NAG Toolbox for MATLAB

f07cn

1 Purpose

f07cn computes the solution to a complex system of linear equations

$$AX = B,$$

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

2 Syntax

```
[dl, d, du, b, info] = f07cn(dl, d, du, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07cn uses the Gaussian elimination with partial pivoting and row interchanges to solve the equations $AX = B$. The matrix A is factorized as $A = PLU$, where P is a permutation matrix, L is unit lower triangular with at most one nonzero subdiagonal element per column, and U is an upper triangular band matrix, with two superdiagonals.

Note that the equation $A^{\mathrm{T}}X = B$ may be solved by interchanging the order of the arguments **du** and **dl**.

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

5 Parameters

5.1 Compulsory Input Parameters

1: **dl**$(*)$ **– complex array**

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

Must contain the $(n - 1)$ subdiagonal elements of the matrix A.

2: **d**$(*)$ **– complex array**

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

Must contain the n diagonal elements of the matrix A.

3: **du**$(*)$ **– complex array**

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

Must contain the $(n - 1)$ superdiagonal elements of the matrix A.

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

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 dimension of the array **d**.

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: $\mathbf{nrhs\_p} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldb

5.4 Output Parameters

1: **dl**$(*)$ **– complex array**

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

If $\mathbf{info} \geq 0$, **dl** contains the $(n-2)$ elements of the second superdiagonal of the upper triangular matrix U from the LU factorization of A, in $\mathbf{dl}(1), \mathbf{dl}(2), \ldots, \mathbf{dl}(n-2)$.

2: **d**$(*)$ **– complex array**

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

If $\mathbf{info} \geq 0$, **d** contains the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

3: **du**$(*)$ **– complex array**

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

If $\mathbf{info} \geq 0$, **du** contains the $(n-1)$ elements of the first superdiagonal of U.

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

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 $\mathbf{ldb} = \max(1, \mathbf{n})$.

If $\mathbf{info} = 0$, the n by r solution matrix X.

5: **info – int32 scalar**

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If $\mathbf{info} = -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **n**, 2: **nrhs_p**, 3: **dl**, 4: **d**, 5: **du**, 6: **b**, 7: **ldb**, 8: **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** $= i$, u_{ii} is exactly zero, and the solution has not been computed. The factorization has not been completed unless $i =$ **n**.

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.

Alternatives to f07cn, which return condition and error estimates are f04cc and f07cp.

8 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to nr.

The real analogue of this function is f07ca.

9 Example

```
dl = [complex(1, -2);
     complex(1, +1);
     complex(2, -3);
     complex(1, +1)];
d = [complex(-1.3, +1.3);
     complex(-1.3, +1.3);
     complex(-1.3, +3.3);
     complex(-0.3, +4.3);
     complex(-3.3, +1.3)];
du = [complex(2, -1);
     complex(2, +1);
     complex(-1, +1);
     complex(1, -1)];
b = [complex(2.4, -5);
     complex(3.4, +18.2);
     complex(-14.7, +9.699999999999999);
     complex(31.9, -7.7);
     complex(-1, +1.6)];
[dlOut, dOut, duOut, bOut, info] = f07cn(dl, d, du, b)
```

```
dlOut =
   2.0000 + 1.0000i
  -1.0000 + 1.0000i
   1.0000 - 1.0000i
   1.0000 + 1.0000i
dOut =
   1.0000 - 2.0000i
```

```
   1.0000 + 1.0000i
   2.0000 - 3.0000i
   1.0000 + 1.0000i
  -1.3399 + 0.2875i
duOut =
  -1.3000 + 1.3000i
  -1.3000 + 3.3000i
  -0.3000 + 4.3000i
  -3.3000 + 1.3000i
bOut =
   1.0000 + 1.0000i
   3.0000 - 1.0000i
   4.0000 + 5.0000i
  -1.0000 - 2.0000i
   1.0000 - 1.0000i
info =
           0
```