NAG Library Routine Document

F08QGF (DTRSEN)

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

F08QGF (DTRSEN) reorders the Schur factorization of a real general matrix so that a selected cluster of eigenvalues appears in the leading elements or blocks 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 F08QGF (JOB, COMPQ, SELECT, N, T, LDT, Q, LDQ, WR, WI, M, S,     &
                   SEP, WORK, LWORK, IWORK, LIWORK, INFO)
INTEGER            N, LDT, LDQ, M, LWORK, IWORK(max(1,LIWORK)), LIWORK,     &
                   INFO
REAL (KIND=nag_wp) T(LDT,*), Q(LDQ,*), WR(*), WI(*), S, SEP,                &
                   WORK(max(1,LWORK))
LOGICAL            SELECT(*)
CHARACTER(1)       JOB, COMPQ
```

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

3 Description

F08QGF (DTRSEN) reorders the Schur factorization of a real general matrix $A = QTQ^{\mathrm{T}}$, so that a selected cluster of eigenvalues appears in the leading diagonal elements or blocks of the Schur form.

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

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 Arguments

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 $\max(1, N)$.

On entry: the eigenvalues in the selected cluster. To select a real eigenvalue λ_j, SELECT(j) must be set .TRUE.. To select a complex conjugate pair of eigenvalues λ_j and λ_{j+1} (corresponding to a 2 by 2 diagonal block), SELECT(j) and/or SELECT($j+1$) must be set to .TRUE.. A complex conjugate pair of eigenvalues **must** be either both included in the cluster or both excluded. See also Section 9.

4: N – INTEGER *Input*

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

Constraint: $N \geq 0$.

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

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

On entry: the n by n upper quasi-triangular matrix T in canonical Schur form, as returned by F08PEF (DHSEQR). See also Section 9.

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 F08QGF (DTRSEN) is called.

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

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

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

On entry: if COMPQ = 'V', Q must contain the n by n orthogonal matrix Q of Schur vectors, as returned by F08PEF (DHSEQR).

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 F08QGF (DTRSEN) is called.

Constraints:

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

9: WR($*$) – REAL (KIND=nag_wp) array *Output*
10: WI($*$) – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the arrays WR and WI must be at least $\max(1, \text{N})$.

On exit: the real and imaginary parts, respectively, of the reordered eigenvalues of $\tilde{T}$. The eigenvalues are stored in the same order as on the diagonal of $\tilde{T}$; see Section 9 for details. Note that if a complex eigenvalue is sufficiently ill-conditioned, then its value may differ significantly from its value before reordering.

11: M – INTEGER *Output*

On exit: m, the dimension of the specified invariant subspace. The value of m is obtained by counting 1 for each selected real eigenvalue and 2 for each selected complex conjugate pair of eigenvalues (see SELECT); $0 \leq m \leq n$.

12: 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 INFO = 1 (see Section 6), S is set to zero.

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

13: 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 INFO = 1 (see Section 6), SEP is set to zero.

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

14: 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.

15: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08QGF (DTRSEN) 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 $\geq \max(1, \text{N})$ or LWORK $= -1$;
if JOB = 'E', LWORK $\geq \max(1, m \times (\text{N} - m))$ or LWORK $= -1$;
if JOB = 'V' or 'B', LWORK $\geq \max(1, 2m \times (\text{N} - m))$ or LWORK $= -1$.

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

16: IWORK($\max(1, \text{LIWORK})$) – INTEGER array *Workspace*

On exit: if INFO = 0, IWORK(1) contains the required minimal size of LIWORK.

17: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08QGF (DTRSEN) is called, unless $\text{LIWORK} = -1$, in which case a workspace query is assumed and the routine only calculates the minimum dimension of IWORK.

Constraints:

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

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

18: INFO – INTEGER *Output*

On exit: $\text{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

$\text{INFO} < 0$

If $\text{INFO} = -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

$\text{INFO} = 1$

The reordering of T failed because a selected eigenvalue was too close to an eigenvalue which was not selected; this error exit can only occur if at least one of the eigenvalues involved was complex. The problem is too ill-conditioned: consider modifying the selection of eigenvalues so that eigenvalues which are very close together are either all included in the cluster or all excluded. On exit, T may have been partially reordered, but WR, WI and Q (if requested) are updated consistently with T; S and SEP (if requested) are both set to zero.

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 $\frac{O(\epsilon)\|A\|_2}{sep}$.

Note that if a 2 by 2 diagonal block is involved in the reordering, its off-diagonal elements are in general changed; the diagonal elements and the eigenvalues of the block are unchanged unless the block is sufficiently ill-conditioned, in which case they may be noticeably altered. It is possible for a 2 by 2 block to break into two 1 by 1 blocks, i.e., for a pair of complex eigenvalues to become purely real. The values of real eigenvalues however are never changed by the reordering.

8 Parallelism and Performance

F08QGF (DTRSEN) 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 input matrix T must be in canonical Schur form, as is the output matrix $\tilde{T}$. This has the following structure.

If all the computed eigenvalues are real, $\tilde{T}$ is upper triangular, and the diagonal elements of $\tilde{T}$ are the eigenvalues; $\mathrm{WR}(i) = \tilde{t}_{ii}$, for $i = 1, 2, \ldots, n$ and $\mathrm{WI}(i) = 0.0$.

If some of the computed eigenvalues form complex conjugate pairs, then $\tilde{T}$ has 2 by 2 diagonal blocks. Each diagonal block has the form

$$\begin{pmatrix} \tilde{t}_{ii} & \tilde{t}_{i,i+1} \\ \tilde{t}_{i+1,i} & \tilde{t}_{i+1,i+1} \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \alpha \end{pmatrix}$$

where $\beta\gamma < 0$. The corresponding eigenvalues are $\alpha \pm \sqrt{\beta\gamma}$; $\mathrm{WR}(i) = \mathrm{WR}(i+1) = \alpha$; $\mathrm{WI}(i) = +\sqrt{|\beta\gamma|}$; $\mathrm{WI}(i+1) = -\mathrm{WI}(i)$.

The complex analogue of this routine is F08QUF (ZTRSEN).

10 Example

This example reorders the Schur factorization of the matrix $A = QTQ^{\mathrm{T}}$ such that the two real eigenvalues appear as the leading elements on the diagonal of the reordered matrix $\tilde{T}$, where

$$T = \begin{pmatrix} 0.7995 & -0.1144 & 0.0060 & 0.0336 \\ 0.0000 & -0.0994 & 0.2478 & 0.3474 \\ 0.0000 & -0.6483 & -0.0994 & 0.2026 \\ 0.0000 & 0.0000 & 0.0000 & -0.1007 \end{pmatrix}$$

and

$$Q = \begin{pmatrix} 0.6551 & 0.1037 & 0.3450 & 0.6641 \\ 0.5236 & -0.5807 & -0.6141 & -0.1068 \\ -0.5362 & -0.3073 & -0.2935 & 0.7293 \\ 0.0956 & 0.7467 & -0.6463 & 0.1249 \end{pmatrix}.$$

The example program for F08QGF (DTRSEN) illustrates the computation of error bounds for the eigenvalues.

The original matrix A is given in Section 10 in F08NFF (DORGHR).

10.1 Program Text

```
    Program f08qgfe

!     F08QGF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dgemm, dlange => f06raf, dtrsen, nag_wp, x02ajf,  &
                             x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: alpha, beta, norm, s, sep
      Integer                         :: i, ifail, info, lda, ldc, ldq, ldt,  &
                                         liwork, lwork, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: a(:,:), c(:,:), q(:,:), t(:,:),      &
                                         wi(:), work(:), wr(:)
      Integer, Allocatable            :: iwork(:)
      Logical, Allocatable            :: select(:)
!     .. Executable Statements ..
```

```
      Write (nout,*) 'F08QGF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldc = n
      ldq = n
      ldt = n
      liwork = (n*n)/4
      lwork = (n*n)/2
      Allocate (a(lda,n),c(ldc,n),q(ldq,n),t(ldt,n),wi(n),work(lwork),wr(n),   &
        iwork(liwork),select(n))

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

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

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

!     Compute Q * T * Q**T to find  A
!     The NAG name equivalent of dgemm is f06yaf
      alpha = 1._nag_wp
      beta = 0._nag_wp
      Call dgemm('N','N',n,n,n,alpha,q,ldq,t,ldt,beta,c,ldc)
      Call dgemm('N','T',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 x04caf('General',' ',n,n,a,lda,'Matrix A computed from Q*T*Q^T',    &
        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 dtrsen is f08qgf
      Call dtrsen('Both','Vectors',select,n,t,ldt,q,ldq,wr,wi,m,s,sep,work,    &
        lwork,iwork,liwork,info)

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

!     Find norm of difference matrix and print warning if it is too large
!     f06raf is the NAG name equivalent of the LAPACK auxiliary dlange
      norm = dlange('O',lda,n,a,lda,work)
      If (norm>x02ajf()**0.8_nag_wp) Then
        Write (nout,*) 'Norm of A - (QT * TT * QT^T) is much greater than 0.'
        Write (nout,*) 'Schur factorization has failed.'
      Else
!       Print Result
        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 spec',            &
```

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

99999 Format (1X,A,A,1P,E10.2)
    End Program f08qgfe
```

10.2 Program Data

```
F08QGF Example Program Data
  4                                           :Value of N
  0.7995  -0.1144   0.0060   0.0336
  0.0000  -0.0994   0.2478   0.3474
  0.0000  -0.6483  -0.0994   0.2026
  0.0000   0.0000   0.0000  -0.1007   :End of matrix T
  0.6551   0.1037   0.3450   0.6641
  0.5236  -0.5807  -0.6141  -0.1068
 -0.5362  -0.3073  -0.2935   0.7293
  0.0956   0.7467  -0.6463   0.1249   :End of matrix Q
  T   F   F   T                       :End of SELECT
```

10.3 Program Results

```
 F08QGF Example Program Results

 Matrix A computed from Q*T*Q^T
          1       2       3       4
 1   0.3500  0.4500 -0.1400 -0.1700
 2   0.0900  0.0700 -0.5399  0.3500
 3  -0.4400 -0.3300 -0.0300  0.1700
 4   0.2500 -0.3200 -0.1300  0.1100

 Condition number estimate of the selected cluster of eigenvalues =   1.75E+00

 Condition number estimate of the specified invariant subspace    =   3.22E+00
```