On a Direct Algorithm for Computing Invariant Subspaces with Specified Eigenvalues*

Zhaojun Bai† and James W. Demmel‡

Abstract

We discuss a direct algorithm for reordering the eigenvalues on the diagonal of a matrix in real Schur form by performing an orthogonal similarity transformation. A new version of the algorithm is given. A detailed error analysis and software description are presented. Numerical examples show the superiority of our algorithm over previous algorithms.

1 Introduction

The problem of reordering the eigenvalues into a desired order along the (block) diagonal of a quasi-triangular real matrix arises in several applications: computing an invariant subspace corresponding to a given group of eigenvalues, estimating condition numbers for a cluster of eigenvalues or their associated invariant subspace in the nonsymmetric eigenvalue problem [3, 20], computing partial eigenvalues of a large nonsymmetric matrix by the simultaneous iteration method [4, 16], computing matrix functions [6, 14], solving the linear quadratic control problem [13], and so on. These problems can be solved in two phases: the first is to compute the Schur decomposition of the given matrix, and the second is to reorder a group of specified eigenvalues to appear at the upper left of the matrix to get the corresponding invariant subspace. In this paper we describe an algorithm and its implementation for this reordering problem. The software is available in LAPACK [2, 1], a public domain numerial linear algebra library.

Specifically, for any given $n \times n$ matrix A, from the QR algorithm we could compute the Schur decomposition of A in the form

$$A = UTU^{\mathrm{H}},$$

where T is an upper triangular matrix, called the *Schur form*, and U is a unitary matrix. U^{H} the conjugate transpose of U, and so $U^{\mathrm{H}}U = I$. The diagonal entries of T are the eigenvalues of A. U and T may be complex even if A is real, since a real matrix may have complex eigenvalues. For a real matrix A, there is a real orthogonal matrix Q such that

$$A = QTQ^{\mathrm{T}}, \tag{1}$$

where T is a real upper quasi-triangular matrix, called the *real Schur form*. T is block upper triangular with 1×1 and 2×2 blocks on the diagonal. The 1×1 blocks contain the real eigenvalues

*This work was supported in part by NSF grant ASC-9005933 and DARPA grant DAAL03-91-C-0047 via a subcontract from the University of Tennessee. The first author was also supported in part by NSF grant ASC-9102963.

†Department of Mathematics, University of Kentucky, Lexington, KY 40506. (na.bai@na-net.ornl.gov)

‡Computer Science Division and Mathematics Department, University of California, Berkeley, CA 94720. (na.demmel@na-net.ornl.gov).

of A. The eigenvalues of the 2×2 diagonal blocks are the complex conjugate eigenvalues of A. The real Schur form may be computed using subroutine HQR from EISPACK [15] or subroutine SHSEQR from LAPACK [2]),

Here U or Q provides an orthonormal basis for the invariant subspace of certain subsets of eigenvalues of the matrix A. If we partition Q and T as

$$Q = [Q_1, Q_2], \quad T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix},$$

then from (1), we have

$$AQ_1 = Q_1 T_{11} \tag{2}$$

and hence Q_1 gives an orthonormal basis for the invariant subspaces of A corresponding to the eigenvalues contained in T_{11}.

Unfortunately, the T given by the QR algorithm will not generally contain the eigenvalues in which we are interested. We must therefore perform some further orthogonal similarities that preserve block triangular form but reorder the desired eigenvalues of A to the upper left corner of the Schur form T, to get the desired invariant subspace as in (2). The crux of such reordering or swapping techniques is how to swap two adjacent 1×1 or 2×2 diagonal blocks by an orthogonal transformation. Formally, let A_{11} be a $p \times p$ matrix, A_{22} be a $q \times q$ matrix, $p, q = 1, 2$; we want to compute an orthogonal $(p+q) \times (p+q)$ matrix Q such that

$$Q^{\mathrm{T}} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} Q = \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{11} \end{bmatrix}, \tag{3}$$

where $\tilde{A}_{ii}$ is similar to A_{ii}, $i = 1, 2$, so that the eigenvalues are unchanged but their positions are exchanged along the (block) diagonal.

To this end, Stewart [17] has described an iterative algorithm for swapping consecutive 1×1 and 2×2 blocks of a quasi-triangular matrix, which we refer to as algorithm EXCHNG. In his method, the first block is used to determine an implicit QR shift. An arbitrary QR step is performed on both blocks to create a dense $(p+q) \times (p+q)$ block. Then a sequence of QR steps using the previously determined shift is performed on both blocks. Theoretically, after one step of QR iteration, the eigenvalues of the first block will emerge in the lower part. But in practice, we may need two or even more QR iterations. This use of QR iteration has been extended by Van Dooren [21] to reordering the eigenvalues of a generalized eigenvalue problem using QZ iteration.

Another algorithm to be further developed in this paper is the so-called *direct swapping method*, which was originally motivated by the work of Dongarra, Hammarling and Wilkinson in 1983, although the paper was finished later (1990) [10]. Ng and Parlett [14] also developed a program to implement the direct swapping algorithm. A similar idea has also been published by Gao and Zhang [8].

This previous work still does not solve the problem satisfactorily. The iterative swapping algorithm has the advantage of guaranteed backward stability, since it just multiples the data by orthogonal matrices. But it may be inaccurate and even fail to reorder the eigenvalues in moderately ill-conditioned cases. On the other hand, the direct swapping algorithm is simple and can better deal with ill-conditioned cases. But the current implementations do not guarantee backward stability.

In this paper, we further improve the direct swapping algorithm. Various strategies have been designed at each stage of the algorithm to improve its accuracy and robustness. A detailed analysis

of the algorithms shows that backward instability is possible only in very ill-conditioned cases, so ill-conditioned in fact that we have been unable to construct a case where it fails. Our goal was to have an absolute stability guarantee, however; we achieved this by directly and cheaply testing for instability and rejecting a swap if it would have been unstable. This can occur only when the eigenvalues are so ill-conditioned as to be indistinguishable in a certain reasonable sense. Numerical experiments show the superiorities of our direct swapping algorithm over previous implementations.

The rest of the paper is organized as follows: Section 2 describes the direct swapping algorithm. Section 3 discusses the algorithm in presence of rounding errors. The error analysis of the algorithm is carried out in Section 4. The software implementation and numerical experments are reported in Section 5. Section 6 draws conclusions. All software related to the algorithms discussed in this paper can be found in the LAPACK library [2].

We assume that any 2×2 diagonal block in the quasi-triangular matrix is in standardized form. This means that its diagonal entries are equal and its off-diagonals nonzero and of opposite sign, that is

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \alpha \end{bmatrix}, \qquad \beta\gamma < 0. \tag{4}$$

Such a block has complex conjugate eigenvalues $\alpha \pm \mu$ where $\mu^2 = -\beta\gamma$. It is known that for any 2×2 block with complex conjugate eigenvalues, an orthogonal similarity transformation will standardize the block. The LAPACK subroutine SHSEQR returns the real Schur factorization with 2×2 blocks in standard form.

2 Direct Swapping Algorithm

As we described in the introduction, the crux of reordering the diagonal blocks to form a specified invariant subspace is to interchange the consecutive diagonal blocks A_{11} and A_{22} in the following block matrix

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \tag{5}$$

where A_{11} is $p \times p$, A_{22} is $q \times q$, $p, q = 1, 2$. Throughout this paper, we assume that A_{11} and A_{22} have no eigenvalue in common, otherwise, they need not to be exchanged. It is seen that the block matrix (5) can be diagonalized as

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} = \begin{bmatrix} I_p & -X \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I_p & X \\ 0 & I_q \end{bmatrix},$$

where X is the solution of the Sylvester equation

$$A_{11}X - XA_{22} = A_{12}. \tag{6}$$

Since it is assumed that A_{11} and A_{22} have no eigenvalue in common, the solution X is unique. If we choose an orthogonal matrix Q such that

$$Q^{\mathrm{T}} \begin{bmatrix} -X \\ I_q \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

and conformally partition Q in the form

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix},$$

then

$$Q^{\mathrm{T}} \begin{bmatrix} -X & I_p \\ I_q & 0 \end{bmatrix} = \begin{bmatrix} R & Q_{11}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}} \end{bmatrix}.$$

Since both matrices on the left are invertible so are R and Q_{12}^{T}. Thus

$$\begin{aligned} Q^{\mathrm{T}} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} Q &= Q^{\mathrm{T}} \begin{bmatrix} I_p & -X \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I_p & X \\ 0 & I_q \end{bmatrix} Q \\ &= \begin{bmatrix} R & Q_{11}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix} \begin{bmatrix} R^{-1} & -R^{-1}Q_{11}^{\mathrm{T}}Q_{12}^{-\mathrm{T}} \\ 0 & Q_{12}^{-\mathrm{T}} \end{bmatrix} \\ &= \begin{bmatrix} RA_{22}R^{-1} & -RA_{22}R^{-1}Q_{11}^{\mathrm{T}}Q_{12}^{-\mathrm{T}} + Q_{11}^{\mathrm{T}}A_{11}Q_{12}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}}A_{11}Q_{12}^{-\mathrm{T}} \end{bmatrix} \\ &\equiv \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{11} \end{bmatrix} \end{aligned}$$

where $\tilde{A}_{ii}$ is similar to A_{ii}, $i = 1, 2$, so that the eigenvalues are invariant, but their positions are exchanged. Furthermore, we have the following theorem to specify such orthogonal transformation, which is due to Ng and Parlett [14].

Theorem 1 (Ng and Parlett). *An orthogonal $(p+q) \times (p+q)$ matrix Q swaps A_{11} and A_{22} if and only if*

$$Q^{\mathrm{T}} \begin{bmatrix} -X \\ I_q \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{7}$$

for some invertible $q \times q$ matrix R where X is defined in (6).

Proof. The *if* part has been shown above, we just need to show the *only if* part. If Q swaps A_{11} and A_{22} then there exist Q_{12}, R and Q_{11}^{T} such that

$$\begin{aligned} \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{11} \end{bmatrix} &= \begin{bmatrix} R & Q_{11}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix} \begin{bmatrix} R^{-1} & -R^{-1}Q_{11}^{\mathrm{T}}Q_{12}^{-\mathrm{T}} \\ 0 & Q_{12}^{-\mathrm{T}} \end{bmatrix} \\ &= Q^{\mathrm{T}} \begin{bmatrix} -X & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix} \begin{bmatrix} 0 & I_q \\ I_p & X \end{bmatrix} Q \end{aligned}$$

It follows that $D \equiv \begin{bmatrix} R & Q_{11}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}} \end{bmatrix}^{-1} Q^{\mathrm{T}} \begin{bmatrix} -X & I_p \\ I_q & 0 \end{bmatrix}$ commutes with $\begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix}$. Since A_{11} and A_{22} have no eigenvalues in common, D must be a polynomial in $\mathrm{diag}(A_{22}, A_{11})$. See [1, vol. 1, page 222].

$$Q^{\mathrm{T}} \begin{bmatrix} -X & I_p \\ I_q & 0 \end{bmatrix} = \begin{bmatrix} R & Q_{11}^{\mathrm{T}} \\ 0 & Q_{12}^{\mathrm{T}} \end{bmatrix} D$$

must be block upper triangular. This completes the proof. □.

Thus we have the following algorithm to swap two adjacent blocks:

Algorithm 1 (Direct Swapping Algorithm)

1. *Solve the $p \times q$ Sylvester equation*

$$A_{11}X - XA_{22} = A_{12};$$

2. Compute an orthogonal matrix Q such that

$$Q^{\mathrm{T}} \begin{bmatrix} -X \\ I_q \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix};$$

3. Perform an orthogonal similarity transformation

$$Q^{\mathrm{T}} A Q = \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{11} \end{bmatrix}.$$

For literature on how to solve the Sylvester equation, see [2, 7]. One direct way is to recast it as a linear system of equations:

$$Kx = b, \tag{8}$$

where

$$K = I_q \otimes A_{11} - A_{22}^{\mathrm{T}} \otimes I_p, \qquad x = \mathrm{col}(X), \qquad b = \mathrm{col}(A_{12}). \tag{9}$$

Here the Kronecker product $W \otimes Z$ of two matrices W and Z is the block matrix whose (i, j) block is $(w_{ij}Z)$. For an $m \times n$ matrix W, $\mathrm{col}(W)$ denotes the column vector formed by taking columns of W and stacking them atop one another from left to right. That is

$$\mathrm{col}(W) = [w_{11}, w_{21}, \ldots, w_{m1}, w_{12}, w_{22}, \ldots, w_{m2}, \ldots, w_{1n}, \ldots, w_{mn}]^{\mathrm{T}}.$$

Since A_{11} and A_{22} have no common eigenvalues, the coefficient matrix K of the linear system (8) is nonsingular. Hence there is a unique solution. Note that the matrices are 1×1 or 2×2 matrices, the linear system (8) can only be up to 4×4. We can simply use Gaussian elimination to solve it.

In the second step, the orthogonal matrix Q which swaps the A_{11} and A_{22} can be computed by the QR factorization of $\begin{bmatrix} -X \\ I_q \end{bmatrix}$ using Householder or Givens transformations.

Finally, we note that from the QR factorization (7), the orthogonal matrix Q which swaps A_{11} and A_{22} can also be explicitly written as

$$Q = \begin{bmatrix} -X & I_p \\ I_q & X^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} C_1^{-1} & 0 \\ 0 & C_2^{-1} \end{bmatrix} \tag{10}$$

where

$$C_1^{\mathrm{T}} C_1 = I_q + X^{\mathrm{T}} X, \qquad C_2^{\mathrm{T}} C_2 = I_p + X X^{\mathrm{T}}. \tag{11}$$

Hence another implementation of the direct swapping algorithm is to use the explicit expression (10) for Q, after computing the Cholesky factorizations (11). Ng and Parlett implemented this scheme, but our numerical experiments show that this scheme is much less robust than Algorithm 1, because of the numerical sensitivity of the Cholesky factorization (11), and the use of the inverses C_1^{-1} and C_2^{-1}. We have not pursued this scheme.

3 The Practical Direct Swapping Algorithm

In the presence of roundoff, the biggest concern is solving the Sylvester equation (6). The linear system (8) could possibly be ill-conditioned if A_{11} and A_{22} have close eigenvalues. In the extreme

case, if A_{11} and A_{22} have the same eigenvalues, the Sylvester equation is singular and the solution X is infinite. To prevent possible overflow, instead we solve the equation

$$A_{11}X - XA_{22} = \gamma A_{12} \tag{12}$$

or the corresponding linear system

$$Kx = \gamma b \tag{13}$$

where γ is a scaling factor ($\gamma \leq 1$), and K is defined as (9). Possible overflow of X is taken care of by choosing a small scaling factor γ. In the extreme case, when A_{11} and A_{22} have the same eigenvalues, we choose $\gamma = 0$. Because the linear system (13) can only be up to 4×4, it does not cost too much to use Gaussian elimination with complete pivoting to solve it with better numerical properties (in particular, the pivots are within a modest factor of the singular values of the 4 by 4 matrix, so setting tiny pivots to a chosen tiny value controls the conditioning of the system and norm of the solution). Applying standard results from [22], straightforward analysis shows that for the computed solution $\bar{X}$ of the Sylvester equation:

$$\frac{\|E\|_F}{\|X\|_F} \leq \rho\varepsilon\,(\|A\|_F + \|B\|_F)\|K^{-1}\|_2, \tag{14}$$

where $E = X - \bar{X}$, ρ is small constant of order $O(1)$, and ε is machine precision. Notice that $\|K^{-1}\|_2$ is the reciprocal of the minimal singular value of K, denoted $\sigma_{min}(K)$. Since $\sigma_{min}(K)$ is closely related to the separation of the spectra of matrices A_{11} and A_{22}, $\sigma_{min}(K)$ is also called the separation of the matrices A_{11} and A_{22}, denoted $sep(A_{11}, A_{22})$ [20]. Therefore (14) can be written as

$$\frac{\|E\|_F}{\|X\|_F} \leq \frac{\rho\varepsilon\,(\|A\|_F + \|B\|_F)}{sep(A_{11}, A_{22})}. \tag{15}$$

In the following error analysis of the algorithm, we will see that the numerical stability is essentially governed by the residual

$$Y \equiv A_{12} - A_{11}\bar{X} + \bar{X}A_{22} = -A_{11}E + EA_{22}.$$

Applying standard error analysis of Gaussian elimination, [12] we have

$$\|Y\|_F = \|A_{12} - A_{11}\bar{X} + \bar{X}A_{22}\|_F \leq \rho\varepsilon\,(\|A_{11}\|_F + \|A_{22}\|_F)\|X\|_F. \tag{16}$$

Notice that the bound does not involve $\|K^{-1}\|_2$, or $sep(A_{11}, A_{22})$.

Next for the QR factorization of the matrix $\begin{bmatrix} -\bar{X} \\ \gamma I \end{bmatrix}$, by Householder elementary reflectors, we know that

$$\begin{bmatrix} -\bar{X} \\ \gamma I \end{bmatrix} = \bar{Q}\begin{bmatrix} \bar{R} \\ 0 \end{bmatrix}, \tag{17}$$

where $\bar{Q} = Q + \delta Q$, $\|\delta Q\| \approx \varepsilon$, $\bar{Q}^T\bar{Q} = I$, i.e., the computed matrix $\bar{Q}$ is orthogonal to machine precision [22].

We will show in the next section, in some pathological cases, the norm of the (2, 1) entry (block) of $\bar{Q}^T A \bar{Q}$ may be larger than $O(\varepsilon)\|A\|$, i.e., it may be backward unstable if we are forced to treat $\bar{Q}^T A \bar{Q}$ as block upper triangular by setting the (2, 1) entry to zero. Therefore we propose to perform adjacent blocks swapping tentatively; if the norm of the (2, 1) entry (block) of $\bar{Q}^T A \bar{Q}$ is less than or equal to $O(\varepsilon)\|A\|$, we swap them, otherwise we return without performing the swap.

If the two blocks are exchanged, then an orthogonal similarity transformation is performed on the 2×2 blocks (if any exist) to return them to standard form (4).

Finally, since the nonsymmetric eigenvalue problem is an ill-conditioned problem, a small perturbation to a 2×2 block (complex conjugate eigenpair) could cause a large perturbation of its eigenvalues. In the extreme case, a 2×2 block could split into two 1×1 blocks if its complex conjugate eigenvalues become real. Carefully designed standardization steps will detect and report such phenomena.

All above considerations are summed up in the following algorithm.

Algorithm SLAEXC (Direct Swapping Algorithm using floating point arithmetic)

1. *Copy A to T:*
$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \leftarrow A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$
2. *Use Gaussian elimination with complete pivoting to solve*
$$T_{11}X - XT_{22} = \gamma T_{12},$$
where γ is a scaling factor to prevent overflow. If there is a small diagonal element during Gaussian elimination, set it to roughly machine precision (scaled by the norm of the matrix if desired).
3. *Compute the QR factorization* $\begin{bmatrix} -X \\ \gamma I \end{bmatrix} = QR$ *by Householder transformations.*
4. *Perform swapping tentatively. Decide whether to accept swap: if the norm of the (2,1) entry (block) of $Q^T T Q$ is less than or equal to $O(\varepsilon)\|T\|_M$, go to the next step, and otherwise exit;*
5. *If the swap is accepted, transform A by Q and set the (2,1) entry (block) of $Q^T A Q$ to zero.*
6. *Standardize 2×2 block(s) if any exist.*

In our implementation of SLAEXC in LAPACK, we have chosen $10\varepsilon \|A\|_M$ as the stability criterion in step 4, where $\|A\|_M = \max_{i,j} |a_{ij}|$.

Finally, we note that we also provide a subroutine STREXC in LAPACK which calls SLAEXC to reorder all the eigenvalues into a user selected order. In particular, the user may select any subset of the spectrum which will be reordered to appear at the top left of the matrix using the fewest possible calls to SLAEXC.

4 Error Analysis of the Direct Swapping Algorithm

In this section, we give an error analysis of the direct swapping algorithm SLAEXC described in the last section. In the interest of brevity, we assume that $p = q = 2$, i.e., we only consider swapping two 2×2 blocks, the hardest case of the problem. In addition, we also assume that the scaling factor $\gamma = 1$. Quantities with bars (like $\bar{X}$) denote computed quantities.

Let $\bar{X}$ be the computed solution of the Sylvester equation, where $\bar{X} = X + E$, X is the exact solution, and E is an error matrix. By the argument of (17), and a result of Stewart [19], the

perturbation of the QR factorization, we know that the QR factorization of $\begin{bmatrix} -\bar{X} \\ I \end{bmatrix}$ can be written as

$$\bar{G} = \begin{bmatrix} -X \\ I \end{bmatrix} + \begin{bmatrix} -E \\ 0 \end{bmatrix} = \bar{Q}\bar{R} = (Q+W) \begin{bmatrix} R+F \\ 0 \end{bmatrix}, \tag{18}$$

where W and F are the perturbations of orthogonal matrix Q and triangular matrix R, respectively, $\bar{Q} = Q + W$ is orthogonal. $\|W\|$ and $\|F\|$ are essentially bounded by the terms of order $\|G\|^{-1}\|E\|$. From $(Q+W)^{\mathrm{T}}(Q+W) = I$, up to the first order of the perturbation, we have

$$Q^{\mathrm{T}}W = -W^{\mathrm{T}}Q.$$

When $\bar{Q} = Q + W$ transforms A, ignoring the second order perturbation we have

$$\begin{aligned} \bar{Q}^{\mathrm{T}}A\bar{Q} &= (Q+W)^{\mathrm{T}}A(Q+W) \\ &= Q^{\mathrm{T}}AQ + W^{\mathrm{T}}AQ + Q^{\mathrm{T}}AW + W^{\mathrm{T}}AW \\ &= \tilde{A} + W^{\mathrm{T}}Q \cdot Q^{\mathrm{T}}AQ + Q^{\mathrm{T}}AQ \cdot Q^{\mathrm{T}}W \\ &= \tilde{A} + \tilde{A}Q^{\mathrm{T}}W - Q^{\mathrm{T}}W\tilde{A}. \end{aligned}$$

Defining $Z = Q^{\mathrm{T}}W$ and partitioning it conformally with $\tilde{A}$ in the form

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix},$$

we have

$$\bar{Q}^{\mathrm{T}}A\bar{Q} = \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} + \begin{bmatrix} E_{22} & E_{12} \\ E_{21} & E_{11} \end{bmatrix}, \tag{19}$$

where

$$\begin{aligned} E_{11} &= \tilde{A}_{11}Z_{22} - Z_{22}\tilde{A}_{11} - Z_{21}\tilde{A}_{12}, \\ E_{22} &= \tilde{A}_{22}Z_{11} - Z_{11}\tilde{A}_{22} + \tilde{A}_{12}Z_{21}, \\ E_{21} &= \tilde{A}_{11}Z_{21} - Z_{21}\tilde{A}_{22}, \end{aligned}$$

E_{11} and E_{22} perturb the eigenvalues directly, E_{21} is of interest because it essentially reflects the numerical stability of swapping, E_{12} is the error to the block $\tilde{A}_{12}$. It is not of interest since it neither affects the numerical stability of the algorithm nor the perturbation of eigenvalues. The following task is to give bounds on the norms of E_{11}, E_{22} and E_{21}. To do so, let us first express Z_{ij} in terms of the blocks Q_{ij} of Q, E and R. From (18), we have

$$(I + Q^{\mathrm{T}}W) \begin{bmatrix} R+F \\ 0 \end{bmatrix} = Q^{\mathrm{T}} \begin{bmatrix} -X \\ I \end{bmatrix} + Q^{\mathrm{T}} \begin{bmatrix} -E \\ 0 \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} + \begin{bmatrix} -Q_{11}^{\mathrm{T}}E \\ -Q_{12}^{\mathrm{T}}E \end{bmatrix}.$$

Postmultiplying by $(R+F)^{-1}$ on both sides of the above equation, and noting that $Z = Q^{\mathrm{T}}W$, the result is

$$(I + Z) \begin{bmatrix} I \\ 0 \end{bmatrix} = \begin{bmatrix} R - Q_{11}^{\mathrm{T}}E \\ -Q_{12}^{\mathrm{T}}E \end{bmatrix} (R+F)^{-1},$$

therefore

$$\begin{aligned} Z_{11} &= -I + (R - Q_{11}^{\mathrm{T}}E)(R+F)^{-1}, \\ Z_{21} &= -Q_{12}^{\mathrm{T}}E(R+F)^{-1}, \end{aligned}$$

and up to the first order perturbations, we have

$$Z_{11} = -Q_{11}^{\mathrm{T}} E R^{-1} \quad (20)$$
$$Z_{21} = -Q_{12}^{\mathrm{T}} E R^{-1} \quad (21)$$

To express Z_{22}, again from (18),

$$(I + QW^{\mathrm{T}}) \begin{bmatrix} -X & -E \\ I & \end{bmatrix} = Q \begin{bmatrix} R \\ 0 \end{bmatrix} + Q \begin{bmatrix} F \\ 0 \end{bmatrix} = \begin{bmatrix} -X \\ I \end{bmatrix} + Q \begin{bmatrix} F \\ 0 \end{bmatrix}.$$

By canceling $\begin{bmatrix} -X \\ I \end{bmatrix}$ from both sides of the equation, and premultiplying by Q, we obtain

$$W^{\mathrm{T}} \begin{bmatrix} -X & -E \\ I & \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} + Q^{\mathrm{T}} \begin{bmatrix} E \\ 0 \end{bmatrix}.$$

Inserting $QQ^{\mathrm{T}} = I$ in the left side of the above equation, we have

$$W^{\mathrm{T}} Q Q^{\mathrm{T}} \begin{bmatrix} -X & -E \\ I & \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} + Q^{\mathrm{T}} \begin{bmatrix} E \\ 0 \end{bmatrix}.$$

Since $W^{\mathrm{T}} Q = -Q^{\mathrm{T}} W = -Z$, we have

$$Z \begin{bmatrix} R \\ 0 \end{bmatrix} - Z \begin{bmatrix} Q_{11}^{\mathrm{T}} E \\ Q_{12}^{\mathrm{T}} E \end{bmatrix} = - \begin{bmatrix} F \\ 0 \end{bmatrix} - \begin{bmatrix} Q_{11}^{\mathrm{T}} E \\ Q_{12}^{\mathrm{T}} E \end{bmatrix}.$$

Thus the "bottom" equation is

$$Z_{21} R - Z_{21} Q_{11}^{\mathrm{T}} E - Z_{22} Q_{12}^{\mathrm{T}} E = -Q_{12}^{\mathrm{T}} E,$$

by (21) and assuming that error matrix E is nonsingular, then

$$Z_{22} = -Z_{21} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}} = Q_{12}^{\mathrm{T}} E R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}}. \quad (22)$$

¿From expressions (20), (21) and (22) of Z_{11}, Z_{21} and Z_{22}, the E_{11}, E_{22} and E_{21} are recast as

$$\begin{aligned} E_{11} &= Q_{12}^{\mathrm{T}} A_{11} Q_{12}^{-\mathrm{T}} Q_{12}^{\mathrm{T}} E R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-T} - Q_{12}^{\mathrm{T}} E R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-T} Q_{12}^{\mathrm{T}} A_{11} Q_{12}^{-\mathrm{T}} \\ &\quad + Q_{12}^{\mathrm{T}} E R^{-1} (-R A_{22} R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}} + Q_{11}^{\mathrm{T}} A_{11} Q_{12}^{-\mathrm{T}}) \\ &= Q_{12}^{\mathrm{T}} (A_{11} E - E A_{22}) R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}} \\ &= -Q_{12}^{\mathrm{T}} Y R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}}, \end{aligned}$$

and

$$\begin{aligned} E_{22} &= -R A_{22} R^{-1} Q_{11}^{\mathrm{T}} E R^{-1} + Q_{11}^{\mathrm{T}} E R^{-1} R A_{22} R^{-1} \\ &\quad - (-R A_{22} R^{-1} Q_{11}^{\mathrm{T}} Q_{12}^{-\mathrm{T}} + Q_{11}^{\mathrm{T}} A_{11} Q_{12}^{-\mathrm{T}}) Q_{12}^{\mathrm{T}} E R^{-1} \\ &= Q_{11}^{\mathrm{T}} (-A_{11} E + E A_{22}) R^{-1} \\ &= Q_{11}^{\mathrm{T}} Y R^{-1}, \end{aligned}$$

and

$$\begin{aligned} E_{21} &= -Q_{12}^{\mathrm{T}} A_{11} Q_{12}^{-\mathrm{T}} Q_{12}^{\mathrm{T}} E R^{-1} + Q_{12}^{\mathrm{T}} E R^{-1} R A_{22} R^{-1} \\ &= -Q_{12}^{\mathrm{T}} (-A_{11} E + E A_{22}) R^{-1} \\ &= Q_{12}^{\mathrm{T}} Y R^{-1}. \end{aligned}$$

We see that up to first order perturbations, E_{11}, E_{22} and E_{21} are essentially related to the residual vector Y of the Sylvester equation solver, R and the subblocks Q_{11} and Q_{12} of Q. Furthermore, rewriting (7) as

$$\begin{bmatrix} -X \\ I \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{21} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

we see that

$$Q_{21} = R^{-1}$$

and

$$R^{\mathrm{T}} R = I + X^{\mathrm{T}} X.$$

Let $\sigma(C)$ denote the set of singular values of matrix C, and $\lambda(C)$ denote the set of eigenvalues of matrix C, then

$$\sigma^2(R) = \lambda(R^{\mathrm{T}} R) = \lambda(I + X^{\mathrm{T}} X) = 1 + \lambda(X^{\mathrm{T}} X) = 1 + \sigma^2(X).$$

Therefore

$$\|Q_{21}\|_2 = \|R^{-1}\|_2 = \frac{1}{\sigma_2(R)} = \frac{1}{(1+\sigma_2^2(X))^{1/2}}, \tag{23}$$

where $\sigma_1(C)$, $\sigma_2(C)$ denote the singular values of 2×2 matrix C with $\sigma_1(C) \geq \sigma_2(C) \geq 0$. Now to estimate the norm of the blocks Q_{ij} of Q, we use the following CS decomposition (cosine-sine decomposition) of a partitioned orthogonal matrix, which was introduced by Stewart [18] although it is implicit in a paper of Davis and Kahan [9]. The decomposition has led to some useful results on the singular values of products and difference of projections. A proof of the existence of the decomposition can be found in [20].

CS Decomposition *Let the orthogonal matrix $Q \in \mathbf{R}^{2k \times 2k}$ be partitioned in the form*

$$Q = \begin{array}{c} \\ k \\ k \end{array} \begin{array}{c} \begin{array}{cc} k & k \end{array} \\ \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \end{array}$$

Then there are orthogonal matrices $U = \mathrm{diag}(U_1, U_2)$ and $V = \mathrm{diag}(V_1, V_2)$ with $U_1, V_1 \in \mathbf{R}^{k \times k}$ such that

$$U^{\mathrm{T}} Q V = \begin{array}{c} \\ k \\ k \end{array} \begin{array}{c} \begin{array}{cc} k & k \end{array} \\ \begin{pmatrix} C & S \\ -S & C \end{pmatrix} \end{array}$$

where

$$C = \mathrm{diag}(c_1, c_2, \ldots, c_k) \geq 0, \quad S = \mathrm{diag}(s_1, s_2, \ldots, s_k) \geq 0, \quad C^2 + S^2 = I.$$

By the CS decomposition of Q and (23), we have

$$\|Q_{11}\|_2 = \frac{\sigma_1(X)}{(1+\sigma_1^2(X))^{1/2}}$$

and

$$\|Q_{12}\|_2 = \|Q_{21}\|_2; \quad \|Q_{22}\|_2 = \|Q_{11}\|_2;$$

Thus, for E_{11}, we have

$$\|E_{11}\|_2 \le \|Q_{12}^{\mathrm{T}}\|_2 \, \|Y\|_{\mathrm{F}} \, \|R^{-1}\|_2 \, \|Q_{11}^{\mathrm{T}}\|_2 \, \|Q_{12}^{-\mathrm{T}}\|_2 = \frac{\sigma_1(X)}{1+\sigma_2^2(X)} \|Y\|_{\mathrm{F}}.$$

Similarly, for E_{22}, we have

$$\|E_{22}\|_2 \le \|Q_{11}^{\mathrm{T}}\|_2 \, \|Y\|_{\mathrm{F}} \, \|R^{-1}\|_2 \le \frac{\sigma_1(X)}{(1+\sigma_1^2(X))^{1/2}(1+\sigma_2^2(X))^{1/2}} \|Y\|_{\mathrm{F}}.$$

Finally, for E_{21}, we have

$$\|E_{21}\|_2 \le \|Q_{12}^{\mathrm{T}}\|_2 \, \|Y\|_{\mathrm{F}} \, \|R^{-1}\|_2 = \frac{1}{1+\sigma_2^2(X)} \|Y\|_{\mathrm{F}}.$$

Hence we have the following theorem

Theorem 2 *Let $Y = A_{12} - A_{11}\bar{X} + \bar{X}A_{22}$, where $\bar{X} = X + E$ is the computed solution of the Sylvester equation (6), assume that the error matrix E is nonsingular, let the QR factorization of $\begin{bmatrix} -\bar{X} \\ I \end{bmatrix}$ satisfies*

$$\begin{bmatrix} -\bar{X} \\ I \end{bmatrix} = \bar{Q} \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix},$$

then

$$\bar{Q}^{\mathrm{T}} A \bar{Q} = \begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{12} \\ 0 & \tilde{A}_{11} \end{bmatrix} + \begin{bmatrix} E_{22} & E_{12} \\ E_{21} & E_{11} \end{bmatrix}$$

where $\tilde{A}_{ii}$ is similar to A_{ii}, $i = 1, 2$, and up to the first order perturbation $O(\|E\|^2)$:

$$\|E_{11}\|_2 \le \frac{\sigma_1(X)}{1+\sigma_2^2(X)} \|Y\|_{\mathrm{F}}, \tag{24}$$

$$\|E_{22}\|_2 \le \frac{\sigma_1(X)}{(1+\sigma_1^2(X))^{1/2}(1+\sigma_2^2(X))^{1/2}} \|Y\|_{\mathrm{F}}, \tag{25}$$

$$\|E_{21}\|_2 \le \frac{1}{1+\sigma_2^2(X)} \|Y\|_{\mathrm{F}}, \tag{26}$$

where $\sigma_1(X) \ge \sigma_2(X) \ge 0$ are the singular values of 2×2 matrix X.

We make the following remarks for the above theorem

Remark 1. From the theorem, we see that the departure from upper block-triangular form (the measure of numerical instability) is dominated by $\|Y\|_{\mathrm{F}}/(1+\sigma_2^2(X))$. From the norm of the residual Y (16), we have

$$\|Y\|_{\mathrm{F}} \le \rho\varepsilon \, (\|A_{11}\|_{\mathrm{F}} + \|A_{22}\|_{\mathrm{F}}) \|X\|_{\mathrm{F}}. \tag{27}$$

On the other hand, it is easy to see that

$$\|X\|_F \leq \frac{\|A_{12}\|_F}{\mathrm{sep}(A_{11}, A_{22})}. \tag{28}$$

where the equality is attained when $\mathrm{vec}(A_{12})$ is a left singular vector of K corresponding to the smallest singular value $\sigma_{mn}(K) = \mathrm{sep}(A_{11}, A_{22})$. Combining the above two inequalities, we have

$$\|E_{21}\|_2 \leq \frac{\rho\varepsilon\,(\|A_{11}\|_F + \|A_{22}\|_F)\|A_{12}\|_F}{(1+\sigma_2^2(X))\mathrm{sep}(A_{11}, A_{22})}.$$

Logically, the above bound indicates that the numerical instability will occur if we have small $\mathrm{sep}(A_{11}, A_{22})$. But in practice, numerical experiments show that this upper bound is very pessimistic. Small $\mathrm{sep}(A_{11}, A_{22})$ does not imply instability. We will discuss this further in the following section.

Remark 2. Iterative refinement applied to the Sylvester equation will improve the accuracy of computed $\bar{X}$, (unless the Sylvester equation is too close to singular), but it need not improve $\|Y\|$ at least when Gaussian elimination with complete pivoting is used to solve the Sylvester equation.

Remark 3. The factor $\rho(X)/(1+\sigma_2^2(X))$ that affects $\|E_{11}\|_2$ and $\|E_{22}\|_2$ is interesting, since it warns that large and ill-conditioned X may endanger accuracy, because of (27) and

$$\frac{\sigma_1(X)}{1+\sigma_2^2(X)} = \frac{\mathrm{cond}(X)}{\sigma_2(X) + \sigma_2^{-1}(X)}.$$

where $\mathrm{cond}(X) = \sigma_1(X)/\sigma_2(X)$. How $\mathrm{cond}(X)$, $\mathrm{sep}(A_{11}, A_{22})$, and the accuracy of the swapped eigenvalues are related in practice needs further investigation.

5 Software Development and Numerical Experiments

In this section, we first discuss the development of software for the swapping algorithm SLAEXC. Then we discuss numerical experiments to show the capability of our software to deal with ill-conditioned cases, compare with Stewart's swapping algorithm EXCHNG, and finally demonstrate the sharpness of our perturbation bounds.

All numerical experiments were carried out on a SUN sparc station 1+. The arithmetic is IEEE standard single precision, with machine precision $\varepsilon = 2^{-23} \approx 1.192 \times 10^{-7}$.

5.1 Software development

A set of FORTRAN subroutines has been developed to implement the direct swapping algorithm described in Section 3. It is part of LAPACK project[2]. As with other LAPACK routines, this algorithm was designed for accuracy, robustness and portability.

The main subroutine is called `STREXC`. `STREXC` moves a given 1×1 or 2×2 diagonal block of a real quasi-triangular matrix to a user specified position. On return, parameter `INFO` reports whether the given block has moved to the desired position, or whether there are blocks too close to swap, and what is the current position of the given block. The subroutine `STREXC` is supported by subroutine `SLAEXC`, which perform a swap to exchange adjacent blocks. The subroutine `SLAEXC` is an implementation of the algorithm SLAEXC described in Section 3, where the subproblem of solving the Sylvester equation (12) by Gaussian elimination with complete pivoting is implemented in

subroutine SLASY2, and the subproblem of standardizing a 2×2 block is implemented in subroutine SLANV2.

In the interest of simplicity, we also used some other subroutines from LAPACK and the BLAS to perform some basic linear algebra operations, such as generating Householder transformations, computing the 2-norm of a vector and so on.

Finally, a test subroutine has been written to automatically test the subroutine SLAEXC. There are nested loops over different block sizes, different numerical scales, and different conditionings of the problem

5.2 Numerical experiments

5.2.1 Backward stability test

To measure the backward stability of a swapping algorithm, we need to test (I) how close the computed orthogonal matrix $\bar{Q}$ is to the identity matrix, and (II) how close $\bar{Q}\bar{\tilde{A}}\bar{Q}^{\mathrm{T}}$ is to the original matrix A. In other words, we need to test whether the two quantities:

$$E_Q = \frac{\|I - \bar{Q}^{\mathrm{T}}\bar{Q}\|_1}{\varepsilon}; \qquad E_A = \frac{\|A - \bar{Q}\bar{\tilde{A}}\bar{Q}^{\mathrm{T}}\|_1}{\varepsilon \|A\|},$$

are around 1, where ε is machine precision. To check the changes among eigenvalues is not required to judge the correctness of an algorithm, since we know that there must have at least an order of $O(\varepsilon \|A\|)$ perturbation to the original matrix after swapping, and the nonsymmetric eigenvalue problem is potentially ill-conditioned. However, for a reasonably conditioned matrices, the changes in the eigenvalues do measure the accuracy of a swapping algorithm. For this reason, in the following numerical examples, we also compare the eigenvalues before and after swapping, besides checking backward stable quantities E_Q and E_A.

We have done extensive testing on matrices with various mixtures of the block sizes, scales and closeness among eigenvalues. More specifically, we show the algorithm SLAEXC on the following four types of matrices:

Test Matrix 1: well separation of A_{11} and A_{22}, the eigenvalues before swapping are

$$\lambda_1 = 0.2000000E+01 \pm i\, 0.2085666E+02$$
$$\lambda_2 = 0.1000000E+01 \pm i\, 0.2017424E+02$$

Test Matrix 2: moderate separation separation of A_{11} and A_{22}, the eigenvalues before swapping are:

$$\lambda_1 = 0.1000000E+01 \pm i\, 0.1732051E+01$$
$$\lambda_2 = 0.1001000E+01 \pm i\, 0.1732916E+01$$

Test Matrix 3: close eigenvalues, the corresponding the Sylvester equation is very ill-conditioned, the eigenvalues before swapping are

$$\lambda_1 = 0.1000000E+01 \pm i\, 0.1000000E+01$$
$$\lambda_2 = 0.1001000E+01 \pm i\, 0.1000000E+01$$

Test Matrix 4: the extreme case, where the eigenvalues of A_{11} and A_{22} is the same, theoretically, the Sylvester equation solution is infinite. This matrix is used to test the robustness of our software against overflow.

Table 1 summarizes the results of algorithm SLAEXC, where $\mathrm{sep}(A_{11}, A_{22})$ is computed by MATLAB, it is included here for the interest of theoretical analysis. From Table 1, we see that both the backward stability and accuracy of the algorithm SLAEXC are satisfactory.

Table 1: numerical tests of algorithm SLAEXC

Test	matrix	$sep(A_{11}, A_{22})$	E_Q	E_A	eigenvalues after swapping
1	$\begin{pmatrix} 2 & -87 & -20000 & 10000 \\ 5 & 2 & -20000 & -10000 \\ 0 & 0 & 1 & -11 \\ 0 & 0 & 37 & 1 \end{pmatrix}$	3.337×10^{-1}	0.260	0.197	$0.1000001E+01 \pm i\,0.2017424E+02$ $0.2000000E+01 \pm i\,0.2085665E+02$
2	$\begin{pmatrix} 1 & -3 & 3576 & 4888 \\ 1 & 1 & -88 & -1440 \\ 0 & 0 & 1.001 & -3 \\ 0 & 0 & 1.001 & 1.001 \end{pmatrix}$	8.442×10^{-4}	0.625	0.423	$0.1001000E+01 \pm i\,0.1732917E+01$ $0.1000000E+01 \pm i\,0.1732051E+01$
3	$\begin{pmatrix} 1 & -100 & 400 & -1000 \\ 0.01 & 1 & 1200 & -10 \\ 0 & 0 & 1.001 & -0.01 \\ 0 & 0 & 100 & 1.001 \end{pmatrix}$	2.000×10^{-7}	0.417	0.001	$0.1000996E+01 \pm i\,0.1000360E+01$ $0.1000003E+01 \pm i\,0.9995396E+00$
4	$\begin{pmatrix} 1 & -3 & 3 & 2 \\ 1 & 1 & 9 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	∞	0.687	0.241	$0.9999987E+00 \pm i\,0.1732051E+01$ $0.1000002E+01 \pm i\,0.1732051E+01$

5.2.2 Comparison with Stewart's algorithm EXCHNG

We have done numerical comparisons between the direct swapping algorithm SLAEXC and Stewart's swapping algorithm EXCHNG [17], which uses QR iteration. Both algorithms perform well in most cases, but in certain cases, the algorithm EXCHNG is inferior to algorithm SLAEXC. For example, let

$$A(\tau) = \begin{pmatrix} 7.001 & -87 & 39.4\tau & 22.2\tau \\ 5 & 7.001 & -12.2\tau & 36.0\tau \\ 0 & 0 & 7.01 & -11.7567 \\ 0 & 0 & 37 & 7.01 \end{pmatrix},$$

where τ is a parameter, the matrix $A(\tau)$ has invariant eigenvalues

$$\begin{aligned} \lambda_1 &= 0.7001000E+01 \pm i\, 0.2085666E+02 \\ \lambda_2 &= 0.7010000E+01 \pm i\, 0.2085660E+02, \end{aligned}$$

$sep(A_{11}, A_{22}) = 0.0024$. When $\tau = 1$, the output matrix of the algorithm SLAEXC is

$$\tilde{\bar{A}} = \begin{pmatrix} 0.70100012E+01 & -0.86993660E+02 & -0.39390938E+02 & -0.22241005E+02 \\ 0.50003409E+01 & 0.70100012E+01 & 0.12191071E+02 & -0.35999401E+02 \\ 0.00000000E+00 & 0.00000000E+00 & 0.70009995E+01 & -0.11755549E+02 \\ 0.00000000E+00 & 0.00000000E+00 & 0.37003792E+02 & 0.70009995E+01 \end{pmatrix}$$

The eigenvalues after swapping are

$$\begin{aligned} \tilde{\lambda}_2 &= 0.7010001E+01 \pm i\, 0.2085661E+02, \\ \tilde{\lambda}_1 &= 0.7000999E+01 \pm i\, 0.2085665E+02, \end{aligned}$$

which is accurate in machine precision. However, the output of algorithm EXCHNG after 8 QR iterations[1] is

$$\tilde{\bar{A}} = \begin{pmatrix} 0.28140299E+02 & -0.81122643E+02 & -0.39849255E+02 & -0.15834051E+02 \\ 0.10856283E+02 & -0.14087547E+02 & -0.23942078E+02 & 0.32877380E+02 \\ 0.00000000E+00 & 0.00000000E+00 & 0.19211971E+02 & 0.21227583E+02 \\ 0.00000000E+00 & 0.00000000E+00 & -0.27540298E+02 & -0.52427406E+01 \end{pmatrix},$$

[1] where the stopping criterion used in QR iteration is $eps = 1.2 \times 10^{-7}$.

Table 2: comparison of algorithms SLAEXC and EXCHNG

τ	SLAEXC	EXCHNG
1	$\tilde{\lambda}_2 = 0.7010001E+01 \pm i\, 0.2085661E+02$	$\tilde{\lambda}_2 = 0.7026377E+01 \pm i\, 0.2085408E+02$
	$\tilde{\lambda}_1 = 0.7000999E+01 \pm i\, 0.2085665E+02$	$\tilde{\lambda}_1 = 0.6984615E+01 \pm i\, 0.2085919E+02$
10	$\tilde{\lambda}_2 = 0.7010000E+01 \pm i\, 0.2085660E+02$	$\tilde{\lambda}_2 = 0.7063053E+01 \pm i\, 0.2086175E+02$
	$\tilde{\lambda}_1 = 0.7000999E+01 \pm i\, 0.2085665E+02$	$\tilde{\lambda}_1 = 0.6947970E+01 \pm i\, 0.2085144E+02$
100	$\tilde{\lambda}_2 = 0.7009999E+01 \pm i\, 0.2085660E+02$	not convergent
	$\tilde{\lambda}_1 = 0.7000999E+01 \pm i\, 0.2085665E+02$	after 30 QR steps

which has eigenvalues

$$\tilde{\lambda}_2 = 0.7026377E+01 \pm i\, 0.2085408E+02$$
$$\tilde{\lambda}_1 = 0.6984615E+01 \pm i\, 0.2085919E+02$$

for λ_2, it still has two decimal digits correct, but, for λ_1, all significant digits have been lost. By the way, after standardization of $\tilde{A}$, it becomes

$$\bar{\tilde{A}} = \begin{pmatrix} 0.70263767E+01 & -0.86978951E+02 & -0.39378300E+02 & 0.22319088E+02 \\ 0.49999757E+01 & 0.70263767E+01 & 0.12174266E+02 & 0.35997513E+02 \\ 0.00000000E+00 & 0.00000000E+00 & 0.69846153E+01 & 0.11755766E+02 \\ 0.00000000E+00 & 0.00000000E+00 & -0.37012115E+02 & 0.69846153E+01 \end{pmatrix}.$$

Table 2 shows the numerical results with different choices of parameter τ, where when $\tau = 10$, it takes 17 QR iterations to convergence. It clearly shows the superiority of algorithm SLAEXC. In particular, we note that algorithm EXCHNG is nonconvergent when $\tau = 100$. It means that the eigenvalues are not able to be exchanged by algorithm EXCHNG. But the algorithm SLAEXC has no difficulty. This convergence difficulty may reflect recent work of Batterson [1] who has discovered classes of nonsymmetric matrices where QR iteration does fail to converge, or converges quite slowly.

5.2.3 On the upper bound (26) of $\|E_{21}\|_2$

Finally, in the interest of theoretical analysis, we discuss the sharpness of the bound on $\|E_{21}\|_2$ which controls the numerical stability of algorithm SLAEXC. In most of the test examples, we see that the bound (26) of $\|E_{21}\|_2$ is very pessimistic. However, we do find some examples indicating that the bound in (26) can roughly be attained. Let us consider the following example[2]:

$$A = \begin{array}{c} \\ 2 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 2 \end{array} \\ \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \end{array} = \begin{pmatrix} 1.0000E+00 & -1.0000E+02 & 1.9900E+04 & 1.0201E+02 \\ 1.0000E-02 & 1.0000E+00 & 1.0000E+02 & -1.9800E+00 \\ 0 & 0 & 1.0100E+00 & -1.0000E-02 \\ 0 & 0 & 1.0000E+02 & 1.0100E+00 \end{pmatrix}$$

where $\mathrm{sep}(A_{11}, A_{22}) = 2 \times 10^{-6}$. The A_{12} block of A is designed so that

$$X = \begin{pmatrix} 1.0000E+00 & -2.0000E+02 \\ 1.0000E+00 & -1.0000E+00 \end{pmatrix}$$

[2]For brevity, only five digits are displayed for all the data in this section though we did run in double precision.

is the solution of the Sylvester equation. Note that $\sigma_{\max}(X) \approx 200.01$, $\sigma_2(X) = 0.99498$. We used MATLAB to compute the different quantities in the bound (where machine precision is doubled $\varepsilon_M = 2.2204 \times 10^{-16}$). First the norm of the residual matrix Y for computed solution $\bar{X}$ of the Sylvester equation is

$$\|Y\|_F = \|A_{12} - A_{11}\bar{X} + \bar{X}A_{22}\|_F = 4.0272 \times 10^{-12},$$

which almost reaches the estimated bound (16) of Y:

$$\varepsilon_M(\|A_{11}\|_F + \|A_{22}\|_F)\|X\|_F = 8.8830 \times 10^{-12}.$$

Furthermore, the observed norm of $(2,1)$ block $\tilde{A}_{21}$ after swapping:

$$\|\bar{\tilde{A}}_{21}\|_2 = 1.2973 \times 10^{-12}.$$

which is also roughly attained to the bound (26) of $\|E_{21}\|_2$:

$$\|E_{21}\|_2 \le \frac{1}{1+\sigma_2^2(X)}\|Y\|_F = 2.0237 \times 10^{-12},$$

Note that for this example, the algorithm is still backward stable, since

$$\|\bar{\tilde{A}}_{21}\|_2 = 1.2973 \times 10^{-12} \le \varepsilon_M\|A\|_F = 4.4189 \times 10^{-12}.$$

After setting $\bar{\tilde{A}}_{21} = 0$, then the measures of backward stability are $\eta_E = 2.2922$ and $E_A = 1.8205$.

¿From Remark 1 after Theorem 2, we might worry that a huge $\|X\|_F$ or tiny $\mathrm{sep}(A_{11}, A_{22})$ could cause numerical instability. However the following example illustrates how in practice a small separation of A_{11} and A_{22} does not necessarily lead to instability. Let

$$A_{11} = \begin{bmatrix} 1 & -10^{-6} \\ 1 & 1 \end{bmatrix}, \qquad A_{22} = A_{11} + \sqrt{\varepsilon_M} I,$$

then the separation of A_{11} and A_{22} is tiny; that is $\mathrm{sep}(A_{11}, A_{22}) = 2.9802 \times 10^{-14}$. Let A_{12} be chosen such that $\mathrm{col}(A_{12})$ is the left singular vector of K corresponding to the smallest singular value $\sigma_{\min}(K)$, so that the norm of the solution X of the Sylvester equation $A_{11}X - XA_{22} = A_{12}$ reaches its upper bound (28), that is

$$\|X\|_F = \frac{\|A_{12}\|_F}{\mathrm{sep}(A_{11}, A_{22})} = 3.3554 \times 10^{13}$$

and $\mathrm{cond}(X) = 10^6$. Hence the estimated bound of the norm of residual Y is

$$\varepsilon\,(\|A_{11}\|_F + \|A_{22}\|_F)\|X\|_F = 2.5810 \times 10^{-2}.$$

However in practice, the observed residual norm $\|Y\|_F = 3.7253 \times 10^{-9}$. After swapping, it turns out that

$$\|\tilde{A}_{21}\|_F = 7.3985 \times 10^{-24} \ll \varepsilon_M\|A\|_F = 5.8747 \times 10^{-16}.$$

So the swapping is perfectly stable!

6 Conclusions

In this paper, we have developed a direct swapping algorithm which reorders the eigenvalues on the diagonal of a matrix in real Schur form by performing an orthogonal similarity transformation. A complete set of FORTRAN subroutines has been developed and included in the LAPACK library [2]. The algorithm is guaranteed to be numerically stable because we explicitly test for instability and do not reorder the eigenvalues if this would be unstable; this can only happen if the eigenvalues are so close as to be indistinguishable. Unfortunately there is no proof of the backward stability of the algorithm without this explicit test, even though we have not seen an example where instability could occur. The detailed error analysis and numerical examples show how well it deals with ill-conditioned cases, whereas the alternative stable algorithm EXCHNG may occasionally fail to converge.

Acknowledgement

The authors would like to thank K. C. Ng and B. Parlett for sharing their programs during our initial work on the subject. The valuable comments of G. W. Stewart and B. Parlett during the development of software are gratefully acknowledged. The authors are also indebted to A. Edelman and N. Higham for their valuable comments on the subject.

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. Mckenney, S. Ostrouchov and D. Sorensen, *LAPACK Users' Guide, Release 1.0*, SIAM, 1992 (to appear)

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. Mckenney, and D. Sorensen, *LAPACK: A Portable Linear Algebra Library for High-Performance Computers*, Supercomputing'90 (J. Martin, ed.) ACM Press, New York, pp. 2-10, 1990.

[3] Z. Bai, J. Demmel and A. Mckenney, *On the conditioning of the nonsymmetric eigenproblem*, LAPACK working note 13, University of Tennessee, CS-89-86, 1989.

[4] Z. Bai and G. W. Stewart, *SRRIT: A Fortran subroutine to calculate the dominant invariant subspace of a nonsymmetric matrix*, to submit to ACM TOMS, 1991.

[5] S. Batterson, *Convergence of the QR algorithm on 3x3 normal matrices*, Num. Math. 58:341–352, 1990.

[6] C. Bavely and G. W. Stewart, *An algorithm for computing reducing subspaces by block diagonalization*, SIAM J. Numer. Anal., 16:359-367(1979).

[7] R. S. Bartels and G. W. Stewart, *Solution of the matrix equation $AX + XB = C$*, Comm. ACM 15:820-826(1972).

[8] Z. Cao and F. Zhang, *Direct methods for ordering eigenvalues of a real matrix*, Chinese University J. of Comp. Math. No. 1, pp. 27-36 (1981), in Chinese.

[9] C. Davis and W. Kahan, *The Rotation of Eigenvectors by a Perturbation III*, SIAM J. Num. Anal. 7:1-46(1970)

[10] J. Dongarra, S. Hammarling and J. Wilkinson, Numerical considerations in computing invariant subspaces, LAPACK working note 25, University of Tennessee, CS-90-117, 1990.

[11] F. R. Gantmacher, *Theory of Matrices*, Vol. I, Chelsea, New York, 1959.

[12] G. Golub, S. Nash and C. Van Loan, *A Hessenberg-Schur Method for the Problem $AX + XB = C$*, IEEE Trans. Automat. Control AC-24: 909-913(1979).

[13] V. Mehrmann, *A Symplectic orthogonal method for single input or single output discrete time optimal control problems*, in Linear Algebra in Signals Systems and Control, B. N. Datta et al eds. SIAM Philadelphia, Penn. pp. 128-140(1988).

[14] K. C. Ng and B. N. Parlett, *Development of an accurate algorithm for $EXP(Bt)$, Part I, Programs to swap diagonal block, Part II*, CPAM 294, University of California, Berkeley. 1988.

[15] B. T. Smith et at, *Matrix Eigensystem Routines - EISPACK guide*, second edition. Lecture Notes in Computer Science, N119, Springer-Verlag, 1976

[16] G. W Stewart, *Simultaneous Iteration for Computing Invariant Subspaces of Non-Hermitian Matrices*, Numer. Math. 25, 12-56 (1976).

[17] G. W Stewart, *Algorithm 506 HQR3 and EXCHNG: Fortran subroutine for calculating and ordering the eigenvalues of a real upper Hessenberg matrix [F2]*, ACM TOMS 2: 275-280 (1976).

[18] G. W Stewart, *On the perturbation of pseudo-Inverse, projections, and linear least squares problems*, SIAM Review 19, 634-62 (1977).

[19] G. W Stewart, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Num Anal. 14, 509-18, (1977).

[20] G. W Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Inc. New Work, 1990

[21] P. Van Dooren, *Algorithm 590, DSUBSP and EXCHQZ: Fortran subroutines for computing deflating subspaces with specified spectrum*, ACM TOMS 8: 376-382 (1982).

[22] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford Univ. Press, 1965