NAG Toolbox for MATLAB

f08cw

1 Purpose

f08cw generates all or part of the complex n by n unitary matrix Q from an RQ factorization computed by f08cv.

2 Syntax

```
[a, info] = f08cw(a, tau, 'm', m, 'n', n, 'k', k)
```

3 Description

f08cw is intended to be used following a call to f08cv, which performs an RQ factorization of a complex matrix A and represents the unitary matrix Q as a product of k elementary reflectors of order n.

This function may be used to generate Q explicitly as a square matrix, or to form only its trailing rows.

Usually Q is determined from the RQ factorization of a p by n matrix A with $p \leq n$. The whole of Q may be computed by:

```
[a, info] = f08cw(a, tau);
```

(note that the array **a** must have at least n rows), or its trailing p rows as:

```
[a, info] = f08cw(a(1:p,:), tau, 'k', p);
```

The rows of Q returned by the last call form an orthonormal basis for the space spanned by the rows of A; thus f08cv followed by f08cw can be used to orthogonalise the rows of A.

The information returned by f08cv also yields the RQ factorization of the trailing k rows of A, where $k < p$. The unitary matrix arising from this factorization can be computed by:

```
[a, info] = f08cw(a, tau, 'k', k);
```

or its leading k columns by:

```
[a, info] = f08cw(a(1:k,:), tau, 'k', k);
```

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D 1999 *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: http://www.netlib.org/lapack/lug

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 f08cv.

2: **tau**$(*)$ **– complex array**

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

tau(i) must contain the scalar factor of the elementary reflector H_i, as returned by f08cv.

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

3: **k – int32 scalar**

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

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

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

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

$$\|E\|_2 = O\epsilon$$

and ϵ is the ***machine precision***.

8 Further Comments

The total number of floating point operations is approximately $16mnk - 8(m+n)k^2 + \frac{16}{3}k^3$; when $m = k$ this becomes $\frac{8}{3}m^2(3n-m)$.

The real analogue of this function is f08cj.

9 Example

```
a    =    [complex(-0.1732849729587456,    -0.353512300439208),    complex(-
0.2462313258920653, ...
       +0.2360567043400598),  complex(-1.835343413987657,  +0),  complex(-
1.000908232487384, ...
     +1.32039822375845), complex(0.4156798554945925, +1.306842647249796),
...
     complex(-0.9547699266414058, +0.7897042065311224);
           complex(0.09114631819652763,  +0.2475043484243413),  complex(-
0.4802181323902914, ...
                 -0.1420168005324323),    complex(0.4821546631632406,    -
0.09601786537269767), ...
         complex(-1.575493318887248,  +0),  complex(-1.101370732269467,  -
0.05966135311459431), ...
     complex(-0.3055741168299267, +0.2217392484654139);
           complex(-0.3728300824744412,  +0.4958059927232391),  complex(-
0.1765491709076989, ...
    +0.1573286421732319), complex(0.21466822640392, +0.1362481009678133),
...
            complex(-0.2486033881873816,   -0.20517929204053),   complex(-
2.991745620440206, ...
     +0), complex(-0.01276691481133434, +0.08423270423682098);
          complex(-0.004843574322500398,  +0.1359553259506703),  complex(-
0.2500132710791688, ...
                    +0.2066154899636792),     complex(-0.3473326302453916,
+0.2282035599626739), ...
                    complex(0.06655374884663712,     -0.0718369909305264),
complex(0.5272403281091034, ...
     +0.480309771000454), complex(-2.764959312539698, +0)];
tau = [complex(1.56428134034193, +0.1176469112365368);
     complex(1.251296240416963, +0.1907664156695332);
     complex(1.137210468449182, +0.3464883871917274);
     complex(1.094033933454587, -0.09403393345458752)];
[aOut, info] = f08cw(a, tau)
```

```
aOut =
  Columns 1 through 4
    0.2810 + 0.5020i    0.2707 - 0.3296i   -0.2864 - 0.0094i    0.2262 -
0.3854i
   -0.2051 - 0.1092i    0.5711 + 0.0432i   -0.5416 + 0.0454i   -0.3387 +
0.2228i
    0.3083 - 0.6874i    0.2251 - 0.1313i   -0.2062 + 0.0691i    0.3259 +
0.1178i
    0.0181 - 0.1483i    0.2930 - 0.2025i    0.4015 - 0.2170i   -0.0796 +
0.0723i
  Columns 5 through 6
   0.0341 - 0.0760i  -0.3936 - 0.2083i
   0.0098 - 0.0712i  -0.1296 + 0.3691i
   0.0753 + 0.1412i   0.0264 - 0.4134i
```

```
  -0.5317 - 0.5751i  -0.0940 - 0.0940i
info =
           0
```