NAG Toolbox for MATLAB

f08fs

1 Purpose

f08fs reduces a complex Hermitian matrix to tridiagonal form.

2 Syntax

```
[a, d, e, tau, info] = f08fs(uplo, a, 'n', n)
```

3 Description

f08fs reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$.

The matrix Q is not formed explicitly but is represented as a product of $n-1$ elementary reflectors (see the F08 Chapter Introduction for details). Functions are provided to work with Q in this representation (see Section 8).

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: **uplo – string**

Indicates whether the upper or lower triangular part of A is stored.

uplo = 'U'

The upper triangular part of A is stored.

uplo = 'L'

The lower triangular part of A is stored.

Constraint: **uplo** = 'U' or 'L'.

2: **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 Hermitian matrix A.

If **uplo** = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

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 tridiagonal matrix T and details of the unitary matrix Q as specified by **uplo**.

2: **d**($*$) – **double array**

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

The diagonal elements of the tridiagonal matrix T.

3: **e**($*$) – **double array**

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

The off-diagonal elements of the tridiagonal matrix T.

4: **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.

5: **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: **uplo**, 2: **n**, 3: **a**, 4: **lda**, 5: **d**, 6: **e**, 7: **tau**, 8: **work**, 9: **lwork**, 10: **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 tridiagonal matrix T 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 T 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 and eigenvectors.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{16}{3}n^3$.

To form the unitary matrix Q f08fs may be followed by a call to f08ft:

```
[a, info] = f08ft(uplo, a, tau);
```

To apply Q to an n by p complex matrix C f08fs may be followed by a call to f08fu. For example,

```
[c, info] = f08fu('Left', uplo, 'No Transpose', a, tau, c);
```

forms the matrix product QC.

The real analogue of this function is f08fe.

9 Example

```
uplo = 'L';
a = [complex(-2.28, +0), complex(0, 0), complex(0, 0), complex(0, 0);
     complex(1.78, +2.03), complex(-1.12, +0), complex(0, 0), complex(0,
0);
       complex(2.26, -0.1), complex(0.01, -0.43), complex(-0.37, +0),
complex(0, 0);
     complex(-0.12, -2.53), complex(-1.07, -0.86), complex(2.31, +0.92),
complex(-0.73, +0)];
[aOut, d, e, tau, info] = f08fs(uplo, a)
```

```
aOut =
  -2.2800                  0                  0                  0
  -4.3385            -0.1285                  0                  0
   0.3279 - 0.1251i  -2.0226            -0.1666                  0
  -0.1413 - 0.3666i  -0.3083 + 0.1763i  -1.8023            -1.9249
d =
   -2.2800
   -0.1285
   -0.1666
   -1.9249
e =
   -4.3385
   -2.0226
   -1.8023
tau =
   1.4103 + 0.4679i
   1.3024 + 0.7853i
   1.0940 - 0.9956i
info =
           0
```