Computing Eigenspaces with Specified Eigenvalues of a Regular Matrix Pair (A, B) and Condition Estimation: Theory, Algorithms and Software

Bo Kågström and Peter Poromaa*

April, 1994
Revised September, 1994
UMINF-94.04,ISSN-0348-0542
(Also published as LAPACK Working Note xx)

Abstract

Theory, algorithms and LAPACK–style software for computing a pair of deflating subspaces with specified eigenvalues of a regular matrix pair (A, B) and error bounds for computed quantities (eigenvalues and eigenspaces) are presented. The *reordering* of specified eigenvalues is performed with a direct orthogonal transformation method with guaranteed numerical stability. Each swap of two adjacent diagonal blocks in the real generalized Schur form, where at least one of them corresponds to a complex conjugate pair of eigenvalues, involves solving a generalized Sylvester equation and the construction of two orthogonal transformation matrices from certain eigenspaces associated with the diagonal blocks. The swapping of two 1×1 blocks is performed using orthogonal (unitary) Givens rotations. The *error bounds* are based on estimates of condition numbers for eigenvalues and eigenspaces. The software computes reciprocal values of a condition number for an individual eigenvalue (or a cluster of eigenvalues), a condition number for an eigenvector (or eigenspace), and spectral projectors onto a selected cluster. By computing reciprocal values we avoid overflow. Changes in eigenvectors and eigenspaces are measured by their change in angle. The condition numbers yield both *asymptotic* and *global* error bounds. The asymptotic bounds are only accurate for small perturbations (E, F) of (A, B), while the global bounds work for all $\|(E, F)\|$ up to a certain bound, whose size is determined by the conditioning of the problem. It is also shown how these upper bounds can be estimated. Fortran 77 *software* that implements our algorithms for reordering eigenvalues, computing (left and right) deflating subspaces with specified eigenvalues and condition number estimation are presented. Computational experiments that illustrate the accuracy, efficiency and reliability of our software are also described.

Key words. Regular matrix pair (pencil), generalized Schur form, direct reordering of eigenvalues, generalized Sylvester equation, eigenvectors, deflating subspaces, condition estimation, error bounds, blocked algorithms, LAPACK–style software.

AMS(MOS) subject classifications. Primary 65F15, 65F05, 65F35, 65Y20.

---

*Department of Computing Science, Umeå University, S-901 87 Umeå, Sweden, Electronic addresses: bokg@cs.umu.se and peterp@cs.umu.se

Contents

1 Introduction **4**
1.1 Notation . . . 5

2 Direct Method for Reordering Eigenvalues in the Generalized Real Schur Form **6**

3 Direct Reordering Algorithms with Guaranteed Backward Stability **9**
3.1 Justification for Rejecting a Swap . . . 11
3.2 Algorithm Variants for the Stability Tests . . . 11

4 Condition Numbers and Error Bounds for Eigenvalues and Eigenspaces of a Regular (A, B) **13**
4.1 A Condition Number and Error Bounds for Simple Eigenvalues . . . 14
4.2 Conditioning and Error Bounds for Left and Right Deflating Subspaces Associated with a Cluster of Eigenvalues . . . 16
4.2.1 Block–diagonalization and Separation of Two Matrix Pairs . . . 17
4.2.2 Conditioning of Left and Right Deflating Subspaces . . . 18
4.2.3 Upper Bound on Perturbations and Global Error Bounds for Deflating Subspaces and Clustered Eigenvalues . . . 19
4.2.4 Optimal Backward Perturbation of Approximate Left and Right Deflating Subspaces . . . 22
4.2.5 Residual–based Error Bound for Approximate Left and Right Deflating Subspaces . . . 22
4.3 Summary of Error Bounds for Eigenvalues and Eigenspaces . . . 23
4.4 Condition Estimates and Error Bounds Computed . . . 24
4.4.1 Estimating Dif_u and Dif_l . . . 25
4.4.2 Estimating Dif_l for Individual Eigenvectors Associated with a Complex Conjugate Pair of Eigenvalues . . . 27

5 Outline of the Software **28**
5.1 Reordering of Diagonal Blocks . . . 29
5.2 Computing Deflating Subspaces with Specified Eigenvalues . . . 29
5.3 Condition Estimation and Approximate Error Bounds . . . 30

6 Computational Experiments **30**
6.1 Accuracy and Reliability Results . . . 30
6.1.1 Test Problems . . . 31
6.1.2 Comparing Different Reordering Methods . . . 33
6.1.3 Results from Condition Estimation and Error Bounds . . . 36
6.2 A Summary of the Results from the Test Programs . . . 36
6.2.1 Test problems for `_CHK3` . . . 37
6.2.2 A Summary of the Results from `_CHK3` . . . 38
6.2.3 Test Problems and a Summary of the Results from `_CHK4` . . . 39

7 Some Conclusions **40**

A Calling sequence `DTGEXC` **49**

B Calling sequence `DTGEX2` **51**

C Calling sequence `DTGSEN` **54**

D Calling sequence `DTGSNA` **58**

E Calling sequence `DGSRBB` **61**

List of Tables

4.1 Asymptotic error bounds for the generalized eigenvalue problem 23
4.2 Global error bounds for the generalized eigenvalue problem 24
6.1 Problem characteristics, chordal distances and reciprocal condition numbers 32
6.2 Eigenvalues after reordering for problems 1, 6 and 11 34
6.3 Eigenvalues after reordering for problems 5, 10 and 15 35
6.4 Eigenvalues after reordering for problem 12 35
7.1 Computed errors after the reordering using Method 1 43
7.2 Computed errors after the reordering using Method 2 44
7.3 Method 2 – Computed stability test values and tolerances 45
7.4 Computed errors after the reordering using Method 3 46
7.5 Computed errors after the reordering using Algorithm 590 47
7.6 Method 2 – Some computed quantities before and after the reordering . . . 48

1 Introduction

Given a matrix pair (A, B), where A and B are general $n \times n$ matrices with real or complex entries. In the generalized eigenvalue problem (GEP) we are interested to find a few or all eigenvalues λ_i and eigenvectors $x_i \neq 0$ such that $Ax_i = \lambda_i B x_i$. Mathematically, the eigenvalues are the roots of the characteristic equation $\det(A - \lambda B) = 0$. If $B = I_n$, GEP reduces to the standard eigenvalue problem (SEP). Moreover, if $\det(B) \neq 0$, the problem can in theory be transformed to $Cx = \lambda x$, with $C = B^{-1}A$. If $\det(B) = 0$ and $x \neq 0$ is a null vector of B, then $Bx = 0Ax$, i.e. x is an eigenvector of the reciprocal problem $Bx = \mu Ax$, with $\mu = \lambda^{-1} = 0$. In other words, $\lambda = \infty$ is an eigenvalue of $Ax = \lambda Bx$. By restricting the matrix pencil $A - \lambda B$ (a family of matrices parameterized by λ) such that $\det(A - \lambda B) = 0$ if and only if λ is an eigenvalue, (A, B) is a regular matrix pair, or similarly, $A - \lambda B$ is a regular matrix pencil. If $\det(A - \lambda B) \equiv 0$ for all λ, $A - \lambda B$ is a singular pencil and (A, B) is a singular matrix pair [12].

From a computational point of view it is more natural to define GEP in cross–product form $\beta Ax = \alpha Bx$ with $\lambda = \alpha/\beta$. An eigenvalue is represented as a pair (α, β), where a finite eigenvalue has $\beta \neq 0$ and an infinite eigenvalue has $\beta = 0$. In this representation an infinite eigenvalue $(\alpha, 0)$ is not essentially different from a zero eigenvalue $(0, \beta)$. As in SEP we both have right and left eigenvectors $x \neq 0$ and $y \neq 0$, respectively, defined as

$$\beta Ax = \alpha Bx, \quad \beta y^H A = \alpha y^H B. \tag{1.1}$$

In several applications (e.g., in control theory [22], [29], [20]) it is not necessary to compute the eigenvectors, but merely to find eigenspaces associated with a specified set of eigenvalues. $\mathcal{L}$ and $\mathcal{R}$ are a pair (left and right) of deflating subspaces of a regular $A - \lambda B$, if $\mathcal{L} = A\mathcal{R} + B\mathcal{R}$ and $\dim\mathcal{L} = \dim\mathcal{R}$ [25, 26]. One way of computing a pair of deflating subspaces (with orthogonal bases) is via the generalized Schur decomposition. As in the standard eigenvalue problem we distinguish two cases.

The *complex* case: let A and B be $n \times n$ with complex entries. Then there exist unitary $U \in \mathbf{C}^{n\times n}$ and $V \in \mathbf{C}^{n\times n}$:

$$U^H(A - \lambda B)V = S - \lambda T, \tag{1.2}$$

where S and T are upper triangular. The eigenvalues are given by the pairs $(s_{ii}, t_{ii}) \neq (0, 0)$. The finite eigenvalues are s_{ii}/t_{ii}, where $t_{ii} \neq 0$. If $(s_{ii}, t_{ii}) = (0, 0)$ for some i, then (A, B) is singular.

The *real* case: let A and B be $n \times n$ with real entries. Then there exist orthogonal $U \in \mathbf{R}^{n\times n}$ and $V \in \mathbf{R}^{n\times n}$:

$$U^T(A - \lambda B)V = S - \lambda T, \tag{1.3}$$

where S is upper quasi–triangular and T is upper triangular. A quasi–triangular matrix is block upper triangular with 1×1 or 2×2 blocks on the diagonal. The 2×2 blocks on the diagonal of $S - \lambda T$ correspond to pairs of complex conjugate eigenvalues.

The columns of U and V, u_i and $v_i, i = 1 : n$, are left and right generalized Schur vectors and the first k columns of U and V span a k–dimensional pair of deflating subspaces associated with the $k \times k$ $(1,1)$–block of (S, T) in generalized Schur form. U and V can be chosen so that the eigenvalues appear in any order along the (block) diagonals of S and T.

More formally, let $U = [U_1, U_2]$ and $V = [V_1, V_2]$ be a conformal partitioning with respect to the cluster of k eigenvalues in the $(1,1)$–block of (S,T):

$$\begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} (A - \lambda B) \begin{bmatrix} V_1 & V_2 \end{bmatrix} = S - \lambda T \equiv \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} - \lambda \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}. \tag{1.4}$$

Now, $\mathcal{L} = \mathrm{span}(U_1)$ and $\mathcal{R} = \mathrm{span}(V_1)$ form a pair of deflating subspaces associated with the cluster of (S_{11}, T_{11}). Moreover, $\mathrm{span}(V_1)$ is a right eigenspace of (A,B), and $\mathrm{span}(U_2)$ is a left eigenspace of (A,B) associated with (S_{22}, T_{22}) [26]. Indeed, we can retrieve a left eigenspace associated with (S_{11}, T_{11}) and a right eigenspace associated with (S_{22}, T_{22}) by a second reordering of the eigenvalues of (S,T) such that, now, the "new" (S_{22}, T_{22}) will correspond to the specified cluster. Moreover, by combining the U_i's and V_i's from the two reorderings it is possible to construct a block–diagonalizing equivalence transformation [20]. Let

$$X^{-1} = \begin{bmatrix} U_2^{(2)^T} \\ U_2^{(1)^T} \end{bmatrix}, \quad Y = \begin{bmatrix} V_1^{(1)} & V_1^{(2)} \end{bmatrix}, \tag{1.5}$$

where $U_i^{(j)}$ and $V_i^{(j)}$ are blocks $i(=1,2)$ of U and V from the reordering $j(=1,2)$. Then $X^{-1}(A - \lambda B)Y$ is block–diagonal with the specified cluster in the $(1,1)$–block [20]. By construction the two block columns of Y and the two block rows of X^{-1} have orthonormal bases which ensure transformation matrices with optimal condition numbers [4]. Alternatively, we can block–diagonalize (S,T) in (1.4) in terms of a non–orthogonal equivalence transformation directly by solving a generalized Sylvester equation (see Section 4.2.1).

In this paper we present underlying theory, algorithms and LAPACK–style software for computing a pair of deflating subspaces with specified eigenvalues of a regular matrix pair (A,B) and error bounds for computed quantities (eigenvalues and eigenspaces). The error bounds are based on estimates of condition numbers for eigenvalues and eigenspaces. Typically, the algorithm for computing a pair of deflating subspaces is a two–step process. First, compute a generalized Schur form of a matrix pair (A,B) using the QZ algorithm [23]. Second, reorder the specified eigenvalues to appear in the $(1,1)$–block of the generalized Schur form. The focus here is to perform the reordering (also in the real case) with a direct method [18]. A reordering method based on the periodic Schur decomposition has been proposed recently [5]. The rest of the paper is outlined as follows. In Section 1.1 we collect our notation. Section 2 gives an overview of the direct method for reordering eigenvalues in the generalized real Schur form. In Section 3 we discuss direct reordering algorithms with guaranteed backward stability. Section 4 collects theory and algorithms for computing condition numbers and error bounds for eigenvalues and deflating subspaces of a regular (A,B). In Section 5 we present our Fortran 77 software for computing deflating subspaces with specified eigenvalues, condition numbers and error bounds. Some computational experiments that illustrate the accuracy and reliability of our software are presented in section 6. Finally, some conclusions are summarized in Section 7.

1.1 Notation

The following notation is used in the paper. I_n denotes an identity matrix of size $n \times n$. $\lambda(A,B)$ denotes the spectrum of a regular matrix pair (A,B) or pencil $A - \lambda B$. $\|A\|_2$

denotes the spectral norm (2-norm) of a matrix A induced by the Euclidean vector norm. $\|A\|_F$ denotes the Frobenius (or Euclidean) matrix norm. $\|A\|_M = \max_{i,j} |a_{ij}|$, i.e. the maximum of the absolute values of the matrix entries. $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ denote the largest and smallest singular values of A, respectively. For a square matrix A we have that $\|A\|_2 = \sigma_{\max}(A)$ and $\|A^{-1}\|_2 = \sigma_{\min}(A)^{-1}$. $A \otimes B$ denotes the Kronecker product of two matrices A and B whose (i,j)-th block element is $a_{ij}B$. The column vector $\mathrm{col}(A)$ denotes an ordered stack of the columns of A from left to right starting with the first column. A^T denotes the transpose of A. A^H denotes the conjugate transpose of A. $|A|$ and $|x|$ denote the matrix and the vector whose elements are $|a_{ij}|$ and $|x_i|$, respectively. Inequalities such as $|A| \leq |B|, |x| \leq |y|$ are interpreted componentwise. $D = \mathrm{diag}(x)$ denotes a diagonal matrix with $d_{ii} = x_i$.

We frequently measure distances between subspaces as their angular distances. $\theta(x, y)$ is the (acute) angle between two 1-dimensional subspaces spanned by the vectors x and y:

$$\cos\theta(x, y) = \frac{|x^T y|}{\|x\|_2 \|y\|_2}.$$

Generalized to the (maximum) angle between two subspaces $\mathcal{X}$ and $\mathcal{Y}$ of equal dimension $k \geq 2$ we have:

$$\theta_{\max}(\mathcal{X}, \mathcal{Y}) = \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \theta(x, y).$$

For computational purposes we use the following definition [13]:

$$\theta_{\max}(\mathcal{X}, \mathcal{Y}) = \arccos \sigma_{\min}(X^T Y),$$

where the columns of X and Y (of size $n \times k$) span orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$, respectively.

2 Direct Method for Reordering Eigenvalues in the Generalized Real Schur Form

A direct reordering method for the $A - \lambda B$ problem is presented in [18], which extends and generalizes the direct SEP method in [2] to regular matrix pairs with real entries. Below we give an overview of the direct method and its numerical properties.

Without loss of generality we consider the problem of reordering the diagonal blocks of a matrix pair (A, B) in the block form,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}, \tag{2.1}$$

where (A_{11}, B_{11}) and (A_{22}, B_{22}) are of size $n_1 \times n_1$ and $n_2 \times n_2$, respectively, and $n_1, n_2 = 1$ or 2. Throughout the paper we assume that (A_{11}, B_{11}) and (A_{22}, B_{22}) have no eigenvalues in common, otherwise, the diagonal blocks need not be swapped.

We want to find orthonormal Q and Z of size $(n_1 + n_2) \times (n_1 + n_2)$ such that

$$Q^T \left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \right) Z = \left(\begin{bmatrix} \hat{A}_{22} & \hat{A}_{12} \\ 0 & \hat{A}_{11} \end{bmatrix}, \begin{bmatrix} \hat{B}_{22} & \hat{B}_{12} \\ 0 & \hat{B}_{11} \end{bmatrix} \right) \equiv (\hat{A}, \hat{B}), \tag{2.2}$$

where (A_{ii}, B_{ii}) and $(\hat{A}_{ii}, \hat{B}_{ii})$ for $i = 1, 2$ are equivalent matrix pairs with the same eigenvalues but their positions are exchanged (swapped) along the block diagonal of (A, B).

The direct method for constructing Q and Z and swapping two diagonal blocks in the generalized real Schur form of (A, B) is outlined below [18]:

- Solve for L and R of size $n_1 \times n_2$ in the *generalized Sylvester equation*:

$$\begin{aligned} A_{11}R - LA_{22} &= -A_{12}, \\ B_{11}R - LB_{22} &= -B_{12}. \end{aligned} \tag{2.3}$$

- Compute an orthogonal matrix Q:

$$Q^T \begin{bmatrix} L \\ I_{n_2} \end{bmatrix} = \begin{bmatrix} T_L \\ 0 \end{bmatrix}. \tag{2.4}$$

- Compute an orthogonal matrix Z:

$$\begin{bmatrix} I_{n_1} & -R \end{bmatrix} Z = \begin{bmatrix} 0 & T_R \end{bmatrix}. \tag{2.5}$$

- Apply Q and Z to (A, B) in an orthogonal equivalence transformation (2.2):

$$\left(\begin{bmatrix} \hat{A}_{22} & \hat{A}_{12} \\ 0 & \hat{A}_{11} \end{bmatrix}, \begin{bmatrix} \hat{B}_{22} & \hat{B}_{12} \\ 0 & \hat{B}_{11} \end{bmatrix} \right) \equiv$$

$$\left(\begin{bmatrix} T_L A_{22} Z_{21} & T_L A_{22} Z_{22} + Q_{11}^T A_{11} T_R \\ 0 & Q_{12}^T A_{11} T_R \end{bmatrix}, \begin{bmatrix} T_L B_{22} Z_{21} & T_L B_{22} Z_{22} + Q_{11}^T B_{11} T_R \\ 0 & Q_{12}^T B_{11} T_R \end{bmatrix} \right), \tag{2.6}$$

  where Q and Z are partitioned conformally with

$$X = \begin{bmatrix} L & I_{n_1} \\ I_{n_2} & 0 \end{bmatrix}, \quad \text{and} \quad Y = \begin{bmatrix} 0 & I_{n_2} \\ I_{n_1} & -R \end{bmatrix}, \quad \text{respectively.}$$

Notice that X and Y above are non–orthogonal transformation matrices that perform the required swapping:

$$\left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \right) = X \left(\begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix}, \begin{bmatrix} B_{22} & 0 \\ 0 & B_{11} \end{bmatrix} \right) Y. \tag{2.7}$$

To solve (2.3) we can use the generalized Schur method [21, 19]. In our case, $(A_{ii}, B_{ii}), i = 1, 2$ are already in generalized Schur form and we end up solving a $2n_1n_2 \times 2n_1n_2$ linear system $Zx = b$, where

$$Z = \begin{bmatrix} I_{n_2} \otimes A_{11} & -A_{22}^T \otimes I_{n_1} \\ I_{n_2} \otimes B_{11} & -B_{22}^T \otimes I_{n_1} \end{bmatrix}, \quad x = \begin{bmatrix} \mathrm{col}(R) \\ \mathrm{col}(L) \end{bmatrix}, \quad b = \begin{bmatrix} -\mathrm{col}(A_{12}) \\ -\mathrm{col}(B_{12}) \end{bmatrix}. \tag{2.8}$$

Since $n_1, n_2 = 1$ or 2 the linear system (2.8) will be of size $2 \times 2, 4 \times 4$ or 8×8 (only 2×2 systems in the complex case). Q in (2.4) and Z in (2.5) can be found by using Householder or Givens transformations to compute a QR factorization and an RQ factorization, respectively. Finally, the equivalence transformation (2.6) is just matrix–matrix multiplication and add operations on A and B.

In the presence of rounding errors the conditioning and the solution of the generalized Sylvester equation will have the greatest impact on the stability of the direct swapping method [17]. Let $(\bar{L}, \bar{R})$ denote the computed solution of the generalized Sylvester equation (2.3), where $\bar{L} = L + \Delta L$, $\bar{R} = R + \Delta R$ and (L, R) is the exact solution. The residuals of the computed solution are

$$\begin{aligned} R_1 &\equiv A_{11}\bar{R} - \bar{L}A_{22} + A_{12}, \\ R_2 &\equiv B_{11}\bar{R} - \bar{L}B_{22} + B_{12}. \end{aligned} \tag{2.9}$$

Moreover, let $\bar{Q}$ and $\bar{Z}$ be the computed transformation matrices in (2.4) and (2.5). The following theorem shows how the errors in these quantities propagate to the results of the direct reordering method for swapping two 2×2 diagonal matrix pairs.

Theorem 2.1 *[18] By applying the computed transformation matrices $\bar{Q}$ and $\bar{Z}$ in an equivalence transformation of (A, B) we get*

$$\bar{Q}^T A \bar{Z} = \begin{bmatrix} \hat{A}_{22} & \hat{A}_{12} \\ 0 & \hat{A}_{11} \end{bmatrix} + \begin{bmatrix} \Delta A_{22} & \Delta A_{12} \\ \Delta A_{21} & \Delta A_{11} \end{bmatrix} \equiv \hat{A} + \Delta A \tag{2.10}$$

and

$$\bar{Q}^T B \bar{Z} = \begin{bmatrix} \hat{B}_{22} & \hat{B}_{12} \\ 0 & \hat{B}_{11} \end{bmatrix} + \begin{bmatrix} \Delta B_{22} & \Delta B_{12} \\ \Delta B_{21} & \Delta B_{11} \end{bmatrix} \equiv \hat{B} + \Delta B, \tag{2.11}$$

where (A_{ii}, B_{ii}) and $(\hat{A}_{ii}, \hat{B}_{ii})$ for $i = 1, 2$ are equivalent matrix pairs as in (2.2) and up to first order perturbations $O(\|(\Delta A, \Delta B)\|_2)$:

$$\|\Delta A_{11}\|_2 \le \frac{1}{(1 + \sigma_{\min}{}^2(L))^{1/2}} \cdot \frac{\sigma_{\max}(R)}{(1 + \sigma_{\max}{}^2(R))^{1/2}} \cdot \|R_1\|_F, \tag{2.12}$$

$$\|\Delta A_{21}\|_2 \le \frac{1}{(1 + \sigma_{\min}{}^2(L))^{1/2}} \cdot \frac{1}{(1 + \sigma_{\min}{}^2(R))^{1/2}} \cdot \|R_1\|_F, \tag{2.13}$$

$$\|\Delta A_{22}\|_2 \le \frac{\sigma_{\max}(L)}{(1 + \sigma_{\max}{}^2(L))^{1/2}} \cdot \frac{1}{(1 + \sigma_{\min}{}^2(R))^{1/2}} \cdot \|R_1\|_F. \tag{2.14}$$

Similar bounds hold for $\|\Delta B_{11}\|_2, \|\Delta B_{21}\|_2, \|\Delta B_{22}\|_2$ with R_1 replaced by R_2.

What can we say about the size of the errors $(\Delta A, \Delta B)$ in Theorem 2.1? First of all, $\|\Delta A_{ij}\|_2$ and $\|\Delta B_{ij}\|_2$ depend on $\|R_1\|_F, \|R_2\|_F$, the norms of the residuals of the computed solution $(\bar{L}, \bar{R})$ to the generalized Sylvester equation, and on the conditioning of the exact solution (L, R). If $\sigma_{\min}(L)$ and $\sigma_{\min}(R)$ are small, the error can be as large as the norms of the residuals.

In [17] a perturbation analysis of the generalized Sylvester equation is presented that takes full account to the structure of the matrix equation, derives expressions for the backward error of an approximate solution $(\bar{L}, \bar{R})$, and derives condition numbers that measure the sensitivity of a solution to perturbations in A_{11}, A_{12}, A_{22} and B_{11}, B_{12}, B_{22}, respectively. Due to the special structure of the (generalized) Sylvester equation the relation for linear systems "relative backward error = relative residual" [24] does not hold in general [16], [17]. Small relative backward errors will always result in small relative residuals. However, the analysis shows that for very ill–conditioned cases the norm of the relative backward errors can greatly exceed the norm of the relative residuals (in fact, by an arbitrary factor [16]). This situation appears when $(\bar{L}, \bar{R})$ is an ill–conditioned (i.e., $\sigma_{\min}(\bar{L})$ and $\sigma_{\min}(\bar{R})$ small) and large–normed (i.e., $\|(\bar{L}, \bar{R})\|_F$ large) solution to the generalized Sylvester equation. However, as we will see later, these situations correspond to extremely ill–conditioned eigenproblems.

3 Direct Reordering Algorithms with Guaranteed Backward Stability

The error analysis of the direct method (summarized in Theorem 2.1) and numerical experiments suggest that a practical implementation should reject a swap if it would result in too large backward errors (i.e. instability). The test for stability can be performed directly and to a small extra cost. A direct algorithm with controlled backward error for swapping two diagonal blocks of a regular matrix pair (A, B) in generalized real Schur form is outlined below [18]. In this section all quantities denote "computed" quantities.

Direct Swapping Algorithm

Step 1 Copy A and B to S and T, respectively.

$$S \leftarrow A, \quad T \leftarrow B.$$

Step 2 Solve for (L, R) in the generalized Sylvester equation:

$$\begin{aligned} S_{11}R - LS_{22} &= -S_{12}\gamma, \\ T_{11}R - LT_{22} &= -T_{12}\gamma. \end{aligned}$$

Use Gaussian elimination with complete pivoting to solve the corresponding linear system and a scaling factor γ to prevent against overflow [19].

Step 3 Compute an orthogonal matrix Q:

$$Q^T \begin{bmatrix} L \\ \gamma I_{n_2} \end{bmatrix} = \begin{bmatrix} T_L \\ 0 \end{bmatrix}.$$

Use Householder transformations to compute a QR factorization [13], [1].

Step 4 Compute an orthogonal matrix Z:

$$\begin{bmatrix} \gamma I_{n_1} & -R \end{bmatrix} Z = \begin{bmatrix} 0 & T_R \end{bmatrix}.$$

Use Householder transformations to compute an RQ factorization [13], [1].

Step 5 Perform the swapping tentatively with backward stability test:

$$S \leftarrow Q^T S Z \equiv \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \qquad T \leftarrow Q^T T Z \equiv \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}.$$

Step 6 If the swap is accepted, apply the equivalence transformation to (A, B):

$$A \leftarrow S, \quad B \leftarrow T.$$

Set the $(2,1)$–blocks to zero.

Step 7 Standardize existing 2×2 blocks.

Use the LAPACK routine `_HGEQZ` to standardize and (possibly) separate 2×2 blocks further [1].

The backward stability test in step 5 is split in two parts:

Weak stability test: Check if $\|(S_{21}, T_{21})\|_F \leq$ *tol1*.

Strong stability test: Check if $\|(A - QSZ^T, B - QTZ^T)\|_F \leq$ *tol2*.

The size of the $(2,1)$-blocks are most crucial since their norms immediately reflect the stability of the swapping. Indeed, $\|(S_{21}, T_{21})\|_F$ is the size of the optimal backward perturbation (E, F) of the reordered (S, T) such that $(S + E, T + F)$ has the n_2 first columns of Q and Z as an exact pair of deflating subspaces (see Section 4.2.4 and [28]). The size of the $(2,1)$-blocks are controlled by *tol1* in the weak stability test and they should not exceed $O(\epsilon\|(S,T)\|_F)$, where ϵ is the relative machine precision. The strong stability test takes all backward errors into account and by choosing *tol2* of the size $O(\epsilon\|(A,B)\|_F)$ and rejecting the swap if the error is larger than *tol2*, we obtain guaranteed backward stability. If both the weak and the strong stability tests hold then the swap is accepted, otherwise rejected. We have used $10\epsilon\|(A,B)\|_F$ for both *tol1* and *tol2*.

One could argue that it is enough with the weak stability test and in fact we have (so far) not been able to construct any example where the strong test fails while the weak test does not. However, since the extra cost of the strong stability test is only marginal it is included in our software.

After step 2 it would be possible to compute an optimal block–diagonalizing equivalence transformation that minimizes the condition numbers of the transformation matrices [6],[20]. Since the scaling factors (which possibly are large numbers) will show up in the S_{ij} and T_{ij} blocks, we do not expect any substantial improvements in performing this block–diagonal scaling. Computational experiments in Matlab confirm this statement too.

The swapping of a 2×2 block and a 1×1 block (or vice versa) is performed similarly as swapping two 2×2 blocks. However, the swapping of two 1×1 blocks is performed using orthogonal (unitary) Givens rotations [30]. In the complex case we perform all reordering with unitary Givens rotations.

3.1 Justification for Rejecting a Swap

It is well–known that the generalized eigenvalue problem (as well as the standard unsymmetric problem) is potentially ill–conditioned in the sense that eigenvalues and eigenspaces may change drastically even under small perturbations of the data (e.g., see [26], [9]). If we insist on performing a reordering of (S_{11}, T_{11}) and (S_{22}, T_{22}) for an ill–conditioned problem, we may destroy any spectral information in $A - \lambda B$. Our computational experiments show that close eigenvalues or small separation between (S_{11}, T_{11}) and (S_{22}, T_{22}) are not enough for rejecting a swap. It is the sensitivity of the eigenspaces that matters most, which in turn is perfectly signaled by the norm of the solutions L and R to the associated generalized Sylvester equation. As before, we illustrate with the case $n_1 = n_2 = 2$. From (2.7) we get that after the reordering

$$\mathcal{L}_1 = \mathrm{span}(\begin{bmatrix} I_2 \\ 0 \end{bmatrix}), \quad \mathcal{R}_1 = \mathrm{span}(\begin{bmatrix} I_2 \\ 0 \end{bmatrix}),$$

and

$$\mathcal{L}_2 = \mathrm{span}(\begin{bmatrix} L \\ I_2 \end{bmatrix}), \quad \mathcal{R}_2 = \mathrm{span}(\begin{bmatrix} R \\ I_2 \end{bmatrix}),$$

are pairs of deflating subspaces associated with the spectrum of the reordered blocks (S_{11}, T_{11}) and (S_{22}, T_{22}), respectively. Informally, we see that when $\|L\|$ and $\|R\|$ are large enough $\mathcal{L}_i$ for $i = 1, 2$ and $\mathcal{R}_i$ for $i = 1, 2$ are almost linearly dependent. More formally; by partitioning Q in (2.4) appropriately and using properties of the CS decomposition of Q, we can show that

$$\cos\theta_{\max}(\mathcal{L}_1, \mathcal{L}_2) = \|Q_{11}\|_2 = \frac{\sigma_{\max}(L)}{(1 + \sigma_{\max}{}^2(L))^{1/2}}.$$

Now, $\theta_{\max}(\mathcal{L}_1, \mathcal{L}_2)$ is close to zero if and only if $\|L\|_2$ is large. A small largest angle between $\mathcal{L}_1$ and $\mathcal{L}_2$, means that the left deflating subspaces associated with (S_{11}, T_{11}) and (S_{22}, T_{22}) are almost linearly dependent. Similarly, it can be shown that $\|R\|_2$ is large if and only if $\theta_{\max}(\mathcal{R}_1, \mathcal{R}_2)$ is close to zero.

3.2 Algorithm Variants for the Stability Tests

In the following we present some different variants to perform the stability tests for accepting or rejecting a swap of two 2×2 diagonal matrix pairs.

Method 1: Perform the weak and strong stability tests on (S, T) as computed in step 5. Here S_{21} and T_{21} are full 2×2 blocks with possibly non–zero entries. If the swap is accepted then set S_{21} and T_{21} to zero.

Method 2: Triangularize T (from step 5) with an orthogonal U from the left and apply the transformation to S, i.e.

$$S \leftarrow U^T S \equiv \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \qquad T \leftarrow U^T T \equiv \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}.$$

Now T is upper triangular and T_{21} is a zero 2×2 block, while S_{21} is still a full matrix block. We also have the freedom to triangularize T (from step 5) with an orthogonal V from the right and apply the transformation to S similarly. The triangularization method (from left or right) that produces the $(2,1)$–block of S with smallest (Frobenius) norm is chosen and checked for stability. If the swap is accepted, then S_{21} is set to zero.

Method 3: Transform (S,T) (from step 5) to generalized Hessenberg form:

$$(S,T) \leftarrow Q_H^T(S,T)Z_H \equiv \left(\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \right),$$

where S_{21} only has one non–zero element, namely in its $(1,2)$ position. Notice that we cannot guarantee that this element is small, even if the original S_{21} is small. We apply the QZ algorithm to (S,T) in generalized Hessenberg form giving

$$(S,T) \leftarrow Q_{QZ}^T(S,T)Z_{QZ} \equiv \left(\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \right).$$

If the strong stability test holds, then the swap is accepted, otherwise rejected.

Methods 1–2 are direct methods and they differ in the way we transform the matrix pair (S,T) before we apply the stability tests. Our aim is to discard as little information as possible in the reordered matrix pair when we impose the $(2,1)$–blocks to be zero. Method 1 is the generic variant and we discard information in both S and T. In method 2 we only discard information in the S–part. Let $\tilde{S}_{ij}$ and $\tilde{T}_{ij}$ denote the (i,j)–blocks of S and T before the triangularization of T and $U_{ij}(V_{ij})$ the corresponding blocks of the orthogonal (unitary) transformation U (or V) used to triangularize from left (or right). The triangularization of T from left results in $T_{21}=0$ and $S_{21} = U_{22}^T(\tilde{S}_{21} - \tilde{T}_{21}\tilde{T}_{11}^{-1}\tilde{S}_{11})$, with

$$\|S_{21}\|_2 \leq \|\tilde{S}_{21} - \tilde{T}_{21}\tilde{T}_{11}^{-1}\tilde{S}_{11}\|_F. \tag{3.1}$$

Similarly, the triangularization from right results in $T_{21}=0$ and $S_{21} = (\tilde{S}_{21} - \tilde{S}_{22}\tilde{T}_{22}^{-1}\tilde{T}_{21})V_{11}^T$, with

$$\|S_{21}\|_2 \leq \|\tilde{S}_{21} - \tilde{S}_{22}\tilde{T}_{22}^{-1}\tilde{T}_{21}\|_F. \tag{3.2}$$

Roughly speaking, if the bound (3.1) is smaller than the bound (3.2), then triangularization from the left is to prefer, otherwise from the right. We have computational evidence that the exact expressions for $\|S_{21}\|$ in (3.1) and (3.2) (and the bounds) can be smaller than the information we discard in method 1 (see bounds for ΔA_{21} and ΔB_{21} in Theorem 2.1).

The expressions for S_{21} above can be traced back to expressions involving blocks of S and T *before* the reordering (i.e. the original A_{ij} and B_{ij} (2.1)) and the residuals of the generalized Sylvester equation in step 2 ($R_1 = A_{11}R - LA_{22} + \gamma A_{12}$ and $R_2 = B_{11}R - LB_{22} + \gamma B_{12}$), resulting in the following bounds on the block S_{21} of S *after* the triangularization in method 2:

Corollary 3.1 (i) *Let $S \leftarrow U^TS$, where U is the exact orthogonal transformation that triangularizes T from the left. Then (up to first order perturbations)*

$$\|\Delta A_{21}\|_2 \equiv \|S_{21}\|_2 \leq \frac{\|R_1 - R_2 B_{22}{}^{-1}A_{22}\|_F}{(1+\sigma_{\min}{}^2(L))^{1/2}(1+\sigma_{\min}^2(R))^{1/2}}. \tag{3.3}$$

(ii) *Similarly, let $S \leftarrow SV$, where V is the exact orthogonal transformation that triangularizes T from the right. Then (up to first order perturbations)*

$$\|\Delta A_{21}\|_2 \equiv \|S_{21}\|_2 \leq \frac{\|R_1 - A_{11}B_{11}{}^{-1}R_2\|_F}{(1+\sigma_{\min}^2(L))^{1/2}(1+\sigma_{\min}^2(R))^{1/2}}. \tag{3.4}$$

In both cases $\Delta B_{21} \equiv T_{21} = 0$.

Proof From the proof of Theorem 2.1 [18], we have (up to first order perturbations) $\tilde{S}_{21} = Q_{12}^T R_1 Z_{21}$ and $\tilde{T}_{21} = Q_{12}^T R_2 Z_{21}$, where Q_{ij} and Z_{ij} are blocks of Q and Z in the orthogonal equivalence transformation (2.2) that performs the reordering of two diagonal blocks. Moreover, $\|Z_{21}\|_2 = 1/(1+\sigma_{\min}^2(R))^{1/2}$ and $\|Q_{12}^T\|_2 = 1/(1+\sigma_{\min}^2(L))^{1/2}$ [18]. From (2.6) we get $\tilde{S}_{11} = T_L A_{22} Z_{21}$ and $\tilde{T}_{11} = T_L B_{22} Z_{21}$. Using these expressions in $S_{21} = U_{22}^T(\tilde{S}_{21} - \tilde{T}_{21}\tilde{T}_{11}^{-1}\tilde{S}_{11})$ we get $S_{21} = U_{22}^T Q_{12}^T (R_1 - R_2 B_{22}{}^{-1} A_{22}) Z_{21}$, and (3.3) follows from well–known inequalities for matrix norms. The bound (3.4) can be proved similarly. $\square$

Besides backward stability of the reordering we strive to affect the eigenvalues as little as possible. We have constructed examples where the triangularization (from left or right) of S has great impact on the reordered eigenvalues.

In method 3 we let the QZ algorithm decide what information to discard, and hopefully this should give us accurate eigenvalues after the swapping of two diagonal blocks. However, applying the QZ algorithm also means that the method may be iterative in the last step. In the worst case, already ordered eigenvalues may be reordered again. We use this variant mainly for comparing results between our direct methods and a “best possible” hybrid method.

4 Condition Numbers and Error Bounds for Eigenvalues and Eigenspaces of a Regular (A, B)

A condition number of a problem measures the sensitivity of the solution to small changes in the problem data. The problem is ill–conditioned if the condition number is large, and ill–posed if the condition number is infinite. Condition numbers can be used to bound errors in computed quantities (e.g., eigenvalues, eigenvectors and deflating subspaces). We construct error bounds from forward perturbation bounds for the problem (that define our condition numbers) and the knowledge of the backward error corresponding to the computed solution. The best we can ask for is to have an explicit expression for the optimal backward error related to the residual of the computed solution. A residual–based expression of the optimal backward error is algorithm–independent, i.e. it can be applied to an approximate solution resulting from any algorithm used to solve the problem. Otherwise, we have to rely to an upper bound on the backward error, resulting from a backward error analysis of the algorithm used to compute the solution. If we use backward stable methods to compute an eigendecomposition of a regular (A, B), then we know that the norm of the backward error in $(A+E, B+F)$ is $\|(E, F)\| = O(\epsilon\|(A, B)\|)$, where ϵ is the relative machine accuracy.

Condition numbers may be very expensive to compute and therefore we will use inexpensive estimates. In the extreme case the exact condition number (separation between two matrix pairs) is an $O(n^6)$ operation while the estimate is computed in only $O(n^3)$ operations [19]. Condition estimators are by definition approximations or bounds to the exact values they try to estimate, and may therefore occasionally overestimate or underestimate the true condition number by a large factor. Extensive computational experiments (on moderately sized problems) show that this seldom happens, but it is of course always possible to construct counter examples.

The condition numbers and estimates computed by our software and discussed here are reciprocal values of a condition number for an individual eigenvalue (or a cluster of eigenvalues), a condition number for an eigenvector (or eigenspace), and spectral projectors onto a selected cluster. By computing reciprocal values we avoid overflow. An infinite value or a condition number that would overflow are reported by the reciprocal value zero. This is in agreement with the condition estimation for the standard eigenvalue problem in LAPACK [3].

These quantities appear in error bounds for eigenvalues and eigenspaces of a regular (A, B), which we also review here. In agreement with the standard eigenvalue problem in LAPACK we measure changes in eigenvectors or eigenspaces by their change in angle. Moreover, our condition numbers yield both *asymptotic* and *global* error bounds. The asymptotic bounds are only accurate for small perturbations (E, F) of (A, B), while the global bounds work for all $\|(E, F)\|$ up to a certain bound. The size of this bound is determined by the conditioning of the problem and may therefore be large (for a well–conditioned problem) or small (for an ill–conditioned problem). We also show how these upper bounds can be estimated. Finally, we present some new results (due to Sun [28]) that give us explicit expressions for the optimal backward error related to the residual of a computed eigenspace, and which also lead to an residual–based global angular error bound for computed left and right deflating subspaces.

4.1 A Condition Number and Error Bounds for Simple Eigenvalues

Assume that $(\alpha, \beta) \neq (0, 0)$ is a simple eigenvalue of a regular matrix pair (A, B) with left and right eigenvectors y and x, respectively, satisfying (1.1). Notice that all non–zero scalar multiples of (α, β) is also an eigenvalue of (A, B). Therefore, it is natural to regard the subspace spanned by the vector $(\alpha, \beta)^T$ as the *generalized eigenvalue* of (A, B) [26]:

$$< \alpha, \beta >= \{\tau(\alpha, \beta)^T : \tau \in \mathbf{C}, (\alpha, \beta) \neq (0, 0)\}. \tag{4.1}$$

In the perturbation theory for generalized eigenvalues we consider the distance between pairs (α, β) and (α', β'). A useful metric is the *chordal distance* of two pairs defined as

$$\mathcal{X}((\alpha, \beta), (\alpha', \beta')) = \frac{|\alpha\beta' - \beta\alpha'|}{\sqrt{|\alpha|^2 + |\beta|^2}\sqrt{|\alpha'|^2 + |\beta'|^2}}. \tag{4.2}$$

If we set $\lambda = \alpha/\beta$ and $\lambda' = \alpha'/\beta'$, then we have

$$\mathcal{X}(\lambda, \lambda') = \frac{|\lambda - \lambda'|}{\sqrt{1 + |\lambda|^2}\sqrt{1 + |\lambda'|^2}}.$$

Some characteristics of the chordal metric are summarized below [26]: The point at infinity is no more than unit distance from any other point ($\mathcal{X}(\lambda, \infty) = 1/\sqrt{1 + |\lambda|^2} \leq 1$). If $|\lambda|, |\lambda'| \leq$

1, then $\mathcal{X}(\lambda, \lambda')$ behaves essentially like the Euclidean metric. The chordal distance between two large numbers can be small (e.g., $\mathcal{X}(\lambda, 2\lambda) \approx 1/|\lambda|$, when $\lambda \to \infty$). Accordingly, large numbers can have very small chordal distances, even when they have large relative errors.

In the following we review how to measure the sensitivity of simple eigenvalues of a regular matrix pair. Let (α, β) be a simple eigenvalue of (A, B) with left and right eigenvectors y and x, respectively. Let (E, F) be a perturbation of (A, B), and $\|(E, F)\|_2 = \epsilon_2$. Then there is an eigenvalue (α', β') of $(A+E, B+F)$ such that the following first order bound holds [26]:

$$\mathcal{X}((\alpha, \beta), (\alpha', \beta')) \leq \nu \cdot \epsilon_2 + O({\epsilon_2}^2), \tag{4.3}$$

where ν is the *condition number* for a *simple generalized eigenvalue*:

$$\nu = \frac{\|x\|_2 \|y\|_2}{\sqrt{|y^H A x|^2 + |y^H B x|^2}}. \tag{4.4}$$

Notice that $y^H Ax / y^H Bx$ is equal to $\lambda = \alpha/\beta$. By using $y^H Ax$ and $y^H Bx$ in (4.4), the condition number is independent of the normalization of the eigenvectors and the corresponding eigenvalue pair. Deleting higher order terms in (4.3) we get an asymptotic error bound for a simple eigenvalue:

$$\mathcal{X}((\alpha, \beta), (\alpha', \beta')) \lesssim \nu \|(E, F)\|_2. \tag{4.5}$$

By replacing $\|(E, F)\|_2$ by $\|(E, F)\|_F$ in (4.5) we get a somewhat weaker but a more computationally attractive bound.

The following example illustrates the definition of ν. Let

$$Y^H = \begin{bmatrix} 1 & 2\delta^{-1} \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & \delta^{-1} \\ 0 & 1 \end{bmatrix}, \quad \text{and}$$

$$Y^H A X = \begin{bmatrix} \epsilon & 0 \\ 0 & -2\epsilon \end{bmatrix}, \quad Y^H B X = \begin{bmatrix} \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}, \quad \text{i.e.}$$

$$A = \begin{bmatrix} \epsilon & 3\epsilon\delta^{-1} \\ 0 & -2\epsilon \end{bmatrix}, \quad B = \begin{bmatrix} \epsilon & -3\epsilon\delta^{-1} \\ 0 & \epsilon \end{bmatrix}.$$

By choosing $\epsilon > 0$ and $\delta > 0$ small we get $\nu_i = O(\epsilon^{-1}\delta^{-1})$ for $i = 1, 2$ from (4.4) and X and Y^H above.

The eigenvalues of (A, B) are 1 $(= \epsilon/\epsilon)$ and -2 $(= -2\epsilon/\epsilon)$. Now, we consider the equivalent pencil

$$(D^{-1}A, D^{-1}B) = \left(\begin{bmatrix} 1 & 3\delta^{-1} \\ 0 & -2 \end{bmatrix}, \begin{bmatrix} 1 & 3\delta^{-1} \\ 0 & 1 \end{bmatrix} \right),$$

where

$$D = \begin{bmatrix} \epsilon^{-1} & 0 \\ 0 & \epsilon^{-1} \end{bmatrix}.$$

It has the same eigenvalues (and eigenvectors too!), but their individual condition numbers are only $\nu_i = O(\delta^{-1})$ for $\epsilon \approx \delta > 0$, showing that the eigenvalues 1 and -2 are better conditioned for the equivalent pencil.

There also exist global error bounds which are not limited by the size of $\|(E,F)\|$ (e.g., Bauer–Fike–style error bounds). In the following we assume that (A,B) is a diagonalizable matrix pair such that

$$Y^H(A,B)X = (\mathrm{diag}(\alpha_i), \mathrm{diag}(\beta_i)). \tag{4.6}$$

Let Y^H and X be normalized such that $|\alpha_i|^2 + |\beta_i|^2 = 1$ and $\|y_i\|_2 = 1$, i.e. we overwrite y_i with $y_i/\|y_i\|_2$, (α_i, β_i) with $(\alpha_i, \beta_i)/(|\alpha_i|^2 + |\beta_i|^2)^{1/2}$ and, finally, x_i with $x_i\|y_i\|_2/(|\alpha_i|^2 + |\beta_i|^2)^{1/2}$. With this normalization the individual condition number for (α_i, β_i) is $\nu_i = \|x_i\|_2$.

Let (α', β') with $|\alpha'|^2 + |\beta'|^2 = 1$ be an eigenvalue of $(A+E, B+F)$. Then we have the following Bauer–Fike–style bound [8]:

$$\min_i \mathcal{X}((\alpha', \beta'), (\alpha_i, \beta_i)) \equiv \min_i |\alpha_i\beta' - \beta_i\alpha'| \leq \|Y^H\|_2\|X\|_2\|(E,F)\|_F. \tag{4.7}$$

If X and Y are normalized as above, then $\|Y^H\|_2 \leq \sqrt{n}$ and $\|X\|_2 \leq \sqrt{n}\max_i \nu_i$, which in (4.7) give us a Bauer–Fike–style bound for the generalized eigenvalue problem:

$$\min_i \mathcal{X}((\alpha', \beta'), (\alpha_i, \beta_i)) \leq n \max_i \nu_i \|(E,F)\|_F. \tag{4.8}$$

As for the standard eigenvalue problem, it is the most ill–conditioned eigenvalue that determines the size of the error bound. In words, (4.7) and (4.8) bound the smallest distance (measured in the chordal metric) between an eigenvalue of the perturbed and unperturbed matrix pairs.

It is possible to strengthen the classical Bauer–Fike bound (for the standard problem) giving a bound for each individual eigenvalue [7, 3], whose size is determined by the conditioning of the individual eigenvalue. By applying the same technique (under the same assumptions as for (4.8)) we can prove that any eigenvalue (α', β') with $|\alpha'|^2 + |\beta'|^2 = 1$ of $(A+E, B+F)$ must lie in one of the regions ("balls")

$$\{(\alpha, \beta), |\alpha|^2 + |\beta|^2 = 1 : \mathcal{X}((\alpha, \beta), (\alpha_i, \beta_i)) \leq n\nu_i\|(E,F)\|_2\}. \tag{4.9}$$

Notice that the sizes of the regions (bounds) are only a factor n larger than the first order error bound (4.5). Moreover, the global error bound (4.9) with respect to an eigenvalue (α_i, β_i) is only useful if it defines a region that does not intersect with regions corresponding to other eigenvalues. If two or more regions intersect, then we can only say that an eigenvalue of the perturbed matrix pair lies in the union of the overlapping regions.

There also exist other Bauer–Fike–style bounds for the generalized eigenvalue problem and other generalizations (see [26] for a review and further references).

4.2 Conditioning and Error Bounds for Left and Right Deflating Subspaces Associated with a Cluster of Eigenvalues

If (A,B) has n distinct eigenvalues, then there exist non–singular matrices Y and X that transform (A,B) to diagonal form (4.6). Moreover, their columns y_i and x_i are left and right eigenvectors associated with the eigenvalues (α_i, β_i) $(i = 1 : n)$. Let $Y^H \equiv P^{-1}$ and from (4.6) we have that $AX + BX = P\mathrm{diag}(\alpha_i + \beta_i)$, i.e. the columns p_i and x_i of P and X, respectively, span pairs of one–dimensional left and right deflating subspaces. Accordingly, conditioning and error bounds for individual eigenvectors can be regarded as a special case of error bounds for left and right deflating subspaces.

Without loss of generality we assume that (A, B) is in generalized Schur form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}. \tag{4.10}$$

In the following we review condition numbers and error bounds for left and right deflating subspaces associated with the cluster of m $(1 \le m \le n-1)$ eigenvalues (counting multiplicities) of (A_{11}, B_{11}). To explain the bounds we need to introduce some definitions.

4.2.1 Block–diagonalization and Separation of Two Matrix Pairs

An equivalence transformation that block–diagonalizes (A, B) in generalized Schur form (4.10) can be expressed as

$$\begin{aligned} &\begin{bmatrix} I_m & -L \\ 0 & I_{n-m} \end{bmatrix} \left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} \right) \begin{bmatrix} I_m & R \\ 0 & I_{n-m} \end{bmatrix} = \\ &\qquad \left(\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \right). \end{aligned} \tag{4.11}$$

Solving for (L, R) in (4.11) is equivalent to solve the generalized Sylvester equation

$$\begin{aligned} A_{11}R - LA_{22} &= -A_{12}, \\ B_{11}R - LB_{22} &= -B_{12}, \end{aligned} \tag{4.12}$$

which can be rewritten as a $2m(n-m) \times 2m(n-m)$ linear system $Z_u x = b$, where

$$Z_u = \begin{bmatrix} I_{n-m} \otimes A_{11} & -A_{22}^T \otimes I_m \\ I_{n-m} \otimes B_{11} & -B_{22}^T \otimes I_m \end{bmatrix} \tag{4.13}$$

and

$$x = \begin{bmatrix} \mathrm{col}(R) \\ \mathrm{col}(L) \end{bmatrix}, \quad b = \begin{bmatrix} -\mathrm{col}(A_{12}) \\ -\mathrm{col}(B_{12}) \end{bmatrix}.$$

Moreover, let

$$p = (1 + \|L\|_F^2)^{1/2}, \quad q = (1 + \|R\|_F^2)^{1/2}. \tag{4.14}$$

In the perturbation theory for the generalized eigenvalue problem, p and q play the same role as the norm of the spectral projector does for the standard eigenvalue problem [9]. Indeed, if $B = I$, then $p = q$ and p equals the norm of the projection onto an invariant subspace of A. For the generalized eigenvalue problem we need both a left and a right projection norm since the left and right deflating subspaces are (normally) different.

Another important quantity involved in the sensitivity analysis of deflating subspaces (and eigenvalues) is the *separation of two matrix pairs* (A_{11}, B_{11}) and (A_{22}, B_{22}) [25]:

$$\mathrm{Dif}_u[(A_{11}, B_{11}), (A_{22}, B_{22})] = \inf_{\|(L,R)\|_F=1} \|(A_{11}R - LA_{22}, B_{11}R - LB_{22})\|_F. \qquad (4.15)$$

The generalized Sylvester operator $(A_{11}R - LA_{22}, B_{11}R - LB_{22})$ in the definition of Dif_u is obtained from block–diagonalizing a regular matrix pair in *upper* block triangular form. Dif_u is a generalization of the separation between two matrices ($\mathrm{Sep}(A_{11}, A_{22}) = \sigma_{\min}(I_{n-m} \otimes A_{11} - A_{22}^T \otimes I_m)$ [25]) to two matrix pairs and it measures the separation of their spectra in the following sense. If (A_{11}, B_{11}) and (A_{22}, B_{22}) have a common eigenvalue, then Dif_u is zero and it is small if there is a small perturbation of either (A_{11}, B_{11}) or (A_{22}, B_{22}) that makes them have a common eigenvalue.

From the matrix representation (4.13) of the generalized Sylvester operator it can be shown [9] that

$$\mathrm{Dif}_u[(A_{11}, B_{11}), (A_{22}, B_{22})] = \sigma_{\min}(Z_u). \qquad (4.16)$$

Moreover, it follows that the generalized Sylvester equation has a unique solution if and only if $\mathrm{Dif}_u > 0$ and we can bound the norm of (L, R) as

$$\|(L, R)\|_F \leq \frac{\|(A_{12}, B_{12})\|_F}{\mathrm{Dif}_u}. \qquad (4.17)$$

Notice that $\mathrm{Dif}_u[(A_{22}, B_{22}), (A_{11}, B_{11})]$ does not generally equal $\mathrm{Dif}_u[(A_{11}, B_{11}), (A_{22}, B_{22})]$ (unless A_{ii} and B_{ii} are symmetric for $i = 1, 2$). Accordingly, the ordering of the arguments plays a role for the separation of two matrix pairs, while it does not for the separation of two matrices ($\mathrm{Sep}(A_{11}, A_{22}) = \mathrm{Sep}(A_{22}, A_{11})$). Therefore, we introduce the notation

$$\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})] = \mathrm{Dif}_u[(A_{22}, B_{22}), (A_{11}, B_{11})]. \qquad (4.18)$$

An associated generalized Sylvester operator $(A_{22}R - LA_{11}, B_{22}R - LB_{11})$ in the definition of Dif_l is obtained from block–diagonalizing a regular matrix pair in *lower* block triangular form:

$$\begin{bmatrix} I_m & 0 \\ -L & I_{n-m} \end{bmatrix} \left(\begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix} \right) \begin{bmatrix} I_m & 0 \\ R & I_{n-m} \end{bmatrix} =$$

$$\left(\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \right).$$

4.2.2 Conditioning of Left and Right Deflating Subspaces

Assume that (A, B) is in generalized Schur form (4.10) and that (A_{11}, B_{11}) contains the cluster of m eigenvalues with left and right deflating subspaces $\mathcal{L}$ and $\mathcal{R}$, respectively. Typically, $\mathcal{L} = \mathrm{span}\{U_1\}$ and $\mathcal{R} = \mathrm{span}\{V_1\}$ where U_1 and V_1 are the leading m columns of the unitary (orthogonal) U and V in (1.2) that transform (A, B) to generalized Schur form. Furthermore, let $\mathcal{L}' = \mathrm{span}\{U'_1\}$ and $\mathcal{R}' = \mathrm{span}\{V'_1\}$ be left and right deflating subspaces of the perturbed matrix pair $(A+E, B+F)$. Then using the technique of Sun [27] one can prove the following first order bounds:

$$\sin\theta_{\max}(\mathcal{L}, \mathcal{L}') \leq \|U_1 - U'_1\|_F \leq \frac{\|(E,F)\|_F}{\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})]} + O(\|(E,F)\|_F^2),$$

$$\sin\theta_{\max}(\mathcal{R}, \mathcal{R}') \leq \|V_1 - V'_1\|_F \leq \frac{\|(E,F)\|_F}{\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})]} + O(\|(E,F)\|_F^2).$$

From the series expansion of the arcsine function we can simplify these bounds further, giving the asymptotic angular bounds

$$\theta_{\max}(\mathcal{L}, \mathcal{L}') \lesssim \frac{\|(E,F)\|_F}{\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})]}, \tag{4.19}$$

$$\theta_{\max}(\mathcal{R}, \mathcal{R}') \lesssim \frac{\|(E,F)\|_F}{\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})]}. \tag{4.20}$$

These bounds imply that Dif_l is the reciprocal of the condition number for eigenvectors $(m = 1)$ and deflating subspaces $(m > 1)$ of a regular (A, B).

4.2.3 Upper Bound on Perturbations and Global Error Bounds for Deflating Subspaces and Clustered Eigenvalues

In order to guarantee that the clusters in the $(1,1)$–blocks of (A, B) and the perturbed matrix pair $(A + E, B + F)$ are of the same size m and uniquely defined, we have to put restrictions on $\|(E,F)\|_F$ [9]:

$$\|(E,F)\|_F \leq \frac{\min(\mathrm{Dif}_u, \mathrm{Dif}_l)}{(p^2+q^2)^{1/2} + 2\max(p,q)}.$$

By imposing a somewhat stronger condition on $\|(E,F)\|_F$, namely

$$\|(E,F)\|_F \leq \frac{\min(\mathrm{Dif}_u, \mathrm{Dif}_l)}{4\max(p,q)} \equiv \Delta, \tag{4.21}$$

Δ in (4.21) conform to the corresponding restriction for the standard eigenvalue problem $(\mathrm{Sep}(A_{11}, A_{22})/4p)$ [3].

We see that Δ may be small if the separation between the two matrix pairs is small or the left and right projection norms are large, indicating that the (deflating subspace) problem is ill–conditioned. A larger $\|(E,F)\|_F$ $(> \Delta)$ may imply that one eigenvalue in the cluster moves and coalesces with another eigenvalue (outside the cluster). Indeed, Δ is a lower bound on the smallest $\|(E,F)\|_F$ such that an eigenvalue of (A_{11}, B_{11}) coalesces with an eigenvalue of (A_{22}, B_{22}) under perturbation (E, F). The bound Δ can be quite conservative but is almost exact in some cases and a good estimate in many others. In particular, the following global error bounds for a pair of deflating subspaces is guaranteed valid for $\|(E,F)\|_F \leq \Delta$ [9].

As before, we assume that (A, B) is in generalized Schur form (4.10) and that (A_{11}, B_{11}) contains the cluster of m eigenvalues with left and right deflating subspaces $\mathcal{L}$ and $\mathcal{R}$, respectively. Further, let $\mathcal{L}'$ and $\mathcal{R}'$ be left and right deflating subspaces of $(A+E, B+F)$. Then we have the following angle bounds for left and right deflating subspaces of the unperturbed and perturbed matrix pairs [9]:

If $\delta \equiv \|(E,F)\|_F/\Delta < 1$, then

$$\theta_{\max}(\mathcal{L},\mathcal{L}') \leq \arctan\left(\frac{\delta}{p-\delta(p^2-1)^{1/2}}\right), \tag{4.22}$$

$$\theta_{\max}(\mathcal{R},\mathcal{R}') \leq \arctan\left(\frac{\delta}{q-\delta(q^2-1)^{1/2}}\right). \tag{4.23}$$

In other words, if δ is small, then the perturbed pair of left and right deflating subspaces are small perturbations of the exact pair of deflating subspaces.

The bounds (4.22), (4.23) are generalized and extended to pairs of reducing subspaces for singular (A, B) [9, 10, 11].

We are also interested to bound the error in the average of the eigenvalues of the cluster in (A_{11}, B_{11}). However, since we are faced with both finite and infinite eigenvalues it is not clear how to define the average of the m eigenvalues $\lambda_i = \alpha_i/\beta_i$. Only if we require that (A_{11}, B_{11}) contains a (proper) subset of the finite eigenvalues or *all* infinite eigenvalues (and no finite eigenvalues) does the average of the cluster make sense. In the following theorem we distinguish these two cases and formulate error bounds for the average of the clustered eigenvalues.

Theorem 4.1 *Let (A_{11}, B_{11}) denote the block of the generalized Schur form of (A, B) that correspond to the unperturbed cluster of eigenvalues, with*

$$\mathrm{Dif}_x[(A_{11},B_{11}),(A_{22},B_{22})] > 0 \quad \text{for} \quad x = l, u.$$

Similarly, let $({A'}_{11}, {B'}_{11})$ denote the corresponding block with perturbed eigenvalues associated with $(A+E, B+F)$, where $\|(E,F)\|_F \leq \Delta$ (4.21).

Case 1. *(A_{11}, B_{11}) contains a (proper) subset of the finite eigenvalues, i.e. B_{11} is non–singular. Let $\bar{\lambda}$ denote the average of the m unperturbed eigenvalues of (A_{11}, B_{11}) and let $\bar{\lambda}'$ be the corresponding average of the perturbed eigenvalues of $({A'}_{11}, {B'}_{11})$. Then*

$$|\bar{\lambda} - \bar{\lambda}'| \leq \frac{1}{\sigma_{\min}(B_{11})}\left(1 + \frac{\sigma_{\max}({A'}_{11})}{\sigma_{\min}({B'}_{11})}\right) 3p\|(E,F)\|_F. \tag{4.24}$$

Case 2. *(A_{11}, B_{11}) contains all infinite eigenvalues and no finite eigenvalues, i.e. A_{11} is non–singular. Let $\bar{\mu} = 0$ denote the average of the m unperturbed eigenvalues of the reciprocal problem (B_{11}, A_{11}) and, similarly, let $\bar{\mu}'$ be the corresponding average of the perturbed eigenvalues of $({B'}_{11}, {A'}_{11})$. Then*

$$|\bar{\mu} - \bar{\mu}'| \leq \frac{1}{\sigma_{\min}(A_{11})}\left(1 + \frac{\sigma_{\max}({B'}_{11})}{\sigma_{\min}({A'}_{11})}\right) 3p\|(E,F)\|_F. \tag{4.25}$$

Proof In general, we can bound the average $\bar{\lambda}$ of the eigenvalues of an $m \times m$ matrix C as

$$|\bar{\lambda}| \leq \frac{1}{m}\sum_{i=1}^{m} |\lambda_i| \leq \max_i |\lambda_i| \leq \|C\|_2 \leq \|C\|_F.$$

In case 1 we have

$$|\bar{\lambda} - \bar{\lambda}'| \leq \|B_{11}^{-1} A_{11} - {B'}_{11}^{-1} {A'}_{11}\|_F \leq \|B_{11}^{-1}\|_2 \|A_{11} - {A'}_{11}\|_F + \|B_{11}^{-1} - {B'}_{11}^{-1}\|_F \|{A'}_{11}\|_2.$$

Since $B_{11}^{-1} - {B'}_{11}^{-1} = -{B'}_{11}^{-1}(B_{11} - {B'}_{11})B_{11}^{-1}$ we have that

$$|\bar{\lambda} - \bar{\lambda}'| \leq \|B_{11}^{-1}\|_2 (\|A_{11} - {A'}_{11}\|_F + \|B_{11} - {B'}_{11}\|_F \|{B'}_{11}^{-1}\|_2 \|{A'}_{11}\|_2). \tag{4.26}$$

With the assumptions in the theorem we can apply the technique of Stewart [25] and prove that

$$\|A_{11} - {A'}_{11}\|_F \leq 3p\|(E, F)\|_F, \quad \|B_{11} - {B'}_{11}\|_F \leq 3p\|(E, F)\|_F,$$

where p (4.14) is the norm of the left spectral projector.

Using these bounds in (4.26) give us the error bound (4.24).

In case 2 we have

$$|\bar{\mu} - \bar{\mu'}| \leq \|A_{11}^{-1} B_{11} - {A'}_{11}^{-1} {B'}_{11}\|_F,$$

and the error bound (4.25) can be proved similarly. □

In the following we make some comments to the error bounds in Theorem 4.1. If B_{11} (in case 1) and A_{11} (in case 2) are well–conditioned with respect to inversion (i.e. $\sigma_{\min}(B_{11}) \gg 0$ and $\sigma_{\min}(A_{11}) \gg 0$, respectively) then the error bounds can be expressed as

$$|\bar{\lambda} - \bar{\lambda}'| \leq c_1 p\|(E, F)\|_F, \quad |\bar{\mu} - \bar{\mu'}| \leq c_2 p\|(E, F)\|_F,$$

where c_1 and c_2 are modest constants. These bounds conform with the corresponding error bound for the average of a cluster of eigenvalues to the standard eigenvalue problem ($2p\|E\|_F$). However, if $\sigma_{\min}(B_{11})$ (and $\sigma_{\min}({B'}_{11})$) are small in case 1, or if $\sigma_{\min}(A_{11})$ (and $\sigma_{\min}({A'}_{11})$) are small in case 2, then the average of the clustered eigenvalues can be quite sensitive to perturbations in (A, B), which is signaled by the quantities

$$\frac{1}{\sigma_{\min}(B_{11})}\left(1 + \frac{\sigma_{\max}({A'}_{11})}{\sigma_{\min}({B'}_{11})}\right),$$

and

$$\frac{1}{\sigma_{\min}(A_{11})}\left(1 + \frac{\sigma_{\max}({B'}_{11})}{\sigma_{\min}({A'}_{11})}\right),$$

respectively, in the bounds (4.24) and (4.25). In case 1 this means that (A_{11}, B_{11}) is nearby a pencil with an infinite eigenvalue and in case 2 (A_{11}, B_{11}) is nearby a singular pencil. Both cases represent ill–conditioned clustering problems. One way to tackle the most ill–conditioned cases with multiple infinite eigenvalues and an almost singular A_{11} is to compute and separate the Jordan structure of the infinite eigenvalue before any clustering takes place [10, 11]. It is well–known that the QZ algorithm applied to defective infinite eigenvalues can affect otherwise well-conditioned eigenvalues of (A, B) [32]. By separating the infinite structure from the rest of the spectrum before applying the QZ algorithm we circumvent this problem.

4.2.4 Optimal Backward Perturbation of Approximate Left and Right Deflating Subspaces

Suppose that $\bar{\mathcal{L}} = \mathrm{span}(\bar{V}_1)$ and $\bar{\mathcal{R}} = \mathrm{span}(\bar{U}_1)$, with $\bar{U}_1{}^H\bar{U}_1 = \bar{V}_1{}^H\bar{V}_1 = I_m$ are approximate left and right deflating subspaces of (A, B). We are interested to find (backward) perturbations of (A, B) such that the perturbed matrix pair has $\bar{\mathcal{L}}$ and $\bar{\mathcal{R}}$ as exact left and right deflating subspaces. Let

$$\mathcal{H} \equiv \{H = (E, F), \quad E, F \in \mathbf{C}^{n\times n} : (A + E)\bar{\mathcal{R}} \subseteq \bar{\mathcal{L}}, (B + F)\bar{\mathcal{R}} \subseteq \bar{\mathcal{L}}\},$$

i.e. $\mathcal{H}$ defines the set of perturbations $H = (E, F)$ such that $(A + E, B + F)$ has $\bar{\mathcal{L}}$ and $\bar{\mathcal{R}}$ as exact left and right deflating subspaces. Moreover, let the right residuals with respect to the approximate deflating subspaces be

$$R_{\mathrm{res}} \equiv (R_{\mathrm{Ares}}, R_{\mathrm{Bres}}) = (A\bar{U}_1 - \bar{V}_1\bar{A}_{11}, B\bar{U}_1 - \bar{V}_1\bar{B}_{11}),$$

where $\bar{A}_{11} = \bar{V}_1{}^H A\bar{U}_1$ and $\bar{B}_{11} = \bar{V}_1{}^H B\bar{U}_1$. Then there exist a unique optimal backward perturbation $H_{\mathrm{opt}} = -R_{\mathrm{res}}\bar{U}_1{}^H \in \mathcal{H}$ [28]:

$$\|H_{\mathrm{opt}}\| = \min_{H\in\mathcal{H}} \|H\| = \|R_{\mathrm{res}}\|, \tag{4.27}$$

for any unitarily invariant norm.

4.2.5 Residual–based Error Bound for Approximate Left and Right Deflating Subspaces

Assume that $\bar{U} = [\bar{U}_1, \bar{U}_2]$ and $\bar{V} = [\bar{V}_1, \bar{V}_2]$ are computed transformations that take (A, B) to generalized Schur form:

$$\bar{V}^H A\bar{U} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \qquad \bar{V}^H B\bar{U} = \begin{bmatrix} \bar{B}_{11} & \bar{B}_{12} \\ \bar{B}_{21} & \bar{B}_{22} \end{bmatrix},$$

where the entries of $\bar{A}_{21}$ and $\bar{B}_{21}$ are small and $\bar{U}, \bar{V}$ unitary (orthogonal) to machine precision accuracy.

Then $\bar{\mathcal{L}} = \mathrm{span}(\bar{V}_1)$ and $\bar{\mathcal{R}} = \mathrm{span}(\bar{U}_1)$ are approximate left and right deflating subspaces of (A, B) and we compute their left and right residuals:

$$L_{\mathrm{res}} \equiv (L_{\mathrm{Ares}}, L_{\mathrm{Bres}}) = (\bar{V}_1{}^H A - \bar{A}_{11}\bar{U}_1{}^H, \bar{V}_1{}^H B - \bar{B}_{11}\bar{U}_1{}^H),$$

$$R_{\mathrm{res}} \equiv (R_{\mathrm{Ares}}, R_{\mathrm{Bres}}) = (A\bar{U}_1 - \bar{V}_1\bar{A}_{11}, B\bar{U}_1 - \bar{V}_1\bar{B}_{11}).$$

It is straightforward to show that

$$\|L_{\mathrm{Ares}}\|_F = \|\bar{A}_{12}\|_F, \quad \|L_{\mathrm{Bres}}\|_F = \|\bar{B}_{12}\|_F, \quad \|R_{\mathrm{Ares}}\|_F = \|\bar{A}_{21}\|_F, \quad \|R_{\mathrm{Bres}}\|_F = \|\bar{B}_{21}\|_F. \tag{4.28}$$

We see that the norm of the right residuals are always small, while in general we cannot expect the norm of the left residuals to be small. However, knowing that there exist an optimal backward perturbation of approximate left and right deflating subspaces it is possible to derive a residual–based error bound. We can rewrite the residual–based error bound for deflating subspaces in [28] as the following angular error bounds:

If $\eta \equiv 4\|L_{\text{res}}\|_F \|R_{\text{res}}\|_F / \text{Dif}_l{}^2 < 1$, then

$$\theta_{\max}(\mathcal{L}, \bar{\mathcal{L}}) \leq \arctan\left(\frac{2\|R_{\text{res}}\|_F}{\text{Dif}_l}\right), \tag{4.29}$$

$$\theta_{\max}(\mathcal{R}, \bar{\mathcal{R}}) \leq \arctan\left(\frac{2\|R_{\text{res}}\|_F}{\text{Dif}_l}\right). \tag{4.30}$$

Notice that the bounds (4.29) and (4.30) are approximate in the sense that the theory assumes that $\bar{U}, \bar{V}$ are exactly unitary (orthogonal), while we can only guarantee that they are unitary (orthogonal) to machine precision accuracy.

From the definition of η we also get a bound on $\|(E,F)\|_F$ similar to (4.21) that guarantees that the residual–based bounds are valid for perturbations (E,F) fulfilling

$$\|(E,F)\|_F \equiv \|R_{\text{res}}\|_F \leq \frac{\text{Dif}_l{}^2}{4\|L_{\text{res}}\|_F} \equiv \Delta_r. \tag{4.31}$$

4.3 Summary of Error Bounds for Eigenvalues and Eigenspaces

In Table 4.1 and Table 4.2 we summarize the error bounds presented in Section 4 (see earlier subsections for definitions and notation used). Table 4.1 shows the asymptotic bounds for a simple eigenvalue (α, β) where $\lambda = \alpha/\beta$, the average of a cluster of eigenvalues $\bar{\lambda}$ (or $\bar{\mu} = 0$ for the infinite eigenvalues, i.e. $\bar{\lambda} = 1/\bar{\mu}$), a left and right eigenvector pair y and x, and a pair (left and right) of deflating subspaces $\mathcal{L}$ and $\mathcal{R}$.

Table 4.1: Asymptotic error bounds for the generalized eigenvalue problem

Bounds for	*Error bound*	*Comment*
Simple eigenvalue:	$\mathcal{X}((\alpha,\beta),(\alpha',\beta')) \lesssim \nu\|(E,F)\|_F$	$\lambda = \alpha/\beta$
Eigenvalue cluster:		
Average of finite λ_i	$\|\bar{\lambda} - \bar{\lambda}'\| \lesssim c_1 p\|(E,F)\|_F$	See also global bounds
Average of infinite $\lambda_i = 1/\mu_i$	$\|\bar{\mu} - \bar{\mu}'\| \lesssim c_2 p\|(E,F)\|_F$	(c_1 and c_2 are constants)
Eigenvector pair:		
Left	$\theta_{\max}(y, y') \lesssim \|(E,F)\|_F/\text{Dif}_l$	$m = 1$
Right	$\theta_{\max}(x, x') \lesssim \|(E,F)\|_F/\text{Dif}_l$	$m = 1$
Deflating subspace pair:		
Left	$\theta_{\max}(\mathcal{L}, \mathcal{L}') \lesssim \|(E,F)\|_F/\text{Dif}_l$	$1 < m \leq n-1$
Right	$\theta_{\max}(\mathcal{R}, \mathcal{R}') \lesssim \|(E,F)\|_F/\text{Dif}_l$	$1 < m \leq n-1$

The asymptotic bounds are only valid for sufficiently small perturbations. If the problem is ill–conditioned, the asymptotic bounds may only hold for extremely small values of $\|(E,F)\|$. Therefore, we also provide similar global error bounds (displayed in Table

4.2), which are valid for all perturbations that satisfy an upper bound on $\|(E, F)\|$. These restrictions are Δ (4.21) and Δ_r (4.31), where Δ_r is associated with residual–based error bounds. For ill–conditioned problems these restrictions will also be small. Indeed, a small value of Δ (or Δ_r) shows that the cluster of eigenvalues in the leading $m \times m$ blocks of (A, B) is ill–conditioned in the sense that small perturbations of (A, B) may imply that one eigenvalue in the cluster moves and coalesces with another eigenvalue (outside the cluster). Accordingly, this also means that the associated (left and right) deflating subspaces are sensitive for small perturbations, since the size of the perturbed subspaces may change for small perturbations of (A, B).

Table 4.2: Global error bounds for the generalized eigenvalue problem

Bounds for	*Error bound*	*Restriction on* $\|(E, F)\|_F$
Simple eigenvalue:	Bound (4.8)	None (holds for all (E, F))
$(\lambda = \alpha/\beta, \|\alpha\|^2 + \|\beta\|^2 = 1)$	Bound (4.9)	None (holds for all (E, F))
Eigenvalue cluster:		
Average of finite λ_i	Bound (4.24)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Average of infinite $\lambda_i = 1/\mu_i$	Bound (4.25)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Eigenvector pair:		
Left	Bound (4.22)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Right	Bound (4.23)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Left (*residual–based*)	Bound (4.29)	$\leq \Delta_r \equiv {\mathrm{Dif}_l}^2/4\|L_{\mathrm{res}}\|_F$
Right (*residual–based*)	Bound (4.30)	$\leq \Delta_r \equiv {\mathrm{Dif}_l}^2/4\|L_{\mathrm{res}}\|_F$
Deflating subspace pair:		
Left	Bound (4.22)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Right	Bound (4.23)	$\leq \Delta \equiv \min(\mathrm{Dif}_u, \mathrm{Dif}_l)/4\max(p, q)$
Left (*residual–based*)	Bound (4.29)	$\leq \Delta_r \equiv {\mathrm{Dif}_l}^2/4\|L_{\mathrm{res}}\|_F$
Right (*residual–based*)	Bound (4.30)	$\leq \Delta_r \equiv {\mathrm{Dif}_l}^2/4\|L_{\mathrm{res}}\|_F$

It is interesting to compare the sizes of Δ and Δ_r. We focus on ill–conditioned problems where the separation between the two clusters are small (i.e. both Dif_l and Dif_u are small) and the deflating subspaces are sensitive (i.e. the associated generalized Sylvester equation has large–normed solutions (L, R)). Since ${\mathrm{Dif}_l}^2$ appears in the nominator of Δ_r while we only have $\min(\mathrm{Dif}_u, \mathrm{Dif}_l)$ in the nominator of Δ it seems as if Δ_r puts harder restrictions on the perturbations. However, if we use the expressions (4.28) in Δ_r, the bound (4.17) on (L, R) to bound p and q giving that they are of size $O(\|(\bar{A}_{12}, \bar{B}_{12})\|_F/\mathrm{Dif}_u)$, then we see that Δ and Δ_r are qualitatively of the same size. It is of course possible to construct examples where Δ is smaller than Δ_r and vice versa.

4.4 Condition Estimates and Error Bounds Computed

Our software (described in more detail in Section 5) compute estimates of the following quantities that appear in the condition numbers and error bounds summarized in tables 4.1 and 4.2.

- $S(\lambda) = \nu^{-1}$, the reciprocal value of the condition number ν (4.4) for an individual eigenvalue $\lambda = \alpha/\beta$.

  Given the left and right eigenvectors y and x corresponding to λ, the reciprocal condition number is computed in $O(n^2)$ flops (floating point operations) as

$$S(\lambda) = \frac{\sqrt{|y^H Ax|^2 + |y^H Bx|^2}}{\|x\|_2 \|y\|_2}. \tag{4.32}$$

  If both α and β are zero, then (A, B) is singular and $S(\lambda) = -1$ is reported.

  The (left and right) eigenvectors computed by LAPACK are normalized such that the largest component will have the sum of the modulus of the real and imaginary parts equal to one (e.g., see _GEGV).

- Dif_u (4.16) and Dif_l (4.18), i.e. the separation(s) between two matrix pairs.

  Dif_l is a reciprocal condition number for an individual (left or right) eigenvector or a (left or right) deflating subspace (see Section 4.2.2). Both Dif_u and Dif_l appear in Δ. Our algorithms for estimating Dif_u and Dif_l are discussed in Section 4.4.1.

- p^{-1} and q^{-1}, the reciprocal values of the left and right projector norms as defined in (4.14).

  These values are computed straightforwardly using L and R from the generalized Sylvester equation (4.12). The cost for solving (4.12) is $2m^2(n-m) + 2m(n-m)^2$ flops, where $m \geq 1$ is the dimension of the deflating subspace(s) corresponding to the selected eigenvalues. Given L and R the cost for computing p^{-1} and q^{-1} is only $O(m(n-m))$ flops.

By using the estimates of these quantities in the error bounds, the user gets enough information for assessing the accuracy of computed eigenvalues (or the average of clustered eigenvalues), eigenvectors or deflating subspaces.

We also compute the algorithm–independent residual–based error bound(s) (4.29), (4.30) and η, the condition that guarantees the validity of the bound(s). The residual–based error bound and η are computed using a Frobenius normed–based estimate of Dif_l (see Section 4.4.1). The residuals (4.28) associated with the approximate deflating subspaces are computed straightforwardly.

4.4.1 Estimating Dif_u and Dif_l

LAPACK–style algorithms and software for estimating Dif_u (4.16) are presented in [19]. The basic problem is to find a lower bound on $\mathrm{Dif}_u^{-1}[(A_{11}, B_{11}), (A_{22}, B_{22})] \equiv \|Z_u^{-1}\|_2$, where Z_u is the matrix representation (4.13) of the generalized Sylvester operator. It is possible to compute lower bounds on Dif_u^{-1} by solving generalized Sylvester equations in triangular form. Both Frobenius norm–based and one–norm–based Dif_u–estimators are discussed and evaluated in [19]. The one–norm–based estimator makes the condition estimation uniform with other parts of LAPACK (e.g., the standard eigenvalue problem). The Frobenius norm-based estimator offers a low–cost and equally reliable estimator. The one–norm–based estimator is a factor 3-10 times more expensive.

By knowing a lower bound DIFINV on $\|Z_u^{-1}\|_2$ we also have an upper bound DIF = 1/DIFINV on the separation between two regular matrix pairs. Since we use blocked algorithms to solve the generalized Sylvester equations involved in computing DIFINV, our estimators will mainly execute Level 3 operations. In the following we outline the algorithms for the Dif_u–estimators. From the definition of Dif_l (4.18) we see that Dif_l–estimators can be computed by using our algorithms for estimating Dif_u. Our software provide (optionally) both Frobenius norm–based and one–normed–based estimators for Dif_u and Dif_l, respectively (see Section 5).

A Frobenius Norm-Based Estimator From the $Z_u x = b$ representation (4.13) of the generalized Sylvester equation (4.12) we get a lower bound on Dif_u^{-1}:

$$\|(L,R)\|_F / \|(C,F)\|_F = \|x\|_2 / \|b\|_2 \leq \|Z_u^{-1}\|_2. \tag{4.33}$$

To get an improved estimate we want to choose right hand sides (C^*, F^*) such that the associated solution (L^*, R^*) has as large norm as possible. Then the quantity

$$\phi_F \equiv \|(L^*, R^*)\|_F / \|(C^*, F^*)\|_F, \tag{4.34}$$

is our lower bound on $\|Z_u^{-1}\|_2$. The work to compute ϕ_F is comparable to solve a generalized Sylvester equation, which costs $O(m^3 + m^2(n-m) + m(n-m)^2 + (n-m)^3)$ flops (only $2m^2(n-m) + 2m(n-m)^2$ if the matrix pairs are in generalized Schur form) [21]. This is a very modest cost compared to compute the exact value of $\sigma_{\min}(Z_u)$, which requires $O(m^3(n-m)^3)$ flops.

Two Frobenius norm–based estimators _TDIFE and _TDIFD are discussed in [19], which are modifications of estimators BSOLVE and BSOLVD in [21]. The main differences concern how contributions to ϕ_F from different subsystems are computed and the look ahead strategies of the estimators. _TDIFE is the default Frobenius norm-based estimator in our software (see Section 5).

An One–Norm–Based Estimator From the relationship

$$\frac{1}{\sqrt{2m(n-m)}} \|Z_u^{-1}\|_1 \leq \|Z_u^{-1}\|_2 \leq \sqrt{2m(n-m)} \|Z_u^{-1}\|_1, \tag{4.35}$$

we know that $\|Z_u^{-1}\|_1$ can never differ more than a factor $\sqrt{2m(n-m)}$ from $\|Z_u^{-1}\|_2$. So it makes sense to compute an one–norm–based estimator of Dif_u^{-1}.

The LAPACK routine _LACON implements a method for estimating the one–norm of a square matrix, using reverse communication for evaluating matrix–vector products [14, 15]. We apply this method to $\|Z_u^{-1}\|_1$ by providing the solution vectors x and y of $Z_u x = z$ and a transposed system $Z_u^T y = z$, where z is determined by _LACON. In each step only one of these generalized Sylvester equations is solved using blocked algorithms [19]. The cost for computing this bound is roughly equal to the number of steps in the reverse communication times the cost for one generalized Sylvester solve.

Notice, $\|Z_u^{-1}\|_\infty$ also satisfy (4.35), i.e. can never differ more than a factor $\sqrt{2m(n-m)}$ from $\|Z_u^{-1}\|_2$. Moreover, since $\|B\|_\infty = \|B^T\|_1$ the same method can be used to compute an infinity–norm–based estimate of Dif_u^{-1}.

4.4.2 Estimating Dif_l for Individual Eigenvectors Associated with a Complex Conjugate Pair of Eigenvalues

The estimation of Dif_l discussed in the preceding subsection is applicable to pairs of deflating subspaces ($m \geq 2$) as well as to individual eigenvector pairs ($m = 1$). As before we assume that (A, B) is transformed to generalized Schur form

$$Q^H AZ = S \equiv \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad Q^H BZ = T \equiv \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}. \tag{4.36}$$

In complex arithmetic we can always choose (S_{11}, T_{11}) to be the individual eigenvalue $\lambda_1 = \alpha_1/\beta_1 = S_{11}/T_{11}$ (real or complex) we want to consider. Moreover, the first column of Q and Z form a pair of deflating subspaces, where q_1 also is a right eigenvector corresponding to λ_1. In this case we can, for example, apply the one–normed–based estimator to estimate Dif_l.

What is said above also applies in real arithmetic to real eigenvalues of (A, B). However, there is an extra complication to estimate Dif_l for the individual eigenvectors corresponding to a complex conjugate pair of eigenvalues. For a real matrix pair, Q and Z in (4.36) are orthogonal and (S_{11}, T_{11}) is a 2×2 matrix pair corresponding to the complex conjugate pair of eigenvalues λ_1 and $\bar{\lambda}_1$. It exits unitary U_1 and V_1 such that

$$U_1^H S_{11} V_1 = \begin{bmatrix} \alpha_1 & x \\ 0 & \alpha_2 \end{bmatrix}, \quad U_1^H T_{11} V_1 = \begin{bmatrix} \beta_1 & y \\ 0 & \beta_2 \end{bmatrix}, \tag{4.37}$$

where $\alpha_1 = \gamma_1 + i\mu_1$, $\alpha_2 = \gamma_2 + i\mu_2$ and β_1, β_2 are real numbers. Now, the complex conjugate pair of eigenvalues is given by $\lambda_1 = \alpha_1/\beta_1$ and $\bar{\lambda}_1 = \alpha_2/\beta_2$ (i.e. $\gamma_1/\beta_1 = \gamma_2/\beta_2$ and $\mu_1/\beta_1 = \mu_2/\beta_2$).

If we are interested to estimate Dif_l associated with λ_1, then we can choose

$$U = \begin{bmatrix} U_1 & 0 \\ 0 & I_{n-2} \end{bmatrix}, \quad V = \begin{bmatrix} V_1 & 0 \\ 0 & I_{n-2} \end{bmatrix},$$

and we get

$$U^H SV = \begin{bmatrix} \alpha_1 & S'_{12} \\ 0 & S'_{22} \end{bmatrix}, \quad U^H TV = \begin{bmatrix} \beta_1 & T'_{12} \\ 0 & T'_{22} \end{bmatrix}. \tag{4.38}$$

Notice that only (S'_{12}, T'_{12}) and the first row of (S'_{22}, T'_{22}) have complex entries. Moreover, $\bar{\lambda}_1 = \alpha_2/\beta_2$ belongs to the spectrum of (S'_{22}, T'_{22}).

From [9] we have the following explicit expression for Dif_l:

$$\mathrm{Dif}_l[(\alpha_1, \beta_1), (S'_{22}, T'_{22})] = \sigma_{\min}(Z_l), \tag{4.39}$$

where

$$Z_l = \begin{bmatrix} \alpha_1 \otimes I_{n-1} & -1 \otimes S'_{22} \\ \beta_1 \otimes I_{n-1} & -1 \otimes T'_{22} \end{bmatrix}, \tag{4.40}$$

and Z_l is a $2(n-1) \times 2(n-1)$ matrix. Dif_l associated with (α_1, β_1) and its conjugate (α_2, β_2) have the same value.

In contrary to the standard eigenvalue problem [3], there is no simple and inexpensive trick to stay in real arithmetic and perform a one–normed–based estimate of $\|Z_l{}^{-1}\|_2$. The LU factorization of Z_l may give L and U with all entries having non–zero imaginary parts. Accordingly, the cost for doing the estimation of Dif_l (in real arithmetic) is similar to the cost for doing it entirely in complex arithmetic. From the definition of Dif_l it is possible to show the following two inequalities:

$$\sigma_{\min}(Z_l) \leq \mathrm{Dif}_l[(\alpha_1, \beta_1), (\alpha_2, \beta_2)] = \sigma_{\min}\left(\begin{bmatrix} \alpha_1 & -\alpha_2 \\ \beta_1 & -\beta_2 \end{bmatrix}\right) \equiv d_1, \tag{4.41}$$

and

$$\sigma_{\min}(Z_l) \leq \max(1, \left|\frac{\gamma_1}{\gamma_2}\right|)\mathrm{Dif}_l[(S_{11}, T_{11}), (S_{22}, T_{22})] \equiv d_2, \tag{4.42}$$

where the S_{ii} and T_{ii} blocks are from the generalized real Schur form. Now, we choose $\min(d_1, d_2)$ as our estimate for $\mathrm{Dif}_l = \sigma_{\min}(Z_l)$. This estimate can be weaker than the estimate computed in complex arithmetic, but is normally a sufficiently good estimate. We report results for both the real and complex estimators in Section 6.1.3 for a selection of problems. If λ_1 and $\bar{\lambda}_1$ are close but well–separated from the rest of the spectrum, then d_1 is a good estimate of $\sigma_{\min}(Z_l)$. Whether d_2 is a good estimate to $\sigma_{\min}(Z_l)$ will mainly depend on the size of (S'_{12}, T'_{12}) in (4.38) (i.e. the "departure from block–diagonality" of the generalized Schur form).

5 Outline of the Software

Following the LAPACK conventions and standards [1], we have developed Fortran 77 routines that perform the following computations for a regular matrix pair (A, B)(in generalized Schur form):

- reorder eigenvalues (diagonal blocks) in the generalized Schur form (routines `_TGEXC` and `_TGEX2`),
- compute (left and right) deflating subspaces with specified eigenvalues (routine `_TGSEN`),
- estimate condition numbers for specified eigenvalues (or a cluster of eigenvalues) and associated eigenvectors or deflating subspaces (routines `_TGSNA`, `_TGSEN`), and compute residual–based approximate error bounds for a pair of deflating subspaces (routine `_GSRBB`).

Following the LAPACK conventions for naming, `_` in `_YYZZZ` stands for `S`(ingle), `D`(ouble), `C`(omplex) or `Z` (Double complex). Routines for all four data types are available. In the following, we describe these top–level computational routines in some detail, while the auxiliary routines are just mentioned briefly. The software uses LAPACK routines to compute machine dependent thresholds, generalized Schur forms of matrix pairs, eigenvalues and eigenvectors, matrix factorizations (QR and RQ), matrix norms, and to copy matrices, perform column– and row–swapping and so on. BLAS routines are used to perform basic linear algebra operations such as matrix–matrix (Level 3), matrix–vector (Level 2) and vector (Level 1) operations.

5.1 Reordering of Diagonal Blocks

_TGEXC reorders the diagonal blocks of (A, B) in generalized Schur form. The reordering is specified by the parameters IFST and ILST. The diagonal block with row index IFST is moved to row ILST by a sequence of transpositions of adjacent blocks. If IFST (on entry) pointed to the second row of a 2×2 block, it is changed to point to the first row. ILST always points to the first row of the block in its final position, which may differ from its input value by $+1$ or -1. Each swap in the reordering is performed with a call to _TGEX2 (see below), and optionally, the matrices of generalized Schur vectors Q and Z are updated with the orthogonal (unitary) equivalence transformations performed. INFO reports if the reordering was successful or if any swap was rejected due to ill-conditioning. The calling sequence and the leading comment lines of DTGEXC are listed in Appendix A.

_TGEX2 implements a direct algorithm with guaranteed backward stability for swapping two adjacent diagonal blocks (A_{11}, B_{11}) and (A_{22}, B_{22}) of a matrix pair (A, B) in generalized Schur form, where the diagonal blocks are of size $\mathtt{N1} \times \mathtt{N1}$ and $\mathtt{N2} \times \mathtt{N2}$, respectively. In the real case N1 and N2 are 1 or 2. If at least one of them is 2, method 2 in Section 3.2 is used to perform the swapping and stability tests. If both eigenvalues are real, $\mathtt{N1} = \mathtt{N2} = 1$ and the swapping is performed using (orthogonal) Givens rotations [30]. In the complex case we perform all reordering with (unitary) Givens rotations. If the problem is too ill–conditioned (i.e. the swap does not pass the stability tests), the swap is rejected. As for _TGEXC, this information is reported on exit by the parameter INFO. The user specifies the index of the first block (A_{11}, B_{11}) in parameter J1. Optionally, the routine also accumulates the orthogonal (unitary) equivalence transformation in Q and Z. The calling sequence and the leading comment lines of DTGEX2 are listed in Appendix B.

5.2 Computing Deflating Subspaces with Specified Eigenvalues

_TGSEN computes (left and right) deflating subspaces associated with some specified eigenvalues of (A, B) in generalized Schur form. Using _TGEXC, (A, B) is reordered so that a selected cluster of M eigenvalues appears in the leading $\mathtt{M} \times \mathtt{M}$ diagonal blocks of A and B. The logical array SELECT specifies the eigenvalues in the cluster, and thereby the value of M. In the real case, if an eigenvalue that belongs to a complex conjugate pair is selected, then by default its conjugate will also belong to the selected cluster (which conforms to the standard eigenvalue problem in LAPACK). Optionally, the matrices of generalized Schur vectors Q and Z are updated with the orthogonal (unitary) equivalence transformations performed. Then, the leading M columns of Q and Z form orthonormal (unitary) bases of the associated left and right deflating subspaces. By calling the LAPACK routine _GEGV, _TGSEN also computes the generalized eigenvalue pairs (α_i, β_i) for $i = 1 : n$, where α_i is a complex number and β_i is a real number. In the real case, real and imaginary parts of α_i are reported in ALPHAR(i) and ALPHAI(i), respectively, and β_i in BETA(i). In the complex case, they are reported in the complex arrays ALPHA and BETA. INFO reports if the reordering was successful or if any swap was rejected due to ill-conditioning. The calling sequence and the leading comment lines of DTGSEN are listed in Appendix C.

5.3 Condition Estimation and Approximate Error Bounds

Optionally, _TGSEN also computes estimates of quantities (condition numbers) that appear in the error bounds summarized in tables 4.1 and 4.2. These are the reciprocal values of the left and right projection norms p and q (defined in (4.14)), and estimates of the separation between two matrix pairs defined by $\mathrm{Dif}_u[(A_{11}, B_{11}), (A_{22}, B_{22})]$ (4.16) and $\mathrm{Dif}_l[(A_{11}, B_{11}), (A_{22}, B_{22})]$ (4.17). The reciprocal values of p and q are reported in PL and PR, respectively, and estimates of Dif_u and Dif_l in DIF(1) and DIF(2), respectively. The functionality obtained from _TGSEN is specified by setting the parameter IJOB, which includes the choice of estimator for Dif_u and Dif_l (one–normed–based or Frobenius normed–based).

_TGSNA estimates the reciprocal condition numbers for specified eigenvalues and eigenvectors of a matrix pair (A, B) in generalized Schur form. The logical array SELECT specifies the M eigenpairs (all or a subset) for which condition numbers are required. The reciprocal values of estimates for the eigenvalue condition numbers $S(\lambda)$ (4.32) are reported in the array S and the corresponding reciprocal values of (Frobenius normed–based) estimates for the eigenvector condition numbers Dif_l are reported in the array DIF. The calling sequence and the leading comment lines of DTGSNA are listed in Appendix D.

_GSRBB computes an algorithm–independent residual–based error bound for a pair (left and right) deflating subspaces of a matrix pair $(C, D) = Q^H(A, B)Z$, where Q and Z transform the original matrix pair (A, B) to generalized Schur canonical form such that the M-by-M (1,1)–block of (C, D) holds a selected cluster of eigenvalues. Optionally, a (Frobenius normed–based) estimate of $\mathrm{Dif}_l[(C_{11}, C_{11}), (D_{22}, D_{22})]$ is reported in DIF. An estimate of the bound(s) (4.29), (4.30) is reported in RBB, and an estimate on η, which should be less than 1, is reported in CNDTN. RRES is the norm of the backward perturbation $\|H_{\mathrm{opt}}\|$ (4.27) associated with the computed pair of deflating subspaces (i.e. the first M columns of Q and Z). INFO is set to 1 if CNDTN ≥ 1. The calling sequence and the leading comment lines of DGSRBB are listed in Appendix E.

Notice, that it is only in _TGSEN where the user has the option to choose between one–normed–based or Frobenius normed–based estimates of Dif_u and Dif_l. The estimation of Dif_l for eigenvectors in _TGSNA and for deflating subspaces in _GSRBB make use of the less expensive but equally reliable Frobenius normed–based estimator (see Section 4.4).

6 Computational Experiments

We have performed an extensive testing of our software on problems ranging from well–conditioned to extremely ill–conditioned. In the following we report detailed results from a selection of test problems as well as a summary of results from the test programs. All results presented in the coming sections are computed on a Sun SPARC station 2 in double precision real (and complex) arithmetic with unit roundoff $\eta =$ EPS $\approx$ 2.2D-16.

6.1 Accuracy and Reliability Results

We have chosen to illustrate the stability and accuracy of our software for a selection of problems "tagged" from 1 to 23, where the basic operation is a swapping of two 2×2 blocks in a 4×4 matrix pair (A, B) in generalized real Schur form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}.$$

The stability tests guarantees that the swapping of two diagonal blocks at most results in $O(\eta\|(A,B)\|_F)$ changes in the original matrix pair. In certain (ill–conditioned) cases this perturbation is enough to change individual eigenvalues a lot (e.g., a real multiple eigenvalue λ of multiplicity k might spread around in a circle in the complex plane with center λ and radius $O(\eta^{1/k})$). However, for well–conditioned or only moderately ill–conditioned cases the change of the eigenvalues is an adequate measure on the reliability and accuracy of a reordering method. Besides, comparing different reordering methods (including the variants discussed in Section 3.2 and the QZ–based method in [30]), we also report estimates of condition numbers and error bounds for eigenvalues and eigenspaces.

6.1.1 Test Problems

The first group of problems (1, 6 and 11) are adopted from [2], and here we treat a standard eigenvalue problem as a generalized one, making it more ill-conditioned. We choose A in (A, B) as

$$\begin{pmatrix} 2 & -87 & -20000 & 1000 \\ 5 & 2 & -20000 & -1000 \\ & & 1 & -11 \\ & & 37 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -3 & 3576 & 4888 \\ 1 & 1 & -88 & -1440 \\ & & 1.0001 & -3 \\ & & 1.0001 & 1.0001 \end{pmatrix}, \begin{pmatrix} 1 & -100 & 400 & -1000 \\ 0.001 & 1 & 1200 & -10 \\ & & 1.0001 & -3 \\ & & 100 & 1.0001 \end{pmatrix},$$

for problems 1, 6 and 11, respectively, and $B = I_4$. Notice that all $(1,2)$–blocks of A have quite large norm.

The following matrix pair (A, B) defines the second group of problems (2, 7 and 12):

$$A = \begin{pmatrix} 1 & 1 & 7 & 5 \\ -1 & 1 & 5 & 9 \\ & & 1 & 1 \\ & & -1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \epsilon & & & \\ & \epsilon & & \\ & & 1 & \\ & & & 1 \end{pmatrix}, \tag{6.1}$$

with the eigenvalues $\lambda_{1,2} = \epsilon^{-1} \pm i\epsilon^{-1}$ and $\lambda_{3,4} = 1 \pm i$, which move along the lines starting at origin and passing $(1, i)$ and $(1, -i)$, respectively. When $0 < \epsilon < 1$ decreases, the eigenvalues $\lambda_{1,2}$ move away from $\lambda_{3,4}$ along these lines. Notice that $\mathrm{Dif}_l = \mathrm{Dif}_u$ is constant $\approx 0.7E-2$ for problems 2, 7 and 12 corresponding to $\epsilon = $ 1E−3, 1E−9 and 1E−15, respectively.

The following matrix pair (A, B) defines the third to fifth group of problems ((3, 8, 13), (4, 9, 14) and (5, 10, 15)):

$$A = \begin{pmatrix} 1 & \delta & x & x \\ -\delta & 1 & 0 & x \\ & & 1+\epsilon & \delta \\ & & -\delta & 1+\epsilon \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & x & x \\ 0 & 1 & 0 & x \\ & & 1 & 0 \\ & & 0 & 1 \end{pmatrix},$$

Table 6.1: Problem characteristics, chordal distances and reciprocal condition numbers

Tag	ϵ	δ	x	$\mathcal{X}(\lambda_1, \lambda_2)$	$\mathcal{X}(\lambda_1, \lambda_3)$	$S(\lambda_{1,2})$	$S(\lambda_{3,4})$	p^{-1}	q^{-1}	Dif_l
1	-	-	-	1E−1	3E−3	6E−4	6E−4	2E−05	2E−05	1E−02
6	-	-	-	7E−1	3E−4	1E−6	1E−6	4E−07	4E−07	3E−04
11	-	-	-	7E−1	3E−4	3E−8	3E−8	2E−08	2E−08	1E−07
2	1E−3	-	-	1E−3	6E−1	1E+0	3E−1	1E−01	1E+00	7E−01
7	1E−9	-	-	1E−9	6E−1	1E+0	3E−1	1E−01	1E+00	7E−01
12	1E−15	-	-	1E−15	6E−1	1E+0	3E−1	1E−01	1E+00	7E−01
3	1E−3	1E+0	5E−1	7E−1	3E−4	4E−3	4E−3	2E−03	2E−03	4E−04
8	1E−3	1E+3	5E−1	2E−3	1E−9	5E−3	5E−3	4E−06	4E−06	7E−07
13	1E−3	1E+9	5E−1	2E−9	1E−21	5E−3	5E−3	8E−07	8E−07	8E−09
4	1E−1	1E−1	5E−1	1E−1	5E−2	3E−1	3E−1	2E−01	2E−01	5E−02
9	1E−4	1E−4	5E−1	1E−4	5E−5	3E−4	3E−4	2E−04	2E−04	5E−05
14	1E−9	1E−9	5E−1	1E−9	5E−10	3E−9	3E−9	2E−09	2E−09	5E−10
5	1E−5	1E−5	0	1E−5	5E−6	1E+0	1E+0	1E+00	1E+00	5E−06
10	1E−5	1E−5	1E+2	1E−5	5E−6	2E−7	2E−7	8E−08	8E−08	5E−06
15	1E−5	1E−5	1E+4	1E−5	5E−6	2E−9	2E−9	8E−10	8E−10	5E−06
16	1E−3	-	-	0	1E−3	3E−22	1E−9	5E−10	5E−10	4E−10
17	1E−6	-	-	0	1E−6	2E−29	1E−18	5E−19	5E−19	2E−16
18	1E−3	-	-	2E−7	1E−3	1E−13	1E−9	5E−10	5E−10	4E−10
19	1E−6	-	-	1E−7	1E−6	8E−20	1E−18	9E−16	9E−16	5E−18
20	1E−3	-	-	0	1E−3	3E−16	3E−16	2E−06	2E−06	4E−10
21	1E−6	-	-	0	1E−6	3E−16	3E−12	2E−12	2E−12	2E−16
22	1E−3	-	-	1E−7	1E−3	7E−8	3E−6	2E−06	2E−06	4E−10
23	1E−6	-	-	2E−7	1E−6	5E−16	3E−12	7E−12	7E−12	2E−16

where x represents a uniformly distributed random number $\in (-x, x)$. The eigenvalues of (A, B) are $\lambda_{1,2} = 1 \pm i\delta$ and $\lambda_{3,4} = 1 + \epsilon \pm i\delta$. For problems 3, 8 and 13 we keep ϵ (= 1E−3) and x (= 0.5) constant and vary δ (= 1, 1E+3 and 1E+9, respectively). This implies that the eigenvalues move along the vertical lines through 1 and $1 + \epsilon$ and form two separated clusters. For problems 4, 9 and 14 we keep x (= 0.5) constant and varies $\epsilon = \delta$ (= 1E−1, 1E−4 and 1E−9, respectively). For ϵ small enough we just have one cluster of close eigenvalues. For problems 5, 10 and 15 we keep $\epsilon = \delta$ (= 1E−5) constant and vary x (= 0, 1E+2 and 1E+4, respectively). Problem 5 corresponds to a homogeneous generalized Sylvester equation and in this group of problems we only increase the departure from "block-diagonality", while keeping one cluster of close eigenvalues.

Problems 16–23 are all modifications of a problem in [16], where (A, B) is defined as [18]:

$$A_{11} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \quad A_{22} = A_{11} - \epsilon \begin{pmatrix} 1+\epsilon & 0 \\ 0 & -1 \end{pmatrix}, \quad B_{11} = B_{22} = I_2.$$

For problems $16 - 19$, $\mathrm{col}(A_{12})$ is equal to the left singular vector corresponding to $\sigma_{\min}(I_2 \otimes A_{11} - A_{22}^T \otimes I_2)$ and $B_{12} = I_2$. For problems $20 - 23$, $\mathrm{col}(A_{12}, B_{12})$ is equal to the left singular vector corresponding to $\sigma_{\min}(Z_u)$, where Z_u is defined in (4.13). The parameter ϵ = 1E−3 for problems 16, 18, 20 and 22 and ϵ = 1E−6 for problems 17, 19, 21 and 23. The last two problems in each group (i.e. 18, 19 and 22, 23) are orthogonally equivalent to the first two (i.e. 16, 17 and 20, 21, respectively).

Table 6.1 shows problem characteristics, including the chordal distance between λ_1 and λ_2 and λ_1 and λ_3, reciprocal values of the individual condition numbers for the two complex conjugate pairs (computed by `DTGSNA`), reciprocal values of (left and right) projector norms (computed by `DTGSEN`) and exact values of Dif_l (computed as $\sigma_{\min}(Z_l)$). Several of these problems represent ill–conditioned eigenvalue problems, where both individual eigenvalues, the cluster of eigenvalues in the $(1, 1)$–block and associated pair of deflating subspaces have large condition numbers.

6.1.2 Comparing Different Reordering Methods

We report results from the three method variants discussed in Section 3.2 and the QZ–based Algorithm 590 [30]. Results from a prototype implementation in Matlab of Method 1 in Section 3.2 (the generic method) has been reported earlier [18]. Besides, the first and the last two groups of problems above, results were reported for problems with finite and infinite eigenvalues (simple as well as defective).

Let $(\bar{A}, \bar{B})$ denote the computed matrix pair after the swapping of two diagonal blocks and $\bar{Q}, \bar{Z}$ be the computed transformation matrices that perform the requested reordering. Now, we consider the following questions:

- How close is $(\bar{Q}\bar{A}\bar{Z}^T, \bar{Q}\bar{B}\bar{Z}^T)$ to the original matrix pair (A, B)?
- How nearly orthogonal are the computed transformation matrices $\bar{Q}$ and $\bar{Z}$?
- How close are the eigenvalues of (A_{ii}, B_{ii}) (before the swapping) and $(\bar{A}_{ii}, \bar{B}_{ii})$ (after the swapping)?

To answer the first two questions we measure the quantities

$$E_{A,B} = \frac{\|(A - \bar{Q}\bar{A}\bar{Z}^T, B - \bar{Q}\bar{B}\bar{Z}^T)\|_F}{\eta\|(A,B)\|_F}, \tag{6.2}$$

$$\|(\bar{A}_{21}, \bar{B}_{21})\|_F, \tag{6.3}$$

$$E_Q = \frac{\|I_n - \bar{Q}^T\bar{Q}\|_F}{\eta}, \quad E_Z = \frac{\|I_n - \bar{Z}^T\bar{Z}\|_F}{\eta}, \tag{6.4}$$

where η is the relative machine precision. Ideally, $E_{A,B}$ and E_Q, E_Z should be of size $O(1)$ and the norms of the $(2,1)$-blocks (6.3) should be of size $O(\eta)$.

In tables 7.1, 7.2, 7.4 and 7.5 we display these quantities and the absolute backward error $\|(E,F)\|_F \equiv \|(A - \bar{Q}\bar{A}\bar{Z}^T, B - \bar{Q}\bar{B}\bar{Z}^T)\|_F$ for Method 1, Method 2, Method 3 and Algorithm 590, respectively. In summary, Method 3 and Algorithm 590 (which both are based on QZ iterations) perform the swapping in all cases with small backward errors in (A, B). The direct methods 1 and 2 reject the swaps for problems 18, 19, 22 and 23. Moreover, Method 2 rejects the swap for problem 13. In Table 7.3 we show computed stability test values and tolerances. We accept a swap if and only if both the weak and the strong stability test values are smaller than *tol1* = *tol2* (see Section 3.2). If a swap is rejected due to (severe) ill–conditioning we can still impose a swap by increasing the tolerances in the stability tests, which is done for methods 1 and 2 to get the results for the "rejected swaps" in tables 7.1 and 7.2.

To answer the last question we display computed eigenvalues after the reordering. Numbers in **bold font** show the absolute error in computed eigenvalues. Table 6.2 and Table 6.3 illustrate the well–known fact that small backward errors do not necessarily imply small errors in the computed eigenvalues. Here the direct Method 2 produces much more accurate eigenvalues than Algorithm 590. For these examples Method 1 and Method 2 produce exactly the same eigenvalues but this is not always the case as we will see later.

Table 6.2: Eigenvalues after reordering for problems 1, 6 and 11

Tag	Alg. 590 – Real parts	Alg. 590 – Imaginary parts
1	0.1000000000**637105**E+01	± 0.2017424100**339241**E+02
	0.1999999999**360595**E+01	± 0.2085665361**375716**E+02
6	0.1000999999**00501**E+01	± 0.1732916616**366925**E+01
	0.1000000000**99329**E+01	± 0.1732050807**522568**E+01
11	0.100099**8004480817**E+01	± 0.100000**4238589241**E+01
	0.100000**1995519146**E+01	±0.99999**57594733729**E+00
Tag	Method 2 – Real parts	Method 2 – Imaginary parts
1	0.100000000000000**1**E+01	± 0.2017424100183201E+02
	0.2000000000000000E+01	± 0.208566536146142**0**E+02
6	0.1001000000000000E+01	± 0.1732916616574496E+01
	0.9999999999999999E+00	± 0.1732050807568877E+01
11	0.100099999999999**5**E+01	± 0.1000000000000**186**E+01
	0.999999999999999**1**E+00	± 0.9999999999999**453**E+00

Table 6.3: Eigenvalues after reordering for problems 5, 10 and 15

Tag	Alg. 590 – Real parts	Alg. 590 – Imaginary parts
5	0.1000010000000000E+01	± 0.100000000000**6551**E-04
	0.1000000000000001E+01	± 0.999999999999**6121**E-05
10	0.10000100**15362113**E+01	± 0.99**86863986519929**E-05
	0.9999999**846379060**E+00	± 0.100**2579812190853**E-04
15	0.10000**63691043548**E+01	±**0.0000000000000000**E+00
	0.10000**11414827676**E+01	± 0.1**394322538135408**E-04
Tag	Method 2 – Real parts	Method 2 – Imaginary parts
5	0.1000010000000000E+01	± 0.1000000000000000E-04
	0.1000000000000000E+01	± 0.1000000000000000E-04
10	0.1000010000000000E+01	± 0.9999999999**295905**E-05
	0.999999999999999**8**E+00	± 0.100000000000**3913**E-04
15	0.1000010000000000E+01	± 0.9999999999**775261**E-05
	0.100000000000000**1E**+01	± 0.10000000000**94131**E-04

Table 6.4: Eigenvalues after reordering for problem 12

Method	Real part	Imaginary part
590	0.999999999999999**8**E+00	± 0.999999999999999**4**E+00
	0.100000**5074603372**E+11	± 0.9999999999**642134**E+10
3	0.999999999999**8981**E+00	± 0.99999999999999**78**E+00
	0.100000000000000**6**E+11	± 0.99999999999999**47**E+10
2	0.999999999999**8981**E+00	± 0.999999999999999**3**E+00
$B = QR$	0.100000000000000**3**E+11	± 0.99999999999999**68**E+10
1	0.999999999999**8976**E+00	± 0.99999999999999**88**E+00
	0.99999**17788454498**E+10	± 0.100000**4769348030**E+11

In Table 6.4 we show that Method 2 and Method 3 compute the most accurate eigenvalues for problem 12, and are superior to Method 1 and Algorithm 590. Moreover, we have not found any example where Method 1 preserves eigenvalues better than Method 2.

Notice that in all cases where Method 2 rejected a swap (except problem 13), Algorithm 590 produced eigenvalues with no accuracy at all (as all methods did when a rejected swap was imposed!). For problem 13, Algorithm 590 produced eigenvalues to half machine precision after the reordering. On the other hand, if SMIN, the threshold for checking non–zero diagonal entries in the LU factorization routine DGELUF in the generalized Sylvester solver (see [19]) is set to the relative machine precision η instead of $\eta\|Z\|_M$ (where $\|Z\|_M$ is the modulus of the largest element in the matrix to factorize), Methods 1–3 perform the swap of the eigenvalues to almost full machine precision. These results are "tagged" 13* in tables 7.1, 7.2, 7.3 and 7.6.

6.1.3 Results from Condition Estimation and Error Bounds

In Table 6.1 we reported some condition estimation results, namely reciprocal values of the individual condition numbers for the two complex conjugate pairs (computed by DTGSNA) and reciprocal values of (left and right) projector norms (computed by DTGSEN). The exact values of $\mathrm{Dif}_l = \mathrm{Dif}_u$ (computed as $\sigma_{\min}(Z_l)$) were also displayed. In Table 7.6 we display Dif_u before and after the reordering for our selection of problems. Moreover, we show the ratios Dif_u/DIF(1)-F and Dif_u/DIF(1)-1, where DIF(1)-F and DIF(1)-1 are the Frobenius normed–based and one–normed–based estimates of Dif_u, respectively. We see that 20 examples are within a factor 10 and the remaining 3 examples are within a factor 100 of the exact value of Dif_u. In the last two columns we report estimates of the reciprocal values of Dif_l, the condition number for the individual eigenvectors corresponding to the complex conjugate pair $\lambda_{1,2}$ computed by DTGSNA and ZTGSNA, respectively.

For a more complete comparison between our Dif_x-estimators ($x = l, u$) including accuracy, performance and reliability results we refer to [19]. Moreover, estimates of condition numbers and error bounds are also checked by the test programs discussed in the next section.

6.2 A Summary of the Results from the Test Programs

We have developed two test programs _CHK3 and _CHK4 for testing and verification of _TGSEN and _GSRBB, and _TGSNA, respectively.

The test program _CHK3 verifies that the backward error is small, the transformation matrices $\bar{Q}$ and $\bar{Z}$ that performed the reordering are orthogonal (unitary), the estimated values DIF(1:2) do not differ too much from the true values of Dif_u and Dif_l, respectively, the chordal distance between "the same" eigenvalues before and after the reordering is small, and that the norm of the $(2,1)$–blocks of the reordered pencil is small. The scheme is to initialize $A_{11}, A_{22}, B_{11}, B_{22}, R$ and L and they define the $(1,2)$–blocks A_{12} and B_{12} (as in (4.12)). The program reorders all eigenvalues in (A_{22}, B_{22}) to the $(1,1)$–block of the matrix pair and checks if everything went well.

The test program _CHK4 verifies that the computed eigenvalue and eigenvector error bounds hold. This is accomplished by using pencils for which the exact eigenvalues and eigenvectors are known.

6.2.1 Test problems for _CHK3

The test problems for _CHK3 are chosen from five different types, generated as follows. The size n of a problem is chosen ≤ 10 and $m = 1, 2, \ldots, 9; n - m = 1, \ldots, 10 - m)$, where (A_{11}, B_{11}) are $m \times m$ and (A_{22}, B_{22}) are $(n-m) \times (n-m)$. Note that Type 1 corresponds to a standard eigenvalue problem but the others are regular generalized eigenvalue problems.

Type 1: Upper triangular problem adapted from [31]: $A_{11} = J_m(1, -1), B_{11} = I_m$ and $A_{22} = J_{n-m}(1-\alpha, 1), B_{22} = I_{n-m}$, where $J_k(d, s)$ denotes a Jordan block of dimension k with d and s as diagonal and superdiagonal elements, respectively. In our tests we use $\alpha = \sqrt{\texttt{EPS}}$, and $1/\sqrt{\texttt{EPS}}$. For $0 < \alpha < 1$, the size of Dif is $O(\alpha^{n-1})$. The entries in $L = R$ are chosen as $r_{ij} = 20(0.5 - \sin(i/j))$, where $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

Type 2: Upper triangular problem with

$$a_{ij} = 2(0.5 - \sin(i)), \quad b_{ij} = 2(0.5 - \sin(i \cdot j)), \quad i = 1, \ldots, m, \quad j = i, \ldots, m.$$

$$a_{ij} = 2(0.5 - \sin(i+j)), \quad b_{ij} = 2(0.5 - \sin(j)), \quad i = m+1, \ldots, n-m, \quad j = i, \ldots, n-m.$$

$$r_{ij} = 20(0.5 - \sin(i \cdot j)), \quad l_{ij} = 20(0.5 - \sin(i+j)), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n-m.$$

Type 3: Quasi upper triangular problem, where the entries in A, B, R and L at first are chosen as for Type 2. Then each second diagonal block in A_{11} and each third diagonal block in A_{22} are made 2×2 by setting $a_{k+1,k+1} = a_{kk}$ and $a_{k+1,k} = -\sin(a_{k,k+1})$ for appropriate values of k.

Type 4: Dense problem at block level, with

$$a_{ij} = 20(0.5 - \sin(i \cdot j)), \quad b_{ij} = 2(0.5 - \sin(i+j)), \quad i = 1, \ldots, m, \quad j = 1, \ldots, m.$$

$$a_{ij} = 20(0.5 - \sin(i+j)), \quad b_{ij} = 2(0.5 - \sin(i \cdot j)), \quad i = m+1, \ldots, n-m, \quad j = m+1, \ldots, n-m.$$

$$r_{ij} = 2(0.5 - \sin(i/j)), \quad l_{ij} = 20(0.5 - \sin(i \cdot j)), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n-m.$$

Type 5: (A, B) has potentially close or common eigenvalues, and large or very large departure from block diagonality. First A_{11} is chosen as the $m \times m$ leading submatrix of A_1, where

$$A_1 = \begin{pmatrix} 1 & \beta & & & & & & & \\ -\beta & 1 & & & & & & & \\ & & 1+\delta & \beta & & & & & \\ & & -\beta & 1+\delta & & & & & \\ & & & & \delta & 1 & & & \\ & & & & -1 & \delta & & & \\ & & & & & & -\delta & 1 & \\ & & & & & & -1 & -\delta & \\ & & & & & & & & 1 \end{pmatrix},$$

then A_{22} is chosen as the $(n-m) \times (n-m)$ leading submatrix of A_2, where

$$A_2 = \begin{pmatrix} -1 & \beta & & & & & & & \\ -\beta & -1 & & & & & & & \\ & & 1-\delta & \beta & & & & & \\ & & -\beta & 1-\delta & & & & & \\ & & & & \delta & 1+\beta & & & \\ & & & & -1-\beta & \delta & & & \\ & & & & & & -\delta & 1+\beta & \\ & & & & & & -1-\beta & -\delta & \\ & & & & & & & & 1-\delta \end{pmatrix}.$$

B_{11} and B_{22} are chosen as the identity matrices I_m and I_{n-m}, respectively. The values of $\beta = 20/\alpha$ and $\delta = -1.5/\alpha$ are used for $\alpha = \sqrt{\texttt{EPS}}$ and $1/\sqrt{\texttt{EPS}}$. The entries in R and L are chosen as $r_{ij} = \alpha(0.5 - \sin(i \cdot j))/20$ and $l_{ij} = \alpha(0.5 - \sin(i + j))/20$, where $i = 1, \ldots, m$ and $j = 1, \ldots, n-m$. All integer values $m, n-m \geq 0$ such that $2 \leq n \leq 10$ are chosen.

6.2.2 A Summary of the Results from _CHK3

The results from the real and complex test runs were similar, only that 60 of the 2700 problems had to be omitted in the complex case because the LAPACK routine CGEGS did not manage to compute the generalized Schur form for these examples. We run both programs on two different target machines, both supporting IEEE floating point standards. A Sun SPARC station only using non–optimized BLAS, and an IBM RS6000/530, where we used the ESSL library for BLAS calls. All programs were compiled with the -O option. The relative machine precision EPS is $\approx 2.22D - 16$ for both machines. In the following, we always start to list a number for Sun SPARC followed by the corresponding number (in parentheses) for IBM RS6000/530.

A total number of 2700 test configurations (including different functionality tests for some problems) are generated in DCHK3 and in total we do around 20790 adjacent swaps. The minimum and maximum values of the backward error were 0.458D-19 (0.458D-19) and 0.193D-14 (0.169D-14), respectively. Minimum and maximum values of $\|I_n - \bar{Q}^T\bar{Q}\|_F$ were 0.0 (0.0) and 0.375D-14 (0.326D-14), respectively. Minimum and maximum values of $\|I_n - \bar{Z}^T\bar{Z}\|_F$ were 0.0 (0.0) and 0.335D-14 (0.289D-14), respectively. These residuals were larger than $10 \times$ EPS in 120 (114) and 144 (78) cases out of 2700, respectively. The estimates of Dif_u and Dif_l differed more than a factor 100 from true values in 170 (184) cases out of 1800, respectively. The maximum value of Dif_x is 1.41 and the corresponding estimate DIF(*) was 1.00 for both machines. The minimum value of Dif_x is 0.210D-19 (0.480D-19) and corresponding estimate DIF(*) was computed as 0.774D-19 (0.774D-19). However, if Dif_x is close to machine epsilon there exists cases where DIF(*) reports an even smaller value (but signals the ill–conditioning correctly). The minimum and maximum values of $\|(A_{21}, B_{21}\|_F$ (i.e. RRES) were 0.0 and 0.360D-6 for both machines and for the same problem. RRES was larger than EPS$\|(A,B)\|_F$ in 18 (0) cases out of 1350, respectively. The chordal distance, $\mathcal{X}(\lambda_{before}, \lambda_{after})$ was larger than EPS$\|(A,B)\|_F$ in 192 (192) cases out of 39600 but never larger than 0.226D-6 (0.266D-6), for the two machines. Minimum and maximum values of the residual based bound on the acute angle between exact and computed deflating subspaces, RBB, were 0.0 (0.0) and 0.184D-6 (0.184D-6), respectively. The maximum value occurred for the very last problem (i.e. Type 5 with $m = 9$, $n - m = 1$ and $\alpha = 1/\sqrt{(\texttt{EPS})}$). Notice that the condition $\eta < 1$ in (4.29) was not fulfilled in 794 (794) cases out of 1350, showing that it is stronger than necessary (since all error bounds gave meaningful information).

We also substituted the *sin*-function in all problems and β and δ in problems of Type 5 with a uniformly random number $\in (0, 1)$, giving similar results as reported above.

6.2.3 Test Problems and a Summary of the Results from _CHK4

The program _CHK4 for testing _TGSNA checks how much the estimates S of the reciprocal value for the eigenvalue condition number $S(\lambda)$ (4.32) differ from the ones computed by using the exact (known) eigenvectors. The program also checks how much the estimates DIF of the reciprocal value for the eigenvector condition number Dif_l differ from the exact computed values $\sigma_{\min}(Z_l)$. In the tests two types of matrix pairs $(A, B) \equiv Y^{-H}(D_a, D_b)X^{-1}$ are used:

Type 1:

$$D_a = \begin{pmatrix} 1+\alpha & & & & \\ & 2+\alpha & & & \\ & & 3+\alpha & & \\ & & & 4+\alpha & \\ & & & & 5+\alpha \end{pmatrix}.$$

Type 2:

$$D_a = \begin{pmatrix} 1 & -1 & & & \\ 1 & 1 & & & \\ & & 1 & & \\ & & & 1+\alpha & 1+\beta \\ & & & -1-\beta & 1+\alpha \end{pmatrix}.$$

For both types $D_b = I_5$ and the exact left and right eigenvectors of (A, B) are the rows and columns of

$$Y^H = \begin{pmatrix} 1 & & -y & y & -y \\ & 1 & -y & y & -y \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}, \quad \text{and} \quad X = \begin{pmatrix} 1 & & -x & -x & x \\ & 1 & x & -x & -x \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix},$$

respectively, where α, β, x and y are given all values independently of each other from $\{\texttt{EPS}^{1/4}, 0.1, 1, 10, \texttt{EPS}^{-1/4}\}$. So, a total of 1250 different pencils (A, B) are generated in the tests. Note that $B \neq I_5$ in $(A, B) = Y^{-H}(D_a, D_b)X^{-1}$.

In summary, the test results are good. The ratio between S and the corresponding $S(\lambda)$ computed by using the exact (known) eigenvectors is 1.0 up to 8 decimal digits for all examples. The reciprocal values for the eigenvector condition numbers Dif_l differ less than a factor 100 from the exact computed value $\sigma_{min}(Z_l)$ in about $90\%, 94\%, 99\%$ and 100% of the 2500 cases for the four different data types `D`, `S`, `Z` and `C`, respectively. This verifies that in real arithmetic `_TGSNA` more often computes a weaker estimate of Dif_l associated with complex conjugate pairs of eigenvalues than in complex arithmetic (see Section 4.4.2).

7 Some Conclusions

Our error analysis of the direct methods and computational experiments (presented in Section 6) give us support to state the following conclusions about our algorithms and software.

- Accuracy and reliability results comparing different reordering methods (Method 1, Method 2 and Method 3 discussed in Section 3.2 and Algorithm 590 [30]) show that Method 2 is to prefer. It is a direct method which is very reliable and it also computes the most accurate eigenvalues of the four methods. Method 2 is implemented in the software presented in Section 5.

- The numerical stability is guaranteed and controlled by computing the size of the backward error and rejecting the swap if it exceeds a certain threshold.

  As mentioned earlier, the generalized eigenvalue problem (as well as the standard unsymmetric problem) is potentially ill–conditioned in the sense that eigenvalues and eigenspaces may change drastically even under small perturbations of the data. If we insist on performing a reordering of (S_{11}, T_{11}) and (S_{22}, T_{22}) for an ill–conditioned problem, we may destroy any spectral information in $A - \lambda B$. Close eigenvalues or small separation between (S_{11}, T_{11}) and (S_{22}, T_{22}) are not enough for rejecting a swap. It is the sensitivity of the eigenspaces that matters most, which in turn is perfectly signaled by the norm of the solutions L and R to the associated generalized Sylvester equation for the direct methods discussed here.

- Qualitative results from our test software on both well–conditioned and ill–conditioned problems, including estimates of reciprocal values of condition numbers for individual eigenvalues, a cluster of eigenvalues, (left and right) eigenvectors, and a pair of (left and right) deflating subspaces, show the reliability and robustness of the algorithms and software presented.

Acknowledgements

We are grateful to Zhaojun Bai and Jim Demmel for fruitful discussions on this work. A special thank to Ji–guang Sun for constructive discussions on error bounds for the generalized eigenvalue problem and comments on an early version of the manuscript.
Financial support has been received from the Swedish National Board of Industrial and Technical Development under grant NUTEK 89-02578P.

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*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[2] Z. Bai and J. Demmel. On swapping diagonal blocks in real Schur form. *Lin. Alg. Appl.*, 186:73–95, 1993.

[3] Z. Bai, J. Demmel, and McKenney A. On computing condition numbers for the nonsymmetric eigenproblem. *ACM Trans. Math. Software*, 19(2):202–223, 1993.

[4] H. Bart, I. Gohberg, M. Kaashoek, and P. Van Dooren. Factorization of transfer functions. *SIAM J. Control Optim.*, 18:675–696, June 1980.

[5] A. Bojanczyk, G. Golub, and P. Van Dooren. The periodic Schur decomposition. Algorithms and applications. SCCM Inter. Rept. NA-92-07, Stanford University, August 1992.

[6] J. Demmel. The condition number of equivalence transformations that block diagonalize matrix pencils. *SIAM J. Num. Anal.*, 20(3):599–610, June 1983.

[7] J. Demmel. On condition numbers and the distance to the nearest ill-posed problem. *Num. Math.*, 51(3):251–289, July 1987.

[8] J. Demmel and B. Kågström. Stable eigendecompositions of matrix pencils $A - \lambda B$. Report UMINF-118.84, Institute of Information Processing, University of Umeå, S-901 87 Umeå, Sweden, 1984.

[9] J. Demmel and B. Kågström. Computing stable eigendecompositions of matrix pencils. *Lin. Alg. Appl.*, 88/89:139–186, April 1987.

[10] J. Demmel and B. Kågström. The Generalized Schur Decomposition of an Arbitrary Pencil $A - \lambda B$: Robust Software with Error Bounds and Applications. Part I: Theory and Algorithms. *ACM Trans. Math. Software*, Vol.19(No. 2):160–174, June 1993.

[11] J. Demmel and B. Kågström. The Generalized Schur Decomposition of an Arbitrary Pencil $A - \lambda B$: Robust Software with Error Bounds and Applications. Part II: Software and Applications. *ACM Trans. Math. Software*, Vol.19(No. 2):175–201, June 1993.

[12] F. Gantmacher. *The Theory of Matrices, Vol. I and II (transl.)*. Chelsea, New York, 1959.

[13] G. Golub and C. Van Loan. *Matrix Computations*. Second Edition. Johns Hopkins University Press, Baltimore, MD, 1989.

[14] W. W. Hager. Condition estimators. *SIAM J. Sci. Stat. Comput.*, 5:311–316, 1984.

[15] N. J. Higham. ALGORITHM 674: Fortran codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. *ACM Trans. Math. Software*, 15(2):168, 1989.

[16] N. J. Higham. Perturbation theory and backward error analysis for $AX - XB = C$. *BIT*, 33(1):124–136, 1993.

[17] B. Kågström. A Perturbation Analysis of the Generalized Sylvester Equation. Report UMINF-92.17, Institute of Information Processing, University of Umeå, S-901 87 Umeå, Sweden, 1992. To appear in *SIAM J. on Matrix Anal. Appl.*, Vol. 15, 1994.

[18] B. Kågström. A Direct Method for Reordering Eigenvalues in the Generalized Real Schur Form of a Regular Matrix Pair (A, B). In M.S. Moonen, G.H. Golub, and B.L.R. De Moor, editors, *Linear Algebra for Large Scale and Real–Time Applications*, pages 195–218. Kluwer Academic Publishers, Amsterdam, 1993.

[19] B. Kågström and P. Poromaa. LAPACK–Style Algorithms and Software for Solving the Generalized Sylvester Equation and Estimating the Separation between Regular Matrix Pairs. Report UMINF-93.23, Institute of Information Processing, University of Umeå, S-901 87 Umeå, Sweden, November, 1993. Also as LAPACK Working Note LAWN 75.

[20] B. Kågström and P. Van Dooren. A generalized state-space approach for the additive decomposition of a transfer matrix. *Int. J. Numerical Linear Algebra with Applications*, 1(2):165–181, 1992.

[21] B. Kågström and L. Westin. Generalized Schur methods with condition estimators for solving the generalized Sylvester equation. *IEEE Trans. Autom. Contr.*, 34(4):745–751, 1989.

[22] A. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Autom. Contr.*, AC-24:913–921, 1979.

[23] C. Moler and G. Stewart. An algorithm for the generalized matrix eigenvalue problem. *SIAM J. Numer. Anal.*, 10:241–256, 1973.

[24] J. L. Rigal and J. Gaches. On the compatability of a given solution with the data of a linear system. *J. Assoc. Comput. Mach.*, 14:543–548, 1967.

[25] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, Oct 1973.

[26] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory.* Academic Press, New York, 1990.

[27] J.-G. Sun. Perturbation expansions for invariant subspaces. *Lin. Alg. Appl.*, 153:85–97, 1991.

[28] J.-G. Sun. Backward perturbation analysis of certain characteristic subspaces. *Num. Math.*, 65:357–382, 1993.

[29] P. Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Stat. Comp.*, 2:121–135, 1981.

[30] P. Van Dooren. ALGORITHM 590: DSUBSP and EXCHQZ: Fortran routines for computing deflating subspaces with specified spectrum. *ACM Trans. Math. Software*, 8:376–382, 1982. and (Corrections) Vol. 4, No. 4, p 787, 1983.

[31] J. Varah. On the separation of two matrices. *SIAM J. Numer. Anal.*, 16:216–222, 1979.

[32] J. H. Wilkinson. Kronecker's canonical form and the QZ algorithm. *Lin. Alg. Appl.*, 28:285–303, 1979.

Table 7.1: Computed errors after the reordering using Method 1

Tag	$\|A_{21}, B_{21}\|_F$	$E_{A,B}$	E_Q	E_Z	$\|(E, F)\|_F$
1	2E−18	4E+00	4E+00	3E+00	3E−11
6	1E−21	3E+00	3E+00	5E+00	4E−12
11	4E−19	2E+00	2E+00	2E+00	5E−13
2	4E−16	2E+00	8E+00	3E+00	7E−15
7	5E−16	2E+00	3E+00	2E+00	5E−15
12	2E−15	3E+00	7E+00	5E+00	7E−14
3	2E−18	3E+00	5E+00	3E+00	2E−15
8	2E−18	3E+00	4E+00	5E+00	2E−12
13	2E−13	3E+00	5E+00	2E+00	1E−06
13*	1E−18	2E+00	2E+00	3E+00	9E−07
4	8E−16	3E+00	5E+00	4E+00	2E−15
9	7E−19	3E+00	6E+00	2E+00	2E−15
14	8E−24	2E+00	3E+00	4E+00	1E−15
5	0E+00	0E+00	0E+00	0E+00	0E+00
10	4E−22	1E+00	3E+00	2E+00	5E−14
15	5E−24	3E+00	7E+00	3E+00	1E−11
16	5E−19	3E+00	6E+00	2E+00	2E−15
17	1E−21	1E+00	2E+00	2E+00	1E−15
18	5E−14	6E+01	4E+00	2E+00	5E−14
19	3E−06	4E+09	6E+00	2E+00	3E−06
20	3E−16	3E+00	7E+00	3E+00	2E−15
21	3E−16	3E+00	2E+00	4E+00	2E−15
22	3E−11	4E+04	4E+00	2E+00	3E−11
23	2E−10	3E+05	4E+00	2E+00	2E−10

Table 7.2: Computed errors after the reordering using Method 2

Tag	$\|A_{21}, B_{21}\|_F$	$E_{A,B}$	E_Q	E_Z	$\|(E,F)\|_F$
1	3E−19	1E+00	3E+00	3E+00	1E−11
6	6E−22	3E+00	3E+00	5E+00	4E−12
11	2E−20	2E+00	3E+00	4E+00	6E−13
2	1E−15	2E+00	8E+00	2E+00	7E−15
7	5E−16	2E+00	9E−01	1E+00	6E−15
12	1E−15	3E+00	6E+00	3E+00	7E−14
3	2E−18	3E+00	5E+00	3E+00	2E−15
8	1E−18	3E+00	4E+00	5E+00	2E−12
13	1E−04	3E+02	4E+00	2E+00	1E−04
13*	1E−18	2E+00	2E+00	3E+00	9E−07
4	3E−16	3E+00	4E+00	4E+00	2E−15
9	7E−19	2E+00	4E+00	2E+00	2E−15
14	6E−24	2E+00	3E+00	4E+00	1E−15
5	0E+00	0E+00	0E+00	0E+00	0E+00
10	4E−22	1E+00	3E+00	2E+00	5E−14
15	2E−24	3E+00	7E+00	3E+00	1E−11
16	6E−19	4E+00	6E+00	2E+00	3E−15
17	9E−22	1E+00	2E+00	2E+00	1E−15
18	5E−14	6E+01	3E+00	2E+00	5E−14
19	3E−06	4E+09	5E+00	1E+00	3E−06
20	1E−16	3E+00	6E+00	4E+00	2E−15
21	2E−16	3E+00	2E+00	5E+00	3E−15
22	6E−12	7E+03	3E+00	3E+00	6E−12
23	2E−10	3E+05	2E+00	2E+00	2E−10

Table 7.3: Method 2 – Computed stability test values and tolerances

Tag	Weak($B = QR$)	Weak($B = RQ$)	Strong	$\mathrm{tol}_{1,2}$
1	8.17E−19	3.16E−19	8.53E−12	7.02E−11
6	7.03E−22	1.05E−21	3.03E−12	1.38E−11
11	2.41E−19	2.55E−21	6.544E−13	3.59E−12
2	9.75E−16	3.88E−13	6.56E−15	3.06E−14
7	3.19E−16	4.75E−08	3.57E−15	3.06E−14
12	1.58E−15	3.99E−6	4.92E−14	3.06E−14
3	6.97E−19	7.27E−19	1.60E−15	7.94E−15
8	5.26E−19	8.56E−19	9.49E−13	4.44E−12
13	1.20E−04	1.20E−04	1.20E−04	4.44E−06
13*	6.09E−19	9.02E−19	7.64E−07	4.44E−06
4	1.41E−16	2.20E−16	1.45E−15	6.75E−15
9	1.33E−19	1.33E−19	1.36E−15	6.58E−15
14	5.91E−25	5.91E−25	8.83E−16	6.58E−15
5	0.00E+00	0.00E+00	0.00E+00	6.28E−15
10	2.38E−22	2.38E−22	5.97E−14	3.93E−13
15	1.41E−24	1.41E−24	1.09E−11	3.93E−11
16	7.82E−17	7.85E−17	1.25E−15	6.28E−15
17	7.85E−17	7.85E−17	7.71E−16	6.28E−15
18	4.62E−14	4.62E−14	4.62E−14	8.01E−15
19	3.05E−06	3.05E−06	3.05E−06	8.01E−15
20	2.94E−16	4.14E−16	1.88E−15	5.91E−15
21	1.95E−16	4.22E−18	1.17E−15	5.88E−15
22	5.73E−12	5.79E−12	5.73E−12	8.01E−15
23	2.38E−10	2.38E−10	2.38E−10	8.01E−15

Table 7.4: Computed errors after the reordering using Method 3

Tag	$\|A_{21}, B_{21}\|_F$	$E_{A,B}$	E_Q	E_Z	$\|(E,F)\|_F$
1	7E−19	5E+00	5E+00	3E+00	3E−11
6	4E−22	1E+00	6E+00	3E+00	1E−12
11	3E−20	3E+00	4E+00	3E+00	9E−13
2	5E−16	4E+00	8E+00	5E+00	1E−14
7	5E−16	9E−01	4E+00	1E+00	3E−15
12	1E−15	4E+00	4E+00	3E+00	1E−14
3	9E−19	3E+00	2E+00	5E+00	3E−15
8	9E−19	5E+00	6E+00	4E+00	2E−12
13	1E−07	4E+00	1E+01	4E+00	2E−06
4	2E−16	4E+00	7E+00	4E+00	3E−15
9	1E−18	2E+00	6E+00	3E+00	1E−15
14	5E−24	3E+00	5E+00	2E+00	2E−15
5	0E+00	0E+00	0E+00	0E+00	0E+00
10	5E−22	2E+00	4E+00	1E+00	8E−14
15	3E−24	4E+00	5E+00	2E+00	2E−11
16	1E−18	4E+00	7E+00	1E+00	3E−15
17	2E−16	2E+00	2E+00	3E+00	2E−15
18	3E−16	2E+00	2E+00	3E+00	2E−15
19	2E−18	5E+00	8E+00	1E+01	4E−15
20	9E−17	4E+00	5E+00	5E+00	3E−15
21	2E−16	3E+00	4E+00	5E+00	3E−15
22	2E−16	3E+00	5E+00	2E+00	2E−15
23	4E−16	1E+01	2E+01	8E+00	8E−15

Table 7.5: Computed errors after the reordering using Algorithm 590

Tag	$\|A_{21}, B_{21}\|_F$	$E_{A,B}$	E_Q	E_Z	$\|(E, F)\|_F$
1	2E−12	3E+00	5E+00	2E+00	2E−11
6	2E−13	2E+00	5E+00	3E+00	2E−12
11	2E−13	6E+00	1E+01	8E+00	2E−12
2	8E−16	5E+00	6E+00	4E+00	2E−14
7	1E−15	6E+00	5E+00	5E+00	2E−14
12	3E−15	7E+00	7E+00	3E+00	2E−14
3	2E−16	3E+00	4E+00	4E+00	2E−15
8	1E−13	2E+00	3E+00	3E+00	7E−13
13	4E−07	7E+00	6E+00	1E+01	3E−06
4	5E−16	3E+00	3E+00	4E+00	2E−15
9	3E−16	3E+00	4E+00	4E+00	2E−15
14	6E−16	3E+00	3E+00	3E+00	2E−15
5	3E−16	4E+00	6E+00	4E+00	2E−15
10	1E−14	3E+00	6E+00	4E+00	1E−13
15	2E−12	2E+00	1E+00	2E+00	8E−12
16	3E−19	2E+00	2E+00	3E+00	2E−15
17	3E−22	3E+00	3E+00	3E+00	2E−15
18	5E−16	4E+00	5E+00	5E+00	3E−15
19	5E−16	4E+00	6E+00	5E+00	3E−15
20	2E−16	1E+00	2E+00	1E+00	1E−15
21	1E−16	1E+00	2E+00	2E+00	1E−15
22	2E−16	3E+00	4E+00	3E+00	2E−15
23	4E−16	4E+01	4E+01	4E+01	3E−14

Table 7.6: Method 2 – Some computed quantities before and after the reordering

Tag	Dif_u-before	Dif_u-after	Dif_u/`DIF(1)`-F	Dif_u/`DIF(1)`-1	Dif_l(`DTGSNA`)	Dif_l(`ZTGSNA`)
1	1E−02	1E−02	0.57	1.90	3E−2	3E−4
6	3E−04	3E−06	0.63	1.30	4E−4	1E−6
11	1E−07	1E−07	0.96	1.20	2E−7	3E−6
2	7E−01	1E−01	0.52	1.00	1E−3	2E−3
7	7E−01	1E−01	0.52	1.00	1E−9	2E−9
12	7E−01	1E−02	0.52	1.00	1E−15	2E−15
3	4E−04	4E−04	0.87	1.60	7E−4	7E−4
8	7E−07	7E−07	0.71	1.40	1E−6	2E−6
13	8E−09	2E−07	0.73	1.50	1E−7	4E−7
13*	8E−09	2E−10	2E+02	4E+02	-	-
4	5E−02	4E−03	0.35	1.30	9E−2	2E−2
9	5E−05	2E−06	0.19	1.10	9E−5	2E−8
14	5E−10	2E−10	0.19	1.10	0	0
5	5E−06	5E−06	0.55	1.40	9E−6	7E−6
10	5E−06	2E−07	0.19	1.10	9E−6	1E−12
15	5E−06	2E−07	0.19	1.10	9E−6	1E−14
16	4E−10	4E−10	0.35	1.40	3E−22	4E−22
17	2E−16	2E−17	20.00	79.00	3E−28	4E−28
18	4E−10	4E−10	0.42	1.60	6E−10	3E−13
19	5E−18	6E−16	0.35	1.40	8E−16	3E−19
20	4E−10	3E−06	0.71	1.50	3E−16	3E−16
21	2E−16	3E−12	0.71	1.50	3E−16	3E−16
22	4E−10	3E−06	0.71	1.50	2E−8	2E−7
23	2E−16	4E−17	44.00	100.00	1E−16	3E−16

A Calling sequence DTGEXC

Here we display the parameter list and the leading comment lines of the double precision routine DTGEXC.

```
      SUBROUTINE DTGEXC(WANTQ, WANTZ, N, A, LDA, B, LDB, Q, LDQ, Z,
     $                  LDZ, IFST, ILST, WORK, LWORK, INFO )
*
       IMPLICIT NONE
*
*     --- (preliminary version) ---
*     Bo Kagstrom and Peter Poromaa, Institute of Information Processing,
*     Univ. of Umea, S-901 87 Sweden.
*     Jan. 1994
*
*     .. Scalar Arguments ..
      LOGICAL            WANTQ, WANTZ
      INTEGER            INFO, LDA, LDB, LDZ, LDQ, N, IFST, ILST,
     $                   LWORK
*     ..
*     .. Array Arguments ..
      DOUBLE PRECISION   Q( LDQ, * ), A( LDA, * ), B(LDB, *),
     $                   Z(LDZ, *), WORK( * )
*     ..
*
*  Purpose
*  =======
*
*  DTGEXC reorders the generalized real Schur decomposition of a real matrix pair,
*  using an orthogonal equivalence transformation (A, B) = Q * (A, B) * Z',
*  so that the diagonal block of (A, B) with row index IFST is moved
*  to row ILST.
*
*  (A, B) must be in generalized real Schur canonical form (as returned by
*  DGEGS), i.e. A is block upper triangular with 1-by-1 and 2-by-2 diagonal
*  blocks. B is upper triangular.
*
*  Optionally, the matrices Q and Z of generalized Schur vectors are updated.
*
*         Q(in) * A(in) * Z(in)' = Q(out) * A(out) * Z(out)'
*         Q(in) * B(in) * Z(in)' = Q(out) * B(out) * Z(out)'
*
*  References
*  ==========
*
*  [1] B. Kagstrom; A Direct Method for Reordering Eigenvalues in the
*      Generalized Real Schur Form of a Regular Matrix Pair (A, B), in
*      M.S. Moonen et al (eds), Linear Algebra for Large Scale and
*      Real-Time Applications, Kluwer Academic Publ. 1993, pp 195-218.
*
*  [2] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified
```

```
*      Eigenvalues of a Regular Matrix Pair (A, B) and Condition
*      Estimation: Theory, Algorithms and Software, Report UMINF - 94.04,
*      Inst. of Information Processing, University of Umea, S-901 87 Umea,
*      Sweden, Febrauary 1994. (also published as LAPACK Working Note xx)
*
*  [3] B. Kagstrom and P. Poromaa, LAPACK-Style Algorithms and Software for
*      Solving the Generalized Sylvester Equation and Estimating the Separa-
*      tion between Regular Matrix Pairs, Report UMINF - 93.23, Inst. of
*      Information Processing, University of Umea, S-901 87 Umea, Sweden,
*      November 1993.(also published as LAPACK Working Note xx)
*
*  Arguments
*  =========
*
*  WANTQ  (input) LOGICAL
*          .TRUE. : update the left transformation matrix Q;
*          .FALSE.: do not update Q.
*
*  WANTZ  (input) LOGICAL
*          .TRUE. : update the right transformation matrix Z;
*          .FALSE.: do not update Z.
*
*  N       (input) INTEGER
*          The order of the matrices A and B. N >= 0.
*
*  A, B    (input/output) DOUBLE PRECISION arrays, dimensions (LDA(B),N)
*          On entry, the matrix pair (A, B), in generalized real Schur
*          canonical form.
*          On exit, the updated matrix pair (A, B), again in generalized
*          real Schur canonical form.
*
*  LDA     (input)  INTEGER
*          The leading dimension of the array A. LDA >= max(1,N).
*
*  LDB     (input)  INTEGER
*          The leading dimension of the array B. LDB >= max(1,N).
*
*  Q       (input/output) DOUBLE PRECISION array, dimension (LDZ,N)
*          On entry, if WANTQ is .TRUE., the orthogonal matrix Q.
*          On exit, if WANTQ is .TRUE., the updated matrix Q.
*          If WANTQ is .FALSE., Q is not referenced.
*
*  LDQ     (input) INTEGER
*          The leading dimension of the array Q.
*          LDQ >= 1; and if WANTQ is .TRUE., LDQ >= N.
*
*  Z       (input/output) DOUBLE PRECISION array, dimension (LDZ,N)
*          On entry, if WANTZ is .TRUE., the orthogonal matrix Z.
*          On exit, if WANTZ is .TRUE., the updated matrix Z.
*          If WANTZ is .FALSE., Z is not referenced.
*
```

```
*  LDZ     (input) INTEGER
*          The leading dimension of the array Z.
*          LDZ >= 1; and if WANTZ is .TRUE., LDZ >= N.
*
*  IFST   (input/output) INTEGER
*  ILST   (input/output) INTEGER
*          Specify the reordering of the diagonal blocks of (A, B).
*          The block with row index IFST is moved to row ILST, by a
*          sequence of transpositions between adjacent blocks.
*          On exit, if IFST pointed on entry to the second row of
*          a 2-by-2 block, it is changed to point to the first row;
*          ILST always points to the first row of the block in its
*          final position (which may differ from its input value by
*          +1 or -1). 1 <= IFST, ILST <= N.
*
*  WORK    (workspace) DOUBLE PRECISION array, dimension (LWORK)
*
*  LWORK   (input) INTEGER
*          The dimension of the array WORK. LWORK >= 4*N + 16.
*
*  INFO    (output) INTEGER
*            0: Successful exit.
*            1: The transformed matrix pair (A, B) would be too far from
*               generalized Schur form; the problem is ill-conditioned.
*               (A, B) may have been partially reordered, and ILST points
*               to the first row of the current position of the block
*               being moved.
*           -k: The k:th argument had an illegal value. If k = 14
*               WORK(1) will hold an appropriate value of LWORK.
*
```

B Calling sequence DTGEX2

Here we display the parameter list and the leading comment lines of the double precision routine DTGEX2.

```
      SUBROUTINE DTGEX2(WANTQ, WANTZ, N, A, LDA, B, LDB, Q, LDQ, Z,
     $                  LDZ, J1, N1, N2, WORK, LWORK, INFO )
*
       IMPLICIT NONE
*
*     --- (preliminary version) ---
*     Bo Kagstrom and Peter Poromaa, Institute of Information Processing,
*     Univ. of Umea, S-901 87 Sweden.
*     Apr. 1994
*
*     .. Scalar Arguments ..
      LOGICAL            WANTQ, WANTZ
      INTEGER            INFO, LDA, LDB, LDZ, LDQ, N, N1, N2,
     $                   LWORK, J1
```

```
*     ..
*     .. Array Arguments ..
      DOUBLE PRECISION   Q( LDQ, * ), A( LDA, * ), B(LDB, *),
     $                   Z(LDZ, *), WORK( * )
*     ..
*
*  Purpose
*  =======
*
*  DTGEX2 swaps adjacent diagonal blocks (A11, B11) and (A22, B22)
*  of size 1-by-1 or 2-by-2 in an upper (quasi) triangular matrix pair
*  (A, B) by an orthogonal equivalence transformation.
*
*  (A, B) must be in generalized real Schur canonical form (as returned by
*  DGEGS), i.e. A is block upper triangular with 1-by-1 and 2-by-2 diagonal
*  blocks. B is upper triangular.
*
*  Optionally, the matrices Q and Z of generalized Schur vectors are updated.
*
*         Q(in) * A(in) * Z(in)' = Q(out) * A(out) * Z(out)'
*         Q(in) * B(in) * Z(in)' = Q(out) * B(out) * Z(out)'
*
*  References
*  ==========
*
*  [1] B. Kagstrom; A Direct Method for Reordering Eigenvalues in the
*      Generalized Real Schur Form of a Regular Matrix Pair (A, B), in
*      M.S. Moonen et al (eds), Linear Algebra for Large Scale and
*      Real-Time Applications, Kluwer Academic Publ. 1993, pp 195-218.
*
*  [2] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified
*      Eigenvalues of a Regular Matrix Pair (A, B) and Condition
*      Estimation: Theory, Algorithms and Software, Report UMINF - 94.04,
*      Inst. of Information Processing, University of Umea, S-901 87 Umea,
*      Sweden, Febrauary 1994. (also published as LAPACK Working Note xx)
*
*  Arguments
*  =========
*
*  WANTQ  (input) LOGICAL
*          .TRUE. : update the left transformation matrix Q;
*          .FALSE.: do not update Q.
*
*  WANTZ  (input) LOGICAL
*          .TRUE. : update the right transformation matrix Z;
*          .FALSE.: do not update Z.
*
*  N       (input) INTEGER
*          The order of the matrices A and B. N >= 0.
*
*  A, B    (input/output) DOUBLE PRECISION arrays, dimensions (LDA(B),N)
```

```
*          On entry, the matrix pair (A, B), in generalized real Schur
*          canonical form.
*          On exit, the updated matrix pair (A, B), again in generalized
*          real Schur canonical form.
*
*  LDA     (input)  INTEGER
*          The leading dimension of the array A. LDA >= max(1,N).
*
*  LDB     (input)  INTEGER
*          The leading dimension of the array B. LDB >= max(1,N).
*
*  Q       (input/output) DOUBLE PRECISION array, dimension (LDZ,N)
*          On entry, if WANTQ is .TRUE., the orthogonal matrix Q.
*          On exit, if WANTQ is .TRUE., the updated matrix Q.
*          If WANTQ is .FALSE., Q is not referenced.
*
*  LDQ     (input) INTEGER
*          The leading dimension of the array Q.
*          LDQ >= 1; and if WANTQ is .TRUE., LDQ >= N.
*
*  Z       (input/output) DOUBLE PRECISION array, dimension (LDZ,N)
*          On entry, if WANTZ is .TRUE., the orthogonal matrix Z.
*          On exit, if WANTZ is .TRUE., the updated matrix Z.
*          If WANTZ is .FALSE., Z is not referenced.
*
*  LDZ     (input) INTEGER
*          The leading dimension of the array Z.
*          LDZ >= 1; and if WANTZ is .TRUE., LDZ >= N.
*
*  J1      (input) INTEGER
*          The index to the first block (A11, B11). 1 <= J1 <= N.
*
*  N1      (input) INTEGER
*          The order of the first block (A11, B11). N1 = 0, 1 or 2.
*
*  N2      (input) INTEGER
*          The order of the second block (A22, B22). N2 = 0, 1 or 2.
*
*  WORK    (workspace) DOUBLE PRECISION array, dimension LWORK.
*
*  LWORK   (input) INTEGER
*          The dimension of the array WORK.
*          LWORK >=  MAX((N * (N2 + N1)), ((N2 + N1) * (N2 + N1) * 2))
*
*  INFO    (output) INTEGER
*            0: Successful exit
*            1: The transformed matrix (A, B) would be too far from
*               Generalized Schur form; the blocks are not swapped
*               and (A, B) and (Q, Z) are unchanged. Problem too
*               ill-conditioned.
*          -14: LWORK is to small. Appropriate value for LWORK is
```

```
*              returned in WORK(1).
*
```

C Calling sequence DTGSEN

Here we display the parameter list and the leading comment lines of the double precision routine DTGSEN.

```
      SUBROUTINE DTGSEN( IJOB, WANTQ, WANTZ, SELECT, N, A, LDA, B, LDB,
     $                   ALPHAR, ALPHAI, BETA,
     $                   Q, LDQ, Z, LDZ, M, PL, PR, DIF, WORK,
     $                   LWORK, IWORK, LIWORK, INFO )
*
       IMPLICIT NONE
*
*     --- (preliminary version) ---
*     Bo Kagstrom and Peter Poromaa, Institute of Information Processing,
*     Univ. of Umea, S-901 87 Sweden.
*     Jan. 1994
*
*     .. Scalar Arguments ..
      LOGICAL            WANTQ, WANTZ
      INTEGER            IJOB, N, LDA, LDB, LDQ, LDZ, LWORK, LIWORK, M,
     $                   INFO
      DOUBLE PRECISION   PL, PR
*     ..
*     .. Array Arguments ..
      LOGICAL            SELECT( * )
      INTEGER            IWORK( * )
      DOUBLE PRECISION   Q( LDQ, * ), A( LDA, * ), B(LDB, *),
     $                   Z(LDZ, *), ALPHAR(*), ALPHAI(*),
     $                   BETA(*), DIF(*), WORK(*)
*     ..
*
*  Purpose
*  =======
*
*  DTGSEN reorders the generalized real Schur decomposition of a real
*  matrix pair (A, B) (in terms of an orthonormal equivalence trans-
*  formation Q' * (A, B) * Z), so that a selected cluster of eigenvalues
*  appears in the leading diagonal blocks of the upper quasi-triangular
*  matrix A and and the upper triangular B. The leading columns of Q and Z
*  form orthonormal bases of the corresponding left and right eigenspaces
*  (deflating subspaces).
*
*  DTGSEN also computes the generalized eigenvalues
*
*             w(j) = (ALPHAR(j) + i*ALPHAI(j))/BETA(j)
*
*  of the reordered matrix pair (A, B).
```

```
*
*  (A, B) must be in generalized real Schur canonical form (as returned by
*  DGEGS), i.e. A is block upper triangular with 1-by-1 and 2-by-2 diagonal
*  blocks. B is upper triangular.
*
*  Optionally, the matrices Q and Z of generalized Schur vectors are updated.
*
*         Q(in) * A(in) * Z(in)' = Q(out) * A(out) * Z(out)'
*         Q(in) * B(in) * Z(in)' = Q(out) * B(out) * Z(out)'
*
*  Optionally, the routine computes estimates of reciprocal condition numbers
*  for eigenvalues and eigenspaces. These are Difu[(A11,B11), (A22,B22)]
*  and Difl[(A11,B11), (A22,B22)], i.e. the separation(s) between the matrix
*  pairs (A11, B11) and (A22,B22) that correspond to the selected cluster and
*  the eigenvalues outside the cluster, respectively, and norms of "projections"
*  onto left and right eigenspaces w.r.t. the selected cluster in the (1,1)-block.
*
*  References
*  ==========
*
*  [1] B. Kagstrom; A Direct Method for Reordering Eigenvalues in the
*      Generalized Real Schur Form of a Regular Matrix Pair (A, B), in
*      M.S. Moonen et al (eds), Linear Algebra for Large Scale and
*      Real-Time Applications, Kluwer Academic Publ. 1993, pp 195-218.
*
*  [2] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified
*      Eigenvalues of a Regular Matrix Pair (A, B) and Condition
*      Estimation: Theory, Algorithms and Software, Report UMINF - 94.04,
*      Inst. of Information Processing, University of Umea, S-901 87 Umea,
*      Sweden, February 1994. (also published as LAPACK Working Note xx)
*
*  [3] B. Kagstrom and P. Poromaa, LAPACK-Style Algorithms and Software for
*      Solving the Generalized Sylvester Equation and Estimating the Separa-
*      tion between Regular Matrix Pairs, Report UMINF - 93.23, Inst. of
*      Information Processing, University of Umea, S-901 87 Umea, Sweden,
*      November 1993.(also published as LAPACK Working Note xx)
*
*  Arguments
*  =========
*
*  IJOB     (input) integer
*            Specifies what functionality is to be obtained.
*           0 : Only reorder w.r.t. SELECT. No extras.
*           1 : Reciprocal of norms of "projections" onto left and right eigenspaces
*               w.r.t. the selected cluster (PL and PR).
*           2 : Upper bounds on Difu and Difl. F-norm-based estimate (DIF(1:2)).
*           3 : Estimate of Difu and Difl. 1-norm-based estimate (DIF(1:2)).
*               About 5 times as expensive as IJOB = 2.
*           4 : Compute PL, PR and DIF (i.e. 0, 1 and 2 above): Economic-version to
*               get it all.
*           5 : Compute PL, PR and DIF (i.e. 0, 1 and 3 above)
```

```
*
*  WANTQ  (input) LOGICAL
*          .TRUE. : update the left transformation matrix Q;
*          .FALSE.: do not update Q.
*
*  WANTZ  (input) LOGICAL
*          .TRUE. : update the right transformation matrix Z;
*          .FALSE.: do not update Z.
*
*  SELECT  (input) LOGICAL array, dimension (N)
*          SELECT specifies the eigenvalues in the selected cluster.
*          To select a real eigenvalue w(j), SELECT(j) must be set to
*          .TRUE.. To select a complex conjugate pair of eigenvalues
*          w(j) and w(j+1), corresponding to a 2-by-2 diagonal block,
*          either SELECT(j) or SELECT(j+1) or both must be set to
*          .TRUE.; a complex conjugate pair of eigenvalues must be
*          either both included in the cluster or both excluded.
*
*  N       (input) INTEGER
*          The order of the matrices A and B. N >= 0.
*
*  A       (input/output) DOUBLE PRECISION array, dimension(LDA,N)
*          On entry, the upper quasi-triangular matrix A, with (A, B) in
*          generalized real Schur canonical form.
*          On exit, A is overwritten by the reordered matrix A, again (A, B)
*          in generalized real Schur canonical form, with the selected
*          eigenvalues in the leading diagonal blocks.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A. LDA >= max(1,N).
*
*  B       (input/output) DOUBLE PRECISION array, dimension(LDB,N)
*          On entry, the upper triangular matrix B, with (A, B) in
*          generalized real Schur canonical form.
*          On exit, B is overwritten by the reordered matrix B, again (A, B)
*          in generalized real Schur canonical form, with the selected
*          eigenvalues in the leading diagonal blocks.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B. LDB >= max(1,N).
*
*  ALPHAR  (output) DOUBLE PRECISION array, dimension (N)
*          ALPHAR(1:N) will be set to the real parts of the diagonal
*          elements of A that would result from reducing (A, B) to
*          generalized Schur form and then further reducing them both to
*          triangular form using unitary transformations s.t. the diagonal
*          of B is non-negative real.  Thus, if A(j,j) is a 1x1 block
*          (i.e., A(j+1,j) = A(j,j+1) = 0), then ALPHAR(j) = A(j,j).
*
*  ALPHAI  (output) DOUBLE PRECISION array, dimension (N)
*          ALPHAI(1:N) will be set to the imaginary parts of the diagonal
```

```
*          elements of A that would result from reducing (A, B) to
*          generalized Schur form and then further reducing them both to
*          triangular form using unitary transformations s.t. the diagonal
*          of B is non-negative real.  Thus, if A(j,j) is a 1x1 block
*          (i.e., A(j+1,j) = A(j,j+1) = 0), then ALPHAI(j) = 0.
*
*  BETA    (output) DOUBLE PRECISION array, dimension (N)
*          BETA(1:N) will be set to the (real) diagonal elements of B
*          that would result from reducing (A, B) to generalized Schur form
*          and then further reducing them both to triangular form using
*          unitary transformations s.t. the diagonal of B is
*          non-negative real.  Thus, if A(j,j) is a 1x1 block (i.e.,
*          A(j+1,j) = A(j,j+1) = 0), then BETA(j) = B(j,j).
*          (Note that BETA(1:N) will always be non-negative real, and
*          no BETAI is necessary.)
*
*  Q       (input/output) DOUBLE PRECISION array, dimension (LDQ,N)
*  Z       (input/output) DOUBLE PRECISION array, dimension (LDZ,N)
*          On entry, if WANTQ and/or WANTZ, the matrices Q and Z of generalized
*          Schur vectors.
*          On exit, if WANTQ and/or WANTZ, Q and/or Z have been postmultiplied
*          by the orthogonal transformation matrices which reorder (A, B);
*          The leading M columns of Q and Z form orthonormal bases for the
*          specified pair of left and right eigenspaces (deflating subspaces).
*          If WANTQ = .FALSE., Q is not referenced.
*          If WANTZ = .FALSE., Z is not referenced.
*
*  LDQ     (input) INTEGER
*          The leading dimension of the array Q.
*          LDQ >= 1; and if WANTQ, LDQ >= N.
*
*  LDZ     (input) INTEGER
*          The leading dimension of the array Z.
*          LDZ >= 1; and if WANTZ, LDZ >= N.
*
*  M       (output) INTEGER
*          The dimension of the specified pair of left and right eigen-
*          spaces (deflating subspaces). 0 <= M <= N.
*
*  PL, PR  (output) DOUBLE PRECISION
*          If IJOB = 1, 4 or 5, PL, PR are lower bounds on the reciprocal
*          of the norm of "projections" onto left and right eigenspaces,
*          respectively, w.r.t. the selected cluster. 0 < PL, PR <= 1.
*          See also further details  and references [2-3].
*          If M = 0 or M = N, PL = PR  = 1.
*          If IJOB = 0, 2 or 3, PL and PR are not referenced.
*
*  DIF    (output) DOUBLE PRECISION array, dimension (2).
*          If IJOB >= 2, DIF(1:2) store the estimates of Difu and Difl,
*          respectively. If IJOB = 2 or 4, DIF(1:2) are F-norm-based upper bounds
*          on Difu and Difl, respectively. If IJOB = 3 or 5, DIF(1:2) are 1-norm-
```

```
*          based estimates of Difu and Difl, respectively, computed using
*          reversed communication with DLACON. See also further details and
*          references [2-3].
*          If M = 0 or N, DIF(1:2) = F-norm([A, B]).
*          If IJOB = 0 or 1, DIF is not referenced.
*
*  WORK    (workspace) DOUBLE PRECISION array, dimension (LWORK)
*
*  LWORK   (input) INTEGER
*          The dimension of the array WORK. LWORK >= (2 * N * N)  + N.
*          If IJOB >= 4 LWORK >= MAX(2*N*N+N, 4*M*(N-M)).
*
*  IWORK   (workspace) INTEGER, dimension (LIWORK)
*          IF IJOB = 0 , IWORK is not referenced.
*
*  LIWORK  (input) INTEGER
*          The dimension of the array IWORK. LIWORK >= N + 6.
*          If IJOB = 3 or 5, LIWORK >= MAX(2*M*(N-M), N+6).
*
*  INFO    (output) INTEGER
*            0: Successful exit.
*          < 0: If INFO = -i, the i-th argument had an illegal value.
*            1: Reordering of (A, B) failed because the transformed
*               matrix pair (A, B) would be too far from generalized
*               Schur form; the problem is very ill-conditioned.
*               (A, B) may have been partially reordered.
*               If requested, 0 is returned in DIF(*), PL and PR.
*
```

D Calling sequence DTGSNA

Here we display the parameter list and the leading comment lines of the double precision routine DTGSNA.

```
      SUBROUTINE DTGSNA( JOB, HOWMNY, SELECT, N, A, LDA, B, LDB,
     $                   VL, LDVL, VR, LDVR, S, DIF, MM, M,
     $                   WORK, LWORK, IWORK, INFO )
       IMPLICIT NONE
*
*     --- (preliminary version) ---
*     Bo Kagstrom and Peter Poromaa, Institute of Information Processing,
*     Univ. of Umea, S-901 87 Sweden.
*     May 1994
*
*
*     .. Scalar Arguments ..
      CHARACTER          HOWMNY, JOB
      INTEGER            INFO, LDA, LDB, LDVL, LDVR, LWORK,
     $                   M, MM, N
*     ..
```

```
*     .. Array Arguments ..
      LOGICAL            SELECT( * )
      INTEGER            IWORK( * )
      DOUBLE PRECISION   S( * ), DIF( * ), A( LDA, * ), B(LDB, *),
     $                   VL( LDVL, * ), VR( LDVR, * ), WORK( * )
*     ..
*
*  Purpose
*  =======
*
*  DTGSNA estimates reciprocal condition numbers for specified
*  eigenvalues and/or eigenvectors of a matrix pair (A, B) in
*  generalized real Schur canonical form (or of any matrix pair
*  (Q*A*Z**T, Q*B*Z**T) with Q and Z orthogonal).
*
*  (A, B) must be in generalized real Schur form (as returned by DGEGS), i.e.
*  A is block upper triangular with 1-by-1 and 2-by-2 diagonal blocks.
*  B is upper triangular.
*
*  References
*  ==========
*
*  [1] B. Kagstrom; A Direct Method for Reordering Eigenvalues in the
*      Generalized Real Schur Form of a Regular Matrix Pair (A, B), in
*      M.S. Moonen et al (eds), Linear Algebra for Large Scale and
*      Real-Time Applications, Kluwer Academic Publ. 1993, pp 195-218.
*
*  [2] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified
*      Eigenvalues of a Regular Matrix Pair (A, B) and Condition
*      Estimation: Theory, Algorithms and Software, Report UMINF - 94.04,
*      Inst. of Information Processing, University of Umea, S-901 87 Umea,
*      Sweden, February 1994. (also published as LAPACK Working Note xx)
*
*  [3] B. Kagstrom and P. Poromaa, LAPACK-Style Algorithms and Software for
*      Solving the Generalized Sylvester Equation and Estimating the Separa-
*      tion between Regular Matrix Pairs, Report UMINF - 93.23, Inst. of
*      Information Processing, University of Umea, S-901 87 Umea, Sweden,
*      November 1993.(also published as LAPACK Working Note xx)
*
*  Arguments
*  =========
*
*  JOB     (input) CHARACTER*1
*          Specifies whether condition numbers are required for
*          eigenvalues (S) or eigenvectors (DIF):
*          = 'E': for eigenvalues only (S);
*          = 'V': for eigenvectors only (DIF);
*          = 'B': for both eigenvalues and eigenvectors (S and DIF).
*
*  HOWMNY  (input) CHARACTER*1
*          = 'A': compute condition numbers for all eigenpairs;
```

```
*          = 'S': compute condition numbers for selected eigenpairs
*                 specified by the array SELECT.
*
*  SELECT  (input) LOGICAL array, dimension (N)
*          If HOWMNY = 'S', SELECT specifies the eigenpairs for which
*          condition numbers are required. To select condition numbers
*          for the eigenpair corresponding to a real eigenvalue w(j),
*          SELECT(j) must be set to .TRUE.. To select condition numbers
*          corresponding to a complex conjugate pair of eigenvalues w(j)
*          and w(j+1), either SELECT(j) or SELECT(j+1) or both, must be
*          set to .TRUE..
*          If HOWMNY = 'A', SELECT is not referenced.
*
*  N       (input) INTEGER
*          The order of the square matrix pair (A, B). N >= 0.
*
*  A       (input) DOUBLE PRECISION array, dimension (LDA,N)
*          The upper quasi-triangular matrix A, where (A, B) is in
*          generalized Schur canonical.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A. LDA >= max(1,N).
*
*  B       (input) DOUBLE PRECISION array, dimension (LDB,N)
*          The upper triangular matrix B, where (A, B) is in
*          generalized Schur canonical.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B. LDB >= max(1,N).
*
*  VL      (input) DOUBLE PRECISION array, dimension (LDVL,M)
*          If JOB = 'E' or 'B', VL must contain left eigenvectors of (A, B),
*          corresponding to the eigenpairs specified by HOWMNY and SELECT.
*          The eigenvectors must be stored in consecutive columns of VL, as
*          returned by DTGEVC or DGEGV. If job = 'V', VL is not referenced.
*
*  LDVL    (input) INTEGER
*          The leading dimension of the array VL.
*          LDVL >= 1; and if JOB = 'E' or 'B', LDVL >= N.
*
*  VR      (input) DOUBLE PRECISION array, dimension (LDVR,M)
*          If JOB = 'E' or 'B', VR must contain right eigenvectors of (A, B),
*          corresponding to the eigenpairs specified by HOWMNY and SELECT.
*          The eigenvectors must be stored in consecutive columns ov VR, as
*          returned by DTGEVC or DGEGV. If job = 'V', VR is not referenced.
*
*  LDVR    (input) INTEGER
*          The leading dimension of the array VR.
*          LDVR >= 1; and if JOB = 'E' or 'B', LDVR >= N.
*
*  S       (output) DOUBLE PRECISION array, dimension (MM)
```

```
*          If JOB = 'E' or 'B', the reciprocal condition numbers of the
*          selected eigenvalues, stored in consecutive elements of the
*          array. For a complex conjugate pair of eigenvalues two
*          consecutive elements of S are set to the same value. Thus
*          S(j), DIF(j), and the j-th columns of VL and VR all
*          correspond to the same eigenpair (but not in general the
*          j-th eigenpair, unless all eigenpairs are selected).
*          If JOB = 'V', S is not referenced.
*
*  DIF     (output) DOUBLE PRECISION array, dimension (MM)
*          If JOB = 'V' or 'B', the estimated reciprocal condition
*          numbers of the selected eigenvectors, stored in consecutive
*          elements of the array. For a complex eigenvector two
*          consecutive elements of DIF are set to the same value. If
*          the eigenvalues cannot be reordered to compute DIF(j), DIF(j)
*          is set to 0; this can only occur when the true value would be
*          very small anyway.
*          If JOB = 'E', DIF is not referenced.
*          For each eigenvalue/vector specified by SELECT, DIF() stores a
*          Frobenius norm-based estimate of Difl.
*
*  MM      (input) INTEGER
*          The number of elements in the arrays S and DIF. MM >= M.
*
*  M       (output) INTEGER
*          The number of elements of the arrays S and DIF used to store
*          the specified condition numbers; for each selected real
*          eigenvalue one element is used, and for each selected complex
*          conjugate pair of eigenvalues, two elements are used. If
*          HOWMNY = 'A', M is set to N.
*
*  WORK    (workspace) DOUBLE PRECISION array, dimension (LWORK)
*
*  LWORK  (input) INTEGER
*          The dimension of the array WORK. LWORK >= 2*N*(N+2)+32.
*
*  IWORK   (workspace) INTEGER array, dimension (N + 6)
*          If JOB = 'E', IWORK is not referenced.
*
*  INFO    (output) INTEGER
*          = 0: successful exit
*          < 0: if INFO = -i, the i-th argument had an illegal value
*
```

E Calling sequence DGSRBB

Here we display the parameter list and the leading comment lines of the double precision routine DGSRBB

```
      SUBROUTINE DGSRBB( INPUTS, N, A, LDA, B, LDB, C, LDC, D, LDD,
     $                   Q, LDQ, Z, LDZ, M, RBB, CNDTN, DIF, RRES, WORK,
```

```
     $                   LWORK, IWORK, LIWORK, INFO )
*
       IMPLICIT NONE
*     --- (preliminary version) ---
*     Bo Kagstrom and Peter Poromaa, Institute of Information Processing,
*     Univ. of Umea, S-901 87 Sweden.
*     Jan. 1994
*
*
*     .. Scalar Arguments ..
      CHARACTER          INPUTS
      INTEGER            N, LDA, LDB, LDC, LDD, LDQ, LDZ, M,
     $                   INFO, LIWORK, LWORK
      DOUBLE PRECISION   DIF, CNDTN, RBB, RRES
*     ..
*     .. Array Arguments ..
      INTEGER            IWORK(*)
      DOUBLE PRECISION   Q( LDQ, * ), A( LDA, * ), B(LDB, *),
     $                   Z(LDZ, *), C(LDC, *), D(LDD, *), WORK(*)
*     ..
*
*  Purpose
*  =======
*
*  DGSRBB computes an algorithm-independent residual-based error bound for
*  left and right eigenspaces (deflating subspaces) of a matrix pair (C, D) =
*  Q' * (A, B) * Z, where Q and Z transform the original matrix pair (A, B) to
*  generalized real Schur canonical form such that the M-by-M (1,1)-block
*  of (C, D) holds a selected cluster of eigenvalues. Such a decomposition
*  can be computed using DGEGS (generalized Schur) and DTGEXC (eigenvalue
*  reordering). The leading M columns of Q and Z form an orthonormal basis
*  for the selected pair of left and right eigenspaces (deflating subspaces).
*
*  (C, D) in generalized Schur form means that C is block upper triangular with
*  1-by-1 and 2-by-2 diagonal blocks. D is upper triangular.
*
*  References
*  ==========
*
*  [1] B. Kagstrom and P. Poromaa; Computing Eigenspaces with Specified
*      Eigenvalues of a Regular Matrix Pair (A, B) and Condition
*      Estimation: Theory, Algorithms and Software, Report UMINF - 94.04,
*      Inst. of Information Processing, University of Umea, S-901 87 Umea,
*      Sweden, February 1994. (also published as LAPACK Working Note xx)
*
*  [2] J-G. Sun; Backward Perturbation Analysis of Certain Characteristic
*      Subspaces, Numerische Mathematik, 65 (1993), pp 357-382.
*
*  Arguments
*  =========
*
```

```
*  INPUTS  (input) CHARACTER
*
*          'N': The values of DIF and (C, D) are ignored and computed by this
*               routine.
*          'D': On input DIF must hold the value of Difl[(C11, D11),(C22, D22)]
*               or an estimate of Difl (e.g., computed by DTGSYL, DTGSYX or DTGSEN).
*          'M': On input C and D must be the transformed matrices
*               C = Q'*A*Z, D = Q'*B*Z, where the (2,1)-blocks may have been set
*               to zero by some algorithm.
*          'B': On input the values of DIF (see above) and (C,D) are all supplied.
*
*
*  N       (input) INTEGER
*          The order of the matrices A and B. N >= 0.
*
*  A       (input/output) DOUBLE PRECISION array, dimension(LDA,N)
*          On entry, the general matrix A.
*          On exit: if INPUTS = 'N', A  = Q'*A*Z, else A is unchanged.
*
*  LDA     (input) INTEGER
*          The leading dimension of the array A. LDA >= max(1,N).
*
*  B       (input/output) DOUBLE PRECISION array, dimension(LDB,N)
*          On entry, the general matrix B.
*          On exit: if INPUTS = 'N', B  = Q'*B*Z, else B is unchanged.
*
*  LDB     (input) INTEGER
*          The leading dimension of the array B. LDB >= max(1,N).
*
*  C       (input/output) DOUBLE PRECISION array, dimension(LDC, N)
*          On entry, if INPUTS = 'M' or 'B', C = Q'*A*Z, where the (2,1)-block
*          may have been set to zero by some algorithm.
*          Note: If A and C are the same formal parameters and INPUTS = 'N',
*          C = A will be overwritten by  Q'*A*Z, else C is unchanged.
*
*  LDC     (input) INTEGER
*          The leading dimension of the array C. LDC >= 1.
*          If INPUTS = 'M' or 'B', LDC >= N.
*
*  D       (input/output) DOUBLE PRECISION array, dimension(LDD, N)
*          On entry, if INPUTS = 'M' or 'B', D = Q'*B*Z, where the (2,1)-block
*          can have been set to zero by some algorithm.
*          Note: If B and D are the same formal parameters and INPUTS = 'N',
*          D = B will be overwritten by  Q'*B*Z, else D is unchanged.
*
*  LDD     (input) INTEGER
*          The leading dimension of the array D. LDD >= 1.
*          If INPUTS = 'M' or 'B', LDD >= N.
*
*  Q       (input) DOUBLE PRECISION array, dimension (LDQ,N)
*  Z       (input) DOUBLE PRECISION array, dimension (LDZ,N)
```

```
*          On entry, Q and Z are the orthogonal transformation matrices
*          which reorder (A, B) with the selected eigenvalues in the
*          (1,1)-block of (C, D); Q'*(A, B)*Z = (C, D). The leading M
*          columns of Q and Z form an orthonormal basis for the specified
*          pair of left and right eigenspaces (deflating subspaces).
*
*  LDQ     (input) INTEGER
*          The leading dimension of the array Q.
*          LDQ >= N.
*
*  LDZ     (input) INTEGER
*          The leading dimension of the array Z.
*          LDZ >= N.
*
*  M       (input) INTEGER
*          The dimension of the specified pair of deflating subspaces.
*
*  RBB    (output) DOUBLE PRECISION
*          On exit, an approximate residual-based error bound for
*          selected left and right eigenspaces (deflating subspaces).
*          See further details.
*
*  CNDTN   (output) DOUBLE PRECISION
*          On exit, the value of the restriction on the perturbations, which
*          should be less than 1 for the bound RBB to hold. See further details.
*
*  DIF    (input/output) DOUBLE PRECISION
*          On entry, if INPUTS = 'D' or 'B', DIF = Difl[(C11, D11),(C22, D22)].
*          This could be an estimate (e.g., computed by DTGSYL, DTGSYX or DTGSEN).
*          On exit, if INPUTS = 'N' or 'M', DIF = an estimate of
*          Difl[(C11, D11),(C22, D22)] computed by DTGSYL.
*          If INPUTS = 'D' or 'B', DIF is not changed.
*
*  RRES    (output) DOUBLE PRECISION
*          On exit, the Frobenius norm of the (2,1)-blocks of
*          (C, D) = Q'*(A, B)*Z, i.e. the norm of the optimal
*          backward error corresponding to the computed pair of
*          deflating subspaces (the first M columns of Q and Z).
*
*  WORK    (workspace) DOUBLE PRECISION array, dimension (LWORK)
*
*  LWORK   (input) INTEGER
*          The dimension of the array WORK. LWORK >= N*N.
*
*  IWORK   (workspace) INTEGER, dimension (LIWORK)
*          If INPUTS = 'D' or 'B', IWORK is not referenced.
*
*  LIWORK  (input) INTEGER
*          The dimension of the array IWORK. LIWORK >= 1.
*          If INPUTS = 'N' or 'M', LIWORK >= N + 6;
*
```

```
*  INFO    (output) INTEGER
*            0: Successful exit
*          < 0: If INFO = -i, the i-th argument had an illegal value
*            1: CNDTN > 1, see further details.
*
```