NAG Fortran Library Routine Document

F08FGF (SORMTR/DORMTR)

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

F08FGF (SORMTR/DORMTR) multiplies an arbitrary real matrix C by the real orthogonal matrix Q which was determined by F08FEF (SSYTRD/DSYTRD) when reducing a real symmetric matrix to tridiagonal form.

2 Specification

```
 SUBROUTINE F08FGF(SIDE, UPLO, TRANS, M, N, A, LDA, TAU, C, LDC, WORK,
1                  LWORK, INFO)
 ENTRY      sormtr (SIDE, UPLO, TRANS, M, N, A, LDA, TAU, C, LDC, WORK,
1                  LWORK, INFO)
 INTEGER           M, N, LDA, LDC, LWORK, INFO
 real              A(LDA,*), TAU(*), C(LDC,*), WORK(*)
 CHARACTER*1       SIDE, UPLO, TRANS
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine is intended to be used after a call to F08FEF (SSYTRD/DSYTRD), which reduces a real symmetric matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation: $A = QTQ^T$. F08FEF represents the orthogonal matrix Q as a product of elementary reflectors.

This routine may be used to form one of the matrix products

$$QC,\ Q^TC,\ CQ \text{ or } CQ^T,$$

overwriting the result on C (which may be any real rectangular matrix).

A common application of this routine is to transform a matrix Z of eigenvectors of T to the matrix QZ of eigenvectors of A.

4 References

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

5 Parameters

1: SIDE – CHARACTER*1 *Input*

On entry: indicates how Q or Q^T is to be applied to C as follows:

if SIDE = 'L', Q or Q^T is applied to C from the left;

if SIDE = 'R', Q or Q^T is applied to C from the right.

Constraint: SIDE = 'L' or 'R'.

2: UPLO – CHARACTER*1 *Input*

On entry: this **must** be the same parameter UPLO as supplied to F08FEF (SSYTRD/DSYTRD).

Constraint: UPLO = 'U' or 'L'.

3: TRANS – CHARACTER*1 *Input*

On entry: indicates whether Q or Q^T is to be applied to C as follows:

if TRANS = 'N', Q is applied to C;

if TRANS = 'T', Q^T is applied to C.

Constraint: TRANS = 'N' or 'T'.

4: M – INTEGER *Input*

On entry: m, the number of rows of the matrix C; m is also the order of Q if SIDE = 'L'.

Constraint: M $\geq$ 0.

5: N – INTEGER *Input*

On entry: n, the number of columns of the matrix C; n is also the order of Q if SIDE = 'R'.

Constraint: N $\geq$ 0.

6: A(LDA,*) – ***real*** array *Input/Output*

Note: the second dimension of the array A must be at least max(1, M) if SIDE = 'L' and at least max(1, N) if SIDE = 'R'.

On entry: details of the vectors which define the elementary reflectors, as returned by F08FEF (SSYTRD/DSYTRD).

On exit: used as internal workspace prior to being restored and hence is unchanged.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08FGF (SORMTR/DORMTR) is called.

Constraints:

LDA $\geq$ max(1, M) if SIDE = 'L',
LDA $\geq$ max(1, N) if SIDE = 'R'.

8: TAU(*) – ***real*** array *Input*

Note: the dimension of the array TAU must be at least max(1, M − 1) if SIDE = 'L' and at least max(1, N − 1) if SIDE = 'R' .

On entry: further details of the elementary reflectors, as returned by F08FEF (SSYTRD/DSYTRD).

9: C(LDC,*) – ***real*** array *Input/Output*

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

On entry: the m by n matrix C.

On exit: C is overwritten by QC or Q^TC or CQ or CQ^T as specified by SIDE and TRANS.

10: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F08FGF (SORMTR/DORMTR) is called.

Constraint: LDC $\geq$ max(1, M).

11: WORK(*) – ***real*** array *Workspace*

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

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

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08FGF (SORMTR/DORMTR) is called, unless LWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $\mathrm{N} \times nb$ if SIDE = 'L' and at least $\mathrm{M} \times nb$ if SIDE = 'R', where nb is the ***blocksize***.

Constraints:

LWORK $\geq \max(1, \mathrm{N})$ or LWORK $= -1$ if SIDE = 'L',
LWORK $\geq \max(1, \mathrm{M})$ or LWORK $= -1$ if SIDE = 'R'.

13: 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.

7 Accuracy

The computed result differs from the exact result by a matrix E such that

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

where ϵ is the ***machine precision***.

8 Further Comments

The total number of floating-point operations is approximately $2m^2n$ if SIDE = 'L' and $2mn^2$ if SIDE = 'R'.

The complex analogue of this routine is F08FUF (CUNMTR/ZUNMTR).

9 Example

To compute the two smallest eigenvalues, and the associated eigenvectors, of the matrix A, where

$$A = \begin{pmatrix} 2.07 & 3.87 & 4.20 & -1.15 \\ 3.87 & -0.21 & 1.87 & 0.63 \\ 4.20 & 1.87 & 1.15 & 2.06 \\ -1.15 & 0.63 & 2.06 & -1.81 \end{pmatrix}.$$

Here A is symmetric and must first be reduced to tridiagonal form T by F08FEF (SSYTRD/DSYTRD). The program then calls F08JJF (SSTEBZ/DSTEBZ) to compute the requested eigenvalues and F08JKF (SSTEIN/DSTEIN) to compute the associated eigenvectors of T. Finally F08FGF (SORMTR/DORMTR) is called to transform the eigenvectors to those of A.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08FGF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDZ, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LDZ=NMAX,LWORK=64*NMAX)
      real             ZERO
      PARAMETER        (ZERO=0.0e0)
*     .. Local Scalars ..
      real             VL, VU
      INTEGER          I, IFAIL, INFO, J, M, N, NSPLIT
      CHARACTER        UPLO
*     .. Local Arrays ..
      real             A(LDA,NMAX), D(NMAX), E(NMAX), TAU(NMAX),
     +                 W(NMAX), WORK(LWORK), Z(LDZ,NMAX)
      INTEGER          IBLOCK(NMAX), IFAILV(NMAX), ISPLIT(NMAX),
     +                 IWORK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         sormtr, sstebz, sstein, ssytrd, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08FGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
         END IF
*
*        Reduce A to tridiagonal form T = (Q**T)*A*Q
*
         CALL ssytrd(UPLO,N,A,LDA,D,E,TAU,WORK,LWORK,INFO)
*
*        Calculate the two smallest eigenvalues of T (same as A)
*
         CALL sstebz('I','B',N,VL,VU,1,2,ZERO,D,E,M,NSPLIT,W,IBLOCK,
     +               ISPLIT,WORK,IWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (W(I),I=1,M)
*
*           Calculate the eigenvectors of T, storing the result in Z
*
            CALL sstein(N,D,E,M,W,IBLOCK,ISPLIT,Z,LDZ,WORK,IWORK,IFAILV,
     +                  INFO)
*
            IF (INFO.GT.0) THEN
               WRITE (NOUT,*) 'Failure to converge.'
            ELSE
*
*              Calculate the eigenvectors of A = Q * (eigenvectors of T)
*
               CALL sormtr('Left',UPLO,'No transpose',N,M,A,LDA,TAU,Z,
     +                     LDZ,WORK,LWORK,INFO)
```

```
*
*              Print eigenvectors
*
               WRITE (NOUT,*)
               IFAIL = 0
*
               CALL X04CAF('General',' ',N,M,Z,LDZ,'Eigenvectors',IFAIL)
*
            END IF
         END IF
      END IF
      STOP
*
99999 FORMAT (3X,(9F8.4))
      END
```

9.2 Program Data

```
F08FGF Example Program Data
  4                           :Value of N
  'L'                         :Value of UPLO
  2.07
  3.87  -0.21
  4.20   1.87   1.15
 -1.15   0.63   2.06  -1.81   :End of matrix A
```

9.3 Program Results

```
 F08FGF Example Program Results

 Eigenvalues
    -5.0034 -1.9987

 Eigenvectors
          1       2
 1   0.5658 -0.2328
 2  -0.3478  0.7994
 3  -0.4740 -0.4087
 4   0.5781  0.3737
```