NAG Library Routine Document

F08XPF (ZGGESX)

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

F08XPF (ZGGESX) 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).

Estimates of condition numbers for selected generalized eigenvalue clusters and Schur vectors are also computed.

2 Specification

```
SUBROUTINE F08XPF (JOBVSL, JOBVSR, SORT, SELCTG, SENSE, N, A, LDA, B, LDB,       &
                   SDIM, ALPHA, BETA, VSL, LDVSL, VSR, LDVSR, RCONDE,            &
                   RCONDV, WORK, LWORK, RWORK, IWORK, LIWORK, BWORK, INFO)

INTEGER                 N, LDA, LDB, SDIM, LDVSL, LDVSR, LWORK,                  &
                        IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp)      RCONDE(2), RCONDV(2), 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, SENSE
EXTERNAL                SELCTG
```

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

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, F08XPF (ZGGESX) 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.

F08XPF (ZGGESX) computes T to have real non-negative diagonal entries. The generalized Schur factorization, before reordering, is computed by the QZ algorithm.

The reciprocals of the condition estimates, the reciprocal values of the left and right projection norms, are returned in RCONDE(1) and RCONDE(2) respectively, for the selected generalized eigenvalues, together with reciprocal condition estimates for the corresponding left and right deflating subspaces, in RCONDV(1) and RCONDV(2). See Section 4.11 of Anderson *et al.* (1999) for further information.

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 F08XPF (ZGGESX), 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 $SELCTG(A(j), B(j))$ is .TRUE..

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

SELCTG must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which F08XPF (ZGGESX) is called. Parameters denoted as *Input* must **not** be changed by this procedure.

5: SENSE – CHARACTER(1) *Input*

On entry: determines which reciprocal condition numbers are computed.

SENSE = 'N'
None are computed.

SENSE = 'E'
Computed for average of selected eigenvalues only.

SENSE = 'V'
Computed for selected deflating subspaces only.

SENSE = 'B'
Computed for both.

If SENSE = 'E', 'V' or 'B', SORT = 'S'.

Constraint: SENSE = 'N', 'E', 'V' or 'B'.

6: N – INTEGER *Input*

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

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

7: A(LDA,$*$) – COMPLEX (KIND=nag_wp) 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: A has been overwritten by its generalized Schur form S.

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08XPF (ZGGESX) is called.

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

9: B(LDB,$*$) – COMPLEX (KIND=nag_wp) 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: B has been overwritten by its generalized Schur form T.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08XPF (ZGGESX) is called.

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

11: SDIM – INTEGER *Output*

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

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

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

On exit: see the description of BETA.

13: 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) and BETA$(j), j = 1, 2, \ldots,$ N are the diagonals of the complex Schur form (S, T). 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||.

14: VSL(LDVSL,$*$) – COMPLEX (KIND=nag_wp) array *Output*

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

15: LDVSL – INTEGER *Input*

On entry: the first dimension of the array VSL as declared in the (sub)program from which F08XPF (ZGGESX) is called.

Constraints:

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

16: VSR(LDVSR,$*$) – COMPLEX (KIND=nag_wp) array *Output*

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

17: LDVSR – INTEGER *Input*

On entry: the first dimension of the array VSR as declared in the (sub)program from which F08XPF (ZGGESX) is called.

Constraints:

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

18: RCONDE(2) – REAL (KIND=nag_wp) array *Output*

On exit: if SENSE = 'E' or 'B', RCONDE(1) and RCONDE(2) contain the reciprocal condition numbers for the average of the selected eigenvalues.

If SENSE = 'N' or 'V', RCONDE is not referenced.

19: RCONDV(2) – REAL (KIND=nag_wp) array *Output*

On exit: if SENSE = 'V' or 'B', RCONDV(1) and RCONDV(2) contain the reciprocal condition numbers for the selected deflating subspaces.

if SENSE = 'N' or 'E', RCONDV is not referenced.

20: WORK(max$(1,$ LWORK$)$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

21: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08XPF (ZGGESX) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the bound on the optimal size of the WORK array and the minimum size of the IWORK array, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if $\mathrm{N} = 0$, $\mathrm{LWORK} \geq 1$;
if SENSE = 'E', 'V' or 'B', $\mathrm{LWORK} \geq \max(1, 2 \times \mathrm{N}, 2 \times \mathrm{SDIM} \times (\mathrm{N} - \mathrm{SDIM}))$;
otherwise $\mathrm{LWORK} \geq \max(1, 2 \times \mathrm{N})$.

Note: $2 \times \mathrm{SDIM} \times (\mathrm{N} - \mathrm{SDIM}) \leq \mathrm{N} \times \mathrm{N}/2$. Note also that an error is only returned if $\mathrm{LWORK} < \max(1, 2 \times \mathrm{N})$, but if SENSE = 'E', 'V' or 'B' this may not be large enough. Consider increasing LWORK by *nb*, where *nb* is the optimal ***block size***.

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

Real workspace.

23: IWORK($\max(1, \mathrm{LIWORK})$) – INTEGER array *Workspace*

On exit: if INFO $= 0$, IWORK(1) returns the minimum LIWORK.

24: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08XPF (ZGGESX) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the bound on the optimal size of the WORK array and the minimum size of the IWORK array, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraints:

if SENSE = 'N' or $\mathrm{N} = 0$, $\mathrm{LIWORK} \geq 1$;
otherwise $\mathrm{LIWORK} \geq \mathrm{N} + 2$.

25: 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.

26: 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 $j = \text{INFO} + 1, \dots, \text{N}$.

INFO $= \text{N} + 1$

Unexpected error returned from F08XSF (ZHGEQZ).

INFO $= \text{N} + 2$

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 $= \text{N} + 3$

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

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

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 F08XBF (DGGESX).

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

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

such that the eigenvalues of (A, B) for which $|\lambda| < 6$ correspond to the top left diagonal elements of the generalized Schur form, (S, T). Estimates of the condition numbers for the selected eigenvalue cluster and corresponding deflating subspaces are also returned.

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

```
!   F08XPF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module f08xpfe_mod

!     F08XPF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                    :: nb = 64, nin = 5, nout = 6
    Contains
      Function selctg(a,b)

!       Logical function selctg for use with ZGGESX (F08XPF)

!       Returns the value .TRUE. if the absolute value of the eigenvalue
!       a/b < 6.0

!       .. Function Return Value ..
        Logical                               :: selctg
!       .. Scalar Arguments ..
        Complex (Kind=nag_wp), Intent (In)    :: a, b
!       .. Local Scalars ..
        Logical                               :: d
!       .. Intrinsic Procedures ..
        Intrinsic                             :: abs
!       .. Executable Statements ..
        If (abs(a)<6.0_nag_wp*abs(b)) Then
          d = .True.
        Else
          d = .False.
        End If

        selctg = d

        Return
      End Function selctg
    End Module f08xpfe_mod
    Program f08xpfe

!     F08XPF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: f06bnf, nag_wp, x02ajf, zggesx, zlange => f06uaf
      Use f08xpfe_mod, Only: nb, nin, nout, selctg
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                    :: abnorm, anorm, bnorm, eps, tol
      Integer                               :: i, info, j, lda, ldb, ldc, ldd,  &
                                               lde, ldvsl, ldvsr, liwork,       &
                                               lwork, n, sdim
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable    :: a(:,:), alpha(:), b(:,:),        &
                                               beta(:), c(:,:), d(:,:), e(:,:), &
                                               vsl(:,:), vsr(:,:), work(:)
      Complex (Kind=nag_wp)                 :: dummy(1)
      Real (Kind=nag_wp)                    :: rconde(2), rcondv(2)
      Real (Kind=nag_wp), Allocatable       :: rwork(:)
      Integer                               :: idum(1)
      Integer, Allocatable                  :: iwork(:)
      Logical, Allocatable                  :: bwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                             :: max, nint, real
!     .. Executable Statements ..
```

```
      Write (nout,*) 'F08XPF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldc = n
      ldd = n
      lde = n
      ldvsl = n
      ldvsr = n
      Allocate (a(lda,n),alpha(n),b(ldb,n),beta(n),c(ldc,n),d(ldd,n),e(lde,n), &
        vsl(ldvsl,n),vsr(ldvsr,n),rwork(8*n),bwork(n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
!     The NAG name equivalent of zggesx is f08xpf
      Call zggesx('Vectors (left)','Vectors (right)','Sort',selctg, &
        'Both reciprocal condition numbers',n,a,lda,b,ldb,sdim,alpha,beta,vsl, &
        ldvsl,vsr,ldvsr,rconde,rcondv,dummy,lwork,rwork,idum,liwork,bwork, &
        info)

!     Make sure that there is enough workspace for blocksize nb.
      lwork = max(n*nb+n*n/2,nint(real(dummy(1))))
      liwork = max(n+2,idum(1))
      Allocate (work(lwork),iwork(liwork))

!     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 Frobenius norms of A and B
!     The NAG name equivalent of the LAPACK auxiliary zlange is f06uaf
      anorm = zlange('Frobenius',n,n,a,lda,rwork)
      bnorm = zlange('Frobenius',n,n,b,ldb,rwork)

!     Find the generalized Schur form
!     The NAG name equivalent of zggesx is f08xpf
      Call zggesx('Vectors (left)','Vectors (right)','Sort',selctg, &
        'Both reciprocal condition numbers',n,a,lda,b,ldb,sdim,alpha,beta,vsl, &
        ldvsl,vsr,ldvsr,rconde,rcondv,work,lwork,rwork,iwork,liwork,bwork, &
        info)

      If (info>0 .And. info/=(n+2)) Then
        Write (nout,99999) 'Failure in ZGGESX. INFO =', info
      Else

!       Print Results
        Write (nout,99999) 'Number of selected eigenvalues  = ', sdim
        Write (nout,*)
        If (info==(n+2)) Then
          Write (nout,99998) '***Note that rounding errors mean ', &
            'that leading eigenvalues in the generalized', &
            'Schur form no longer satisfy SELCTG = .TRUE.'
          Write (nout,*)
        End If

        Flush (nout)

!       Print information on generalized eigenvalues.
        Write (nout,*) 'Selected Generalized Eigenvalues'
        Write (nout,*)
        Write (nout,99996)(j,alpha(j)/beta(j),j=1,sdim)

!       Compute the machine precision and sqrt(anorm**2+bnorm**2)
        eps = x02ajf()
        abnorm = f06bnf(anorm,bnorm)
        tol = eps*abnorm
```

```
!       Print out the reciprocal condition numbers and error bound for
!       selected eigenvalues
        Write (nout,*)
        Write (nout,99997) &
          'Reciprocal condition numbers for the average of the', &
          'selected eigenvalues and their asymptotic error bound', &
          'rcond-left = ', rconde(1), ', rcond-right = ', rconde(2), &
          ', error = ', tol/rconde(1)

        Write (nout,*)
        Write (nout,99997) &
          'Reciprocal condition numbers for the deflating subspaces', &
          'and their approximate asymptotic error bound', 'rcond-left = ', &
          rcondv(1), ', rcond-right = ', rcondv(2), ', error = ', &
          tol/rcondv(2)

      End If

99999 Format (1X,A,I4/1X,A)
99998 Format (1X,2A/1X,A)
99997 Format (1X,A/1X,A/1X,3(A,1P,E8.1))
99996 Format (1X,I2,1X,'(',1P,E11.4,',',E11.4,')')
    End Program f08xpfe
```

9.2 Program Data

```
F08XPF 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

```
 F08XPF Example Program Results

 Number of selected eigenvalues  =    2


 Selected Generalized Eigenvalues

  1 ( 2.0000E+00,-5.0000E+00)
  2 ( 3.0000E+00,-1.0000E+00)

 Reciprocal condition numbers for the average of the
 selected eigenvalues and their asymptotic error bound
 rcond-left =  1.2E-01, rcond-right =  1.6E-01, error =  1.9E-13

 Reciprocal condition numbers for the deflating subspaces
 and their approximate asymptotic error bound
 rcond-left =  4.8E-01, rcond-right =  4.7E-01, error =  4.9E-14
```