NAG Library Routine Document

F08NFF (DORGHR)

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

F08NFF (DORGHR) generates the real orthogonal matrix Q which was determined by F08NEF (DGEHRD) when reducing a real general matrix A to Hessenberg form.

2 Specification

```
SUBROUTINE F08NFF(N, ILO, IHI, A, LDA, TAU, WORK, LWORK, INFO)
INTEGER            N, ILO, IHI, LDA, LWORK, INFO
double precision   A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

3 Description

F08NFF (DORGHR) 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 generate Q explicitly as a square matrix. Q has the structure:

$$Q = \begin{pmatrix} I & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & I \end{pmatrix}$$

where Q_{22} occupies rows and columns i_{lo} to i_{hi}.

4 References

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

5 Parameters

1: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

2: ILO – INTEGER *Input*
3: IHI – INTEGER *Input*

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

Constraints:

if $\mathrm{N} > 0$, $1 \leq \mathrm{ILO} \leq \mathrm{IHI} \leq \mathrm{N}$;
if $\mathrm{N} = 0$, $\mathrm{ILO} = 1$ and $\mathrm{IHI} = 0$.

4: A(LDA,$*$) – ***double precision*** array *Input/Output*

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

On entry: details of the vectors which define the elementary reflectors, as returned by F08NEF (DGEHRD).

On exit: the n by n orthogonal matrix Q.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NFF (DORGHR) is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

6: TAU($*$) – ***double precision*** array *Input*

Note: the dimension of the array TAU must be at least $\max(1, \mathrm{N} - 1)$.

On entry: further details of the elementary reflectors, as returned by F08NEF (DGEHRD).

7: WORK($\max(1, \mathrm{LWORK})$) – ***double precision*** array *Workspace*

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

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08NFF (DORGHR) is called, unless $\mathrm{LWORK} = -1$, 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 optimal performance LWORK should be at least $(\mathrm{IHI} - \mathrm{ILO}) \times nb$, where nb is the ***block size***.

Constraint: $\mathrm{LWORK} \geq \max(1, \mathrm{IHI} - \mathrm{ILO})$ or $\mathrm{LWORK} = -1$.

9: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\mathrm{INFO} < 0$

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

7 Accuracy

The computed matrix Q differs from an exactly orthogonal matrix by a matrix E such that

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

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

8 Further Comments

The total number of floating-point operations is approximately $\frac{4}{3}q^3$, where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The complex analogue of this routine is F08NTF (ZUNGHR).

9 Example

This example computes the Schur factorization of the matrix A, where

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

Here A is general and must first be reduced to Hessenberg form by F08NEF (DGEHRD). The program then calls F08NFF (DORGHR) to form Q, and passes this matrix to F08PEF (DHSEQR) which computes the Schur factorization of A.

9.1 Program Text

```
*     F08NFF 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-1))
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), TAU(NMAX), WI(NMAX), WORK(LWORK),
     +                 WR(NMAX), Z(LDZ,NMAX)
*     .. External Subroutines ..
      EXTERNAL         DGEHRD, DHSEQR, DORGHR, F06QFF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08NFF 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)
*
*        Reduce A to upper Hessenberg form H = (Q**T)*A*Q
*
         CALL DGEHRD(N,1,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Copy A into Z
*
         CALL F06QFF('General',N,N,A,LDA,Z,LDZ)
*
*        Form Q explicitly, storing the result in Z
*
         CALL DORGHR(N,1,N,Z,LDZ,TAU,WORK,LWORK,INFO)
*
*        Calculate the Schur factorization of H = Y*T*(Y**T) and form
*        Q*Y explicitly, storing the result in Z
*
*        Note that A = Z*T*(Z**T), where Z = Q*Y
*
         CALL DHSEQR('Schur form','Vectors',N,1,N,A,LDA,WR,WI,Z,LDZ,
     +               WORK,LWORK,INFO)
*
*        Print Schur form
*
         WRITE (NOUT,*)
         IFAIL = 0
*
         CALL X04CAF('General',' ',N,N,A,LDA,'Schur form',IFAIL)
*
*        Print Schur vectors
*
```

```
         WRITE (NOUT,*)
         IFAIL = 0
*
         CALL X04CAF('General',' ',N,N,Z,LDZ,'Schur vectors of A',IFAIL)
*
      END IF
*
      END
```

9.2 Program Data

```
F08NFF 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
```

9.3 Program Results

```
 F08NFF Example Program Results

 Schur form
          1       2       3       4
 1   0.7995  0.0060 -0.1144 -0.0336
 2   0.0000 -0.0994 -0.6483 -0.2026
 3   0.0000  0.2478 -0.0994 -0.3474
 4   0.0000  0.0000  0.0000 -0.1007

 Schur vectors of A
          1       2       3       4
 1  -0.6551 -0.3450 -0.1037  0.6641
 2  -0.5236  0.6141  0.5807 -0.1068
 3   0.5362  0.2935  0.3073  0.7293
 4  -0.0956  0.6463 -0.7467  0.1249
```