

NAG Toolbox for MATLAB

f08hs

1 Purpose

f08hs reduces a complex Hermitian band matrix to tridiagonal form.

2 Syntax

```
[ab, d, e, q, info] = f08hs(vect, uplo, kd, ab, q, 'n', n)
```

3 Description

f08hs reduces a Hermitian band matrix A to real symmetric tridiagonal form T by a unitary similarity transformation:

$$T = Q^H A Q.$$

The unitary matrix Q is determined as a product of Givens rotation matrices, and may be formed explicitly by the function if required.

The function uses a vectorizable form of the reduction, due to Kaufman 1984.

4 References

Kaufman L 1984 Banded eigenvalue solvers on vector machines *ACM Trans. Math. Software* **10** 73–86

Parlett B N 1998 *The Symmetric Eigenvalue Problem* SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **vect** – string

Indicates whether Q is to be returned.

vect = 'V'

Q is returned.

vect = 'U'

Q is updated (and the array **q** must contain a matrix on entry).

vect = 'N'

Q is not required.

Constraint: **vect** = 'V', 'U' or 'N'.

2: **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'.

3: **kd** – int32 scalar

If **uplo** = 'U', the number of superdiagonals, k_d , of the matrix A .

If **uplo** = 'L', the number of subdiagonals, k_d , of the matrix A .

Constraint: **kd** ≥ 0 .

4: **ab(ldab,*)** – complex array

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

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

The upper or lower triangle of the n by n Hermitian band matrix A .

The matrix is stored in rows 1 to $k_d + 1$, more precisely,

if **uplo** = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in **ab**($k_d + 1 + i - j, j$) for $\max(1, j - k_d) \leq i \leq j$;

if **uplo** = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in **ab**($1 + i - j, j$) for $j \leq i \leq \min(n, j + k_d)$.

5: **q(ldq,*)** – complex array

The first dimension, **ldq**, of the array **q** must satisfy

if **vect** = 'V' or 'U', **ldq** $\geq \max(1, \mathbf{n})$;

if **vect** = 'N', **ldq** ≥ 1 .

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **vect** = 'V' or 'U' and at least 1 if **vect** = 'N'

If **vect** = 'U', **q** must contain the matrix formed in a previous stage of the reduction (for example, the reduction of a banded Hermitian-definite generalized eigenproblem); otherwise **q** need not be set.

5.2 Optional Input Parameters

1: **n** – int32 scalar

Default: The first dimension of the array **ab** and the second dimension of the array **ab**. (An error is raised if these dimensions are not equal.)

n , the order of the matrix A .

Constraint: **n** ≥ 0 .

5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldq, work

5.4 Output Parameters

1: **ab(ldab,*)** – complex array

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

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

ab contains values generated during the reduction to tridiagonal form.

The first superdiagonal and the diagonal of the tridiagonal matrix T are returned in **ab** using the same storage format as described above.

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: **q(ldq,*)** – **complex array**

The first dimension, **ldq**, of the array **q** must satisfy

if **vect** = 'V' or 'U', **ldq** $\geq \max(1, \mathbf{n})$;
if **vect** = 'N', **ldq** ≥ 1 .

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **vect** = 'V' or 'U' and at least 1 if **vect** = 'N'

If **vect** = 'V' or 'U', the n by n matrix Q .

If **vect** = 'N', **q** is not referenced.

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: **vect**, 2: **uplo**, 3: **n**, 4: **kd**, 5: **ab**, 6: **ldab**, 7: **d**, 8: **e**, 9: **q**, 10: **ldq**, 11: **work**, 12: **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.

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 Further Comments

The total number of real floating-point operations is approximately $20n^2k$ if **vect** = 'N' with $10n^3(k - 1)/k$ additional operations if **vect** = 'V'.

The real analogue of this function is f08he.

9 Example

```

vect = 'V';
uplo = 'L';
kd = int32(2);
ab = [complex(-3.13, +0), complex(-1.91, +0), complex(-2.87, +0),
      complex(0.5, +0);
      complex(1.94, +2.1), complex(-0.82, +0.89), complex(-2.1, +0.16),
      complex(0, 0);
      complex(-3.4, -0.25), complex(-0.67, -0.34), complex(0, 0),
      complex(0, 0)];
q = complex(zeros(4, 4));
[abOut, d, e, qOut, info] = f08hs(vect, uplo, kd, ab, q)

```

```

abOut =
    -3.1300         -1.2853         -0.9962         -1.9985
     4.4493         1.7002         2.5144             0
   -3.4000 - 0.2500i    0.6614 - 1.4067i          0          0
d =
    -3.1300
    -1.2853
    -0.9962
    -1.9985
e =
     4.4493
     1.7002
     2.5144
qOut =
    1.0000             0             0             0
             0      0.4360 + 0.4720i    0.1789 + 0.2537i   -0.6192 -
0.3275i
             0      -0.7642 - 0.0562i    0.2549 + 0.0530i   -0.5669 +
0.1539i
             0             0      0.8717 - 0.2757i    0.2891 -
0.2839i
info =
             0

```