NAG Library Routine Document

F08NTF (ZUNGHR)

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

F08NTF (ZUNGHR) generates the complex unitary matrix Q which was determined by F08NSF (ZGEHRD) when reducing a complex general matrix A to Hessenberg form.

2 Specification

```
SUBROUTINE F08NTF (N, ILO, IHI, A, LDA, TAU, WORK, LWORK, INFO)

INTEGER                 N, ILO, IHI, LDA, LWORK, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), TAU(*), WORK(max(1,LWORK))
```

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

3 Description

F08NTF (ZUNGHR) is intended to be used following a call to F08NSF (ZGEHRD), which reduces a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation: $A = QHQ^{\mathrm{H}}$. F08NSF (ZGEHRD) 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 F08NVF (ZGEBAL) 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 F08NSF (ZGEHRD).

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, $*$) – COMPLEX (KIND=nag_wp) 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 F08NSF (ZGEHRD).

On exit: the n by n unitary matrix Q.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NTF (ZUNGHR) is called.

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

6: 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 F08NSF (ZGEHRD).

7: WORK($\max(1, \mathrm{LWORK})$) – COMPLEX (KIND=nag_wp) array *Workspace*

On exit: if INFO $= 0$, the real part of 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 F08NTF (ZUNGHR) 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 $(\mathrm{IHI} - \mathrm{ILO}) \times nb$, where nb is the ***block size***.

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

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

F08NTF (ZUNGHR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08NTF (ZUNGHR) 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 $\frac{16}{3}q^3$, where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The real analogue of this routine is F08NFF (DORGHR).

10 Example

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

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix}.$$

Here A is general and must first be reduced to Hessenberg form by F08NSF (ZGEHRD). The program then calls F08NTF (ZUNGHR) to form Q, and passes this matrix to F08PSF (ZHSEQR) which computes the Schur factorization of A.

10.1 Program Text

```
    Program f08ntfe

!     F08NTF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ajf, x04dbf, zgehrd, zgemm, zhseqr,    &
                             zlange => f06uaf, zunghr
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: alpha, beta
      Real (Kind=nag_wp)               :: norm
      Integer                          :: i, ifail, info, lda, ldc, ldd, ldz,  &
                                          lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), c(:,:), d(:,:), tau(:),    &
                                            w(:), work(:), z(:,:)
      Real (Kind=nag_wp), Allocatable  :: rwork(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: cmplx
!     .. Executable Statements ..
      Write (nout,*) 'F08NTF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldc = n
      ldd = n
      ldz = n
      lwork = 64*(n-1)
      Allocate (a(lda,n),c(ldc,n),d(ldd,n),rwork(lda),tau(n),w(n),work(lwork), &
        z(ldz,n))

!     Read A from data file
```

```
      Read (nin,*)(a(i,1:n),i=1,n)

!     Store A in D
      d(1:ldd,1:n) = a(1:lda,1:n)

!     Print matrix A
!     ifail: behaviour on error exit
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.4','Matrix A', &
        'Integer',rlabs,'Integer',clabs,80,0,ifail)
      Write (nout,*)
      Flush (nout)

!     Reduce A to upper Hessenberg form H = (Q**H)*A*Q

!     The NAG name equivalent of zgehrd is f08nsf
      Call zgehrd(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 zunghr is f08ntf
      Call zunghr(n,1,n,z,ldz,tau,work,lwork,info)

!     Calculate the Schur factorization of H = Y*T*(Y**H) and form
!     Q*Y explicitly, storing the result in Z

!     Note that A = Z*T*(Z**H), where Z = Q*Y

!     The NAG name equivalent of zhseqr is f08psf
      Call zhseqr('Schur form','Vectors',n,1,n,a,lda,w,z,ldz,work,lwork,info)

!     Compute A - Z*T*Z^H from Schur factorization of A, and store in matrix D
!     The NAG name equivelent of zgemm is f06zaf
      alpha = cmplx(1,kind=nag_wp)
      beta = cmplx(0,kind=nag_wp)
      Call zgemm('N','N',n,n,n,alpha,z,ldz,a,lda,beta,c,ldc)
      alpha = cmplx(-1,kind=nag_wp)
      beta = cmplx(1,kind=nag_wp)
      Call zgemm('N','C',n,n,n,alpha,c,ldc,z,ldz,beta,d,ldd)

!     Find norm of matrix D and print warning if it is too large
!     f06uaf is the NAG name equivalent of the LAPACK auxiliary zlange
      norm = zlange('O',ldd,n,d,ldd,rwork)
      If (norm>x02ajf()**0.8_nag_wp) Then
        Write (nout,*) 'Norm of A-(Z*T*Z^H) is much greater than 0.'
        Write (nout,*) 'Schur factorization has failed.'
      Else
!       Print eigenvalues.
        Write (nout,*) 'Eigenvalues'
        Write (nout,99999)(w(i),i=1,n)
      End If

99999 Format ((3X,4(' (',F7.4,',',F7.4,')':)))

    End Program f08ntfe
```

10.2 Program Data

```
F08NTF Example Program Data
  4                                                          :Value of N
 (-3.97,-5.04) (-4.11, 3.70) (-0.34, 1.01) ( 1.29,-0.86)
 ( 0.34,-1.50) ( 1.52,-0.43) ( 1.88,-5.38) ( 3.36, 0.65)
 ( 3.31,-3.85) ( 2.50, 3.45) ( 0.88,-1.08) ( 0.64,-1.48)
 (-1.10, 0.82) ( 1.81,-1.59) ( 3.25, 1.33) ( 1.57,-3.44)   :End of matrix A
```

10.3 Program Results

```
F08NTF Example Program Results

Matrix A
                    1                   2                   3                   4
1  (-3.9700,-5.0400) (-4.1100, 3.7000) (-0.3400, 1.0100) ( 1.2900,-0.8600)
2  ( 0.3400,-1.5000) ( 1.5200,-0.4300) ( 1.8800,-5.3800) ( 3.3600, 0.6500)
3  ( 3.3100,-3.8500) ( 2.5000, 3.4500) ( 0.8800,-1.0800) ( 0.6400,-1.4800)
4  (-1.1000, 0.8200) ( 1.8100,-1.5900) ( 3.2500, 1.3300) ( 1.5700,-3.4400)

Eigenvalues
   (-6.0004,-6.9998) (-5.0000, 2.0060) ( 7.9982,-0.9964) ( 3.0023,-3.9998)
```