NAG Toolbox for MATLAB

f08ns

1 Purpose

f08ns reduces a complex general matrix to Hessenberg form.

2 Syntax

```
[a, tau, info] = f08ns(ilo, ihi, a, 'n', n)
```

3 Description

f08ns reduces a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation: $A = QHQ^{\mathrm{H}}$. 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). Functions are provided to work with Q in this representation (see Section 8).

The function can take advantage of a previous call to f08nv, 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 f08nv can be supplied to the function directly. If f08nv 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 Parameters

5.1 Compulsory Input Parameters

1: **ilo – int32 scalar**

2: **ihi – int32 scalar**

If A has been output by f08nv, then **ilo** and **ihi must** contain the values returned by that function. Otherwise, **ilo** must be set to 1 and **ihi** to **n**.

Constraints:

if $\mathbf{n} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{n}$;
if $\mathbf{n} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$.

3: **a**(**lda**,$*$) **– complex array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

The n by n general matrix A.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a**.

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, work, lwork

5.4 Output Parameters

1: **a(lda,$*$) – complex array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

a contains the upper Hessenberg matrix H and details of the unitary matrix Q. The subdiagonal elements of H are real.

2: **tau($*$) – complex array**

Note: the dimension of the array **tau** must be at least $\max(1, \mathbf{n} - 1)$.

Further details of the unitary matrix Q.

3: **info – int32 scalar**

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **n**, 2: **ilo**, 3: **ihi**, 4: **a**, 5: **lda**, 6: **tau**, 7: **work**, 8: **lwork**, 9: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

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 Further Comments

The total number of real floating-point operations is approximately $\frac{8}{3}q^2(2q + 3n)$, where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$; if $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$, the number is approximately $\frac{40}{3}n^3$.

To form the unitary matrix Q f08ns may be followed by a call to f08nt:

```
[a, info] = f08nt(ilo, ihi, a, tau);
```

To apply Q to an m by n complex matrix C f08ns may be followed by a call to f08nu. For example,

```
[c, info] = f08nu('Left', 'No Transpose', ilo, ihi, a, tau, c);
```

forms the matrix product QC.

The real analogue of this function is f08ne.

9 Example

```
ilo = int32(1);
ihi = int32(4);
a = [complex(-3.97, -5.04), complex(-4.11, +3.7), complex(-0.34, +1.01),
complex(1.29, -0.86);
       complex(0.34, -1.5), complex(1.52, -0.43), complex(1.88, -5.38),
complex(3.36, +0.65);
       complex(3.31, -3.85), complex(2.5, +3.45), complex(0.88, -1.08),
complex(0.64, -1.48);
      complex(-1.1, +0.82), complex(1.81, -1.59), complex(3.25, +1.33),
complex(1.57, -3.44)];
[aOut, tau, info] = f08ns(ilo, ihi, a)
```

```
aOut =
   -3.9700 - 5.0400i   -1.1318 - 2.5693i   -4.6027 - 0.1426i   -1.4249 +
1.7330i
   -5.4797               1.8585 - 1.5502i    4.4145 - 0.7638i   -0.4805 -
1.1976i
    0.6932 - 0.4829i    6.2673              -0.4504 - 0.0290i   -1.3467 +
1.6579i
   -0.2113 + 0.0864i    0.1242 - 0.2289i   -3.5000               2.5619 -
3.3708i
tau =
   1.0620 - 0.2737i
   1.8059 + 0.3479i
   1.1818 + 0.9833i
info =
           0
```