

## NAG Library Routine Document

### F08FGF (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 (DORMTR) multiplies an arbitrary real matrix  $C$  by the real orthogonal matrix  $Q$  which was determined by F08FEF (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, &
                  LWORK, INFO)
INTEGER          M, N, LDA, LDC, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), TAU(*), C(LDC,*), WORK(max(1,LWORK))
CHARACTER(1)    SIDE, UPLO, TRANS
```

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

#### 3 Description

F08FGF (DORMTR) is intended to be used after a call to F08FEF (DSYTRD), which reduces a real symmetric matrix  $A$  to symmetric tridiagonal form  $T$  by an orthogonal similarity transformation:  $A = QTQ^T$ . F08FEF (DSYTRD) represents the orthogonal matrix  $Q$  as a product of elementary reflectors.

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

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

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  $C$ .

SIDE = 'L'

$Q$  or  $Q^T$  is applied to  $C$  from the left.

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 (DSYTRD).  
*Constraint:* UPLO = 'U' or 'L'.
- 3: TRANS – CHARACTER(1) *Input*  
*On entry:* indicates whether  $Q$  or  $Q^T$  is to be applied to  $C$ .  
 TRANS = 'N'  
 $Q$  is applied to  $C$ .  
 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 (KIND=nag\_wp) array *Input*  
**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 (DSYTRD).
- 7: LDA – INTEGER *Input*  
*On entry:* the first dimension of the array A as declared in the (sub)program from which F08FGF (DORMTR) is called.  
*Constraints:*  
     if SIDE = 'L',  $LDA \geq \max(1, M)$ ;  
     if SIDE = 'R',  $LDA \geq \max(1, N)$ .
- 8: TAU(\*) – REAL (KIND=nag\_wp) 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 (DSYTRD).
- 9: C(LDC,\*) – REAL (KIND=nag\_wp) 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  $Q^T C$  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 (DORMTR) is called.  
*Constraint:*  $LDC \geq \max(1, M)$ .

- 11: WORK(max(1,LWORK)) – REAL (KIND=nag\_wp) array Workspace  
*On exit:* if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimal performance.
- 12: LWORK – INTEGER Input  
*On entry:* the dimension of the array WORK as declared in the (sub)program from which F08FGF (DORMTR) is called.  
 If LWORK = -1, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.  
*Suggested value:* for optimal performance, LWORK  $\geq N \times nb$  if SIDE = 'L' and at least  $M \times nb$  if SIDE = 'R', where *nb* is the optimal **block size**.  
*Constraints:*  
     if SIDE = 'L', LWORK  $\geq \max(1, N)$  or LWORK = -1;  
     if SIDE = 'R', LWORK  $\geq \max(1, M)$  or LWORK = -1.
- 13: INFO – INTEGER Output  
*On exit:* INFO = 0 unless the routine detects an error (see Section 6).

## 6 Error Indicators and Warnings

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 result differs from the exact result by a matrix *E* such that

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

where  $\epsilon$  is the *machine precision*.

## 8 Parallelism and Performance

F08FGF (DORMTR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08FGF (DORMTR) 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 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 (ZUNMTR).

## 10 Example

This example computes 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 (DSYTRD). The program then calls F08JJF (DSTEBZ) to compute the requested eigenvalues and F08JKF (DSTEIN) to compute the associated eigenvectors of  $T$ . Finally F08FGF (DORMTR) is called to transform the eigenvectors to those of  $A$ .

### 10.1 Program Text

```

Program f08fgfe

!      F08FGF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: dormtr, dstebz, dstein, dsytrd, nag_wp, x04caf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Real (Kind=nag_wp), Parameter      :: zero = 0.0E0_nag_wp
Integer, Parameter                 :: nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)                 :: vl, vu
Integer                             :: i, ifail, info, lda, ldc, lwork, m, &
                                     n, nsplit
Character (1)                       :: uplo
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable    :: a(:,,:), c(:,,:), d(:), e(:), tau(:), &
                                     w(:), work(:)
Integer, Allocatable                :: iblock(:), ifailv(:), isplit(:), &
                                     iwork(:)
!      .. Executable Statements ..
Write (nout,*) 'F08FGF Example Program Results'
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n
lda = n
ldc = n
lwork = 64*n
Allocate (a(lda,n),c(ldc,n),d(n),e(n),tau(n),w(n),work(lwork),iblock(n), &
         ifailv(n),isplit(n),iwork(3*n))

!      Read A from data file

Read (nin,*) uplo
If (uplo=='U') Then
  Read (nin,*)(a(i,i:n),i=1,n)
Else If (uplo=='L') Then
  Read (nin,*)(a(i,1:i),i=1,n)
End If

!      Reduce A to tridiagonal form T = (Q**T)*A*Q
Call dsytrd(uplo,n,a,lda,d,e,tau,work,lwork,info)

!      Calculate the two smallest eigenvalues of T (same as A)

!      The NAG name equivalent of dstebz is f08jjf
Call dstebz('I','B',n,vl,vu,1,2,zero,d,e,m,nsplit,w,iblock,isplit,work, &
         iwork,info)

```

```

Write (nout,*)
If (info>0) Then
  Write (nout,*) 'Failure to converge.'
Else
  Write (nout,*) 'Eigenvalues'
  Write (nout,99999) w(1:m)

!   Calculate the eigenvectors of T, storing the result in C
!   The NAG name equivalent of dstein is f08jkg
  Call dstein(n,d,e,m,w,iblock,isplit,c,ldc,work,iwork,ifailv,info)

  If (info>0) Then
    Write (nout,*) 'Failure to converge.'
  Else

!   Calculate the eigenvectors of A = Q * (eigenvectors of T)
!   The NAG name equivalent of dormtr is f08fgf
  Call dormtr('Left',uplo,'No transpose',n,m,a,lda,tau,c,ldc,work, &
    lwork,info)

!   Print eigenvectors

  Write (nout,*)
  Flush (nout)

!   ifail: behaviour on error exit
!   =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
  ifail = 0
  Call x04caf('General', ' ',n,m,c,ldc,'Eigenvalues',ifail)

  End If
End If

99999 Format (3X,(9F8.4))
End Program f08fgfe

```

## 10.2 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

```

## 10.3 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

```

---