NAG Fortran Library Routine Document

F08XEF (SHGEQZ/DHGEQZ)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F08XEF (SHGEQZ/DHGEQZ) implements the QZ method for finding generalized eigenvalues of the real matrix pair (A, B) of order n, which is in the generalized upper Hessenberg form.

2 Specification

```
 SUBROUTINE F08XEF(JOB, COMPQ, COMPZ, N, ILO, IHI, A, LDA, B, LDB,
1                  ALPHAR, ALPHAI, BETA, Q, LDQ, Z, LDZ, WORK, LWORK,
2                  INFO)
 ENTRY      shgeqz (JOB, COMPQ, COMPZ, N, ILO, IHI, A, LDA, B, LDB,
1                  ALPHAR, ALPHAI, BETA, Q, LDQ, Z, LDZ, WORK, LWORK,
2                  INFO)
 INTEGER           N, ILO, IHI, LDA, LDB, LDQ, LDZ, LWORK, INFO
 real              A(LDA,*), B(LDB,*), ALPHAR(*), ALPHAI(*), BETA(*),
1                  Q(LDQ,*), Z(LDZ,*), WORK(*)
 CHARACTER*1       JOB, COMPQ, COMPZ
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

F08XEF (SHGEQZ/DHGEQZ) implements a single-double-shift version of the QZ method for finding the generalized eigenvalues of the real matrix pair (A, B) which is in the generalized upper Hessenberg form. If the matrix pair (A, B) is not in the generalized upper Hessenberg form, then the routine F08WEF (SGGHRD/DGGHRD) should be called before invoking F08XEF (SHGEQZ/DHGEQZ).

This problem is mathematically equivalent to solving the equation

$$\det(A - \lambda B) = 0.$$

Note that, to avoid underflow, overflow and other arithmetic problems, the generalized eigenvalues λ_j are never computed explicitly by this routine but defined as ratios between two computed values, α_j and β_j:

$$\lambda_j = \alpha_j / \beta_j.$$

The parameters α_j, in general, are finite complex values and β_j are finite real non-negative values.

If desired, the matrix pair (A, B) may be reduced to generalized Schur form. That is, the transformed matrix B is upper triangular and the transformed matrix A is block upper triangular, where the diagonal blocks are either 1 by 1 or 2 by 2. The 1 by 1 blocks provide generalized eigenvalues which are real and the 2 by 2 blocks give complex generalized eigenvalues.

The parameter JOB specifies two options. If JOB = 'S' then the matrix pair (A, B) is simultaneously reduced to Schur form by applying one orthogonal transformation (usually called Q) on the left and another (usually called Z) on the right. That is,

$$\begin{array}{l} A \leftarrow Q^T A Z \\ B \leftarrow Q^T B Z \end{array}$$

The 2 by 2 upper-triangular diagonal blocks of B corresponding to 2 by 2 blocks of A will be reduced to non-negative diagonal matrices. That is, if A$(j+1, j)$ is non-zero, then B$(j+1, j) = $ B$(j, j+1) = 0$ and B(j, j) and B$(j+1, j+1)$ will be non-negative.

If JOB = 'E', then at each iteration, the same transformations are computed, but they are only applied to those parts of A and B which are needed to compute α and β. This option could be used if generalized eigenvalues are required but not generalized eigenvectors.

If JOB = 'S' and COMPQ and COMPZ are 'V' or 'I', then the orthogonal transformations used to reduce the pair (A, B) are accumulated into the input arrays Q and Z. If generalized eigenvectors are required then JOB must be set to 'S' and if left (right) generalized eigenvectors are to be computed then COMPQ(COMPZ) must be set to 'V' or 'I' and not 'N'.

If COMPQ is set to 'I' then eigenvectors are accumulated on the identity matrix and on exit the array Q contains the left eigenvector matrix Q. However, if COMPQ is set to 'S' then the transformations are accumulated on the user supplied matrix Q_0 in array Q on entry and thus on exit Q contains the matrix product QQ_0. A similar convention is used for COMPZ.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia

Moler C B and Stewart G W (1973) An algorithm for generalized matrix eigenproblems *SIAM J. Numer. Anal.* **10** 241–256

Golub G H and van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Stewart G W and Sun J-G (1990) *Matrix Perturbation Theory* Academic Press, London

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: specifies the operations to be performed on (A, B):

if JOB = 'E', the matrix pair (A, B) on exit might not be in the generalized Schur form;

if JOB = 'S', the matrix pair (A, B) on exit will be in the generalized Schur form.

Constraint: JOB = 'E' or 'S'.

2: COMPQ – CHARACTER*1 *Input*

On entry: specifies the operations to be performed on Q:

if COMPQ = 'N', the array Q is unchanged;

if COMPQ = 'V', the left transformation Q is accumulated on the array Q;

if COMPQ = 'I', the array Q is initialised to the identity matrix before the left transformation Q is accumulated in Q.

Constraint: COMPQ = 'N', 'V' or 'I'.

3: COMPZ – CHARACTER*1 *Input*

On entry: specifies the operations to be performed on Z:

if COMPZ = 'N', the array Z is unchanged;

if COMPZ = 'V', the right transformation Z is accumulated on the array Z;

if COMPZ = 'I', the array Z is initialised to the identity matrix before the right transformation Z is accumulated in Z.

Constraint: COMPZ = 'N', 'V' or 'I'.

4: N – INTEGER *Input*

On entry: n, the order of the matrices A, B, Q and Z.

Constraint: $\mathrm{N} \geq 0$.

5: ILO – INTEGER *Input*
6: IHI – INTEGER *Input*

On entry: the indices i_{lo} and i_{hi}, respectively which define the upper triangular parts of A. The submatrices $\mathrm{A}(1 : i_{lo} - 1, 1 : i_{lo} - 1)$ and $\mathrm{A}(i_{hi} + 1 : n, i_{hi} + 1 : n)$ are then upper triangular. These parameters are provided by F08WHF (SGGBAL/DGGBAL) if the matrix pair was previously balanced; otherwise, $\mathrm{ILO} = 1$ and $\mathrm{IHI} = \mathrm{N}$.

Constraints:

$1 \leq \mathrm{ILO} \leq \mathrm{IHI} \leq \mathrm{N}$ if $\mathrm{N} > 0$;
$\mathrm{ILO} = 1$ and $\mathrm{IHI} = 0$ if $\mathrm{N} = 0$.

7: A(LDA,*) – ***real*** array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the n by n upper Hessenberg matrix A. The elements below the first subdiagonal must be set to zero. If JOB = 'S', the matrix pair (A, B) will be simultaneously reduced to generalized Schur form. If JOB = 'E', the 1 by 1 and 2 by 2 diagonal blocks of the matrix pair (A, B) will give generalized eigenvalues but the remaining elements will be irrelevant.

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08XEF (SHGEQZ/DHGEQZ) is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

9: B(LDB,*) – ***real*** array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, \mathrm{N})$.

On entry: the n by n upper triangular matrix B. The elements below the diagonal must be zero.

On exit: if JOB = 'S', the matrix pair (A, B) will be simultaneously reduced to generalized Schur form. If JOB = 'E', the 1 by 1 and 2 by 2 diagonal blocks of the matrix pair (A, B) will give generalized eigenvalues but the remaining elements will be irrelevant.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08XEF (SHGEQZ/DHGEQZ) is called.

Constraint: $\mathrm{LDB} \geq \max(1, \mathrm{N})$.

11: ALPHAR($*$) – ***real*** array *Output*

Note: the dimension of the array ALPHAR must be at least $\max(1, \mathrm{N})$.

On exit: the real parts of α_j, for $j = 1, \ldots, n$.

12: ALPHAI($*$) – ***real*** array *Output*

Note: the dimension of the array ALPHAI must be at least $\max(1, \mathrm{N})$.

On exit: the imaginary parts of α_j, for $j = 1, \ldots, n$.

13: BETA($*$) – ***real*** array *Output*

Note: the dimension of the array BETA must be at least $\max(1, \mathrm{N})$.

On exit: β_j, for $j = 1, \ldots, n$.

14: Q(LDQ,*) – ***real*** array *Input/Output*

Note: the second dimension of the array Q must be at least max(1, N) if COMPQ = 'V' or 'I'. If COMPQ = 'N', Q is not referenced.

On entry: if COMPQ = 'V', the matrix Q_0. The matrix Q_0 is usually the matrix Q returned by F08WEF (SGGHRD/DGGHRD). If COMPQ = 'N', Q is not referenced.

On exit: if COMPQ = 'V', Q contains the matrix product QQ_0; if COMPQ = 'I', Q contains the transformation matrix Q.

15: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08XEF (SHGEQZ/DHGEQZ) is called.

Constraints:

LDQ $\geq$ N if COMPQ = 'V' or 'I';
LDQ $\geq$ 1 if COMPQ = 'N'.

16: Z(LDZ,*) – ***real*** array *Input/Output*

Note: the second dimension of the array Z must be at least max(1, N) if COMPZ = 'V' or 'I'. If COMPZ = 'N', Z is not referenced.

On entry: if COMPZ = 'V', the matrix Z_0. The matrix Z_0 is usually the matrix Z returned by F08WEF (SGGHRD/DGGHRD). If COMPZ = 'N', Z is not referenced.

On exit: if COMPZ = 'V', Z contains the matrix product ZZ_0; if COMPZ = 'I', Z contains the transformation matrix Z.

17: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08XEF (SHGEQZ/DHGEQZ) is called.

Constraints:

LDZ $\geq$ N if COMPZ = 'V' or 'I';
LDZ $\geq$ 1 if COMPZ = 'N'.

18: WORK($*$) – ***real*** array *Workspace*

Note: the dimension of the array WORK must be at least max(1, LWORK).

On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimum performance.

19: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08XEF (SHGEQZ/DHGEQZ) is called unless LWORK = -1, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK.

Constraint: LWORK $\geq$ max(1, N) or LWORK = -1.

20: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If $1 \leq$ INFO $\leq$ N, the QZ iteration did not converge and the matrix pair (A, B) is not in the generalized Schur form at exit. However, if INFO $<$ N, then the computed α_i and β_i should be correct for $i =$ INFO $+ 1, \ldots,$ N.

If N $+ 1 \leq$ INFO $\leq 2 \times$ N, the computation of shifts failed and the matrix pair (A, B) is not in the generalized Schur form at exit. However, if INFO $< 2 \times$ N, then the computed α_i and β_i should be correct for $i =$ INFO $-$ N $+ 1, \ldots,$ N.

If INFO $> 2 \times$ N, then it indicates a variety of highly unusual failures.

7 Accuracy

Please consult section 4.11 of the LAPACK Users' Guide (Anderson *et al.* (1999)) and Chapter 6 of Stewart and Sun (1990), for more information.

8 Further Comments

This routine is the fifth step in the solution of the real generalized eigenvalue problem and is called after F08WEF (SGGHRD/DGGHRD).

The complex analogue of this routine is F08XSF (CHGEQZ/ZHGEQZ).

9 Example

The example program computes the α and β parameters, which defines the generalized eigenvalues, of the matrix pair (A, B) given by

$$A = \begin{pmatrix} 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 2.0 & 4.0 & 8.0 & 16.0 & 32.0 \\ 3.0 & 9.0 & 27.0 & 81.0 & 243.0 \\ 4.0 & 16.0 & 64.0 & 256.0 & 1024.0 \\ 5.0 & 25.0 & 125.0 & 625.0 & 3125.0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1.0 & 2.0 & 3.0 & 4.0 & 5.0 \\ 1.0 & 4.0 & 9.0 & 16.0 & 25.0 \\ 1.0 & 8.0 & 27.0 & 64.0 & 125.0 \\ 1.0 & 16.0 & 81.0 & 256.0 & 625.0 \\ 1.0 & 32.0 & 243.0 & 1024.0 & 3125.0 \end{pmatrix}.$$

This requires calls to five routines: F08WHF (SGGBAL/DGGBAL) to balance the matrix, F08AEF (SGEQRF/DGEQRF) to perform the QR factorization of B, F08AGF (SORMQR/DORMQR) to apply Q to A, F08WEF (SGGHRD/DGGHRD) to reduce the matrix pair to the generalized Hessenberg form and F08XEF (SHGEQZ/DHGEQZ) to compute the eigenvalues using the QZ algorithm.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08XEF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDB, LDQ, LDZ, LWORK
      PARAMETER        (NMAX=10,LDA=NMAX,LDB=NMAX,LDQ=1,LDZ=1,
     +                 LWORK=6*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, IHI, ILO, INFO, IROWS, J, JWORK, N
      CHARACTER        COMPQ, COMPZ, JOB
*     .. Local Arrays ..
      real             A(LDA,NMAX), ALPHAI(NMAX), ALPHAR(NMAX),
     +                 B(LDB,NMAX), BETA(NMAX), LSCALE(NMAX),
     +                 Q(LDQ,LDQ), RSCALE(NMAX), TAU(NMAX), WORK(LWORK),
     +                 Z(LDZ,LDZ)
*     .. External Subroutines ..
      EXTERNAL         sgeqrf, sggbal, sgghrd, shgeqz, sormqr, X04CAF
*     .. Intrinsic Functions ..
      INTRINSIC        NINT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08XEF Example Program Results'
*
*     Skip heading in data file
*
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        READ matrix A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*        READ matrix B from data file
*
         READ (NIN,*) ((B(I,J),J=1,N),I=1,N)
*
*        Balance matrix pair (A,B)
*
         JOB = 'B'
         CALL sggbal(JOB,N,A,LDA,B,LDB,ILO,IHI,LSCALE,RSCALE,WORK,INFO)
*
*        Matrix A after balancing
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,A,LDA,'Matrix A after balancing',
     +               IFAIL)
         WRITE (NOUT,*)
*
*        Matrix B after balancing
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,B,LDB,'Matrix B after balancing',
     +               IFAIL)
         WRITE (NOUT,*)
*
*         Reduce B to triangular form using QR
*
         IROWS = IHI + 1 - ILO
         CALL sgeqrf(IROWS,IROWS,B(ILO,ILO),LDB,TAU,WORK,LWORK,INFO)
*
*         Apply the orthogonal transformation to matrix A
*
         CALL sormqr('L','T',IROWS,IROWS,IROWS,B(ILO,ILO),LDB,TAU,
     +               A(ILO,ILO),LDA,WORK,LWORK,INFO)
```

```
*
*        Compute the generalized Hessenberg form of (A,B)
*
         COMPQ = 'N'
         COMPZ = 'N'
         CALL sgghrd(COMPQ,COMPZ,IROWS,1,IROWS,A(ILO,ILO),LDA,B(ILO,ILO)
     +               ,LDB,Q,LDQ,Z,LDZ,INFO)
*
*        Matrix A in generalized Hessenberg form
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,A,LDA,
     +               'Matrix A in Hessenberg form',IFAIL)
         WRITE (NOUT,*)
*
*        Matrix B in generalized Hessenberg form
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,B,LDB,'Matrix B is triangular',
     +               IFAIL)
*
*        Routine shgeqz
*        Workspace query: JWORK = -1
*
         JWORK = -1
         JOB = 'E'
         CALL shgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHAR,
     +               ALPHAI,BETA,Q,LDQ,Z,LDZ,WORK,JWORK,INFO)
         WRITE (NOUT,*)
         WRITE (NOUT,99999) NINT(WORK(1))
         WRITE (NOUT,99998) LWORK
         WRITE (NOUT,*)
*
*        Compute the generalized Schur form
*        if the workspace LWORK is adequate
*
         IF (NINT(WORK(1)).LE.LWORK) THEN
            CALL shgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHAR,
     +                  ALPHAI,BETA,Q,LDQ,Z,LDZ,WORK,LWORK,INFO)
*
*        Print the generalized eigenvalues
*
            WRITE (NOUT,99997)
*
            DO 20 I = 1, N
               IF (BETA(I).NE.0.0e0) THEN
                  WRITE (NOUT,99996) I, '(', ALPHAR(I)/BETA(I), ',',
     +              ALPHAI(I)/BETA(I), ')'
               ELSE
                  WRITE (NOUT,99994) I
               END IF
   20       CONTINUE
         ELSE
            WRITE (NOUT,99995)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,'Minimal required LWORK = ',I6)
99998 FORMAT (1X,'Actual value of  LWORK = ',I6)
99997 FORMAT (1X,'Generalized eigenvalues')
99996 FORMAT (1X,I4,5X,A,F7.3,A,F7.3,A)
99995 FORMAT (1X,'Insufficient workspace for array WORK',/' in F08XEF/',
     +        'shgeqz')
99994 FORMAT (1X,I4,'Eigenvalue is infinite')
      END
```

9.2 Program Data

```
F08XEF Example Program Data
   5                                                         :Value of N
1.00        1.00        1.00        1.00        1.00
2.00        4.00        8.00       16.00       32.00
3.00        9.00       27.00       81.00      243.00
4.00       16.00       64.00      256.00     1024.00
5.00       25.00      125.00      625.00     3125.00          :End of matrix A
1.00        2.00        3.00        4.00        5.00
1.00        4.00        9.00       16.00       25.00
1.00        8.00       27.00       64.00      125.00
1.00       16.00       81.00      256.00      625.00
1.00       32.00      243.00     1024.00     3125.00          :End of matrix B
```

9.3 Program Results

```
 F08XEF Example Program Results
 Matrix A after balancing
             1          2          3          4          5
 1      1.0000     1.0000     0.1000     0.1000     0.1000
 2      2.0000     4.0000     0.8000     1.6000     3.2000
 3      0.3000     0.9000     0.2700     0.8100     2.4300
 4      0.4000     1.6000     0.6400     2.5600    10.2400
 5      0.5000     2.5000     1.2500     6.2500    31.2500

 Matrix B after balancing
             1          2          3          4          5
 1      1.0000     2.0000     0.3000     0.4000     0.5000
 2      1.0000     4.0000     0.9000     1.6000     2.5000
 3      0.1000     0.8000     0.2700     0.6400     1.2500
 4      0.1000     1.6000     0.8100     2.5600     6.2500
 5      0.1000     3.2000     2.4300    10.2400    31.2500

 Matrix A in Hessenberg form
             1          2          3          4          5
 1     -2.1898    -0.3181     2.0547     4.7371    -4.6249
 2     -0.8395    -0.0426     1.7132     7.5194   -17.1850
 3      0.0000    -0.2846    -1.0101    -7.5927    26.4499
 4      0.0000     0.0000     0.0376     1.4070    -3.3643
 5      0.0000     0.0000     0.0000     0.3813    -0.9937

 Matrix B is triangular
             1          2          3          4          5
 1     -1.4248    -0.3476     2.1175     5.5813    -3.9269
 2      0.0000    -0.0782     0.1189     8.0940   -15.2928
 3      0.0000     0.0000     1.0021   -10.9356    26.5971
 4      0.0000     0.0000     0.0000     0.5820    -0.0730
 5      0.0000     0.0000     0.0000     0.0000     0.5321

 Minimal required LWORK =      5
 Actual value of  LWORK =     60

 Generalized eigenvalues
    1     ( -2.437,  0.000)
    2     (  0.607,  0.795)
    3     (  0.607, -0.795)
    4     (  1.000,  0.000)
    5     ( -0.410,  0.000)
```