# NAG Fortran Library Routine Document

# F08NUF (ZUNMHR)

**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

F08NUF (ZUNMHR) multiplies an arbitrary complex matrix $C$ by the complex unitary matrix $Q$ which was determined by F08NSF (ZGEHRD) when reducing a complex general matrix to Hessenberg form.

## 2 Specification

```
SUBROUTINE F08NUF (SIDE, TRANS, M, N, ILO, IHI, A, LDA, TAU, C, LDC,
1                  WORK, LWORK, INFO)
INTEGER            M, N, ILO, IHI, LDA, LDC, LWORK, INFO
complex*16         A(LDA,*), TAU(*), C(LDC,*), WORK(*)
CHARACTER*1        SIDE, TRANS
```

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

## 3 Description

F08NUF (ZUNMHR) 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 $i_{\mathrm{lo}}$ and $i_{\mathrm{hi}}$ 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 form one of the matrix products

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

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

A common application of this routine is to transform a matrix $V$ of eigenvectors of $H$ to the matrix $QV$ 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 Parameters

1: SIDE – CHARACTER*1 *Input*

*On entry*: indicates how $Q$ or $Q^{\mathrm{H}}$ is to be applied to $C$.

SIDE = 'L'

$Q$ or $Q^{\mathrm{H}}$ is applied to $C$ from the left.

SIDE = 'R'

$Q$ or $Q^{\mathrm{H}}$ is applied to $C$ from the right.

*Constraint*: SIDE = 'L' or 'R'.

2: TRANS – CHARACTER*1 *Input*

*On entry*: indicates whether $Q$ or $Q^{\mathrm{H}}$ is to be applied to $C$.

TRANS = 'N'

$Q$ is applied to $C$.

TRANS = 'C'

$Q^{\mathrm{H}}$ is applied to $C$.

*Constraint*: TRANS = 'N' or 'C'.

3: 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*: $\mathrm{M} \geq 0$.

4: 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*: $\mathrm{N} \geq 0$.

5: ILO – INTEGER *Input*
6: IHI – INTEGER *Input*

*On entry*: these **must** be the same parameters ILO and IHI, respectively, as supplied to F08NSF (ZGEHRD).

*Constraints*:

if SIDE = 'L' and $\mathrm{M} > 0$, $1 \leq \mathrm{ILO} \leq \mathrm{IHI} \leq \mathrm{M}$;
if SIDE = 'L' and $\mathrm{M} = 0$, $\mathrm{ILO} = 1$ and $\mathrm{IHI} = 0$;
if SIDE = 'R' and $\mathrm{N} > 0$, $1 \leq \mathrm{ILO} \leq \mathrm{IHI} \leq \mathrm{N}$;
if SIDE = 'R' and $\mathrm{N} = 0$, $\mathrm{ILO} = 1$ and $\mathrm{IHI} = 0$.

7: A(LDA,$*$) – ***complex*16*** array *Input*

**Note**: the second dimension of the array A must be at least $\max(1, \mathrm{M})$ if SIDE = 'L' and at least $\max(1, \mathrm{N})$ if SIDE = 'R'.

*On entry*: details of the vectors which define the elementary reflectors, as returned by F08NSF (ZGEHRD).

8: LDA – INTEGER *Input*

*On entry*: the first dimension of the array A as declared in the (sub)program from which F08NUF (ZUNMHR) is called.

*Constraints*:

if SIDE = 'L', $\mathrm{LDA} \geq \max(1, \mathrm{M})$;
if SIDE = 'R', $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

9: TAU($*$) – ***complex*16*** array *Input*

**Note**: the dimension of the array TAU must be at least $\max(1, \mathrm{M} - 1)$ if SIDE = 'L' and at least $\max(1, \mathrm{N} - 1)$ if SIDE = 'R'.

*On entry*: further details of the elementary reflectors, as returned by F08NSF (ZGEHRD).

10: C(LDC,$*$) – ***complex*16*** array *Input/Output*

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

*On entry*: the $m$ by $n$ matrix $C$.

*On exit*: is overwritten by $QC$ or $Q^{\mathrm{H}}C$ or $CQ$ or $CQ^{\mathrm{H}}$ as specified by SIDE and TRANS.

11: LDC – INTEGER *Input*

*On entry*: the first dimension of the array C as declared in the (sub)program from which F08NUF (ZUNMHR) is called.

*Constraint*: LDC $\geq \max(1, \mathrm{M})$.

12: WORK$(*)$ – ***complex*16*** array *Workspace*

**Note**: the dimension of the array WORK must be at least $\max(1, \mathrm{LWORK})$.

*On exit*: if INFO $= 0$, the real part of WORK$(1)$ contains the minimum value of LWORK required for optimal performance.

13: LWORK – INTEGER *Input*

*On entry*: the dimension of the array WORK as declared in the (sub)program from which F08NUF (ZUNMHR) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

*Suggested value*: for optimal performance, LWORK $\geq \mathrm{N} \times nb$ if SIDE $=$ 'L' and at least $\mathrm{M} \times nb$ if SIDE $=$ 'R', where $nb$ is the optimal ***block size***.

*Constraints*:

if SIDE $=$ 'L', LWORK $\geq \max(1, \mathrm{N})$ or LWORK $= -1$;
if SIDE $=$ 'R', LWORK $\geq \max(1, \mathrm{M})$ or LWORK $= -1$.

14: 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 $= -i$, the $i$th parameter 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 $\epsilon$ is the ***machine precision***.

## 8 Further Comments

The total number of real floating-point operations is approximately $8nq^2$ if SIDE $=$ 'L' and $8mq^2$ if SIDE $=$ 'R', where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The real analogue of this routine is F08NGF (DORMHR).

# 9 Example

This example computes all the eigenvalues of the matrix $A$, where

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

and those eigenvectors which correspond to eigenvalues $\lambda$ such that $\mathrm{Re}(\lambda) < 0$. Here $A$ is general and must first be reduced to upper Hessenberg form $H$ by F08NSF (ZGEHRD). The program then calls F08PSF (ZHSEQR) to compute the eigenvalues, and F08PXF (ZHSEIN) to compute the required eigenvectors of $H$ by inverse iteration. Finally F08NUF (ZUNMHR) is called to transform the eigenvectors of $H$ back to eigenvectors of the original matrix $A$.

## 9.1 Program Text

```
*     F08NUF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDH, LDZ, LWORK, LDVL, LDVR
      PARAMETER        (NMAX=8,LDA=NMAX,LDH=NMAX,LDZ=1,LWORK=64*NMAX,
     +                 LDVL=NMAX,LDVR=NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION THRESH
      INTEGER          I, IFAIL, INFO, J, M, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), H(LDH,NMAX), TAU(NMAX),
     +                 VL(LDVL,NMAX), VR(LDVR,NMAX), W(NMAX),
     +                 WORK(LWORK), Z(LDZ,1)
      DOUBLE PRECISION RWORK(NMAX)
      INTEGER          IFAILL(NMAX), IFAILR(NMAX)
      LOGICAL          SELECT(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F06TFF, X04DBF, ZGEHRD, ZHSEIN, ZHSEQR, ZUNMHR
*     .. Intrinsic Functions ..
      INTRINSIC        DBLE, AIMAG
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08NUF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
         READ (NIN,*) THRESH
*
*        Reduce A to upper Hessenberg form H = (Q**H)*A*Q
*
         CALL ZGEHRD(N,1,N,A,LDA,TAU,WORK,LWORK,INFO)
*
*        Copy A to H
*
         CALL F06TFF('General',N,N,A,LDA,H,LDH)
*
*        Calculate the eigenvalues of H (same as A)
*
         CALL ZHSEQR('Eigenvalues','No vectors',N,1,N,H,LDH,W,Z,LDZ,
     +               WORK,LWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
```

```
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (' (',DBLE(W(I)),',',AIMAG(W(I)),')',I=1,
     +        N)
*
            DO 20 I = 1, N
               SELECT(I) = DBLE(W(I)) .LT. THRESH
   20       CONTINUE
*
*           Calculate the eigenvectors of H (as specified by SELECT),
*           storing the result in VR
*
            CALL ZHSEIN('Right','QR','No initial vectors',SELECT,N,A,
     +                  LDA,W,VL,LDVL,VR,LDVR,N,M,WORK,RWORK,IFAILL,
     +                  IFAILR,INFO)
*
*           Calculate the eigenvectors of A = Q * (eigenvectors of H)
*
            CALL ZUNMHR('Left','No transpose',N,M,1,N,A,LDA,TAU,VR,LDVR,
     +                  WORK,LWORK,INFO)
*
*           Print eigenvectors
*
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04DBF('General',' ',N,M,VR,LDVR,'Bracketed','F7.4',
     +                  'Contents of array VR','Integer',RLABS,
     +                  'Integer',CLABS,80,0,IFAIL)
*
         END IF
       END IF
       STOP
*
99999 FORMAT ((3X,4(A,F7.4,A,F7.4,A,:)))
      END
```

## 9.2 Program Data

```
F08NUF 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
  0.0                                                       :Value of THRESH
```

## 9.3 Program Results

```
 F08NUF Example Program Results

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

 Contents of array VR
                      1                  2
 1  ( 1.0000, 0.0000) ( 0.2613, 0.5284)
 2  (-0.0210, 0.3590) ( 0.6485, 0.4683)
 3  ( 0.1035, 0.3683) (-0.0323,-0.8516)
 4  (-0.0664,-0.3436) (-0.4521, 0.1368)
```