NAG Toolbox for MATLAB

f07mn

1 Purpose

f07mn computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian matrix and X and B are n by r matrices.

2 Syntax

```
[a, ipiv, b, info] = f07mn(uplo, a, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07mn uses the diagonal pivoting method to factor A as $A = UDU^{\mathrm{H}}$ if **uplo** = 'U' or $A = LDL^{\mathrm{H}}$ if **uplo** = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, and D is Hermitian and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of A is then used to solve the system of equations $AX = B$.

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: **uplo – string**

If **uplo** = 'U', the upper triangle of A is stored.

If **uplo** = 'L', the lower triangle of A is stored.

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

The n by n Hermitian matrix A.

If **uplo** = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

3: **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 $\max(1, \mathbf{nrhs\_p})$

Note: To solve the equations $Ax = b$, where b is a single right-hand side, **b** may be supplied as a one-dimensional array with length **ldb** $= \max(1, \mathbf{n})$.

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**.

n, the number of linear equations, i.e., 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, i.e., the number of columns of the matrix B.

Constraint: **nrhs_p** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, work, lwork

5.4 Output Parameters

1: **a(lda,$*$) – complex array**

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

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

If **info** $= 0$, the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $\mathbf{a} = UDU^{\mathrm{H}}$ or $\mathbf{a} = LDL^{\mathrm{H}}$ as computed by f07mr.

2: **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 determined by f07mr.

ipiv$(k) > 0$

Rows and columns k and **ipiv**(k) were interchanged, and $D(k, k)$ is a 1 by 1 diagonal block.

uplo $=$ 'U' and **ipiv**$(k) =$ **ipiv**$(k - 1) < 0$

Rows and columns $k - 1$ and $-$**ipiv**(k) were interchanged and $D(k - 1 : k, k - 1 : k)$ is a 2 by 2 diagonal block.

uplo $=$ 'L' and **ipiv**$(k) =$ **ipiv**$(k + 1) < 0$

Rows and columns $k + 1$ and $-$**ipiv**(k) were interchanged and $D(k : k + 1, k : k + 1)$ is a 2 by 2 diagonal block.

3: **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 $\max(1, \textbf{nrhs\_p})$

Note: To solve the equations $Ax = b$, where b is a single right-hand side, **b** may be supplied as a one-dimensional array with length **ldb** $= \max(1, \mathbf{n})$.

If **info** $= 0$, the n by r solution matrix X.

4: **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: **work**, 10: **lwork**, 11: **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.

info > 0

If **info** $= i$, d_{ii} is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* 1999 for further details.

f07mp is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, f04ch solves $Ax = b$ and returns a forward error bound and condition estimate. f04ch calls f07mn to solve the equations.

8 Further Comments

The total number of floating-point operations is approximately $\frac{4}{3}n^3 + 8n^2r$, where r is the number of right-hand sides.

The real analogue of this function is f07ma.

9 Example

```
uplo = 'Upper';
a  =  [complex(-1.84,  +0),  complex(0.11,  -0.11),  complex(-1.78,  -1.18),
complex(3.91, -1.5);
           complex(0,  0),  complex(-4.63,  +0),  complex(-1.84,  +0.03),
complex(2.21, +0.21);
      complex(0, 0), complex(0, 0), complex(-8.87, +0), complex(1.58, -
0.9);
     complex(0, 0), complex(0, 0), complex(0, 0), complex(-1.36, +0)];
b = [complex(2.98, -10.18);
```

```
     complex(-9.58, +3.88);
     complex(-0.77, -16.05);
     complex(7.79, +5.48)];
[aOut, ipiv, bOut, info] = f07mn(uplo, a, b)
```

```
aOut =
   -7.1028                0.2997 + 0.1578i    0.3397 + 0.0303i   -0.1518 +
0.3743i
         0             -5.4176                0.5637 + 0.2850i    0.3100 +
0.0433i
         0                   0             -1.8400                3.9100 -
1.5000i
        0                 0                 0            -1.3600
ipiv =
           1
           2
          -1
          -1
bOut =
   2.0000 + 1.0000i
   3.0000 - 2.0000i
  -1.0000 + 2.0000i
   1.0000 - 1.0000i
info =
           0
```