NAG Toolbox for MATLAB

f04cc

1 Purpose

f04cc 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. An estimate of the condition number of A and an error bound for the computed solution are also returned.

2 Syntax

```
[dl, d, du, du2, ipiv, b, rcond, errbnd, ifail] = f04cc(dl, d, du, b,
'n', n, 'nrhs_p', nrhs_p)
```

3 Description

The LU decomposition with partial pivoting and row interchanges is used to factor A as $A = PLU$, where P is a permutation matrix, L is unit lower triangular with at most one nonzero subdiagonal element, and U is an upper triangular band matrix with two superdiagonals. The factored form of A is then used to solve the system of equations $AX = B$.

Note that the equations $A^{\mathrm{T}}X = B$ 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

Higham N J 2002 *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

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

The n by r matrix of right-hand sides B.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

The number of linear equations n, 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**.

The number of right-hand sides r, 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 **ifail** ≥ 0, **dl** contains the $(n - 1)$ multipliers that define the matrix L from the LU factorization of A.

2: **d**$(*)$ **– complex array**

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

If **ifail** ≥ 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 **ifail** ≥ 0, **du** 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 $\max(1, \mathbf{n} - 2)$.

If **ifail** ≥ 0, **du2** returns 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 $\max(1, \mathbf{n})$.

If **ifail** ≥ 0, the 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**$(i) = i$ indicates a row interchange was not required.

6: **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 $\max(1, \mathbf{nrhs\_p})$. 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 **ifail** $= 0$ or $Np1$, the n by r solution matrix X.

7: **rcond – double scalar**

If **ifail** ≥ 0, an estimate of the reciprocal of the condition number of the matrix A, computed as **rcond** $= 1/\left(\|A\|_1 \|A^{-1}\|_1\right)$.

8: **errbnd – double scalar**

If **ifail** $= 0$ or $Np1$, an estimate of the forward error bound for a computed solution $\hat{x}$, such that $\|\hat{x} - x\|_1 / \|x\|_1 \leq$ **errbnd**, where $\hat{x}$ is a column of the computed solution returned in the array **b** and x is the corresponding column of the exact solution X. If **rcond** is less than ***machine precision***, then **errbnd** is returned as unity.

9: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail < 0 and **ifail** $\neq -999$

If **ifail** $= -i$, the ith argument had an illegal value.

ifail $= -999$

Allocation of memory failed. The ***complex*16*** allocatable memory required is $2 \times$ **n**. In this case the factorization and the solution X have been computed, but **rcond** and **errbnd** have not been computed.

ifail > 0 and **ifail** $\leq N$

If **ifail** $= i$, u_{ii} is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution could not be computed.

ifail $= N + 1$

rcond is less than ***machine precision***, so that the matrix A is numerically singular. A solution to the equations $AX = B$ has nevertheless been 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 $\kappa(A) = \|A^{-1}\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. f04cc uses the approximation $\|E\|_1 = \epsilon\|A\|_1$ to estimate **errbnd**. See Section 4.4 of Anderson *et al.* 1999 for further details.

8 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to nr. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham 2002 for further details.

The real analogue of f04cc is f04bc.

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(2.7, +6.9);
     complex(3.4, +18.2), complex(-6.9, -5.3);
     complex(-14.7, +9.699999999999999), complex(-6, -0.6);
     complex(31.9, -7.7), complex(-3.9, +9.300000000000001);
     complex(-1, +1.6), complex(-3, +12.2)];
[dlOut, dOut, duOut, du2, ipiv, bOut, rcond, errbnd, ifail] = f04cc(dl,
d, du, b)
```

```
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
bOut =
   1.0000 + 1.0000i   2.0000 - 1.0000i
   3.0000 - 1.0000i   1.0000 + 2.0000i
   4.0000 + 5.0000i  -1.0000 + 1.0000i
  -1.0000 - 2.0000i   2.0000 + 1.0000i
   1.0000 - 1.0000i   2.0000 - 2.0000i
```

```
rcond =
    0.0054
errbnd =
   2.0445e-14
ifail =
           0
```