NAG Toolbox for MATLAB

f08at

1 Purpose

f08at generates all or part of the complex unitary matrix Q from a QR factorization computed by f08as or f08bs.

2 Syntax

```
[a, info] = f08at(a, tau, 'm', m, 'n', n, 'k', k)
```

3 Description

f08at 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 generate Q explicitly as a square matrix, or to form only its leading columns.

Usually Q is determined from the QR factorization of an m by p matrix A with $m \geq p$. The whole of Q may be computed by:

```
[a, info] = f08at(a, tau);
```

(note that the array **a** must have m columns) or its leading p columns by:

```
[a, info] = f08at(a(1:p,:), tau);
```

The columns of Q returned by the last call form an orthonormal basis for the space spanned by the columns of A; thus f08as followed by f08at can be used to orthogonalise the columns of A.

The information returned by the QR factorization functions also yields the QR factorization of the leading k columns of A, where $k < p$. The unitary matrix arising from this factorization can be computed by:

```
[a, info] = f08at(a, tau);
```

or its leading k columns by:

```
[a, info] = f08at(a(:,1:k), tau);
```

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 $\max(1, \mathbf{n})$

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

2: **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.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the order of the unitary matrix Q.

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 Q.

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

3: **k – int32 scalar**

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

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

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

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 $\max(1, \mathbf{n})$

The m by n matrix Q.

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: **m**, 2: **n**, 3: **k**, 4: **a**, 5: **lda**, 6: **tau**, 7: **work**, 8: **lwork**, 9: **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 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 $16mnk - 8(m+n)k^2 + \frac{16}{3}k^3$; when $n = k$, the number is approximately $\frac{8}{3}n^2(3m - n)$.

The real analogue of this function is f08af.

9 Example

```
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)];
[aOut, info] = f08at(a, tau)
```

```
aOut =
   -0.3110 +  0.2624i   -0.3175 +  0.4835i    0.4966 -  0.2997i   -0.0072 -
0.3718i
    0.3175 -  0.6414i   -0.2062 +  0.1577i   -0.0793 -  0.3094i   -0.0282 -
0.1491i
   -0.2008 +  0.1490i    0.4892 -  0.0900i    0.0357 -  0.0219i    0.5625 -
0.0710i
    0.1199 -  0.1231i    0.2566 -  0.3055i    0.4489 -  0.2141i   -0.1651 +
0.1800i
   -0.2689 -  0.1652i    0.1697 -  0.2491i   -0.0496 +  0.1158i   -0.4885 -
0.4540i
   -0.3499 +  0.0907i   -0.0491 -  0.3133i   -0.1256 -  0.5300i    0.1039 +
0.0450i
info =
              0
```