NAG Library Routine Document

F08YGF (DTGSEN)

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

F08YGF (DTGSEN) reorders the generalized Schur factorization of a matrix pair in real generalized Schur form, so that a selected cluster of eigenvalues appears in the leading elements, or blocks on the diagonal of the generalized Schur form. The routine also, optionally, computes the reciprocal condition numbers of the cluster of eigenvalues and/or corresponding deflating subspaces.

2 Specification

```
SUBROUTINE F08YGF (IJOB, WANTQ, WANTZ, SELECT, N, A, LDA, B, LDB,          &
                   ALPHAR, ALPHAI, BETA, Q, LDQ, Z, LDZ, M, PL, PR, DIF,    &
                   WORK, LWORK, IWORK, LIWORK, INFO)
INTEGER            IJOB, N, LDA, LDB, LDQ, LDZ, M, LWORK, IWORK(*),         &
                   LIWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), ALPHAR(N), ALPHAI(N), BETA(N),       &
                   Q(LDQ,*), Z(LDZ,*), PL, PR, DIF(*),                      &
                   WORK(max(1,LWORK))
LOGICAL            WANTQ, WANTZ, SELECT(N)
```

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

3 Description

F08YGF (DTGSEN) factorizes the generalized real n by n matrix pair (S,T) in real generalized Schur form, using an orthogonal equivalence transformation as

$$S = \hat{Q}\hat{S}\hat{Z}^{\mathrm{T}}, \quad T = \hat{Q}\hat{T}\hat{Z}^{\mathrm{T}},$$

where $\left(\hat{S},\hat{T}\right)$ are also in real generalized Schur form and have the selected eigenvalues as the leading diagonal elements, or diagonal blocks. The leading columns of Q and Z are the generalized Schur vectors corresponding to the selected eigenvalues and form orthonormal subspaces for the left and right eigenspaces (deflating subspaces) of the pair (S,T).

The pair (S,T) are in real generalized Schur form if S is block upper triangular with 1 by 1 and 2 by 2 diagonal blocks and T is upper triangular as returned, for example, by F08XAF (DGGES), or F08XEF (DHGEQZ) with JOB = 'S'. The diagonal elements, or blocks, define the generalized eigenvalues (α_i,β_i), for $i=1,2,\ldots,n$, of the pair (S,T). The eigenvalues are given by

$$\lambda_i = \alpha_i/\beta_i,$$

but are returned as the pair (α_i,β_i) in order to avoid possible overflow in computing λ_i. Optionally, the routine returns reciprocals of condition number estimates for the selected eigenvalue cluster, p and q, the right and left projection norms, and of deflating subspaces, Dif_u and Dif_l. For more information see Sections 2.4.8 and 4.11 of Anderson *et al.* (1999).

If S and T are the result of a generalized Schur factorization of a matrix pair (A,B)

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

then, optionally, the matrices Q and Z can be updated as $Q\hat{Q}$ and $Z\hat{Z}$. Note that the condition numbers of the pair (S,T) are the same as those of the pair (A,B).

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

5 Arguments

1: IJOB – INTEGER *Input*

On entry: specifies whether condition numbers are required for the cluster of eigenvalues (p and q) or the deflating subspaces (Dif_u and Dif_l).

IJOB $= 0$
: Only reorder with respect to SELECT. No extras.

IJOB $= 1$
: Reciprocal of norms of 'projections' onto left and right eigenspaces with respect to the selected cluster (p and q).

IJOB $= 2$
: The upper bounds on Dif_u and Dif_l. F-norm-based estimate (DIF$(1:2)$).

IJOB $= 3$
: Estimate of Dif_u and Dif_l. 1-norm-based estimate (DIF$(1:2)$). About five times as expensive as IJOB $= 2$.

IJOB $= 4$
: Compute PL, PR and DIF as in IJOB $= 0$, 1 and 2. Economic version to get it all.

IJOB $= 5$
: Compute PL, PR and DIF as in IJOB $= 0$, 1 and 3.

Constraint: $0 \leq \text{IJOB} \leq 5$.

2: WANTQ – LOGICAL *Input*

On entry: if WANTQ $=$ .TRUE., update the left transformation matrix Q.

If WANTQ $=$ .FALSE., do not update Q.

3: WANTZ – LOGICAL *Input*

On entry: if WANTZ $=$ .TRUE., update the right transformation matrix Z.

If WANTZ $=$ .FALSE., do not update Z.

4: SELECT(N) – LOGICAL array *Input*

On entry: specifies the eigenvalues in the selected cluster. To select a real eigenvalue λ_j, SELECT(j) must be set to .TRUE..

To select a complex conjugate pair of eigenvalues λ_j and λ_{j+1}, corresponding to a 2 by 2 diagonal block, either SELECT(j) or SELECT$(j+1)$ or both must be set to .TRUE.; a complex conjugate pair of eigenvalues must be either both included in the cluster or both excluded.

5: N – INTEGER *Input*

On entry: n, the order of the matrices S and T.

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

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

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

On entry: the matrix S in the pair (S, T).

On exit: the updated matrix $\hat{S}$.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08YGF (DTGSEN) is called.

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

8: $\text{B}(\text{LDB}, *)$ – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the matrix T, in the pair (S, T).

On exit: the updated matrix $\hat{T}$

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08YGF (DTGSEN) is called.

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

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

On exit: see the description of BETA.

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

On exit: see the description of BETA.

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

On exit: $\text{ALPHAR}(j)/\text{BETA}(j)$ and $\text{ALPHAI}(j)/\text{BETA}(j)$ are the real and imaginary parts respectively of the jth eigenvalue, for $j = 1, 2, \ldots, \text{N}$.

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

Conjugate pairs of eigenvalues correspond to the 2 by 2 diagonal blocks of $\hat{S}$. These 2 by 2 blocks can be reduced by applying complex unitary transformations to $\left(\hat{S}, \hat{T}\right)$ to obtain the complex Schur form $(\tilde{S}, \tilde{T})$, where $\tilde{S}$ is triangular (and complex). In this form $\text{ALPHAR} + i\text{ALPHAI}$ and BETA are the diagonals of $\tilde{S}$ and $\tilde{T}$ respectively.

13: $\text{Q}(\text{LDQ}, *)$ – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array Q must be at least $\max(1, \text{N})$ if WANTQ = .TRUE., and at least 1 otherwise.

On entry: if WANTQ = .TRUE., the n by n matrix Q.

On exit: if WANTQ = .TRUE., the updated matrix $Q\hat{Q}$.

If WANTQ = .FALSE., Q is not referenced.

14: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08YGF (DTGSEN) is called.

Constraints:

if WANTQ = .TRUE., $\text{LDQ} \geq \max(1, \text{N})$;
otherwise $\text{LDQ} \geq 1$.

15: Z(LDZ, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array Z must be at least $\max(1, \mathrm{N})$ if WANTZ = .TRUE., and at least 1 otherwise.

On entry: if WANTZ = .TRUE., the n by n matrix Z.

On exit: if WANTZ = .TRUE., the updated matrix $Z\hat{Z}$.

If WANTZ = .FALSE., Z is not referenced.

16: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08YGF (DTGSEN) is called.

Constraints:

if WANTZ = .TRUE., LDZ $\geq \max(1, \mathrm{N})$;
otherwise LDZ ≥ 1.

17: M – INTEGER *Output*

On exit: the dimension of the specified pair of left and right eigenspaces (deflating subspaces).

18: PL – REAL (KIND=nag_wp) *Output*
19: PR – REAL (KIND=nag_wp) *Output*

On exit: if IJOB = 1, 4 or 5, PL and PR are lower bounds on the reciprocal of the norm of 'projections' p and q onto left and right eigenspaces with respect to the selected cluster. $0 <$ PL, PR ≤ 1.

If M = 0 or M = N, PL = PR = 1.

If IJOB = 0, 2 or 3, PL and PR are not referenced.

20: DIF($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array DIF must be at least 2.

On exit: if IJOB ≥ 2, DIF$(1:2)$ store the estimates of Dif_u and Dif_l.

If IJOB = 2 or 4, DIF$(1:2)$ are F-norm-based upper bounds on Dif_u and Dif_l.

If IJOB = 3 or 5, DIF$(1:2)$ are 1-norm-based estimates of Dif_u and Dif_l.

If M = 0 or n, DIF$(1:2) = \|(A, B)\|_F$.

If IJOB = 0 or 1, DIF is not referenced.

21: WORK(max(1, LWORK)) – REAL (KIND=nag_wp) array *Workspace*

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

If IJOB = 0, WORK is not referenced.

22: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08YGF (DTGSEN) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, 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 LWORK $\neq -1$,

if N = 0, LWORK ≥ 1;
if IJOB = 1, 2 or 4, LWORK $\geq \max(4 \times \mathrm{N} + 16, 2 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}))$;

if IJOB $= 3$ or 5, LWORK $\geq \max(4 \times \mathrm{N} + 16, 4 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}))$;
otherwise LWORK $\geq 4 \times \mathrm{N} + 16$.

23: IWORK($*$) – INTEGER array *Workspace*

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

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

If IJOB $= 0$, IWORK is not referenced.

24: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08YGF (DTGSEN) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, 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 LIWORK $\neq -1$,

if IJOB $= 1$, 2 or 4, LIWORK $\geq \mathrm{N} + 6$;
if IJOB $= 3$ or 5, LIWORK $\geq \max(2 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}), \mathrm{N} + 6)$;
otherwise LIWORK ≥ 1.

25: 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 $= 1$

Reordering of (S, T) failed because the transformed matrix pair $\left(\hat{S}, \hat{T}\right)$ would be too far from generalized Schur form; the problem is very ill-conditioned. (S, T) may have been partially reordered. If requested, 0 is returned in DIF(1 : 2), PL and PR.

7 Accuracy

The computed generalized Schur form is nearly the exact generalized Schur form for nearby matrices $(S + E)$ and $(T + F)$, where

$$\|E\|_2 = O\,\epsilon\|S\|_2 \quad \text{and} \quad \|F\|_2 = O\,\epsilon\|T\|_2,$$

and ϵ is the ***machine precision***. See Section 4.11 of Anderson *et al.* (1999) for further details of error bounds for the generalized nonsymmetric eigenproblem, and for information on the condition numbers returned.

8 Parallelism and Performance

F08YGF (DTGSEN) 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 complex analogue of this routine is F08YUF (ZTGSEN).

10 Example

This example reorders the generalized Schur factors S and T and update the matrices Q and Z given by

$$S = \begin{pmatrix} 4.0 & 1.0 & 1.0 & 2.0 \\ 0 & 3.0 & 4.0 & 1.0 \\ 0 & 1.0 & 3.0 & 1.0 \\ 0 & 0 & 0 & 6.0 \end{pmatrix}, \quad T = \begin{pmatrix} 2.0 & 1.0 & 1.0 & 3.0 \\ 0 & 1.0 & 2.0 & 1.0 \\ 0 & 0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 2.0 \end{pmatrix},$$

$$Q = \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{pmatrix} \quad \text{and} \quad Z = \begin{pmatrix} 1.0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \end{pmatrix},$$

selecting the first and fourth generalized eigenvalues to be moved to the leading positions. Bases for the left and right deflating subspaces, and estimates of the condition numbers for the eigenvalues and Frobenius norm based bounds on the condition numbers for the deflating subspaces are also output.

10.1 Program Text

```
    Program f08ygfe

!     F08YGF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dtgsen, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: pl, pr
      Integer                          :: i, ijob, info, lda, ldb, ldc, ldq,   &
                                          ldz, liwork, lwork, m, n
      Logical                          :: wantq, wantz
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), alphai(:), alphar(:),        &
                                          b(:,:), beta(:), c(:,:), q(:,:),     &
                                          work(:), z(:,:)
      Real (Kind=nag_wp)               :: dif(2)
      Integer, Allocatable             :: iwork(:)
      Logical, Allocatable             :: select(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08YGF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      ldc = n
      ldq = n
      ldz = n
      liwork = (n*n)/2 + 6
      lwork = n*(n+4) + 16
      Allocate (a(lda,n),alphai(n),alphar(n),b(ldb,n),beta(n),c(ldc,n),        &
        q(ldq,n),work(lwork),z(ldz,n),iwork(liwork),select(n))

!     Read A, B, Q, Z and the logical array SELECT from data file
```

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

      Read (nin,*) select(1:n)

!     Set ijob, wantq and wantz
      ijob = 4
      wantq = .True.
      wantz = .True.

!     Reorder the Schur factors A and B and update the matrices
!     Q and Z

!     The NAG name equivalent of dtgsen is f08ygf
      Call dtgsen(ijob,wantq,wantz,select,n,a,lda,b,ldb,alphar,alphai,beta,q,  &
        ldq,z,ldz,m,pl,pr,dif,work,lwork,iwork,liwork,info)

      If (info>0) Then
        Write (nout,99999) info
        Write (nout,*)
        Flush (nout)
      End If

!     Print Results
      Write (nout,99996) 'Number of selected eigenvalues  = ', m
      Write (nout,*)
      Write (nout,*) 'Selected Generalized Eigenvalues'
      Write (nout,*)
      Write (nout,99997)(i,alphar(i)/beta(i),alphai(i)/beta(i),i=1,m)

      Write (nout,*)
      Write (nout,99998) 'Norm estimate of projection onto',                   &
        ' left  eigenspace for selected cluster', 1.0_nag_wp/pl
      Write (nout,*)
      Write (nout,99998) 'Norm estimate of projection onto',                   &
        ' right eigenspace for selected cluster', 1.0_nag_wp/pr
      Write (nout,*)
      Write (nout,99998) 'F-norm based upper bound on', ' Difu', dif(1)
      Write (nout,*)
      Write (nout,99998) 'F-norm based upper bound on', ' Difl', dif(2)

99999 Format (' Reordering could not be completed. INFO = ',I3)
99998 Format (1X,2A,/,1X,1P,E10.2)
99997 Format (1X,I2,1X,'(',1P,E11.4,',',E11.4,')')
99996 Format (1X,A,I4)
    End Program f08ygfe
```

10.2 Program Data

```
F08YGF Example Program Data
  4                    :Value of N
  4.0  1.0  1.0  2.0
  0.0  3.0  4.0  1.0
  0.0  1.0  3.0  1.0
  0.0  0.0  0.0  6.0   :End of matrix A
  2.0  1.0  1.0  3.0
  0.0  1.0  2.0  1.0
  0.0  0.0  1.0  1.0
  0.0  0.0  0.0  2.0   :End of matrix B
  1.0  0.0  0.0  0.0
  0.0  1.0  0.0  0.0
  0.0  0.0  1.0  0.0
  0.0  0.0  0.0  1.0   :End of matrix Q
  1.0  0.0  0.0  0.0
  0.0  1.0  0.0  0.0
  0.0  0.0  1.0  0.0
  0.0  0.0  0.0  1.0   :End of matrix Z
  T    F    F    T     :End of SELECT
```

10.3 Program Results

```
 F08YGF Example Program Results

 Number of selected eigenvalues  =    2

 Selected Generalized Eigenvalues

  1 ( 2.0000E+00, 0.0000E+00)
  2 ( 3.0000E+00, 0.0000E+00)

 Norm estimate of projection onto left  eigenspace for selected cluster
   2.69E+00

 Norm estimate of projection onto right eigenspace for selected cluster
   1.50E+00

 F-norm based upper bound on Difu
   2.52E-01

 F-norm based upper bound on Difl
   2.45E-01
```