NAG Library Routine Document

F08XCF (DGGES3)

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

F08XCF (DGGES3) computes the generalized eigenvalues, the generalized real Schur form (S, T) and, optionally, the left and/or right generalized Schur vectors for a pair of n by n real nonsymmetric matrices (A, B).

2 Specification

```
SUBROUTINE F08XCF (JOBVSL, JOBVSR, SORT, SELCTG, N, A, LDA, B, LDB,        &
                   SDIM, ALPHAR, ALPHAI, BETA, VSL, LDVSL, VSR, LDVSR,     &
                   WORK, LWORK, BWORK, INFO)

INTEGER            N, LDA, LDB, SDIM, LDVSL, LDVSR, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), ALPHAR(N), ALPHAI(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 ***dgges3***.

3 Description

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

$$A = QSZ^{\mathrm{T}}, \quad B = QTZ^{\mathrm{T}},$$

where Q and Z are orthogonal, T is upper triangular and S is upper quasi-triangular with 1 by 1 and 2 by 2 diagonal blocks. The generalized eigenvalues, λ, of (A, B) are computed from the diagonals of S and T 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, F08XCF (DGGES3) 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.

F08XCF (DGGES3) computes T to have non-negative diagonal elements, and the 2 by 2 blocks of S correspond to complex conjugate pairs of generalized eigenvalues. 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 (2012) *Matrix Computations* (4th Edition) Johns Hopkins University Press, Baltimore

5 Arguments

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 be moved to the top left of the generalized Schur form.

If SORT = 'N', SELCTG is not referenced by F08XCF (DGGES3), and may be called with the dummy function F08XAZ.

The specification of SELCTG is:

```
FUNCTION SELCTG (AR, AI, B)
LOGICAL SELCTG
REAL (KIND=nag_wp) AR, AI, B
```

1: AR – REAL (KIND=nag_wp) *Input*
2: AI – REAL (KIND=nag_wp) *Input*
3: B – REAL (KIND=nag_wp) *Input*

On entry: an eigenvalue $\left(\mathrm{AR}(j)+\sqrt{-1}\times\mathrm{AI}(j)\right)/\mathrm{B}(j)$ is selected if $\mathrm{SELCTG}(\mathrm{AR}(j),\mathrm{AI}(j),\mathrm{B}(j))=$ .TRUE.. If either one of a complex conjugate pair is selected, then both complex generalized eigenvalues are selected.

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

SELCTG must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which F08XCF (DGGES3) is called. Arguments 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: $\mathrm{N}\geq 0$.

6: A(LDA, $*$) – REAL (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 F08XCF (DGGES3) is called.

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

8: B(LDB, $*$) – REAL (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 F08XCF (DGGES3) 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.. (Complex conjugate pairs for which SELCTG is .TRUE. for either eigenvalue count as 2.)

11: ALPHAR(N) – REAL (KIND=nag_wp) array *Output*

On exit: see the description of BETA.

12: ALPHAI(N) – REAL (KIND=nag_wp) array *Output*

On exit: see the description of BETA.

13: BETA(N) – REAL (KIND=nag_wp) array *Output*

On exit: (ALPHAR(j) + ALPHAI(j) $\times$ i)/BETA(j), for $j = 1, 2, \ldots,$ N, will be the generalized eigenvalues. ALPHAR(j) + ALPHAI(j) $\times$ i, and BETA(j), for $j = 1, 2, \ldots,$ N, are the diagonals of the complex Schur form (S, T) that would result if the 2 by 2 diagonal blocks of the real Schur form of (A, B) were further reduced to triangular form using 2 by 2 complex unitary transformations.

If ALPHAI(j) is zero, then the jth eigenvalue is real; if positive, then the jth and $(j+1)$st eigenvalues are a complex conjugate pair, with ALPHAI($j + 1$) negative.

Note: the quotients ALPHAR(j)/BETA(j) and ALPHAI(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, ALPHAR and ALPHAI will always be less than and usually comparable with $\|A\|_2$ in magnitude, and BETA will always be less than and usually comparable with $\|B\|_2$.

14: VSL(LDVSL, $*$) – REAL (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.

15: LDVSL – INTEGER *Input*

On entry: the first dimension of the array VSL as declared in the (sub)program from which F08XCF (DGGES3) is called.

Constraints:

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

16: VSR(LDVSR, $*$) – REAL (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.

17: LDVSR – INTEGER *Input*

On entry: the first dimension of the array VSR as declared in the (sub)program from which F08XCF (DGGES3) is called.

Constraints:

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

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

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

19: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08XCF (DGGES3) 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 mimimum; add, say $nb \times (N \times 6)$, where nb is the optimal ***block size***.

Constraints:

if N $= 0$, LWORK ≥ 1;
otherwise LWORK $\geq \max(8 \times N, 6 \times N + 16)$.

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

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. No eigenvectors have been calculated but ALPHAR(j), ALPHAI(j) and BETA(j) should be correct from element $\langle value \rangle$.

INFO $=$ N $+ 1$

The QZ iteration failed with an unexpected error, please contact NAG.

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

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

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 Parallelism and Performance

F08XCF (DGGES3) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08XCF (DGGES3) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

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

The complex analogue of this routine is F08XQF (ZGGES3).

10 Example

This example finds the generalized Schur factorization of the matrix pair (A, B), where

$$A = \begin{pmatrix} 3.9 & 12.5 & -34.5 & -0.5 \\ 4.3 & 21.5 & -47.5 & 7.5 \\ 4.3 & 21.5 & -43.5 & 3.5 \\ 4.4 & 26.0 & -46.0 & 6.0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1.0 & 2.0 & -3.0 & 1.0 \\ 1.0 & 3.0 & -5.0 & 4.0 \\ 1.0 & 3.0 & -4.0 & 3.0 \\ 1.0 & 3.0 & -4.0 & 4.0 \end{pmatrix},$$

such that the real positive eigenvalues of (A, B) correspond to the top left diagonal elements of the generalized Schur form, (S, T).

10.1 Program Text

```
!   F08XCF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module f08xcfe_mod

!     F08XCF Example Program Module:
!            Parameters and User-defined Routines

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

!       Logical function selctg for use with DGGES3 (F08XCF)
!       Returns the value .TRUE. if the eigenvalue is real and positive

!       .. Function Return Value ..
        Logical                         :: selctg
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: ai, ar, b
!       .. Executable Statements ..
        selctg = (ar>0._nag_wp .And. ai==0._nag_wp .And. b/=0._nag_wp)
        Return
      End Function selctg
    End Module f08xcfe_mod
    Program f08xcfe

!     F08XCF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: dgemm, dgges3, dlange => f06raf, nag_wp, x02ajf,  &
                             x04caf
      Use f08xcfe_mod, Only: nb, nin, nout, selctg
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: alph, bet, normd, norme
      Integer                           :: i, ifail, info, lda, ldb, ldc, ldd,  &
                                           lde, ldvsl, ldvsr, lwork, n, sdim
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable   :: a(:,:), alphai(:), alphar(:),        &
                                           b(:,:), beta(:), c(:,:), d(:,:),     &
                                           e(:,:), vsl(:,:), vsr(:,:), work(:)
      Real (Kind=nag_wp)                :: dummy(1)
      Logical, Allocatable              :: bwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: max, nint
!     .. Executable Statements ..
```

```
      Write (nout,*) 'F08XCF 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),alphai(n),alphar(n),b(ldb,n),beta(n),vsl(ldvsl,n),    &
        vsr(ldvsr,n),bwork(n),c(ldc,n),d(ldd,n),e(lde,n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
!     The NAG name equivalent of dgges3 is f08xcf
      Call dgges3('Vectors (left)','Vectors (right)','Sort',selctg,n,a,lda,b,  &
        ldb,sdim,alphar,alphai,beta,vsl,ldvsl,vsr,ldvsr,dummy,lwork,bwork,     &
        info)

!     Make sure that there is enough workspace for block size nb.
      lwork = max(8*n+16+n*nb,nint(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)

!     Copy A and B into D and E respectively
      d(1:n,1:n) = a(1:n,1:n)
      e(1:n,1:n) = b(1:n,1:n)

!     Print matrices A and B
!     ifail: behaviour on error exit
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',n,n,a,lda,'Matrix A',ifail)
      Write (nout,*)
      Flush (nout)

      ifail = 0
      Call x04caf('General',' ',n,n,b,ldb,'Matrix B',ifail)
      Write (nout,*)
      Flush (nout)

!     Find the generalized Schur form
!     The NAG name equivalent of dgges3 is f08xcf
      Call dgges3('Vectors (left)','Vectors (right)','Sort',selctg,n,a,lda,b,  &
        ldb,sdim,alphar,alphai,beta,vsl,ldvsl,vsr,ldvsr,work,lwork,bwork,info)

      If (info==0 .Or. info==(n+2)) Then

!       Compute A - Q*S*Z^T from the factorization of (A,B) and store in
!       matrix D
!       The NAG name equivalent of dgemm is f06yaf
        alph = 1.0_nag_wp
        bet = 0.0_nag_wp
        Call dgemm('N','N',n,n,n,alph,vsl,ldvsl,a,lda,bet,c,ldc)
        alph = -1.0_nag_wp
        bet = 1.0_nag_wp
        Call dgemm('N','T',n,n,n,alph,c,ldc,vsr,ldvsr,bet,d,ldd)

!       Compute B - Q*T*Z^T from the factorization of (A,B) and store in
!       matrix E
        alph = 1.0_nag_wp
        bet = 0.0_nag_wp
        Call dgemm('N','N',n,n,n,alph,vsl,ldvsl,b,ldb,bet,c,ldc)
        alph = -1.0_nag_wp
```

```
          bet = 1.0_nag_wp
          Call dgemm('N','T',n,n,n,alph,c,ldc,vsr,ldvsr,bet,e,lde)

!         Find norms of matrices D and E and warn if either is too large
!         f06raf is the NAG name equivalent of the LAPACK auxiliary dlange
          normd = dlange('O',ldd,n,d,ldd,work)
          norme = dlange('O',lde,n,e,lde,work)
          If (normd>x02ajf()**0.8_nag_wp .Or. norme>x02ajf()**0.8_nag_wp) Then
            Write (nout,*)                                                     &
              'Norm of A-(Q*S*Z^T) or norm of B-(Q*T*Z^T) is much greater than 0.'
            Write (nout,*) 'Schur factorization has failed.'
          Else

!           Print solution
            Write (nout,99999)                                                 &
              'Number of eigenvalues for which SELCTG is true = ', sdim,       &
              '(dimension of deflating subspaces)'

            Write (nout,*)
!           Print generalized eigenvalues
            Write (nout,*) 'Selected generalized eigenvalues'

            Do i = 1, sdim
              If (beta(i)/=0.0_nag_wp) Then
                Write (nout,99997) i, '(', alphar(i)/beta(i), ',',             &
                  alphai(i)/beta(i), ')'
              Else
                Write (nout,99996) i
              End If
            End Do
            Write (nout,*)

            If (info==(n+2)) Then
              Write (nout,99995) '***Note that rounding errors mean ',         &
                'that leading eigenvalues in the generalized',                 &
                'Schur form no longer satisfy SELCTG = .TRUE.'
              Write (nout,*)
            End If

          End If

        Else
          Write (nout,99998) 'Failure in DGGES3. INFO =', info
        End If

99999 Format (1X,A,I4,/,1X,A)
99998 Format (1X,A,I4)
99997 Format (1X,I4,5X,A,F7.3,A,F7.3,A)
99996 Format (1X,I4,'Eigenvalue is infinite')
99995 Format (1X,2A,/,1X,A)
    End Program f08xcfe
```

10.2 Program Data

```
F08XCF Example Program Data
   4                       :Value of N
   3.9  12.5 -34.5  -0.5
   4.3  21.5 -47.5   7.5
   4.3  21.5 -43.5   3.5
   4.4  26.0 -46.0   6.0 :End of matrix A
   1.0   2.0  -3.0   1.0
   1.0   3.0  -5.0   4.0
   1.0   3.0  -4.0   3.0
   1.0   3.0  -4.0   4.0 :End of matrix B
```

10.3 Program Results

```
F08XCF Example Program Results

Matrix A
          1          2          3          4
1      3.9000    12.5000   -34.5000    -0.5000
2      4.3000    21.5000   -47.5000     7.5000
3      4.3000    21.5000   -43.5000     3.5000
4      4.4000    26.0000   -46.0000     6.0000

Matrix B
          1          2          3          4
1      1.0000     2.0000    -3.0000     1.0000
2      1.0000     3.0000    -5.0000     4.0000
3      1.0000     3.0000    -4.0000     3.0000
4      1.0000     3.0000    -4.0000     4.0000

Number of eigenvalues for which SELCTG is true =    2
(dimension of deflating subspaces)

Selected generalized eigenvalues
   1     (  2.000,  0.000)
   2     (  4.000,  0.000)
```