NAG Fortran 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,
1                   ALPHAR, ALPHAI, BETA, Q, LDQ, Z, LDZ, M, PL, PR, DIF,
2                   WORK, LWORK, IWORK, LIWORK, INFO)
 INTEGER            IJOB, N, LDA, LDB, LDQ, LDZ, M, LWORK, IWORK(*),
1                   LIWORK, INFO
 double precision   A(LDA,*), B(LDB,*), ALPHAR(*), ALPHAI(*), BETA(*),
1                   Q(LDQ,*), Z(LDZ,*), PL, PR, DIF(*), WORK(*)
 LOGICAL            WANTQ, WANTZ, SELECT(*)
```

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 $(\hat{S},\hat{T})$ 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 Anderson *et al.* (1999) (Sections 2.4.8 and 4.11).

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

5 Parameters

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 5 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.

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($*$) – LOGICAL array *Input*

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

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

6: A(LDA,$*$) – ***double precision*** array *Input/Output*

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

On entry: the matrix S, in generalized real Schur form.

On exit: the reordered matrix $\hat{S}$ in generalized real Schur canonical form.

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

8: B(LDB,$*$) – ***double precision*** array *Input/Output*

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

On entry: the matrix T, in generalized real Schur form.

On exit: the reordered matrix $\hat{T}$ in generalized real Schur canonical form.

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

10: ALPHAR($*$) – ***double precision*** array *Output*

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

On exit: see the description of BETA.

11: ALPHAI($*$) – ***double precision*** array *Output*

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

On exit: see the description of BETA.

12: BETA($*$) – ***double precision*** array *Output*

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

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

If ALPHAI(j) is zero, then the jth eigenvalue is real; if positive then 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 $(\hat{S}, \hat{T})$ to obtain the complex Schur form $(\tilde{S}, \tilde{T})$, where $\tilde{S}$ is triangular (and complex). In this form ALPHAR + iALPHAI and BETA are the diagonals of $\tilde{S}$ and $\tilde{T}$ respectively.

13: Q(LDQ,$*$) – ***double precision*** array *Input/Output*

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

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

15: Z(LDZ,*) – ***double precision*** array *Input/Output*

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

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

17: M – INTEGER *Output*

On exit: the dimension of the specified pair of left and right eigen-spaces (deflating subspaces) with $0 \leq M \leq N$.

18: PL – ***double precision*** *Output*
19: PR – ***double precision*** *Output*

On exit: if IJOB = 1, 4 or 5, PL, 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 $0 < PL, PR \leq 1$.

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

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

20: DIF(*) – ***double precision*** array *Output*

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

On exit: if $IJOB \geq 2$, 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(*) – ***double precision*** array *Workspace*

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

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 $\geq \max(4 \times \mathrm{N} + 16, 4 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}))$;
  LWORK $\geq 4 \times \mathrm{N} + 16$ otherwise.

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 $\geq \max(2 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}), \mathrm{N} + 6)$;
  LIWORK ≥ 1 otherwise.

25: 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 parameter 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 $(\hat{S}, \hat{T})$ 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 Anderson *et al.* (1999) (Section 4.11) for further details of error bounds for the generalized nonsymmetric eigenproblem, and for information on the condition numbers returned.

8 Further Comments

The complex analogue of this routine is F08YUF (ZTGSEN).

9 Example

To reorder 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.

9.1 Program Text

```
*     F08YGF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=8)
      INTEGER          LDQ, LDS, LDT, LDZ, LIWORK, LWORK
      PARAMETER        (LDQ=NMAX,LDS=NMAX,LDT=NMAX,LDZ=NMAX,
     +                 LIWORK=(NMAX*NMAX)/2+6,LWORK=NMAX*(NMAX+4)+16)
*     .. Local Scalars ..
      DOUBLE PRECISION PL, PR
      INTEGER          I, IFAIL, IJOB, INFO, J, M, N
      LOGICAL          WANTQ, WANTZ
*     .. Local Arrays ..
      DOUBLE PRECISION ALPHAI(NMAX), ALPHAR(NMAX), BETA(NMAX), DIF(2),
     +                 Q(LDQ,NMAX), S(LDS,NMAX), T(LDT,NMAX),
     +                 WORK(LWORK), Z(LDZ,NMAX)
      INTEGER          IWORK(LIWORK)
      LOGICAL          SELECT(NMAX)
*     .. External Subroutines ..
      EXTERNAL         DTGSEN, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08YGF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read S, T, Q, Z and the logical array SELECT from data file
*
         READ (NIN,*) ((S(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((T(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((Q(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((Z(I,J),J=1,N),I=1,N)
*
         READ (NIN,*) (SELECT(I),I=1,N)
*
*        Set IJOB, WANTQ and WANTZ
         IJOB = 4
         WANTQ = .TRUE.
```

```
         WANTZ = .TRUE.
*
*        Reorder the Schur factors S and T and update the matrices
*        Q and Z
*
         CALL DTGSEN(IJOB,WANTQ,WANTZ,SELECT,N,S,LDS,T,LDT,ALPHAR,
     +               ALPHAI,BETA,Q,LDQ,Z,LDZ,M,PL,PR,DIF,WORK,LWORK,
     +               IWORK,LIWORK,INFO)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,99999) INFO
            WRITE (NOUT,*)
         END IF
*
*        Print reordered generalized Schur form
*
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,S,LDS,'Reordered Schur matrix S',
     +               IFAIL)
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CAF('General',' ',N,N,T,LDT,'Reordered Schur matrix T',
     +               IFAIL)
*
*        Print deflating subspaces
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CAF('General',' ',N,M,Q,LDQ,
     +               'Basis of left deflating invariant subspace',IFAIL)
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CAF('General',' ',N,M,Z,LDZ,
     +               'Basis of right deflating invariant subspace',
     +               IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'Norm estimate of projection onto',
     +     ' left  eigenspace for selected cluster', 1.0D0/PL
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'Norm estimate of projection onto',
     +     ' right eigenspace for selected cluster', 1.0D0/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)
      ELSE
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
99999 FORMAT (' Reordering could not be completed. INFO = ',I3)
99998 FORMAT (1X,2A,/1X,1P,E10.2)
      END
```

9.2 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 S
  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 T
```

```
 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
```

9.3 Program Results

```
F08YGF Example Program Results

Reordered Schur matrix S
            1          2          3          4
1      4.0000     1.2247    -1.7055    -1.2615
2      0.0000     2.7386    -3.4009    -4.4423
3      0.0000     0.0000     4.9328    -2.4277
4      0.0000     0.0000    -0.9368    -1.7597

Reordered Schur matrix T
            1          2          3          4
1      2.0000     1.6330    -1.9307    -2.1461
2      0.0000     0.9129    -1.4726    -1.7315
3      0.0000     0.0000     2.2471     0.0000
4      0.0000     0.0000     0.0000    -0.9750

Basis of left deflating invariant subspace
              1            2
1    1.0000E+00   0.0000E+00
2    0.0000E+00   4.4721E-01
3    0.0000E+00   1.1159E-17
4    0.0000E+00   8.9443E-01

Basis of right deflating invariant subspace
         1       2
1   1.0000  0.0000
2   0.0000  0.8165
3   0.0000 -0.4082
4   0.0000  0.4082

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
```