NAG Fortran Library Routine Document

F08XSF (CHGEQZ/ZHGEQZ)

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

F08XSF (CHGEQZ/ZHGEQZ) implements the QZ method for finding generalized eigenvalues of the complex matrix pair (A, B) of order n, which is in the generalized upper Hessenberg form.

2 Specification

```
 SUBROUTINE F08XSF(JOB, COMPQ, COMPZ, N, ILO, IHI, A, LDA, B, LDB, ALPHA,
1                  BETA, Q, LDQ, Z, LDZ, WORK, LWORK, RWORK, INFO)
 ENTRY      chgeqz (JOB, COMPQ, COMPZ, N, ILO, IHI, A, LDA, B, LDB, ALPHA,
1                  BETA, Q, LDQ, Z, LDZ, WORK, LWORK, RWORK, INFO)
 INTEGER           N, ILO, IHI, LDA, LDB, LDQ, LDZ, LWORK, INFO
 real              RWORK(*)
 complex           A(LDA,*), B(LDB,*), ALPHA(*), BETA(*), Q(LDQ,*),
1                  Z(LDZ,*), WORK(*)
 CHARACTER*1       JOB, COMPQ, COMPZ
```

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

3 Description

F08XSF (CHGEQZ/ZHGEQZ) implements a single-shift version of the QZ method for finding the generalized eigenvalues of the complex 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 F08WSF (CGGHRD/ZGGHRD) should be called before invoking F08XSF (CHGEQZ/ZHGEQZ).

This problem is mathematically equivalent to solving the matrix 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 matrices A and B are upper triangular and the diagonal values of A and B provide α and β.

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

$$\begin{aligned} A &\leftarrow Q^H A Z \\ B &\leftarrow Q^H B Z \end{aligned}$$

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 unitary 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' rather than '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 in 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: N ≥ 0.

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

On entry: the indices i_{lo} and i_{hi}, respectively which defines the upper triangular parts of A. The submatrices A$(1 : i_{lo} - 1, 1 : i_{lo} - 1)$ and A$(i_{hi} + 1 : n, i_{hi} + 1 : n)$ are then upper triangular. These parameters are provided by F08WVF (CGGBAL/ZGGBAL) if the matrix pair was previously balanced; otherwise, ILO $= 1$ and IHI $=$ N.

Constraints:

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

7: A(LDA,*) – ***complex*** 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 F08XSF (CHGEQZ/ZHGEQZ) is called.

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

9: B(LDB,*) – ***complex*** 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 F08XSF (CHGEQZ/ZHGEQZ) is called.

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

11: ALPHA($*$) – ***complex*** array *Output*

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

On exit: α_j, for $j = 1, \ldots, n$.

12: BETA($*$) – ***complex*** array *Output*

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

On exit: β_j, for $j = 1, \ldots, n$.

13: Q(LDQ,*) – ***complex*** array *Input/Output*

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

On entry: if COMPQ = 'V', the matrix Q_0 is usually the matrix Q returned by F08NSF (CGEHRD/ZGEHRD).

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.

14: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08XSF (CHGEQZ/ZHGEQZ) is called.

Constraints:

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

15: Z(LDZ,*) – ***complex*** array *Input/Output*

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

On entry: if COMPZ = 'V', the matrix Z_0. Usually, Z_0 is the matrix Z returned by F08WSF (CGGHRD/ZGGHRD). 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.

16: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08XSF (CHGEQZ/ZHGEQZ) is called.

Constraints:

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

17: WORK($*$) – ***complex*** array *Workspace*

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

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

18: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the subprogram from which F08XSF (CHGEQZ/ZHGEQZ) 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 $\geq \max(1, \mathrm{N})$ or LWORK $= -1$.

19: RWORK($*$) – ***real*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 6 * \mathrm{N})$.

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 complex generalized eigenvalue problem and is called after F08WSF (CGGHRD/ZGGHRD).

The number of floating-point operations taken by this routine is proportional to n^3.

The real analogue of this routine is F08XEF (SHGEQZ/DHGEQZ).

9 Example

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

$$A = \begin{pmatrix} 1.0 + 3.0i & 1.0 + 4.0i & 1.0 + 5.0i & 1.0 + 6.0i \\ 2.0 + 2.0i & 4.0 + 3.0i & 8.0 + 4.0i & 16.0 + 5.0i \\ 3.0 + 1.0i & 9.0 + 2.0i & 27.0 + 3.0i & 81.0 + 4.0i \\ 4.0 + 0.0i & 16.0 + 1.0i & 64.0 + 2.0i & 256.0 + 3.0i \end{pmatrix}$$

$$B = \begin{pmatrix} 1.0 + 0.0i & 2.0 + 1.0i & 3.0 + 2.0i & 4.0 + 3.0i \\ 1.0 + 1.0i & 4.0 + 2.0i & 9.0 + 3.0i & 16.0 + 4.0i \\ 1.0 + 2.0i & 8.0 + 3.0i & 27.0 + 4.0i & 64.0 + 5.0i \\ 1.0 + 3.0i & 16.0 + 4.0i & 81.0 + 5.0i & 256.0 + 6.0i \end{pmatrix}.$$

This requires calls to five routines: F08WVF (CGGBAL/ZGGBAL) to balance the matrix, F08ASF (CGEQRF/ZGEQRF) to perform the QR factorization of B, F08AUF (CUNMQR/ZUNMQR) to apply Q to A, F08WSF (CGGHRD/ZGGHRD) to reduce the matrix pair to the generalized Hessenberg form and F08XSF (CHGEQZ/ZHGEQZ) to compute the eigenvalues via 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.

```
*     F08XSF 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 ..
      complex          E
      INTEGER          I, IFAIL, IHI, ILO, INFO, IROWS, J, JWORK, N
      CHARACTER        COMPQ, COMPZ, JOB
```

```
*     .. Local Arrays ..
      complex          A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(NMAX), Q(LDQ,LDQ), TAU(NMAX), WORK(LWORK),
     +                 Z(LDZ,LDZ)
      real             LSCALE(NMAX), RSCALE(NMAX), RWORK(6*NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         X04DBF, cgeqrf, cggbal, cgghrd, chgeqz, cunmqr
*     .. Intrinsic Functions ..
      INTRINSIC        real, imag, NINT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08XSF 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 cggbal(JOB,N,A,LDA,B,LDB,ILO,IHI,LSCALE,RSCALE,RWORK,INFO)
*
*        Matrix A after balancing
*
         IFAIL = 0
         CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed','F7.4',
     +               'Matrix A after balancing','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
*
*        Matrix B after balancing
*
         IFAIL = 0
         CALL X04DBF('General',' ',N,N,B,LDB,'Bracketed','F7.4',
     +               'Matrix B after balancing','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
*
*        Reduce B to triangular form using QR
*
         IROWS = IHI + 1 - ILO
         CALL cgeqrf(IROWS,IROWS,B(ILO,ILO),LDB,TAU,WORK,LWORK,INFO)
*
*        Apply the orthogonal transformation to A
*
         CALL cunmqr('L','C',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 cgghrd(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 X04DBF('General',' ',N,N,A,LDA,'Bracketed','F7.3',
     +               'Matrix A in Hessenberg form','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
```

```
         WRITE (NOUT,*)
*
*        Matrix B in generalized Hessenberg form
*
         IFAIL = 0
         CALL X04DBF('General',' ',N,N,B,LDB,'Bracketed','F7.3',
     +               'Matrix B is triangular','Integer',RLABS,'Integer',
     +               CLABS,80,0,IFAIL)
*
*        Routine chgeqz
*        Workspace query: JWORK = -1
*
         JWORK = -1
         JOB = 'E'
         CALL chgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHA,BETA,Q,
     +               LDQ,Z,LDZ,WORK,JWORK,RWORK,INFO)
         WRITE (NOUT,*)
         WRITE (NOUT,99999) NINT(real(WORK(1)))
         WRITE (NOUT,99998) LWORK
         WRITE (NOUT,*)
         WRITE (NOUT,99997)
         WRITE (NOUT,99996)
*
*        Compute the generalized Schur form
*        if the workspace LWORK is adequate
*
         IF (NINT(real(WORK(1))).LE.LWORK) THEN
            CALL chgeqz(JOB,COMPQ,COMPZ,N,ILO,IHI,A,LDA,B,LDB,ALPHA,
     +                  BETA,Q,LDQ,Z,LDZ,WORK,LWORK,RWORK,INFO)
*
*        Print the generalized eigenvalues
*        Note: the actual values of beta are real and non-negative
*
            DO 20 I = 1, N
               IF (real(BETA(I)).NE.0.0e0) THEN
                  E = ALPHA(I)/BETA(I)
                  WRITE (NOUT,99995) I, '(', real(E), ',', imag(E), ')'
               ELSE
                  WRITE (NOUT,99996) I
               END IF
   20       CONTINUE
         ELSE
            WRITE (NOUT,99994)
         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,'Infinite eigenvalue')
99995 FORMAT (1X,I4,5X,A,F7.3,A,F7.3,A)
99994 FORMAT (1X,'Insufficient workspace for array WORK',/' in F08XSF/',
     +       'chgeqz')
      END
```

9.2 Program Data

```
F08XSF Example Program Data
   4                                                          :Value of N
(  1.00, 3.00)  (  1.00, 4.00)  (  1.00, 5.00)  (  1.00, 6.00)
(  2.00, 2.00)  (  4.00, 3.00)  (  8.00, 4.00)  ( 16.00, 5.00)
(  3.00, 1.00)  (  9.00, 2.00)  ( 27.00, 3.00)  ( 81.00, 4.00)
(  4.00, 0.00)  ( 16.00, 1.00)  ( 64.00, 2.00)  (256.00, 3.00)  :End of matrix A
(  1.00, 0.00)  (  2.00, 1.00)  (  3.00, 2.00)  (  4.00, 3.00)
(  1.00, 1.00)  (  4.00, 2.00)  (  9.00, 3.00)  ( 16.00, 4.00)
(  1.00, 2.00)  (  8.00, 3.00)  ( 27.00, 4.00)  ( 64.00, 5.00)
(  1.00, 3.00)  ( 16.00, 4.00)  ( 81.00, 5.00)  (256.00, 6.00)  :End of matrix B
```

9.3 Program Results

```
F08XSF Example Program Results
Matrix A after balancing
                  1                  2                  3                  4
1  ( 1.0000, 3.0000) ( 1.0000, 4.0000) ( 0.1000, 0.5000) ( 0.1000, 0.6000)
2  ( 2.0000, 2.0000) ( 4.0000, 3.0000) ( 0.8000, 0.4000) ( 1.6000, 0.5000)
3  ( 0.3000, 0.1000) ( 0.9000, 0.2000) ( 0.2700, 0.0300) ( 0.8100, 0.0400)
4  ( 0.4000, 0.0000) ( 1.6000, 0.1000) ( 0.6400, 0.0200) ( 2.5600, 0.0300)

Matrix B after balancing
                  1                  2                  3                  4
1  ( 1.0000, 0.0000) ( 2.0000, 1.0000) ( 0.3000, 0.2000) ( 0.4000, 0.3000)
2  ( 1.0000, 1.0000) ( 4.0000, 2.0000) ( 0.9000, 0.3000) ( 1.6000, 0.4000)
3  ( 0.1000, 0.2000) ( 0.8000, 0.3000) ( 0.2700, 0.0400) ( 0.6400, 0.0500)
4  ( 0.1000, 0.3000) ( 1.6000, 0.4000) ( 0.8100, 0.0500) ( 2.5600, 0.0600)

Matrix A in Hessenberg form
                  1                  2                  3                  4
1  ( -2.868, -1.595) ( -0.809, -0.328) ( -4.900, -0.987) ( -0.048,  1.163)
2  ( -2.672,  0.595) ( -0.790,  0.049) ( -4.955, -0.163) ( -0.439, -0.574)
3  (  0.000,  0.000) ( -0.098, -0.011) ( -1.168, -0.137) ( -1.756, -0.205)
4  (  0.000,  0.000) (  0.000,  0.000) (  0.087,  0.004) (  0.032,  0.001)

Matrix B is triangular
                  1                  2                  3                  4
1  ( -1.775,  0.000) ( -0.721,  0.043) ( -5.021,  1.190) ( -0.145,  0.726)
2  (  0.000,  0.000) ( -0.218,  0.035) ( -2.541, -0.146) ( -0.823, -0.418)
3  (  0.000,  0.000) (  0.000,  0.000) ( -1.396, -0.163) ( -1.747, -0.204)
4  (  0.000,  0.000) (  0.000,  0.000) (  0.000,  0.000) ( -0.100, -0.004)

Minimal required LWORK =      4
Actual value of  LWORK =     60

Generalized eigenvalues

   1     ( -0.635,  1.653)
   2     (  0.493,  0.910)
   3     (  0.674, -0.050)
   4     (  0.458, -0.843)
```