NAG Toolbox for MATLAB

f07cr

1 Purpose

f07cr computes the LU factorization of a complex n by n tridiagonal matrix A.

2 Syntax

```
[dl, d, du, du2, ipiv, info] = f07cr(dl, d, du, 'n', n)
```

3 Description

f07cr uses Gaussian elimination with partial pivoting and row interchanges to factorize the matrix A as

$$A = PLU,$$

where P is a permutation matrix, L is unit lower triangular with at most one nonzero subdiagonal element in each column, and U is an upper triangular band matrix, with two superdiagonals.

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.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the order of the matrix A.

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

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **dl**$(*)$ – **complex array**

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

Contains the $(n - 1)$ multipliers that define the matrix L of the LU factorization of A.

2: **d**$(*)$ – **complex array**

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

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

Contains the $(n - 1)$ elements of the first superdiagonal of U.

4: **du2**$(*)$ – **complex array**

Note: the dimension of the array **du2** must be at least $\max(1, \mathbf{n} - 2)$.

Contains the $(n - 2)$ elements of the second superdiagonal of U.

5: **ipiv**$(*)$ – **int32 array**

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

Contains the n pivot indices that define the permutation matrix P. At the ith step, row i of the matrix was interchanged with row **ipiv**(i). **ipiv**(i) will always be either i or $(i + 1)$, **ipiv**$(i) = i$ indicating that a row interchange was not performed.

6: **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: **n**, 2: **dl**, 3: **d**, 4: **du**, 5: **du2**, 6: **ipiv**, 7: **info**.

info > 0

If **info** $= i$, $U(i, i)$ is exactly zero. The factorization has been completed, but the factor U is exactly singular, and division by zero will occur if it is used to solve a system of equations.

7 Accuracy

The computed factorization satisfies an equation of the form

$$A + E = PLU,$$

where

$$\|E\|_\infty = O(\epsilon)\|A\|_\infty$$

and ϵ is the ***machine precision***.

Following the use of this function, f07cs can be used to solve systems of equations $AX = B$ or $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$, and f07cu can be used to estimate the condition number of A.

8 Further Comments

The total number of floating-point operations required to factorize the matrix A is proportional to n.

The real analogue of this function is f07cd.

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)];
[dlOut, dOut, duOut, du2, ipiv, info] = f07cr(dl, d, du)
```

```
dlOut =
  -0.7800 - 0.2600i
   0.1620 - 0.4860i
  -0.0452 - 0.0010i
  -0.3979 - 0.0562i
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
du2 =
   2.0000 + 1.0000i
  -1.0000 + 1.0000i
   1.0000 - 1.0000i
ipiv =
           2
           3
           4
           5
           5
info =
           0
```