NAG Library Routine Document

F08GTF (ZUPGTR)

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

F08GTF (ZUPGTR) generates the complex unitary matrix Q, which was determined by F08GSF (ZHPTRD) when reducing a Hermitian matrix to tridiagonal form.

2 Specification

```
SUBROUTINE F08GTF (UPLO, N, AP, TAU, Q, LDQ, WORK, INFO)

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

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

3 Description

F08GTF (ZUPGTR) is intended to be used after a call to F08GSF (ZHPTRD), which reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. F08GSF (ZHPTRD) represents the unitary 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 Arguments

1: UPLO – CHARACTER(1) *Input*

On entry: this **must** be the same argument UPLO as supplied to F08GSF (ZHPTRD).

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($*$) – COMPLEX (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 F08GSF (ZHPTRD).

4: TAU($*$) – COMPLEX (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 F08GSF (ZHPTRD).

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

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

On exit: the n by n unitary matrix Q.

6: LDQ – INTEGER *Input*

On entry: the first dimension of the array Q as declared in the (sub)program from which F08GTF (ZUPGTR) is called.

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

7: WORK(N – 1) – COMPLEX (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

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 unitary matrix by a matrix E such that

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

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

8 Parallelism and Performance

F08GTF (ZUPGTR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08GTF (ZUPGTR) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of real floating-point operations is approximately $\frac{16}{3}n^3$.

The real analogue of this routine is F08GFF (DOPGTR).

10 Example

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

$$A = \begin{pmatrix} -2.28 + 0.00i & 1.78 - 2.03i & 2.26 + 0.10i & -0.12 + 2.53i \\ 1.78 + 2.03i & -1.12 + 0.00i & 0.01 + 0.43i & -1.07 + 0.86i \\ 2.26 - 0.10i & 0.01 - 0.43i & -0.37 + 0.00i & 2.31 - 0.92i \\ -0.12 - 2.53i & -1.07 - 0.86i & 2.31 + 0.92i & -0.73 + 0.00i \end{pmatrix},$$

using packed storage. Here A is Hermitian and must first be reduced to tridiagonal form by F08GSF

(ZHPTRD). The program then calls F08GTF (ZUPGTR) to form Q, and passes this matrix to F08JSF (ZSTEQR) which computes the eigenvalues and eigenvectors of A.

10.1 Program Text

```
    Program f08gtfe

!     F08GTF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dznrm2, nag_wp, x04dbf, zhptrd, zsteqr, zupgtr
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: scal
      Integer                          :: i, ifail, info, j, k, ldq, n
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ap(:), q(:,:), tau(:), work(:)
      Real (Kind=nag_wp), Allocatable  :: d(:), e(:), rwork(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, conjg, maxloc
!     .. Executable Statements ..
      Write (nout,*) 'F08GTF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldq = n
      Allocate (ap(n*(n+1)/2),q(ldq,n),tau(n),work(n-1),d(n),e(n),rwork(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**H)*A*Q
!     The NAG name equivalent of zhptrd is f08gsf
      Call zhptrd(uplo,n,ap,d,e,tau,info)

!     Form Q explicitly, storing the result in Q
!     The NAG name equivalent of zupgtr is f08gtf
      Call zupgtr(uplo,n,ap,tau,q,ldq,work,info)

!     Calculate all the eigenvalues and eigenvectors of A
!     The NAG name equivalent of zsteqr is f08jsf
      Call zsteqr('V',n,d,e,q,ldq,rwork,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, largest element real
```

```
        Do i = 1, n
          rwork(1:n) = abs(q(1:n,i))
          k = maxloc(rwork(1:n),1)
          scal = conjg(q(k,i))/abs(q(k,i))/dznrm2(n,q(1,i),1)
          q(1:n,i) = q(1:n,i)*scal
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,n,q,ldq,'Bracketed','F7.4','Eigenvectors', &
          'Integer',rlabs,'Integer',clabs,80,0,ifail)

      End If

99999 Format (8X,4(F7.4,11X,:))
    End Program f08gtfe
```

10.2 Program Data

```
F08GTF Example Program Data
  4                                                        :Value of N
  'L'                                                      :Value of UPLO
 (-2.28, 0.00)
 ( 1.78, 2.03) (-1.12, 0.00)
 ( 2.26,-0.10) ( 0.01,-0.43) (-0.37, 0.00)
 (-0.12,-2.53) (-1.07,-0.86) ( 2.31, 0.92) (-0.73, 0.00)   :End of matrix A
```

10.3 Program Results

```
 F08GTF Example Program Results

 Eigenvalues
        -6.0002            -3.0030             0.5036             3.9996

 Eigenvectors
                  1                  2                  3                  4
 1  ( 0.7299, 0.0000) (-0.2120, 0.1497) ( 0.1000,-0.3570) ( 0.1991, 0.4720)
 2  (-0.1663,-0.2061) ( 0.7307,-0.0000) ( 0.2863,-0.3353) (-0.2467, 0.3751)
 3  (-0.4165,-0.1417) (-0.3291, 0.0479) ( 0.6890, 0.0000) ( 0.4468, 0.1466)
 4  ( 0.1743, 0.4162) ( 0.5200, 0.1329) ( 0.0662, 0.4347) ( 0.5612, 0.0000)
```