ALGORITHM 590
DSUBSP and **EXCHQZ**: FORTRAN Subroutines for Computing Deflating Subspaces with Specified Spectrum

P. VAN DOOREN
Stanford University

Categories and Subject Descriptors: G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra—*eigenvalues*; G.m [**Mathematics of Computing**]: Miscellaneous—*FORTRAN*

General Terms: Algorithms

Additional Key Words and Phrases: Generalized eigenvalue, *QZ* algorithm

---

1. DESCRIPTION

A reliable and widely available method to compute the generalized eigenvalues of an $n \times n$ real regular (i.e., invertible) pencil $\lambda B - A$ is the so-called QZ-algorithm [1, 3]. This algorithm constructs orthogonal row and column transformations Q_1 and Z_1 such that the transformed pencil

$$\lambda B_1 - A_1 = Q_1(\lambda B - A)Z_1 \tag{1}$$

is in "quasi-triangular" form, that is, with B_1 in upper triangular form and A_1 in block triangular Hessenberg form with 1×1 and 2×2 diagonal blocks, as illustrated below:

$$B_1 = \begin{bmatrix} x & x & x & x & x & x \\ & x & x & x & x & x \\ & & x & x & x & x \\ & & & x & x & x \\ & & & & x & x \\ & & & & & x \end{bmatrix}, \quad A_1 = \begin{bmatrix} x & x & x & x & x & x \\ & x & x & x & x & x \\ & x & x & x & x & x \\ & & & x & x & x \\ & & & & x & x \\ & & & & x & x \end{bmatrix}. \tag{2}$$

The 1×1 diagonal pencils of $\lambda B_1 - A_1$ contain the real generalized eigenvalues of the pencil $\lambda B - A$, and the 2×2 diagonal pencils of $\lambda B_1 - A_1$ contain the complex generalized eigenvalues, a conjugate pair to each 2×2 pencil.

Received 11 November 1981; revised 2 June 1982; accepted 2 August 1982
This work was supported by the National Science Foundation under Grant ENG78-10003 and by the U.S. Air Force under Grant AFOSR-79-0094.
Author's present address: Philips Research Laboratory, Avenue Van Becelaere 2, Box 9, B-1170, Brussels, Belgium.

The FORTRAN subroutines **DSUBSP** and **EXCHQZ** allow one to update the decomposition (1) by postmultiplying Z_1 by Z_2 and premultiplying Q_1 by Q_2 such that

$$\lambda B_2 - A_2 = Q_2 Q_1 (\lambda B - A) Z_1 Z_2 \tag{3}$$

is still in quasi-triangular form, but in addition, a specific ordering of generalized eigenvalues is obtained in this form. This is important in several applications [6, 7]. Indeed when the generalized eigenvalues $\lambda_1, \ldots, \lambda_n$ are ordered such that $\lambda_1, \ldots, \lambda_k$ are inside a region Γ and $\lambda_{k+1}, \ldots, \lambda_n$ are outside this region, then the space spanned by the first k orthonormal columns of $Z = Z_1 Z_2$ is the deflating subspace [4] of $\lambda B - A$ corresponding to the spectrum inside Γ. In several applications, deflating subspaces have to be computed for different regions Γ. Of course, Γ has to be symmetric with respect to the real axis, since complex pairs of eigenvalues have to be categorized both outside or inside Γ in order for this deflating subspace to be real. No further assumptions have to be imposed on Γ: it can be open or not, connected or not.

The user has to provide a function describing the region Γ by testing whether the generalized eigenvalues of a 1×1 or 2×2 (diagonal) pencil lies inside or outside the region Γ. This function must be of the type

INTEGER FUNCTION FTEST (LSIZE, ALPHA, BETA, S, P)

with parameters:

LSIZE	An integer containing the size of the considered pencil (1 or 2).
ALPHA,BETA	Two real variables. In case **LSIZE**=1, the generalized eigenvalue of the considered pencil is given by **ALPHA/BETA**, which may be infinite when **BETA**=0.
S, P	Two real variables. In case **LSIZE**=2, they contain the sum and product of the two complex conjugate generalized eigenvalues of the considered pencil.
FTEST	The function value, which is put equal to 1 when the generalized eigenvalue(s) of the considered pencil is (are) inside the specified region Γ, and equal to -1 otherwise.

Simple examples for such routines are given by the functions **FIN**, **FOUT**, **FOLHP**, and **FCRHP**, describing the regions inside and outside the unit circle, the open left half-plane, and the closed right half-plane, respectively. Their listings are included below as templates for the user.

This routine is then used as parameter for the subroutine **DSUBSP**, which reorders the 1×1 and 2×2 diagonal pencils of the quasi-triangular form $\lambda B_1 - A_1$ such that those with generalized eigenvalues inside Γ appear first. The calling sequence for **DSUBSP** is

CALL DSUBSP (NMAX, N, A, B, Z, FTEST, EPS, NDIM, FAIL, IND)

with (parameters preceded by an asterisk are altered by the subroutine):

NMAX	An integer containing the first dimension of the arrays **A**, **B**, and **Z**.
N	An integer containing the current order of **A**, **B**, and **Z**.

;axA,*B Doubly subscripted real arrays containing the pencil to be reordered. On return, $\lambda B - A$ contains the final quasi-triangular pencil, rearranged with respect to the region Γ specified by **FTEST**.

***Z** A doubly subscripted real array into which the reducing column transformation is postmultiplied.

FTEST The integer function provided by the user to describe the region Γ of interest.

EPS A real number used as the convergence criterion. Maximal accuracy is obtained when **EPS** is set equal to relpr $\times$ $\max\{\|\mathbf{A}\|_2, \|\mathbf{B}\|_2\}$, where relpr is the relative precision of the computer used. Smaller values of **EPS** will increase the amount of work without significantly improving the accuracy.

***NDIM** An integer giving the dimension of the computed deflating subspace.

***FAIL** A logical variable that on normal return is **.FALSE.** If the iterative part of the algorithm does not converge, **FAIL** is set to **.TRUE.**.

*IND An integer working array of dimension at least **N**.

DSUBSP is to be used together with the EISPACK programs **QZHES**, **QZIT**, and **QZVAL** [1] to reduce a full pencil $\lambda B - A$ to quasi-triangular form with the eigenvalues inside the contour Γ appearing first on diagonal. (For the explanation of the parameters **EPS1**, **IERR**, **ALPHAR**, **ALPHAI**, and **BETA** in these routines, see [1].):

```
CALL QZHES (NMAX, N, A, B, .TRUE., Z)
CALL QZIT (NMAX, N, A, B, EPS1, .TRUE., Z, IERR)
CALL QZVAL (NMAX, N, A, B, ALPHAR, ALPHAI, BETA, .TRUE., Z)
CALL DSUBSP (NMAX, N, A, B, Z, FTEST, EPS, NDIM, FAIL, IND)
```

Besides the function **FTEST**, describing the region Γ of interest, **DSUBSP** also uses the subroutines **EXCHQZ**, **GIV**, and **SROT**. **EXCHQZ** is a FORTRAN subroutine to interchange two adjacent (1×1 and/or 2×2) pencils of a quasi-triangular form. Specifically, it is supposed that A has a block of order $l1$ starting at the lth diagonal element, and a block of order $l2$ starting at the $(l + l1)$th diagonal element (illustrated below for $n = 5$, $l = 2$, $l1 = 2$, $l2 = 1$):

$$B = \begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix}, \quad A = \begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & x & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix}. \tag{4}$$

EXCHQZ constructs orthogonal row and column transformations V and W such that $V(\lambda B - A)W$ has consecutive blocks of order $l2$ and $l1$ at the lth diagonal element (illustrated for the example above):

$$VBW = \begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix} \qquad VAW = \begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & x & x & x \\ & & & & x \end{bmatrix}. \tag{5}$$

The generalized eigenvalues associated with each diagonal block are interchanged along with the blocks. The column transformation W is postmultiplied into the array **Z**; the row transformation V is not stored since it is not required in the computation of bases of deflating subspaces.

The calling sequence for **EXCHQZ** is

CALL EXCHQZ (NMAX, N, A, B, Z, L, LS1, LS2, EPS, FAIL)

with (parameters preceded by an asterisk are altered by the subroutine):

NMAX	An integer containing the first dimension of **A, B**, and **Z**.
N	An integer containing the current order of **A, B**, and **Z**.
***A,*B**	Doubly subscripted real arrays containing the pencil to be reordered.
***Z**	A doubly subscripted real array into which the updating column transformation is postmultiplied.
L	An integer containing the leading diagonal position of the first block to be interchanged.
LS1	An integer containing the size of the first block.
LS2	An integer containing the size of the second block.
EPS	A convergence criterion (cf. **EPS** in the calling sequence of **DSUBSP**).
***FAIL**	A logical variable that on normal return is **.FALSE.**. If the iterative part of the algorithm does not converge, **FAIL** is set to **.TRUE.**.

EXCHQZ requires the subroutines **GIV** and **SROT**, which are elementary routines that construct and perform Givens rotations on column or rows. **SROT** is a BLA subroutine and **GIV** is a modification of **SROTG** of the BLAS package [2] in order to allow a more compact code for **EXCHQZ**, which contains several calls to these routines.

2. METHOD AND PROGRAMMING DETAILS

The subroutine **DSUBSP** makes a first pass through the diagonal blocks of the quasi-triangular form $\lambda B - A$ in order to determine the sizes of the diagonal blocks and the locations of their generalized eigenvalues with respect to Γ. During this pass the integer vector **IND(•)** is created with entries ± 1 or ± 2. Here **sign** [**IND(I)**] refers to the location of the generalized eigenvalues of block **I** with respect to Γ (+ for inside Γ, − for outside Γ) and **abs**[**IND(I)**] indicates the size of this block. The entries of this vector are then rearranged such that the plus signs appear first. This is done using a "bubble sort," that is, each time a plus sign follows a minus sign, it is moved in front of all its preceding minus signs via consecutive permutations. Each permutation, of course, involves an interchanging of two consecutive blocks using the subroutine **EXCHQZ**.

EXCHQZ works in a fashion similar to the routine **EXCHNG**, developed for the reordering of the standard eigenvalue problem [5] (this is also the reason why an apparent parallelism with that paper has been pursued here). To interchange two consecutive blocks where at least one of them has order 1, a shift is performed to the real eigenvalue of a 1×1 block (in [5], this was only done for the interchange of two 1×1 blocks). Givens rotations are then constructed in a straightforward manner to interchange the shifted zero eigenvalue to the other block. The construction of the Givens transformations needed for the exchange of the blocks is done in such a way as to ensure backward stability.

In the case of two 2×2 blocks, an arbitrary *QZ*-step is performed on both blocks in order to eliminate the uncoupling between them. Then a sequence of *QZ*-steps using a previously determined shift is performed on both blocks. A decoupling with the blocks in the desired order is usually obtained with a few steps (rarely more than two). If within 30 iterations no decoupling is obtained, the subroutine gives an error return. The criterion used is that the coupling element of the two blocks in the Hessenberg form of A is smaller than **EPS**. Since it does not make sense to force this coupling element to be smaller than the errors in the rest of the pencil, one should not choose **EPS** smaller than relpr $\times \max\{\|A\|_2, \|B\|_2\}$, where relpr is the relative precision of the computer used. A good choice for **EPS** is the "estimated" absolute precision of the (sometimes measured) data in A and B. Since the pencils used here are of dimension 4×4, the *QZ* steps are implemented with Givens transformations instead of Householder transformations, which turns out to be economical. More details are given in [7], where a proof of the backward stability of the method is also given.

EXCHQZ uses the routines **GIV** and **SROT**, which construct a 2×2 Givens rotation to zero out an element of a 2-vector, and perform it, respectively, on two columns or rows of a specified matrix.

ACKNOWLEDGMENTS

The valuable comments of Sven Hammarling and the helpful guidance distilled from Algorithms 506 and 539 are gratefully acknowledged.

REFERENCES

1. GARBOW, B.S., BOYLE, J.M., DONGARRA, J.J., AND MOLER, C.B. Matrix Eigensystem Routines—EISPACK Guide Extension. In *Lecture Notes in Computer Science*, vol. 51, Springer, New York, 1977.
2. LAWSON, C.L., HANSON, R.J., KINCAID, D.R., AND KROGH, F.T. Basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Softw. 5*, 3 (Sept. 1979), 324–325.
3. MOLER, C.B., AND STEWART, G.W. An algorithm for the generalized eigenvalue problem. *SIAM J. Numer. Anal. 10* (1973), 241–256.
4. STEWART, G.W. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Rev. 15* (1973), 727–764.
5. STEWART, G.W. HQR3 and EXCHNG. Fortran subroutines for calculating and ordering the

eigenvalues of a real upper Hessenberg matrix. *ACM Trans. Math. Softw. 2*, 3 (Sept. 1976), 275–280.

6. Van Dooren, P. The generalized eigenstructure problem in linear system theory. *IEEE Trans. Automat. Contr. AC-26* (1981), 111–129.
7. Van Dooren, P. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Stat. Sci. Comput. 2* (1981), 121–135.

ALGORITHM

[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 409 for order form).]

```
      SUBROUTINE DSUBSP(NMAX, N, A, B, Z, FTEST, EPS, NDIM, FAIL, IND)
      INTEGER NMAX, N, FTEST, NDIM, IND(N)
      LOGICAL FAIL
      REAL A(NMAX,N), B(NMAX,N), Z(NMAX,N), EPS
C*
C* GIVEN THE UPPER TRIANGULAR MATRIX B AND UPPER HESSENBERG MATRIX A
C* WITH 1X1 OR 2X2 DIAGONAL BLOCKS, THIS ROUTINE REORDERS THE DIAGONAL
C* BLOCKS ALONG WITH THEIR GENERALIZED EIGENVALUES BY CONSTRUCTING EQUI-
C* VALENCE TRANSFORMATIONS QT AND ZT. THE ROW TRANSFORMATION ZT IS ALSO
C* PERFORMED ON THE GIVEN (INITIAL) TRANSFORMATION Z (RESULTING FROM A
C* POSSIBLE PREVIOUS STEP OR INITIALIZED WITH THE IDENTITY MATRIX).
C* AFTER REORDERING, THE EIGENVALUES INSIDE THE REGION SPECIFIED BY THE
C* FUNCTION FTEST APPEAR AT THE TOP. IF NDIM IS THEIR NUMBER THEN THE
C* NDIM FIRST COLUMNS OF Z SPAN THE REQUESTED SUBSPACE. DSUBSP REQUIRES
C* THE SUBROUTINE EXCHQZ AND THE INTEGER FUNCTION FTEST WHICH HAS TO BE
C* PROVIDED BY THE USER. THE PARAMETERS IN THE CALLING SEQUENCE ARE :
C* (STARRED PARAMETERS ARE ALTERED BY THE SUBROUTINE)
C*
C*    NMAX     THE FIRST DIMENSION OF A, B AND Z
C*    N        THE ORDER OF A, B AND Z
C*   *A,*B     THE MATRIX PAIR WHOSE BLOCKS ARE TO BE REORDERED.
C*   *Z        UPON RETURN THIS ARRAY IS MULTIPLIED BY THE COLUMN
C*             TRANSFORMATION ZT.
C*    FTEST(LS,ALPHA,BETA,S,P) AN INTEGER FUNCTION DESCRIBING THE
C*             SPECTRUM OF THE DEFLATING SUBSPACE TO BE COMPUTED:
C*             WHEN LS=1 FTEST CHECKS IF ALPHA/BETA IS IN THAT SPECTRUM
C*             WHEN LS=2 FTEST CHECKS IF THE TWO COMPLEX CONJUGATE
C*             ROOTS WITH SUM S AND PRODUCT P ARE IN THAT SPECTRUM
C*             IF THE ANSWER IS POSITIVE, FTEST=1, OTHERWISE FTEST=-1
C*    EPS      THE REQUIRED ABSOLUTE ACCURACY OF THE RESULT
C*   *NDIM     AN INTEGER GIVING THE DIMENSION OF THE COMPUTED
C*             DEFLATING SUBSPACE
C*   *FAIL     A LOGICAL VARIABLE WHICH IS FALSE ON A NORMAL RETURN,
C*             TRUE OTHERWISE (WHEN EXCHQZ FAILS)
C*   *IND      AN INTEGER WORKING ARRAY OF DIMENSION AT LEAST N
C*
      INTEGER L, LS, LS1, LS2, L1, LL, NUM, IS, L2I, L2K, I, K, II,
     * ISTEP, IFIRST
      REAL S, P, D, ALPHA, BETA
      FAIL = .TRUE.
      NDIM = 0
      NUM = 0
      L = 0
      LS = 1
C*** CONSTRUCT ARRAY IND(I) WHERE :
C***     IABS(IND(I)) IS THE SIZE OF THE BLOCK I
```

```
C***     SIGN(IND(I)) INDICATES THE LOCATION OF ITS EIGENVALUES
C***                  (AS DETERMINED BY FTEST).
C*** NUM IS THE NUMBER OF ELEMENTS IN THIS ARRAY
      DO 30 LL=1,N
        L = L + LS
        IF (L.GT.N) GO TO 40
        L1 = L + 1
        IF (L1.GT.N) GO TO 10
        IF (A(L1,L).EQ.0.) GO TO 10
C* HERE A 2X2  BLOCK IS CHECKED *
        LS = 2
        D = B(L,L)*B(L1,L1)
        S = (A(L,L)*B(L1,L1)+A(L1,L1)*B(L,L)-A(L1,L)*B(L,L1))/D
        P = (A(L,L)*A(L1,L1)-A(L,L1)*A(L1,L))/D
        IS = FTEST(LS,ALPHA,BETA,S,P)
        GO TO 20
C* HERE A 1X1  BLOCK IS CHECKED *
   10   LS = 1
        IS = FTEST(LS,A(L,L),B(L,L),S,P)
   20   NUM = NUM + 1
        IF (IS.EQ.1) NDIM = NDIM + LS
        IND(NUM) = LS*IS
   30 CONTINUE
C***  REORDER BLOCKS SUCH THAT THOSE WITH POSITIVE VALUE
C***    OF IND(.) APPEAR FIRST.
```