

NAG Library Routine Document

F08GUF (ZUPMTR)

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

F08GUF (ZUPMTR) multiplies an arbitrary complex matrix C by the complex unitary matrix Q which was determined by F08GSF (ZHPTRD) when reducing a complex Hermitian matrix to tridiagonal form.

2 Specification

```
SUBROUTINE F08GUF (SIDE, UPLO, TRANS, M, N, AP, TAU, C, LDC, WORK, INFO)
INTEGER                M, N, LDC, INFO
COMPLEX (KIND=nag_wp) AP(*), TAU(*), C(LDC,*), WORK(*)
CHARACTER(1)          SIDE, UPLO, TRANS
```

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

3 Description

F08GUF (ZUPMTR) 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: $A = QTQ^H$. F08GSF (ZHPTRD) represents the unitary matrix Q as a product of elementary reflectors.

This routine may be used to form one of the matrix products

$$QC, Q^H C, CQ \text{ or } CQ^H,$$

overwriting the result on C (which may be any complex rectangular matrix).

A common application of this routine is to transform a matrix Z of eigenvectors of T to the matrix QZ 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 Arguments

- 1: SIDE – CHARACTER(1) *Input*
On entry: indicates how Q or Q^H is to be applied to C .
 SIDE = 'L'
 Q or Q^H is applied to C from the left.
 SIDE = 'R'
 Q or Q^H is applied to C from the right.
Constraint: SIDE = 'L' or 'R'.
- 2: UPLO – CHARACTER(1) *Input*
On entry: this **must** be the same argument UPLO as supplied to F08GSF (ZHPTRD).
Constraint: UPLO = 'U' or 'L'.

- 3: TRANS – CHARACTER(1) *Input*
On entry: indicates whether Q or Q^H is to be applied to C .
 TRANS = 'N'
 Q is applied to C .
 TRANS = 'C'
 Q^H is applied to C .
Constraint: TRANS = 'N' or 'C'.
- 4: 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: $M \geq 0$.
- 5: 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: $N \geq 0$.
- 6: AP(*) – COMPLEX (KIND=nag_wp) array *Input/Output*
Note: the dimension of the array AP must be at least $\max(1, M \times (M + 1)/2)$ if SIDE = 'L' and at least $\max(1, N \times (N + 1)/2)$ if SIDE = 'R'.
On entry: details of the vectors which define the elementary reflectors, as returned by F08GSF (ZHPTRD).
On exit: is used as internal workspace prior to being restored and hence is unchanged.
- 7: TAU(*) – COMPLEX (KIND=nag_wp) 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 F08GSF (ZHPTRD).
- 8: C(LDC, *) – COMPLEX (KIND=nag_wp) 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 $Q^H C$ or CQ or CQ^H as specified by SIDE and TRANS.
- 9: LDC – INTEGER *Input*
On entry: the first dimension of the array C as declared in the (sub)program from which F08GUF (ZUPMTR) is called.
Constraint: $LDC \geq \max(1, M)$.
- 10: WORK(*) – COMPLEX (KIND=nag_wp) array *Workspace*
Note: the dimension of the array WORK must be at least $\max(1, N)$ if SIDE = 'L' and at least $\max(1, M)$ if SIDE = 'R'.
- 11: 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 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 Parallelism and Performance

F08GUF (ZUPMTR) 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 $8m^2n$ if SIDE = 'L' and $8mn^2$ if SIDE = 'R'.

The real analogue of this routine is F08GGF (DOPMTR).

10 Example

This example computes the two smallest eigenvalues, and the associated 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 T by F08GSF (ZHPTRD). The program then calls F08JJF (DSTEBZ) to compute the requested eigenvalues and F08JXF (ZSTEIN) to compute the associated eigenvectors of T . Finally F08GUF (ZUPMTR) is called to transform the eigenvectors to those of A .

10.1 Program Text

Program f08gufe

```
!      F08GUF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
!      Use nag_library, Only: dstebz, dznrm2, nag_wp, x04dbf, zhptrd, zstein, &
!                               zupmtr
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Real (Kind=nag_wp), Parameter      :: zero = 0.0E0_nag_wp
!      Integer, Parameter                  :: nin = 5, nout = 6
```

```

! .. Local Scalars ..
Complex (Kind=nag_wp)      :: scal
Real (Kind=nag_wp)        :: vl, vu
Integer                    :: i, ifail, info, j, k, ldc, m, n,      &
                           nsplit
Character (1)              :: uplo
! .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: ap(:), c(:,,:), tau(:), work(:)
Real (Kind=nag_wp), Allocatable   :: d(:), e(:), rwork(:), w(:)
Integer, Allocatable              :: iblock(:), ifailv(:), isplit(:),  &
                                   iwork(:)
Character (1)                     :: clabs(1), rlabs(1)
! .. Intrinsic Procedures ..
Intrinsic                          :: abs, conjg, maxloc
! .. Executable Statements ..
Write (nout,*) 'F08GUF Example Program Results'
! Skip heading in data file
Read (nin,*)
Read (nin,*) n
ldc = n
Allocate (ap(n*(n+1)/2),c(ldc,n),tau(n),work(n),d(n),e(n),rwork(5*n),w(n &
),iblock(n),ifailv(n),isplit(n),iwork(3*n))

! 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)

! Calculate the two smallest eigenvalues of T (same as A)

! The NAG name equivalent of dstebz is f08jjf
Call dstebz('I','B',n,vl,vu,1,2,zero,d,e,m,nsplit,w,iblock,isplit,rwork, &
iwork,info)

Write (nout,*)
If (info>0) Then
  Write (nout,*) 'Failure to converge.'
Else
  Write (nout,*) 'Eigenvalues'
  Write (nout,99999) w(1:m)

! Calculate the eigenvectors of T, storing the result in C
! The NAG name equivalent of zstein is f08jxf
Call zstein(n,d,e,m,w,iblock,isplit,c,ldc,rwork,iwork,ifailv,info)

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

! Calculate the eigenvectors of A = Q * (eigenvectors of T)
! The NAG name equivalent of zupmtr is f08guf
Call zupmtr('Left',uplo,'No transpose',n,m,ap,tau,c,ldc,work,info)

! Print eigenvectors

Write (nout,*)
Flush (nout)

! Normalize the eigenvectors, largest element real
Do i = 1, m
  rwork(1:n) = abs(c(1:n,i))
  k = maxloc(rwork(1:n),1)
  scal = conjg(c(k,i))/abs(c(k,i))/dznrm2(n,c(1,i),1)

```

```

        c(1:n,i) = c(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,m,c,ldc,'Bracketed','F7.4',      &
        'Eigenvectors','Integer',rlabs,'Integer',clabs,80,0,ifail)

    End If
End If

99999 Format (8X,4(F7.4,11X,:))
End Program f08gufe

```

10.2 Program Data

F08GUF 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

F08GUF Example Program Results

```

Eigenvalues
    -6.0002          -3.0030

```

```

Eigenvectors
           1           2
1  ( 0.7299, 0.0000) (-0.2120, 0.1497)
2  (-0.1663,-0.2061) ( 0.7307,-0.0000)
3  (-0.4165,-0.1417) (-0.3291, 0.0479)
4  ( 0.1743, 0.4162) ( 0.5200, 0.1329)

```
