

$$W^T F_2 V = \begin{matrix} \\ m-\tilde{\mu}_2 \\ \tilde{\mu}_2 \end{matrix} \begin{matrix} \begin{matrix} \mu_1 & \mu_2 \end{matrix} \\ \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & X \end{bmatrix} \end{matrix}. \tag{3.7}$$

Since (3.1) holds, using the condensed form (2.5) we get

$$\begin{bmatrix} 0 & \xi_{22} \end{bmatrix} \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}-\Psi_{12}F_{21} & s\Theta_{12}-\Phi_{12}-\Psi_{11}F_{12}-\Psi_{12}X \\ -\Phi_{21}-\Psi_{22}F_{21} & s\Theta_{22}-\Phi_{22}-\Psi_{22}X \\ 0 & s\Theta_{32}-\Phi_{32} \end{bmatrix}^{-1} \begin{bmatrix} \Psi_{11}\mathcal{H}+\Psi_{12}\mathcal{H}_{21} \\ \Psi_{22}\mathcal{H}_{21} \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} C_2 \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \begin{bmatrix} I_{m-1} \\ 0 \end{bmatrix} = 0.$$

Thus, Lemma 2.3 gives that

$$\operatorname{rank}_g \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}-\Psi_{12}F_{21} & s\Theta_{12}-\Phi_{12}-\Psi_{11}F_{12}-\Psi_{12}X & \Psi_{11}\mathcal{H}+\Psi_{12}\mathcal{H}_{21} \\ -\Phi_{21}-\Psi_{22}F_{21} & s\Theta_{22}-\Phi_{22}-\Psi_{22}X & \Psi_{22}\mathcal{H}_{21} \\ 0 & s\Theta_{32}-\Phi_{32} & 0 \\ 0 & \xi_{22} & 0 \end{bmatrix}$$

$$= n_2 = \mu_1 + \mu_2, \tag{3.8}$$

which, along with property (2.7), yields that

$$\mu_1 = \operatorname{rank}_g \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}-\Psi_{12}F_{21} & \Psi_{11}\mathcal{H}+\Psi_{12}\mathcal{H}_{21} \\ -\Phi_{21}-\Psi_{22}F_{21} & \Psi_{22}\mathcal{H}_{21} \end{bmatrix}. \tag{3.9}$$

However, $\Theta_{11} \in \mathbf{R}^{\mu_1\times\mu_1}$ is nonsingular, and thus

$$\operatorname{rank}_g \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}-\Psi_{12}F_{21} & \Psi_{11}\mathcal{H}+\Psi_{12}\mathcal{H}_{21} \\ -\Phi_{21}-\Psi_{22}F_{21} & \Psi_{22}\mathcal{H}_{21} \end{bmatrix} \geq \mu_1 + \operatorname{rank}(\Psi_{22}\mathcal{H}_{21}),$$

which implies that $\Psi_{22}\mathcal{H}_{21} = 0$. Note that Ψ_{22} is nonsingular. Thus, we obtain

$$\mathcal{H}_{21} = 0. \tag{3.10}$$

Consequently, (3.1) and the condensed form (2.5) yield

$$\begin{aligned} n_2 + 1 &= \operatorname{rank}_g \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix} \\ &\quad + \operatorname{rank}_g \left(\begin{bmatrix} 0 & 1 \end{bmatrix} C_2 \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \right) \\ &= \operatorname{rank}_g \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 & B_2H \\ sE_{32}-A_{32} & 0 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} C_2 & 0 \end{bmatrix} = \operatorname{rank}_g \begin{bmatrix} sE_{22}-A_{22} & B_2 \\ sE_{32}-A_{32} & 0 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} C_2 & 0 \end{bmatrix} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(since } H \text{ is nonsingular)} \\ &= \mu_2 + \mu_1 + \tilde{\mu}_2 = n_2 + \tilde{\mu}_2. \end{aligned}$$

Therefore, $\tilde{\mu}_2 = 1$, i.e., the first equality in condition (3.2) holds.

Since H is nonsingular, as a direct consequence of condition (3.2) and the equality (3.10), we know that

$$\mathcal{H} \text{ and } \mathcal{H}_{22} \text{ are nonsingular.} \tag{3.11}$$

Now, $\tilde{\mu}_2 = 1$, and using property (2.9), equations (3.9), (3.10), (3.11), and Lemma 2.4 we have

$$\Phi_{21} + \Psi_{22}F_{21} = 0, \text{ i.e., } F_{21} = -\Psi_{22}^{-1}\Phi_{21} = -\Phi_{21}/\Psi_{22}, \tag{3.12}$$

which also implies (3.5).

By (3.1) and the condensed form (2.8), we also have

$$\begin{bmatrix} 0 & \mathcal{C}_{12} \end{bmatrix} \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & s\mathcal{E}_{12} - \mathcal{A}_{12} - \mathcal{B}_{11}\tilde{\mathcal{F}} - \mathcal{B}_{12}\mathcal{F}_{22} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & s\mathcal{E}_{22} - \mathcal{A}_{22} - \mathcal{B}_{21}\tilde{\mathcal{F}} - \mathcal{B}_{22}\mathcal{F}_{22} \\ 0 & s\mathcal{E}_{32} - \mathcal{A}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{B}_{11}\mathcal{H}_{12} + \mathcal{B}_{12}\mathcal{H}_{22} \\ \mathcal{B}_{21}\mathcal{H}_{12} + \mathcal{B}_{22}\mathcal{H}_{22} \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} I_{m-1} & 0 \end{bmatrix} C_2 \begin{bmatrix} sE_{22} - A_{22} - B_2F_2 \\ sE_{32} - A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

By Lemma 2.3 we get

$$\operatorname{rank}_g \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & s\mathcal{E}_{12} - \mathcal{A}_{12} - \mathcal{B}_{11}\tilde{\mathcal{F}} - \mathcal{B}_{12}\mathcal{F}_{22} & \mathcal{B}_{11}\mathcal{H}_{12} + \mathcal{B}_{12}\mathcal{H}_{22} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & s\mathcal{E}_{22} - \mathcal{A}_{22} - \mathcal{B}_{21}\tilde{\mathcal{F}} - \mathcal{B}_{22}\mathcal{F}_{22} & \mathcal{B}_{21}\mathcal{H}_{12} + \mathcal{B}_{22}\mathcal{H}_{22} \\ 0 & s\mathcal{E}_{32} - \mathcal{A}_{32} & 0 \\ 0 & \mathcal{C}_{12} & 0 \end{bmatrix}$$
$$= n_2 = \nu_1 + \nu_2.$$

Using (2.10) we have

$$\nu_1 = \operatorname{rank}_g \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & \mathcal{B}_{11}\mathcal{H}_{12} + \mathcal{B}_{12}\mathcal{H}_{22} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & \mathcal{B}_{21}\mathcal{H}_{12} + \mathcal{B}_{22}\mathcal{H}_{22} \end{bmatrix}. \tag{3.13}$$

Note that $\mathcal{E}_{11}$ and $\mathcal{H}_{22}$ are nonsingular, and thus

$$\mathcal{B}_{21}\mathcal{H}_{12} + \mathcal{B}_{22}\mathcal{H}_{22} = 0, \text{ i.e., } \mathcal{B}_{22} = -\mathcal{B}_{21}\mathcal{H}_{12}\mathcal{H}_{22}^{-1}. \tag{3.14}$$

Hence, $\operatorname{rank}\begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix} = \operatorname{rank}(\mathcal{B}_{21})$. Following that $\begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}$ is of full row rank, we know that $\mathcal{B}_{21}$ is also of full row rank, i.e.,

$$\operatorname{rank}(\mathcal{B}_{21}) = \tilde{\nu}_2. \tag{3.15}$$

Since $\mathcal{H}$ is nonsingular, (3.1), (3.2), and (3.10) give

$$\operatorname{rank}_g \left(\begin{bmatrix} 0 & \mathcal{C}_{12} \end{bmatrix} \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & s\mathcal{E}_{12} - \mathcal{A}_{12} - \mathcal{B}_{11}\tilde{\mathcal{F}} - \mathcal{B}_{12}\mathcal{F}_{22} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & s\mathcal{E}_{22} - \mathcal{A}_{22} - \mathcal{B}_{21}\tilde{\mathcal{F}} - \mathcal{B}_{22}\mathcal{F}_{22} \\ 0 & s\mathcal{E}_{32} - \mathcal{A}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{B}_{11} \\ \mathcal{B}_{21} \\ 0 \end{bmatrix} \right)$$
$$= \operatorname{rank}_g \left(\begin{bmatrix} I_{m-1} & 0 \end{bmatrix} C_2 \begin{bmatrix} sE_{22} - A_{22} - B_2F \\ sE_{32} - A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \begin{bmatrix} I_{m-1} \\ 0 \end{bmatrix} \right) = m - 1,$$

so property (2.9) and the fact that $\mathcal{B}_{21}$ is of full row rank imply that

$$\nu_1 + \nu_2 + m - 1 = \operatorname{rank}_g \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & s\mathcal{E}_{12} - \mathcal{A}_{12} - \mathcal{B}_{12}\tilde{\mathcal{F}} - \mathcal{B}_{12}\mathcal{F}_{22} & \mathcal{B}_{11} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & s\mathcal{E}_{22} - \mathcal{A}_{22} - \mathcal{B}_{21}\tilde{\mathcal{F}} - \mathcal{B}_{22}\mathcal{F}_{22} & \mathcal{B}_{21} \\ 0 & s\mathcal{E}_{32} - \mathcal{A}_{32} & 0 \\ 0 & \mathcal{C}_{12} & 0 \end{bmatrix}$$
$$= \nu_2 + \operatorname{rank}_g \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} - \mathcal{B}_{11}\mathcal{F}_{11} - \mathcal{B}_{12}\mathcal{F}_{21} & \mathcal{B}_{11} \\ -\mathcal{A}_{21} - \mathcal{B}_{21}\mathcal{F}_{11} - \mathcal{B}_{22}\mathcal{F}_{21} & \mathcal{B}_{21} \end{bmatrix} = \nu_2 + (\nu_1 + \tilde{\nu}_2).$$

Obviously, we get the second equality in condition (3.2), i.e., $\tilde{\nu}_2 = m-1$, and furthermore (3.15) yields that $\mathcal{B}_{21}$ is nonsingular. Consequently, we have from (3.14) that

$$\mathcal{H}_{12} = -\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{H}_{22},$$

which, along with (3.10) and (3.11), yields that H is of the form in (3.4).

Now property (2.9) is equivalent to

$$\text{(3.16)}\quad \operatorname{rank}\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11}+\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21} & \mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22} \end{bmatrix} = \nu_1 \qquad \forall s\in\mathbf{C}.$$

Returning to (3.13) and using (3.14), (3.11), and $\mathcal{H}_{12} = -\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{H}_{22}$ we get

$$\begin{aligned}
\nu_1 &= \operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11}-\mathcal{B}_{11}\mathcal{F}_{11}-\mathcal{B}_{12}\mathcal{F}_{21} & \mathcal{B}_{11}\mathcal{H}_{12}+\mathcal{B}_{12}\mathcal{H}_{22} \\ -\mathcal{A}_{21}-\mathcal{B}_{21}\mathcal{F}_{11}-\mathcal{B}_{22}\mathcal{F}_{21} & 0 \end{bmatrix} \\
&= \operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11}-\mathcal{B}_{11}\mathcal{F}_{11}-\mathcal{B}_{12}\mathcal{F}_{21} & (\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})\mathcal{H}_{22} \\ -\mathcal{A}_{21}-\mathcal{B}_{21}\mathcal{F}_{11}-\mathcal{B}_{22}\mathcal{F}_{21} & 0 \end{bmatrix} \\
&= \operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11}+\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21}-(\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})\mathcal{F}_{21} & (\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})\mathcal{H}_{22} \\ -\mathcal{A}_{21}-\mathcal{B}_{21}\mathcal{F}_{11}-\mathcal{B}_{22}\mathcal{F}_{21} & 0 \end{bmatrix} \\
&= \operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11}+\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21} & \mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22} \\ -\mathcal{A}_{21}-\mathcal{B}_{21}\mathcal{F}_{11}-\mathcal{B}_{22}\mathcal{F}_{21} & 0 \end{bmatrix},
\end{aligned} \tag{3.17}$$

which, along with (3.16) and Lemma 2.4, leads to the fact that

$$\text{(3.18)}\qquad \mathcal{A}_{21}+\mathcal{B}_{21}\mathcal{F}_{11}+\mathcal{B}_{22}\mathcal{F}_{21} = 0,$$

i.e., (3.6) is true.

From (3.1), (3.10), and (3.18) we know that

$$\text{(3.19)}\quad \mathcal{C}_{12}\begin{bmatrix} s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\tilde{\mathcal{F}}-\mathcal{B}_{22}\mathcal{F}_{22} \\ s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix}\mathcal{B}_{21}\mathcal{H} \text{ is nonsingular and diagonal.}$$

Now, taking

$$\text{(3.20)}\qquad \mathcal{F} = \tilde{\mathcal{F}}+\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{F}_{22},$$

we have the condition (3.3).

Sufficiency. We prove "sufficiency" constructively. Since $\left[\begin{smallmatrix} E_{22} \\ E_{32}\end{smallmatrix}\right]$ is nonsingular, we know $\left[\begin{smallmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32}\end{smallmatrix}\right]$ is nonsingular. Similarly, because of condition (3.3), $\left[\begin{smallmatrix} \mathcal{B}_{21} \\ 0\end{smallmatrix}\right]$ must be of full column rank, but condition (3.2) holds, so $\mathcal{B}_{21}$ is nonsingular. Motivated by (3.10), (3.14), (3.12), (3.18), and (3.20), define F_2 and H by (3.4), (3.5), and (3.6). We have the following:

(a) Since W^TF_2V is of the form

$$W^TF_2V = \begin{bmatrix} F_{11} & F_{12} \\ -\Phi_{21}/\Psi_{22} & X \end{bmatrix},$$

by the condensed form (2.5) we obtain

$$\begin{aligned}
&\begin{bmatrix} 0 & 1 \end{bmatrix} C_2\begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1}\begin{bmatrix} B_2 \\ 0 \end{bmatrix} H\begin{bmatrix} I_{m-1} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & \xi_{22} \end{bmatrix}\begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}+\Psi_{12}\Phi_{21}/\Psi_{22} & s\Theta_{12}-\Phi_{12}-\Psi_{11}F_{12}-\Psi_{12}X \\ 0 & s\Theta_{22}-\Phi_{22}-\Psi_{22}X \\ 0 & s\Theta_{32}-\Phi_{32} \end{bmatrix}^{-1}\begin{bmatrix} \Psi_{11}\mathcal{H} \\ 0 \\ 0 \end{bmatrix} = 0.
\end{aligned} \tag{3.21}$$

Furthermore,

$$\begin{aligned}
&\operatorname{rank}_g\left(\begin{bmatrix} 0 & 1 \end{bmatrix} C_2 \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)\\
&= \operatorname{rank}_g \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}F_{11}+\Psi_{12}\Psi_{22}^{-1}\Phi_{21} & s\Theta_{12}-\Phi_{12}-\Psi_{11}F_{12}-\Psi_{12}X & -\Psi_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{H}_{22}+\Psi_{12}\mathcal{H}_{22} \\ 0 & s\Theta_{22}-\Phi_{22}-\Psi_{22}X & \Psi_{22}\mathcal{H}_{22} \\ 0 & s\Theta_{32}-\Phi_{32} & 0 \\ 0 & \xi_{22} & 0 \end{bmatrix} - n_2\\
&= \mu_1+\mu_2+1-n_2 = n_2+1-n_2 = 1,
\end{aligned}$$

so

$$\begin{bmatrix} 0 & 1 \end{bmatrix} C_2 \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H \begin{bmatrix} 0 \\ 1 \end{bmatrix} \neq 0. \tag{3.22}$$

(b) By the condensed form (2.8) we have

$$\begin{aligned}
&\begin{bmatrix} I_{m-1} & 0 \end{bmatrix} C_2 \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H\\
&= \begin{bmatrix} 0 & \mathcal{C}_{12} \end{bmatrix} \begin{bmatrix} s\mathcal{E}_{11}-\tilde{\mathcal{A}}_{11} & s\mathcal{E}_{12}-\tilde{\mathcal{A}}_{12} \\ 0 & s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\mathcal{F} \\ 0 & s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix} \begin{bmatrix} \mathcal{B}_{11}\mathcal{H} & \tilde{\mathcal{B}}_{12}\mathcal{H}_{22} \\ \mathcal{B}_{21}\mathcal{H} & 0 \end{bmatrix} = \begin{bmatrix} T(s) & 0 \end{bmatrix},
\end{aligned} \tag{3.23}$$

where

$$\begin{aligned}
T(s) &= \mathcal{C}_{12} \begin{bmatrix} s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\mathcal{F} \\ s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix} \begin{bmatrix} \mathcal{B}_{21} \\ 0 \end{bmatrix} \mathcal{H},\\
\tilde{\mathcal{A}}_{11} &= \mathcal{A}_{11}+\mathcal{B}_{11}\mathcal{F}_{11}+\mathcal{B}_{12}\mathcal{F}_{21},\\
\tilde{\mathcal{A}}_{12} &= \mathcal{A}_{12}+\mathcal{B}_{11}\mathcal{F}+\mathcal{B}_{12}\mathcal{F}_{22}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{F}_{22},\\
\tilde{\mathcal{B}}_{12} &= \mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22}.
\end{aligned}$$

Hence, (3.1) follows directly from (3.21)–(3.23) and condition (3.3). ∎

REMARK 1. *Condition* (3.2) *is equivalent to saying that the row by row decoupling problem without stability requirement for the following linear time-invariant system:*

$$\begin{bmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{bmatrix} \dot{\tilde{x}} = \begin{bmatrix} \mathcal{A}_{22} \\ \mathcal{A}_{32} \end{bmatrix} \tilde{x} + \begin{bmatrix} \mathcal{B}_{21} \\ 0 \end{bmatrix} \tilde{u},\ \tilde{y} = \mathcal{C}_{12}\tilde{x}$$

is solvable.

Partition $\mathcal{V}$ in the condensed form (2.8) as

$$\mathcal{V} = \begin{matrix} \\ \mu_1 \\ \mu_2 \end{matrix} \begin{matrix} \begin{matrix} \nu_1 & \nu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{V}_{11} & \mathcal{V}_{12} \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix} \end{matrix}. \tag{3.24}$$

For any F_2 defined by (3.4)–(3.6), a simple calculation yields that

$$\begin{aligned}
&\mathcal{U}U \begin{bmatrix} sE_{22}-A_{22}-B_2F_2 \\ sE_{32}-A_{32} \end{bmatrix} V\mathcal{V} \qquad (3.25)\\
&= \begin{bmatrix} s\mathcal{E}_{11}-(\mathcal{A}_{11}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21}-(\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})\Phi_{21}\mathcal{V}_{11}/\Psi_{22})-(\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})X\mathcal{V}_{21} & \star \\ 0 & s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\mathcal{F} \\ 0 & s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix},
\end{aligned}$$

where $\star$ denotes some block matrix which we are not interested in. If $\mathcal{V}_{21}$ is not of full column rank, then the choice of X such that

$$\operatorname{rank}(s\mathcal{E}_{11}-\mathcal{A}_{11}+\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21}+(\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})\Phi_{21}\mathcal{V}_{11}/\Psi_{22}-(\mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22})X\mathcal{V}_{21}) \tag{3.26}$$
$$= \nu_1 \qquad \forall s \in \mathbf{C}/\mathbf{C}^-$$

is a stabilization problem by output feedback, which is still an open question in control theory [20]. Fortunately, just because the quadruplet $(\left[\begin{smallmatrix}E_{22}\\E_{32}\end{smallmatrix}\right], \left[\begin{smallmatrix}A_{22}\\A_{32}\end{smallmatrix}\right], \left[\begin{smallmatrix}B_2\\0\end{smallmatrix}\right], C_2)$ is from the condensed form (2.1), we can show in the next theorem that if the RRDPS for system (1.1) is solvable, then $\mathcal{V}_{21}$ is of full column rank, and hence the above stabilization problem is equivalent to one by state feedback, which has been investigated extensively (see [16, 19]).

We are now ready to present our main result in this subsection.

THEOREM 3.2 (reduction property for the RRDPS). *Given system* (1.1) *with* $(E^{-1}A, E^{-1}B)$ *being controllable, assume that the condensed forms* (2.1), (2.5), *and* (2.8) *have been determined and partition* $\mathcal{V}$ *as* (3.24). *Then the RRDPS is solvable if and only if* $\mathcal{V}_{21}$ *is of full column rank, condition* (3.2) *is true, and, furthermore, there exist matrices* $\mathcal{F} \in \mathbf{R}^{(m-1)\times\nu_2}$ *and* $\mathcal{H} \in \mathbf{R}^{(m-1)\times(m-1)}$ *with* $\mathcal{H}$ *nonsingular such that*

$$\begin{bmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{A}_{22}+\mathcal{B}_{21}\mathcal{F} \\ \mathcal{A}_{32} \end{bmatrix} \text{ is stable} \tag{3.27}$$

and (3.3) *holds. Moreover, in the case that the RRDPS is solvable,* $(\left[\begin{smallmatrix}\mathcal{E}_{22}\\\mathcal{E}_{32}\end{smallmatrix}\right]^{-1}\left[\begin{smallmatrix}\mathcal{A}_{22}\\\mathcal{A}_{32}\end{smallmatrix}\right], \left[\begin{smallmatrix}\mathcal{E}_{22}\\\mathcal{E}_{32}\end{smallmatrix}\right]^{-1}\left[\begin{smallmatrix}\mathcal{B}_{21}\\0\end{smallmatrix}\right])$ *is controllable,* $\mathcal{B}_{21}$ *is nonsingular, and the desired feedback matrices* F *and* H *can be chosen to be*

$$H = W\begin{bmatrix} \mathcal{H} & -\mathcal{B}_{21}^{-1}\mathcal{B}_{22}\mathcal{H}_{22} \\ 0 & \mathcal{H}_{22} \end{bmatrix}, \quad F = \begin{matrix} n_1 & n_2 \\ \end{matrix} \begin{bmatrix} F_1 & F_2 \end{bmatrix} Q^T, \tag{3.28}$$

where $\mathcal{H}_{22} \in \mathbf{R}$, $\mathcal{H}_{22} \neq 0$, F_1 *satisfies*

$$B_2F_1 = -A_{21}, \quad E_{11}^{-1}(A_{11}+B_1F_1) \text{ is stable}, \tag{3.29}$$

and F_2 *is determined by* (3.4)–(3.6) *with* X, $(\mathcal{F}, \mathcal{H})$ *satisfying* (3.26), *and* (3.27) *and* (3.3), *respectively.*

Proof. We prove the "necessity" first and then the "sufficiency."

Necessity. Define

$$s\Theta - \Phi = \begin{bmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{bmatrix}, \quad \Psi = \begin{bmatrix} B_2 \\ 0 \end{bmatrix},$$

and denote the ith rows of C_2 in (2.1) and C as c_i and $\tilde{c}_i$, respectively. It is easy to see from the condensed form (2.1) that

$$Q^{-1} = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \\ 0 & E_{32} \end{bmatrix}^{-1} PE = \begin{bmatrix} E_{11} & E_{12} \\ 0 & \Theta \end{bmatrix}^{-1} PE.$$

Because B_2 is of full row rank and (2.2) holds, then by Lemma 2.6 there exists F_1 such that $B_2F_1 + A_{21} = 0$ and

$$
\begin{aligned}
&Q^{-1}(sI - E^{-1}A - E^{-1}B\begin{bmatrix} F_1 & 0 \end{bmatrix}Q^T)Q \\
&= \begin{matrix} \\ n_1 \\ n_2 \end{matrix}\begin{matrix} \begin{matrix} n_1 & \quad n_2 \end{matrix} \\ \begin{bmatrix} sI - E_{11}^{-1}A_{11} - E_{11}^{-1}B_1F_1 & -\tilde{A}_{12} \\ 0 & sI - \Theta^{-1}\Phi \end{bmatrix} \end{matrix}, \\
Q^{-1}E^{-1}B &= \begin{matrix} n_1 \\ n_2 \end{matrix}\begin{bmatrix} \tilde{B}_1 \\ \Theta^{-1}\Psi \end{bmatrix}, \qquad CQ = \begin{matrix} \begin{matrix} n_1 & \quad n_2 \end{matrix} \\ \begin{bmatrix} 0 & C_2 \end{bmatrix} \end{matrix},
\end{aligned}
$$

where $E_{11}^{-1}A_{11} + E_{11}^{-1}B_1F_1$ is stable, and $\tilde{B}_1$ and $\tilde{A}_{12}$ are two constant matrices. Consequently, we have

$$
\begin{aligned}
\mathcal{V}^\star(E^{-1}A, E^{-1}B, \mathrm{Ker}(C)) &= Q\begin{bmatrix} I_{n_1} & 0 \\ 0 & \mathcal{V}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(C_2)) \end{bmatrix}, \\
\mathcal{V}^\star(E^{-1}A, E^{-1}B, \mathrm{Ker}(c_i)) &= Q\begin{bmatrix} I_{n_1} & 0 \\ 0 & \mathcal{V}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\tilde{c}_i)) \end{bmatrix}, \quad i = 1, \ldots, m, \\
\mathcal{V}^\star_{stab}(E^{-1}A, E^{-1}B, \mathrm{Ker}(c_i)) &= Q\begin{bmatrix} I_{n_1} & 0 \\ 0 & \mathcal{V}^\star_{stab}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\tilde{c}_i)) \end{bmatrix}, \quad i = 1, \ldots, m.
\end{aligned}
\tag{3.30}
$$

Moreover, we also have from (2.3) that

$$
\mathrm{rank}\begin{bmatrix} sI - \Theta^{-1}\Phi - \Theta^{-1}\Psi F_2 \\ C_2 \end{bmatrix} = n_2 \qquad \forall F_2 \in \mathbf{R}^{m\times n_2} \quad \text{and} \quad \forall s \in \mathbf{C}^-.
$$

So

$$
\mathcal{V}^\star_{stab}(E^{-1}A, E^{-1}B, \mathrm{Ker}(C)) = Q\begin{bmatrix} I_{n_1} \\ 0 \end{bmatrix}, \quad \mathcal{V}^\star_{stab}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(C_2)) = \{0\}.
\tag{3.31}
$$

Similarly, by the condensed forms (2.5) and (2.8), we get

$$
V^{-1} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ 0 & \Theta_{22} \\ 0 & \Theta_{32} \end{bmatrix}^{-1} U\Theta, \quad (V\mathcal{V})^{-1} = \begin{bmatrix} \mathcal{E}_{11} & \mathcal{E}_{12} \\ 0 & \mathcal{E}_{22} \\ 0 & \mathcal{E}_{32} \end{bmatrix}^{-1} \mathcal{U}U\Theta.
$$

Since Ψ_{22} and $\begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}$ are of full row rank and properties (2.6), (2.7), (2.9), and (2.10) hold, then for any conjugate set $\tilde{\Lambda} = \{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{\mu_1}\}$ and $\hat{\Lambda} = \{\hat{\lambda}_1, \ldots, \hat{\lambda}_{\nu_1}\}$, there exist matrices $\tilde{F}_2$ and $\hat{F}_2$ such that

$$
\begin{aligned}
V^{-1}(sI - \Theta^{-1}\Phi - \Theta^{-1}\Psi\tilde{F}_2)V &= \begin{matrix} \\ \mu_1 \\ \mu_2 \end{matrix}\begin{matrix} \begin{matrix} \mu_1 & \quad \mu_2 \end{matrix} \\ \begin{bmatrix} sI - \tilde{\Phi}_{11} & -\tilde{\Phi}_{12} \\ 0 & sI - \Phi_2 \end{bmatrix} \end{matrix}, \\
V^{-1}\Theta^{-1}\Psi &= \begin{matrix} \mu_1 \\ \mu_2 \end{matrix}\begin{bmatrix} \tilde{\Psi}_1 \\ \tilde{\Psi}_2 \end{bmatrix}, \quad \tilde{c}_mV = \begin{matrix} \begin{matrix} \mu_1 & \quad \mu_2 \end{matrix} \\ \begin{bmatrix} 0 & \xi_{22} \end{bmatrix} \end{matrix}, \\
(V\mathcal{V})^{-1}(sI - \Theta^{-1}\Phi - \Theta^{-1}\Psi\hat{F}_2)(V\mathcal{V}) &= \begin{matrix} \\ \nu_1 \\ \nu_2 \end{matrix}\begin{matrix} \begin{matrix} \nu_1 & \quad \nu_2 \end{matrix} \\ \begin{bmatrix} sI - \hat{\mathcal{A}}_{11} & -\hat{\mathcal{A}}_{12} \\ 0 & sI - \mathcal{A}_2 \end{bmatrix} \end{matrix}, \\
(V\mathcal{V})^{-1}\Theta^{-1}\Psi &= \begin{matrix} \nu_1 \\ \nu_2 \end{matrix}\begin{bmatrix} \hat{\mathcal{B}}_1 \\ \hat{\mathcal{B}}_2 \end{bmatrix}, \quad \begin{bmatrix} I_{m-1} & 0 \end{bmatrix}C_2V\mathcal{V} = \begin{matrix} \begin{matrix} \nu_1 & \quad \nu_2 \end{matrix} \\ \begin{bmatrix} 0 & \mathcal{C}_{12} \end{bmatrix} \end{matrix},
\end{aligned}
$$

where

$$\sigma(\tilde{\Phi}_{11}) = \tilde{\Lambda}, \quad \Phi_2 = \begin{bmatrix} \Theta_{22} \\ \Theta_{32} \end{bmatrix}^{-1} \begin{bmatrix} \Phi_{22} \\ \Phi_{32} \end{bmatrix}, \quad \tilde{\Psi}_2 = \begin{bmatrix} \Theta_{22} \\ \Theta_{32} \end{bmatrix}^{-1} \begin{bmatrix} 0 & \Psi_{22} \\ 0 & 0 \end{bmatrix} W^T,$$
$$\sigma(\hat{\mathcal{A}}_{11}) = \hat{\Lambda}, \quad \mathcal{A}_2 = \begin{bmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{A}_{22} \\ \mathcal{A}_{32} \end{bmatrix}, \quad \hat{\mathcal{B}}_2 = \begin{bmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{B}_{21} & \mathcal{B}_{22} \\ 0 & 0 \end{bmatrix} W^T,$$

and

$$\operatorname{rank}_g \begin{bmatrix} \mathcal{T}_\infty^T(\tilde{\Psi}_2)(sI - \Phi_2) \\ \xi_{22} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} s\Theta_{32} - \Phi_{32} \\ \xi_{22} \end{bmatrix} = \mu_2,$$
$$\operatorname{rank}_g \begin{bmatrix} \mathcal{T}_\infty^T(\hat{\mathcal{B}}_2)(sI - \mathcal{A}_2) \\ \mathcal{C}_{12} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} s\mathcal{E}_{32} - \mathcal{A}_{32} \\ \mathcal{C}_{12} \end{bmatrix} = \nu_2.$$

In the above, $\mathcal{T}_\infty(M)$ denotes the null space of matrix M^T. Thus, we have

$$\mathcal{R}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(\tilde{c}_m)) = V \begin{bmatrix} I_{\mu_1} \\ 0 \end{bmatrix},$$
$$\mathcal{R}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(\begin{bmatrix} I_{m-1} & 0 \end{bmatrix} C_2)) = V\mathcal{V} \begin{bmatrix} I_{\nu_1} \\ 0 \end{bmatrix}. \tag{3.32}$$

Assume that the RRDPS is solvable. Then Theorem 1.2(iii) holds, i.e.,

$$\mathcal{V}^\star(E^{-1}A, E^{-1}B, \operatorname{Ker}(C)) = \bigcap_{i=1}^m \mathcal{V}^\star(E^{-1}A, E^{-1}B, \operatorname{Ker}(c_i)),$$
$$\mathcal{V}^\star_{stab}(E^{-1}A, E^{-1}B, \operatorname{Ker}(C)) = \bigcap_{i=1}^m \mathcal{V}^\star_{stab}(E^{-1}A, E^{-1}B, \operatorname{Ker}(c_i)),$$

which, along with (3.30) and (3.31), gives

$$\bigcap_{i=1}^m \mathcal{V}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(\tilde{c}_i)) = \mathcal{V}^\star(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(C_2)),$$
$$\bigcap_{i=1}^m \mathcal{V}^\star_{stab}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(\tilde{c}_i)) = \{0\} = \mathcal{V}^\star_{stab}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \operatorname{Ker}(C_2)). \tag{3.33}$$

Note that B_2 is of full row rank and the controllability of $(E^{-1}A, E^{-1}B)$ yields that $\operatorname{rank}(sE_{32} - A_{32}) = \tilde{n}_3$ for all $s \in \mathbf{C}$, so

$$\operatorname{rank} \begin{bmatrix} sE_{22} - A_{22} & B_2 \\ sE_{32} - A_{32} & 0 \end{bmatrix} = \tilde{n}_2 + \tilde{n}_3 = n_2 \qquad \forall s \in \mathbf{C}. \tag{3.34}$$

Hence, Theorem 1.2(iii) implies that there exist matrices F_2 and H with H nonsingular such that

$$C_2 \begin{bmatrix} sE_{22} - A_{22} - B_2F_2 \\ sE_{32} - A_{32} \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} H = C_2(sI - \Theta^{-1}\Phi - \Theta^{-1}\Psi F_2)^{-1}(\Theta^{-1}\Psi)H$$
$$\text{is nonsingular and diagonal,} \tag{3.35}$$

and

$$\left[\begin{array}{c} E_{22} \\ E_{32} \end{array}\right]^{-1} \left[\begin{array}{c} A_{22}+B_2F_2 \\ A_{32} \end{array}\right] = \Theta^{-1}\Phi + \Theta^{-1}\Psi F_2 \text{ is stable.} \tag{3.36}$$

Therefore, from Theorem 3.1, we get conditions (3.2) and (3.3). Moreover, $\mathcal{B}_{21}$ is nonsingular and F_2 and H satisfy (3.4); thus (3.25) is also true, which, along with (3.36), gives us condition (3.27).

In the following we show that $\mathcal{V}_{21}$ is of full column rank. In fact, we have from (3.35), (3.36), and Theorem 1.2(ii) that

$$\bigcap_{i=1}^{m} \mathcal{R}^{\star}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\tilde{c}_i)) \subset \mathcal{V}^{\star}_{stab}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(C_2)) = \{0\}. \tag{3.37}$$

However,

$$\mathcal{R}^{\star}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\left[\begin{array}{cc} I_{m-1} & 0 \end{array}\right] C_2)) \subset \bigcap_{i=1}^{m-1} \mathcal{R}^{\star}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\tilde{c}_i)).$$

So we get

$$\mathcal{R}^{\star}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\tilde{c}_m)) \bigcap \mathcal{R}^{\star}(\Theta^{-1}\Phi, \Theta^{-1}\Psi, \mathrm{Ker}(\left[\begin{array}{cc} I_{m-1} & 0 \end{array}\right] C_2)) = \{0\}. \tag{3.38}$$

From (3.38) and (3.32) we have

$$\begin{aligned} \{0\} &= \mathrm{Range}\left(\left[\begin{array}{c} I_{\mu_1} \\ 0 \end{array}\right]\right) \bigcap \mathrm{Range}\left(\mathcal{V}\left[\begin{array}{c} I_{\nu_1} \\ 0 \end{array}\right]\right) \\ &= \mathrm{Range}\left(\left[\begin{array}{c} I_{\mu_1} \\ 0 \end{array}\right]\right) \bigcap \mathrm{Range}\left(\left[\begin{array}{c} \mathcal{V}_{11} \\ \mathcal{V}_{21} \end{array}\right]\right), \end{aligned}$$

which is equivalent to saying that $\mathcal{V}_{21}$ is of full column rank because $\left[\begin{smallmatrix} \mathcal{V}_{11} \\ \mathcal{V}_{21} \end{smallmatrix}\right]$ is of full column rank.

Sufficiency. We have shown (3.34) because $(E^{-1}A, E^{-1}B)$ is controllable and B_2 is of full row rank. By the condensed form (2.8), we get

$$\mathrm{rank}(s\mathcal{E}_{32} - \mathcal{A}_{32}) = \tilde{\nu}_3 \qquad \forall s \in \mathbf{C}.$$

Because $\mathcal{B}_{21}$ is nonsingular, thus

$$\mathrm{rank}\left[\begin{array}{cc} s\mathcal{E}_{22} - \mathcal{A}_{22} & \mathcal{B}_{21} \\ s\mathcal{E}_{32} - \mathcal{A}_{32} & 0 \end{array}\right] = \tilde{\nu}_2 + \tilde{\nu}_3 = \nu_2 \qquad \forall s \in \mathbf{C},$$

i.e., $(\left[\begin{smallmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{smallmatrix}\right]^{-1}\left[\begin{smallmatrix} \mathcal{A}_{22} \\ \mathcal{A}_{32} \end{smallmatrix}\right], \left[\begin{smallmatrix} \mathcal{E}_{22} \\ \mathcal{E}_{32} \end{smallmatrix}\right]^{-1}\left[\begin{smallmatrix} \mathcal{B}_{21} \\ 0 \end{smallmatrix}\right])$ is controllable.

Since B_2 is of full row rank and (2.2) holds, Lemma 2.6 yields that there exists F_1 that satisfies (3.29). Moreover, $\mathcal{V}_{21}$ is of full column rank, which means that the choice of X satisfying (3.26) is equivalent to a stabilization problem by state feedback. Because (2.9) is true, by Lemma 2.6, the X in (3.26) is also well-defined. Thus, F in (3.28) is well-defined. For any F in (3.28), (3.25) is true and

$$P(sE - A - BF)Q = \left[\begin{array}{ccc} sE_{11} - A_{11} - B_1F_1 & sE_{12} - A_{12} - B_1F_2 \\ 0 & sE_{22} - A_{22} - B_2F_2 \\ 0 & sE_{32} - A_{32} \end{array}\right], \tag{3.39}$$

where the orthogonal matrices P, Q are determined in Theorem 2.1. Hence, the stability of $E^{-1}(A+BF)$ follows directly from (3.29), (3.27), and (3.26).

On the other hand, using (3.39), the condensed form (2.1), and Theorem 3.1, we have that for any pair (F, H) in (3.28), $C(sE-A-BF)^{-1}BH$ is nonsingular and diagonal. ∎

REMARK 2. *When $m=2$, then the condition* (3.3) *becomes $\mathcal{B}_{21}\neq 0$. In this case,*

$$\operatorname{rank}_g\begin{bmatrix} s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\mathcal{F} & \mathcal{B}_{21}\\ s\mathcal{E}_{32}-\mathcal{A}_{32} & 0\\ \mathcal{C}_{12} & 0\end{bmatrix}=\mu_2+1,\ \mathcal{C}_{12}\begin{bmatrix} s\mathcal{E}_{22}-\mathcal{A}_{22}-\mathcal{B}_{21}\mathcal{F}\\ s\mathcal{E}_{32}-\mathcal{A}_{32}\end{bmatrix}^{-1}\begin{bmatrix}\mathcal{B}_{21}\\0\end{bmatrix}\neq 0\qquad \forall \mathcal{F}\in\mathbf{R}^{1\times\nu_2};$$

i.e., (3.3) *is true for any $\mathcal{F}\in\mathbf{R}^{1\times\nu_2}$ and $\mathcal{H}\in\mathbf{R}$ with $\mathcal{H}\neq 0$ in* (3.4). *Hence, we can take $\mathcal{H}=1$ and choose any $\mathcal{F}$ satisfying* (3.27) *when we construct the desired feedback matrices F and H using Theorem* 3.2.

REMARK 3. *Theorem* 3.1 *cannot be proved based on the geometric approach. It is a result of using appropriate numerical linear algebra technique. Consequently, Theorem* 3.2 *cannot be proved only by using the geometric approach; its successful proof involves a combination of the geometric approach and numerical linear algebra technique.*

4. A complete numerical algorithm. Obviously, Theorem 3.2 can lead directly to a numerically reliable algorithm using only orthogonal transformations to verify the solvability of the RRDPS for system (1.1). In the following we consider the numerical computation of the desired feedback matrices F and H.

Let the QR factorization of $\begin{bmatrix}\mathcal{B}_{21} & \mathcal{B}_{22}\end{bmatrix}$ in the condensed form (2.8) be

$$\begin{bmatrix}\mathcal{B}_{21} & \mathcal{B}_{22}\end{bmatrix}\mathcal{Z}=\begin{array}{c}\\ m-1\\ 1\end{array}\begin{array}{c}\begin{array}{cc} m-1 & 1\end{array}\\ \begin{bmatrix}\tilde{\mathcal{B}}_{21} & 0\end{bmatrix}\end{array},\quad \mathcal{Z}=\begin{array}{c}\\ m-1\\ 1\end{array}\begin{array}{c}\begin{array}{cc} m-1 & 1\end{array}\\ \begin{bmatrix}\mathcal{Z}_{11} & \mathcal{Z}_{12}\\ \mathcal{Z}_{21} & \mathcal{Z}_{22}\end{bmatrix}\end{array},\tag{4.1}$$

where $\tilde{\mathcal{B}}_{21}$ is nonsingular. Then, $\begin{bmatrix} I_{m-1} & 0\\ \mathcal{B}_{21} & \mathcal{B}_{22}\end{bmatrix}\mathcal{Z}=\begin{bmatrix}\mathcal{Z}_{11} & \mathcal{Z}_{12}\\ \tilde{\mathcal{B}}_{21} & 0\end{bmatrix}$. From the transposed version of Lemma 2.5 we have that $\mathcal{Z}_{22}\neq 0$ and

$$\mathcal{B}_{21}^{-1}\mathcal{B}_{22}=-\mathcal{Z}_{12}/\mathcal{Z}_{22}.\tag{4.2}$$

Thus, let $\mathcal{H}_{22}=\mathcal{Z}_{22}$; then the formula in (3.28) for matrix H is replaced by

$$H=W\begin{bmatrix}\mathcal{H} & \mathcal{Z}_{12}\\ 0 & \mathcal{Z}_{22}\end{bmatrix}.\tag{4.3}$$

Since $\mathcal{Z}_{12}, \mathcal{Z}_{22}$ are obtained by orthogonal transformations, we can construct the matrix H using orthogonal transformations completely, which can be implemented in numerically stable ways.

We continue to consider the computation of matrix F in (3.28). From (3.4), a simple calculation yields that

$$F_2=W\left(\begin{bmatrix}0 & \mathcal{F}\\ 0 & 0\end{bmatrix}+D_2\right)\mathcal{V}^TV^T,\tag{4.4}$$

where

$$D_2 = \begin{bmatrix} -\mathcal{B}_{21} & \\ & 1 \end{bmatrix}^{-1} \begin{bmatrix} I & \mathcal{B}_{22} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{A}_{21} & 0 \\ X\mathcal{V}_{21} - \Phi_{21}\mathcal{V}_{11}/\Psi_{22} & X\mathcal{V}_{22} - \Phi_{21}\mathcal{V}_{12}/\Psi_{22} \end{bmatrix}. \tag{4.5}$$

This implies that

$$F = \begin{bmatrix} F_1 & F_2 \end{bmatrix} Q^T = W \left(D + \begin{bmatrix} 0 & \mathcal{F} \\ 0 & 0 \end{bmatrix} \right) \left(Q \begin{bmatrix} I & \\ & V\mathcal{V} \end{bmatrix} \right)^T, \tag{4.6}$$

where

$$D = \begin{bmatrix} W^T F_1 & D_2 \end{bmatrix}. \tag{4.7}$$

Note that in the condensed form (2.1), B_2 is of full row rank, so we can get orthogonal matrix $\tilde{Q}$ by the QR factorization of B_2^T such that

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tilde{Q} = \begin{matrix} \\ \tilde{n}_2 \end{matrix} \begin{matrix} \begin{matrix} \tilde{n}_2 & m - \tilde{n}_2 \end{matrix} \\ \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & 0 \end{bmatrix} \end{matrix}, \tag{4.8}$$

where B_{21} is lower triangular and nonsingular. Now we compute an orthogonal matrix $\tilde{P}$ such that

$$\tilde{P} \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} = \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \begin{bmatrix} 0 \\ \tilde{B}_{21} \end{bmatrix}, \quad \tilde{P} = \begin{matrix} \\ \tilde{n}_2 \end{matrix} \begin{matrix} \begin{matrix} n_1 & \tilde{n}_2 \end{matrix} \\ \begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{21} & \tilde{P}_{22} \end{bmatrix} \end{matrix}, \tag{4.9}$$

where $\tilde{B}_{21}$ is nonsingular. Denote

$$\tilde{P} \begin{bmatrix} E_{11} & A_{11} & B_{12} \\ 0 & A_{21} & 0 \end{bmatrix} = \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \begin{matrix} \begin{matrix} n_1 & n_1 & m - \tilde{n}_2 \end{matrix} \\ \begin{bmatrix} \tilde{E}_{11} & \tilde{A}_{11} & \tilde{B}_{12} \\ \tilde{E}_{21} & \tilde{A}_{21} & \tilde{B}_{22} \end{bmatrix} \end{matrix}, \tag{4.10}$$

and

$$\tilde{Q}^T F_1 = \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix}, \quad F_{11} \in \mathbf{R}^{\tilde{n}_2 \times n_1}, \quad F_{21} \in \mathbf{R}^{(m-\tilde{n}_2) \times n_1}. \tag{4.11}$$

As an application of Lemma 2.5, we have that $\tilde{P}_{22}$ is nonsingular and

$$E_{11} = \tilde{P}_{11}^{-1}\tilde{E}_{11}, \quad A_{11} - B_{11}B_{21}^{-1}A_{21} = \tilde{P}_{11}^{-1}\tilde{A}_{11}, \quad B_{12} = \tilde{P}_{11}^{-1}\tilde{B}_{12}.$$

Furthermore, we have that $B_2F_1 = -A_{21}$ and

$$E_{11}^{-1}(A_{11} + B_1F_1) \text{ is stable}$$

is equivalent to

$$B_{21}F_{11} = -A_{21} \tag{4.12}$$

and

$$\operatorname{rank}(s\tilde{E}_{11} - \tilde{A}_{11} - \tilde{B}_{12}F_{21}) = n_1 \qquad \forall s \in \mathbf{C}/\mathbf{C}^-. \tag{4.13}$$

This means that F_{11} can be obtained by solving (4.12), and F_{21} can be computed by the numerically reliable methods in [15, 19]. Consequently, F_1 can be computed in a numerically reliable manner.

In order to compute D_2, assume that the QR factorization of $\left[\begin{smallmatrix}\mathcal{B}_{11}\\ \mathcal{B}_{21}\end{smallmatrix}\right]$ in the condensed form (2.8) is given by

$$\mathcal{Y}\begin{bmatrix}\mathcal{B}_{21}\\ \mathcal{B}_{21}\end{bmatrix}=\begin{matrix}\nu_1\\ \tilde{\nu}_2\end{matrix}\begin{bmatrix}0\\ \tilde{\mathcal{B}}_{21}\end{bmatrix},\quad \mathcal{Y}=\begin{matrix}\nu_1\\ \tilde{\nu}_2\end{matrix}\begin{matrix}\begin{bmatrix}\mathcal{Y}_{11} & \mathcal{Y}_{12}\\ \mathcal{Y}_{21} & \mathcal{Y}_{22}\end{bmatrix}\\ \nu_1\quad \tilde{\nu}_2\end{matrix}. \tag{4.14}$$

Since $\mathcal{B}_{21}$ is nonsingular, $\tilde{\mathcal{B}}_{21}$ is also nonsingular. Denote

$$\mathcal{Y}\begin{bmatrix}\mathcal{E}_{11} & \mathcal{A}_{11} & \mathcal{B}_{12}\\ 0 & \mathcal{A}_{21} & \mathcal{B}_{22}\end{bmatrix}=\begin{matrix}\nu_1\\ \tilde{\nu}_2\end{matrix}\begin{matrix}\begin{bmatrix}\tilde{\mathcal{E}}_{11} & \tilde{\mathcal{A}}_{11} & \tilde{\mathcal{B}}_{12}\\ \tilde{\mathcal{E}}_{21} & \tilde{\mathcal{A}}_{21} & \tilde{\mathcal{B}}_{22}\end{bmatrix}\\ \nu_1\quad \nu_1\quad 1\end{matrix}. \tag{4.15}$$

By Lemma 2.5 we have that $\mathcal{Y}_{11}$ and $\tilde{\mathcal{E}}_{11}$ are nonsingular and

$$\mathcal{E}_{11}=\mathcal{Y}_{11}^{-1}\tilde{\mathcal{E}}_{11},\quad \mathcal{A}_{11}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{A}_{21}=\mathcal{Y}_{11}^{-1}\tilde{\mathcal{A}}_{11},\quad \mathcal{B}_{12}-\mathcal{B}_{11}\mathcal{B}_{21}^{-1}\mathcal{B}_{22}=\mathcal{Y}_{11}^{-1}\tilde{\mathcal{B}}_{12}.$$

Hence, (3.26) is equivalent to

$$\tilde{\mathcal{E}}_{11}^{-1}(\tilde{\mathcal{A}}_{11}-\tilde{\mathcal{B}}_{12}\Phi_{21}\mathcal{V}_{11}/\Psi_{22}+\tilde{\mathcal{B}}_{12}X\mathcal{V}_{21}) \text{ is stable.}$$

Thus, we only need to compute Y by the numerically reliable methods in [19, 15] such that

$$\operatorname{rank}(s\tilde{\mathcal{E}}_{11}-(\tilde{\mathcal{A}}_{11}-\tilde{\mathcal{B}}_{12}\Phi_{21}\mathcal{V}_{11}/\Psi_{22})-\tilde{\mathcal{B}}_{12}Y)=\nu_1\qquad \forall s\in\mathbf{C}/\mathbf{C}^- \tag{4.16}$$

and then get X by solving the equation

$$X\mathcal{V}_{21}=Y. \tag{4.17}$$

Obviously, the term

$$\begin{bmatrix}I & \mathcal{B}_{22}\\ & 1\end{bmatrix}\begin{bmatrix}\mathcal{A}_{21} & 0\\ X\mathcal{V}_{21}-\Phi_{21}\mathcal{V}_{11}/\Psi_{22} & X\mathcal{V}_{22}-\Phi_{21}\mathcal{V}_{12}/\Psi_{22}\end{bmatrix}$$

can be computed via a numerically reliable way and D_2 can be obtained by solving the equation

$$\begin{bmatrix}-\mathcal{B}_{21} & \\ & 1\end{bmatrix}D_2=\begin{bmatrix}I_{m-1} & \mathcal{B}_{22}\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathcal{A}_{21} & 0\\ X\mathcal{V}_{21}-\Phi_{21}\mathcal{V}_{11}/\Psi_{22} & X\mathcal{V}_{22}-\Phi_{21}\mathcal{V}_{12}/\Psi_{22}\end{bmatrix}. \tag{4.18}$$

The discussion above shows that D can be computed in a numerically reliable way. Furthermore, the rounding errors in computing the term D have no influence on the computation of $\mathcal{F}$. Therefore, F can be computed via a numerically reliable method.

We are now ready to present our numerical algorithm for the RRDPS in which Remark 2 is taken into account.

ALGORITHM 1.
Input: $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, *and* $C \in \mathbf{R}^{m\times n}$.
Output: *Matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *solving the RRDPS.*
Step 0. *Set*

$$F_0 \in \mathbf{R}^{m\times n}, \quad H_0 \in \mathbf{R}^{m\times m}, \quad F_0 := 0, \quad H_0 := 0, \quad W_0 = I_m, \quad Q_0 = I_n.$$

Step 1. *Compute the condensed forms* (2.1) *and* (2.5). *If* $\tilde{\mu}_2 \neq 1$, *then print "The RRDPS is unsolvable" and stop. Otherwise, compute the condensed form* (2.8).
Step 2. *Compute the QR factorizations of* $\mathcal{B}_{21}$ *and* $\mathcal{V}_{21}$. *If* $\mathcal{B}_{21}$ *is singular or* $\mathcal{V}_{21}$ *is not of full column rank, then print "The RRDPS is unsolvable" and stop. Otherwise, perform* (4.1), (4.8)–(4.10), (4.14), *and* (4.15).
Step 3. *Compute* F_{21} *and* Y *using the methods in* [19, 15] *such that* (4.13) *and* (4.16) *hold, respectively.*
Step 4. *Solve* (4.12) *and* (4.17) *by QR factorization to get* F_{11} *and* X. *Furthermore, compute* D_2 *by solving* (4.18) *using the QR factorization of* $\mathcal{B}_{21}$, *and get* F_1 *by* (4.11). *Then compute* D *by* (4.7).
Step 5. *Set*

$$W_0 := W_0 \begin{bmatrix} W & \\ & I \end{bmatrix}, \quad \hat{Q}_0 = Q \begin{bmatrix} I & \\ & V\mathcal{V} \end{bmatrix}, \quad Q_0 := Q_0 \begin{bmatrix} I & \\ & \hat{Q}_0 \end{bmatrix},$$

$$F_0 := \begin{bmatrix} W & \\ & I \end{bmatrix}^T F_0 \begin{bmatrix} I & \\ & \hat{Q}_0 \end{bmatrix} + \begin{bmatrix} 0 & D \\ 0 & 0 \end{bmatrix},$$

$$H_0 := \begin{bmatrix} W & \\ & I \end{bmatrix}^T H_0 + \begin{array}{c} \begin{matrix} m-1 & 1 \end{matrix} \\ \begin{bmatrix} 0 & \mathcal{Z}_{12} & 0 \\ 0 & \mathcal{Z}_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}.$$

If $m = 2$, *go to Step* 6. *Otherwise, if* $m > 2$, *set*

$$sE - A := \begin{bmatrix} s\mathcal{E}_{22} - \mathcal{A}_{22} \\ s\mathcal{E}_{32} - \mathcal{A}_{32} \end{bmatrix}, \quad B := \begin{bmatrix} \mathcal{B}_{21} \\ 0 \end{bmatrix}, \quad C := \mathcal{C}_{12}, \quad n := \nu_2, \quad m := m - 1,$$

and go to Step 1.
Step 6. *If* $\mathcal{B}_{21} = 0$, *then print "The RRDPS is unsolvable" and stop. Otherwise, compute* $\mathcal{F}$ *using the methods in* [19, 15] *such that* (3.27) *holds. Then compute* F *and* H *by*

$$F := W_0 \left(F_0 + \begin{bmatrix} 0 & \mathcal{F} \\ 0 & 0 \end{bmatrix} \right) Q_0^T, \quad H := W_0 \left(H_0 + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \right).$$

Output F *and* H.

In the following we present a numerical example using Algorithm 1. In this example, the original data matrices E, A, B, and C were generated using the MATLAB command `randn` and all subsequent calculations were carried out using MATLAB 5.3 on an HP 712/80 workstation with IEEE standard, i.e., the machine accuracy is about $\epsilon \cong 10^{-16}$. Due to the length limitation of this paper, we give only the final result. The details of all intermediate computations can be obtained from the authors.

EXAMPLE 2. *Suppose there is a system of the form* (1.1) *with*

$$E = \begin{bmatrix} 1.3894941016041 & 2.8773508670257 & 0.2254236217237 & -0.1924005899501 & -5.2172271998520 & 1.1979322072823 \\ 0.4561230628610 & 0.8529343697912 & -0.1203480101854 & 0.6219233511796 & -1.8532903681842 & 0.8187885350812 \\ -0.0583279014123 & -0.1451475204220 & -0.0891807358259 & 0.1020733874504 & -0.0315731413154 & 0.0106538391242 \\ -0.5160868552020 & -1.0743534709506 & -0.0651803246756 & 0.3078528106434 & 1.6646937680349 & -0.0962126616017 \\ -1.0388915782354 & -3.2013267952723 & -0.8511366013008 & 1.4981997899661 & 5.6599971844308 & -1.7967594859402 \\ -0.3408357784176 & -0.6642282085079 & -0.5843332003168 & 0.2574681041184 & 1.9048019648284 & -0.8776049968164 \end{bmatrix},$$

$$A = \begin{bmatrix} 3.7618688503302 & 1.4534396306785 & -2.5928123960916 & 4.0507325536819 & -6.1976217923044 & -2.5488686731466 \\ 0.7439564521050 & 0.1275500731638 & -0.9710266379526 & 1.4505878792942 & -0.8750848524142 & -0.6461986468818 \\ 1.1582190944175 & 1.5535353319471 & -0.3959391178398 & -0.3693076185632 & -3.6897842316839 & -0.4961305538884 \\ -1.3442823973504 & 0.2997769470154 & 1.3705594260599 & -2.1313731454811 & 1.3823906057240 & 1.5909923699461 \\ -3.2286388077973 & -1.3335790375774 & 1.7665042794420 & -2.6646848942675 & 5.0827326482614 & 2.1847489726994 \\ -1.4621882316854 & -4.7908980622290 & -2.0774176289798 & 2.7166042000452 & 7.1727103453469 & -2.773993179975 \end{bmatrix},$$

$$B = \begin{bmatrix} 2.12055770435426 & 1.00661557403426 & 1.95683404123065 \\ 0.88892004016864 & 0.33241927378497 & 0.74071011016848 \\ 0.31163435340972 & 0.37071970874182 & 0.45009376598337 \\ -0.87363963677912 & -0.49683601091227 & -0.81737747267883 \\ -1.32613521275035 & -0.78528412691089 & -1.33784251083343 \\ 0.62329434673124 & 0.61139637181498 & 0.51224198001395 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.0092738121968 & -0.0158138239954 & -0.0062525703881 & 0.0206707842160 & 0.0023945971994 & 0.0188773963767 \\ -0.4722469002929 & -1.1665817063750 & -0.4527399982410 & 0.5968064514640 & 2.0109513620752 & -0.5336352249465 \\ 0.1412527284128 & 0.4070293405785 & -0.2195263818174 & -0.0564929866156 & -0.1932345015239 & -0.1775481769430 \end{bmatrix}.$$

Now we perform Algorithm 1 *and get the desired* F *and* H *for solving the RRDPS, as follows:*

$$H = \begin{bmatrix} 0.54803522302892 & -0.20199465930860 & -0.70998878873329 \\ 0.14814298152094 & -0.65998549562213 & -0.24482058326485 \\ -0.82323207623715 & 0.72361405678664 & 0.66028690876232 \end{bmatrix},$$

$$F = \begin{bmatrix} 1.9687489604506 & 1.0905360571177 & 2.5519292168400 & -1.4564243641146 & -16.1322181168766 & 1.4110078104903 \\ 4.8457993652431 & 0.5913364485116 & -1.1629494431679 & 5.0189870624541 & -5.1724262490250 & -2.6898922595821 \\ -6.9774481790293 & -0.0300578156656 & 1.7181386312414 & -6.0839168311771 & 19.9810453051219 & 4.7575592152078 \end{bmatrix}.$$

We verify that the (F, H) *above solves the RRDPS.*

If we only choose the first 4 decimal digits by rounding off the other decimal digits in the matrices E, A, B, and C in Example 2, then we get system

$$\tilde{E}\dot{x} = \tilde{A}x + \tilde{B}u, \quad y = \tilde{C}x \tag{4.19}$$

with matrices

$$\tilde{E} = \begin{bmatrix} 1.3895 & 2.8774 & 0.2254 & -0.1924 & -5.2172 & 1.1979 \\ 0.4561 & 0.8529 & -0.1203 & 0.6219 & -1.8533 & 0.8188 \\ -0.0583 & -0.1451 & -0.0892 & 0.1021 & -0.0316 & 0.0107 \\ -0.5161 & -1.0744 & -0.0652 & 0.3079 & 1.6647 & -0.0962 \\ -1.0389 & -3.2013 & -0.8511 & 1.4982 & 5.6600 & -1.7968 \\ -0.3408 & -0.6642 & -0.5843 & 0.2575 & 1.9048 & -0.8776 \end{bmatrix},$$

$$\tilde{A} = \begin{bmatrix} 3.7619 & 1.4534 & -2.5928 & 4.0507 & -6.1976 & -2.5489 \\ 0.7440 & 0.1276 & -0.9710 & 1.4506 & -0.8751 & -0.6462 \\ 1.1582 & 1.5535 & -0.3959 & -0.3693 & -3.6898 & -0.4961 \\ -1.3443 & 0.2998 & 1.3706 & -2.1314 & 1.3824 & 1.5910 \\ -3.2286 & -1.3336 & 1.7665 & -2.6647 & 5.0827 & 2.1847 \\ -1.4622 & -4.7909 & -2.0774 & 2.7166 & 7.1727 & -2.7740 \end{bmatrix},$$

$$\tilde{B} = \begin{bmatrix} 2.1206 & 1.0066 & 1.9568 \\ 0.8889 & 0.3324 & 0.7407 \\ 0.3116 & 0.3707 & 0.4501 \\ -0.8736 & -0.4968 & -0.8174 \\ -1.3261 & -0.7853 & -1.3378 \\ 0.6233 & 0.6114 & 0.5122 \end{bmatrix},$$

$$\tilde{C} = \begin{bmatrix} -0.0093 & -0.0158 & -0.0063 & 0.0207 & 0.0024 & 0.0189 \\ -0.4722 & -1.1666 & -0.4527 & 0.5968 & 2.0110 & -0.5336 \\ 0.1413 & 0.4070 & -0.2195 & -0.0565 & -0.1932 & -0.1775 \end{bmatrix}.$$

However, in this case the RRDPS for system (4.19) is unsolvable because $\mathcal{V}_{21}$ in the case $n = 6, m = 3$ is not of full column rank.[1] The poles and invariant zeros of system (1.1) are

$$\Lambda = \begin{bmatrix} 5.41814291005592 - 3.23024976667490i \\ 5.41814291005593 + 3.23024976667490i \\ -1.87864288202711 \\ 1.56316607778768 \\ 0.36477560252640 - 0.32084261817080i \\ 0.36477560252640 + 0.32084261817080i \end{bmatrix}, \quad \mathcal{O} = \begin{bmatrix} -1.88406066392001 \\ 1.97486631764478 \\ 1.97486631764458 \end{bmatrix},$$

respectively, while the poles and invariant zeros of system (4.19) are

$$\tilde{\Lambda} = \begin{bmatrix} 5.42133770871955 - 3.23587512736390i \\ 5.42133770871955 + 3.23587512736390i \\ -1.88035964076856 \\ 1.56137864611542 \\ 0.36362154553414 - 0.32258108140166i \\ 0.36362154553414 + 0.32258108140166i \end{bmatrix}, \quad \tilde{\mathcal{O}} = \begin{bmatrix} -1.91123137997939 \\ 1.96608208697235 \\ 2.02959696122931 \end{bmatrix},$$

respectively. Since

$$\|\Lambda - \tilde{\Lambda}\|_2 / \|\Lambda\|_2 = 1.0703 \times 10^{-3}, \quad \|\mathcal{O} - \tilde{\mathcal{O}}\|_2 / \|\mathcal{O}\|_2 = 1.8324 \times 10^{-2},$$

therefore the structures of systems (1.1) and (4.19) are totally different. Note that

$$\|E - \tilde{E}\|_2 / \|E\|_2 = 1.0866 \times 10^{-5}, \quad \|A - \tilde{A}\|_2 / \|A\|_2 = 7.0301 \times 10^{-6},$$
$$\|B - \tilde{B}\|_2 / \|B\|_2 = 2.0461 \times 10^{-5}, \quad \|C - \tilde{C}\|_2 / \|C\|_2 = 4.8808 \times 10^{-5},$$

so the above truncation errors are large enough to change the structure of system (1.1); equivalently, we cannot regard system (4.19) as an appropriate approximation of system (1.1). Consequently, such truncation errors are large enough to perturb the decouplable system (1.1) with stability to a nondecouplable system (4.19) with stability! This result highlights some important points worth noting for the RRDPS:

[1] This interesting observation is due to Prof. M. Malabre!

- In general, engineers may like to truncate the data obtained from experiments and work with these truncated data. However, when we deal with the RRDPS, we must avoid any truncation of the data, and the input data should be as accurate as possible, otherwise, the results may be in error!
- The various row by row decoupling problems, including the RRDPS, are rather sensitive to the input data. This is an inherent property of the row by row decoupling problems, *which does not depend on the numerical methods used*! Therefore, it is interesting to study the distance between a given decouplable system (with or without stability) and the set of all nondecouplable systems (with or without stability). There are no results available for this topic yet.
- Because the RRDPS is a rather sensitive problem, it requires reliable numerical methods. Generally, the error during the numerical computation may be interpreted into the perturbation in the input data, so if the used numerical methods are not reliable, the errors produced during the computations may increase to a certain level such that the perturbed system becomes a nondecouplable one.
- Algorithm 1 is implemented using orthogonal transformations, and hence it is reliable. This point has been verified through a step response analysis. For the result in Example 2, it has been found that the response to steps or sinusoidal inputs applied on each separate control channel effectively gives terms on the off-diagonal parts which are about 10^{-15}—almost the same as the machine accuracy.[2]

5. Concluding remarks. We have studied the RRDPS for system (1.1). We first proved a reduction property for the RRDPS based on the condensed forms (2.1), (2.5), and (2.8) using Theorems 1.2 and 3.1 and then developed the numerically reliable Algorithm 1.

The condensed forms (2.1), (2.5), and (2.8) are all based on orthogonal transformations, and the main ingredients of their constructions are the generalized upper triangular forms of matrix pencils (see Lemma A.1 in the appendix) and QR factorization with pivoting. Therefore, our results lead to numerically reliable methods for solving the RRDPS using existing numerical algebra software such as MATLAB.

We should mention that row by row decoupling is not always possible, and thus one can resort to triangular decoupling, as follows, because it requires less restrictive conditions.

Problem 5.1. Given a system of the form (1.1), find matrices $F \in \mathbf{R}^{m\times n}$ and $H \in \mathbf{R}^{m\times m}$ such that $E^{-1}(A+BF)$ is stable and $T(s) := C(sE-A-BF)^{-1}BH$ is lower triangular and nonsingular.

In the sequel [6], we will study the triangular decoupling problem. We will show that the numerical linear algebra technique used in the present paper can be extended not only to derive explicit solvability conditions but also to parameterize all solutions for the triangular decoupling problem, although we cannot expect that similar parameterization of all solutions can be obtained for the RRDPS.

Appendix. Before we construct the condensed forms (2.1) and (2.5), we recall the generalized upper triangular form of matrix pencils and QR factorization with column pivoting.

[2]This verification was done by Dr. Jean-Francois Camalt in IRCCyN of Ecole Centrale De Nantes.

It is well known that any matrix pencil $sE - A$ can be transformed into its generalized upper triangular (GUPTRI) form under orthogonal transformations. This GUPTRI is well studied in [8, 9, 18], and numerically stable algorithms are available via www.netlib.org/linalg/guptri. There also exists a MATLAB MEX-interface for the GUPTRI software.

LEMMA A.1 (see [8, 9, 18]). *Given a matrix pencil (E, A), $E, A \in \mathbf{R}^{l\times n}$ there exist orthogonal matrices $P \in \mathbf{R}^{l\times l}$, $Q \in \mathbf{R}^{n\times n}$ such that (PEQ, PAQ) are in the following GUPTRI:*

$$P(sE-A)Q = \begin{array}{c} \\ l_1 \\ n_2 \\ n_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{cccc} n_1 & n_2 & n_3 & n_4 \end{array} \\ \begin{bmatrix} sE_{11}-A_{11} & sE_{12}-A_{12} & sE_{13}-A_{13} & sE_{14}-A_{14} \\ 0 & sE_{22}-A_{22} & sE_{23}-A_{23} & sE_{24}-A_{24} \\ 0 & 0 & sE_{33}-A_{33} & sE_{34}-A_{34} \\ 0 & 0 & 0 & sE_{44}-A_{44} \end{bmatrix} \end{array}, \tag{A.1}$$

where

$$\begin{aligned} &\operatorname{rank}(E_{11}) = l_1, \quad \operatorname{rank}(E_{22}) = n_2, \quad \operatorname{rank}(E_{44}) = n_4, \\ &\operatorname{rank}(sE_{11} - A_{11}) = l_1, \quad \operatorname{rank}(sE_{33} - A_{33}) = n_3, \quad \operatorname{rank}(sE_{44} - A_{44}) = n_4 \quad \forall s \in \mathbf{C}. \end{aligned}$$

It is also well known that any matrix $A \in \mathbf{R}^{m\times n}$ can be factorized as

$$UA = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} \Pi, \tag{A.2}$$

where U and Π are orthogonal and permutation matrices, respectively, and R_1 is nonsingular and upper triangular. The factorization (A.2) is called the QR factorization of A with column pivoting. Furthermore, if we denote $R = \begin{bmatrix} R_1 & R_2 \end{bmatrix} \Pi$, then R is of full row rank and $UA = \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right]$. Moreover, if we perform QR factorization of A^T with column pivoting, we can get the orthogonal matrix V such that $V^T A^T = \left[\begin{smallmatrix} \mathcal{R}^T \\ 0 \end{smallmatrix}\right]$ with $\mathcal{R}^T$ being of full row rank. Hence, V satisfies $AV = \begin{bmatrix} \mathcal{R} & 0 \end{bmatrix}$, where $\mathcal{R}$ is of full column rank.

In this appendix, for any matrix M, $\mathcal{S}_\infty(M)$ denotes a full column rank matrix whose columns span the null space of M and $\mathcal{T}_\infty(M) = \mathcal{S}_\infty(M^T)$.

Now we give the numerical constructions of the condensed forms (2.1) and (2.5).

The construction of the condensed form (2.1). We construct the condensed form (2.1) by the following algorithm.

ALGORITHM 2.

Input: *$E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, and $C \in \mathbf{R}^{m\times n}$ with E nonsingular.*

Output: *Orthogonal matrices $P, Q \in \mathbf{R}^{n\times n}$ and the condensed form* (2.1).

Step 1. *Perform the row compression of B and the column compression of C, and then perform the GUPTRI form of the pencil $(\mathcal{T}_\infty^T(B)E\mathcal{S}_\infty(C), \mathcal{T}_\infty^T(B)A\mathcal{S}_\infty(C))$ to get orthogonal matrices P_1 and Q_1 such that*

$$P_1(sE-A)Q_1 =: \begin{array}{c} \\ \tilde{n}_1^{(1)} \\ \tilde{n}_2^{(1)} \\ n_2^{(1)} \\ \tilde{n}_4^{(1)} \end{array} \begin{array}{c} \begin{array}{cccc} n_1^{(1)} & n_2^{(1)} & n_3^{(1)} & n_4^{(1)} \end{array} \\ \begin{bmatrix} sE_{11}^{(1)}-A_{11}^{(1)} & sE_{12}^{(1)}-A_{12}^{(1)} & sE_{13}^{(1)}-A_{13}^{(1)} & sE_{14}^{(1)}-A_{14}^{(1)} \\ sE_{21}^{(1)}-A_{21}^{(1)} & sE_{22}^{(1)}-A_{22}^{(1)} & sE_{23}^{(1)}-A_{23}^{(1)} & sE_{24}^{(1)}-A_{24}^{(1)} \\ 0 & sE_{32}^{(1)}-A_{32}^{(1)} & sE_{33}^{(1)}-A_{33}^{(1)} & sE_{34}^{(1)}-A_{34}^{(1)} \\ 0 & 0 & sE_{43}^{(1)}-A_{43}^{(1)} & sE_{44}^{(1)}-A_{44}^{(1)} \end{bmatrix} \end{array},$$

$$P_1B =: \begin{matrix}\tilde{n}_1^{(1)}\\ \tilde{n}_2^{(1)}\\ n_2^{(1)}\\ \tilde{n}_4^{(1)}\end{matrix}\begin{bmatrix} B_1^{(1)}\\ 0\\ 0\\ 0\end{bmatrix}, \quad CQ_1 =: \begin{matrix}\end{matrix}\begin{matrix} n_1^{(1)} & n_2^{(1)} & n_3^{(1)} & n_4^{(1)}\end{matrix}\\ \begin{bmatrix} 0 & 0 & 0 & C_4^{(1)}\end{bmatrix},$$

where

$$\operatorname{rank}(B_1^{(1)}) = \tilde{n}_1^{(1)}, \quad \operatorname{rank}(E_{21}^{(1)}) = \tilde{n}_2^{(1)}, \quad \operatorname{rank}(C_4^{(1)}) = n_4^{(1)}, \quad \operatorname{rank}(E_{32}^{(1)}) = n_2^{(1)}, \tag{A.3}$$

$$\operatorname{rank}(sE_{21}^{(1)} - A_{21}^{(1)}) = \tilde{n}_2^{(1)}, \quad \operatorname{rank}(sE_{43}^{(1)} - A_{43}^{(1)}) = n_3^{(1)} \qquad \forall s \in \mathbf{C}. \tag{A.4}$$

Step 2. *Perform the QZ method* [11] *to get orthogonal matrices* P_2 *and* Q_2 *such that*

$$P_2(sE_{32}^{(1)} - A_{32}^{(1)})Q_2 =: \begin{matrix} n_2^{(2)}\\ n_2^{(1)} - n_2^{(2)}\end{matrix}\begin{matrix}\end{matrix}\begin{matrix} n_2^{(2)} & n_2^{(1)} - n_2^{(2)}\end{matrix}\\ \begin{bmatrix} sE_{32}^{(2)} - A_{32}^{(2)} & sE_{33}^{(2)} - A_{33}^{(2)}\\ 0 & sE_{43}^{(2)} - A_{43}^{(2)}\end{bmatrix},$$

where all eigenvalues of $(E_{32}^{(2)})^{-1}A_{32}^{(2)}$ *and* $(E_{43}^{(2)})^{-1}A_{43}^{(2)}$ *are in* $\mathbf{C}^-$ *and* $\mathbf{C}/\mathbf{C}^-$*, respectively, i.e.,*

$$\operatorname{rank}(sE_{32}^{(2)} - A_{32}^{(2)}) = n_2^{(2)} \qquad \forall s \in \mathbf{C}/\mathbf{C}^-, \tag{A.5}$$

$$\operatorname{rank}(sE_{43}^{(2)} - A_{43}^{(2)}) = n_2^{(1)} - n_2^{(2)} \qquad \forall s \in \mathbf{C}^-. \tag{A.6}$$

Set

$$P_2\begin{bmatrix} sE_{33}^{(1)} - A_{33}^{(1)} & sE_{34}^{(1)} - A_{34}^{(1)}\end{bmatrix} =: \begin{matrix} n_2^{(2)}\\ n_2^{(1)} - n_2^{(2)}\end{matrix}\begin{matrix}\end{matrix}\begin{matrix} n_3^{(1)} & n_4^{(1)}\end{matrix}\\ \begin{bmatrix} sE_{34}^{(2)} - A_{34}^{(2)} & sE_{35}^{(2)} - A_{35}^{(2)}\\ sE_{44}^{(2)} - A_{44}^{(2)} & sE_{45}^{(2)} - A_{45}^{(2)}\end{bmatrix},$$

$$\begin{bmatrix} sE_{12}^{(1)} - A_{12}^{(1)}\\ sE_{22}^{(1)} - A_{22}^{(1)}\end{bmatrix} Q_2 =: \begin{matrix}\tilde{n}_1^{(1)}\\ \tilde{n}_2^{(1)}\end{matrix}\begin{matrix}\end{matrix}\begin{matrix} n_2^{(1)} & n_2^{(1)} - n_2^{(2)}\end{matrix}\\ \begin{bmatrix} sE_{12}^{(2)} - A_{12}^{(2)} & sE_{13}^{(2)} - A_{13}^{(2)}\\ sE_{22}^{(2)} - A_{22}^{(2)} & sE_{23}^{(2)} - A_{23}^{(2)}\end{bmatrix}.$$

Step 3. *Note that* E *is nonsingular, so*

$$\begin{bmatrix} E_{11}^{(1)} & E_{12}^{(2)}\\ E_{21}^{(1)} & E_{22}^{(2)}\\ 0 & E_{32}^{(2)}\end{bmatrix}$$

is of full column rank. Hence, we can perform a QR factorization of

$$\begin{bmatrix} E_{11}^{(1)} & E_{12}^{(2)}\\ E_{21}^{(1)} & E_{22}^{(2)}\\ 0 & E_{32}^{(2)}\end{bmatrix}$$

to get the orthogonal matrix P_3 *such that*

$$P_3\begin{bmatrix} sE_{11}^{(1)} - A_{11}^{(1)} & sE_{12}^{(2)} - A_{12}^{(2)}\\ sE_{21}^{(1)} - A_{21}^{(1)} & sE_{22}^{(2)} - A_{22}^{(2)}\\ 0 & sE_{32}^{(2)} - A_{32}^{(2)}\end{bmatrix} =: \begin{matrix} n_1\\ \tilde{n}_2\end{matrix}\begin{matrix}\end{matrix}\begin{matrix} n_1\end{matrix}\\ \begin{bmatrix} sE_{11} - A_{11}\\ -A_{21}\end{bmatrix},$$

where

$$n_1 = n_1^{(1)} + n_2^{(2)}, \quad \tilde{n}_2 = \tilde{n}_1^{(1)} + \tilde{n}_2^{(1)} + n_2^{(2)} - n_1, \quad \operatorname{rank}(E_{11}) = n_1. \tag{A.7}$$

Set

$$P_3 \begin{bmatrix} sE_{13}^{(2)} - A_{13}^{(2)} & sE_{13}^{(1)} - A_{13}^{(1)} & sE_{14}^{(1)} - A_{14}^{(1)} \\ sE_{23}^{(2)} - A_{23}^{(2)} & sE_{23}^{(1)} - A_{23}^{(1)} & sE_{24}^{(1)} - A_{24}^{(1)} \\ sE_{33}^{(2)} - A_{33}^{(2)} & sE_{33}^{(1)} - A_{33}^{(1)} & sE_{34}^{(1)} - A_{34}^{(1)} \end{bmatrix} =: \begin{matrix} \\ n_1 \\ \tilde{n}_2 \end{matrix} \begin{matrix} n_2 \\ \begin{bmatrix} sE_{12} - A_{12} \\ sE_{22} - A_{22} \end{bmatrix} \end{matrix},$$

$$P_3 \begin{bmatrix} B_1^{(1)} \\ 0 \\ 0 \end{bmatrix} =: \begin{matrix} n_1 \\ \tilde{n}_2 \end{matrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

$$\begin{bmatrix} sE_{43}^{(2)} - A_{43}^{(2)} & sE_{44}^{(2)} - A_{44}^{(2)} & sE_{45}^{(2)} - A_{45}^{(2)} \\ 0 & sE_{43}^{(1)} - A_{43}^{(1)} & sE_{44}^{(1)} - A_{44}^{(1)} \end{bmatrix} =: sE_{32} - A_{32},$$

$$\begin{bmatrix} 0 & 0 & C_4^{(1)} \end{bmatrix} =: C_2,$$

$$n_2 := n - n_1, \quad \tilde{n}_3 := n - n_1 - \tilde{n}_2.$$

By (A.3) *and* (A.7) *we have*

$$\begin{aligned} \operatorname{rank}(B_2) &= \operatorname{rank} \begin{bmatrix} E_{11} & B_1 \\ 0 & B_2 \end{bmatrix} - n_1 = \operatorname{rank} \begin{bmatrix} E_{11}^{(1)} & E_{12}^{(2)} & B_1^{(1)} \\ E_{21}^{(1)} & E_{22}^{(2)} & 0 \\ 0 & E_{32}^{(2)} & 0 \end{bmatrix} - n_1 \\ &= \tilde{n}_1^{(1)} + \tilde{n}_2^{(1)} + n_2^{(2)} - n_1 = n_2; \end{aligned}$$

equivalently, B_2 *is of full row rank. Furthermore, by* (A.3), (A.4), (A.5), *and* (A.6) *we get*

$$\begin{aligned} \operatorname{rank} \begin{bmatrix} sE_{11} - A_{11} & B_1 \\ -A_{21} & B_2 \end{bmatrix} &= \operatorname{rank} \begin{bmatrix} sE_{11}^{(1)} - A_{11}^{(1)} & sE_{12}^{(2)} - A_{12}^{(2)} & B_1^{(1)} \\ sE_{21}^{(1)} - A_{21}^{(1)} & sE_{22}^{(2)} - A_{22}^{(2)} & 0 \\ 0 & sE_{32}^{(2)} - A_{32}^{(2)} & 0 \end{bmatrix} \\ &= \tilde{n}_1^{(1)} + \tilde{n}_2^{(1)} + n_2^{(2)} = n_1 + n_2 \qquad \forall s \in \mathbf{C}/\mathbf{C}^- \end{aligned}$$

and

$$\begin{aligned} \operatorname{rank} \begin{bmatrix} sE_{32} - A_{32} \\ C_2 \end{bmatrix} &= \operatorname{rank} \begin{bmatrix} sE_{43}^{(2)} - A_{43}^{(2)} & sE_{44}^{(2)} - A_{44}^{(2)} & sE_{45}^{(2)} - A_{45}^{(2)} \\ 0 & sE_{43}^{(1)} - A_{43}^{(1)} & sE_{44}^{(1)} - A_{44}^{(1)} \\ 0 & 0 & C_4^{(1)} \end{bmatrix} \\ &= (n_2^{(1)} - n_2^{(2)}) + n_3^{(1)} + n_4^{(1)} = n_2 \qquad \forall s \in \mathbf{C}^-; \end{aligned}$$

i.e., the properties (2.2) *and* (2.3) *hold.*

Step 4. *Set*

$$P = \begin{bmatrix} P_3 & \\ & I_{\tilde{n}_3} \end{bmatrix} \begin{bmatrix} I_{\tilde{n}_1^{(1)} + \tilde{n}_2^{(1)}} & & \\ & P_2 & \\ & & I_{\tilde{n}_4^{(1)}} \end{bmatrix} P_1, \quad Q = Q_1 \begin{bmatrix} I_{n_1^{(1)}} & & \\ & Q_2 & \\ & & I_{n_3^{(1)} + n_4^{(1)}} \end{bmatrix}.$$

Then $(P(sE - A)Q, PB, CQ)$ *are in the condensed form* (2.1).

The construction of the condensed form (2.5). We construct the condensed form (2.5) by the following algorithm.

ALGORITHM 3.

Input: $\left[\begin{smallmatrix} E_{22} \\ E_{32} \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} A_{22} \\ A_{32} \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right]$, *and* C_2 *in the condensed form* (2.5).

Output: *Orthogonal matrices* U, V, W, *and the condensed form* (2.5).

Step 1. *Perform QR factorizations of* $\left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right]$ *and* $(\left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right] C_2)^T$ *with column pivoting, respectively, and then perform the GUPTRI of* $T_\infty^T(\left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right])\left[\begin{smallmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{smallmatrix}\right] S_\infty(\left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right] C_2)$ *to get the orthogonal matrices* U_1 *and* V *such that*

$$U_1 \begin{bmatrix} B_2 \\ 0 \end{bmatrix} =: \begin{matrix} \tilde{\mu}_1^{(1)} \\ \tilde{\mu}_2^{(1)} \\ \tilde{\mu}_3 \end{matrix} \begin{bmatrix} \Psi_1^{(1)} \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \end{bmatrix} C_2 V =: \begin{matrix} \mu_1 & \mu_2^{(1)} & \mu_3^{(1)} \\ \multicolumn{3}{c}{\begin{bmatrix} 0 & 0 & \xi_{23}^{(1)} \end{bmatrix}} \end{matrix},$$

$$U_1 \begin{bmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{bmatrix} V := \begin{matrix} \tilde{\mu}_1^{(1)} \\ \tilde{\mu}_2^{(1)} \\ \tilde{\mu}_3 \end{matrix} \begin{matrix} \mu_1 & \mu_2^{(1)} & \mu_3^{(1)} \\ \multicolumn{3}{c}{\begin{bmatrix} s\Theta_{11}^{(1)}-\Phi_{11}^{(1)} & s\Theta_{12}^{(1)}-\Phi_{12}^{(1)} & s\Theta_{13}^{(1)}-\Phi_{13}^{(1)} \\ s\Theta_{21}^{(1)}-\Phi_{21}^{(1)} & s\Theta_{22}^{(1)}-\Phi_{22}^{(1)} & s\Theta_{23}^{(1)}-\Phi_{23}^{(1)} \\ 0 & s\Theta_{32}^{(1)}-\Phi_{32}^{(1)} & s\Theta_{33}^{(1)}-\Phi_{33}^{(1)} \end{bmatrix}} \end{matrix},$$

where

$$\operatorname{rank}(\Psi_1^{(1)}) = \tilde{\mu}_1^{(1)}, \quad \operatorname{rank}(\xi_{23}^{(1)}) = \mu_3^{(1)}, \quad \operatorname{rank}(\Theta_{21}^{(1)}) = \tilde{\mu}_2^{(1)},$$
$$\operatorname{rank}_g(s\Theta_{32}^{(1)} - \Phi_{32}^{(1)}) = \mu_2^{(1)}, \operatorname{rank}(s\Theta_{21}^{(1)} - \Phi_{21}^{(1)}) = \tilde{\mu}_2^{(1)} \qquad \forall s \in \mathbf{C}.$$

Step 2. E *is nonsingular from the form* (2.1), *and* $\left[\begin{smallmatrix} E_{22} \\ E_{32} \end{smallmatrix}\right]$ *is also nonsingular; therefore* $\left[\begin{smallmatrix} \Theta_{11}^{(1)} \\ \Theta_{21}^{(1)} \end{smallmatrix}\right]$ *is of full column rank. Hence, perform the QR factorization of* $\left[\begin{smallmatrix} \Theta_{11}^{(1)} \\ \Theta_{21}^{(1)} \end{smallmatrix}\right]$ *with column pivoting to get the orthogonal matrix* U_2 *such that*

$$U_2 \begin{bmatrix} s\Theta_{11}^{(1)} - \Phi_{11}^{(1)} \\ s\Theta_{21}^{(1)} - \Phi_{21}^{(1)} \end{bmatrix} =: \begin{matrix} \mu_1 \\ \tilde{\mu}_2 \end{matrix} \begin{bmatrix} s\Theta_{11} - \Phi_{11} \\ -\Phi_{21} \end{bmatrix}$$

with Θ_{11} *nonsingular. Set*

$$U_2 \begin{bmatrix} s\Theta_{12}^{(1)} - \Phi_{12}^{(1)} & s\Theta_{13}^{(1)} - \Phi_{13}^{(1)} \\ s\Theta_{22}^{(1)} - \Phi_{22}^{(1)} & s\Theta_{23}^{(1)} - \Phi_{23}^{(1)} \end{bmatrix} =: \begin{matrix} \mu_1 \\ \tilde{\mu}_2 \end{matrix} \begin{matrix} \mu_2 \\ \multicolumn{1}{c}{\begin{bmatrix} s\Theta_{12} - \Phi_{12} \\ s\Theta_{22} - \Phi_{22} \end{bmatrix}} \end{matrix}, \quad U_2 \begin{bmatrix} \Psi_1^{(1)} \\ 0 \end{bmatrix} =: \begin{matrix} \mu_1 \\ \tilde{\mu}_2 \end{matrix} \begin{bmatrix} \Psi_1^{(2)} \\ \Psi_2^{(2)} \end{bmatrix},$$

$$\begin{bmatrix} s\Theta_{32}^{(1)} - \Phi_{32}^{(1)} & s\Theta_{33}^{(1)} - \Phi_{33}^{(1)} \end{bmatrix} =: s\Theta_{32} - \Phi_{32}, \quad \begin{bmatrix} 0 & \xi_{23}^{(1)} \end{bmatrix} =: \xi_{22}.$$

Step 3. *Note that* $\left[\begin{smallmatrix} \Theta_{11}^{(1)} & \Psi_1^{(1)} \\ \Theta_{21}^{(1)} & 0 \end{smallmatrix}\right]$ *is of full row rank, so* $\Psi_2^{(2)}$ *is also of full row rank. Therefore, by performing the QR factorization of* $(\Psi_2^{(2)})^T$ *with column pivoting, we can get the orthogonal matrix* W *such that*

$$\begin{bmatrix} \Psi_1^{(2)} \\ \Psi_2^{(2)} \end{bmatrix} W =: \begin{matrix} \mu_1 \\ \tilde{\mu}_2 \end{matrix} \begin{matrix} m-\tilde{\mu}_2 & \tilde{\mu}_2 \\ \multicolumn{2}{c}{\begin{bmatrix} \Psi_{11} & \Psi_{12} \\ 0 & \Psi_{22} \end{bmatrix}} \end{matrix}$$

with Ψ_{22} *nonsingular.*

Step 4. *Set*

$$U := \begin{bmatrix} U_2 & \\ & I \end{bmatrix} U_1, \quad \begin{bmatrix} I_{n_2-1} & 0 \end{bmatrix} C_2 V =: \begin{bmatrix} \mu_1 & \mu_2 \\ \Xi_{11} & \Xi_{12} \end{bmatrix}.$$

Then $(U\left[\begin{smallmatrix} sE_{22}-A_{22} \\ sE_{32}-A_{32} \end{smallmatrix}\right]V, U\left[\begin{smallmatrix} B_2 \\ 0 \end{smallmatrix}\right]W, C_2V)$ *is in the condensed form* (2.5).

Acknowledgments. We are grateful to three referees for their helpful comments and suggestions on earlier versions of this paper. We wish to thank Professor Michel Malabre and Dr. Jean-Francois Camalt for their step response analysis on Example 2 and observation on the unsolvability of the RRDPS for system (4.19). We would like to thank Professors Daniel Boley, Volker Mehrmann, and Paul Van Dooren for their kind encouragement and advice.

REFERENCES

[1] G. Basile and G. Marro, *Controlled and Conditioned Invariants in Linear System Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[2] M. Chang and I.B. Rhodes, *Disturbance localization in linear systems with simultaneous decoupling, pole assignment, or stabilization*, IEEE Trans. Automat. Control, 20 (1975), pp. 518–523.
[3] C.-T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, New York, 1984.
[4] D. Chu and Y.S. Hung, *Row by Row Decoupling Problem for Descriptor Systems*, Technical Report, Department of Mathematics, National University of Singapore, 2000.
[5] D. Chu and V. Mehrmann, *Disturbance decoupling for linear time-invariant systems: A matrix pencil approach*, IEEE Trans. Automat. Control, 46 (2001), pp. 802–808.
[6] D. Chu and Roger C.E. Tan, *Solvability conditions and parameterization of all solutions for the triangular decoupling problem*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 1171–1182.
[7] C. Commault, J.M. Dion, and J. Montoya, *Simultaneous decoupling and disturbance rejection: A structural approach*, Internat. J. Control, 59 (1994), pp. 1325–1344.
[8] J.W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part* I: *Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.
[9] J.W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications. Part* II: *Software and applications*, ACM Trans. Math. Software, 19 (1993), pp. 175–201.
[10] P.L. Falb and W.A. Wolovich, *Decoupling in the design and synthesis of multivariable control systems*, IEEE Trans. Automat. Control, 12 (1967), pp. 651–659.
[11] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[12] J.F. Lafay, J. Descusse, and M. Malabre, *Solution to Morgan's problem*, IEEE Trans. Automat. Control, 33 (1988), pp. 732–739.
[13] J.C. Martinez Garcia and M. Malabre, *The row by row decoupling problem with stability: A structural approach*, IEEE Trans. Automat. Control, 39 (1994), pp. 2457–2460.
[14] J.C. Martinez Garcia and M. Malabre, *The simultaneous disturbance rejection and regular row by row decoupling with stability: A geometric approach*, IEEE Trans. Automat. Control, 40 (1995), pp. 365–369,.
[15] G.S. Miminis, *Deflation in eigenvalue assignment of descriptor systems using state feedback*, IEEE Trans. Automat. Control, 38 (1993), pp. 1322–1336.
[16] 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.
[17] P.H. Petkov, N.D. Christov, and M.M. Konstantinov, *Computational Methods in Linear Control Systems*, Prentice-Hall, Hertfordshire, UK, 1991.
[18] P. Van Dooren, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, 26 (1981), pp. 111–129.
[19] A. Varga, *On stabilization methods of descriptor systems*, Systems Control Lett., 24 (1995), pp. 133–138.
[20] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed., Springer-Verlag, New York, 1985.
[21] W.M. Wonham and A.S. Morse, *Decoupling and pole assignment in linear multivariable systems: A geometric approach*, SIAM J. Control Optim., 8 (1970), pp. 1–18.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1171–1182
© 2002 Society for Industrial and Applied Mathematics

SOLVABILITY CONDITIONS AND PARAMETERIZATION OF ALL SOLUTIONS FOR THE TRIANGULAR DECOUPLING PROBLEM*

DELIN CHU† AND ROGER C. E. TAN†

Abstract. This is the sequel to [D. Chu and R. C. E. Tan, *SIAM J. Matrix Anal. Appl.*, 23 (2002), pp. 1143-1170]. In that paper we studied the row by row decoupling problem with stability in control theory and developed a numerically reliable method for solving it. In this paper we study a related problem—the triangular decoupling problem. We not only give new and explicit solvability conditions but also parameterize all the solutions. The basis of our result is a condensed form which is computed using only orthogonal transformations. Hence, our new solvability conditions can be verified and all solutions can be parameterized in a numerically stable manner.

Key words. triangular decoupling, condensed form, orthogonal transformation

AMS subject classifications. 93B05, 93B40, 93B52, 65F35

PII. S0895479801389710

1. Introduction.

Consider the linear time-invariant system

$$\begin{aligned} E\dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned} \tag{1.1}$$

where $E, A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{m\times n}$, E is nonsingular, $x \in \mathbf{R}^n$ is the state, $u \in \mathbf{R}^m$ is the control input, and $y \in \mathbf{R}^m$ is the output. If a state feedback of the form

$$u = Fx + Hv \tag{1.2}$$

is applied to system (1.1), then the closed-loop system becomes

$$E\dot{x} = (A + BF)x + BHv, \quad y = Cx. \tag{1.3}$$

The triangular decoupling problem with or without stability can be formulated as follows.

DEFINITION 1.1. *Suppose there is a system of the form* (1.1).

(i) Triangular decoupling problem (TDP): *Find matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *such that the transfer matrix* $T(s)$ *from output* y *to the input* v *in* (1.3) *is lower triangular and nonsingular, i.e.,*

$$T(s) := C(sE - A - BF)^{-1}BH \text{ is lower triangular and nonsingular.} \tag{1.4}$$

(ii) Triangular decoupling problem with stability (TDPS): *Find matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *such that* (1.4) *is true and* $E^{-1}(A + BF)$ *is stable.*

In [8] we studied the row by row decoupling problem with stability in control theory. However, in general row by row decoupling is not always possible. In such cases, one can resort to triangular decoupling because it requires less restrictive conditions.

*Received by the editors April 8, 2001; accepted for publication (in revised form) by D. Boley December 14, 2001; published electronically May 10, 2002. This research was supported by NUS Research grant R-146-000-016-112.

http://www.siam.org/journals/simax/23-4/38971.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@math.nus.edu.sg, mattance@math.nus.edu.sg).

The TDP has been treated in the existing literature by many researchers using various approaches (see [1, 2, 3, 4, 5, 6, 7]). However, these studies are all incomplete, since

(a) all solutions for the TDP and TDPS have not been parameterized explicitly;
(b) no numerically stable algorithms are available to verify the existing solvability conditions and compute a desired solution and parameterize all solutions for TDP and TDPS.

In this paper we will study the TDP and TDPS. Using a condensed form, we give new and explicit solvability conditions and parameterize all solutions for the TDP and TDPS. This condensed form is computed using only orthogonal transformations. Hence, our new solvability conditions can be easily verified and all solutions can be parameterized in a numerically stable manner using standard numerical linear algebra software such as MATLAB.

In this paper we denote the generic rank of a rational matrix function by $\operatorname{rank}_g[\cdot]$.

2. The main result. First, we need to obtain a condensed form for the system (1.1) and then from it develop our main result in this section.

LEMMA 2.1. *Let $\mathcal{E}, \mathcal{A} \in \mathbf{R}^{n\times n}$, $\mathcal{B} \in \mathbf{R}^{n\times m}$, $\mathcal{C} \in \mathbf{R}^{m\times n}$ with $\mathcal{E}$ nonsingular. There exist orthogonal matrices $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ such that*

$$\mathcal{U}(s\mathcal{E}-\mathcal{A})\mathcal{V} = \begin{matrix} \\ \nu_1 \\ \nu_2 \\ \mu_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} \mu_1 \end{matrix} & \begin{matrix} \mu_2 \end{matrix} \\ \begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} & 0 \\ s\mathcal{E}_{21}-\mathcal{A}_{21} & -\mathcal{A}_{22} \\ s\mathcal{E}_{31}-\mathcal{A}_{31} & s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix} \end{matrix}, \qquad \mathcal{U}\mathcal{B}\mathcal{W} = \begin{matrix} \\ \nu_1 \\ \nu_2 \\ \mu_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} \nu_2 \end{matrix} & \begin{matrix} m-\nu_2 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ \mathcal{B}_{21} & 0 \\ \mathcal{B}_{31} & \mathcal{B}_{32} \end{bmatrix} \end{matrix},$$

$$\mathcal{C}\mathcal{V} = \begin{matrix} \\ 1 \\ m-1 \end{matrix}\!\!\begin{matrix} \begin{matrix} \mu_1 \end{matrix} & \begin{matrix} \mu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{C}_{11} & 0 \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \end{matrix}, \tag{2.1}$$

where $\mathcal{E}_{32}$ and $\mathcal{B}_{21}$ are nonsingular, $\mu_1 = \nu_1 + \nu_2$, $\nu_2 = 0$ or $\nu_2 = 1$, and

$$\operatorname{rank}\begin{bmatrix} -\mathcal{A}_{22} & \mathcal{B}_{21} & 0 \\ s\mathcal{E}_{32}-\mathcal{A}_{32} & \mathcal{B}_{31} & \mathcal{B}_{32} \end{bmatrix} = \nu_2 + \mu_2 \qquad \forall s \in \mathbf{C}, \tag{2.2}$$

$$\operatorname{rank}_g \begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} \\ \mathcal{C}_{11} \end{bmatrix} = \mu_1. \tag{2.3}$$

Proof. The condensed form (2.1) is the transposed version of the form (2.5) in [8]. Since property (2.3) and the nonsingularity of $\mathcal{E}_{32}$ and $\mathcal{B}_{21}$ give

$$n+1 \geq \operatorname{rank}_g \begin{bmatrix} s\mathcal{E}-\mathcal{A} & \mathcal{B} \\ \begin{bmatrix} 1 & 0 \end{bmatrix}\mathcal{C} & 0 \end{bmatrix} = \mu_1 + \nu_2 + \mu_2 = n + \nu_2, \tag{2.4}$$

thus, $\nu_2 \leq 1$, i.e., $\nu_2 = 0$ or $\nu_2 = 1$. ∎

Remark 1. From (2.4), we know that

$$\nu_2 = \operatorname{rank}_g \begin{bmatrix} s\mathcal{E}-\mathcal{A} & \mathcal{B} \\ \begin{bmatrix} 1 & 0 \end{bmatrix}\mathcal{C} & 0 \end{bmatrix} - n,$$

so ν_2 is independent of orthogonal matrices $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ in the condensed form (2.1).

Obviously, in (2.1) if $\nu_2 = 0$, then the second row blocks in $\mathcal{U}(s\mathcal{E}-\mathcal{A})\mathcal{V}$ and $\mathcal{U}\mathcal{B}\mathcal{W}$ and the first column block in $\mathcal{U}\mathcal{B}\mathcal{W}$ disappear, and (2.1) is reduced to

$$\mathcal{U}(s\mathcal{E}-\mathcal{A})\mathcal{V} = \begin{matrix} \\ \nu_1 \\ \mu_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} \mu_1 \end{matrix} & \begin{matrix} \mu_2 \end{matrix} \\ \begin{bmatrix} s\mathcal{E}_{11}-\mathcal{A}_{11} & 0 \\ s\mathcal{E}_{31}-\mathcal{A}_{31} & s\mathcal{E}_{32}-\mathcal{A}_{32} \end{bmatrix} \end{matrix}, \qquad \mathcal{U}\mathcal{B}\mathcal{W} = \begin{matrix} \\ \nu_1 \\ \mu_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} m \end{matrix} \\ \begin{bmatrix} 0 \\ \mathcal{B}_{32} \end{bmatrix} \end{matrix},$$

$$\mathcal{CV} = \begin{matrix} \\ 1 \\ m-1 \end{matrix} \begin{matrix} \begin{matrix} \mu_1 & \mu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{C}_{11} & 0 \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \end{matrix},$$

where $\nu_1 = \mu_1$, $\mathcal{E}_{11}$ is nonsingular, (2.3) holds, and

$$\operatorname{rank} \begin{bmatrix} s\mathcal{E}_{32} - \mathcal{A}_{32} & \mathcal{B}_{32} \end{bmatrix} = \mu_2 \qquad \forall s \in \mathbf{C}.$$

The following simple example shows that ν_2 in (2.1) may attain the value zero for some $(\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{C})$. Let

$$\mathcal{E} = I_4, \quad \mathcal{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Obviously, $(\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ is in the condensed form (2.1) with

$$\mathcal{U} = \mathcal{V} = I_4, \quad \mathcal{W} = I_2, \quad \nu_1 = \mu_1 = \mu_2 = 2, \quad \nu_2 = 0.$$

By applying the transposed version of Lemma 2.5 in [8] to the condensed form (2.1), we immediately get the following corollary.

COROLLARY 2.2. *Let $\mathcal{E}, \mathcal{A} \in \mathbf{R}^{n\times n}, \mathcal{B} \in \mathbf{R}^{n\times m}$, and $\mathcal{C} \in \mathbf{R}^{m\times n}$ with $\mathcal{E}$ nonsingular. Assume that orthogonal matrices $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ have been determined such that $(\mathcal{U}(s\mathcal{E} - \mathcal{A})\mathcal{V}, \mathcal{U}\mathcal{B}\mathcal{W}, \mathcal{C}\mathcal{V})$ is in the condensed form* (2.1). *Let the QR factorization of $\begin{bmatrix} \mathcal{B}_{21} \\ \mathcal{B}_{31} \end{bmatrix}$ with column pivoting be*

$$\mathcal{P} \begin{bmatrix} \mathcal{B}_{21} \\ \mathcal{B}_{31} \end{bmatrix} = \begin{matrix} \nu_2 \\ \mu_2 \end{matrix} \begin{bmatrix} \tilde{\mathcal{B}}_{21} \\ 0 \end{bmatrix}, \quad \mathcal{P} = \begin{matrix} \\ \nu_2 \\ \mu_2 \end{matrix} \begin{matrix} \begin{matrix} \nu_2 & \mu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{P}_{22} & \mathcal{P}_{23} \\ \mathcal{P}_{32} & \mathcal{P}_{33} \end{bmatrix} \end{matrix},$$

where $\mathcal{P}$ is orthogonal. Denote

$$\mathcal{P} \left[\begin{array}{cc|c} s\mathcal{E}_{21} - \mathcal{A}_{21} & -\mathcal{A}_{22} & 0 \\ s\mathcal{E}_{31} - \mathcal{A}_{31} & s\mathcal{E}_{32} - \mathcal{A}_{32} & \mathcal{B}_{32} \end{array}\right] = \begin{matrix} \\ \nu_2 \\ \mu_2 \end{matrix} \begin{matrix} \begin{matrix} \mu_1 & \mu_2 & m-\nu_2 \end{matrix} \\ \begin{bmatrix} * & * & * \\ s\tilde{\mathcal{E}}_{31} - \tilde{\mathcal{A}}_{31} & s\tilde{\mathcal{E}}_{32} - \tilde{\mathcal{A}}_{32} & \tilde{\mathcal{B}}_{32} \end{bmatrix} \end{matrix},$$

$$\mathcal{Y} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \mathcal{P}_{32} & \mathcal{P}_{33} \end{bmatrix} \mathcal{U}.$$

Then $\mathcal{Y}$ is nonsingular and

$$\mathcal{Y}(s\mathcal{E} - \mathcal{A})\mathcal{V} = \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} & 0 \\ s\mathcal{E}_{21} - \mathcal{A}_{21} & -\mathcal{A}_{22} \\ s\tilde{\mathcal{E}}_{31} - \tilde{\mathcal{A}}_{31} & s\tilde{\mathcal{E}}_{32} - \tilde{\mathcal{A}}_{32} \end{bmatrix}, \quad \mathcal{Y}\mathcal{B}\mathcal{W} = \begin{bmatrix} 0 & 0 \\ \mathcal{B}_{21} & 0 \\ 0 & \tilde{\mathcal{B}}_{32} \end{bmatrix},$$

where $\tilde{\mathcal{E}}_{32}$ is nonsingular and

$$\operatorname{rank} \begin{bmatrix} s\tilde{\mathcal{E}}_{32} - \tilde{\mathcal{A}}_{32} & \tilde{\mathcal{B}}_{32} \end{bmatrix} = \mu_2 \qquad \forall s \in \mathbf{C}. \tag{2.5}$$

As a direct consequence of Lemma 2.1 and Corollary 2.2, we have the following important theorem.

THEOREM 2.3. *Given $E, A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}$, and $C \in \mathbf{R}^{m\times n}$, there exist an integer $0 \le k \le m$, a nonsingular matrix $X \in \mathbf{R}^{n\times n}$, and orthogonal matrices $V \in \mathbf{R}^{n\times n}$ and $W \in \mathbf{R}^{m\times m}$ such that*

$$X(sE-A)V = \begin{array}{c} \\ \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \\ \tilde{n}_4 \\ \vdots \\ \tilde{n}_{2i-1} \\ \tilde{n}_{2i} \\ \vdots \\ \tilde{n}_{2k-1} \\ \tilde{n}_{2k} \\ n_{k+1} \end{array} \begin{array}{c} \begin{array}{ccccccc} n_1 & n_2 & n_3 & \cdots & n_k & n_{k+1} \end{array} \\ \begin{bmatrix} sE_{11}-A_{11} & 0 & 0 & \cdots & 0 & 0 \\ sE_{21}-A_{21} & -A_{22} & -A_{23} & \cdots & -A_{2k} & -A_{2(k+1)} \\ sE_{31}-A_{31} & sE_{32}-A_{32} & 0 & \cdots & 0 & 0 \\ sE_{41}-A_{41} & sE_{42}-A_{42} & -A_{43} & \cdots & -A_{4k} & -A_{4(k+1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ sE_{(2i-1)1}-A_{(2i-1)1} & sE_{(2i-1)2}-A_{(2i-1)2} & sE_{(2i-1)3}-A_{(2i-1)3} & \cdots & 0 & 0 \\ sE_{(2i)1}-A_{(2i)1} & sE_{(2i)2}-A_{(2i)2} & sE_{(2i)3}-A_{(2i)3} & \cdots & -A_{(2i)k} & -A_{(2i)(k+1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ sE_{(2k-1)1}-A_{(2k-1)1} & sE_{(2k-1)2}-A_{(2k-1)2} & sE_{(2k-1)3}-A_{(2k-1)3} & \cdots & sE_{(2k-1)k}-A_{(2k-1)k} & 0 \\ sE_{(2k)1}-A_{(2k)1} & sE_{(2k)2}-A_{(2k)2} & sE_{(2k)3}-A_{(2k)3} & \cdots & sE_{(2k)k}-A_{(2k)k} & -A_{(2k)(k+1)} \\ sE_{(2k+1)1}-A_{(2k+1)1} & sE_{(2k+1)2}-A_{(2k+1)2} & sE_{(2k+1)3}-A_{(2k+1)3} & \cdots & sE_{(2k+1)k}-A_{(2k+1)k} & D(s) \end{bmatrix} \end{array},$$

$$XBW = \begin{array}{c} \\ \tilde{n}_1 \\ \tilde{n}_2 \\ \tilde{n}_3 \\ \tilde{n}_4 \\ \vdots \\ \tilde{n}_{2i-1} \\ \tilde{n}_{2i} \\ \vdots \\ \tilde{n}_{2k-1} \\ \tilde{n}_{2k} \\ n_{k+1} \end{array} \begin{array}{c} \begin{array}{ccccccc} \tilde{n}_2 & \tilde{n}_4 & \cdots & \tilde{n}_{2i} & \cdots & \tilde{n}_{2k} & m-\sum_{j=1}^{k}\tilde{n}_{2j} \end{array} \\ \begin{bmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ B_{21} & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & B_{42} & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & B_{(2i)i} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & B_{(2k)k} & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & B_{(2k+1)(k+1)} \end{bmatrix} \end{array}, \tag{2.6}$$

$$CV = \begin{array}{c} \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ m-k \end{array} \begin{array}{c} \begin{array}{cccccc} n_1 & n_2 & n_3 & \cdots & n_k & n_{k+1} \end{array} \\ \begin{bmatrix} C_{11} & 0 & 0 & \cdots & 0 & 0 \\ C_{21} & C_{22} & 0 & \cdots & 0 & 0 \\ C_{31} & C_{32} & C_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{k1} & C_{k2} & C_{k3} & \cdots & C_{kk} & 0 \\ C_{(k+1)1} & C_{(k+1)2} & C_{(k+1)3} & \cdots & C_{(k+1)k} & 0 \end{bmatrix} \end{array},$$

where $D(s) = sE_{(2k+1)(k+1)} - A_{(2k+1)(k+1)}$, $E_{(2k+1)(k+1)}$, $B_{(2i)i}$, and $\left[\begin{smallmatrix} E_{(2i-1)i} \\ E_{(2i)i} \end{smallmatrix}\right]$, $i = 1, \ldots, k$, are nonsingular, and

$$\tilde{n}_{2i-1} + \tilde{n}_{2i} = n_i, \quad \tilde{n}_{2i} = 0 \text{ or } \tilde{n}_{2i} = 1, \quad i = 1, \ldots, k, \tag{2.7}$$

$$\operatorname{rank}(sE_{(2i-1)i} - A_{(2i-1)i}) = \tilde{n}_{2i-1}, \quad i = 2, \ldots, k \qquad \forall s \in \mathbf{C}, \tag{2.8}$$

$$\operatorname{rank}\begin{bmatrix} D(s) & B_{(2k+1)(k+1)} \end{bmatrix} = n_{k+1} \qquad \forall s \in \mathbf{C}, \tag{2.9}$$

$$\operatorname{rank}_g \begin{bmatrix} sE_{(2i-1)i} - A_{(2i-1)i} \\ C_{ii} \end{bmatrix} = n_i, \quad i = 1, \ldots, k. \tag{2.10}$$

Moreover, although X is not orthogonal, it and the condensed form (2.6) *are computed using* only orthogonal transformations, *which can be implemented in a numerically stable way.*

Proof. See the appendix.

We are now ready to present our main result.

THEOREM 2.4. *Given a system of the form* (1.1), *assume that the integer* k, *nonsingular matrix* X, *and orthogonal matrices* V *and* W *have been determined such that* $(X(sE-A)V, XBW, CV)$ *is in the condensed form* (2.6).

(i) *The TDP is solvable if and only if*

$$k = m, \quad \tilde{n}_{2i} = 1, \quad i = 1, 2, \ldots, m. \tag{2.11}$$

All matrices F *and* H *in* (1.2) *solving the TDP are given by*

$$F = W \begin{bmatrix} F_{11} & -\frac{A_{22}}{B_{21}} & -\frac{A_{23}}{B_{21}} & \cdots & -\frac{A_{2(m-1)}}{B_{21}} & -\frac{A_{2m}}{B_{21}} \\ F_{21} & F_{22} & -\frac{A_{43}}{B_{42}} & \cdots & -\frac{A_{4(m-1)}}{B_{42}} & -\frac{A_{4m}}{B_{42}} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ F_{(m-1)1} & F_{(m-1)2} & F_{(m-1)3} & \ddots & F_{(m-1)(m-1)} & -\frac{A_{(2(m-1))m}}{B_{(2(m-1))(m-1)}} \\ F_{m1} & F_{m2} & F_{m3} & \cdots & F_{m(m-1)} & F_{mm} \end{bmatrix} V^T, \tag{2.12}$$

$$H = W \begin{bmatrix} h_{11} & 0 & \cdots & 0 \\ h_{21} & h_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mm} \end{bmatrix}, \tag{2.13}$$

where $F_{ij} \in \mathbf{R}^{1\times n_j}, h_{ij} \in \mathbf{R}, i = 1, \ldots, m$, $j = 1, \ldots, i$, *are arbitrary,* $h_{ii}, i = 1, \ldots, m$, *are nonzero.*

(ii) *The TDPS is solvable if and only if condition* (2.11) *is true and furthermore*

$$\operatorname{rank}(sE_{11} - A_{11}) = \tilde{n}_1 \qquad \forall s \in \mathbf{C}/\mathbf{C}^-. \tag{2.14}$$

Moreover, all matrices F *and* H *in* (1.2) *solving the TDPS are those in* (2.12) *and* (2.13) *with*

$$\begin{bmatrix} E_{(2i-1)i} \\ E_{(2i)i} \end{bmatrix}^{-1} \left(\begin{bmatrix} A_{(2i-1)i} \\ A_{(2i)i} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{(2i)i} \end{bmatrix} F_{ii} \right)$$

being stable, i.e.,

$$\operatorname{rank} \begin{bmatrix} sE_{(2i-1)i} - A_{(2i-1)i} \\ sE_{(2i)i} - A_{(2i)i} - B_{(2i)i}F_{ii} \end{bmatrix} = n_i, \quad i = 1, \ldots, m \qquad \forall s \in \mathbf{C}/\mathbf{C}^-. \tag{2.15}$$

Proof. We first prove the "necessity" and then the "sufficiency."

(i) *Necessity.* Let $F \in \mathbf{R}^{m\times n}$ and $H \in \mathbf{R}^{m\times m}$ be such that (1.4) holds true. Then H is nonsingular and we have

$$\begin{aligned} \operatorname{rank}_g \begin{bmatrix} X(sE-A)V & XBW \\ CV & 0 \end{bmatrix} &= \operatorname{rank}_g \begin{bmatrix} sE - A - BF & BH \\ C & 0 \end{bmatrix} \\ &= n + \operatorname{rank}_g(C(sE - A - BF)^{-1}BH) = n + m. \end{aligned} \tag{2.16}$$

However, using the condensed form (2.6), we know that

$$\operatorname{rank}_g \begin{bmatrix} X(sE-A)V & XBW \\ CV & 0 \end{bmatrix} = \sum_{i=1}^{k} n_i + \sum_{i=1}^{k} \tilde{n}_{2i} + n_{k+1} = n + \sum_{i=1}^{k} \tilde{n}_{2i}. \tag{2.17}$$

Note that $k \le m$ and $\tilde{n}_{2i} \le 1$, $i = 1, \ldots, k$. Thus, condition (2.11) follows directly from (2.16) and (2.17). Furthermore, $B_{(2k+1)(k+1)}$ vanishes, so property (2.9) and the nonsingularity of $E_{(2k+1)(k+1)}$ yield $n_{k+1} = 0$.

In the following we show that F and H are of the forms (2.12) and (2.13), respectively. Let

$$W^T F V = \begin{array}{c} \\ 1 \\ \vdots \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} n_1 & \cdots & n_m \end{array} \\ \begin{bmatrix} F_{11} & \cdots & F_{1m} \\ \vdots & \ldots & \vdots \\ F_{m1} & \cdots & F_{mm} \end{bmatrix} \end{array}, \qquad W^T H = \begin{array}{c} \\ 1 \\ \vdots \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \cdots & 1 \end{array} \\ \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ldots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{bmatrix} \end{array}.$$

Now, in the condensed form (2.6), for $i = 1, \ldots, m$, denote

$$s\mathcal{E}_i - \mathcal{A}_i = \begin{bmatrix} sE_{(2i-1)i} - A_{(2i-1)i} & 0 & \cdots & 0 \\ sE_{(2i)i} - A_{(2i)i} & -A_{(2i)(i+1)} & \cdots & -A_{(2i)m} \\ \vdots & \vdots & \cdots & \vdots \\ sE_{(2m-1)i} - A_{(2m-1)i} & sE_{(2m-1)(i+1)} - A_{(2m-1)(i+1)} & \cdots & sE_{(2m-1)m} - A_{(2m-1)m} \\ sE_{(2m)i} - A_{(2m)i} & sE_{(2m)(i+1)} - A_{(2m)(i+1)} & \cdots & sE_{(2m)m} - A_{(2m)m} \end{bmatrix},$$

$$\mathcal{B}_i = \begin{bmatrix} 0 & \cdots & 0 \\ B_{(2i)i} & \cdots & 0 \\ \vdots & \ldots & \vdots \\ 0 & \cdots & 0 \\ 0 & \cdots & B_{(2m)m} \end{bmatrix}, \qquad \mathcal{C}_i = \begin{bmatrix} C_{ii} & & \\ \vdots & \ddots & \\ C_{mi} & \cdots & C_{mm} \end{bmatrix}. \tag{2.18}$$

Then property (2.7) implies that

$$\mathcal{E}_i,\ i = 1, \ldots, m, \text{ are nonsingular.} \tag{2.19}$$

At the same time, property (2.8) and the nonsingularity of $B_{(2i)i}$ yield

$$\operatorname{rank} \begin{bmatrix} s\mathcal{E}_i - \mathcal{A}_i & \mathcal{B}_i \end{bmatrix} = \sum_{j=2i-1}^{2m} \tilde{n}_j = \sum_{j=i}^{m} n_j \qquad \forall s \in \mathbf{C}. \tag{2.20}$$

Set

$$\mathcal{F}_i = \begin{bmatrix} F_{ii} & \cdots & F_{im} \\ \vdots & \ldots & \vdots \\ F_{mi} & \cdots & F_{mm} \end{bmatrix}, \qquad \mathcal{H}_i = \begin{bmatrix} h_{ii} & \cdots & h_{im} \\ \vdots & \ldots & \vdots \\ h_{mi} & \cdots & h_{mm} \end{bmatrix}.$$

From (1.4) and using (2.11) we have $\mathcal{C}_1 = CV$,

$$\mathcal{C}_1(s\mathcal{E}_1 - \mathcal{A}_1 - \mathcal{B}_1\mathcal{F}_1)^{-1}\mathcal{B}_1\mathcal{H}_1 = C(sI - A - BF)^{-1}BH \text{ is lower triangular and nonsingular,} \tag{2.21}$$

and $\mathcal{H}_1$ is nonsingular. Hence, for $i = 1, \ldots, m-1$, we have the following:

(a) Since

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \mathcal{C}_i(s\mathcal{E}_i - \mathcal{A}_i - \mathcal{B}_i\mathcal{F}_i)^{-1}\mathcal{B}_i\mathcal{H}_i \begin{bmatrix} 0 \\ I_{\sum_{j=i}^{m} n_j - 1} \end{bmatrix} = 0,$$

by Lemma 2.3 in [8] and property (2.10) we get

$$\begin{aligned}\sum_{j=i}^{m} n_j &= \operatorname{rank}_g \left[\begin{array}{cc} s\mathcal{E}_i - \mathcal{A}_i - \mathcal{B}_i\mathcal{F}_i & \mathcal{B}_i\mathcal{H}_i \left[\begin{array}{c} 0 \\ I_{\sum_{j=i}^{m} n_j - 1} \end{array}\right] \\ \left[\begin{array}{cc} 1 & 0 \end{array}\right] \mathcal{C}_i & 0 \end{array}\right] \\ &= \operatorname{rank}_g \left[\begin{array}{ccc} sE_{(2i-1)i} - A_{(2i-1)i} & 0 & 0 \\ sE_{(2i)i} - A_{(2i)i} - B_{(2i)i}F_{ii} & -X_i & Y_i \\ Z_i & s\mathcal{E}_{i+1} - \mathcal{A}_{i+1} - \mathcal{B}_{i+1}\mathcal{F}_{i+1} & \mathcal{B}_{i+1}\mathcal{H}_{i+1} \\ C_{ii} & 0 & 0 \end{array}\right] \\ &= n_i + \operatorname{rank}_g \left[\begin{array}{cc} -X_i & Y_i \\ s\mathcal{E}_{i+1} - \mathcal{A}_{i+1} - \mathcal{B}_{i+1}\mathcal{F}_{i+1} & \mathcal{B}_{i+1}\mathcal{H}_{i+1} \end{array}\right],\end{aligned}$$

i.e.,

$$\sum_{j=i+1}^{m} n_j = \operatorname{rank}_g \left[\begin{array}{cc} -X_i & Y_i \\ s\mathcal{E}_{i+1} - \mathcal{A}_{i+1} - \mathcal{B}_{i+1}\mathcal{F}_{i+1} & \mathcal{B}_{i+1}\mathcal{H}_{i+1} \end{array}\right], \tag{2.22}$$

where

$$\begin{aligned} X_i &= \left[\begin{array}{ccc} A_{(2i)(i+1)} & \cdots & A_{(2i)k} \end{array}\right] + B_{(2i)i} \left[\begin{array}{ccc} F_{i(i+1)} & \cdots & F_{ik} \end{array}\right], \\ Z_i &= \left[\begin{array}{c} sE_{(2i+1)i} - A_{(2i+1)i} \\ \vdots \\ sE_{(2k)i} - A_{(2k)i} \end{array}\right] - \mathcal{B}_{i+1} \left[\begin{array}{c} F_{(i+1)i} \\ \vdots \\ F_{ki} \end{array}\right], \\ Y_i &= B_{(2i)i} \left[\begin{array}{ccc} h_{i(i+1)} & \cdots & h_{im} \end{array}\right]. \end{aligned}$$

However, $\mathcal{E}_{i+1}$ is nonsingular (see (2.19)), i.e., $\operatorname{rank}(\mathcal{E}_{i+1}) = \sum_{j=i+1}^{m} n_j$; thus, it follows directly from (2.22) and Lemma 2.3 in [8] that

$$Y_i = 0. \tag{2.23}$$

However, $B_{(2i)i}$ is nonsingular, so

$$h_{ij} = 0, \quad j = i+1, \ldots, m. \tag{2.24}$$

Consequently, since $\mathcal{H}_i$ is nonsingular,

$$h_{ii} \neq 0, \quad \mathcal{H}_{i+1} \text{ is nonsingular.} \tag{2.25}$$

We also have from (2.20) that

$$\begin{aligned} &\operatorname{rank} \left[\begin{array}{cc} s\mathcal{E}_{i+1} - \mathcal{A}_{i+1} - \mathcal{B}_{i+1}\mathcal{F}_{i+1} & \mathcal{B}_{i+1}\mathcal{H}_{i+1} \end{array}\right] \\ &\quad = \operatorname{rank} \left[\begin{array}{cc} s\mathcal{E}_{i+1} - \mathcal{A}_{i+1} & \mathcal{B}_{i+1} \end{array}\right] = \sum_{j=i+1}^{m} n_j \qquad \forall s \in \mathbf{C}. \end{aligned}$$

Therefore, it follows from Lemma 2.4 in [8] and the equalities (2.23) and (2.22) that $X_i = 0$; equivalently,

$$\begin{aligned} \left[\begin{array}{ccc} F_{i(i+1)} & \cdots & F_{im} \end{array}\right] &= -B_{(2i)i}^{-1} \left[\begin{array}{ccc} A_{(2i)(i+1)} & \cdots & A_{(2i)m} \end{array}\right] \\ &= -\left[\begin{array}{ccc} A_{(2i)(i+1)} & \cdots & A_{(2i)m} \end{array}\right] / B_{(2i)i}. \end{aligned} \tag{2.26}$$

(b) Furthermore, a simple calculation using (2.26) and (2.24) gives

$$\mathcal{C}_i(s\mathcal{E}_i-\mathcal{A}_i-\mathcal{B}_i\mathcal{F}_i)^{-1}\mathcal{B}_i\mathcal{H}_i = \begin{bmatrix} d_{ii} & 0 \\ \star & \mathcal{C}_{i+1}(s\mathcal{E}_{i+1}-\mathcal{A}_{i+1}-\mathcal{B}_{i+1}\mathcal{F}_{i+1})^{-1}\mathcal{B}_{i+1}\mathcal{H}_{i+1} \end{bmatrix} \tag{2.27}$$

with

$$d_{ii} = C_{ii}\begin{bmatrix} sE_{(2i-1)i}-A_{(2i-1)i} \\ sE_{(2i)i}-A_{(2i)i}-B_{(2i)i}F_{ii} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ B_{(2i)i} \end{bmatrix} h_{ii}.$$

Since $\mathcal{C}_i(s\mathcal{E}_i-\mathcal{A}_i-\mathcal{B}_i\mathcal{F}_i)^{-1}\mathcal{B}_i\mathcal{H}_i$ is lower triangular and nonsingular, hence $\mathcal{C}_{i+1}(s\mathcal{E}_{i+1}-\mathcal{A}_{i+1}-\mathcal{B}_{i+1}\mathcal{F}_{i+1})^{-1}\mathcal{B}_{i+1}\mathcal{H}_{i+1}$ is also lower triangular and nonsingular.

Finally, from (2.26) and (2.24) we know that F and H are of the form (2.12) and (2.13), respectively.

(ii) *Necessity.* Let $F \in \mathbf{R}^{m\times n}$ and $H \in \mathbf{R}^{m\times m}$ be such that (1.4) holds true and $E^{-1}(A+BF)$ is stable. Then condition (2.11) follows directly from part (i). Furthermore, $n_{k+1} = 0$ and

$$\operatorname{rank}\begin{bmatrix} sE-A & B \end{bmatrix} = \operatorname{rank}\begin{bmatrix} X(sE-A-BF) & XB \end{bmatrix} = n \qquad \forall s \in \mathbf{C}/\mathbf{C}^-,$$

which, along with the condensed form (2.6), give condition (2.14) immediately. Note that in the necessity proof of part (i), we have shown that F and H must be of the form (2.12) and (2.13), respectively. Hence,

$$X(sE-A-BF)V = \begin{bmatrix} sE_{11}-A_{11} & 0 & \cdots & 0 \\ sE_{21}-A_{21}-B_{21}F_{11} & 0 & \cdots & 0 \\ \star & sE_{32}-A_{32} & \cdots & 0 \\ \star & sE_{42}-A_{42}-B_{42}F_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \star & \star & \cdots & sE_{(2m-1)m}-A_{(2m-1)m} \\ \star & \star & \cdots & sE_{(2m)m}-A_{(2m)m}-B_{(2m)m}F_{mm} \end{bmatrix}, \tag{2.28}$$

i.e., $X(sE-A-BF)V$ is a block lower triangular matrix with diagonal blocks $\left[\begin{smallmatrix} sE_{(2i-1)i}-A_{(2i-1)i} \\ sE_{(2i)i}-A_{(2i)i}-B_{(2i)i}F_{ii} \end{smallmatrix}\right]$. Hence, (2.15) holds.

Sufficiency in (i) *and* (ii). Since condition (2.11) holds true, the F in (2.12) and H in (2.13) are well-defined. Moreover, $B_{(2k+1)(k+1)}$ vanishes, and thus property (2.9) and the nonsingularity of $E_{(2k+1)(k+1)}$ yield $n_{k+1} = 0$. For any F in (2.12) and H in (2.13), $X(sE-A-BF)V$ is of the form (2.28). Hence, a simple calculation yields that $C(sE-A-BF)^{-1}BH$ is lower triangular. Furthermore, using condition (2.11), the condensed form (2.6) and the nonsingularity of H, we have

$$\begin{aligned} &\operatorname{rank}_g(C(sE-A-BF)^{-1}BH) \\ &= \operatorname{rank}_g(C(sE-A-BF)^{-1}B) = \operatorname{rank}_g\begin{bmatrix} sE-B-BF & B \\ C & 0 \end{bmatrix} - n \\ &= \operatorname{rank}_g\begin{bmatrix} X(sE-A)V & XBW \\ CV & 0 \end{bmatrix} - n = \left(\sum_{i=1}^{m}\tilde{n}_{2i} + \sum_{i=1}^{m} n_i\right) - n = (m+n) - n = m. \end{aligned}$$

So, $C(sE-A-BF)^{-1}BH$ is nonsingular. This completes the sufficiency in (i).

In addition, if condition (2.14) holds, then this and property (2.15) imply that for $i = 1, \ldots, m$,

$$\operatorname{rank} \begin{bmatrix} sE_{(2i-1)i} - A_{(2i-1)i} & 0 \\ sE_{(2i)i} - A_{(2i)i} & B_{(2i)i} \end{bmatrix} = \tilde{n}_{2i-1} + \tilde{n}_{2i} = n_i \qquad \forall s \in \mathbf{C}/\mathbf{C}^-.$$

Hence, the matrices F in (2.12) satisfying (2.15) are well-defined. For such matrices F, from (2.28) and (2.14) directly, we have that $E^-(A + BF)$ is stable. □

In the following we present a numerical example to illustrate Theorem 2.4. In this example, $E = I$, the matrices A, B, and C were generalized using the MATLAB command `randn`, and all calculations were carried out in MATLAB version 5.3 on an HP 712/80 workstation with IEEE standard.

EXAMPLE 1. *Consider a linear system of the form* (1.1) *with* $E = I_6$ *and*

$$A = \begin{bmatrix} 2.283129391333 & -0.130522011520 & 0.630405883785 & -0.303950548401 & -0.566033905599 & -0.067245268998 \\ -0.933627536796 & 0.282685786981 & -0.346639989395 & -0.190398640676 & 0.443759807696 & 0.272460381754 \\ 0.322264058481 & 0.506376716138 & -0.116692390185 & -0.342638135001 & -0.433525158152 & -0.059952105006 \\ 0.376346092436 & -0.373763124448 & -0.032644121361 & 0.287240739246 & 0.003781864195 & 0.185700287271 \\ 0.788496794340 & -0.318850761740 & -0.214568292491 & -0.204082647281 & 0.160466695221 & -0.171958221696 \\ -0.486066607654 & -0.062860458751 & -0.326559142051 & -0.132193719163 & 0.083112560686 & -0.708225624792 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.611474009782 & -0.139351387817 & -0.108822413255 & -0.271835728453 & 0.549162999423 & -0.119043605507 \\ -0.884529647289 & 0.276792028460 & -0.002460107216 & -0.233329865913 & -0.331123636929 & 0.497195966565 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.81174572654609 & -0.47427470270648 \\ -0.18840562963458 & 0.37121415857186 \\ -0.15125144403907 & 0.12361535544134 \\ 0.46660377223427 & -0.10777472484658 \\ 0.74731545307112 & -0.42985039663091 \\ -0.05574426645597 & 0.34130956398515 \end{bmatrix}, \quad n = 6, \ m = 2.$$

By performing Algorithm 1 *in the appendix, we have that both the TDP and the TDPS are solvable. Moreover, all solutions* (F, H) *for the TDP are given by*

$$F = W \begin{bmatrix} f_{11} & f_{12} & f_{13} & -1.28062115170497 & -0.22193226560205 & 0.95260605649937 \\ f_{21} & f_{22} & f_{23} & f_{24} & f_{25} & f_{26} \end{bmatrix} V^T,$$
$$H = W \begin{bmatrix} h_{11} & 0 \\ h_{21} & h_{22} \end{bmatrix} \tag{2.29}$$

with f_{1i}, f_{2j}, $i = 1, 2, 3$, $j = 1, \ldots, 6$, *and* h_{11}, h_{21}, h_{22} *being arbitrary, and* $h_{11} \neq 0$, $h_{22} \neq 0$. *Furthermore, all solutions* (F, H) *for the TDPS are the solutions of the TDP with*

$$\begin{bmatrix} 0.9501292851472 & 0.4564676651683 & -0.0000000000000 \\ 0.2311385135743 & 0.0185036432482 & 0.7382072458107 \\ 0.6068425835418 + 0.304617366869 f_{11} & 0.8214071642953 + 0.304617366869 f_{12} & 0.176266144946 + 0.304617366869 f_{13} \end{bmatrix}$$

and

$$\begin{bmatrix} -0.10273688201010 & 0.37198279155256 & 0.53127157025587 \\ -0.13708943833174 & 0.91051717539050 & -0.15724825069291 \\ 0.53190307087149 & 0.15005295135054 & -0.68206629178869 \end{bmatrix}^{-1} \times$$
$$\begin{bmatrix} 0.0110878551004 & -0.0235197788574 & 0.0944294923097 \\ 0.4403669435638 & -0.2327460410701 & -0.5125399306208 \\ 0.232437276865 - 0.479839406517 f_{24} & 0.087827156236 - 0.479839406517 f_{25} & -0.096974113324 - 0.479839406517 f_{26} \end{bmatrix}$$

being stable. Here, V and W are

$$V = \begin{bmatrix} -0.509190448771 & -0.4649700530129 & -0.3649627226701 & 0.5580681216908 & 0.2333258084520 & 0.1595280113520 \\ -0.369014659501 & -0.2353526949515 & 0.1683609035794 & -0.3725764697591 & -0.5301780983677 & 0.600158191139 \\ -0.348229242975 & 0.3747797475288 & -0.5510010824179 & -0.0326781540433 & -0.5253749672967 & -0.396973077601 \\ 0.693529633294 & -0.2277272385532 & -0.4578500695054 & 0.2169642378212 & -0.3652180418808 & 0.277619465833 \\ -0.044743431581 & 0.6833703203034 & 0.1688057406093 & 0.5246333477518 & -0.0506989338864 & 0.4740222791450 \\ -0.017445210383 & 0.2628638616699 & -0.5447316405100 & -0.4757785765628 & 0.5024905917994 & 0.393705024005 \end{bmatrix},$$

$$W = \begin{bmatrix} -0.99060382389925 & -0.13676280223870 \\ -0.13676280223870 & 0.99060382389925 \end{bmatrix}.$$

3. Conclusions. We have studied the triangular decoupling problem with or without stability based on matrix pencil theory. We have not only given new verifiable solvability conditions but also parameterized all the solutions. The basis of our result is a condensed form which is computed using only orthogonal transformations. Hence, our solvability conditions can be verified and all solutions can be characterized in a numerically stable manner.

Appendix. The proof of Theorem 2.3. We prove Theorem 2.3 constructively by the following algorithm.

ALGORITHM 1.

Input: *E, $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, and $C \in \mathbf{R}^{m\times n}$ with E nonsingular.*

Output: *Nonsingular matrix $X \in \mathbf{R}^{n\times n}$, orthogonal matrices $V \in \mathbf{R}^{n\times n}$ and $W \in \mathbf{R}^{m\times m}$, and the condensed form* (2.6).

Step 0. *Set*

$$E_1 := E,\ A_1 := A,\ B_1 := B,\ C_1 := C,\ X := I_n,\ V := I_n,\ W := I_m,\ k := 1.$$

Step 1. *Compute the transposed version of the condensed form* (2.5) *in* [8] *for (E_k, A_k, B_k, C_k) to get orthogonal matrices U_k, V_k, W_k such that*

$$U_k(sE_k - A_k)V_k =: \begin{matrix} \\ \tilde{n}_{2k-1} \\ \tilde{n}_{2k} \\ n - \sum_{i=1}^{k} n_i \end{matrix} \begin{matrix} \begin{matrix} n_k & \qquad\qquad & n - \sum_{i=1}^{k} n_i \end{matrix} \\ \begin{bmatrix} sE_{(2k-1)k} - A_{(2k-1)k} & 0 \\ sE_{(2k)k} - A_{(2k)k} & -A^{(k)}_{(2k)(k+1)} \\ s\mathcal{E}^{(k)}_{(2k+1)k} - \mathcal{A}^{(k)}_{(2k+1)k} & s\mathcal{E}^{(k)}_{(2k+1)(k+1)} - \mathcal{A}^{(k)}_{(2k+1)(k+1)} \end{bmatrix} \end{matrix},$$

$$U_k B_k W_k =: \begin{matrix} \\ \tilde{n}_{2k-1} \\ \tilde{n}_{2k} \\ n - \sum_{i=1}^{k} n_i \end{matrix} \begin{matrix} \begin{matrix} \tilde{n}_{2k} & \qquad\qquad & m - \sum_{i=1}^{k} \tilde{n}_{2i} \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ B_{(2k)k} & 0 \\ \mathcal{B}^{(k)}_{(2k+1)k} & \mathcal{B}^{(k)}_{(2k+1)(k+1)} \end{bmatrix} \end{matrix},$$

$$C_k V_k =: \begin{matrix} \\ 1 \\ m-k \end{matrix} \begin{matrix} \begin{matrix} n_k & \qquad\qquad & n - \sum_{i=1}^{k} n_i \end{matrix} \\ \begin{bmatrix} C_{kk} & 0 \\ C^{(k)}_{(k+1)k} & C_{k+1} \end{bmatrix} \end{matrix},$$

where $\mathcal{E}^{(k)}_{(2k+1)(k+1)}$ and $B_{(2k)k}$ are nonsingular, $n_k = \tilde{n}_{2k-1} + \tilde{n}_{2k}$, $\tilde{n}_{2k} = 0$ or 1, and

$$\operatorname{rank} \begin{bmatrix} -A^{(k)}_{(2k)(k+1)} & B_{(2k)k} & 0s\mathcal{E}^{(k)}_{(2k+1)(k+1)} - \mathcal{A}^{(k)}_{(2k+1)(k+1)} \\ \mathcal{B}^{(k)}_{(2k+1)k} & \mathcal{B}^{(k)}_{(2k+1)(k+1)} & \end{bmatrix}$$

$$= \tilde{n}_{2k} + \left(n - \sum_{i=1}^{k} n_i \right) \qquad \forall s \in \mathbf{C}, \tag{A.1}$$

$$\text{(A.2)}\qquad \operatorname{rank}_g \begin{bmatrix} sE_{(2k-1)k} - A_{(2k-1)k} \\ C_{kk} \end{bmatrix} = n_k.$$

Denote

$$-A^{(k-1)}_{(2i)k} V_k =: \begin{matrix} \\ \tilde{n}_{2i} \end{matrix} \begin{matrix} \begin{matrix} n_k & n - \sum_{j=1}^{k} n_j \end{matrix} \\ \begin{bmatrix} -A_{(2i)k} & -A^{(k)}_{(2i)(k+1)} \end{bmatrix} \end{matrix}, \qquad i = 1, \ldots, k-1,$$

$$U_k(sE^{(k-1)}_{(2k-1)j} - A^{(k-1)}_{(2k-1)j}) =: \begin{matrix} \\ \tilde{n}_{2k-1} \\ \tilde{n}_{2k} \\ n - \sum_{i=1}^{k} n_i \end{matrix} \begin{matrix} n_j \\ \begin{bmatrix} sE_{(2k-1)j} - A_{(2k-1)j} \\ sE_{(2k)j} - A_{(2k)j} \\ s\mathcal{E}^{(k)}_{(2k+1)j} - \mathcal{A}^{(k)}_{(2k+1)j} \end{bmatrix} \end{matrix}, \qquad j = 1, \ldots, k-1,$$

$$C^{(k-1)}_{kj} =: \begin{matrix} \\ 1 \\ m-k \end{matrix} \begin{matrix} n_j \\ \begin{bmatrix} C_{kj} \\ C^{(k)}_{(k+1)j} \end{bmatrix} \end{matrix}, \qquad j = 1, \ldots, k-1,$$

and

$$X := \begin{bmatrix} I_{\sum_{i=1}^{k-1} n_i} & 0 \\ 0 & U_k \end{bmatrix} X,\ V := V \begin{bmatrix} I_{\sum_{i=1}^{k-1} n_i} & 0 \\ 0 & V_k \end{bmatrix},\ W := W \begin{bmatrix} I_{\sum_{i=1}^{k-1} \tilde{n}_{2i}} & 0 \\ 0 & W_k \end{bmatrix}.$$

Step 2. *Perform the QR factorization of* $\left[\begin{smallmatrix} B_{(2k)k} \\ \mathcal{B}^{(k)}_{(2k+1)k} \end{smallmatrix}\right]$ *with column pivoting to get an orthogonal matrix* $\mathcal{P}_k$ *such that*

$$\mathcal{P}_k \begin{bmatrix} B_{(2k)k} \\ \mathcal{B}^{(k)}_{(2k+1)k} \end{bmatrix} =: \begin{matrix} \tilde{n}_{2k} \\ n - \sum_{i=1}^{k} n_i \end{matrix} \begin{bmatrix} \tilde{B}_{(2k)k} \\ 0 \end{bmatrix}, \quad \mathcal{P}_k = \begin{matrix} \\ \tilde{n}_{2k} \\ n - \sum_{i=1}^{k} n_i \end{matrix} \begin{matrix} \begin{matrix} \tilde{n}_{2k} & n - \sum_{i=1}^{k} n_i \end{matrix} \\ \begin{bmatrix} \mathcal{P}^{(k)}_{11} & \mathcal{P}^{(k)}_{12} \\ \mathcal{P}^{(k)}_{21} & \mathcal{P}^{(k)}_{22} \end{bmatrix} \end{matrix}.$$

Denote

$$\begin{bmatrix} \mathcal{P}^{(k)}_{21} & \mathcal{P}^{(k)}_{22} \end{bmatrix} \begin{bmatrix} sE_{(2k)j} - A_{(2k)j} \\ s\mathcal{E}^{(k)}_{(2k+1)j} - \mathcal{A}^{(k)}_{(2k+1)j} \end{bmatrix} =: sE^{(k)}_{(2k+1)j} - A^{(k)}_{(2k+1)j}, \qquad j = 1, \ldots, k,$$

$$\begin{bmatrix} \mathcal{P}^{(k)}_{21} & \mathcal{P}^{(k)}_{22} \end{bmatrix} \begin{bmatrix} -A_{(2k)(k+1)} \\ s\mathcal{E}^{(k)}_{(2k+1)(k+1)} - \mathcal{A}^{(k)}_{(2k+1)(k+1)} \end{bmatrix} =: sE_{k+1} - A_{k+1},$$

$$\begin{bmatrix} \mathcal{P}^{(k)}_{21} & \mathcal{P}^{(k)}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \mathcal{B}^{(k)}_{(2k+1)(k+1)} \end{bmatrix} =: B_{k+1}, \quad \begin{bmatrix} I_{\sum_{i=1}^{k-1} n_i} & 0 & 0 \\ 0 & I_{n_k} & 0 \\ 0 & \mathcal{P}^{(k)}_{21} & \mathcal{P}^{(k)}_{22} \end{bmatrix} X =: X.$$

Since $\mathcal{E}^{(k)}_{(2k+1)(k+1)}$ *and* $B_{(2k)k}$ *are nonsingular, by Corollary* 2.2 *we know that* E_{k+1} *and* X *are nonsingular, and*

$$\text{(A.3)}\qquad \operatorname{rank} \begin{bmatrix} sE_{k+1} - A_{k+1} & B_{k+1} \end{bmatrix} = n - \sum_{i=1}^{k} n_i \qquad \forall s \in \mathbf{C}.$$

Note that if $k < m$, *then* $m - \sum_{i=1}^{k} \tilde{n}_{2i} > 0$, *so if* $k < m$ *and* $C_{k+1} \neq 0$, *set* $k := k+1$

and return to Step 1. *Otherwise, set*

$$\begin{aligned} sE_{(2k+1)(k+1)} - A_{(2k+1)(k+1)} &:= sE_{k+1} - A_{k+1}, \\ B_{(2k+1)(k+1)} &:= B_{k+1}, \\ n_{k+1} &:= n - \sum_{i=1}^{k} n_i, \\ A_{(2i)(k+1)} &:= A^{(k)}_{(2i)(k+1)}, \\ sE_{(2k+1)i} - A_{(2k+1)i} &:= sE^{(k)}_{(2k+1)i} - A^{(k)}_{(2k+1)i}, \\ C_{(k+1)i} &:= C^{(k)}_{(k+1)i}, \qquad i = 1, \ldots, k, \end{aligned}$$

and stop.

Now, properties (2.10) and (A.2) are the same, properties (2.9) and (A.3) are the same, and property (2.8) follows directly from (A.1) and the nonsingularity of $B_{(2i)i}$, $i = 1, \ldots, k$. Hence, a simple calculation using Corollary 2.2 yields that $(X(sE - A)V, XBW, CV)$ is in the condensed form (2.6). □

Acknowledgments. We are grateful to the two anonymous referees for their helpful comments and suggestions on an earlier version of this paper. The work reported here has also benefitted much from the valuable suggestions given by Professors Daniel Boley, Volker Mehrmann, and Paul Van Dooren.

REFERENCES

[1] N. Ito and H. Inaba *Block triangular decoupling for linear systems over principal ideal domains*, SIAM J. Control Optim., 35 (1997), pp. 744–765.

[2] H. Inaba and N. Otsuka, *Triangular decoupling and stabilization for linear control systems in Hilbert spaces*, IMA Math. Control Inform., 6 (1989), pp. 317–332.

[3] J. Descusse and R. Lizarzaburu, *Triangular decoupling and pole placement in linear control multivariable systems: A direct algebraic approach*, Internat. J. Control, 30 (1979), pp. 139–152.

[4] S.H. Wang, *Relationship between triangular decoupling and invertibility of linear control multivariable systems*, Internat. J. Control, 15 (1972), pp. 395–399.

[5] A.S. Morse and W.M. Wonham, *Triangular decoupling of linear control multivariable systems*, IEEE Trans. Automat. Control, 15 (1970), pp. 447–449.

[6] F.N. Koumboulis, *Input-output triangular decoupling and data sensitivity*, Automatica, 32 (1996), pp. 569–573.

[7] J.S. Liu and K. Yuan, *Characterization of allowable perturbation for robust decoupling of affine nonlinear systems*, Internat. J. Control, 54 (1991), pp. 729–735.

[8] D. Chu and R.C.E. Tan, *Numerically reliable computing for the row by row decoupling problem with stability*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 1143-1170.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1183–1193
© 2002 Society for Industrial and Applied Mathematics

PERTURBATION BOUNDS OF UNITARY AND SUBUNITARY POLAR FACTORS*

WEN LI† AND WEIWEI SUN‡

Abstract. In this paper, we present some new perturbation bounds for (generalized) polar decompositions under the Frobenius norm for both complex and real matrices. For subunitary polar factors, we show that our bounds always improve the existing bounds. Based on some interesting properties of the matrix equation $W + W^* = W^*W$, some new bounds involving both the Frobenius norm and the spectral norm of the perturbation are given. The optimality of bounds is discussed.

Key words. generalized polar decomposition, perturbation bound, singular value

AMS subject classifications. 65F10, 15A45

PII. S0895479801394623

1. Introduction.

Let $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) be the set of $m\times n$ complex (real) matrices and let $\mathbb{C}_r^{m\times n} \subset \mathbb{C}^{m\times n}$ ($\mathbb{R}_r^{m\times n} \subset \mathbb{R}^{m\times n}$) be the set of $m \times n$ complex (real) matrices having rank r. Here we always assume that $m \geq n$. Q is called a *unitary matrix* if $Q^*Q = I$ and a *subunitary matrix* if $\|Qx\|_F = ||x||_F$ for any $x \in R(Q^*)$ (e.g., see [13]), where the superscript * denotes conjugate transpose, $\|\cdot\|_F$ denotes the Frobenius norm, and $R(A)$ denotes a subspace spanned by columns of A. For a complex $m \times n$ matrix $A \in \mathbb{C}_r^{m\times n}$, there are a symmetric positive semidefinite matrix H and a subunitary matrix Q such that

$$A = QH. \tag{1.1}$$

This decomposition is called the *generalized polar decomposition* of A, and Q is called the *subunitary polar factor* of this decomposition. Usually, when

$$r = \operatorname{rank}(A) = n \tag{1.2}$$

(1.1) is called the *polar decomposition* and Q is the *unitary polar factor*. The decomposition (1.1) can be calculated from the *singular value decomposition* (SVD)

$$A = U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^*$$

by

$$H = V_1\Sigma_1V_1^*, \qquad Q = U_1V_1^*, \tag{1.3}$$

where $U = (U_1, U_2) \in \mathbb{C}^{m\times m}$ and $V = (V_1, V_2) \in \mathbb{C}^{n\times n}$ are unitary, $U_1 \in \mathbb{C}^{m\times r}$, $V_1 \in \mathbb{C}^{n\times r}$, $\Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$, and $\sigma_i, i = 1, 2, \ldots, r$, define the singular values of

*Received by the editors September 4, 2001; accepted for publication (in revised form) by R. Bhatia December 21, 2001; published electronically May 10, 2002. This work was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (project CityU 1141/01P).

http://www.siam.org/journals/simax/23-4/39462.html

†Research Center for Science, Xi'an Jiaotong University, Xi'an, 710049, People's Republic of China, and Department of Mathematics, South China Normal University, Guangzhou, 510631, People's Republic of China (liwen@scnu.edu.cn).

‡Department of Mathematics, City University of Hong Kong, Hong Kong, People's Republic of China (maweiw@math.cityu.edu.hk).

A with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. For any matrix $A \in \mathbb{C}_n^{m\times n}$, there exists a unique polar decomposition. But it is not true in general for $A \in \mathbb{C}_r^{m\times n}$ with $r < n$. It has been proved that the generalized polar decomposition (1.1) satisfying

$$R(Q^*) = R(H) \tag{1.4}$$

is unique (e.g., see [13]).

A variety of applications of the polar decomposition and generalized polar decomposition can be found in [6, 8], and some numerical methods for the polar decomposition were given in [6, 13]. In this paper we are concerned with perturbation bounds of the unitary polar factor Q. The details for perturbation bounds of eigenvalues and singular values can be found in [2, 12]. The perturbation problem arising in the polar decomposition has been studied by many authors. It should be noted that perturbation bounds of (sub)unitary polar factors depend heavily upon the number field, the rank, and the dimension of A.

Let $A \in \mathbb{C}_r^{m\times n}$ and $\widetilde{A} \in \mathbb{C}_r^{m\times n}$ with

$$A = QH, \qquad \widetilde{A} = \widetilde{Q}\widetilde{H},$$

and $E = A - \widetilde{A}$. A simple perturbation bound of subunitary polar factors in the generalized polar decomposition was given by Sun and Chen [13]:

$$\|Q - \widetilde{Q}\|_F \leq \frac{2}{\max\{\tilde{\sigma}_r, \sigma_r\}}\|E\|_F, \tag{1.5}$$

where $\tilde{\sigma}_i$, $i = 1, 2, \ldots, r$, denote the singular values of $\widetilde{A}$ with $0 < \tilde{\sigma}_r \leq \tilde{\sigma}_{r-1} \leq \cdots \leq \tilde{\sigma}_1$. A different bound obtained by Li [9] is

$$\|Q - \widetilde{Q}\|_F \leq \frac{1}{\min\{\tilde{\sigma}_r, \sigma_r\}}\|E\|_F\,. \tag{1.6}$$

The above inequality improves the bound (1.5) by a factor of 2 when the perturbation E is small enough. However, for some large scale problems, $\min\{\tilde{\sigma}_r, \sigma_r\}$ may be smaller.

Perturbation bounds for unitary polar factors, i.e., $r = n$, have been studied by Barrlund [1], Bhatia [3], Bhatia and Mukherjea [4], Chatelin and Gratton [5], Mathias [11], and Li [10]. In the special case $m = n = r$ and $\|E\|_2 < \sigma_n$, Mathias [11] proved that for complex matrices and a general unitarily invariant norm $\|\cdot\|$,

$$\|\widetilde{Q} - Q\| \leq -\frac{\|E\|}{\|E\|_2} \times \log\left(1 - \frac{\|E\|_2}{\sigma_n}\right), \tag{1.7}$$

where $||\cdot||_2$ defines the spectral norm. A further bound given by Li [10] for complex matrices is

$$\|Q - \widetilde{Q}\| \leq \frac{2}{\tilde{\sigma}_n + \sigma_n}\|E\|\,. \tag{1.8}$$

The above bound is always sharper than Mathias's bound in (1.7). However, it should be noted that the bound in (1.8) was obtained only in the special case $r = m = n$. When $r = n < m$, the best bound obtained in [10] is

$$\|Q - \widetilde{Q}\|_F \leq \sqrt{\left(\frac{2}{\sigma_n + \tilde{\sigma}_n}\right)^2 + \left(\frac{1}{\max\{\sigma_n, \tilde{\sigma}_n\}}\right)^2}\,\|E\|_F\,. \tag{1.9}$$

The organization of this paper is as follows. In section 2, we prove that for complex matrices, the bound in (1.8) holds for generalized polar decompositions, i.e., for any r and $m \geq n$. The bound obtained here is sharper than both (1.5) and (1.6) for generalized polar decompositions and (1.9) for full rank matrices. In section 3, we present some interesting features of the solutions of the matrix equation $W + W^* = W^*W = WW^*$ and the perturbation of unitary polar factors. It is easy to see from these features why perturbation bounds in real and complex cases have different forms. Some new perturbation bounds for $n \times n$ real and complex nonsingular matrices are obtained. The optimality of bounds in both complex and real fields is discussed and more counterexamples are given in section 4.

2. Subunitary polar factors. Let I_p be the $p \times p$ identity matrix and

$$I_{m,n}^{(p)} \equiv \begin{pmatrix} I_p & 0 \\ 0 & 0 \end{pmatrix}.$$

For simplicity we write $I^{(p)}$ for $I_{m,n}^{(p)}$. Let

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \in \mathbb{C}^{m\times m} \quad \text{and} \quad T = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \in \mathbb{C}^{n\times n}$$

be two unitary matrices, where both S_{11} and T_{11} are $r \times r$. Let

$$M = 2I - S_{11}^*T_{11} - T_{11}^*S_{11}, \qquad \widetilde{M} = 2I - T_{11}S_{11}^* - S_{11}T_{11}^*$$

and m_{ij} and $\widetilde{m}_{ij}$ denote the (i, j) entry of M and $\widetilde{M}$, respectively. Some basic properties for M and $\widetilde{M}$ are given below.

LEMMA 2.1. *Both M and $\widetilde{M}$ are Hermitian positive semidefinite and* $\mathrm{tr}(M) = \mathrm{tr}(\widetilde{M})$.

Proof. Since S and T are unitary, $\|S_{11}\|_2 \leq 1$ and $\|T_{11}\|_2 \leq 1$. Then the magnitude of each eigenvalue of $S_{11}^*T_{11} + T_{11}^*S_{11}$ is less than or equal to 2. M is Hermitian positive semidefinite and so is $\widetilde{M}$. The second part of this lemma follows from the definition and by using the relation $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. ∎

The following lemma can be easily obtained.

LEMMA 2.2. *Let A and B be $n \times n$ Hermitian matrices and $A - B$ be positive semidefinite. Then* $\mathrm{tr}(A) \geq \mathrm{tr}(B)$.

LEMMA 2.3. *Let $S \in \mathbb{C}^{m\times m}$ and $T \in \mathbb{C}^{n\times n}$ be two unitary matrices, and*

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{C}_r^{m\times n}, \qquad \widetilde{\Sigma} = \begin{pmatrix} \widetilde{\Sigma}_1 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{C}_r^{m\times n},$$

where $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ and $\widetilde{\Sigma}_1 = \mathrm{diag}(\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_r)$ are two $r \times r$ diagonal matrices with $0 < \sigma_r \leq \cdots \leq \sigma_1$ and $0 < \widetilde{\sigma}_r \leq \cdots \leq \widetilde{\sigma}_1$. Let $\Gamma = \Sigma - \sigma I^{(r)}$ and $\widetilde{\Gamma} = \widetilde{\Sigma} - \sigma I^{(r)}$, $\sigma > 0$. Then

$$\begin{aligned} (2.1) \qquad & 2\mathcal{R}e\,\mathrm{tr}[(SI^{(r)} - I^{(r)}T)(S\Gamma - \widetilde{\Gamma}T)^*] \\ & \geq (\sigma_{r-1} + \widetilde{\sigma}_{r-1} - 2\sigma)\mathrm{tr}(M) - (\sigma_{r-1} - \sigma_r)m_{rr} - (\widetilde{\sigma}_{r-1} - \widetilde{\sigma}_r)\widetilde{m}_{rr}\,, \end{aligned}$$

where $\mathcal{R}e$ denotes the real part of a complex number.

Proof. We only prove the inequality (2.1) in the case $m = n$. When $m > n$, the result can be proved similarly by considering the square matrices $\mathcal{S} = S$, $\mathcal{T} = \begin{pmatrix} T & 0 \\ 0 & I_{m-n} \end{pmatrix}$, $\Omega = [\Sigma, 0]$, and $\widetilde{\Omega} = [\widetilde{\Sigma}, 0]$.

From the proof of Lemma 1 of [9] we obtain

$$\begin{aligned}(2.2)\qquad &2\mathcal{R}e\,\mathrm{tr}[(SI^{(r)}-I^{(r)}T)(S\Gamma-\widetilde{\Gamma}T)^*]\\ &=\mathrm{tr}[(2I^{(r)}-S^*I^{(r)}T-T^*I^{(r)}S)\Gamma]+\mathrm{tr}[(2I^{(r)}-TI^{(r)}S^*-SI^{(r)}T^*)\widetilde{\Gamma}].\end{aligned}$$

It follows that

$$\begin{aligned}(2.3)\qquad &2\mathcal{R}e\,\mathrm{tr}[(SI^{(r)}-I^{(r)}T)(S\Gamma-\widetilde{\Gamma}T)^*]\\ &=\mathrm{tr}[(2I-S_{11}^*T_{11}-T_{11}^*S_{11})\Gamma_1]+\mathrm{tr}[(2I-T_{11}S_{11}^*-S_{11}T_{11}^*)\widetilde{\Gamma}_1]\\ &=\mathrm{tr}(M\Gamma_1)+\mathrm{tr}(\widetilde{M}\widetilde{\Gamma}_1),\end{aligned}$$

where $\Gamma_1=\mathrm{diag}(\sigma_1-\sigma,\ldots,\sigma_r-\sigma)$ and $\widetilde{\Gamma}_1=\mathrm{diag}(\widetilde{\sigma}_1-\sigma,\ldots,\widetilde{\sigma}_r-\sigma)$. By Lemma 2.1 both M and $\widetilde{M}$ are Hermitian positive semidefinite. Hence M and $\widetilde{M}$ have the unique Hermitian positive semidefinite square roots $M^{\frac{1}{2}}$ and $\widetilde{M}^{\frac{1}{2}}$. Clearly,

$$(2.4)\qquad \mathrm{tr}(M\Gamma_1)=\mathrm{tr}(M^{\frac{1}{2}}\Gamma_1M^{\frac{1}{2}})$$

and $M^{\frac{1}{2}}\Gamma_1M^{\frac{1}{2}}=M^{\frac{1}{2}}(\sigma_r-\sigma)M^{\frac{1}{2}}+M^{\frac{1}{2}}\mathrm{diag}(\sigma_1-\sigma_r,\ldots,\sigma_{r-1}-\sigma_r,0)M^{\frac{1}{2}}$. It follows from Lemma 2.2 and the fact $\mathrm{tr}(BC)=\mathrm{tr}(CB)$ that

$$\begin{aligned}(2.5)\qquad &\mathrm{tr}(M^{\frac{1}{2}}\Gamma_1M^{\frac{1}{2}})\\ &=\mathrm{tr}(M^{\frac{1}{2}}(\sigma_r-\sigma)M^{\frac{1}{2}})+\mathrm{tr}(M^{\frac{1}{2}}\mathrm{diag}(\sigma_1-\sigma_r,\ldots,\sigma_{r-1}-\sigma_r,0)M^{\frac{1}{2}})\\ &\geq(\sigma_r-\sigma)\mathrm{tr}(M)+(\sigma_{r-1}-\sigma_r)\mathrm{tr}(MI^{(r-1)}).\end{aligned}$$

It is easy to see that $\mathrm{tr}(MI^{(r-1)})=\mathrm{tr}(M_{r-1})$, where M_{r-1} is the $(r-1)\times(r-1)$ leading principal submatrix of M. Hence $\mathrm{tr}(M)=\mathrm{tr}(M_{r-1})+m_{rr}$. From (2.5),

$$\begin{aligned}(2.6)\qquad \mathrm{tr}(M^{\frac{1}{2}}\Gamma_1M^{\frac{1}{2}})&\geq(\sigma_r-\sigma)\mathrm{tr}(M)+(\sigma_{r-1}-\sigma_r)(\mathrm{tr}(M)-m_{rr})\\ &=(\sigma_{r-1}-\sigma)\mathrm{tr}(M)-(\sigma_{r-1}-\sigma_r)m_{rr}\,.\end{aligned}$$

Similarly we have

$$(2.7)\qquad \mathrm{tr}(\widetilde{M}^{\frac{1}{2}}\widetilde{\Gamma}_1\widetilde{M}^{\frac{1}{2}})\geq(\widetilde{\sigma}_{r-1}-\sigma)\mathrm{tr}(\widetilde{M})-(\widetilde{\sigma}_{r-1}-\widetilde{\sigma}_r)\widetilde{m}_{rr}\,.$$

By Lemma 2.1 and (2.3)–(2.7) one may deduce that

$$\begin{aligned}&2\mathcal{R}e\,\mathrm{tr}[(SI^{(r)}-I^{(r)}T)(S\Gamma-\widetilde{\Gamma}T)^*]\\ &\geq(\sigma_{r-1}-\sigma)\mathrm{tr}(M)-(\sigma_{r-1}-\sigma_r)m_{rr}+(\widetilde{\sigma}_{r-1}-\sigma)\mathrm{tr}(\widetilde{M})-(\widetilde{\sigma}_{r-1}-\widetilde{\sigma}_r)\widetilde{m}_{rr}\\ &=(\sigma_{r-1}+\widetilde{\sigma}_{r-1}-2\sigma)\mathrm{tr}(M)-(\sigma_{r-1}-\sigma_r)m_{rr}-(\widetilde{\sigma}_{r-1}-\widetilde{\sigma}_r)\widetilde{m}_{rr},\end{aligned}$$

which proves the lemma. □

A new perturbation bound for subunitary polar factors is given in the following theorem.

THEOREM 2.4. *Let* $A,\ \widetilde{A}\in\mathbb{C}_r^{m\times n}$ *and*

$$A=QH\quad\textit{and}\quad\widetilde{A}=\widetilde{Q}\widetilde{H}$$

be the generalized polar decomposition of A *and* $\widetilde{A}$*, respectively, and satisfy* (1.4)*. Then*

$$(2.8)\qquad \|Q-\widetilde{Q}\|_F\leq\frac{2}{\widetilde{\sigma}_r+\sigma_r}\|E\|_F\,.$$

Proof. Let

$$A = U\Sigma V^* \quad \text{and} \quad \widetilde{A} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^* \tag{2.9}$$

be SVDs of A and $\widetilde{A}$, respectively. Let $S = \widetilde{U}^*U$, $T = \widetilde{V}^*V$, and $\sigma = \frac{\sigma_r+\tilde{\sigma}_r}{2}$. It is easy to see that

$$\begin{aligned}\|S\Sigma - \widetilde{\Sigma}T\|_F^2 &= \sigma^2\|SI^{(r)} - I^{(r)}T\|_F^2 + \|S\Gamma - \widetilde{\Gamma}T\|_F^2 \\ &\quad + 2\sigma\mathcal{R}e\,\mathrm{tr}[(SI^{(r)} - I^{(r)}T)(S\Gamma - \widetilde{\Gamma}T)^*].\end{aligned} \tag{2.10}$$

By Lemma 2.3, we have $2\mathcal{R}e\,\mathrm{tr}[(SI^{(r)} - I^{(r)}T)(S\Gamma - \widetilde{\Gamma}T)^*] \geq 0$, which together with (2.10) gives

$$\|S\Sigma - \Sigma T\|_F \geq \frac{\sigma_r + \tilde{\sigma}_r}{2}\|SI^{(r)} - I^{(r)}T\|_F\,.$$

Hence we have

$$\begin{aligned}\|A - \widetilde{A}\|_F &= \|U\Sigma V^* - \widetilde{U}\widetilde{\Sigma}\widetilde{V}^*\|_F \\ &= \|\widetilde{U}^*U\Sigma - \widetilde{\Sigma}\widetilde{V}^*V\|_F \\ &\geq \tfrac{\sigma_r+\tilde{\sigma}_r}{2}\|UI^{(r)}V^* - \widetilde{U}I^{(r)}\widetilde{V}^*\|_F \\ &= \tfrac{\sigma_r+\tilde{\sigma}_r}{2}\|Q - \widetilde{Q}\|_F,\end{aligned}$$

which proves the theorem. ∎

Remark 1. It is noted that our result in Theorem 2.4 always improves the bounds in (1.5), (1.6), and (1.9). The following example shows that the inequality

$$\|Q - \widetilde{Q}\|_F \leq \frac{1}{\max\{\tilde{\sigma}_r\,,\ \sigma_r\}}\|E\|_F$$

does not hold even for $r = n = m$. More examples will be given in section 4 to show that the bound in (2.8) is optimal in some sense.

Example 1. Let

$$A = I \quad \text{and} \quad \widetilde{A} = -2I$$

be $n \times n$ matrices. Then $2\sqrt{n} = \|Q - \widetilde{Q}\|_F > \frac{1}{\max\{\tilde{\sigma}_n,\ \sigma_n\}}\|E\|_F = \frac{3}{2}\sqrt{n}$.

3. Normal matrix and perturbation bounds of unitary polar factors. In this section we consider only $n \times n$ matrices. First we present some interesting features for some special normal matrices, which will be used to provide new perturbation bounds of unitary polar factors.

LEMMA 3.1. *Let W be an $n \times n$ matrix satisfying*

$$W + W^* = W^*W = WW^*\,. \tag{3.1}$$

(i) *If $W \in \mathbb{C}^{n\times n}$, then there exists a unitary matrix P such that*

$$P^*WP = \mathrm{diag}(d_1, d_2, \ldots, d_n), \tag{3.2}$$

where

$$d_j = a_j \pm i\sqrt{2a_j - a_j^2}, \quad 0 \leq a_j \leq 2. \tag{3.3}$$

(ii) *If $W \in \mathbb{R}^{n\times n}$, there exists a real orthonormal matrix P such that*

$$P^T W P = \operatorname{diag}(W_1, W_2, \ldots, W_k), \tag{3.4}$$

where each W_j is either a real 1×1 matrix, in this case $W_j = 0$ or 2, or a real 2×2 matrix of the form

$$W_j = \begin{pmatrix} a_j & \pm\sqrt{2a_j - a_j^2} \\ \mp\sqrt{2a_j - a_j^2} & a_j \end{pmatrix}, \quad 0 < a_j < 2. \tag{3.5}$$

Proof. By (3.1) W is a normal matrix. By some basic matrix theory (e.g., see [7]), there exists a unitary (real orthonormal matrix) P such that (3.2) (equation (3.4)) holds. Then by using the unitary (orthonormal) transformation, (3.1) becomes

$$\widehat{W} + \widehat{W}^* = \widehat{W}^*\widehat{W} = \widehat{W}\widehat{W}^*,$$

where $\widehat{W} = P^* W P$.

In the complex case,

$$2\mathcal{R}e\,(d_i) = |d_i|^2 .$$

Solving the above equation gives (3.3).

In the real case, each W_j is either a real 1×1 matrix or a real 2×2 matrix of the form (e.g., see [7, Theorem 2.5.8])

$$W_j = \begin{pmatrix} w_{j1} & w_{j2} \\ -w_{j2} & w_{j1} \end{pmatrix},$$

where w_{j1} and w_{j2} are real. Equation (3.5) can be obtained by solving

$$W_j + W_j^T = W_j^T W_j = W_j W_j^T . \tag{3.6}$$

□

Lemma 3.1 describes the structure of matrices satisfying (3.1). For those W_j in (3.5),

$$||W_j||_F^2 = 2||W_j||_2^2 = 4a_j .$$

More general discussion for the solution of (3.1) can be found in [14]. By noting these features in Lemma 3.1, we can obtain some new perturbation bounds.

Let A and $\widetilde{A}$ be $n\times n$ nonsingular matrices having the SVDs in (2.9). Let $S = \widetilde{U}^* U$ and $T = \widetilde{V}^* V$. Then S and T are unitary and

$$\|S - T\| = \|\widetilde{U}^*(UV^* - \widetilde{U}\widetilde{V}^*)V\| = \|Q - \widetilde{Q}\|$$

for any unitarily invariant norm $\|\cdot\|$. Since $m = n = r$, by the definition of M and $\widetilde{M}$ in section 2 we have

$$M = 2I - S^*T - T^*S \quad \text{and} \quad \widetilde{M} = 2I - TS^* - ST^* .$$

Let

$$W = I - S^*T \quad \text{and} \quad \widetilde{W} = I - TS^* .$$

Then $W = U^*(Q - \widetilde{Q})V$, $\widetilde{W} = \widetilde{V}^*(\widetilde{Q}^* - Q^*)\widetilde{U}$, and

$$\|W\| = \|\widetilde{W}\| = \|Q - \widetilde{Q}\| \tag{3.7}$$

by noting $Q = UV^*$ and $\widetilde{Q} = \widetilde{U}\widetilde{V}^*$. Since S and T are unitary, we have

$$\begin{aligned} M &= (I - S^*T)(I - S^*T)^* = WW^* = W^*W, \\ \widetilde{M} &= (I - TS^*)(I - TS^*)^* = \widetilde{W}\widetilde{W}^* = \widetilde{W}^*\widetilde{W}\,. \end{aligned} \tag{3.8}$$

By (3.8) and the definition of M and $\widetilde{M}$, it is easy to see that W and $\widetilde{W}$ satisfy (3.1).

LEMMA 3.2. *Let* $\Gamma = \operatorname{diag}(\sigma_1 - \sigma, \ldots, \sigma_n - \sigma)$ *and* $\widetilde{\Gamma} = \operatorname{diag}(\widetilde{\sigma}_1 - \sigma, \ldots, \widetilde{\sigma}_n - \sigma)$ *be two* $n \times n$ *diagonal matrices, where* $0 < \sigma_n \le \cdots \le \sigma_1$ *and* $0 < \widetilde{\sigma}_n \le \cdots \le \widetilde{\sigma}_1$. *Then*

$$\begin{aligned} &2\mathcal{R}e\,\operatorname{tr}[(S - T)(S\Gamma - \widetilde{\Gamma}T)^T] \\ &\quad \ge (\widetilde{\sigma}_{n-1} + \sigma_{n-1} - 2\sigma)||Q - \widetilde{Q}||_F^2 - (\sigma_{n-1} + \widetilde{\sigma}_{n-1} - \sigma_n - \widetilde{\sigma}_n)||Q - \widetilde{Q}||_2^2. \end{aligned} \tag{3.9}$$

Proof. By (3.7) and (3.8),

$$\operatorname{tr}(M) = ||W||_F^2 = ||Q - \widetilde{Q}||_F^2 \tag{3.10}$$

and

$$m_{nn} = W_{n*}W_{n*}^* = ||W_{n*}||_F^2, \tag{3.11}$$

where B_{i*} denotes the ith row of the matrix B. Since $W = U^*(Q - \widetilde{Q})V$, we have $W_{n*} = U_{n*}^*(Q - \widetilde{Q})V$. Then

$$\|W_{n*}\|_F = \|U_{n*}^*(Q - \widetilde{Q})\|_F \le \|Q - \widetilde{Q}\|_2, \tag{3.12}$$

and therefore

$$m_{nn} \le ||Q - \widetilde{Q}||_2\,.$$

Similarly,

$$\widetilde{m}_{nn} \le ||Q - \widetilde{Q}||_2\,.$$

Equation (3.9) is obtained by using Lemma 2.3 and noting $r = n = m$. ∎

Using Lemma 3.2 and noting the proof of Theorem 2.4 leads to

$$\sigma(\widetilde{\sigma}_{n-1} + \sigma_{n-1} - \sigma)\|Q - \widetilde{Q}\|_F^2 - \sigma(\sigma_{n-1} + \widetilde{\sigma}_{n-1} - \sigma_n - \widetilde{\sigma}_n)\|Q - \widetilde{Q}\|_2^2 \le \|E\|_F^2 \tag{3.13}$$

for any $\sigma > 0$.

The following theorem gives a perturbation bound involving both the Frobenius norm and the spectral norm.

THEOREM 3.3. *Let* A *and* $\widetilde{A}$ *be two* $n \times n$ *nonsingular matrices with SVDs given in* (2.9), *and let* $\sigma_1 \ge \cdots \ge \sigma_n > 0$ *and* $\widetilde{\sigma}_1 \ge \cdots \ge \widetilde{\sigma}_n > 0$ *be the singular values of* A *and* $\widetilde{A}$, *respectively. Then*

$$\left[(1 - \alpha)\left(\frac{\sigma_{n-1} + \widetilde{\sigma}_{n-1}}{2}\right) + \alpha\left(\frac{\sigma_n + \widetilde{\sigma}_n}{2}\right)\right] ||Q - \widetilde{Q}||_F \le \|E\|_F\,, \tag{3.14}$$

where $\alpha = \frac{||Q-\widetilde{Q}||_2^2}{||Q-\widetilde{Q}||_F^2}$.

Proof. Equation (3.14) can be obtained by maximizing the left side of (3.13) over σ. ∎

Remark 2. For the Frobenius norm, we can obtain Li's bound (1.8) by noting the fact that $\alpha \leq 1$. When $\alpha < 1$, we can obtain some better bounds. Particularly, if

$$\alpha = \frac{\|Q-\widetilde{Q}\|_2^2}{\|Q-\widetilde{Q}\|_F^2} \leq \frac{1}{2} \tag{3.15}$$

or

$$\frac{\|E\|_F}{\|E\|_2} \geq \sqrt{2} + \frac{\sigma_{n-1}+\widetilde{\sigma}_{n-1}}{\sigma_n+\widetilde{\sigma}_n},$$

we have

$$\|Q-\widetilde{Q}\|_F \leq \frac{4}{\widetilde{\sigma}_{n-1}+\widetilde{\sigma}_n+\sigma_{n-1}+\sigma_n}\|E\|_F \tag{3.16}$$

by using (1.8) and (3.13) with $\sigma = \frac{\widetilde{\sigma}_{n-1}+\widetilde{\sigma}_n+\sigma_{n-1}+\sigma_n}{4}$. Condition (3.15) is not always satisfied; some examples will be given in section 4. However, it is always true for real matrices with the small perturbation E.

THEOREM 3.4. *Let* $A, \widetilde{A} \in \mathbb{R}_n^{n\times n}$ *and* $||E||_2 < \sigma_n + \widetilde{\sigma}_n$. *Then the inequality* (3.16) *holds.*

Proof. By the assumption of the theorem and (1.8) we have

$$\|\widetilde{Q}-Q\|_2 \leq \frac{2}{\sigma_n+\widetilde{\sigma}_n}\|E\|_2 < \frac{2}{\sigma_n+\widetilde{\sigma}_n}(\sigma_n+\widetilde{\sigma}_n) = 2\,.$$

By (3.7) and Lemma 3.1(ii),

$$\max_j \|W_j\|_2 = \|W\|_2 = \|\widetilde{Q}-Q\|_2 < 2\,.$$

Let $\|W_p\|_2 = \max_j \|W_j\|_2$. Then by (3.5) and (3.7),

$$\|\widetilde{Q}-Q\|_F = \|W\|_F \geq \|W_p\|_F = \sqrt{2}\|W_p\|_2 = \sqrt{2}\|\widetilde{Q}-Q\|_2\,.$$

Equation (3.16) follows immediately from Remark 2. ∎

Remark 3. For real matrices, perturbation bounds of unitary polar factors have been studied by Barrlund [1] and Mathias [11], respectively. Mathias's bounds (Theorems 2.3 and 2.4 of [11]) when restricted to the Frobenius norm are as follows: (i) If $\sigma_1(E) < \sigma_n(A)$,

$$\|Q-\widetilde{Q}\|_F \leq -\frac{2\|E\|_F}{\|E\|_2}\log\left(1-\frac{\|E\|_2}{\sigma_n+\sigma_{n-1}}\right), \tag{3.17}$$

and (ii) if $A+tE$ is nonsingular for all $t \in [0,1]$,

$$\|Q-\widetilde{Q}\|_F \leq \max_{0\leq t\leq 1}\left\{\frac{2}{\sigma_n(A+tE)+\sigma_{n-1}(A+tE)}\right\}\|E\|_F. \tag{3.18}$$

Neither of these two bounds is uniformly better than the other. Our bound in (3.16) is slightly better than the second one. The conditions are different. The following

example shows that for any real matrices, the condition $\|E\|_2 < \sigma_n + \widetilde{\sigma}_n$ is also necessary for our bound.

Example 2. For any $A \in \mathbb{R}^{2\times 2}$ with the SVD $A = U\Sigma V^T$, let

$$\widetilde{A} = U\Sigma((I + D)V)^T$$

be the SVD of $\widetilde{A}$, where

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix}$$

with $\sigma_1 > \sigma_2$. Clearly, $\|E\|_F = \|E\|_2 = 2\sigma_2 = \sigma_2 + \widetilde{\sigma}_2$. On the other hand, we have $\|Q - \widetilde{Q}\|_F^2 = 4$. Then

$$2 = \|Q - \widetilde{Q}\|_F > \frac{4}{\sigma_2 + \widetilde{\sigma}_2 + \sigma_1 + \widetilde{\sigma}_1}\|E\|_F = \frac{4\sigma_2}{\sigma_2 + \sigma_1}.$$

4. Optimal bounds and more examples. It has been noted by many people and also can be seen from those previous bounds [1, 9, 10, 11, 13] and bounds obtained in this paper that there is some difference between the bounds in real and complex cases.

- For complex matrices, perturbation bounds of unitary polar factors are proportional to the reciprocal of the smallest singular values of A and $\widetilde{A}$.
- For real matrices, perturbation bounds are proportional to the reciprocal of the sum of the two smallest singular values of A and $\widetilde{A}$ under two conditions: (i) $r = m = n$ and (ii) $||E||_2 < \sigma_n + \widetilde{\sigma}_n$.

For real matrices, Li [10] gave an example to show that it is not true even in the case $m > n = r$. The following example shows that without the condition $||E||_2 < \sigma_n + \widetilde{\sigma}_n$, it is also not true even in the case $r = n = m$.

Example 3. Let A and $\widetilde{A}$ be defined in Example 2 with

$$D = \begin{pmatrix} a & \sqrt{-2a - a^2} \\ \sqrt{-2a - a^2} & -2 - a \end{pmatrix}, \quad -2 \le a \le 0.$$

We have $\|E\|_F^2 = \|E\|_2^2 = 2|a|(\sigma_1^2 - \sigma_2^2) + 4\sigma_2^2$ and $\|\widetilde{Q} - Q\|_F^2 = 4$. Then

$$\|Q - \widetilde{Q}\|_F^2 = \frac{4}{2|a|(\sigma_1^2 - \sigma_2^2) + 4\sigma_2^2}\|E\|_F^2. \tag{4.1}$$

The bound in (4.1) mainly depends upon σ_2 when $|a|$ is very small.

Without any restriction on $||E||$, $\|Q - \widetilde{Q}\|$ in both real and complex cases should be proportional to the reciprocal of the smallest singular value. In fact, the perturbation matrix W ($||W||_F = ||Q - \widetilde{Q}||_F$) in complex cases is always similar to a diagonal matrix, and in real cases it is similar to a block diagonal matrix with 1×1 block or 2×2 block in (3.5). In the first case the perturbation may be in rank-1 matrix space, and the second case means that the perturbation belongs to the matrix space of at least rank 2 except the trivial case $Q = \widetilde{Q}$. When the condition $\|E\|_2 < \sigma_n + \tilde{\sigma}_n$ is imposed, the rank of the perturbation is either zero or larger than or equal to 2, where $\|Q - \widetilde{Q}\|_F^2 \ge 2\|Q - \widetilde{Q}\|_2^2$. This feature has been used in the proof of Theorem 3.4.

To confirm the sharpness of a perturbation bound, one often gives specific A and $\widetilde{A}$, such as unitary matrices, such that the bounds can be achieved. One question arising here is whether a perturbation bound is achieved for any matrix A and some

E. For complex matrices, the following example shows that our bound in (2.8) for any r and $m \geq n$ with or without the restriction of $\|E\|$ being small enough is optimal in this sense.

Example 4. Let A be a 2×2 matrix and let $A = U\Sigma V^*$ be its SVD. Let $\widetilde{A} = U\Sigma(V(I+D))^*$, i.e., $\widetilde{U} = U$, $\widetilde{\Sigma} = \Sigma$, and $\widetilde{V} = V(I+D)$, where

$$D = \begin{pmatrix} 0 & 0 \\ 0 & -a + i\sqrt{2a - a^2} \end{pmatrix}, \quad 0 < a < 2.$$

A straightforward calculation gives

$$\|Q - \widetilde{Q}\|_F = \|UD^*V^*\|_F = \|D^*\|_F = \sqrt{2a}$$

and

$$\|A - \widetilde{A}\|_F = \|U\Sigma DV^*\|_F = \|\Sigma D\|_F = \sqrt{2a}\sigma_2\,.$$

Thus the bound in (2.8) is achieved. It is easy to extend this example to the $m \times n$ case.

It is more complicated in the real case. The optimal bound without the restriction $\|E\|_2 < \sigma_n + \tilde{\sigma}_n$ is the same as in (2.8), which has been confirmed by Example 3. However, the optimal bound with the above restriction is not clear. Mathias claimed the following bound in the abstract of [11]:

$$\|\widetilde{Q} - Q\|_F \leq \frac{2}{\sigma_n + \sigma_{n-1}}\|E\|_F \tag{4.2}$$

with the restriction $\|E\|_2 < \sigma_n$. There could be some typographical error since in the text of [11] he proved only the bounds (3.17) and (3.18) for unitarily invariant norm. Although the bound in (4.2) looks very close to our bound in (3.16), the following example shows that (4.2) is not true.

Example 5. Let A, $\widetilde{A} \in \mathbb{R}^{2\times 2}$ with the SVDs $A = U\Sigma V^T$ and $\widetilde{A} = U\widetilde{\Sigma}(V(I+D))^T$, where

$$D = \begin{pmatrix} -a & \sqrt{2a - a^2} \\ -\sqrt{2a - a^2} & -a \end{pmatrix}.$$

Let $\widetilde{\sigma}_1 = \widetilde{\sigma}_2 = 1$, $\sigma_1 = \sigma_2 = 6$, and $a = 0.7$. Then $\|E\|_2 = 5.7793 < \sigma_2 = 6$. However,

$$1.8556 = \left(\frac{2}{\sigma_1 + \sigma_2}\right)^2 \|E\|_F^2 < \|Q - \widetilde{Q}\|_F^2 = 2.8\,.$$

Acknowledgment. The authors would like to thank the anonymous referees for their valuable comments which led to the improvement of Theorem 3.3.

REFERENCES

[1] A. Barrlund, *Perturbation bounds on the polar decomposition*, BIT, 30 (1989), pp. 101–113.
[2] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[3] R. Bhatia, *Matrix factorizations and their perturbations*, Linear Algebra Appl., 197/198 (1994), pp. 245–276.
[4] R. Bhatia and K. Mukherjea, *Variation of the unitary part of a matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1007–1014.

[5] F. Chatelin and S. Gratton, *One the condition numbers associated with the polar factorization of a matrix*, Numer. Linear Algebra Appl., 7 (2000), pp. 337–354.
[6] N.J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
[7] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[8] C. Kenney and A.J. Laub, *Polar decomposition and matrix sign function condition estimates*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 488–504.
[9] R.C. Li, *A perturbation bound for the generalized polar decomposition*, BIT, 33 (1993), pp. 304–308.
[10] R.C. Li, *New perturbation bounds for the unitary polar factor*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 327–332.
[11] R. Mathias, *Perturbation bounds for the polar decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 588–597.
[12] G.W. Stewart and J.G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[13] J.G. Sun and C.H. Chen, *Generalized polar decomposition*, Math. Numer. Sinica, 11 (1989), pp. 262–273.
[14] X. Sun and C. Bischof, *A basis-kernel representation of orthogonal matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1184–1196.

SIAM J. MATRIX ANAL. APPL.
Vol. 23, No. 4, pp. 1194–1196
© 2002 Society for Industrial and Applied Mathematics

ON THE LIMIT OF SOME TOEPLITZ-LIKE DETERMINANTS*

CRAIG A. TRACY† AND HAROLD WIDOM‡

Abstract. In this article we derive, using standard methods of Toeplitz theory, an asymptotic formula for certain large minors of Toeplitz matrices. Bump and Diaconis obtained the same asymptotics using representation theory, with an answer having a different form.

Key words. Toeplitz determinant, Szegö limit theorem, Wiener–Hopf factorization

AMS subject classification. 47B35

PII. S0895479801395367

Our Toeplitz-like matrices are of the form

$$M = (c_{p_i - q_j}), \quad i, j = 0, 1, \ldots,$$

where $\{p_i\}$ and $\{q_i\}$ are sequences of integers satisfying $p_i = q_i = i$ for i sufficiently large, say for $i \geq m$. These are a particular class of finite-rank perturbations of Toeplitz matrices. If the p_i and the q_i are all different, then after rearranging the first m rows and columns, these become minors of the Toeplitz matrix (c_{i-j}) obtained by removing finitely many rows and columns.

Recently Bump and Diaconis [1], using the representation theory of the symmetric group, obtained an asymptotic formula for the determinants of large sections of these minors. Here we use "Toeplitz" methods to obtain the asymptotics in quite a different form, although the answers must be the same.

We assume that $\{c_i\}$ is the sequence of Fourier coefficients of a bounded function φ, so that $(c_{i-j}) = T(\varphi)$ in the usual notation. We assume also that $T(\varphi)$ is invertible on the space $\ell^2(\mathbf{Z}^+)$, and that is almost all. (We shall explain this below.)

For convenience in notation we consider the $(m+n) \times (m+n)$ sections of M,

$$M_{m+n} = (c_{p_i - q_j}), \quad i, j = 0, 1, \ldots, m+n-1,$$

and denote by $T_n(\varphi)$ the $n \times n$ Toeplitz matrix $(c_{i-j})_{i,j=0,1,\ldots,n-1}$.

If $T(\varphi)$ is invertible, then φ has a factorization $\varphi = \varphi^- \varphi^+$ where the functions $(\varphi^+)^{\pm 1}$ and $(\varphi^-)^{\pm 1}$ belong to H^2 and $\overline{H^2}$, respectively.[1] This is the "Wiener–Hopf factorization" of φ. In terms of these factors (and with subscripts denoting Fourier coefficients) our limit formula is

$$\lim_{n\to\infty} \frac{\det M_{m+n}}{\det T_n(\varphi)} = \det \left(\sum_{k=1}^{\infty} (\varphi^-)_{p_i+k-m} \, (\varphi^+)_{-q_j-k+m} \right)_{i,\, j=0,\ldots,m-1}. \tag{1}$$

*Received by the editors September 18, 2001; accepted for publication by L. Eldén December 19, 2001; published electronically May 10, 2002.
http://www.siam.org/journals/simax/23-4/39536.html

†Department of Mathematics and Institute of Theoretical Dynamics, University of California, Davis, CA 95616 (tracy@itd.ucdavis.edu). The research of this author was supported by National Science Foundation grant DMS-9802122.

‡Department of Mathematics, University of California, Santa Cruz, CA 95064 (widom@math.ucsc.edu). The research of this author was supported by National Science Foundation grant DMS-9732687.

[1] Recall that H^2 consists of the L^2 functions whose Fourier coefficients with negative index all vanish. The sequences of Fourier coefficients with nonnegative indices of φ^+ and $\overline{\varphi^-}$ are, up to constant factors, $T(\varphi)^{-1}\delta$, respectively, $T(\overline{\varphi})^{-1}\delta$, where δ is the sequence $\{1, 0, 0, \ldots\}$.

(The sum in the determinant on the right side of (1) has only finitely many nonzero terms since the Fourier coefficient $(\varphi^-)_k$ vanishes for $k > 0$ and $(\varphi^+)_k$ vanishes for $k < 0$.)

Under some additional hypotheses the strong Szegö limit theorem gives the asymptotics of the Toeplitz determinant:

$$\det T_n(\varphi) \sim G(\varphi)^n \, E(\varphi),$$

where

$$G(\varphi) = \exp\left\{\frac{1}{2\pi}\int \log\varphi(\theta)\,d\theta\right\}, \quad E(\varphi) = \exp\left\{\sum_{k=1}^{\infty} k(\log\varphi)_k\,(\log\varphi)_{-k}\right\}.$$

Our result holds without these extra hypotheses. What we do need, which is a little stronger than the invertibility of $T(\varphi)$, is the uniform invertibility of the $T_n(\varphi)$. That is, we require that the $T_n(\varphi)$ be invertible for sufficiently large n and that the norms of the inverse matrices $T_n(\varphi)^{-1}$ be bounded as $n \to \infty$. This holds in all "normal" cases where $T(\varphi)$ is invertible. The known counterexamples are not simple. (For a discussion of these points, see [2], especially Chapter 2.) If φ has a continuous logarithm, for example, then the $T_n(\varphi)$ are uniformly invertible.[2]

THEOREM. *If the $T_n(\varphi)$ are uniformly invertible, then* (1) *holds.*

Proof. We use the fact from linear algebra that if we have a matrix

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

with A and D square and if D is invertible, then

$$\det M = \det D \, \det(A - BD^{-1}C).$$

In our case $M = M_{m+n}$, $A = (c_{p_i-q_j})_{i,j=0,\dots,m-1}$ and D is the Toeplitz matrix $T_n(\varphi)$. If the indices for B and C start at 0, then their entries are given by

$$B_{i,k} = c_{p_i-m-k}, \quad C_{k,j} = c_{m+k-q_j} \qquad (0 \le i,j \le m-1, \;\; 0 \le k \le n-1), \tag{2}$$

and we are interested in the limit as $n \to \infty$ of the i,j entry of the $m \times m$ matrix $BT_n(\varphi)^{-1}C$.

Now we use the fact that if the $T_n(\varphi)$ are uniformly invertible, then $T_n(\varphi)^{-1}$ converges strongly to the infinite Toeplitz matrix $T(\varphi)^{-1}$. (See [2, Proposition 2.2].) It follows that each entry of $BT_n(\varphi)^{-1}C$ converges to the corresponding entry of $BT(\varphi)^{-1}C$, where now B and C are the $m \times \infty$ and $\infty \times m$ matrices, respectively, with entries given by (2) but now with $k \in \mathbf{Z}^+$. What remains is to show that the i,j entry of $A - BT(\varphi)^{-1}C$ is given by the summand on the right side of (1).

It is convenient to extend the range of the row or column indices of our Toeplitz matrices so that one or the other can run over $\mathbf{Z}$ rather than $\mathbf{Z}^+$. So we introduce the notations $T^{(r)}(\varphi)$ and $T^{(c)}(\varphi)$ for the matrices in which the row, respectively, the column, index runs over $\mathbf{Z}$. With this notation, we see that

$$B_{i,k} = T^{(r)}(\varphi)_{p_i-m,\,k}, \quad C_{k,j} = T^{(c)}(\varphi)_{k,\,q_j-m}.$$

[2]For such a φ the Wiener–Hopf factors φ^- and φ^+ are, up to constant factors, the exponentials of the portions of the Fourier series of $\log\varphi$ corresponding to negative, respectively, positive, indices.

Thus

$$(BT(\varphi)^{-1}C)_{i,j} = (T^{(r)}(\varphi)\,T(\varphi)^{-1}\,T^{(c)})_{p_i-m,\,q_j-m}.$$

Now it is well known and easy to check (see [2, Proposition 1.13]) that

$$T(\varphi) = T(\varphi^-)\,T(\varphi^+), \quad T(\varphi)^{-1} = T(\varphi^+)^{-1}\,T(\varphi^-)^{-1}$$

and just as easy to check that

$$T^{(r)}(\varphi) = T^{(r)}(\varphi^-)\,T(\varphi^+), \quad T^{(c)}(\varphi) = T(\varphi^-)\,T^{(c)}(\varphi^+).$$

It follows that

$$T^{(r)}(\varphi)\,T(\varphi)^{-1}\,T^{(c)}(\varphi) = T^{(r)}(\varphi^-)\,T^{(c)}(\varphi^+).$$

Hence

$$(BT(\varphi)^{-1}C)_{i,j} = \sum_{k=0}^{\infty}(\varphi^-)_{p_i-m-k}\,(\varphi^+)_{m+k-q_j}$$

$$= \sum_{k=-\infty}^{\infty}(\varphi^-)_{p_i-m-k}\,(\varphi^+)_{k-q_j+m} - \sum_{k=-\infty}^{-1}(\varphi^-)_{p_i-m-k}\,(\varphi^+)_{k-q_j+m}$$

$$= \varphi_{p_i-q_j} - \sum_{k=1}^{\infty}(\varphi^-)_{p_i-m+k}\,(\varphi^+)_{-k-q_j+m}.$$

The first term on the right is $A_{i,j}$, and so the theorem is established.

REFERENCES

[1] D. Bump and P. Diaconis, *Toeplitz minors*, J. Combin. Theory Ser. A., 97 (2002), pp. 252–271.
[2] A. Böttcher and B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices*, Springer-Verlag, Heidelberg, 1999.