NAG Toolbox for MATLAB

f08ks

1 Purpose

f08ks reduces a complex m by n matrix to bidiagonal form.

2 Syntax

```
[a, d, e, tauq, taup, info] = f08ks(a, 'm', m, 'n', n)
```

3 Description

f08ks reduces a complex m by n matrix A to real bidiagonal form B by a unitary transformation: $A = QBP^{\mathrm{H}}$, where Q and P^{H} are unitary matrices of order m and n respectively.

If $m \geq n$, the reduction is given by:

$$A = Q\begin{pmatrix} B_1 \\ 0 \end{pmatrix}P^{\mathrm{H}} = Q_1 B_1 P^{\mathrm{H}},$$

where B_1 is a real n by n upper bidiagonal matrix and Q_1 consists of the first n columns of Q.

If $m < n$, the reduction is given by

$$A = Q(\, B_1 \quad 0\,)P^{\mathrm{H}} = QB_1P_1^{\mathrm{H}},$$

where B_1 is a real m by m lower bidiagonal matrix and P_1^{H} consists of the first m rows of P^{H}.

The unitary matrices Q and P are not formed explicitly but are represented as products of elementary reflectors (see the F08 Chapter Introduction for details). Functions are provided to work with Q and P 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: **a(lda,$*$) – complex array**

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

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

The m by n matrix A.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the number of rows of the matrix A.

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

2: **n – int32 scalar**

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

n, the number of columns 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 $\max(1, \mathbf{m})$

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

If $m \geq n$, the diagonal and first superdiagonal are overwritten by the upper bidiagonal matrix B, elements below the diagonal are overwritten by details of the unitary matrix Q and elements above the first superdiagonal are overwritten by details of the unitary matrix P.

If $m < n$, the diagonal and first subdiagonal are overwritten by the lower bidiagonal matrix B, elements below the first subdiagonal are overwritten by details of the unitary matrix Q and elements above the diagonal are overwritten by details of the unitary matrix P.

2: **d($*$) – double array**

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

The diagonal elements of the bidiagonal matrix B.

3: **e($*$) – double array**

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

The off-diagonal elements of the bidiagonal matrix B.

4: **tauq($*$) – complex array**

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

Further details of the unitary matrix Q.

5: **taup($*$) – complex array**

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

Further details of the unitary matrix P.

6: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

$\mathbf{info} = -i$

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

1: **m**, 2: **n**, 3: **a**, 4: **lda**, 5: **d**, 6: **e**, 7: **tauq**, 8: **taup**, 9: **work**, 10: **lwork**, 11: **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 bidiagonal form B satisfies $QBP^{\mathrm{H}} = A + E$, 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 B 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 singular values and vectors.

8 Further Comments

The total number of real floating-point operations is approximately $16n^2(3m - n)/3$ if $m \geq n$ or $16m^2(3n - m)/3$ if $m < n$.

If $m \gg n$, it can be more efficient to first call f08as to perform a QR factorization of A, and then to call f08ks to reduce the factor R to bidiagonal form. This requires approximately $8n^2(m + n)$ floating-point operations.

If $m \ll n$, it can be more efficient to first call f08av to perform an LQ factorization of A, and then to call f08ks to reduce the factor L to bidiagonal form. This requires approximately $8m^2(m + n)$ operations.

To form the unitary matrices P^{H} and/or Q f08ks may be followed by calls to f08kt:

to form the m by m unitary matrix Q

```
[a, info] = f08kt('Q', n, a, tauq);
```

but note that the second dimension of the array **a** must be at least **m**, which may be larger than was required by f08ks;

to form the n by n unitary matrix P^{H}

```
[a, info] = f08kt('P', m, a, taup);
```

but note that the first dimension of the array **a**, specified by the parameter **lda**, must be at least **n**, which may be larger than was required by f08ks.

To apply Q or P to a complex rectangular matrix C, f08ks may be followed by a call to f08ku.

The real analogue of this function is f08ke.

9 Example

```
a = [complex(0.96, -0.8100000000000001), complex(-0.03, +0.96), complex(-
0.91, +2.06), complex(-0.05, +0.41);
      complex(-0.98, +1.98), complex(-1.2, +0.19), complex(-0.66, +0.42),
...
     complex(-0.8100000000000001, +0.5600000000000001);
       complex(0.62, -0.46), complex(1.01, +0.02), complex(0.63, -0.17),
complex(-1.11, +0.6);
      complex(-0.37, +0.38), complex(0.19, -0.54), complex(-0.98, -0.36),
complex(0.22, -0.2);
       complex(0.83, +0.51), complex(0.2, +0.01), complex(-0.17, -0.46),
complex(1.47, +1.59);
             complex(1.08,   -0.28),   complex(0.2,   -0.12),   complex(-
0.07000000000000001, +1.23), complex(0.26, +0.26)];
[aOut, d, e, tauq, taup, info] = f08ks(a)
```

```
aOut =
```

```
  -3.0870               2.1126              0.0543 + 0.4543i   0.3757 +
0.1070i
  -0.3270 + 0.4238i   2.0660              1.2628              0.0283 +
0.1650i
   0.1692 - 0.0798i  -0.2585 - 0.0137i   1.8731            -1.6126
  -0.1060 + 0.0727i   0.0582 + 0.0068i  -0.3219 + 0.3404i   2.0022
    0.1729 + 0.1606i    0.0884 - 0.1430i   -0.4052 - 0.2475i    0.2871 +
0.1826i
    0.2699 - 0.0152i   -0.0551 - 0.1065i    0.2172 + 0.2910i    0.5596 -
0.0569i
d =
   -3.0870
    2.0660
    1.8731
    2.0022
e =
    2.1126
    1.2628
   -1.6126
tauq =
   1.3110 - 0.2624i
   1.7965 - 0.0234i
   1.2420 - 0.1807i
   1.0144 + 0.6225i
taup =
   1.2312 - 0.5404i
   1.2623 - 0.9286i
   1.7829 - 0.6221i
        0
info =
           0
```