NAG Toolbox for MATLAB

f08ls

1 Purpose

f08ls reduces a complex m by n band matrix to real upper bidiagonal form.

2 Syntax

```
[ab, d, e, q, pt, c, info] = f08ls(vect, m, kl, ku, ab, c, 'n', n,
'ncc', ncc)
```

3 Description

f08ls reduces a complex m by n band matrix to real upper bidiagonal form B by a unitary transformation: $A = QBP^{\mathrm{H}}$. The unitary matrices Q and P^{H}, of order m and n respectively, are determined as a product of Givens rotation matrices, and may be formed explicitly by the function if required. A matrix C may also be updated to give $\tilde{C} = Q^{\mathrm{H}}C$.

The function uses a vectorizable form of the reduction.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **vect – string**

Indicates whether the matrices Q and/or P^{H} are generated.

vect = 'N'

Neither Q nor P^{H} is generated.

vect = 'Q'

Q is generated.

vect = 'P'

P^{H} is generated.

vect = 'B'

Both Q and P^{H} are generated.

Constraint: **vect** = 'N', 'Q', 'P' or 'B'.

2: **m – int32 scalar**

m, the number of rows of the matrix A.

Constraint: **m** ≥ 0.

3: **kl – int32 scalar**

The number of subdiagonals, k_l, within the band of A.

Constraint: **kl** ≥ 0.

4: **ku – int32 scalar**

The number of superdiagonals, k_u, within the band of A.

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

5: **ab(ldab,$*$) – complex array**

The first dimension of the array **ab** must be at least $\mathbf{kl} + \mathbf{ku} + 1$

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

The original m by n band matrix A.

The matrix is stored in rows 1 to $+ + 1$, more precisely, the element A_{ij} must be stored in

$$\mathbf{ab}(+1 + i - j, j) \qquad \text{for } \max(1j-) \leq i \leq \min(mj+).$$

6: **c(ldc,$*$) – complex array**

The first dimension, **ldc**, of the array **c** must satisfy

if $\mathbf{ncc} > 0$, $\mathbf{ldc} \geq \max(1, \mathbf{m})$;
if $\mathbf{ncc} = 0$, $\mathbf{ldc} \geq 1$.

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

An m by n_C matrix C.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the number of columns of the matrix A.

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

2: **ncc – int32 scalar**

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

n_C, the number of columns of the matrix C.

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

5.3 Input Parameters Omitted from the MATLAB Interface

ldab, ldq, ldpt, ldc, work, rwork

5.4 Output Parameters

1: **ab(ldab,$*$) – complex array**

The first dimension of the array **ab** must be at least $\mathbf{kl} + \mathbf{ku} + 1$

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

ab contains values generated during the reduction.

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 superdiagonal elements of the bidiagonal matrix B.

4: **q**(**ldq**,$*$) – **complex array**

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

if **vect** = 'Q' or 'B', $\mathbf{ldq} \geq \max(1, \mathbf{m})$;
$\mathbf{ldq} \geq 1$ otherwise.

The second dimension of the array must be at least $\max(1, \mathbf{m})$ if **vect** = 'Q' or 'B', and at least 1 otherwise

If **vect** = 'Q' or 'B', contains the m by m unitary matrix Q.

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

5: **pt**(**ldpt**,$*$) – **complex array**

The first dimension, **ldpt**, of the array **pt** must satisfy

if **vect** = 'P' or 'B', $\mathbf{ldpt} \geq \max(1, \mathbf{n})$;
$\mathbf{ldpt} \geq 1$ otherwise.

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

The n by n unitary matrix P^{H}, if **vect** = 'P' or 'B'. If **vect** = 'N' or 'Q', **pt** is not referenced.

6: **c**(**ldc**,$*$) – **complex array**

The first dimension, **ldc**, of the array **c** must satisfy

if $\mathbf{ncc} > 0$, $\mathbf{ldc} \geq \max(1, \mathbf{m})$;
if $\mathbf{ncc} = 0$, $\mathbf{ldc} \geq 1$.

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

c contains $Q^{\mathrm{H}}C$. If $\mathbf{ncc} = 0$, **c** is not referenced.

7: **info** – **int32 scalar**

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

6 Error Indicators and Warnings

$\mathbf{info} = -i$

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

1: **vect**, 2: **m**, 3: **n**, 4: **ncc**, 5: **kl**, 6: **ku**, 7: **ab**, 8: **ldab**, 9: **d**, 10: **e**, 11: **q**, 12: **ldq**, 13: **pt**, 14: **ldpt**, 15: **c**, 16: **ldc**, 17: **work**, 18: **rwork**, 19: **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.

The computed matrix Q differs from an exactly unitary matrix by a matrix F such that

$$\|F\|_2 = O(\epsilon).$$

A similar statement holds for the computed matrix P^{H}.

8 Further Comments

The total number of real floating-point operations is approximately the sum of:

$20n^2k$, if **vect** = 'N' and **ncc** $= 0$, and

$10n^2n_C(k-1)/k$, if C is updated, and

$10n^3(k-1)/k$ if either Q or P^{H} is generated (double this if both),

where $k = k_l + k_u$, assuming $n \gg k$. For this section we assume that $m = n$.

The real analogue of this function is f08le.

9 Example

```
vect = 'N';
m = int32(6);
kl = int32(2);
ku = int32(1);
ab  =  [complex(0,  +0),  complex(-0.03,  +0.96),  complex(-0.66,  +0.42),
complex(-1.11, +0.6);
       complex(0.96,  -0.81), complex(-1.2,  +0.19), complex(0.63,  -0.17),
complex(0.22, -0.2);
      complex(-0.98, +1.98), complex(1.01, +0.02), complex(-0.98, -0.36),
complex(1.47, +1.59);
       complex(0.62,  -0.46), complex(0.19, -0.54), complex(-0.17, -0.46),
complex(0.26, +0.26)];
c = [];
[abOut, d, e, q, pt, cOut, info] = f08ls(vect, m, kl, ku, ab, c)
```

```
abOut =
          0                1.4909 + 0.8236i   -0.1892 + 1.2659i    0.0524 +
0.1370i
    2.0300 - 1.7128i   -1.0428 + 1.4054i    1.5055 - 1.4070i   -0.2913 -
0.8154i
   -1.0381 + 2.0974i    1.4552 + 0.1747i   -1.3984 + 1.0897i    0.0392 +
0.4466i
    0.6200 - 0.4600i   -0.1954 - 1.1588i   -0.3272 - 1.6175i    0.0224 +
0.0483i
d =
    2.6560
    1.7501
    2.0607
    0.8658
e =
    1.7033
    1.2800
    0.1467
q =
  5.5851e-209 - 1.3684e-54i
pt =
  4.2440e-314 - 3.9257e-54i
cOut =
info =
           0
```