NAG Toolbox for MATLAB

f07cp

1 Purpose

f07cp uses the LU factorization to compute the solution to a complex system of linear equations

$$AX = B, \quad A^{T}X = B \quad \text{or} \quad A^{H}X = B,$$

where A is a tridiagonal matrix of order n and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Syntax

```
[dlf, df, duf, du2, ipiv, x, rcond, ferr, berr, info] = f07cp(fact,
trans, dl, d, du, dlf, df, duf, du2, ipiv, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07cp performs the following steps:

1. If **fact** = 'N', the LU decomposition is used to factor the matrix A as $A = LU$, where L is a product of permutation and unit lower bidiagonal matrices and U is upper triangular with nonzeros in only the main diagonal and first two superdiagonals.
2. If some $u_{ii} = 0$, so that U is exactly singular, then the function returns with **info** $= i$. Otherwise, the factored form of A is used to estimate the condition number of the matrix A. If the reciprocal of the condition number is less than ***machine precision***, **info** $\geq N + 1$ is returned as a warning, but the function still goes on to solve for X and compute error bounds as described below.
3. The system of equations is solved for X using the factored form of A.
4. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.

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

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

5 Parameters

5.1 Compulsory Input Parameters

1: **fact – string**

Specifies whether or not the factorized form of the matrix A has been supplied.

fact = 'F'

dlf, **df**, **duf**, **du2** and **ipiv** contain the factorized form of the matrix A. **dl**, **d**, **du**, **dlf**, **df**, **duf**, **du2** and **ipiv** will not be modified.

fact = 'N'

The matrix A will be copied to **dlf**, **df** and **duf** and factorized.

Constraint: **fact** = 'F' or 'N'.

2: **trans – string**

Specifies the form of the system of equations.

trans = 'N'

$AX = B$ (No transpose).

trans = 'T'

$A^{T}X = B$ (Transpose).

trans = 'C'

$A^{H}X = B$ (Conjugate transpose).

Constraint: **trans** = 'N', 'T' or 'C'.

3: **dl**$(*)$ **– complex array**

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

The $(n - 1)$ subdiagonal elements of A.

4: **d**$(*)$ **– complex array**

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

The n diagonal elements of A.

5: **du**$(*)$ **– complex array**

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

The $(n - 1)$ superdiagonal elements of A.

6: **dlf**$(*)$ **– complex array**

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

If **fact** = 'F', **dlf** contains the $(n - 1)$ multipliers that define the matrix L from the LU factorization of A.

7: **df**$(*)$ **– complex array**

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

If **fact** = 'F', **df** contains the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

8: **duf**$(*)$ **– complex array**

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

If **fact** = 'F', **duf** contains the $(n - 1)$ elements of the first superdiagonal of U.

9: **du2**$(*)$ **– complex array**

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

If **fact** = 'F', **du2** contains the $(n - 2)$ elements of the second superdiagonal of U.

10: **ipiv**$(*)$ **– int32 array**

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

If **fact** = 'F', **ipiv** contains the pivot indices from the LU factorization of A.

11: **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 dimension of the array **d** The dimension of the array **df** 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, 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, ldx, work, rwork

5.4 Output Parameters

1: **dlf**$(*)$ **– complex array**

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

If **fact** = 'N', **dlf** contains the $(n - 1)$ multipliers that define the matrix L from the LU factorization of A.

2: **df**$(*)$ **– complex array**

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

If **fact** = 'N', **df** contains the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

3: **duf**$(*)$ **– complex array**

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

If **fact** = 'N', **duf** 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, **n** $-$ 2).

If **fact** = 'N', **du2** 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, **n**).

If **fact** = 'N', **ipiv** contains the pivot indices from the LU factorization of A; 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: **x**(**ldx**,$*$) – **complex array**

The first dimension of the array **x** must be at least max(1, **n**)

The second dimension of the array must be at least max(1, **nrhs_p**)

If **info** = 0 or **info** $\geq N+1$, the n by r solution matrix X.

7: **rcond** – **double scalar**

The estimate of the reciprocal condition number of the matrix A. If **rcond** = 0, the matrix may be exactly singular. This condition is indicated by a return code of **info** > 0leqN. Otherwise, if **rcond** is less than the ***machine precision***, the matrix is singular to working precision. This condition is indicated by a return code of **info** $\geq N+1$.

8: **ferr**($*$) – **double array**

Note: the dimension of the array **ferr** must be at least max(1, **nrhs_p**).

If **info** = 0 or **info** $\geq N+1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq$ **ferr**(j) where $\hat{x}_j$ is the jth column of the computed solution returned in the array **x** and x_j is the corresponding column of the exact solution X. The estimate is as reliable as the estimate for **rcond**, and is almost always a slight overestimate of the true error.

9: **berr**($*$) – **double array**

Note: the dimension of the array **berr** must be at least max(1, **nrhs_p**).

If **info** = 0 or **info** $\geq N+1$, an estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

10: **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: **fact**, 2: **trans**, 3: **n**, 4: **nrhs_p**, 5: **dl**, 6: **d**, 7: **du**, 8: **dlf**, 9: **df**, 10: **duf**, 11: **du2**, 12: **ipiv**, 13: **b**, 14: **ldb**, 15: **x**, 16: **ldx**, 17: **rcond**, 18: **ferr**, 19: **berr**, 20: **work**, 21: **rwork**, 22: **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 and **info** $\leq N$

If **info** $= i$, $u(i,i)$ is exactly zero. The factorization has not been completed unless $i = \mathbf{n}$, but the factor U is exactly singular, so the solution and error bounds could not be computed. **rcond** $= 0$ is returned.

info $= N + 1$

U is nonsingular, but **rcond** is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of **rcond** would suggest.

7 Accuracy

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

$$|E| \leq c(n)\epsilon|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. See Section 9.3 of Higham 2002 for further details.

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

$$\frac{\|x-\hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \operatorname{cond}(A,\hat{x},b)$$

where $\operatorname{cond}(A,\hat{x},b) = \left\| |A^{-1}|(|A||\hat{x}|+|b|)\right\|_\infty / \|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth column of X, then w_c is returned in $\mathbf{berr}(j)$ and a bound on $\|x-\hat{x}\|_\infty / \|\hat{x}\|_\infty$ is returned in $\mathbf{ferr}(j)$. 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. The solution is then refined, and the errors estimated, using iterative refinement.

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 this function is f07cb.

9 Example

```
fact = 'No factors';
trans = 'No transpose';
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)];
dlf = complex(zeros(4,1));
```

```
df  = complex(zeros(5,1));
duf = complex(zeros(4,1));
du2 = complex(zeros(3,1));
ipiv = [int32(8186736);
     int32(-16);
     int32(-1232603712);
     int32(-1208358296);
     int32(7)];
b = [complex(2.4, -5), complex(2.7, +6.9);
     complex(3.4, +18.2), complex(-6.9, -5.3);
     complex(-14.7, +9.7), complex(-6, -0.6);
     complex(31.9, -7.7), complex(-3.9, +9.3);
     complex(-1, +1.6), complex(-3, +12.2)];
[dlfOut, dfOut, dufOut, du2Out, ipivOut, x, rcond, ferr, berr, info] =
...
    f07cp(fact, trans, dl, d, du, dlf, df, duf, du2, ipiv, b)
```

```
dlfOut =
  -0.7800 - 0.2600i
   0.1620 - 0.4860i
  -0.0452 - 0.0010i
  -0.3979 - 0.0562i
dfOut =
   1.0000 - 2.0000i
   1.0000 + 1.0000i
   2.0000 - 3.0000i
   1.0000 + 1.0000i
  -1.3399 + 0.2875i
dufOut =
  -1.3000 + 1.3000i
  -1.3000 + 3.3000i
  -0.3000 + 4.3000i
  -3.3000 + 1.3000i
du2Out =
   2.0000 + 1.0000i
  -1.0000 + 1.0000i
   1.0000 - 1.0000i
ipivOut =
           2
           3
           4
           5
           5
x =
   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
ferr =
   1.0e-13 *
    0.5290
    0.7736
berr =
   1.0e-15 *
    0.0223
    0.1009
info =
           0
```