NAG Library Routine Document

F08YUF (ZTGSEN)

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

F08YUF (ZTGSEN) reorders the generalized Schur factorization of a complex matrix pair in generalized Schur form, so that a selected cluster of eigenvalues appears in the leading elements 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 F08YUF(IJOB, WANTQ, WANTZ, SELECT, N, A, LDA, B, LDB, ALPHA,
1                  BETA, Q, LDQ, Z, LDZ, M, PL, PR, DIF, WORK, LWORK,
2                  IWORK, LIWORK, INFO)
 INTEGER           IJOB, N, LDA, LDB, LDQ, LDZ, M, LWORK,
1                  IWORK(max(1,LIWORK)), LIWORK, INFO
 double precision  PL, PR, DIF(*)
 complex*16        A(LDA,*), B(LDB,*), ALPHA(N), BETA(N), Q(LDQ,*),
1                  Z(LDZ,*), WORK(max(1,LWORK))
 LOGICAL           WANTQ, WANTZ, SELECT(N)
```

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

3 Description

F08YUF (ZTGSEN) factorizes the generalized complex n by n matrix pair (S,T) in generalized Schur form, using a unitary equivalence transformation as

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

where $\left(\hat{S},\hat{T}\right)$ are also in generalized Schur form and have the selected eigenvalues as the leading diagonal elements. 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 generalized Schur form if S and T are upper triangular as returned, for example, by F08XNF (ZGGES), or F08XSF (ZHGEQZ) with JOB = 'S'. The diagonal elements 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)

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

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, Philadelphiahttp://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 ($\mathrm{DIF}(1:2)$).

IJOB $= 3$
Estimate of Dif_u and Dif_l. 1-norm-based estimate ($\mathrm{DIF}(1:2)$). 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.

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*

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 an eigenvalue λ_j, SELECT(j) must be set to .TRUE..

5: N – INTEGER *Input*

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

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

6: A(LDA,$*$) – ***complex*16*** array *Input/Output*

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

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 F08YUF (ZTGSEN) is called.

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

8: $\text{B}(\text{LDB},*)$ – ***complex*16*** 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 F08YUF (ZTGSEN) is called.

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

10: $\text{ALPHA}(\text{N})$ – ***complex*16*** array *Output*
11: $\text{BETA}(\text{N})$ – ***complex*16*** array *Output*

On exit: ALPHA and BETA contain diagonal elements of $\hat{S}$ and $\hat{T}$, respectively, when the pair (S, T) has been reduced to generalized Schur form. $\text{ALPHA}(i)/\text{BETA}(i)$, for $i = 1, 2, \ldots, \text{N}$, are the eigenvalues.

12: $\text{Q}(\text{LDQ},*)$ – ***complex*16*** 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.

13: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08YUF (ZTGSEN) is called.

Constraints:

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

14: $\text{Z}(\text{LDZ},*)$ – ***complex*16*** array *Input/Output*

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

15: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08YUF (ZTGSEN) is called.

Constraints:

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

16: M – INTEGER *Output*

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

Constraint: $0 \leq M \leq N$.

17: PL – ***double precision*** *Output*
18: PR – ***double precision*** *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 eigenspace 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.

19: DIF($*$) – ***double precision*** 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.

20: WORK(max(1, LWORK)) – ***complex*16*** array *Workspace*

On exit: if INFO = 0, the real part of WORK(1) contains the minimum 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 F08YUF (ZTGSEN) 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 IJOB = 1, 2 or 4, $LWORK \geq \max(1, 2 \times M \times (N - M))$;
if IJOB = 3 or 5, $LWORK \geq \max(1, 4 \times M \times (N - M))$;
otherwise $LWORK \geq 1$.

.

22: IWORK(max(1, LIWORK)) – INTEGER array *Workspace*

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

23: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08YUF (ZTGSEN) 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 $\geq \mathrm{N} + 2$;
if IJOB $= 3$ or 5, LIWORK $\geq \max(\mathrm{N} + 2, 2 \times \mathrm{M} \times (\mathrm{N} - \mathrm{M}))$;
otherwise LIWORK ≥ 1.

.

24: 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 $= 1$

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

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

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 Further Comments

The real analogue of this routine is F08YGF (DTGSEN).

9 Example

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

$$S = \begin{pmatrix} 4.0 + 4.0i & 1.0 + 1.0i & 1.0 + 1.0i & 2.0 - 1.0i \\ 0 & 2.0 + 1.0i & 1.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 2.0 - 1.0i & 1.0 + 1.0i \\ 0 & 0 & 0 & 6.0 - 2.0i \end{pmatrix},$$

$$T = \begin{pmatrix} 2.0 & 1.0 + 1.0i & 1.0 + 1.0i & 3.0 - 1.0i \\ 0 & 1.0 & 2.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 1.0 & 1.0 + 1.0i \\ 0 & 0 & 0 & 2.0 \end{pmatrix},$$

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

selecting the second and third 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

```
*     F08YUF 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, LDA, LDB, LDZ, LIWORK, LWORK
      PARAMETER        (LDQ=NMAX,LDA=NMAX,LDB=NMAX,LDZ=NMAX,
     +                 LIWORK=(NMAX*NMAX)/2+2,LWORK=NMAX*NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION PL, PR
      INTEGER          I, IFAIL, IJOB, INFO, J, M, N
      LOGICAL          WANTQ, WANTZ
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), ALPHA(NMAX), B(LDB,NMAX),
     +                 BETA(NMAX), Q(LDQ,NMAX), WORK(LWORK), Z(LDZ,NMAX)
      DOUBLE PRECISION DIF(2)
      INTEGER          IWORK(LIWORK)
      LOGICAL          SELECT(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         X04DBF, ZTGSEN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08YUF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A, B, Q, Z and the logical array SELECT from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((B(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 A and B and update the matrices
*        Q and Z
*
         CALL ZTGSEN(IJOB,WANTQ,WANTZ,SELECT,N,A,LDA,B,LDB,ALPHA,BETA,Q,
     +               LDQ,Z,LDZ,M,PL,PR,DIF,WORK,LWORK,IWORK,LIWORK,INFO)
         IF (INFO.NE.0) THEN
            WRITE (NOUT,99999) INFO
            WRITE (NOUT,*)
         END IF
*
*        Print reordered generalized Schur form
```

```
*
         IFAIL = 0
         CALL X04DBF('General',' ',N,N,A,LDA,'Bracketed','F7.4',
     +               'Reordered Schur matrix A','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04DBF('General',' ',N,N,B,LDB,'Bracketed','F7.4',
     +               'Reordered Schur matrix B','Integer',RLABS,
     +               'Integer',CLABS,80,0,IFAIL)
*
*        Print deflating subspaces
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04DBF('General',' ',N,M,Q,LDQ,'Bracketed','F7.4',
     +               'Basis of left deflating invariant subspace',
     +               'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04DBF('General',' ',N,M,Z,LDZ,'Bracketed','F7.4',
     +               'Basis of right deflating invariant subspace',
     +               'Integer',RLABS,'Integer',CLABS,80,0,IFAIL)
*
*        Print norm estimates and F-norm upper bounds
*
         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
*
99999 FORMAT (' Reordering could not be completed. INFO = ',I3)
99998 FORMAT (1X,2A,/1X,1P,E10.2)
      END
```

9.2 Program Data

```
F08YUF Example Program Data
  4                                                           :Value of N
 ( 4.0, 4.0) ( 1.0, 1.0) ( 1.0, 1.0) ( 2.0,-1.0)
 ( 0.0, 0.0) ( 2.0, 1.0) ( 1.0, 1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 2.0,-1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 6.0,-2.0)   :End of matrix A
 ( 2.0, 0.0) ( 1.0, 1.0) ( 1.0, 1.0) ( 3.0,-1.0)
 ( 0.0, 0.0) ( 1.0, 0.0) ( 2.0, 1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 2.0, 0.0)   :End of matrix B
 ( 1.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 1.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0)   :End of matrix Q
 ( 1.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 1.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0) ( 0.0, 0.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0)   :End of matrix Z
   F           T           T           F           :End of SELECT
```

9.3 Program Results

```
 F08YUF Example Program Results

 Reordered Schur matrix A
                    1                 2                 3                 4
 1  ( 4.6904, 2.3452) (-2.1563, 0.1192) ( 1.9599,-0.5174) ( 1.8091,-1.2060)
 2  ( 0.0000, 0.0000) ( 2.0084,-1.0042) ( 0.9161,-0.2762) ( 1.8574,-0.5326)
 3  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 1.6985, 1.6985) ( 0.1270, 0.7231)
 4  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 6.0000,-2.0000)

 Reordered Schur matrix B
                    1                 2                 3                 4
 1  ( 2.3452, 0.0000) (-0.6181, 1.8237) ( 0.9290, 0.5409) ( 2.7136,-1.5076)
 2  ( 0.0000, 0.0000) ( 1.0042, 0.0000) ( 1.2251,-1.1857) ( 1.8541,-0.2929)
 3  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.8492, 0.0000) ( 0.1435, 0.9053)
 4  ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 2.0000, 0.0000)

 Basis of left deflating invariant subspace
                    1                 2
 1  ( 0.9045, 0.3015) (-0.0033,-0.2397)
 2  ( 0.3015, 0.0000) ( 0.2497, 0.7157)
 3  ( 0.0000, 0.0000) ( 0.0549, 0.6042)
 4  ( 0.0000, 0.0000) ( 0.0000, 0.0000)

 Basis of right deflating invariant subspace
                    1                 2
 1  ( 0.7071, 0.0000) (-0.5607, 0.0000)
 2  ( 0.7071, 0.0000) ( 0.5607, 0.0000)
 3  ( 0.0000, 0.0000) ( 0.0552, 0.6067)
 4  ( 0.0000, 0.0000) ( 0.0000, 0.0000)

 Norm estimate of projection onto left  eigenspace for selected cluster
   8.90E+00

 Norm estimate of projection onto right eigenspace for selected cluster
   7.02E+00

 F-norm based upper bound on Difu
   2.18E-01

 F-norm based upper bound on Difl
   2.62E-01
```