

NAG Toolbox for MATLAB

f07ms

1 Purpose

f07ms solves a complex Hermitian indefinite system of linear equations with multiple right-hand sides,

$$AX = B,$$

where A has been factorized by f07mr.

2 Syntax

```
[b, info] = f07ms(uplo, a, ipiv, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07ms is used to solve a complex Hermitian indefinite system of linear equations $AX = B$, this function must be preceded by a call to f07mr which computes the Bunch–Kaufman factorization of A .

If **uplo** = 'U', $A = PUDU^H P^T$, where P is a permutation matrix, U is an upper triangular matrix and D is an Hermitian block diagonal matrix with 1 by 1 and 2 by 2 blocks; the solution X is computed by solving $PUDY = B$ and then $U^H P^T X = Y$.

If **uplo** = 'L', $A = PLDL^H P^T$, where L is a lower triangular matrix; the solution X is computed by solving $PLDY = B$ and then $L^H P^T X = Y$.

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: **uplo** – string

Indicates how A has been factorized.

uplo = 'U'

$A = PUDU^H P^T$, where U is upper triangular.

uplo = 'L'

$A = PLDL^H P^T$, where L is lower triangular.

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

2: **a(lda,*)** – complex array

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

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

Details of the factorization of A , as returned by f07mr.

3: **ipiv**(*) – **int32** array

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

Details of the interchanges and the block structure of D , as returned by f07mr.

4: **b**(ldb,*) – **complex** array

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

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

The n by r right-hand side matrix B .

5.2 Optional Input Parameters

1: **n** – **int32** scalar

Default: The second dimension of the array **a** The dimension of the array **ipiv**.
 n , the order of the matrix A .

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

2: **nrhs_p** – **int32** scalar

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

r , the number of right-hand sides.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb

5.4 Output Parameters

1: **b**(ldb,*) – **complex** array

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

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

The n by r solution matrix X .

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: **uplo**, 2: **n**, 3: **nrhs_p**, 4: **a**, 5: **lda**, 6: **ipiv**, 7: **b**, 8: **ldb**, 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

For each right-hand side vector b , the computed solution x is the exact solution of a perturbed system of equations $(A + E)x = b$, where

$$\begin{aligned} \text{if } \mathbf{uplo} = 'U', & \quad |E| \leq c(n)\epsilon P|U||D||U^H|P^T; \\ \text{if } \mathbf{uplo} = 'L', & \quad |E| \leq c(n)\epsilon P|L||D||L^H|P^T, \end{aligned}$$

$c(n)$ is a modest linear function of n , and ϵ is the *machine precision*.

If \hat{x} is the true solution, then the computed solution x satisfies a forward error bound of the form

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq c(n) \text{cond}(A, x) \epsilon$$

where $\text{cond}(A, x) = \| |A^{-1}| |A| |x| \|_\infty / \|x\|_\infty \leq \text{cond}(A) = \| |A^{-1}| |A| \|_\infty \leq \kappa_\infty(A)$.

Note that $\text{cond}(A, x)$ can be much smaller than $\text{cond}(A)$.

Forward and backward error bounds can be computed by calling f07mv, and an estimate for $\kappa_\infty(A)$ ($= \kappa_1(A)$) can be obtained by calling f07mu.

8 Further Comments

The total number of real floating-point operations is approximately $8n^2r$.

This function may be followed by a call to f07mv to refine the solution and return an error estimate.

The real analogue of this function is f07me.

9 Example

```

uplo = 'L';
a = [complex(-1.36, +0), complex(0, +0), complex(0, +0), complex(0, +0);
      complex(3.91, -1.5), complex(-1.84, +0), complex(0, +0), complex(0,
+0);
      complex(0.3100287981271241, +0.04333020743962702),
complex(0.5637050486508776, ...
+0.2850349501519716), complex(-5.417624387291579, +0), complex(0,
+0);
      complex(-0.1518120207240102, +0.3742958425613706),
complex(0.339658279960361, ...
+0.03031451811355637), complex(0.2997244646075835,
+0.1578268372785777), complex(-7.102809895801842, +0)];
ipiv = [int32(-4);
        int32(-4);
        int32(3);
        int32(4)];
b = [complex(7.79, +5.48), complex(-35.39, +18.01);
      complex(-0.77, -16.05), complex(4.23, -70.02);
      complex(-9.58, +3.88), complex(-24.79, -8.4);
      complex(2.98, -10.18), complex(28.68, -39.89)];
[bOut, info] = f07ms(uplo, a, ipiv, b)

bOut =
    1.0000 - 1.0000i    3.0000 - 4.0000i
   -1.0000 + 2.0000i   -1.0000 + 5.0000i
    3.0000 - 2.0000i    7.0000 - 2.0000i
    2.0000 + 1.0000i   -8.0000 + 6.0000i
info =
      0

```