NAG Library Routine Document

F08NGF (DORMHR)

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

F08NGF (DORMHR) multiplies an arbitrary real matrix C by the real orthogonal matrix Q which was determined by F08NEF (DGEHRD) when reducing a real general matrix to Hessenberg form.

2 Specification

```
SUBROUTINE F08NGF (SIDE, TRANS, M, N, ILO, IHI, A, LDA, TAU, C, LDC, WORK,      &
                   LWORK, INFO)

INTEGER            M, N, ILO, IHI, LDA, LDC, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), TAU(*), C(LDC,*), WORK(max(1,LWORK))
CHARACTER(1)       SIDE, TRANS
```

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

3 Description

F08NGF (DORMHR) is intended to be used following a call to F08NEF (DGEHRD), which reduces a real general matrix A to upper Hessenberg form H by an orthogonal similarity transformation: $A = QHQ^{\mathrm{T}}$. F08NEF (DGEHRD) represents the matrix Q as a product of $i_{\mathrm{hi}} - i_{\mathrm{lo}}$ elementary reflectors. Here i_{lo} and i_{hi} are values determined by F08NHF (DGEBAL) when balancing the matrix; if the matrix has not been balanced, $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$.

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

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

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

A common application of this routine is to transform a matrix V of eigenvectors of H to the matrix QV 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: 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'.

3: 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: $M \geq 0$.

4: 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: $N \geq 0$.

5: ILO – INTEGER *Input*
6: IHI – INTEGER *Input*

On entry: these **must** be the same parameters ILO and IHI, respectively, as supplied to F08NEF (DGEHRD).

Constraints:

if SIDE = 'L' and $M > 0$, $1 \leq ILO \leq IHI \leq M$;
if SIDE = 'L' and $M = 0$, $ILO = 1$ and $IHI = 0$;
if SIDE = 'R' and $N > 0$, $1 \leq ILO \leq IHI \leq N$;
if SIDE = 'R' and $N = 0$, $ILO = 1$ and $IHI = 0$.

7: 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 F08NEF (DGEHRD).

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NGF (DORMHR) is called.

Constraints:

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

9: 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 F08NEF (DGEHRD).

10: 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^TC or CQ or CQ^T as specified by SIDE and TRANS.

11: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F08NGF (DORMHR) is called.

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

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

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

13: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08NGF (DORMHR) 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 $\geq \max(1, \mathrm{N})$ or LWORK $= -1$;
if SIDE = 'R', LWORK $\geq \max(1, \mathrm{M})$ or LWORK $= -1$.

14: 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 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 $2nq^2$ if SIDE = 'L' and $2mq^2$ if SIDE = 'R', where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The complex analogue of this routine is F08NUF (ZUNMHR).

9 Example

This example computes all the eigenvalues of the matrix A, where

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix},$$

and those eigenvectors which correspond to eigenvalues λ such that $\mathrm{Re}(\lambda) < 0$. Here A is general and must first be reduced to upper Hessenberg form H by F08NEF (DGEHRD). The program then calls F08PEF (DHSEQR) to compute the eigenvalues, and F08PKF (DHSEIN) to compute the required eigenvectors of H by inverse iteration. Finally F08NGF (DORMHR) is called to transform the eigenvectors of H back to eigenvectors of the original matrix A.

9.1 Program Text

```
    Program f08ngfe

!     F08NGF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dgehrd, dhsein, dhseqr, dormhr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: eig, eig1
      Real (Kind=nag_wp)               :: thresh
      Integer                          :: i, ifail, info, j, k, lda, ldc, ldh, &
                                          ldvl, ldz, lwork, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), c(:,:), h(:,:), tau(:),      &
                                          vl(:,:), wi(:), work(:), wr(:), z(:,:)
      Integer, Allocatable             :: ifaill(:), ifailr(:)
      Logical, Allocatable             :: select(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: aimag, cmplx, real
!     .. Executable Statements ..
      Write (nout,*) 'F08NGF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldz = 1
      lda = n
      ldc = n
      ldh = n
      ldvl = n
      lwork = 64*n
      Allocate (a(lda,n),c(ldc,n),h(ldh,n),tau(n),vl(ldvl,n),wi(n), &
        work(lwork),wr(n),z(ldz,1),ifaill(n),ifailr(n),select(n))

!     Read A from data file

      Read (nin,*)(a(i,1:n),i=1,n)

      Read (nin,*) thresh

!     Reduce A to upper Hessenberg form H = (Q**T)*A*Q
!     The NAG name equivalent of dgehrd is f08nef
      Call dgehrd(n,1,n,a,lda,tau,work,lwork,info)

!     Copy A to H
      h(1:n,1:n) = a(1:n,1:n)
```

```
!     Calculate the eigenvalues of H (same as A)
!     The NAG name equivalent of dhseqr is f08pef
      Call dhseqr('Eigenvalues','No vectors',n,1,n,h,ldh,wr,wi,z,ldz,work, &
        lwork,info)

      Write (nout,*)
      If (info>0) Then
        Write (nout,*) 'Failure to converge.'
      Else
        Write (nout,*) 'Eigenvalues'
        Write (nout,99999)(' (',wr(i),',',wi(i),')',i=1,n)

        Do i = 1, n
          select(i) = wr(i) < thresh
        End Do

!       Calculate the eigenvectors of H (as specified by SELECT),
!       storing the result in C
!       The NAG name equivalent of dhsein is f08pkf
        Call dhsein('Right','QR','No initial vectors',select,n,a,lda,wr,wi,vl, &
          ldvl,c,ldc,n,m,work,ifaill,ifailr,info)

!       Calculate the eigenvectors of A = Q * (eigenvectors of H)
!       The NAG name equivalent of dormhr is f08ngf
        Call dormhr('Left','No transpose',n,m,1,n,a,lda,tau,c,ldc,work,lwork, &
          info)

!       Print eigenvectors

        Write (nout,*)
        Flush (nout)

!       Normalize selected eigenvectors
        j = 0
        k = 1
        Do While (k<=n)
          If (select(k)) Then
            j = j + 1
            If (wi(k)==0.0_nag_wp) Then
              Do i = 2, n
                c(i,j) = c(i,j)/c(1,j)
              End Do
              c(1,j) = 1.0_nag_wp
            Else
              eig1 = cmplx(c(1,j),c(1,j+1),kind=nag_wp)
              c(1,j) = 1.0_nag_wp
              c(1,j+1) = 0.0_nag_wp
              Do i = 2, n
                eig = cmplx(c(i,j),c(i,j+1),kind=nag_wp)
                eig = eig/eig1
                c(i,j) = real(eig)
                c(i,j+1) = aimag(eig)
              End Do
              j = j + 1
              k = k + 1
            End If
          End If
          k = k + 1
        End Do

!       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,'Contents of array C',ifail)

      End If

99999 Format (1X,A,F8.4,A,F8.4,A)
    End Program f08ngfe
```

9.2 Program Data

```
F08NGF Example Program Data
  4                                :Value of N
  0.35   0.45  -0.14  -0.17
  0.09   0.07  -0.54   0.35
 -0.44  -0.33  -0.03   0.17
  0.25  -0.32  -0.13   0.11   :End of matrix A
  0.0                          :Value of THRESH
```

9.3 Program Results

```
 F08NGF Example Program Results

 Eigenvalues
  (  0.7995,  0.0000)
  ( -0.0994,  0.4008)
  ( -0.0994, -0.4008)
  ( -0.1007,  0.0000)

 Contents of array C
             1          2          3
 1      1.0000     0.0000     1.0000
 2     -1.7779     0.3606     2.6491
 3     -0.9521     0.3411     4.7381
 4     -1.2785    -1.6841     5.7614
```