NAG Toolbox for MATLAB

f08au

1 Purpose

f08au multiplies an arbitrary complex matrix C by the complex unitary matrix Q from a QR factorization computed by f08as or f08bs.

2 Syntax

```
[c, info] = f08au(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

f08au is intended to be used after a call to f08as or f08bs, which perform a QR factorization of a complex matrix A. The unitary matrix Q is represented as a product of elementary reflectors.

This function may be used to form one of the matrix products

$$QC, Q^{\mathrm{H}}C, CQ \text{ or } CQ^{\mathrm{H}},$$

overwriting the result on **c** (which may be any complex rectangular matrix).

A common application of this function is in solving linear least-squares problems, as described in the F08 Chapter Introduction and illustrated in Section 9 of the document for f08as.

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: **side – string**

Indicates how Q or Q^{H} is to be applied to C.

side = 'L'

Q or Q^{H} is applied to C from the left.

side = 'R'

Q or Q^{H} is applied to C from the right.

Constraint: **side** = 'L' or 'R'.

2: **trans – string**

Indicates whether Q or Q^{H} is to be applied to C.

trans = 'N'

Q is applied to C.

trans = 'C'

Q^{H} is applied to C.

Constraint: **trans** = 'N' or 'C'.

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

The first dimension, **lda**, of the array **a** must satisfy

if **side** = 'L', **lda** $\geq \max(1, \mathbf{m})$;
if **side** = 'R', **lda** $\geq \max(1, \mathbf{n})$.

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

Details of the vectors which define the elementary reflectors, as returned by f08as or f08bs.

4: **tau**(∗) – **complex array**

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

Further details of the elementary reflectors as returned by f08as or f08bs.

5: **c**(**ldc**,∗) – **complex array**

The first dimension of the array **c** 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 C.

5.2 Optional Input Parameters

1: **m** – **int32 scalar**

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

m, the number of rows of the matrix C.

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

2: **n** – **int32 scalar**

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

n, the number of columns of the matrix C.

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

3: **k** – **int32 scalar**

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

k, the number of elementary reflectors whose product defines the matrix Q.

Constraints:

if **side** = 'L', $\mathbf{m} \geq \mathbf{k} \geq 0$;
if **side** = 'R', $\mathbf{n} \geq \mathbf{k} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldc, work, lwork

5.4 Output Parameters

1: **c**(**ldc**,∗) – **complex array**

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

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

c contains QC or $Q^{\mathrm{H}}C$ or CQ or CQ^{H} as specified by **side** and **trans**.

2: **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: **side**, 2: **trans**, 3: **m**, 4: **n**, 5: **k**, 6: **a**, 7: **lda**, 8: **tau**, 9: **c**, 10: **ldc**, 11: **work**, 12: **lwork**, 13: **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 result differs from the exact result by a matrix E such that

$$\|E\|_2 = O(\epsilon)\|C\|_2,$$

where ϵ is the ***machine precision***.

8 Further Comments

The total number of real floating-point operations is approximately $8nk(2m-k)$ if **side** = 'L' and $8mk(2n-k)$ if **side** = 'R'.

The real analogue of this function is f08ag.

9 Example

```
side = 'Left';
trans = 'Conjugate transpose';
a  =  [complex(-3.087005021051958,  +0),  complex(-0.4884993674179767,  -
1.141689105124614), ...
           complex(0.3773560431732106,   -1.243729755480491),   complex(-
0.8551654376967619, -0.7073198731811355);
                    complex(-0.326978431123342,    +0.4238066080640211),
complex(1.516316047290931, ...
       +0),  complex(1.373055096626977,  -0.8176293354211591),  complex(-
0.2508627202628476, +0.8203486040451443);
          complex(0.1691724764304651,  -0.07980476733072399),  complex(-
0.4537104861498296, ...
      -0.006491499591352979),  complex(-2.17134536255717,  +0),  complex(-
0.2272676203283603, -0.2957314059070643);
          complex(-0.1059736295130279,  +0.0726861860966964),  complex(-
0.2734071741396468, ...
                    +0.0978078838704349),     complex(-0.291822737804996,
+0.4888081441553061), complex(-2.353376106555421, +0);
           complex(0.1729396325459321,  +0.1606326404292985),  complex(-
0.3236304714632618, ...
                    +0.1230007002199739),     complex(0.2727685061644792,
+0.04697693306903757), ...
    complex(0.7054226886031557, +0.2515080566109891);
          complex(0.2698996744687472,  -0.01516708364852971),  complex(-
0.1645935439354584, ...
                    +0.3389007203482612),     complex(0.5348395253617789,
+0.3988290677840221), ...
    complex(0.2703069905230761, -0.07268783264065712)];
tau = [complex(1.310981029656006, -0.2623902437722555);
```

```
     complex(1.105103989110574, -0.450362538745018);
     complex(1.040251871615519, +0.2121758107261096);
     complex(1.18595901116611, +0.2011836003307436)];
c = [complex(-2.09, +1.93), complex(3.26, -2.7);
     complex(3.34, -3.53), complex(-6.22, +1.16);
     complex(-4.94, -2.04), complex(7.94, -3.13);
     complex(0.17, +4.23), complex(1.04, -4.26);
     complex(-5.19, +3.63), complex(-2.31, -2.12);
     complex(0.98, +2.53), complex(-1.39, -4.05)];
[cOut, info] = f08au(side, trans, a, tau, c)
```

```
cOut =
   5.3510 - 0.1638i  -4.7626 - 2.8427i
  -5.7559 - 0.2004i   6.3325 - 2.7406i
  -2.5366 + 4.0215i   6.4835 - 3.8629i
  -1.0677 - 6.3316i   6.4968 - 0.7809i
  -0.0381 - 0.0273i  -0.1320 - 0.0612i
  -0.0144 + 0.0483i   0.0906 - 0.0740i
info =
           0
```