NAG Library Routine Document

F08QUF (ZTRSEN)

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

F08QUF (ZTRSEN) reorders the Schur factorization of a complex general matrix so that a selected cluster of eigenvalues appears in the leading elements on the diagonal of the Schur form. The routine also optionally computes the reciprocal condition numbers of the cluster of eigenvalues and/or the invariant subspace.

2 Specification

```
SUBROUTINE F08QUF (JOB, COMPQ, SELECT, N, T, LDT, Q, LDQ, W, M, S, SEP,         &
                   WORK, LWORK, INFO)

INTEGER                 N, LDT, LDQ, M, LWORK, INFO
REAL (KIND=nag_wp)      S, SEP
COMPLEX (KIND=nag_wp)   T(LDT,*), Q(LDQ,*), W(*), WORK(max(1,LWORK))
LOGICAL                 SELECT(*)
CHARACTER(1)            JOB, COMPQ
```

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

3 Description

F08QUF (ZTRSEN) reorders the Schur factorization of a complex general matrix $A = QTQ^{\mathrm{H}}$, so that a selected cluster of eigenvalues appears in the leading diagonal elements of the Schur form.

The reordered Schur form $\tilde{T}$ is computed by a unitary similarity transformation: $\tilde{T} = Z^{\mathrm{H}}TZ$. Optionally the updated matrix $\tilde{Q}$ of Schur vectors is computed as $\tilde{Q} = QZ$, giving $A = \tilde{Q}\tilde{T}\tilde{Q}^{\mathrm{H}}$.

Let $\tilde{T} = \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix}$, where the selected eigenvalues are precisely the eigenvalues of the leading m by m sub-matrix T_{11}. Let $\tilde{Q}$ be correspondingly partitioned as $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ where Q_1 consists of the first m columns of Q. Then $AQ_1 = Q_1T_{11}$, and so the m columns of Q_1 form an orthonormal basis for the invariant subspace corresponding to the selected cluster of eigenvalues.

Optionally the routine also computes estimates of the reciprocal condition numbers of the average of the cluster of eigenvalues and of the invariant subspace.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: JOB – CHARACTER(1) *Input*

On entry: indicates whether condition numbers are required for the cluster of eigenvalues and/or the invariant subspace.

JOB = 'N'
No condition numbers are required.

JOB = 'E'
Only the condition number for the cluster of eigenvalues is computed.

JOB = 'V'
Only the condition number for the invariant subspace is computed.

JOB = 'B'
Condition numbers for both the cluster of eigenvalues and the invariant subspace are computed.

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

2: COMPQ – CHARACTER(1) *Input*

On entry: indicates whether the matrix Q of Schur vectors is to be updated.

COMPQ = 'V'
The matrix Q of Schur vectors is updated.

COMPQ = 'N'
No Schur vectors are updated.

Constraint: COMPQ = 'V' or 'N'.

3: SELECT($*$) – LOGICAL array *Input*

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

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

4: N – INTEGER *Input*

On entry: n, the order of the matrix T.

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

5: T(LDT,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n upper triangular matrix T, as returned by F08PSF (ZHSEQR).

On exit: T is overwritten by the updated matrix $\tilde{T}$.

6: LDT – INTEGER *Input*

On entry: the first dimension of the array T as declared in the (sub)program from which F08QUF (ZTRSEN) is called.

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

7: Q(LDQ,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: if COMPQ = 'V', Q must contain the n by n unitary matrix Q of Schur vectors, as returned by F08PSF (ZHSEQR).

On exit: if COMPQ = 'V', Q contains the updated matrix of Schur vectors; the first m columns of Q form an orthonormal basis for the specified invariant subspace.

If COMPQ = 'N', Q is not referenced.

8: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08QUF (ZTRSEN) is called.

Constraints:

if COMPQ = 'V', LDQ $\geq \max(1, \mathrm{N})$;
if COMPQ = 'N', LDQ ≥ 1.

9: W($*$) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: the reordered eigenvalues of $\tilde{T}$. The eigenvalues are stored in the same order as on the diagonal of $\tilde{T}$.

10: M – INTEGER *Output*

On exit: m, the dimension of the specified invariant subspace, which is the same as the number of selected eigenvalues (see SELECT); $0 \leq m \leq n$.

11: S – REAL (KIND=nag_wp) *Output*

On exit: if JOB = 'E' or 'B', S is a lower bound on the reciprocal condition number of the average of the selected cluster of eigenvalues. If M = 0 or N, S = 1.

If JOB = 'N' or 'V', S is not referenced.

12: SEP – REAL (KIND=nag_wp) *Output*

On exit: if JOB = 'V' or 'B', SEP is the estimated reciprocal condition number of the specified invariant subspace. If M = 0 or N, SEP = $\|T\|$.

If JOB = 'N' or 'E', SEP is not referenced.

13: WORK($\max(1, \mathrm{LWORK})$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

14: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08QUF (ZTRSEN) is called, unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the minimum dimension of WORK.

Constraints:

if JOB = 'N', LWORK ≥ 1 or LWORK $= -1$;
if JOB = 'E', LWORK $\geq \max(1, m \times (\mathrm{N} - m))$ or LWORK $= -1$;
if JOB = 'V' or 'B', LWORK $\geq \max(1, 2m \times (\mathrm{N} - m))$ or LWORK $= -1$.

The actual amount of workspace required cannot exceed $\mathrm{N}^2/4$ if JOB = 'E' or $\mathrm{N}^2/2$ if JOB = 'V' or 'B'.

15: 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.

7 Accuracy

The computed matrix $\tilde{T}$ is similar to a matrix $(T + E)$, where

$$\|E\|_2 = O(\epsilon)\|T\|_2,$$

and ϵ is the ***machine precision***.

S cannot underestimate the true reciprocal condition number by more than a factor of $\sqrt{\min(m, n-m)}$. SEP may differ from the true value by $\sqrt{m(n-m)}$. The angle between the computed invariant subspace and the true subspace is $\frac{O(\epsilon)\|A\|_2}{sep}$.

The values of the eigenvalues are never changed by the reordering.

8 Further Comments

The real analogue of this routine is F08QGF (DTRSEN).

9 Example

This example reorders the Schur factorization of the matrix $A = QTQ^{\mathrm{H}}$ such that the eigenvalues stored in elements t_{11} and t_{44} appear as the leading elements on the diagonal of the reordered matrix $\tilde{T}$, where

$$T = \begin{pmatrix} -6.0004 - 6.9999i & 0.3637 - 0.3656i & -0.1880 + 0.4787i & 0.8785 - 0.2539i \\ 0.0000 + 0.0000i & -5.0000 + 2.0060i & -0.0307 - 0.7217i & -0.2290 + 0.1313i \\ 0.0000 + 0.0000i & 0.0000 + 0.0000i & 7.9982 - 0.9964i & 0.9357 + 0.5359i \\ 0.0000 + 0.0000i & 0.0000 + 0.0000i & 0.0000 + 0.0000i & 3.0023 - 3.9998i \end{pmatrix}$$

and

$$Q = \begin{pmatrix} -0.8347 - 0.1364i & -0.0628 + 0.3806i & 0.2765 - 0.0846i & 0.0633 - 0.2199i \\ 0.0664 - 0.2968i & 0.2365 + 0.5240i & -0.5877 - 0.4208i & 0.0835 + 0.2183i \\ -0.0362 - 0.3215i & 0.3143 - 0.5473i & 0.0576 - 0.5736i & 0.0057 - 0.4058i \\ 0.0086 + 0.2958i & -0.3416 - 0.0757i & -0.1900 - 0.1600i & 0.8327 - 0.1868i \end{pmatrix}.$$

The original matrix A is given in Section 9 in F08NTF (ZUNGHR).

9.1 Program Text

```
    Program f08qufe

!     F08QUF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ajf, x04dbf, zgemm, zlange => f06uaf,  &
                             ztrsen
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: alpha, beta
      Real (Kind=nag_wp)               :: norm, s, sep
      Integer                          :: i, ifail, info, lda, ldc, ldq, ldt,  &
                                          lwork, m, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), c(:,:), q(:,:), t(:,:),    &
                                            w(:), work(:)
      Real (Kind=nag_wp)               :: rwork(1)
      Logical, Allocatable             :: select(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
```

```
      Intrinsic                        :: cmplx
!     .. Executable Statements ..
      Write (nout,*) 'F08QUF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldc = n
      lda = n
      ldq = n
      ldt = n
      lwork = (n*n)/2
      Allocate (a(lda,n),c(ldc,n),q(ldq,n),t(ldt,n),w(n),work(lwork), &
        select(n))

!     Read T, Q and the logical array SELECT from data file

      Read (nin,*)(t(i,1:n),i=1,n)
      Read (nin,*)
      Read (nin,*)(q(i,1:n),i=1,n)
      Read (nin,*)
      Read (nin,*) select(1:n)

!     Compute Q * T * Q**T to find  A
!     The NAG name equivelent of zgemm is f06zaf
      alpha = cmplx(1,kind=nag_wp)
      beta = cmplx(0,kind=nag_wp)
      Call zgemm('N','N',n,n,n,alpha,q,ldq,t,ldt,beta,c,ldc)
      Call zgemm('N','C',n,n,n,alpha,c,ldc,q,ldq,beta,a,lda)

!     Print Matrix A, as computed from Q * T * Q**T
!     ifail: behaviour on error exit
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.4', &
        'Matrix A created from Q*T*Q^T','Integer',rlabs,'Integer',clabs,80,0, &
        ifail)

      Write (nout,*)
      Flush (nout)

!     Reorder the Schur factor T and update the matrix Q to obtain TT and QT

!     The NAG name equivalent of ztrsen is f08quf
      Call ztrsen('Both','Vectors',select,n,t,ldt,q,ldq,w,m,s,sep,work,lwork, &
        info)

!     Compute (Q * T * Q^H) - (QT * TT * QT^H) and store in A,
!     i.e. the difference between reconstructed A using Schur and reordered
!          Schur decompositions.
      alpha = cmplx(1,kind=nag_wp)
      beta = cmplx(0,kind=nag_wp)
      Call zgemm('N','N',n,n,n,alpha,q,ldq,t,ldt,beta,c,ldc)
      alpha = cmplx(-1,kind=nag_wp)
      beta = cmplx(1,kind=nag_wp)
      Call zgemm('N','C',n,n,n,alpha,c,ldc,q,ldq,beta,a,lda)

!     Find norm of difference matrix and print warning if it is too large
!     f06uaf is the NAG name equivalent of the LAPACK auxiliary zlange
      norm = zlange('O',lda,n,a,lda,rwork)
      If (norm>x02ajf()**0.8_nag_wp) Then
        Write (nout,*) 'Norm of A - (QT * TT * QT^H) is much greater than 0.'
        Write (nout,*) 'Schur factorization has failed.'
      Else
!       Print condition estimates
        Write (nout,99999) 'Condition number estimate', &
          ' of the selected cluster of eigenvalues = ', 1.0_nag_wp/s
        Write (nout,*)
        Write (nout,99999) 'Condition number estimate of the specified ', &
```

```
         'invariant subspace    = ', 1.0_nag_wp/sep
      End If

99999 Format (1X,A,A,1P,E10.2)
    End Program f08qufe
```

9.2 Program Data

```
F08QUF Example Program Data
  4                                                        :Value of N
 (-6.0004,-6.9999) ( 0.3637,-0.3656) (-0.1880, 0.4787) ( 0.8785,-0.2539)
 ( 0.0000, 0.0000) (-5.0000, 2.0060) (-0.0307,-0.7217) (-0.2290, 0.1313)
 ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 7.9982,-0.9964) ( 0.9357, 0.5359)
 ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 0.0000, 0.0000) ( 3.0023,-3.9998)
                                                           :End of matrix T
 (-0.8347,-0.1364) (-0.0628, 0.3806) ( 0.2765,-0.0846) ( 0.0633,-0.2199)
 ( 0.0664,-0.2968) ( 0.2365, 0.5240) (-0.5877,-0.4208) ( 0.0835, 0.2183)
 (-0.0362,-0.3215) ( 0.3143,-0.5473) ( 0.0576,-0.5736) ( 0.0057,-0.4058)
 ( 0.0086, 0.2958) (-0.3416,-0.0757) (-0.1900,-0.1600) ( 0.8327,-0.1868)
                                                           :End of matrix Q
  T   F   F   T                                            :End of SELECT
```

9.3 Program Results

```
 F08QUF Example Program Results

 Matrix A created from Q*T*Q^T
                    1                 2                 3                 4
 1  (-3.9702,-5.0406) (-4.1108, 3.7002) (-0.3403, 1.0098) ( 1.2899,-0.8590)
 2  ( 0.3397,-1.5006) ( 1.5201,-0.4301) ( 1.8797,-5.3804) ( 3.3606, 0.6498)
 3  ( 3.3101,-3.8506) ( 2.4996, 3.4504) ( 0.8802,-1.0802) ( 0.6401,-1.4800)
 4  (-1.0999, 0.8199) ( 1.8103,-1.5905) ( 3.2502, 1.3297) ( 1.5701,-3.4397)

 Condition number estimate of the selected cluster of eigenvalues =   1.02E+00

 Condition number estimate of the specified invariant subspace    =   1.82E-01
```