NAG Library Routine Document

F08GFF (DOPGTR)

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

F08GFF (DOPGTR) generates the real orthogonal matrix Q, which was determined by F08GEF (DSPTRD) when reducing a symmetric matrix to tridiagonal form.

2 Specification

```
SUBROUTINE F08GFF (UPLO, N, AP, TAU, Q, LDQ, WORK, INFO)

INTEGER            N, LDQ, INFO
REAL (KIND=nag_wp) AP(*), TAU(*), Q(LDQ,*), WORK(N-1)
CHARACTER(1)       UPLO
```

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

3 Description

F08GFF (DOPGTR) is intended to be used after a call to F08GEF (DSPTRD), which reduces a real symmetric matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation: $A = QTQ^{\mathrm{T}}$. F08GEF (DSPTRD) represents the orthogonal matrix Q as a product of $n-1$ elementary reflectors.

This routine may be used to generate Q explicitly as a square matrix.

4 References

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

5 Parameters

1: UPLO – CHARACTER(1) *Input*

On entry: this **must** be the same parameter UPLO as supplied to F08GEF (DSPTRD).

Constraint: UPLO = 'U' or 'L'.

2: N – INTEGER *Input*

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

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

3: AP($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array AP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: details of the vectors which define the elementary reflectors, as returned by F08GEF (DSPTRD).

4: TAU($*$) – REAL (KIND=nag_wp) 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 F08GEF (DSPTRD).

5: Q(LDQ,$*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: the n by n orthogonal matrix Q.

6: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08GFF (DOPGTR) is called.

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

7: WORK(N $-$ 1) – REAL (KIND=nag_wp) array *Workspace*

8: 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 $\frac{4}{3}n^3$.

The complex analogue of this routine is F08GTF (ZUPGTR).

9 Example

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

$$A = \begin{pmatrix} 2.07 & 3.87 & 4.20 & -1.15 \\ 3.87 & -0.21 & 1.87 & 0.63 \\ 4.20 & 1.87 & 1.15 & 2.06 \\ -1.15 & 0.63 & 2.06 & -1.81 \end{pmatrix},$$

using packed storage. Here A is symmetric and must first be reduced to tridiagonal form by F08GEF (DSPTRD). The program then calls F08GFF (DOPGTR) to form Q, and passes this matrix to F08JEF (DSTEQR) which computes the eigenvalues and eigenvectors of A.

9.1 Program Text

```
    Program f08gffe

!     F08GFF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dopgtr, dsptrd, dsteqr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, j, ldq, n
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ap(:), d(:), e(:), q(:,:), tau(:),   &
                                          work(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08GFF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldq = n
      Allocate (ap(n*(n+1)/2),d(n),e(n),q(ldq,n),tau(n),work(2*n-2))

!     Read A from data file

      Read (nin,*) uplo
      If (uplo=='U') Then
        Read (nin,*)((ap(i+j*(j-1)/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+(2*n-j)*(j-1)/2),j=1,i),i=1,n)
      End If

!     Reduce A to tridiagonal form T = (Q**T)*A*Q
!     The NAG name equivalent of dsptrd is f08gef
      Call dsptrd(uplo,n,ap,d,e,tau,info)

!     Form Q explicitly, storing the result in Q
!     The NAG name equivalent of dopgtr is f08gff
      Call dopgtr(uplo,n,ap,tau,q,ldq,work,info)

!     Calculate all the eigenvalues and eigenvectors of A
!     The NAG name equivalent of dsteqr is f08jef
      Call dsteqr('V',n,d,e,q,ldq,work,info)

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

!       Print eigenvalues and eigenvectors

        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) d(1:n)
        Write (nout,*)
        Flush (nout)

!       Normalize the eigenvectors
        Do i = 1, n
          q(1:n,i) = q(1:n,i)/q(1,i)
        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,n,q,ldq,'Eigenvectors',ifail)
```

```
      End If

99999 Format (3X,(8F8.4))
    End Program f08gffe
```

9.2 Program Data

```
F08GFF 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
```

9.3 Program Results

```
 F08GFF Example Program Results

 Eigenvalues
    -5.0034 -1.9987  0.2013  8.0008

 Eigenvectors
             1          2          3          4
 1      1.0000     1.0000     1.0000     1.0000
 2     -0.6148    -3.4333     0.4489     0.6668
 3     -0.8378     1.7553    -1.3572     0.8248
 4      1.0219    -1.6052    -1.8213     0.0988
```