NAG Fortran Library Routine Document

F08NGF (SORMHR/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 (SORMHR/DORMHR) multiplies an arbitrary real matrix C by the real orthogonal matrix Q which was determined by F08NEF (SGEHRD/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,
     1                  WORK, LWORK, INFO)
      ENTRY      sormhr (SIDE, TRANS, M, N, ILO, IHI, A, LDA, TAU, C, LDC,
     1                  WORK, LWORK, INFO)
      INTEGER           M, N, ILO, IHI, LDA, LDC, LWORK, INFO
      real              A(LDA,*), TAU(*), C(LDC,*), WORK(*)
      CHARACTER*1       SIDE, TRANS
```

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

3 Description

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

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 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 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: 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'.

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 (SGEHRD/DGEHRD).

Constraints:

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

7: 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 F08NEF (SGEHRD/DGEHRD).

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

8: LDA – INTEGER *Input*

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

Constraints:

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

9: 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 F08NEF (SGEHRD/DGEHRD).

10: 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 $Q^T C$ 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 (SORMHR/DORMHR) is called.

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

12: WORK($*$) – ***real*** array *Workspace*

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

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

13: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08NGF (SORMHR/DORMHR) 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 $N \times nb$ if SIDE = 'L' and at least $M \times nb$ if SIDE = 'R', where nb is the ***blocksize***.

Constraints:

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

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

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

9 Example

To compute 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 (SGEHRD/DGEHRD). The program then calls F08PEF (SHSEQR/DHSEQR) to compute the eigenvalues, and F08PKF (SHSEIN/DHSEIN) to compute the required eigenvectors of H by inverse iteration. Finally F08NGF (SORMHR/DORMHR) is called to transform the eigenvectors of H back to eigenvectors of the original matrix 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.

```
*     F08NGF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDH, LDZ, LWORK, LDVL, LDVR
      PARAMETER        (NMAX=8,LDA=NMAX,LDH=NMAX,LDZ=1,LWORK=64*NMAX,
     +                 LDVL=NMAX,LDVR=NMAX)
*     .. Local Scalars ..
      real             THRESH
      INTEGER          I, IFAIL, INFO, J, M, N
*     .. Local Arrays ..
      real             A(LDA,NMAX), H(LDH,NMAX), TAU(NMAX),
     +                 VL(LDVL,NMAX), VR(LDVR,NMAX), WI(NMAX),
     +                 WORK(LWORK), WR(NMAX), Z(LDZ,1)
      INTEGER          IFAILL(NMAX), IFAILR(NMAX)
      LOGICAL          SELECT(NMAX)
*     .. External Subroutines ..
      EXTERNAL         sgehrd, shsein, shseqr, sormhr, F06QFF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08NGF 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,*) ((A(I,J),J=1,N),I=1,N)
*
         READ (NIN,*) THRESH
*
*        Reduce A to upper Hessenberg form H = (Q**T)*A*Q
*
         CALL sgehrd(N,1,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Copy A to H
*
         CALL F06QFF('General',N,N,A,LDA,H,LDH)
*
*        Calculate the eigenvalues of H (same as A)
*
         CALL shseqr('Eigenvalues','No vectors',N,1,N,H,LDH,WR,WI,Z,LDZ,
     +               WORK,LWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
```

```
         ELSE
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (' (',WR(I),',',WI(I),')',I=1,N)
*
            DO 20 I = 1, N
               SELECT(I) = WR(I) .LT. THRESH
   20       CONTINUE
*
*           Calculate the eigenvectors of H (as specified by SELECT),
*           storing the result in VR
*
            CALL shsein('Right','QR','No initial vectors',SELECT,N,A,
     +                  LDA,WR,WI,VL,LDVL,VR,LDVR,N,M,WORK,IFAILL,
     +                  IFAILR,INFO)
*
*           Calculate the eigenvectors of A = Q * (eigenvectors of H)
*
            CALL sormhr('Left','No transpose',N,M,1,N,A,LDA,TAU,VR,LDVR,
     +                  WORK,LWORK,INFO)
*
*           Print eigenvectors
*
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04CAF('General',' ',N,M,VR,LDVR,
     +                  'Contents of array VR',IFAIL)
*
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,F8.4,A,F8.4,A)
      END
```

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 VR
          1       2       3
 1   0.3881  0.0574  0.1493
 2  -0.7107  0.0380  0.3956
 3  -0.3891  0.0778  0.7075
 4  -0.3996 -0.7270  0.8603
```