Numer. Math. (1997) 76: 279–308

Numerische Mathematik

Electronic Edition

An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems

Zhaojun Bai[1,⋆], **James Demmel**[2,⋆⋆], **Ming Gu**[3,⋆⋆⋆]

[1] Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA
[2] Computer Science Division and Mathematics Department, University of California, Berkeley, CA 94720, USA
[3] Department of Mathematics and Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720, USA

Received September 20, 1994 / Revised version received February 5, 1996

Summary. We discuss an inverse-free, highly parallel, spectral divide and conquer algorithm. It can compute either an invariant subspace of a nonsymmetric matrix A, or a pair of left and right deflating subspaces of a regular matrix pencil $A - \lambda B$. This algorithm is based on earlier ones of Bulgakov, Godunov and Malyshev, but improves on them in several ways. This algorithm only uses easily parallelizable linear algebra building blocks: matrix multiplication and QR decomposition, but not matrix inversion. Similar parallel algorithms for the nonsymmetric eigenproblem use the matrix sign function, which requires matrix inversion and is faster but can be less stable than the new algorithm.

Mathematics Subject Classification (1991): 65F15

1. Introduction

We are concerned with the following two computational problems.

1. For a given $n \times n$ nonsymmetric matrix A, we want to find an invariant subspace $\mathscr{R}$ (i.e. $A\mathscr{R} \subseteq \mathscr{R}$) corresponding to the eigenvalues of A in a specified region $\mathscr{D}$ of the complex plane. In other words, we want to find a unitary matrix $Q = (Q_1,\ Q_2)$ with $\mathscr{R} = \text{span}\{Q_1\}$ such that

⋆ The author was supported in part by NSF grant ASC-9102963 and in part by ARPA grant DM28E04120 via a subcontract from Argonne National Laboratory
⋆⋆ The author was supported in part by NSF grant ASC-9005933, ARPA contact DAAL03-91-C-0047 via a subcontract from the University of Tennessee, DOE grant DE-FG03-94ER25219, NSF grant ASC-9313958, and ARPA grant DM28E04120 via a subcontract from Argonne National Laboratory
⋆⋆⋆ The author was supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098
Correspondence to: J. Demmel

$$Q^{\mathrm{H}}AQ = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \tag{1.1}$$

and the eigenvalues of A_{11} are the eigenvalues of A in $\mathscr{D}$. We shall call this problem an *(ordinary) spectral divide and conquer (SDC) problem*.

2. A *regular* matrix pencil $A - \lambda B$ is a square pencil such that $\det(A - \lambda B)$ is not identically zero. Given such an n by n nonsymmetric pencil, we want to find a pair of left and right deflating subspaces $\mathscr{L}$ and $\mathscr{R}$ (i.e. $A\mathscr{R} \subseteq \mathscr{L}$ and $B\mathscr{R} \subseteq \mathscr{L}$) corresponding to the eigenvalues of the pair $A - \lambda B$ in a specified region $\mathscr{D}$ on complex plane. In other words, we want to find a unitary matrix $Q_{\mathrm{L}} = (Q_{\mathrm{L}1},\ Q_{\mathrm{L}2})$ with $\mathscr{L} = \mathrm{span}\{Q_{\mathrm{L}1}\}$, and a unitary matrix $Q_{\mathrm{R}} = (Q_{\mathrm{R}1},\ Q_{\mathrm{R}2})$ with $\mathscr{R} = \mathrm{span}\{Q_{\mathrm{R}1}\}$, such that

$$Q_{\mathrm{L}}^{\mathrm{H}}AQ_{\mathrm{R}} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \text{ and } Q_{\mathrm{L}}^{\mathrm{H}}BQ_{\mathrm{R}} = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}, \tag{1.2}$$

and the eigenvalues of $A_{11} - \lambda B_{11}$ are the eigenvalues of $A - \lambda B$ in the region $\mathscr{D}$. We shall call this problem a *generalized spectral divide and conquer (SDC) problem*.

The region $\mathscr{D}$ in the above problems will initially just be the interior (or exterior) of the unit disk. By employing Möbius transformations $(\alpha A+\beta B)(\gamma A+\delta B)^{-1}$ and divide-and-conquer, $\mathscr{D}$ can be the union of intersections of arbitrary half planes and (complemented) disks, and so a rather general region. We will assume that the given matrix A or matrix pencil $A - \lambda B$ has no eigenvalues on the boundary $\mathscr{D}$ (in practice this means we might enlarge or shrink $\mathscr{D}$ slightly if we fail to converge).

The nonsymmetric eigenproblem and its generalized counterpart are important problems in numerical linear algebra, and have until recently resisted attempts at effective parallelization. The standard serial algorithm for the spectral divide and conquer problem is to use the QR algorithm (or the QZ algorithm in the generalized case) to reduce the matrix (or pencil) to Schur form, and then to reorder the eigenvalues on the diagonal of the Schur form to put the eigenvalues in $\mathscr{D}$ in the upper left corner, as shown in (1.1) and (1.2) (see [8] and the references therein). The approach is numerically stable, although in some extremely ill-conditioned cases, the swapping process may fail[1]. Although some thought this approach was too fine grain to parallelize easily [23], the QR iteration itself was recently parallelized successfully [34]. While this parallelization scheme works well for a modest number of processors, it may not scale as well to very large numbers of processors as our approach. Also, it must compute all or most eigenvalues even if only a few are desired. For these reasons, we will pursue the divide and conquer approach.

There are two highly parallel algorithms for the spectral divide and conquer problem, those based on the *matrix sign function* (which we describe in Sect. 3),

[1] Recently Bojanczyk and Van Dooren [13] have found a way to eliminate this possibility, although the theoretical possibility of nonconvergence of the QR algorithm remains [11]

and an *inverse-free method* based on original algorithms of Bulgakov, Godunov and Malyshev [30, 16, 40, 41, 42], which is the main topic of this paper. Both kinds of algorithms are easy to parallelize because they require only large matrix operations which have been successfully parallelized on most existing machines: matrix-matrix multiplication, QR decomposition and (for the sign function) matrix inversion. The price paid for the easy parallelization of these algorithms is potential loss of stability compared to the QR or QZ algorithms; they can fail to converge in a number of circumstances in which the QR and QZ algorithms succeed. Fortunately, it is usually easy to detect and compensate for this loss of stability, by choosing to divide and conquer the spectrum in a slightly different location.

In brief, the difference between the sign-function and inverse-free methods is as follows. The sign-function method is significantly faster than inverse-free when it converges, but there are some very difficult problems where the inverse-free algorithm gives a more accurate answer than the sign-function. This leads us to propose the following 3-step algorithm [22, 25]:

1. Try to use the matrix sign-function to split the spectrum. If it succeeds, stop.
2. Otherwise, if the sign-function fails, try to split the spectrum using the inverse-free algorithm. If it succeeds, stop.
3. Otherwise, if the inverse-free methods fails, use the QR (or QZ) algorithm.

This 3-step approach works by trying the fastest but least stable method first, falling back to slower but more stable methods only if necessary.

This paper is primarily concerned with an algorithm based on the pioneering work of Godunov, Bulgakov and Malyshev [30, 16, 40], in particular on the work of Malyshev [41, 42]. We have made the following improvements on their work:

- We have eliminated the need for matrix exponentials, thus making their algorithm truly practical. By expressing the algorithms for computing the ordinary and generalized spectral divide and conquer decompositions in a single framework, we in fact show it is equally easy to divide the complex plane along arbitrary circles and lines with the same amount of work.
- Our error analysis is simpler and tighter. In particular, our condition number can be as small as the square root of the condition number in [41], and is precisely the square of the reciprocal of the distance from $A - \lambda B$ to a natural set of ill-posed problems, those pencils which have an eigenvalue on the unit circle.
- We have simplified their algorithm by eliminating all inversions and related factorizations.
- We propose a realistic and inexpensive stopping criterion for the inner loop iteration.

Many simplifications in these algorithms are possible in case the matrix A is symmetric. The PRISM project, with which this work is associated, is also producing algorithms for the symmetric case; see [6, 12] for more details.

The rest of this paper is organized as follows. In Sect. 2 we present our algorithm for the ordinary and generalized spectral divide and conquer problems, discuss some implementation details and options, and show how to divide the spectrum along arbitrary circles and lines in the complex plane. In Sect. 3, we compare the cost of the new algorithm with the matrix sign function based algorithms. In Sect. 4, we explain why the new algorithm works, using a simpler explanation than in [41]. Section 5 derives a condition number, and Sect. 6 uses it to analyze convergence of the new algorithm. Section 7 does error analysis, and Sect. 8 contrasts our bounds to those of Malyshev [41]. Section 9 discusses the stopping criterion of the new algorithm. Section 10 presents numerical examples, Sect. 11 lists open problems, and Sect. 12 draws conclusions.

Throughout this paper we shall use the notational conventions in [31]: Matrices are denoted by upper case italic and Greek letters, vectors by lower-case italic letters, and scalars by lower-case Greek letters or lower-case italic if there is no confusion. The matrix A^{T} is the transpose of A, and A^{H} is the complex conjugate transpose of A. $\|\cdot\|$, $\|\cdot\|_{\mathrm{F}}$, and $\|\cdot\|_1$ are the spectral norm, Frobenius norm, and 1-norm of a vector or matrix, respectively. The condition number $\|A\| \cdot \|A^{-1}\|$ will be denoted $\kappa(A)$. $\lambda(A)$ and $\lambda(A, B)$ denote the sets of eigenvalues of the matrix A and the matrix pencil $A - \lambda B$, respectively. span$\{X\}$ is a subspace spanned by the columns of the matrix X. det(A) is the determinant of matrix A. The lower-case italic letter i equals $\sqrt{-1}$ throughout. Machine precision is denoted by ε.

2. Algorithm

Algorithm 1 below computes left and right deflating subspaces of a matrix pencil $A - \lambda B$ corresponding to the eigenvalues inside (or outside) the unit disk. When $B = I$, these left and right deflating subspaces are identical, both are equal to an invariant subspace of A, and only the first half of Algorithm 1 is necessary to compute this space. Since most of the results in this paper do not change when $B = I$, we will describe the case of general B, and remark on any simplifications when $B = I$.

Algorithm 1 is similar to the matrix sign function based algorithm in that it begins by computing orthogonal projectors onto the desired subspaces. Later, we will show how to divide into more general regions. Algorithm 1 applies to complex matrices A and B. But if A and B are real, then Algorithm 1 requires only real arithmetic.

2.1. Algorithm for spectral division of (A, B)

Algorithm 1. Given $n \times n$ matrices A and B, compute two unitary matrices Q_{L} and Q_{R}, such that

$$Q_{\mathrm{L}}^{\mathrm{H}} A Q_{\mathrm{R}} = \begin{pmatrix} A_{11} & A_{12} \\ E_{21} & A_{22} \end{pmatrix}, \quad Q_{\mathrm{L}}^{\mathrm{H}} B Q_{\mathrm{R}} = \begin{pmatrix} B_{11} & B_{12} \\ F_{21} & B_{22} \end{pmatrix},$$

and where in exact arithmetic we would have $\lambda(A_{11}, B_{11}) \subseteq \mathscr{D}$, $\lambda(A_{22}, B_{22}) \cap \mathscr{D} = \emptyset$, and $E_{21} = F_{21} = 0$. $\mathscr{D}$ can be the interior (or exterior) of the unit disk. We assume that no eigenvalues of the pencil (A, B) are on the unit circle. On return, the generally nonzero quantities $\|E_{21}\|_1/\|A\|_1$ and $\|F_{21}\|/\|B\|_1$ measure the stability of the computed decomposition.

When $B = I$, $Q_{\mathrm{L}} = Q_{\mathrm{R}}$, so $Q_{\mathrm{L}}^{\mathrm{H}} B Q_{\mathrm{R}} = I$ need not be computed.

/* *Part 1: Compute the right deflating subspace* */

1) Let $A_0 = A$ and $B_0 = B$.
2) For $j = 0, 1, 2, \ldots$ until convergence or $j > maxit$

$$\begin{pmatrix} B_j \\ -A_j \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} R_j \\ 0 \end{pmatrix}, \quad \text{(QR decomposition)}$$

$A_{j+1} = Q_{12}^{\mathrm{H}} A_j$;
$B_{j+1} = Q_{22}^{\mathrm{H}} B_j$;
if $\|R_j - R_{j-1}\|_1 \leq \tau \|R_{j-1}\|_1$, $p = j + 1$, exit;
End for
3) For the exterior of the unit disk, compute
$(A_p + B_p)^{-1} A_p = Q_{\mathrm{R}} R_{\mathrm{R}} \Pi_{\mathrm{R}}$, (rank revealing QR decomposition)
or for the interior of the unit disk, compute
$(A_p + B_p)^{-1} B_p = Q_{\mathrm{R}} R_{\mathrm{R}} \Pi_{\mathrm{R}}$, (rank revealing QR decomposition)
4) $l_{\mathrm{R}} = \mathrm{rank}(R_R)$, (the number of eigenvalues in the selected region.)
5) If $B = I$, set $Q_{\mathrm{L}} = Q_{\mathrm{R}}$ and go to step 11).

/* *Part 2: Compute the left deflating subspace* */

6) Let $A_0 = A^{\mathrm{H}}$ and $B_0 = B^{\mathrm{H}}$.
7) For A_0 and B_0 do the loop 2).
8) For the exterior of the unit disk, compute
$A_p^{\mathrm{H}} (A_p + B_p)^{-\mathrm{H}} = Q_{\mathrm{L}} R_{\mathrm{L}} \Pi_{\mathrm{L}}$, (rank revealing QR decomposition)
or for the interior of the unit disk, compute
$B_p^{\mathrm{H}} (A_p + B_p)^{-\mathrm{H}} = Q_{\mathrm{L}} R_{\mathrm{L}} \Pi_{\mathrm{L}}$, (rank revealing QR decomposition)
9) $l_{\mathrm{L}} = \mathrm{rank}(R_{\mathrm{L}})$, (the number of eigenvalues in the selected region.)
10) If $l_{\mathrm{R}} \neq l_{\mathrm{L}}$, signal an error and quit, otherwise let $l = l_{\mathrm{R}} = l_{\mathrm{L}}$;

/* *Part 3: Divide the pencil.* */

11) Compute $Q_{\mathrm{L}}^{\mathrm{H}} A Q_{\mathrm{R}} = \begin{matrix} \\ l \\ n-l \end{matrix} \begin{matrix} \begin{matrix} l & n-l \end{matrix} \\ \begin{pmatrix} A_{11} & A_{12} \\ E_{21} & A_{22} \end{pmatrix} \end{matrix}$, $Q_{\mathrm{L}}^{\mathrm{H}} B Q_{\mathrm{R}} = \begin{matrix} \\ l \\ n-l \end{matrix} \begin{matrix} \begin{matrix} l & n-l \end{matrix} \\ \begin{pmatrix} B_{11} & B_{12} \\ F_{21} & B_{22} \end{pmatrix} \end{matrix}$.
and $\|E_{21}\|_1/\|A\|_1$ and $\|F_{21}\|/\|B\|_1$ (If $B = I$, only $Q_{\mathrm{L}}^{\mathrm{H}} A Q_{\mathrm{R}}$ should be computed).

2.2. Implementation details and options

The main costs of Algorithm 1 are the matrix-matrix multiplications and the QR decomposition in the inner loop, and the rank-revealing QR following the inner loop. There is a large literature on parallel matrix-matrix multiplication and QR decomposition. They are usually among the first algorithms to be implemented quickly on a high performance architecture [26, 3].

In step 2), we assume that the QR decomposition of $\begin{pmatrix} B_j \\ -A_j \end{pmatrix}$ is computed so that the diagonal elements of R_j are all positive, so the matrix R_j is uniquely defined. This is needed for the stopping criterion "If $\|R_j - R_{j-1}\|_1 \leq \tau \|R_{j-1}\|_1$, $p = j + 1$, exit" to function correctly.

$\|E_{21}\|_1/\|A\|_1$ and $\|F_{21}\|_1/\|B\|_1$ are accurate measures of the backward stability of the algorithm because one proceeds by setting E_{21} and F_{21} to zero and continuing to divide and conquer. This introduces a backward error of precisely $\|E_{21}\|_1/\|A\|_1$ in A and $\|F_{21}\|_1/\|B\|_1$ in B.

We need to choose a stopping criterion τ in the inner loop of step 2), as well as a limit *maxit* on the maximum number of iterations. So far we have used $\tau \approx n\varepsilon$ (where ε is the machine precision) and *maxit* = 60. In Sect. 10 we shall discuss these issues again.

In finite precision arithmetic, it is possible that we might get two different numbers l_{R} and l_{L} of eigenvalues in region $\mathscr{D}$ in steps 4) and 9). Therefore, we need the test in step 10). In our numerical experiments, l_{R} and l_{L} have always been equal. If they were not, we would handle it the same way we handle other convergence failures: the spectral decomposition based on $\mathscr{D}$ is rejected, and a new region $\mathscr{D}$ must be selected (see Sect. 2.3).

Now we show how to compute Q_{R} in step 3) and Q_{L} in step 8) without computing the explicit inverse $(A_p + B_p)^{-1}$ and subsequent products. This yields the ultimate *inverse-free* algorithm. For simplicity, let us use column pivoting to reveal rank, although more sophisticated rank-revealing schemes exist [20, 32, 37, 50]. Recall that for our purposes, we only need the unitary factor Q and the rank of $C^{-1}D$ (or $D^{\mathrm{H}}C^{-\mathrm{H}}$). It turns out that by using the generalized QR (GQR) decomposition technique developed in [43, 4], we can get the desired information without computing C^{-1} or $C^{-\mathrm{H}}$. In fact, in order to compute the QR decomposition with pivoting of $C^{-1}D$, we first compute the QR decomposition with pivoting of the matrix D:

$$D = Q_1 R_1 \Pi, \tag{2.3}$$

and then we compute the RQ factorization of the matrix $Q_1^{\mathrm{H}}C$:

$$Q_1^{\mathrm{H}}C = R_2 Q_2. \tag{2.4}$$

From (2.3) and (2.4), we have $C^{-1}D = Q_2^{\mathrm{H}}(R_2^{-1}R_1)\Pi$. The Q_2 is the desired unitary factor. The rank of R_1 is also the rank of the matrix $C^{-1}D$. The rank revealing QR decomposition of $D^{\mathrm{H}}C^{-\mathrm{H}}$ is computed analogously, starting from the QL decomposition of C.

Note that the above GQR decomposition will not necessarily always reveal the numerical rank, even though it works much of the time. In particular, the permutation Π should really depend on both C and D. Another way to compute a *rank-revealing GQR decomposition* is to explicitly form $C^{-1}D$, compute its rank revealing QR, take the resulting permutation Π, and use this Π in decomposition (2.3). This costs quite a bit more, and Π is still not guaranteed to be correct

if $C^{-1}D$ is computed sufficient inaccurately. However, a more sophisticated implementation of this later idea can indeed reveal the numerical rank of $C^{-1}D$. Given the recently increasing speed of SVD implementations based on divide-and-conquer [33], one may just want to use the SVD instead.

The GQR decomposition is always backward stable in the following sense. The computed Q_2 is nearly the exact orthogonal factor for matrices $C + \delta C$ and $D + \delta D$, where $\|\delta C\| = O(\varepsilon)\|C\|$ and $\|\delta D\| = O(\varepsilon)\|D\|$.

Finally, we note that in some applications, we may only want the eigenvalues of the reduced matrix A_{11} or of the matrix pencil (A_{11}, B_{11}) or their subblocks. In this case, we do not need to compute the blocks A_{12}, A_{22}, B_{12} or B_{22} in step 11) of Algorithm 1, and so we can save some computations.

2.3. Other kinds of regions

Although Algorithm 1 only divides the spectrum along the unit circle, we can use Möbius transformations of the input matrix A or matrix pair (A, B) to divide along other curves (we treat A as the pair (A, I)). By transforming the eigenproblem $Az = \lambda Bz$ to

$$(\alpha A + \beta B)z = \frac{\alpha\lambda + \beta}{\gamma\lambda + \delta}(\gamma A + \delta B)z$$

and applying Algorithm 1 to $A_0 = \alpha A + \beta B$ and $B_0 = \gamma A + \delta B$, we see that we can split along the curve where $|\tilde{\lambda}| = |\frac{\alpha\lambda+\beta}{\gamma\lambda+\delta}| = 1$. This lets us divide the spectrum along arbitrary circles and straight lines, since any circle or straight line is the image of the unit circle under an appropriate Möbius transformation [1]. This is a major attraction of Algorithm 1: it can handle an arbitrary line or circle just by setting A_0 and B_0 to appropriate linear combinations of A and B. In contrast, using the matrix sign function to split the spectrum along an arbitrary line or circle will generally require a matrix inversion. This also eliminates the need for matrix exponentiation in Malyshev's algorithm [42], which was used to split along lines. We note that if the chosen circle is centered on the real axis, or if the chosen line is vertical, then all arithmetic will be real if A and B are real.

3. Inverse-free iteration vs. the matrix sign function

In this section we compare the cost of a single iteration of the new algorithm with the matrix sign function based algorithm. Numerical experiments will be presented in Sect. 10.

We begin by reviewing the matrix sign function. The sign function sign(A) of a matrix A with no eigenvalues on the imaginary axis can be defined via the Jordan canonical form of A: Let

$$A = X \begin{pmatrix} J_+ & 0 \\ 0 & J_- \end{pmatrix} X^{-1}$$

be the Jordan canonical form of A, where the eigenvalues of J_+ are in the open right half plane, and the eigenvalues of J_- are in the open left half plane. Then sign(A), as introduced by Roberts [45], is

$$\text{sign}(A) \equiv X \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} X^{-1}.$$

It is easy to show that the two matrices

$$P_+ = \frac{1}{2}(I + \text{sign}(A)) \quad \text{and} \quad P_- = \frac{1}{2}(I - \text{sign}(A)) \tag{3.5}$$

are the spectral projectors onto the invariant subspaces corresponding to the eigenvalues of A in the open right and open left half planes, respectively. Now let the *rank revealing QR decomposition* of the matrix P_+ be $P_+ = QR\Pi$, so that R is upper triangular, Q is unitary, and Π is a permutation matrix chosen so that the leading columns of Q span the range space of P_+. Then Q yields the desired spectral decomposition [7]:

$$Q^{\mathrm{H}} A Q = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

where the eigenvalues of A_{11} are the eigenvalues of A in open right half plane, and the eigenvalues of A_{22} are the eigenvalues of A in the open left half plane. By computing the sign function of Möbius transformations of A, the spectrum can be divided along arbitrary lines and circles.

The simplest scheme for computing the matrix sign function is the Newton iteration applied to $(\text{sign}(A))^2 = I$:

$$A_{j+1} = \frac{1}{2}(A_j + A_j^{-1}), \quad j = 0, 1, 2, \ldots \quad \text{with} \quad A_0 = A. \tag{3.6}$$

The iteration is globally and ultimately quadratically convergent with $\lim_{j\to\infty} A_j = \text{sign}(A)$ [45, 38]. The iteration could fail to converge if A has pure imaginary eigenvalues (or, in finite precision, if A is "close" to having pure imaginary eigenvalues.) There are many ways to improve the accuracy and convergence rates of this basic iteration [18, 35, 39].

The matrix sign function may also be used in the generalized eigenproblem $A - \lambda B$ by implicitly applying (3.6) to AB^{-1} [29]. We do not want to invert B if it is ill-conditioned, which is why we want to apply the previous algorithm implicitly. This leads to the following iteration:

$$A_{j+1} = \frac{1}{2}(A_j + BA_j^{-1}B), \quad j = 0, 1, 2, \ldots \quad \text{with} \quad A_0 = A. \tag{3.7}$$

A_j converges quadratically to a matrix C if B is nonsingular and $A - \lambda B$ has no pure imaginary eigenvalues. In this case CB^{-1} is the matrix sign function of AB^{-1}, and so following (3.5) we want to use the QR decomposition to calculate the range space of $P_\pm = \frac{1}{2}(I \pm CB^{-1})$, which has the same range space as $2P_\pm B = B \pm C$. Thus we can compute the invariant subspace of AB^{-1} (left deflating

subspace of $A - \lambda B$) without inverting B, by computing the rank revealing QR decomposition of $B \pm C$. The right deflating subspace of $A - \lambda B$ can be obtained by applying this algorithm to $A^{\mathrm{H}} - \lambda B^{\mathrm{H}}$, since transposing swaps right and left spaces.

Now we consider the convergence of (3.7) when B is singular, and $A - \lambda B$ has no pure imaginary eigenvalues. By considering the Weierstrass Canonical Form of $A - \lambda B$ [28], it suffices to consider $A_0 = I$ and B a nilpotent Jordan block. Then it is easy to show by induction that

$$A_j = 2^{-j} I + \frac{2^j - 2^{-j}}{3} B^2 + O(B^4)$$

so that A_j diverges to infinity if B is 3-by-3 or larger, and converges to 0 otherwise. In the latter case, the range space of $B \pm A_j$ converges to the space spanned by $e_1 = [1, 0, \ldots, 0]^{\mathrm{T}}$, which is indeed a left deflating subspace. The situation is more complicated in the former case.

By avoiding all explicit matrix inversions, and requiring only QR decomposition and matrix-matrix multiplication instead, our new algorithm may eliminate the possible instability associated with inverting ill-conditioned matrices. However, it does not avoid all accuracy or convergence difficulties associated with eigenvalues very close to the unit circle. In addition, the generalized eigenproblem has another possible source of difficulty: when $A - \lambda B$ is close to a singular pencil [28, 24]. We shall discuss this further in Sects. 5 and 7.

The advantage of the new approach is obtained at the cost of more storage and more arithmetic. For example, when the matrix A is real and $B = I$, Algorithm 1 needs $4n^2$ more storage space than standard Newton iteration (some other iterations for the sign function which converge faster than Newton require more storage). This will certainly limit the problem size we will be able to solve. The one loop of the inverse-free iteration for the standard SDC problem does about 6.7 times more arithmetic than the one loop of the Newton iteration. For the generalized SDC problem, it is about 2.2 times more arithmetic (see [10] for details). We expect that these extra expenses of the new approach will be compensated by better numerical stability in some cases, especially for the generalized eigenproblem; see Sect. 10.

4. Why the algorithm works

The simplest way we know to see why the algorithm works is as follows. We believe this is much simpler than the explanation in [41], for example.

For simplicity we will assume that all matrices we want to invert are invertible. Our later error analysis will not depend on this. It suffices to consider the first half of Algorithm 1. We will exhibit a basis for the pencil $A - \lambda B$ in which the transformations of the algorithm will be transparent. From step 2) of Algorithm 1, we see that

$$\begin{pmatrix} Q_{11}^{\mathrm{H}} & Q_{21}^{\mathrm{H}} \\ Q_{12}^{\mathrm{H}} & Q_{22}^{\mathrm{H}} \end{pmatrix} \begin{pmatrix} B_j \\ -A_j \end{pmatrix} = \begin{pmatrix} Q_{11}^{\mathrm{H}} B_j - Q_{21}^{\mathrm{H}} A_j \\ Q_{12}^{\mathrm{H}} B_j - Q_{22}^{\mathrm{H}} A_j \end{pmatrix} = \begin{pmatrix} R \\ 0 \end{pmatrix}$$

so $Q_{12}^{\mathrm{H}} B_j = Q_{22}^{\mathrm{H}} A_j$ or $B_j A_j^{-1} = Q_{12}^{-\mathrm{H}} Q_{22}^{H}$. Therefore

$$A_{j+1}^{-1} B_{j+1} = A_j^{-1} Q_{12}^{-\mathrm{H}} Q_{22}^{\mathrm{H}} B_j = (A_j^{-1} B_j)^2$$

so the algorithm is simply repeatedly squaring the eigenvalues, driving the ones inside the unit disk to 0 and those outside to ∞. Repeated squaring yields quadratic convergence. This is analogous to the sign function iteration where computing $(A+A^{-1})/2$ is equivalent to taking the Cayley transform $(A-I)(A+I)^{-1}$ of A, squaring, and taking the inverse Cayley transform. Therefore, in step 3) of Algorithm 1 we have

$$(A_p + B_p)^{-1} A_p = (I + A_p^{-1} B_p)^{-1} = (I + (A^{-1}B)^{2^p})^{-1} \ . \tag{4.8}$$

To see that this approaches a projector onto the right deflating subspace corresponding to eigenvalues outside the unit circle as required by the algorithm, we will use the the Weierstrass Canonical Form of the pencil $A - \lambda B$ [28]. Write

$$A - \lambda B = P_{\mathrm{L}}' \begin{pmatrix} J_0 - \lambda I & \\ & J_\infty - \lambda N \end{pmatrix} P_{\mathrm{R}}^{-1}$$

where P_{L}' and P_{R} are nonsingular, J_0 contains the Jordan blocks of eigenvalues inside the unit circle, J_∞ contains the Jordan blocks of eigenvalues outside the unit circle, and N is block diagonal with identity blocks corresponding to blocks of finite eigenvalues in J_∞, and nilpotent blocks corresponding to infinite eigenvalues (identity blocks in J_∞) [28]. In this notation, the projector first mentioned in Sect. 2.2 is

$$P_{\mathrm{R},|z|>1} = P_{\mathrm{R}} \begin{pmatrix} 0 & \\ & I \end{pmatrix} P_{\mathrm{R}}^{-1}$$

and the deflating subspace in question is spanned by the trailing columns of P_{R}.

Since J_∞ is nonsingular, we may write

$$\begin{aligned} A - \lambda B &= P_{\mathrm{L}}' \begin{pmatrix} I & \\ & J_\infty \end{pmatrix} \begin{pmatrix} J_0 - \lambda I & \\ & I - \lambda J_\infty^{-1} N \end{pmatrix} P_{\mathrm{R}}^{-1} \\ &\equiv P_{\mathrm{L}} \begin{pmatrix} J_0 - \lambda I & \\ & I - \lambda J_0' \end{pmatrix} P_{\mathrm{R}}^{-1} \ , \end{aligned} \tag{4.9}$$

where $J_0' = J_\infty^{-1} N$ has all its eigenvalues either nonzero and inside the unit circle (corresponding to finite eigenvalues of J_∞) or at zero (corresponding to nilpotent blocks of N). Thus

$$A^{-1}B = \left(P_{\mathrm{L}} \begin{pmatrix} J_0 & \\ & I \end{pmatrix} P_{\mathrm{R}}^{-1} \right)^{-1} \left(P_{\mathrm{L}} \begin{pmatrix} I & \\ & J_0' \end{pmatrix} P_{\mathrm{R}}^{-1} \right) = P_{\mathrm{R}} \begin{pmatrix} J_0^{-1} & \\ & J_0' \end{pmatrix} P_{\mathrm{R}}^{-1}$$

and

$$
\begin{aligned}
(A_p + B_p)^{-1}A_p &= (I + (A^{-1}B)^{2^p})^{-1} \\
&= P_{\mathrm{R}} \begin{pmatrix} (I + J_0^{-2^p})^{-1} & \\ & (I + J'^{2^p}_0)^{-1} \end{pmatrix} P_{\mathrm{R}}^{-1} \; . \qquad (4.10)
\end{aligned}
$$

Since $J_0^{-2^p} \to \infty$ and $J'^{2^p}_0 \to 0$ as $p \to \infty$, the last displayed expression converges to $P_{\mathrm{R},|z|>1}$ as desired. The approximate projector $(A_p + B_p)^{-1}B_p$ onto the other right deflating subspace is just

$$
\begin{aligned}
I - (A_p + B_p)^{-1}A_p &= (I + (A^{-1}B)^{2^p})^{-1}(A^{-1}B)^{2^p} \\
&= P_{\mathrm{R}} \begin{pmatrix} (I + J_0^{2^p})^{-1} & \\ & (I + J'^{-2^p}_0)^{-1} \end{pmatrix} P_{\mathrm{R}}^{-1} \qquad (4.11)
\end{aligned}
$$

which converges to

$$
P_{\mathrm{R},|z|<1} = I - P_{\mathrm{R},|z|>1} = P_{\mathrm{R}} \begin{pmatrix} I & \\ & 0 \end{pmatrix} P_{\mathrm{R}}^{-1} \; . \qquad (4.12)
$$

The projectors

$$
P_{\mathrm{L},|z|>1} = P_{\mathrm{L}} \begin{pmatrix} 0 & \\ & I \end{pmatrix} P_{\mathrm{L}}^{-1} \text{ and } P_{\mathrm{L},|z|<1} = I - P_{\mathrm{L},|z|>1} = P_{\mathrm{L}} \begin{pmatrix} I & \\ & 0 \end{pmatrix} P_{\mathrm{L}}^{-1}
$$

onto left deflating subspaces are computed in Algorithm 1 by applying the same procedure to $A^{\mathrm{H}} - \lambda B^{\mathrm{H}}$, since taking the conjugate transpose swaps right and left spaces.

We discuss the convergence rate of this iteration in the next section, after we have introduced the condition number.

Here is an alternative approach to computing the left deflating space, which saves about half cost of Algorithm 1, but requires the solution of a possibly ill-conditioned linear system. Note that

$$
\begin{aligned}
P_{\mathrm{L},|z|>1} \cdot (A, B) &= (P_{\mathrm{L}} \begin{pmatrix} 0 & \\ & I \end{pmatrix} P_{\mathrm{R}}^{-1}, P_{\mathrm{L}} \begin{pmatrix} 0 & \\ & J'_0 \end{pmatrix} P_{\mathrm{R}}^{-1}) \\
&= (A, B) \begin{pmatrix} P_{\mathrm{R},|z|>1} & \\ & P_{\mathrm{R},|z|>1} \end{pmatrix} \; .
\end{aligned}
$$

We can solve this for $P_{\mathrm{L},|z|>1}$ by using the decomposition

$$
\begin{pmatrix} A^{\mathrm{H}} \\ B^{\mathrm{H}} \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}
$$

so

$$
P_{\mathrm{L},|z|>1}[R^{\mathrm{H}}, 0] = (AP_{\mathrm{R},|z|>1}, BP_{\mathrm{R},|z|>1})Q
$$

and thus

$$
P_{\mathrm{L},|z|>1} = (AP_{\mathrm{R},|z|>1}, BP_{\mathrm{R},|z|>1})Q \begin{pmatrix} I \\ 0 \end{pmatrix} R^{-\mathrm{H}} \; .
$$

The condition number of R is the same as the condition number of the $n \times 2n$ matrix $(A,\ B)$. If $(A,\ B)$ is nearly singular, this means the pencil $A - \lambda B$ is nearly singular, which means its eigenvalues are all very ill-conditioned, among other things [24]. We discuss this further below.

5. Perturbation theory

Algorithm 1 will work (in exact arithmetic) unless there is an eigenvalue on the unit circle. This includes the case of singular pencils, since if $A-\lambda B$ is a singular pencil then $A-zB$ will be singular for any z, including the unit circle. Thus the set of matrices with an eigenvalue on the unit circle, or pencils such that $A-zB$ is singular for some z on the unit circle, are the sets of "ill-posed problems" for Algorithm 1.

Our goal is to show that the reciprocal of the distance to this set of ill-posed problems is a natural condition number for this problem. This will rely on a clever expression for the orthogonal projectors by Malyshev [41]. In contrast to Malyshev's work, however, our analysis will be much simpler and lead to a potentially much smaller condition number.

We begin with a simple formula for the distance to the nearest ill-posed problem. We define this distance as follows:

$$d_{(A,B)} \equiv \inf\{\|E\|+\|F\| : (A+E)-z(B+F) \text{ is singular for some } z \text{ where } |z| = 1\} \quad . \tag{5.13}$$

This infimum is clearly attained for some E and F by compactness. Note also that $d_{(A,B)} = d_{(B,A)} = d_{(A^{\mathrm{H}},B^{\mathrm{H}})} = d_{(B^{\mathrm{H}},A^{\mathrm{H}})}$.

Lemma 1. $d_{(A,B)} = \min_\theta \sigma_{\min}(A - \mathrm{e}^{\mathrm{i}\theta}B)$.

Proof. Let $\sigma = \min_\theta \sigma_{\min}(A-\mathrm{e}^{\mathrm{i}\theta}B)$. Then there is a θ and an E such that $\|E\| = \sigma$ and $A + E - \mathrm{e}^{\mathrm{i}\theta}B$ is singular, implying $d_{(A,B)} \le \|E\| = \sigma$. To prove the opposite inequality, the definition of $d_{(A,B)}$ implies that there are a θ and matrices E and F with $\|E\| + \|F\| = d_{(A,B)}$ such that $A+E-\mathrm{e}^{\mathrm{i}\theta}(B+F) = (A-\mathrm{e}^{\mathrm{i}\theta}B)+(E-\mathrm{e}^{\mathrm{i}\theta}F)$ is singular. Thus

$$d_{(A,B)} = \|E\| + \|F\| \ge \|E - \mathrm{e}^{\mathrm{i}\theta}F\| \ge \sigma_{\min}(A - \mathrm{e}^{\mathrm{i}\theta}B) \ge \sigma$$

as desired. □

As a remark, note that essentially the same proof shows that for *any* domain $\mathscr{D}$

$$\begin{aligned} &\min\{\|E,F\|_{\mathrm{F}} : \det((A+E)-z(B+F)) = 0 \text{ for some } z \in \mathscr{D}\} \\ &\quad = \min_{\substack{s,c \\ z=s/c\in\mathscr{D} \\ |s|^2+|c|^2=1}} \sigma_{\min}(cA-sB) \ , \end{aligned}$$

which is the natural way to extend the notion of pseudospectrum to matrix pencils [52]. An analogous formula appears in [19].

Now we turn to the perturbation theory of the approximate projector computed in step 3) of Algorithm 1, $(A_p + B_p)^{-1}B_p$, which is also given by the formula in (4.11). Following Malyshev [41], we will express this approximate projector as one block component of the solution of a particular linear system (our linear

system differs slightly from his). Let $m = 2^p$. All the subblocks in the following mn-by-mn linear system are n-by-n. All subblocks not shown in the coefficient matrix are zero.

$$M_m(A,B)\begin{pmatrix} Z_{m-1} \\ \vdots \\ Z_0 \end{pmatrix} \equiv \begin{pmatrix} -A & & & -B \\ B & \ddots & & \\ & \ddots & \ddots & \\ & & B & -A \end{pmatrix}\begin{pmatrix} Z_{m-1} \\ \vdots \\ Z_0 \end{pmatrix}$$

$$= \begin{pmatrix} -B \\ 0 \\ \vdots \\ 0 \end{pmatrix} \equiv \tilde{B}_m \tag{5.14}$$

If B or A were nonsingular, we could easily confirm that the solution of (5.14) would be

$$\begin{pmatrix} Z_{m-1} \\ Z_{m-2} \\ \vdots \\ Z_0 \end{pmatrix} = \begin{pmatrix} (B^{-1}A)^{m-1}(I+(B^{-1}A)^m)^{-1} \\ (B^{-1}A)^{m-2}(I+(B^{-1}A)^m)^{-1} \\ \vdots \\ (I+(B^{-1}A)^m)^{-1} \end{pmatrix}$$

$$\text{or} \quad \begin{pmatrix} (A^{-1}B)(I+(A^{-1}B)^m)^{-1} \\ (A^{-1}B)^2(I+(A^{-1}B)^m)^{-1} \\ \vdots \\ (A^{-1}B)^m(I+(A^{-1}B)^m)^{-1} \end{pmatrix} .$$

Thus we see that $Z_0 = (A^{-1}B)^m(I+(A^{-1}B)^m)^{-1}$ as in (4.11). Since this algebraic formula, $(A_p+B_p)^{-1}B_p = Z_0$, is true on an open dense set of matrix pairs (A,B), it is reasonable to suspect that it is true everywhere. We can prove this by using the Weierstrass Canonical Form of $A-\lambda B$ as in Sect. 4, assuming only that $A-\lambda B$ is nonsingular for all $|\lambda| = 1$ (see [10] for details).

The motivation for (5.14) in [41] is from a recurrence for the coefficients of the Fourier expansion of $(B - e^{i\theta}A)^{-1}$, but that will not concern us here.

By using standard perturbation theory for the linear system (5.14), we will get the perturbation theory for $(A_p+B_p)^{-1}B_p$ (or $(A_p+B_p)^{-1}A_p = I-(A_p+B_p)^{-1}B_p$) that we want. We will use a slight variation on the usual normwise perturbation theory, and take full account of the structure of the coefficient matrix. In fact, we will see that we get the same condition number whether or not we take the structure into account or not. Let I_m be an m-by-m identity matrix, and J_m be an m-by-m matrix with 1 on the subdiagonal, and -1 in position $(1,m)$. Then one can easily confirm that the coefficient matrix in (5.14) can be written using the Kronecker product $\otimes$ as

$$M_m(A,B) = -I_m \otimes A + J_m \otimes B \quad .$$

Since J_m is orthogonal, and hence normal, its eigendecomposition can be written $J_m = U\Lambda U^{\mathrm{H}}$, where U is a unitary matrix and $\Lambda = \mathrm{diag}(e^{i\theta_1}, ..., e^{i\theta_m})$ is the diagonal matrix of eigenvalues, all of which must lie on the unit circle. In fact, one can easily confirm that the characteristic polynomial of J_m is $\det(\lambda I - J_m) = \lambda^m + 1$, so the eigenvalues are m-th roots of -1. Then transforming $M_m(A, B)$ using the unitary similarity $U \otimes I_n$, we get

$$\begin{aligned}(U \otimes I_n)^{\mathrm{H}} M_m(A,B)(U \otimes I_n) &= -U^{\mathrm{H}} I_m U \otimes A + U J_m U^{\mathrm{H}} \otimes B \\ &= -I_m \otimes A + \Lambda \otimes B \\ &= \mathrm{diag}(-A + e^{i\theta_1} B, ..., -A + e^{i\theta_m} B) \ .\end{aligned}$$

Therefore, the smallest singular value of $M_m(A, B)$ is $\min_{1 \le j \le m} \sigma_{\min}(-A + e^{i\theta_j} B)$. As m grows, and the process converges, this smallest singular value decreases to $\min_\theta \sigma_{\min}(-A + e^{i\theta} B) = d_{(A,B)}$. This shows that $d_{(A,B)}^{-1}$ is a condition number for $(A_p + B_p)^{-1} B_p$, and in fact a lower bound bound for all finite m. We may also bound

$$\left\| \begin{pmatrix} Z_{m-1} \\ \vdots \\ Z_0 \end{pmatrix} \right\|_2 \le \frac{\|B\|}{d_{(A,B)}} \ . \tag{5.15}$$

6. Convergence analysis

Using equation (4.10), we will bound the error

$$\|(A_p + B_p)^{-1} A_p - P_{\mathrm{R},|z|>1}\| = \|(I + (A^{-1}B)^{2^p})^{-1} - P_{\mathrm{R},|z|>1}\|$$

after p steps of the algorithm. Our bound will be in terms of $\|P_{\mathrm{R},|z|>1}\|$ and $d_{(A,B)}$. It can be much tighter than the corresponding bound in Theorem 1.4 of [41], for reasons discussed in Sect. 8.

Theorem 1. *Let $d_{(A,B)}$ be defined as in (5.13). Then if*

$$p \ge \log_2 \frac{\|(A,B)\| - d_{(A,B)}}{d_{(A,B)}}$$

we may bound

$$\frac{\|(A_p + B_p)^{-1} A_p - P_{\mathrm{R},|z|>1}\|}{\|P_{\mathrm{R},|z|>1}\|} \le \frac{2^{p+3}(1 - \frac{d_{(A,B)}}{\|(A,B)\|})^{2^p}}{\max(0, 1 - 2^{p+2}(1 - \frac{d_{(A,B)}}{\|(A,B)\|})^{2^p})} \ . \tag{6.16}$$

Thus, we see that convergence is quadratic, and depends on the smallest relative perturbation $\frac{d_{(A,B)}}{\|(A,B)\|}$ that makes $A - \lambda B$ have an eigenvalue on the unit circle; the smaller this perturbation, the slower the convergence.

We begin the proof with an estimate on the growth of matrix powers. Many related bounds are in the literature [52, 36]; ours differs slightly because it involves powers of the matrix $Y^{-1}X$.

Lemma 2. *Let $X - \lambda Y$ have all its eigenvalues inside the unit circle. Then*

$$\|(Y^{-1}X)^m\| \leq \begin{cases} e_m \cdot m \cdot \left(1 - \frac{d_{(X,Y)}}{\|Y\|}\right)^m & \text{if } m > \frac{\|Y\| - d_{(X,Y)}}{d_{(X,Y)}} \\ \frac{\|Y\|}{d_{(X,Y)}} & \text{if } m \leq \frac{\|Y\| - d_{(X,Y)}}{d_{(X,Y)}} \end{cases} .$$

where $e \leq e_m \equiv (1 + m^{-1})^{m+1} \leq 4$, and $\lim_{m\to\infty} e_m = e$. We may also bound $e_m \cdot m \leq e \cdot (m + 1)$.

Proof of Lemma 2. Let r satisfy $\rho(Y^{-1}X) < r \leq 1$, where $\rho(Y^{-1}X)$ is the spectral radius of $Y^{-1}X$. Then

$$\begin{aligned} \|(Y^{-1}X)^m\| &= \left\| \frac{1}{2\pi \mathrm{i}} \oint_{\text{circle of radius } r} z^m (zI - Y^{-1}X)^{-1} dz \right\| \\ &= \left\| \frac{1}{2\pi \mathrm{i}} \int_0^{2\pi} (r\mathrm{e}^{\mathrm{i}\theta})^m (r\mathrm{e}^{\mathrm{i}\theta} Y - X)^{-1} d(r\mathrm{e}^{\mathrm{i}\theta}) Y \right\| \\ &\leq \frac{r^{m+1}\|Y\|}{\min_\theta \sigma_{\min}(r\mathrm{e}^{\mathrm{i}\theta} Y - X)} = \frac{r^{m+1}\|Y\|}{\min_\theta \sigma_{\min}(\mathrm{e}^{\mathrm{i}\theta} Y - X + Y\mathrm{e}^{\mathrm{i}\theta}(r-1))} \\ &\leq \frac{r^{m+1}\|Y\|}{\min_\theta \sigma_{\min}(\mathrm{e}^{\mathrm{i}\theta} Y - X) - \|Y\|(1-r)} \\ &= \frac{r^{m+1}}{d_{(X,Y)}/\|Y\| - 1 + r} \equiv f(r) . \end{aligned}$$

We may easily show that if $m \geq [\|Y\| - d_{(X,Y)}]/d_{(X,Y)}$, then $f(r)$ has a minimum at $\rho(Y^{-1}X) < r = \frac{m+1}{m}(1 - d_{(X,Y)}/\|Y\|) \leq 1$, and the value of this minimum is

$$m \cdot (1 + m^{-1})^{m+1} \cdot (1 - d_{(X,Y)}/\|Y\|)^m \equiv m \cdot e_m \cdot (1 - d_{(X,Y)}/\|Y\|)^m .$$

If $m \leq [\|Y\| - d_{(X,Y)}]/d_{(X,Y)}$, then the upper bound is attained at $r = 1$. □

Completely analogously, one may prove the following lemma, which is a special case of a bound in [52].

Lemma 3. *Let X have all its eigenvalues inside the unit circle. Let $d_X \equiv \min_\theta \sigma_{\min}(\mathrm{e}^{\mathrm{i}\theta} I - X)$; d_X is the smallest perturbation of X that will make it have an eigenvalue on the unit circle. Then*

$$\|X^m\| \leq \begin{cases} e_m \cdot m \cdot (1 - d_X)^m & \text{if } m > \frac{1 - d_X}{d_X} \\ \frac{1}{d_X} & \text{if } m \leq \frac{1 - d_X}{d_X} \end{cases} .$$

where e_m is as defined in Lemma 2.

Proof of Theorem 1. By a unitary change of basis, we may without loss of generality assume that

$$A - \lambda B = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} - \lambda \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}$$

where the eigenvalues of $A_{11}-\lambda B_{11}$ are inside the unit circle, and the eigenvalues of $A_{22}-\lambda B_{22}$ are outside the unit circle. Let L and R be the unique matrices such that [24, 48]

$$\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} - \lambda \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} = \begin{pmatrix} I & L \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{11}-\lambda B_{11} & 0 \\ 0 & A_{22}-\lambda B_{22} \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix}^{-1} .$$

Then, assuming for the moment that A is invertible, we get

$$A^{-1}B = \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{11}^{-1}B_{11} & 0 \\ 0 & A_{22}^{-1}B_{22} \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix}^{-1}$$

and

$$P_{\mathrm{R},|z|>1} = \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix}^{-1} = \begin{pmatrix} 0 & R \\ 0 & I \end{pmatrix} .$$

Then we see that $E_p \equiv (I+(A^{-1}B)^{2^p})^{-1} - P_{\mathrm{R},|z|>1}$ may be written

$$\begin{aligned} E_p &= \begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \begin{pmatrix} (I+(A_{11}^{-1}B_{11})^{2^p})^{-1} & 0 \\ 0 & (I+(A_{22}^{-1}B_{22})^{2^p})^{-1} - I \end{pmatrix} \begin{pmatrix} I & R \\ 0 & I \end{pmatrix}^{-1} \\ &= \begin{pmatrix} (I+(B_{11}^{-1}A_{11})^{2^p})^{-1}(B_{11}^{-1}A_{11})^{2^p} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & -R \\ 0 & 0 \end{pmatrix} \\ &\quad - \begin{pmatrix} 0 & R \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & (I+(A_{22}^{-1}B_{22})^{2^p})^{-1}(A_{22}^{-1}B_{22})^{2^p} \end{pmatrix} . \end{aligned}$$

The derivation of this formula used the fact that A, and so A_{11}, were nonsingular, but the final formula does not require this. Thus

$$\begin{aligned} \|E_p\| &\le \|P_{\mathrm{R},|z|>1}\| (\|(I+(B_{11}^{-1}A_{11})^{2^p})^{-1}(B_{11}^{-1}A_{11})^{2^p}\| \\ &\quad + \|(I+(A_{22}^{-1}B_{22})^{2^p})^{-1}(A_{22}^{-1}B_{22})^{2^p}\|) \\ &\le \|P_{\mathrm{R},|z|>1}\| \left(\frac{\|(B_{11}^{-1}A_{11})^{2^p}\|}{1-\|(B_{11}^{-1}A_{11})^{2^p}\|} + \frac{\|(A_{22}^{-1}B_{22})^{2^p}\|}{1-\|(A_{22}^{-1}B_{22})^{2^p}\|} \right) \end{aligned}$$

provided the denominators are positive. From Lemma 2, we may bound

$$\|(B_{11}^{-1}A_{11})^{2^p}\| \le 4\cdot 2^p \cdot \left(1 - \frac{d_{(A_{11},B_{11})}}{\|B_{11}\|}\right)^{2^p}$$

$$\text{and} \quad \|(A_{22}^{-1}B_{22})^{2^p}\| \le 4\cdot 2^p \cdot \left(1 - \frac{d_{(A_{22},B_{22})}}{\|A_{22}\|}\right)^{2^p}$$

for p sufficiently large. Since

$$\frac{d_{(A_{11},B_{11})}}{\|B_{11}\|} \ge \frac{d_{(A,B)}}{\|(A,B)\|} \quad \text{and} \quad \frac{d_{(A_{22},B_{22})}}{\|A_{22}\|} \ge \frac{d_{(A,B)}}{\|(A,B)\|}$$

this yields the desired bound. □

A weakness in Lemmas 2 and 3 comes from using the single number $d_{(A,B)}$ (or d_A) to characterize a matrix. For example,

$$A_1 = \begin{pmatrix} .5 & 1000 & 0 \\ 0 & .5 & 1000 \\ 0 & 0 & .5 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix} ,$$

where $\alpha \approx 1 - 1.25 \cdot 10^{-7}$ have the same value of d_A, namely about $1.25 \cdot 10^{-7}$. $\|A_2^n\|$ clearly never increases, let alone to $1/d_A \approx 8 \cdot 10^6$ as predicted by Lemma 3; in contrast $\|A_1^n\|$ gets as large as $1.5 \cdot 10^6$. For large n, $\|A_2^n\|$ decreases precisely as $(1-d_A)^n \approx .999999875^n$, as predicted by Lemma 3; in contrast $\|A_1^n\|$ decreases much faster, as $.5^n$. To see that both parts of the bound can be attained simultaneously, consider $\mathrm{diag}(A_1, A_2)$. Despite the potential overestimation, we will use $d_{(A,B)}$ in all our analyses in the paper, both because it gives tighter bounds than those previously published, and in the inevitable tradeoff between accuracy and simplicity of bounds of this sort, we have chosen simplicity.

One can use the bound in Lemma 3 to bound the norm of A^n computed in floating point [36]; this work will appear elsewhere.

7. Error analysis

Following Malyshev [41], the analysis depends on the observation that step 2) of Algorithm 1 is just computing the QR decomposition of $M_m(A, B)$, in a manner analogous to block cyclic reduction [17]. Malyshev works hard to derive a rigorous *a priori* bound on the total roundoff error, yielding an expression which is complicated and possibly much too large. It can be too large because it depends on his condition number ω (see Sect. 8) instead of our smaller $d_{(A,B)}^{-1}$, and because worst case roundoff analysis is often pessimistic. In algorithmic practice, we will use an *a posteriori* bound $\max(\|E_{21}\|, \|F_{21}\|)$, which will be a precise measure of the backward error in one spectral decomposition, rather than the *a priori* bounds presented here.

We begin by illustrating why step 2 of Algorithm 1 is equivalent to solving (5.14) using QR decomposition. We take $p = 3$, which means $m = 2^3 = 8$. Let

$$\begin{pmatrix} Q_{11}^{(j)} & Q_{12}^{(j)} \\ Q_{21}^{(j)} & Q_{22}^{(j)} \end{pmatrix}$$

be the orthogonal matrix computed in the jth iteration of step 2), and let

$$\tilde{Q}^{(j)} = \begin{pmatrix} Q_{21}^{(j)} & Q_{22}^{(j)} \\ Q_{11}^{(j)} & Q_{12}^{(j)} \end{pmatrix} .$$

Then we see that step 2) of algorithm 2 is equivalent to the identity

$$\tilde{Q}^{(j)\mathrm{H}} \begin{pmatrix} -A_j & 0 & B_j \\ B_j & A_j & 0 \end{pmatrix} = \begin{pmatrix} R_j & \star & \star \\ 0 & A_{j+1} & B_{j+1} \end{pmatrix} \tag{7.17}$$

where the $\star$s are entries which do not interest us. Multiplying block rows 1 and 2, 3 and 4, 5 and 6, and 7 and 8 in (5.14) by $\tilde{Q}^{(0)\mathrm{H}}$ and using (7.17) yields

$$\begin{pmatrix} R_1 & \star & & & & & & \star \\ 0 & -A_1 & & & & & & -B_1 \\ & \star & R_1 & \star & & & & \\ & B_1 & 0 & -A_1 & & & & \\ & & & \star & R_1 & \star & & \\ & & & B_1 & 0 & -A_1 & & \\ & & & & & \star & R_1 & \star \\ & & & & & B_1 & 0 & -A_1 \end{pmatrix} \begin{pmatrix} Z_7 \\ Z_6 \\ Z_5 \\ Z_4 \\ Z_3 \\ Z_2 \\ Z_1 \\ Z_0 \end{pmatrix} = \begin{pmatrix} \star \\ -B_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Reordering the odd-numbered blocks before the even ones results in

$$\left(\begin{array}{cccc|cccc} R_1 & & & & \star & & & \star \\ & R_1 & & & \star & \star & & \\ & & R_1 & & & \star & \star & \\ & & & R_1 & & & \star & \star \\ \hline & & & & -A_1 & & & -B_1 \\ & & & & B_1 & -A_1 & & \\ & & & & & B_1 & -A_1 & \\ & & & & & & B_1 & -A_1 \end{array}\right) \cdot \left(\begin{array}{c} Z_7 \\ Z_5 \\ Z_3 \\ Z_1 \\ \hline Z_6 \\ Z_4 \\ Z_2 \\ Z_0 \end{array}\right) = \left(\begin{array}{c} \star \\ 0 \\ 0 \\ 0 \\ \hline -B_1 \\ 0 \\ 0 \\ 0 \end{array}\right). \tag{7.18}$$

Repeating this with $\tilde{Q}^{(1)\mathrm{H}}$ on the lower right corner of (7.18), and similarly reordering blocks, we get

$$\left(\begin{array}{cccc|cc|cc} R_1 & & & & \star & & & \star \\ & R_1 & & & \star & \star & & \\ & & R_1 & & & \star & \star & \\ & & & R_1 & & & \star & \star \\ \hline & & & & R_2 & & \star & \star \\ & & & & & R_2 & \star & \star \\ \hline & & & & & & -A_2 & -B_2 \\ & & & & & & B_2 & -A_2 \end{array}\right) \left(\begin{array}{c} Z_7 \\ Z_5 \\ Z_3 \\ Z_1 \\ \hline Z_6 \\ Z_2 \\ \hline Z_4 \\ Z_0 \end{array}\right) = \left(\begin{array}{c} \star \\ 0 \\ 0 \\ 0 \\ \hline \star \\ 0 \\ \hline -B_2 \\ 0 \end{array}\right). \tag{7.19}$$

One more step with $\tilde{Q}^{(2)\mathrm{H}}$ on the lower right corner of (7.19) yields

$$\left(\begin{array}{cccc|cc|cc} R_1 & & & & \star & & & \star \\ & R_1 & & & \star & \star & & \\ & & R_1 & & & \star & \star & \\ & & & R_1 & & & \star & \star \\ \hline & & & & R_2 & & \star & \star \\ & & & & & R_2 & \star & \star \\ \hline & & & & & & R_3 & \star \\ & & & & & & & -A_3 - B_3 \end{array}\right) \left(\begin{array}{c} Z_7 \\ Z_5 \\ Z_3 \\ Z_1 \\ \hline Z_6 \\ Z_2 \\ \hline Z_4 \\ Z_0 \end{array}\right) = \left(\begin{array}{c} \star \\ 0 \\ 0 \\ 0 \\ \hline \star \\ 0 \\ \hline \star \\ -B_3 \end{array}\right). \tag{7.20}$$

Thus, we see again that $Z_0 = (A_3 + B_3)^{-1}B_3$ as desired. It is clear from this development that the process is backward stable in the following sense: the computed $A_3 + B_3$ (or more generally $A_m + B_m$) in the transformed coefficient matrix, and B_3 (or B_m) in the transformed right hand side, are the exact results corresponding to a slightly perturbed $M_{2^m}(A,B)+\delta M_{2^m}$ and initial right hand side $\tilde{B}_{2^m} + \delta\tilde{B}_{2^m}$, where $\|\delta M_{2^m}\| = O(\varepsilon)\|(A,B)\|$ and $\|\delta\tilde{B}_{2^m}\| = O(\varepsilon)\|B\|$.

Next we must analyze the computation of Q_{R} in step 3) of Algorithm 1. As described in Sect. 2.2, if we use the GQR decomposition to compute Q_{R} without inverses, then Q_{R} is nearly the exact orthogonal factor of $(A_m+B_m+E)^{-1}(B_m+F)$ where $\|E\| = O(\varepsilon)\|A_m + B_m\| = O(\varepsilon)\|(A,B)\|$, and $\|F\| = O(\varepsilon)\|B_m\| = O(\varepsilon)\|B\|$. We can take these E and F and "push them back" into δM and $\delta\tilde{B}_m$, respectively, since the mapping from $M_{2^m}(A,B)+\delta M_{2^m}$ to A_m+B_m is an orthogonal projection, as is the map from $\tilde{B}_{2^m}$ to B_m. So altogether, combining the analysis of steps 1) and 2), we can say that Q_{R} is nearly the exact answer for $M_{2^m}(A,B) + \delta M'_{2^m}$ and $\tilde{B}_{2^m} + \delta\tilde{B}'_{2^m}$ where $\|\delta M'_{2^m}\| = O(\varepsilon)\|(A,B)\|$ and $\|\delta\tilde{B}'_{2^m}\| = O(\varepsilon)\|B\|$. Since the condition number of the linear system (5.14) is (no larger than) $d^{-1}_{(A,B)}$, and the norm of the solution is bounded by (5.15), the absolute error in the computed Z_0 of which Q_{R} is nearly the true factor is bounded by[2] $O(\varepsilon)\cdot\|B\|\cdot\|(A,B)\|d^{-2}_{(A,B)} \le O(\varepsilon)\cdot\|(A,B)\|^2 d^{-2}_{(A,B)}$.

To bound the error in the space spanned by the leading columns of Q_{R}, which is our approximate deflating subspace, we need to know how much a right singular subspace of a matrix Z_0, i.e. the space spanned by the right singular vectors corresponding to a subset $\mathscr{S}$ of the singular values, is perturbed when Z_0 is perturbed by a matrix of norm η. If Z_0 were the exact projector in (4.12), $\mathscr{S}$ would consist of all the nonzero singular values. In practice, of course, this is a question of rank determination. No matter what $\mathscr{S}$ is, the space spanned by the corresponding singular vectors is perturbed by at most $O(\eta)/\mathrm{gap}_{\mathscr{S}}$ [44, 51, 48], where $\mathrm{gap}_{\mathscr{S}}$ is the shortest distance from any singular value in $\mathscr{S}$ to any singular value not in $\mathscr{S}$:

$$\mathrm{gap}_{\mathscr{S}} \equiv \min_{\substack{\sigma\in\mathscr{S}\\ \bar{\sigma}\notin\mathscr{S}}} |\sigma-\bar{\sigma}| \ .$$

so we need to estimate $\mathrm{gap}_{\mathscr{S}}$ in order to compute an error bound. We will do this for Z_0 equal to its limit $P_{\mathrm{R},|z|<1}$ in (4.12). There is always a unitary change of basis in which a projector is of the form $\begin{pmatrix} I & \Sigma \\ 0 & 0 \end{pmatrix}$, where $\Sigma = \mathrm{diag}(\sigma_1,\ldots,\sigma_{l_{\mathrm{R}}})$ is diagonal with $\sigma_1 \ge \cdots \ge \sigma_{l_{\mathrm{R}}} \ge 0$. From this it is easy to compute the singular values of the projector: $\{\sqrt{1+\sigma_1^2},\ldots,\sqrt{1+\sigma_{l_{\mathrm{R}}}^2},1,\cdots,1,0,\ldots,0\}$, where the number of ones in the set of singular values is equal to $\max(2l_{\mathrm{R}}-n,0)$. Since $\mathscr{S} = \{\sqrt{1+\sigma_1^2},\ldots,\sqrt{1+\sigma_{l_{\mathrm{R}}}^2},1,\cdots,1\}$, we get

[2] This bound is true even if we compute the inverse of $A_m + B_m$ explicitly

$$\mathrm{gap}_{\mathscr{S}} = \begin{cases} \sqrt{1+\sigma_{l_R}^2} & \text{if } 2l_R \le n \\ 1 & \text{if } 2l_R > n \end{cases} .$$

Thus, we get that in the limit as $m \to \infty$, the error δQ_R in Q_R is bounded by

$$\|\delta Q_R\| = \frac{O(\varepsilon)\cdot\|(A,B)\|^2}{d_{(A,B)}^2 \cdot \mathrm{gap}_{\mathscr{S}}} . \tag{7.21}$$

A similar bound holds for $\|\delta Q_L\|$ in Algorithm 1. Thus

$$\begin{aligned} \|E_{21}\| &\le \|(Q_L+\delta Q_L)^H A(Q_R+\delta Q_R) - Q_L^H A Q_R\| \\ &= \|\delta Q_L^H A Q_R + Q_L^H A \delta Q_R\| + O(\varepsilon^2) \\ &\le (\|\delta Q_L\| + \|\delta Q_R\|)\|A\| + O(\varepsilon^2) \end{aligned}$$

with a similar bound for $\|F_{21}\|$.

So altogether, in the limit as $m \to \infty$, we expect the following bound on backward stability[3]:

$$\begin{aligned} \max\left(\frac{\|E_{21}\|}{\|A\|}, \frac{\|F_{21}\|}{\|B\|}\right) &\le \frac{O(\varepsilon)\cdot\|(A,B)\|^2}{d_{(A,B)}^2 \cdot \min(\mathrm{gap}_{\mathscr{S}_R}, \mathrm{gap}_{\mathscr{S}_L})} \\ &\le \frac{O(\varepsilon)\cdot\|(A,B)\|^2}{d_{(A,B)}^2} , \end{aligned} \tag{7.22}$$

where $\mathrm{gap}_{\mathscr{S}_R}$ refers to the gap in the singular values of $P_{R,|z|<1}$, and $\mathrm{gap}_{\mathscr{S}_L}$ refers to the gap in the singular values of $P_{L,|z|<1}$,

For simplicity, consider Algorithm 1 when $B = I$, where $P_{R,|z|<1} = P_{L,|z|<1}$. An interesting feature of the error bound is that it may be smaller if $2l_R \le n$ than otherwise. This is borne out by numerical experiments, where it can be more accurate to make the choice in step 3) of Algorithm 1 which leads to A_{11} being smaller than A_{22}. Also, when $2l_R \le n$, the error bound is a decreasing function of σ_{l_R}. On the other hand, If σ_{l_R} is large, this means σ_1 and so $\|P_{R,|z|<1}\| = \sqrt{1+\sigma_1^2}$ are large, and this in turn means the eigenvalues inside the unit circle are ill-conditioned [24]. This should mean the eigenvalues are *harder* to divide, not easier. Of course as they become more ill-conditioned, $d_{(A,B)}$ decreases at the same time, which counterbalances the increase in σ_{l_R}.

In practice, we will use the *a posteriori* bounds $\|E_{21}\|$ and $\|F_{21}\|$ anyway, since if we block upper-triangularize $Q_L^H(A-\lambda B)Q_R$ by setting the $(2,1)$ blocks to zero, $\|E_{21}\|$ and $\|F_{21}\|$ are precisely the backward errors we commit. If the next section, we will compare our error bound with those in [41].

[3] In fact this bound holds for sufficiently large m as well

8. Remark on Malyshev's condition number

We have just shown that $d_{(A,B)}^{-1}$ is a natural condition number for this problem. In this subsection, we will show that Malyshev's condition number can be much larger [41]. Malyshev's condition number is

$$\begin{aligned} \omega &\equiv \left\| \frac{1}{2\pi} \int_0^{2\pi} (B - e^{i\phi}A)^{-1}(AA^H + BB^H)(B - e^{i\phi}A)^{-H} d\phi \right\| \\ &= \left\| \frac{1}{2\pi} \int_0^{2\pi} (B' - e^{i\phi}A')^{-1}(B' - e^{i\phi}A')^{-H} d\phi \right\| \end{aligned} \tag{8.23}$$

where $A' = (AA^H + BB^H)^{-1/2}A$ and $B' = (AA^H + BB^H)^{-1/2}B$; this means $A'A'^H + B'B'^H = I$. Malyshev begins his version of the algorithm by replacing A by A' and B by B', which we could too if we wanted to.

Malyshev's absolute error bound on the computed Z_0 is essentially $O(\varepsilon)\omega^2$, whereas ours is $O(\varepsilon)d_{(A,B)}^{-2}$, assuming $\|(A, B)\| \approx 1$. We will show that $d_{(A,B)}^{-1}$ can be as small as the square root of ω.

Since

$$\sigma_{\min}(AA^H + BB^H) \leq \frac{d_{(A,B)}}{d_{(A',B')}} \leq \sigma_{\max}(AA^H + BB^H)$$

it is sufficient to compare ω and $d_{(A,B)}^{-1}$ when $AA^H + BB^H$ is well-conditioned. Malyshev shows that, in our notation, $d_{(A',B')}^{-1} < 5\pi\omega$, showing that $d_{(A',B')}^{-1}$ is never much larger than ω. Malyshev shows that $d_{(A,B)}^{-1}$ and ω can be close when $B = I$ and A is real symmetric. By taking norms inside the integral in (8.23), one gets the other bound $\sqrt{\omega} \leq d_{(A,B)}^{-1}$, showing that $d_{(A,B)}^{-1}$ can be as small as the square root of ω. To see that $d_{(A,B)}^{-1}$ can indeed be this small, consider the following example. Let $A = I$ and $B = D - N$, where D is diagonal with entries equally spaced along any arc of the circle centered at the origin with radius $0 < d < 1$ and angular extent $\pi/8$, and N has ones on the superdiagonal and zeros elsewhere. When d is close to 1 and the dimension of A is at least about 20, one can computationally confirm that $d_{(A,B)}^{-1}$ is close to $\sqrt{\omega}$. This example works because when $e^{i\theta}$ is in the same sector as the eigenvalues of B, $(B - e^{i\theta}A)^{-1}$ is as large as it can get, and its largest entry is in position $(1, n)$:

$$\frac{1}{\prod_{k=1}^{n}(B_{kk} - e^{i\theta})}$$

Thus the integral for ω is bounded above by a modest multiple of the integral of the square of the magnitude of the quantity just displayed (times $\sigma_{\max}(AA^H + BB^H)$), which is near its maximum value $d_{(A,B)}^{-2}$ for a range of θ close to $[0, \pi/8]$, so the integral is within a constant of $d_{(A,B)}^{-2}$.

9. Stopping criterion

In this section we justify the stopping criterion used in Algorithm 1 by showing that R_j converges quadratically. From step 2) of Algorithm 1, we see that

$$B_{j+1} = Q_{22}^{\mathrm{H}} B_j = Q_{22}^{\mathrm{H}} Q_{11} R_j \quad \text{and} \quad A_{j+1} = Q_{12}^{\mathrm{H}} A_j = -Q_{12}^{\mathrm{H}} Q_{21} R_j \ .$$

For two symmetric non-negative definite matrices P_1 and P_2, we use the relation $P_1 \leq P_2$ to mean that $P_2 - P_1$ is non-negative definite. The above relations imply

$$\begin{aligned} R_{j+1}^{\mathrm{H}} R_{j+1} &= B_{j+1}^{\mathrm{H}} B_{j+1} + A_{j+1}^{\mathrm{H}} A_{j+1} \\ &= R_j^{\mathrm{H}} \left(Q_{11}^{\mathrm{H}} Q_{22} Q_{22}^{\mathrm{H}} Q_{11} + Q_{21}^{\mathrm{H}} Q_{12} Q_{12}^{\mathrm{H}} Q_{21} \right) R_j \\ &\leq R_j^{\mathrm{H}} \left(Q_{11}^{\mathrm{H}} Q_{11} + Q_{21}^{\mathrm{H}} Q_{21} \right) R_j \\ &= R_j^{\mathrm{H}} R_j \ . \end{aligned}$$

Since $R_j^{\mathrm{H}} R_j \geq 0$ for all j, the above relation implies that the sequence $\{R_j^{\mathrm{H}} R_j\}$ converges. On the other hand, since R_j can be viewed as a diagonal block in the upper triangular matrix of the cyclic QR decomposition of the coefficient matrix in (5.14), we have $\sigma_{\min}(R_j) \geq d_{(A,B)}$. Hence the sequence $\{R_j^{\mathrm{H}} R_j\}$ converges to a symmetric positive definite matrix. Let this limit matrix be $R^{\mathrm{H}} R$, where R is upper triangular with positive diagonal elements. It follows that the sequence $\{R_j\}$ converges to R.

Now we sketch a proof of quadratic convergence of $\{R_j\}$. For details see [10]. Note that

$$\begin{aligned} R_{j+1}^{\mathrm{H}} R_{j+1} &= R_j^{\mathrm{H}} \left(Q_{11}^{\mathrm{H}} Q_{22} Q_{22}^{\mathrm{H}} Q_{11} + Q_{21}^{\mathrm{H}} Q_{12} Q_{12}^{\mathrm{H}} Q_{21} \right) R_j \\ &= R_j^{\mathrm{H}} (I - S_j S_j^{\mathrm{H}} - S_j^{\mathrm{H}} S_j) R_j \end{aligned}$$

where $S_j = Q_{11}^{\mathrm{H}} Q_{21}$. It then follows that S_j converges to the zero matrix. If we define E_j by $R_{j+1} = (I + E_j) R_j$, then E_j is upper triangular and satisfies

$$(I + E_j)^{\mathrm{H}} (I + E_j) = I - S_j S_j^{\mathrm{H}} - S_j^{\mathrm{H}} S_j \ .$$

In other words, $(I + E_j)^{\mathrm{H}} (I + E_j)$ is the Cholesky factorization of $I - S_j S_j^{\mathrm{H}} - S_j^{\mathrm{H}} S_j$. Hence $\|E_j\| = O(\|S_j\|^2)$ and

$$\|R_{j+1} - R_j\| \leq \|E_j\| \ \|R_j\| = O(\|S_j\|^2 \|R_j\|) \ .$$

Finally, by next proving the identity $S_j = -R_j^{-\mathrm{H}} B_j^H A_j R_j^{-1}$, we can complete our set of recurrences for R_j, E_j and S_j with the formula

$$S_{j+1} = -(I + E_j)^{-\mathrm{H}} S_j^2 (I + E_j)^{-1} \ . \tag{9.24}$$

This establishes the quadratic convergence of $\{S_j\}$ to 0 and hence $\{R_j\}$ to R. We point out that this implies that the sequence $\{B_j^H A_j\}$ also converges quadratically to 0.

10. Numerical experiments

In this section, we present results of our numerical experiments with Algorithm 1 and compare them with the matrix sign function based algorithm. In all experiments we split the spectrum along the imaginary axis. When $B = I$, this means we apply Algorithm 1 to $A_0 = I - A$ and $B_0 = I + A$. For general B, this means that we apply Algorithm 1 to $A_0 = B - A$ and $B_0 = B + A$. We focus primarily on the ordinary SDC problem ($B = I$). The algorithm was implemented in MATLAB version 4.0a on a SUN workstation 1+ using IEEE standard double precision arithmetic with machine precision $\varepsilon \approx 2.2 \times 10^{-16}$.

The Newton iteration (3.6) for computing the matrix sign function of a matrix A is terminated if

$$\|A_{j+1} - A_j\|_1 \leq 10n\varepsilon \|A_j\|_1.$$

The inner loop iteration in Algorithm 1 for computing the desired projector is terminated if

$$\|R_j - R_{j-1}\|_1 \leq 10n\varepsilon \|R_{j-1}\|_1.$$

We set the maximal number of iterations *maxit*=60 for both the Newton iteration and the inverse-free iteration.

Algorithm 1 and the matrix sign function based algorithm work well for the numerous random matrices we tested. In a typical example for the standard SDC problem ($B = I$), we let A be a 100 by 100 random matrix with entries independent and normally distributed with mean 0 and variance 1; A has condition number about 10^4. Algorithm 1 took 13 inverse-free iterations to converge and returned with $\|E_{21}\|_1/\|A_{21}\|_1 \approx 5.44 \times 10^{-15}$. The matrix sign function took 12 Newton iterations to converge and returned with $\|E_{21}\|_1/\|A_{21}\|_1 \approx 2.12 \times 10^{-14}$. Both algorithms determined 48 eigenvalues in the open left half plane, all of which agreed with the eigenvalues computed by the QR algorithm to 12 decimal digits.

In a typical example for the generalized SDC problem (general B), we let A and B be 50 by 50 random matrices with entries distributed as above. Algorithm 1 took 10 inverse-free iterations to compute the right deflating subspace, and 10 inverse-free iterations for the left deflating subspace, and returned with $\|E_{21}\|_1/\|A_{21}\|_1 \approx 3.31 \times 10^{-15}$ and $\|F_{21}\|_1/\|B_{21}\|_1 \approx 2.64 \times 10^{-15}$. Using the QZ algorithm, we found that the closest distance of the eigenvalues of the pencil $A - \lambda B$ to the imaginary axis was about 10^{-3}.

We now present three examples, where test matrices are constructed so that they are ill-conditioned for inversion, have eigenvalues close to the imaginary axis, and/or have large norm of the spectral projector corresponding to the eigenvalues we want to split. Thus, they should be difficult cases for our algorithm.

In the following tables, we use rcond(A) to denote the estimate of the reciprocal condition number of matrix A computed by MATLAB function `rcond`. $\Delta(A) = \min_{\lambda_j \in \lambda(A)} |\Re\lambda_j|$ is the distance of the nearest eigenvalue to the imaginary axis. sep = sep$(A_{11}, A_{22}) = \sigma_{\min}(I \otimes A_{11} - A_{22}^{\mathrm{T}} \otimes I)$ is the separation of

matrices A_{11} and A_{22} [48], and $\|P\| = \sqrt{1+\|R\|^2}$ is the norm of the spectral projector $P = \begin{pmatrix} I & R \\ 0 & 0 \end{pmatrix}$ corresponding the eigenvalues of A_{11}; R satisfies $A_{11}R - RA_{22} = -A_{12}$. A number 10^{α} in parenthesis next to an iteration number *iter* in the following tables indicates that the convergence of the Newton iteration or the inverse-free iteration was stationary at about 10^{α} from the $iter^{\text{th}}$ iteration forward, and failed to satisfy the stopping criterion even after 60 iterations.

All random matrices used below have entries independent and normally distributed with mean 0 and variance 1.

Example 1. This example is taken from [5, 2]. Let

$$B = \begin{pmatrix} -\eta & 1 & 0 & 0 \\ -1 & -\eta & 0 & 0 \\ 0 & 0 & \eta & 1 \\ 0 & 0 & -1 & \eta \end{pmatrix}, \quad G = R = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\text{and} \quad A = Q^{\mathrm{T}} \begin{pmatrix} B & R \\ G & -B^{\mathrm{T}} \end{pmatrix} Q \ ,$$

where Q is an orthogonal matrix generated from the QR decomposition of a random matrix. As $\eta \to 0$, two pairs of complex conjugate eigenvalues of A approach the imaginary axis, one pair at about $-\eta^2 \pm \mathrm{i}$ and the other pair at about $\eta^2 \pm \mathrm{i}$.

Table 1 lists the results computed by Algorithm 1 and the matrix sign function based algorithm. From Table 1, we see that if a matrix is not ill-conditioned to invert, the Newton iteration performs as well as the inverse-free iteration. When there are eigenvalues close to the boundary of our selected region (the imaginary axis), the inverse-free iteration suffers the same slow convergence and the large backward error as the Newton iteration. These eigenvalues are simply too close to separate. Note that the Newton iteration takes about 6 to 7 times less work than the inverse-free iteration.

Table 1. Numerical results for Example 1

		Newton iteration		Inverse-free iteration	
$\Delta(A) \approx \eta^2$	rcond(A)	*iter*	$\frac{\|E_{21}\|_1}{\|A\|_1}$	*iter*	$\frac{\|E_{21}\|_1}{\|A\|_1}$
1	$6.83e-2$	7	$2.19e-16$	7	$3.14e-16$
10^{-2}	$3.18e-2$	14	$1.26e-15$	14	$1.75e-15$
10^{-6}	$3.12e-2$	27	$2.21e-11$	27	$1.94e-11$
10^{-10}	$4.28e-2$	41	$3.65e-07$	40	$1.56e-07$

For this example, we also compared the observed numerical convergence rate of Algorithm 1 with the theoretical prediction of the convergence rate given in Theorem 1. To compute the theoretical prediction, we need to estimate $d_{(A,B)}$. Algorithms for computing d_A and related problems are given in [15, 14, 19]. Since our examples are quite small, and we needed little accuracy, we used "intelligent brute force" to estimate $d_{(A,B)}$.

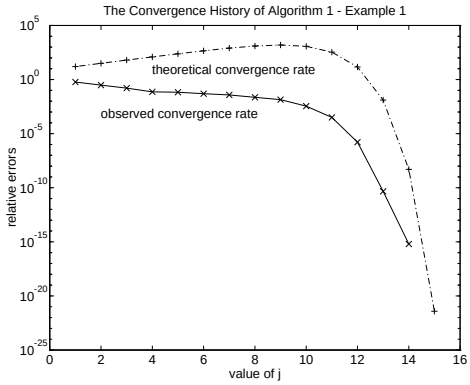

Fig. 1. Convergence History of Example 1, $\eta = 0.1$

Figure 1 plots the observed convergence rate of Algorithm 1 and the theoretical convergence rate, which is the upper bound in (6.16), for the matrix A with $\eta = 0.1$. We estimated $d_{(A_0,B_0)} \approx 9.72 \times 10^{-3}$, and $\|(A_0, B_0)\| \approx 6.16$. Although the theoretical convergence rate is an overestimate, it does reproduce the basic convergence behavior of the algorithm, in particular the ultimate quadratic convergence. Regarding the analysis of the backward accuracy as given in (7.22), for this example, we have

$$\frac{\|E_{21}\|}{\|A\|} \approx 7.87 \times 10^{-15} < \frac{\varepsilon\, \|(A_0, B_0)\|^2}{d^2_{(A_0,B_0)}} \approx 8.89 \times 10^{-11}.$$

As we have observed in many experiments, the bound in (7.22) is often pessimistic, and so the algorithm works much better than we can prove. More study is needed.

Example 2. In this example, A is a parameterized matrix of the form $A = Q^{\mathrm{T}}\tilde{A}Q$, where Q is an orthogonal matrix generated from the QR decomposition of a random matrix,

$$\tilde{A} = \begin{array}{c} \\ k \\ k \end{array} \begin{array}{c} \begin{array}{cc} k & k \end{array} \\ \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \end{array}, \quad A_{11} = \begin{pmatrix} 1-\alpha & & & \alpha \\ \alpha & 1-\alpha & & \\ & \ddots & \ddots & \\ & & \alpha & 1-\alpha \end{pmatrix},$$

$$A_{22} = -A_{11}^{\mathrm{T}}, \qquad 0 \le \alpha \le 0.5,$$

and A_{12} is a random matrix. Note that the eigenvalues of A_{11} lie on a circle with center $1 - \alpha$ and radius α and those of A_{22} lie on a circle with center $-1 + \alpha$ and radius α. The closest distance of the eigenvalues of A to the imaginary axis is $\Delta(A) = 1 - 2\alpha$. As $\alpha \to 0.5$, two eigenvalues of A simultaneously approach the imaginary axis from the right and left. Figure 2 is the eigenvalue distribution when $k = 20$ and $\alpha = .45$.

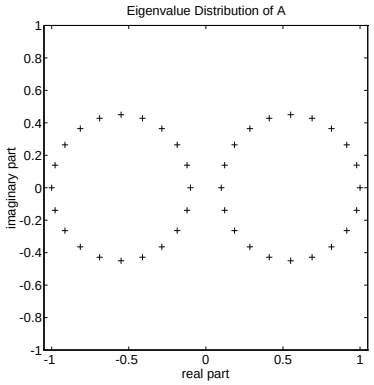

Fig. 2. Eigenvalue distribution of 40 by 40 matrix A with $k = 20$, $\alpha = 0.45$

Table 2 reports the computed results for different values of α with $k = 10$. From this data, we see that when the eigenvalues of A are adequately separated from the imaginary axis ($\Delta(A) \geq \sqrt{\varepsilon}$), the results computed by the inverse-free iteration are superior to the ones from Newton iteration, especially when the matrix is ill-conditioned with respect to inversion. This is what we expect from the theoretical analysis of the algorithms. The following example further confirms this observation.

Table 2. Numerical results for Example 2

				Newton iteration		Inverse-free iteration	
$\Delta(A)$	rcond(A)	sep	$\|\|P\|\|$	*iter*	$\frac{\|\|E_{21}\|\|_1}{\|\|A\|\|_1}$	*iter*	$\frac{\|\|E_{21}\|\|_1}{\|\|A\|\|_1}$
10^{-1}	$8.19e-04$	$2.00e-1$	$6.42e+0$	9	$8.15e-16$	9	$2.49e-16$
10^{-3}	$1.61e-07$	$2.00e-3$	$2.07e+2$	$15(10^{-13})$	$4.23e-12$	15	$1.19e-15$
10^{-5}	$4.12e-12$	$2.00e-5$	$8.06e+4$	$21(10^{-09})$	$3.27e-07$	22	$8.46e-15$
10^{-7}	$1.38e-15$	$2.00e-7$	$2.29e+6$	$28(10^{-05})$	$2.09e-04$	$28(10^{-13})$	$2.44e-13$

Example 3. The test matrices in this example are specially constructed random matrices of the form

$$A = Q^{\mathrm{T}} \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} Q, \tag{10.25}$$

where Q is an orthogonal matrix generated from the QR decomposition of a random matrix. Submatrices A_{11} and A_{22} are first set to be 5×5 random upper triangular matrices, and then their diagonal elements replaced by $d|a_{ii}|$ and $-d|a_{ii}|$, respectively, where $a_{ii} (1 \leq i \leq n)$ are other random numbers and d is a positive parameter. A_{12} is another 5×5 random matrix. As d gets small, all the eigenvalues get close to the origin and become ill-conditioned. This is the hardest kind of spectrum to divide.

The numerical results are reported in Table 3. All eigenvalues are fairly distant from the imaginary axis ($\Delta(A) \approx O(10^{-3})$), but the conditioning of the generated matrices with respect to inversion can be quite large. The separation of

A_{11} and A_{22} can also become small, and $\|P\|$ large, indicating that the eigenvalues are hard to separate. Table 3 gives results for d in the set $\{1, 0.5, 0.3, 0.2, 0.1\}$. Again, Newton iteration is inferior to inverse-free iteration for the ill-conditioned problems. In particular, in the case of $d = 0.1$, we observed that from the fourth Newton iteration onward rcond(A_4) was about $O(10^{-18})$, and that Newton failed to converge. However, the inverse-free iteration is still fairly accurate, although the convergence rate and the backward accuracy do deteriorate.

Table 3. Numerical results for Example 3

				Newton iteration		Inverse-free iteration	
d	rcond(A)	sep	$\|P\|$	*iter*	$\frac{\|E_{21}\|_1}{\|A\|_1}$	*iter*	$\frac{\|E_{21}\|_1}{\|A\|_1}$
1.0	$4.09e-06$	$1.36e-03$	$7.39e+1$	$9(10^{-13})$	$4.56e-14$	10	$7.08e-16$
0.5	$1.29e-06$	$2.37e-04$	$4.32e+2$	$11(10^{-12})$	$1.99e-12$	10	$1.66e-15$
0.3	$3.43e-10$	$4.71e-06$	$2.76e+5$	$14(10^{-07})$	$4.55e-09$	15	$1.64e-15$
0.2	$6.82e-11$	$3.94e-07$	$5.48e+4$	$16(10^{-07})$	$2.76e-08$	12	$1.43e-13$
0.1	$8.12e-14$	$1.54e-10$	$7.48e+8$	–	(fail)	$15(10^{-13})$	$3.66e-11$

11. Open problems

Here we propose some open problems about the spectral divide and conquer algorithm.

1. In Algorithm 1 with general B, we test whether l_{L} is equal to l_{R}, where l_{L} is the number of eigenvalues in the specified region determined from computing the left deflating space, and l_{R} is the number of eigenvalues in the specified region determined from computing the right deflating space. Normally, we expect them to be the same, however, what does it mean when $l_{\mathrm{L}} \neq l_{\mathrm{R}}$? Perhaps this is an indicator that the pencil is nearly singular.
2. Iterative refinement, based either on nonsymmetric Jacobi iteration [23, 27, 49, 47, 46] or refining invariant subspaces ([21] and the references therein) could be used to make E_{21} (and F_{21}) smaller if they are unacceptably large.

12. Conclusions and future work

In this paper, we have further developed the algorithm proposed by Godunov, Bulgakov and Malyshev for doing spectral divide and conquer. With reasonable storage and arithmetic cost, the new algorithm applies equally well to the standard and generalized eigenproblem, and avoids all matrix inversions in the inner loop, instead requiring QR decompositions and matrix multiplication. It forms an alternative to the matrix sign function for the parallel solution of the nonsymmetric eigenproblem.

Although the new approach eliminates the possible instability associated with inverting ill-conditioned matrices, it does not eliminate the problem of slow or misconvergence when eigenvalues lie too close to the boundary of the selected

region. Numerical experiments indicate that the distance of the eigenvalues to the boundary affects the speed of convergence of the new approach as it does to the matrix sign function based algorithm, but the new approach can yield an accurate solution even when the sign function fails. The backward error bounds given in Sect. 7 are often pessimistic. The new algorithm performs much better than our error analysis can justify. We believe that in dealing with the standard spectral divide and conquer problem, the matrix sign function based algorithm is still generally superior.

Future work includes building a "rank-revealing" generalized QR decomposition, devising an inexpensive condition estimator, incorporating iterative refinement, and understanding how to deal with (nearly) singular pencils. The applications of the inverse-free iteration for solving algebraic Riccati equations deserves closer study too.

The performance evaluation of the new algorithms on massively parallel machines, such as the Intel Delta and Thinking Machines CM-5, will appear in [9].

References

1. Ahlfors, L. (1966): Complex Analysis. McGraw-Hill
2. Ammar, G., Benner, P., Mehrmann, V. (1993): A multishift algorithm for the numerical solution of algebraic Riccati equations. Elect. Trans. on Numer. Anal., **1**:33–48
3. Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., Sorensen, D. (1995): LAPACK Users' Guide (second edition). SIAM, Philadelphia 324 pages
4. Anderson, E., Bai, Z., Dongarra, J. (1992): Generalized QR factorization and its applictions. Lin. Alg. Appl., **162–164**:243–271
5. Arnold, W.F., Laub., A.J. (1984): Generalized eigenproblem algorithms and software for algebraic Riccati equations. Proc. IEEE, **72**:1746–1754
6. Auslander, L., Tsao, A. (1992): On parallelizable eigensolvers. Advances in Applied Mathematics, **13**:253–261
7. Bai, Z., Demmel, J. (1993): Design of a parallel nonsymmetric eigenroutine toolbox, Part I. In: Proceedings of the Sixth SIAM Conference on Parallel Proceesing for Scientific Computing. SIAM. Long version available as UC Berkeley Computer Science report all.ps.Z via anonymous ftp from tr-ftp.cs.berkeley.edu, directory pub/tech-reports/csd/csd-92-718
8. Bai, Z., Demmel, J. (1993): On swapping diagonal blocks in real Schur form. Lin. Alg. Appl., **186**:73–96
9. Bai, Z., Demmel, J., Dongarra, J., Petitet, A., Robinson, H. (1995): The spectral decomposition of nonsymmetric matrices on distributed memory parallel computers. Compute Science Dept. Technical Report CS-95-273, University of Tennessee at Knoxville to appear in SIAM J. Sci. Comp.
10. Bai, Z., Demmel, J., Gu, M. (1994): Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems. Computer Science Division Report UCB//CSD-94-793, UC Berkeley, February 1994. available by anonymous ftp to tr-ftp.cs.berkeley.edu in directory pub/tech-reports/csd/csd-94-79
11. Batterson, S. (1990): Convergence of the shifted QR algorithm on 3 by 3 normal matrices. Num. Math., **58**:341–352
12. Bischof, C., Huss-Lederman, S., Sun, X., Tsao, A. (1993): The PRISM project: Infrastructure and algorithms for parallel eigensolvers. In Proceedings of the Scalable Parallel Libraries Conference, Mississippi State, Mississippi. IEEE Computer Society
13. Bojanczyk, A., Van Dooren, P. (1993): personal communication

14. Boyd, S., Balakrishnan, V. (1990): A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. Systems Control Letters, **15**:1–7
15. Boyd, S., Balakrishnan, V., Kabamba, P. (1989): A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems. Mathematics of Control, Signals, and Systems, **2(3)**:207–219
16. Bulgakov, A.Ya., Godunov, S.K. (1988): Circular dichotomy of the spectrum of a matrix. Siberian Math. J., **29(5)**:734–744
17. Buzbee, B., Golub, G., Nielsen, C. (1970): On direct methods for solving Poisson's equation. SIAM J. Num. Anal., **7**:627–656
18. Byers, R. (1987): Solving the algebraic Riccati equation with the matrix sign function. Lin. Alg. Appl., **85**:267–279
19. Byers, R. (1988): A bisection method for measuring the distance of a stable matrix to the unstable matrices. SIAM J. Sci. Stat. Comp., **9(5)**:875–881
20. Chan, T. (1987): Rank revealing QR factorizations. Lin. Alg. Appl., **88/89**:67–82
21. Demmel, J. (1987): Three methods for refining estimates of invariant subspaces. Computing, **38**:43–57
22. Demmel, J. (1993): Trading off parallelism and numerical stability. In G. Golub, M. Moonen, B. de Moor, editors, Linear Algebra for Large Scale and Real-Time Applications, pages 49–68. Kluwer Academic Publishers NATO-ASI Series E: Applied Sciences, Vol. 232; Available as all.ps.Z via anonymous ftp from tr-ftp.cs.berkeley.edu, in directory pub/tech-reports/csd/csd-92-702
23. Demmel, J., Heath, M., van der Vorst, H. (1993): Parallel numerical linear algebra. In Iserles A., editor, Acta Numerica, volume 2. Cambridge University Press
24. Demmel, J., Kågström, B. (1987) Computing stable eigendecompositions of matrix pencils. Lin. Alg. Appl., **88/89**:139–186
25. Demmel, J., Li, X. (1994): Faster numerical algorithms via exception handling. IEEE Trans. Comp., **43(8)**:983–992. LAPACK Working Note 59
26. Dongarra, J., Du Croz, J., Duff, I., Hammarling, S. (1990): A set of Level 3 Basic Linear Algebra Subprograms. ACM Trans. Math. Soft., **16(1)**:1–17
27. Eberlein, P. (1987): On the Schur decomposition of a matrix for parallel computation. IEEE Trans. Comput. P., **36**:167–174
28. Gantmacher, F. (1959): The Theory of Matrices, vol. II (transl.). Chelsea, New York
29. Gardiner, J., Laub, A. (1986): A generalization of the matrix-sign function solution for algebraic Riccati equations. Int. J. Control, **44**:823–832
30. Godunov, S.K. (1986): Problem of the dichotomy of the spectrum of a matrix. Siberian Math. J., **27(5)**:649–660
31. Golub, G., Van Loan, C. (1989): Matrix Computations. Johns Hopkins University Press, Baltimore, MD, 2nd edition
32. Gu, M., Eisenstat, S. (1993): An efficient algorithm for computing a rank-revealing QR decomposition. Computer Science Dept. Report YALEU/DCS/RR-967, Yale University
33. Gu, M., Eisenstat, S.C. (1995): A divide-and-conquer algorithm for the bidiagonal SVD. SIAM J. Mat. Anal. Appl., **16(1)**:79–92
34. Henry, G., van de Geijn, R. (1994): Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problem: myths and reality. Computer Science Dept. Technical Report CS-94-244, University of Tennessee, Knoxville. (LAPACK Working Note #79); to appear in SIAM SISC
35. Higham, N.J. (1986): Computing the polar decomposition - with applications. SIAM J. Sci. Stat. Comput., **7**:1160–1174
36. Higham, N.J., Philip, A. (1995): Knight. Matrix powers in finite precision arithmetic. SIAM J. Mat. Anal. Appl., **16**:343–358
37. Hong, P., Pan, C.T. (1992): The rank revealing QR and SVD. Math. Comp., **58**:575–232
38. Howl, J. (1983): The sign matrix and the separation of matrix eigenvalues. Lin. Alg. Appl., **49**:221–232
39. Kenney, C., Laub, A. (1991): Rational iteration methods for the matrix sign function. SIAM J. Mat. Anal. Appl., **21**:487–494
40. Malyshev, A.N. (1989): Computing invariant subspaces of a regular linear pencil of matrices. Siberian Math. J., **30(4)**:559–567

41. Malyshev, A.N. (1992): Guaranteed accuracy in spectral problems of linear algebra, I,II. Siberian Adv. in Math., **2(1,2)**:144–197,153–204
42. Malyshev, A.N. (1993): Parallel algorithm for solving some spectral problems of linear algebra. Lin. Alg. Appl., **188,189**:489–520
43. Paige, C. (1990): Some aspects of generalized QR factorization. In M. Cox and S. Hammarling, editors, Reliable Numerical Computations. Clarendon Press, Oxford
44. Parlett, B. (1980): The Symmetric Eigenvalue Problem. Prentice Hall, Englewood Cliffs, NJ
45. Roberts, J. (1980): Linear model reduction and solution of the algebraic Riccati equation. Inter. J. Control, **32**:677–687
46. Sameh, A. (1971): On Jacobi and Jacobi-like algorithms for a parallel computer. Math. Comp., **25**:579–590
47. Shroff, G. (1991): A parallel algorithm for the eigenvalues and eigenvectors of a general complex matrix. Num. Math., **58**:779–805
48. Stewart, G.W. (1973): Error and perturbation bounds for subspaces associated with certain eigenvalue problems. SIAM Review, 15(4):727–764
49. Stewart, G.W. (1985): A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix. SIAM J. Sci. Stat. Comput., **6**:853–864
50. Stewart, G.W. (1993): Updating a rank-revealing ULV decomposition. SIAM J. Mat. Anal. Appl., **14(2)**:494–499
51. Stewart, G.W., Sun, J.-G. (1990): Matrix Perturbation Theory. Academic Press, New York
52. Trefethen, L.N. (1991): Pseudospectra of matrices. In 1991 Dundee Numerical Analysis Conference Proceedings, Dundee, Scotland

This article was processed by the author using the LaTeX style file *pljour1m* from Springer-Verlag.