NAG Fortran Library Routine Document

F08XNF (ZGGES)

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

F08XNF (ZGGES) computes the generalized eigenvalues, the generalized Schur form (S,T), and, optionally the left and/or right generalized Schur vectors for a pair of n by n complex nonsymmetric matrices (A,B).

2 Specification

```
 SUBROUTINE F08XNF (JOBVSL, JOBVSR, SORT, SELCTG, N, A, LDA, B, LDB,
1                   SDIM, ALPHA, BETA, VSL, LDVSL, VSR, LDVSR, WORK,
2                   LWORK, RWORK, BWORK, INFO)
 INTEGER            N, LDA, LDB, SDIM, LDVSL, LDVSR, LWORK, INFO
 double precision   RWORK(*)
 complex*16         A(LDA,*), B(LDB,*), ALPHA(*), BETA(*), VSL(LDVSL,*),
1                   VSR(LDVSR,*), WORK(*)
 LOGICAL            SELCTG, BWORK(*)
 CHARACTER*1        JOBVSL, JOBVSR, SORT
 EXTERNAL           SELCTG
```

The routine may be called by its LAPACK name ***zgges***.

3 Description

The generalized Schur factorization for a pair of complex matrices (A,B) is given by

$$A = QSZ^H, \quad B = QTZ^H,$$

where Q and Z are unitary, T and S are upper triangular. The generalized eigenvalues, λ, of (A,B) are computed from the diagonals of T and S and satisfy

$$Az = \lambda Bz,$$

where z is the corresponding generalized eigenvector. λ is actually returned as the pair (α,β) such that

$$\lambda = \alpha/\beta$$

since β, or even both α and β can be zero. The columns of Q and Z are the left and right generalized Schur vectors of (A,B).

Optionally, F08XNF (ZGGES) can order the generalized eigenvalues on the diagonals of (S,T) so that selected eigenvalues are at the top left. The leading columns of Q and Z then form an orthonormal basis for the corresponding eigenspaces, the deflating subspaces.

F08XNF (ZGGES) computes T to have real non-negative diagonal entries. The generalized Schur factorization, before reordering, is computed by the QZ algorithm.

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 URL: http://www.netlib.org/lapack/lug

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

5 Parameters

1: JOBVSL – CHARACTER*1 *Input*

On entry: if JOBVSL = 'N', do not compute the left Schur vectors.

If JOBVSL = 'V', compute the left Schur vectors.

2: JOBVSR – CHARACTER*1 *Input*

On entry: if JOBVSR = 'N', do not compute the right Schur vectors.

If JOBVSR = 'V', compute the right Schur vectors.

3: SORT – CHARACTER*1 *Input*

On entry: specifies whether or not to order the eigenvalues on the diagonal of the generalized Schur form:

if SORT = 'N', eigenvalues are not ordered;
if SORT = 'S', eigenvalues are ordered (see SELCTG).

4: SELCTG – LOGICAL FUNCTION, supplied by the user. *External Procedure*

If SORT = 'S', SELCTG is used to select generalized eigenvalues to the top left of the generalized Schur form.

If SORT = 'N', SELCTG is not referenced and F08XNF (ZGGES) may be called with the dummy function F08XNZ.

Its specification is:

```
LOGICAL FUNCTION SELCTG (A, B)
complex*16              A, B
```

1: A – ***complex*16*** *Input*
2: B – ***complex*16*** *Input*

On entry: an eigenvalue $\mathrm{A}(j)/\mathrm{B}(j)$ is selected if $\mathrm{SELCTG}(\mathrm{A}(j),\mathrm{B}(j))$ is true.

Note that in the ill-conditioned case, a selected generalized eigenvalue may no longer satisfy $\mathrm{SELCTG}(\mathrm{A}(j),\mathrm{B}(j))$ = .TRUE. after ordering. INFO is set to $\mathrm{N}+2$ in this case. (See INFO below).

SELCTG must be declared as EXTERNAL in the (sub)program from which F08XNF (ZGGES) is called. Parameters denoted as *Input* must **not** be changed by this procedure.

5: N – INTEGER *Input*

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

6: A(LDA,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the first of the pair of matrices, A.

On exit: has been overwritten by its generalized Schur form S.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08XNF (ZGGES) is called.

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

8: B(LDB,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the second of the pair of matrices, B.

On exit: has been overwritten by its generalized Schur form T.

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08XNF (ZGGES) is called.

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

10: SDIM – INTEGER *Output*

On exit: if SORT $=$ 'N', SDIM $= 0$.

If SORT $=$ 'S', SDIM $=$ number of eigenvalues (after sorting) for which SELCTG is true.

11: ALPHA($*$) – ***complex*16*** array *Output*

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

On exit: see the description of BETA below.

12: BETA($*$) – ***complex*16*** array *Output*

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

On exit: $\mathrm{ALPHA}(j)/\mathrm{BETA}(j)$, $j = 1, \ldots, \mathrm{N}$, will be the generalized eigenvalues. $\mathrm{ALPHA}(j)$, $j = 1, \ldots, \mathrm{N}$ and $\mathrm{BETA}(j)$, $j = 1, \ldots, \mathrm{N}$ are the diagonals of the complex Schur form (A, B) output by ZGGES. The $\mathrm{BETA}(j)$ will be non-negative real.

Note: the quotients $\mathrm{ALPHA}(j)/\mathrm{BETA}(j)$ may easily over- or underflow, and $\mathrm{BETA}(j)$ may even be zero. Thus, the user should avoid naively computing the ratio α/β. However, ALPHA will be always less than and usually comparable with $\|\mathrm{A}\|$ in magnitude, and BETA always less than and usually comparable with $\|\mathrm{B}\|$.

13: VSL(LDVSL,$*$) – ***complex*16*** array *Output*

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

On exit: if JOBVSL $=$ 'V', VSL will contain the left Schur vectors, Q.

If JOBVSL $=$ 'N', VSL is not referenced.

14: LDVSL – INTEGER *Input*

On entry: the first dimension of the array VSL as declared in the (sub)program from which F08XNF (ZGGES) is called.

Constraints:

if JOBVSL $=$ 'V', $\mathrm{LDVSL} \geq \max(1, \mathrm{N})$;
$\mathrm{LDVSL} \geq 1$ otherwise.

15: VSR(LDVSR,$*$) – ***complex*16*** array *Output*

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

On exit: if JOBVSR $=$ 'V', VSR will contain the right Schur vectors, Z.

If JOBVSR $=$ 'N', VSR is not referenced.

16: LDVSR – INTEGER *Input*

On entry: the first dimension of the array VSR as declared in the (sub)program from which F08XNF (ZGGES) is called.

Constraints:

if JOBVSR = 'V', LDVSR $\geq \max(1, \mathrm{N})$;
LDVSR ≥ 1 otherwise.

17: WORK($*$) – ***complex*16*** array *Workspace*

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

On exit: if INFO $= 0$, WORK(1) returns the optimal LWORK.

18: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08XNF (ZGGES) is called.

For good performance, LWORK must generally be larger than the minimum, say $2 \times \mathrm{N} + nb \times \mathrm{N}$, where nb is the optimal block size for F08NSF (ZGEHRD).

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraint: LWORK $\geq \max(1, 2 \times \mathrm{N})$.

19: RWORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 8 \times \mathrm{N})$.

20: BWORK($*$) – LOGICAL array *Workspace*

Note: the dimension of the array BWORK must be at least 1 if SORT = 'N' and at least $\max(1, \mathrm{N})$ otherwise.

If SORT = 'N', BWORK is not referenced.

21: 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 argument had an illegal value.

INFO $= 1$ to N

The QZ iteration failed. (A, B) are not in Schur form, but ALPHA(j) and BETA(j) should be correct for $j = \mathrm{INFO} + 1, \ldots, \mathrm{N}$.

INFO $>$ N

$= \mathrm{N} + 1$: other than QZ iteration failed in F08XSF (ZHGEQZ)

$= \mathrm{N} + 2$: after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the generalized Schur form no longer satisfy SELZTG = .TRUE.. This could also be caused due to scaling.

$= \mathrm{N} + 3$: reordering failed because some eigenvalues were too close to separate (the problem is very ill-conditioned).

7 Accuracy

The computed generalized Schur factorization satisfies

$$A + E = QSZ^H, \quad B + F = QTZ^H,$$

where

$$\|(E,F)\|_F = \mathrm{O}(\epsilon)\|(A,B)\|_F$$

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating-point operations is proportional to n^3.

The real analogue of this routine is F08XAF (DGGES).

9 Example

To find the generalized Schur factorization of the matrix pair (A, B), where

$$A = \begin{pmatrix} -21.10 - 22.50i & 53.50 - 50.50i & -34.50 + 127.50i & 7.50 + 0.50i \\ -0.46 - 7.78i & -3.50 - 37.50i & -15.50 + 58.50i & -10.50 - 1.50i \\ 4.30 - 5.50i & 39.70 - 17.10i & -68.50 + 12.50i & -7.50 - 3.50i \\ 5.50 + 4.40i & 14.40 + 43.30i & -32.50 - 46.00i & -19.00 - 32.50i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.00 - 5.00i & 1.60 + 1.20i & -3.00 + 0.00i & 0.00 - 1.00i \\ 0.80 - 0.60i & 3.00 - 5.00i & -4.00 + 3.00i & -2.40 - 3.20i \\ 1.00 + 0.00i & 2.40 + 1.80i & -4.00 - 5.00i & 0.00 - 3.00i \\ 0.00 + 1.00i & -1.80 + 2.40i & 0.00 - 4.00i & 4.00 - 5.00i \end{pmatrix}.$$

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

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.

```
*     F08XNF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX
      PARAMETER        (NB=64,NMAX=10)
      INTEGER          LDA, LDB, LDVSL, LDVSR, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LDVSL=NMAX,LDVSR=NMAX,
     +                 LWORK=NMAX+NMAX*NB)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, LWKOPT, N, SDIM
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(NMAX), VSL(LDVSL,NMAX), VSR(LDVSR,NMAX),
     +                 WORK(LWORK)
      DOUBLE PRECISION RWORK(8*NMAX)
      LOGICAL          BWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Functions ..
      LOGICAL          F08XNZ
      EXTERNAL         F08XNZ
*     .. External Subroutines ..
      EXTERNAL         X04DBF, ZGGES
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08XNF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read in the matrices A and B
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((B(I,J),J=1,N),I=1,N)
*
*        Find the generalized Schur form
*
         CALL ZGGES('Vectors (left)','Vectors (right)','No sort',F08XNZ,
     +              N,A,LDA,B,LDB,SDIM,ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,
     +              WORK,LWORK,RWORK,BWORK,INFO)
*
         IF (INFO.GT.0) THEN
            WRITE (NOUT,99999) 'Failure in ZGGES. INFO =', INFO
         ELSE
*
*           Print out the factors of the generalized Schur factorization
*
            IFAIL = 0
            CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed','F7.3',
     +                  'Generalized Schur matrix S','Integer',RLABS,
     +                  'Integer',CLABS,80,0,IFAIL)
*
            WRITE (NOUT,*)
            CALL X04DBF('General',' ',N,N,B,LDB,'Bracketed','F7.3',
     +                  'Generalized Schur matrix T','Integer',RLABS,
     +                  'Integer',CLABS,80,0,IFAIL)
*
            WRITE (NOUT,*)
            CALL X04DBF('General',' ',N,N,VSL,LDVSL,'Bracketed','F7.4',
     +                  'Matrix of left generalized Schur vectors',
     +                  'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
*
            WRITE (NOUT,*)
            CALL X04DBF('General',' ',N,N,VSR,LDVSR,'Bracketed','F7.4',
     +                  'Matrix of right generalized Schur vectors',
     +                  'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
*
            LWKOPT = WORK(1)
            IF (LWORK.LT.LWKOPT) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,99998) 'Optimum workspace required = ',
     +           LWKOPT, 'Workspace provided         = ', LWORK
            END IF
         END IF
      ELSE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
99999 FORMAT (1X,A,I4)
99998 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08XNF Example Program Data
  4                                                               : Value of N
  (-21.10,-22.50) ( 53.50,-50.50) (-34.50,127.50) (  7.50,  0.50)
  ( -0.46, -7.78) ( -3.50,-37.50) (-15.50, 58.50) (-10.50, -1.50)
  (  4.30, -5.50) ( 39.70,-17.10) (-68.50, 12.50) ( -7.50, -3.50)
  (  5.50,  4.40) ( 14.40, 43.30) (-32.50,-46.00) (-19.00,-32.50) : End of A
  (  1.00, -5.00) (  1.60,  1.20) ( -3.00,  0.00) (  0.00, -1.00)
  (  0.80, -0.60) (  3.00, -5.00) ( -4.00,  3.00) ( -2.40, -3.20)
  (  1.00,  0.00) (  2.40,  1.80) ( -4.00, -5.00) (  0.00, -3.00)
  (  0.00,  1.00) ( -1.80,  2.40) (  0.00, -4.00) (  4.00, -5.00) : End of B
```

9.3 Program Results

```
 F08XNF Example Program Results

 Generalized Schur matrix S
                   1                  2                  3                  4
 1  ( 19.033,-57.099) ( 53.591,-89.817) (-81.314,-63.230) (106.662,-44.786)
 2  (  0.000,  0.000) ( 11.882,-29.705) (  3.562, 27.629) ( -0.671,-16.421)
 3  (  0.000,  0.000) (  0.000,  0.000) ( 10.961, -3.654) (-25.023, -8.201)
 4  (  0.000,  0.000) (  0.000,  0.000) (  0.000,  0.000) ( 21.872,-27.340)

 Generalized Schur matrix T
                   1                  2                  3                  4
 1  (  6.344,  0.000) (  3.399,  0.712) ( -0.515, -2.382) (  6.582,  2.430)
 2  (  0.000,  0.000) (  5.941,  0.000) ( -2.448, -0.343) (  5.739, -0.702)
 3  (  0.000,  0.000) (  0.000,  0.000) (  3.654,  0.000) ( -1.410, -3.933)
 4  (  0.000,  0.000) (  0.000,  0.000) (  0.000,  0.000) (  5.468,  0.000)

 Matrix of left generalized Schur vectors
                   1                  2                  3                  4
 1  (-0.3347, 0.7387) ( 0.2872,-0.4789) ( 0.1725, 0.0093) ( 0.0144,-0.0212)
 2  (-0.1277, 0.2493) (-0.0282, 0.4999) ( 0.1541,-0.8008) (-0.0087,-0.0767)
 3  (-0.3557, 0.0396) (-0.4615,-0.0822) (-0.3939, 0.0258) (-0.1464,-0.6892)
 4  (-0.0126,-0.3682) ( 0.1508,-0.4417) ( 0.1517,-0.3555) (-0.7049,-0.0133)

 Matrix of right generalized Schur vectors
                   1                  2                  3                  4
 1  (-0.9240,-0.1977) ( 0.2460, 0.2090) (-0.0054, 0.0542) ( 0.0000, 0.0000)
 2  (-0.1716, 0.0793) (-0.5943, 0.0905) ( 0.7467,-0.2127) ( 0.0000, 0.0000)
 3  (-0.0793,-0.1716) ( 0.0943,-0.5082) ( 0.0102,-0.4438) ( 0.7034,-0.0728)
 4  ( 0.1716,-0.0793) ( 0.5082, 0.0943) ( 0.4438, 0.0102) (-0.0728,-0.7034)
```