NAG 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, SDIM,    &
                   ALPHA, BETA, VSL, LDVSL, VSR, LDVSR, WORK, LWORK, RWORK,   &
                   BWORK, INFO)

INTEGER            N, LDA, LDB, SDIM, LDVSL, LDVSR, LWORK, INFO
REAL (KIND=nag_wp) RWORK(max(1,8*N))
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*), ALPHA(N), BETA(N), VSL(LDVSL,*),    &
                   VSR(LDVSR,*), WORK(max(1,LWORK))
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

$$A = QSZ^{\mathrm{H}}, \qquad B = QTZ^{\mathrm{H}},$$

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 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.

Constraint: JOBVSL = 'N' or 'V'.

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.

Constraint: JOBVSR = 'N' or 'V'.

3:    SORT – CHARACTER(1)   *Input*

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

SORT = 'N'
      Eigenvalues are not ordered.

SORT = 'S'
      Eigenvalues are ordered (see SELCTG).

Constraint: SORT = 'N' or 'S'.

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 by F08XNF (ZGGES), and may be called with the dummy function F08XNZ.

The specification of SELCTG is:

```
FUNCTION SELCTG (A, B)

LOGICAL SELCTG
COMPLEX (KIND=nag_wp) A, B
```

1:     A – COMPLEX (KIND=nag_wp)   *Input*
2:     B – COMPLEX (KIND=nag_wp)   *Input*

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

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

SELCTG must either be a module subprogram USEd by, or 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.

Constraint: $N \geq 0$.

6:      A(LDA,∗) – COMPLEX (KIND=nag_wp) array   *Input/Output*

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

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

        On exit: A 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: $LDA \geq \max(1, N)$.

8:      B(LDB,∗) – COMPLEX (KIND=nag_wp) array   *Input/Output*

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

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

        On exit: B 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: $LDB \geq \max(1, N)$.

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(N) – COMPLEX (KIND=nag_wp) array   *Output*

        On exit: see the description of BETA.

12:     BETA(N) – COMPLEX (KIND=nag_wp) array   *Output*

        On exit: ALPHA(j)/BETA(j), for $j = 1, 2, \ldots, N$, will be the generalized eigenvalues. ALPHA(j), for $j = 1, 2, \ldots, N$ and BETA(j), for $j = 1, 2, \ldots, N$, are the diagonals of the complex Schur form (A, B) output by F08XNF (ZGGES). The BETA(j) will be non-negative real.

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

13:     VSL(LDVSL,∗) – COMPLEX (KIND=nag_wp) array   *Output*

        Note: the second dimension of the array VSL must be at least $\max(1, N)$ if JOBVSL = 'V', and at least 1 otherwise.

        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', $LDVSL \geq \max(1, N)$;
                otherwise $LDVSL \geq 1$.

15: VSR(LDVSR,∗) – COMPLEX (KIND=nag_wp) array *Output*

   Note: the second dimension of the array VSR must be at least $\max(1, N)$ if JOBVSR = 'V', and at least 1 otherwise.

   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 $\geq \max(1, N)$;
   > otherwise LDVSR ≥ 1.

17: WORK($\max(1, \text{LWORK})$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

18: LWORK – INTEGER *Input*

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

   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.

   Suggested value: for optimal performance, LWORK must generally be larger than the minimum, say $2 \times N + nb \times N$, where nb is the optimal **block size** for F08NSF (ZGEHRD).

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

19: RWORK($\max(1, 8 \times N)$) – REAL (KIND=nag_wp) array *Workspace*

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

   Note: the dimension of the array BWORK must be at least 1 if SORT = 'N', and at least $\max(1, N)$ 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 = $-i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

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 $j = \text{INFO} + 1, \ldots, N$.

INFO = N + 1

Unexpected error returned from F08XSF (ZHGEQZ).

INFO = N + 2

After reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the generalized Schur form no longer satisfy SELCTG = .TRUE.. This could also be caused by underflow due to scaling.

INFO = N + 3

The eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned).

7   Accuracy

The computed generalized Schur factorization satisfies

$$A + E = QSZ^{H}, \qquad B + F = QTZ^{H},$$

where

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

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

This example finds 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

```
    Program f08xnfe

!       F08XNF Example Program Text

!       Mark 24 Release. NAG Copyright 2012.

!       .. Use Statements ..
        Use nag_library, Only: f08xnz, nag_wp, x02amf, zgges
!       .. Implicit None Statement ..
        Implicit None
!       .. Parameters ..
```

```
      Integer, Parameter                :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)             :: eig
      Real (Kind=nag_wp)                :: small
      Integer                           :: i, info, j, lda, ldb, ldc, ldvsl,    &
                                           ldvsr, lwork, n, sdim
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), alpha(:), b(:,:), beta(:), &
                                            c(:,:), vsl(:,:), vsr(:,:), work(:)
      Complex (Kind=nag_wp)             :: dummy(1)
      Real (Kind=nag_wp), Allocatable   :: rwork(:)
      Logical, Allocatable              :: bwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: abs, max, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08XNF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldc = n
      ldvsl = n
      ldvsr = n
      Allocate (a(lda,n),alpha(n),b(ldb,n),beta(n),c(ldc,n),vsl(ldvsl,n), &
        vsr(ldvsr,n),rwork(8*n),bwork(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of zgges is f08xnf
      Call zgges('Vectors (left)','Vectors (right)','No sort',f08xnz,n,a,lda, &
        b,ldb,sdim,alpha,beta,vsl,ldvsl,vsr,ldvsr,dummy,lwork,rwork,bwork, &
        info)

!     Make sure that there is enough workspace for blocksize nb.
      lwork = max((nb+1)*n,nint(real(dummy(1))))
      Allocate (work(lwork))

!     Read in the matrices A and B

      Read (nin,*)(a(i,1:n),i=1,n)
      Read (nin,*)(b(i,1:n),i=1,n)

!     Find the generalized Schur form
!     The NAG name equivalent of zgges is f08xnf
      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>0) Then
        Write (nout,99999) 'Failure in ZGGES. INFO =', info
      Else

        small = x02amf()
        Write (nout,*) 'Generalized Eigenvalues'
        Write (nout,*)
        Do j = 1, n

!         Print out information on the jth eigenvalue

          If ((abs(a(j,j)))*small>=abs(b(j,j))) Then
            Write (nout,99997) j
          Else
            eig = a(j,j)/b(j,j)
            Write (nout,99998) j, eig
          End If

        End Do

      End If
```

```
99999 Format (1X,A,I4)
99998 Format (1X,I2,1X,'(',1P,E11.4,',',E11.4,')')
99997 Format (1X,I2,1X,'  Infinite or undetermined')
   End Program f08xnfe
```

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 Eigenvalues

  1 (  3.0000E+00,-9.0000E+00)
  2 (  2.0000E+00,-5.0000E+00)
  3 (  3.0000E+00,-1.0000E+00)
  4 (  4.0000E+00,-5.0000E+00)
```

_____