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
REAL (KIND=nag_wp)  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,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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: $LDA \geq \max(1, N)$.

6: TAU($*$) – REAL (KIND=nag_wp) array *Input*

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

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

7: 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.

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 $LWORK = -1$, 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 $(IHI - ILO) \times nb$, where nb is the ***block size***.

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

9: 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 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 $q = i_{hi} - i_{lo}$.

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

```
    Program f08nffe

!     F08NFF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dgehrd, dgemm, dhseqr, dlange => f06raf, dorghr,  &
                             nag_wp, x02ajf, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: alpha, beta, norm
      Integer                          :: i, ifail, info, lda, ldc, ldd, ldz,  &
                                          lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), c(:,:), d(:,:), tau(:),      &
                                          wi(:), work(:), wr(:), z(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F08NFF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldz = n
      ldc = n
      ldd = n
      lwork = 64*(n-1)
      Allocate (a(lda,n),c(ldc,n),d(ldd,n),tau(n),wi(n),work(lwork),wr(n), &
        z(ldz,n))

!     Read A from data file
      Read (nin,*)(a(i,1:n),i=1,n)

!     Copy A into D.
      d(1:n,1:n) = a(1:n,1:n)

      Write (nout,*)
      Flush (nout)

!     Print Matrix A
!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',n,n,a,lda,'Matrix A',ifail)

      Write (nout,*)
      Flush (nout)

!     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 into Z
      z(1:n,1:n) = a(1:n,1:n)

!     Form Q explicitly, storing the result in Z
!     The NAG name equivalent of dorghr is f08nff
      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
!     The NAG name equivalent of dhseqr is f08pef
      Call dhseqr('Schur form','Vectors',n,1,n,a,lda,wr,wi,z,ldz,work,lwork, &
        info)

!     Compute A - Z*T*Z^T from the factorization of A and store in matrix D.
!     The NAG name equivelent of dgemm is f06yaf.
      alpha = 1.0_nag_wp
      beta = 0.0_nag_wp
      Call dgemm('N','N',n,n,n,alpha,z,ldz,a,lda,beta,c,ldc)
      alpha = -1.0_nag_wp
      beta = 1.0_nag_wp
      Call dgemm('N','T',n,n,n,alpha,c,ldc,z,ldz,beta,d,ldd)

!     Find norm of difference matrix D and warn if it is too large;
!     f06raf is the NAG name equivalent of the LAPACK auxiliary dlange
      norm = dlange('O',ldd,n,d,ldd,work)
      If (norm>x02ajf()**0.8_nag_wp) Then
        Write (nout,*) 'Norm of A-(Z*T*Z^T) is much greater than 0.'
        Write (nout,*) 'Schur factorization has failed.'
      Else
!       Print eigenvalues.
        Write (nout,*) 'Eigenvalues'
        Write (nout,99999)(' (',wr(i),',',wi(i),')',i=1,n)
      End If

99999 Format (1X,A,F8.4,A,F8.4,A)

    End Program f08nffe
```

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

 Matrix A
          1       2       3       4
 1   0.3500  0.4500 -0.1400 -0.1700
 2   0.0900  0.0700 -0.5400  0.3500
 3  -0.4400 -0.3300 -0.0300  0.1700
 4   0.2500 -0.3200 -0.1300  0.1100

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