# NAG Toolbox for MATLAB

# f08fu

## 1 Purpose

f08fu multiplies an arbitrary complex matrix $C$ by the complex unitary matrix $Q$ which was determined by f08fs when reducing a complex Hermitian matrix to tridiagonal form.

## 2 Syntax

```
[c, info] = f08fu(side, uplo, trans, a, tau, c, 'm', m, 'n', n)
```

## 3 Description

f08fu is intended to be used after a call to f08fs, which reduces a complex Hermitian matrix $A$ to real symmetric tridiagonal form $T$ by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. f08fs represents the unitary matrix $Q$ 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 to transform a matrix $Z$ of eigenvectors of $T$ to the matrix $QZ$ of eigenvectors of $A$.

## 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 $Q^{\mathrm{H}}$ is to be applied to $C$.

**side** = 'L'

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

**side** = 'R'

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

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

2: **uplo – string**

This **must** be the same parameter **uplo** as supplied to f08fs.

*Constraint*: **uplo** = 'U' or 'L'.

3: **trans – string**

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

**trans** = 'N'

$Q$ is applied to $C$.

**trans** = 'C'

$Q^{\mathrm{H}}$ is applied to $C$.

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

4: **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 $\max(1, \mathbf{m})$ if **side** = 'L' and at least $\max(1, \mathbf{n})$ if **side** = 'R'

Details of the vectors which define the elementary reflectors, as returned by f08fs.

5: **tau**(∗) – **complex array**

**Note**: the dimension of the array **tau** must be at least $\max(1, \mathbf{m} - 1)$ if **side** = 'L' and at least $\max(1, \mathbf{n} - 1)$ if **side** = 'R'.

Further details of the elementary reflectors, as returned by f08fs.

6: **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 $C$; $m$ is also the order of $Q$ if **side** = 'L'.

*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 $C$; $n$ is also the order of $Q$ if **side** = 'R'.

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

## 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 $CQ^{\mathrm{H}}$ 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: **uplo**, 3: **trans**, 4: **m**, 5: **n**, 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 $\epsilon$ is the ***machine precision***.

## 8 Further Comments

The total number of real floating-point operations is approximately $8m^2n$ if **side** = 'L' and $8mn^2$ if **side** = 'R'.

The real analogue of this function is f08fg.

## 9 Example

```
side = 'Left';
uplo = 'L';
trans = 'No transpose';
a = [complex(-2.28, +0), complex(0, 0), complex(0, 0), complex(0, 0);
     complex(1.78, +2.03), complex(-1.12, +0), complex(0, 0), complex(0,
0);
        complex(2.26, -0.1), complex(0.01, -0.43), complex(-0.37, +0),
complex(0, 0);
     complex(-0.12, -2.53), complex(-1.07, -0.86), complex(2.31, +0.92),
complex(-0.73, +0)];
range = 'I';
order = 'B';
vl = 0;
vu = 0;
il = int32(1);
iu = int32(2);
abstol = 0;
[a, d, e, tau, info] = f08fs(uplo, a);
[m, nsplit, w, iblock, isplit, info] = ...
    f08jj(range, order, vl, vu, il, iu, abstol, d, e);
[c, ifailv, info] = f08jx(d, e, m, w, iblock, isplit);
[cOut, info] = f08fu(side, uplo, trans, a, tau, c)
```

```
cOut =
   0.7299            -0.2595
  -0.1663 - 0.2061i   0.5969 + 0.4214i
  -0.4165 - 0.1417i  -0.2965 - 0.1507i
```

```
   0.1743 + 0.4162i   0.3482 + 0.4085i
info =
           0
```