

NAG Library Routine Document

F08NSF (ZGEHRD)

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

F08NSF (ZGEHRD) reduces a complex general matrix to Hessenberg form.

2 Specification

```
SUBROUTINE F08NSF (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 *zgehrd*.

3 Description

F08NSF (ZGEHRD) reduces a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation: $A = QHQ^H$. H has real subdiagonal elements.

The matrix Q is not formed explicitly, but is represented as a product of elementary reflectors (see the F08 Chapter Introduction for details). Routines are provided to work with Q in this representation (see Section 9).

The routine can take advantage of a previous call to F08NVF (ZGEBAL), which may produce a matrix with the structure:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ & A_{22} & A_{23} \\ & & A_{33} \end{pmatrix}$$

where A_{11} and A_{33} are upper triangular. If so, only the central diagonal block A_{22} , in rows and columns i_{lo} to i_{hi} , needs to be reduced to Hessenberg form (the blocks A_{12} and A_{23} will also be affected by the reduction). Therefore the values of i_{lo} and i_{hi} determined by F08NVF (ZGEBAL) can be supplied to the routine directly. If F08NVF (ZGEBAL) has not previously been called however, then i_{lo} must be set to 1 and i_{hi} to n .

4 References

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

5 Arguments

- 1: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 2: ILO – INTEGER *Input*
- 3: IHI – INTEGER *Input*
On entry: if A has been output by F08NVF (ZGEBAL), then ILO and IHI **must** contain the values returned by that routine. Otherwise, ILO must be set to 1 and IHI to N.

Constraints:

if $N > 0$, $1 \leq \text{ILO} \leq \text{IHI} \leq N$;
 if $N = 0$, $\text{ILO} = 1$ and $\text{IHI} = 0$.

- 4: $A(\text{LDA}, *)$ – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n general matrix A .

On exit: A is overwritten by the upper Hessenberg matrix H and details of the unitary matrix Q . The subdiagonal elements of H are real.

- 5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08NSF (ZGEHRD) is called.

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

- 6: $\text{TAU}(*)$ – COMPLEX (KIND=nag_wp) array *Output*

Note: the dimension of the array TAU must be at least $\max(1, N - 1)$.

On exit: further details of the unitary matrix Q .

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

On exit: if $\text{INFO} = 0$, the real part of $\text{WORK}(1)$ 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 F08NSF (ZGEHRD) is called.

If $\text{LWORK} = -1$, 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, $\text{LWORK} \geq N \times nb$, where nb is the optimal **block size**.

Constraint: $\text{LWORK} \geq \max(1, N)$ or $\text{LWORK} = -1$.

- 9: INFO – INTEGER *Output*

On exit: $\text{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

$\text{INFO} < 0$

If $\text{INFO} = -i$, argument i had an illegal value.

If $\text{INFO} = -999$, dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

An explanatory message is output, and execution of the program is terminated.

7 Accuracy

The computed Hessenberg matrix H is exactly similar to a nearby matrix $(A + E)$, where

$$\|E\|_2 \leq c(n)\epsilon\|A\|_2,$$

$c(n)$ is a modestly increasing function of n , and ϵ is the *machine precision*.

The elements of H themselves may be sensitive to small perturbations in A or to rounding errors in the computation, but this does not affect the stability of the eigenvalues, eigenvectors or Schur factorization.

8 Parallelism and Performance

F08NSF (ZGEHRD) 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 $\frac{8}{3}q^2(2q + 3n)$, where $q = i_{hi} - i_{lo}$; if $i_{lo} = 1$ and $i_{hi} = n$, the number is approximately $\frac{40}{3}n^3$.

To form the unitary matrix Q F08NSF (ZGEHRD) may be followed by a call to F08NTF (ZUNGHR):

```
CALL ZUNGHR(N,ILO,IHI,A,LDA,TAU,WORK,LWORK,INFO)
```

To apply Q to an m by n complex matrix C F08NSF (ZGEHRD) may be followed by a call to F08NUF (ZUNMHR). For example,

```
CALL ZUNMHR('Left','No Transpose',M,N,ILO,IHI,A,LDA,TAU,C,LDC, &
            WORK,LWORK,INFO)
```

forms the matrix product QC .

The real analogue of this routine is F08NEF (DGEHRD).

10 Example

This example computes the upper Hessenberg form of the matrix A , where

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

10.1 Program Text

```
Program f08nsfe

!      F08NSF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgehrd
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Complex (Kind=nag_wp), Parameter :: zero = (0.0E0_nag_wp,0.0E0_nag_wp)
      Integer, Parameter :: nin = 5, nout = 6
!      .. Local Scalars ..
```

```

Integer                                :: i, ifail, info, lda, lwork, n
! .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: a(:, :), tau(:), work(:)
Character (1)                        :: clabs(1), rlabs(1)
! .. Executable Statements ..
Write (nout,*) 'F08NSF Example Program Results'
! Skip heading in data file
Read (nin,*)
Read (nin,*) n
lda = n
lwork = 64*n
Allocate (a(lda,n),tau(n-1),work(lwork))

! Read A from data file

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

! Reduce A to upper Hessenberg form

! The NAG name equivalent of zgehrd is f08nsf
Call zgehrd(n,1,n,a,lda,tau,work,lwork,info)

! Set the elements below the first subdiagonal to zero

Do i = 1, n - 2
  a(i+2:n,i) = zero
End Do

! Print upper Hessenberg form

Write (nout,*)
Flush (nout)

! 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',
  'Upper Hessenberg form','Integer',rlabs,'Integer',clabs,80,0,ifail) &

End Program f08nsfe

```

10.2 Program Data

F08NSF 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

F08NSF Example Program Results

Upper Hessenberg form

	1	2	3	4
1	(-3.9700,-5.0400)	(-1.1318,-2.5693)	(-4.6027,-0.1426)	(-1.4249, 1.7330)
2	(-5.4797, 0.0000)	(1.8585,-1.5502)	(4.4145,-0.7638)	(-0.4805,-1.1976)
3	(0.0000, 0.0000)	(6.2673, 0.0000)	(-0.4504,-0.0290)	(-1.3467, 1.6579)
4	(0.0000, 0.0000)	(0.0000, 0.0000)	(-3.5000, 0.0000)	(2.5619,-3.3708)
