NAG Toolbox for MATLAB

f07bn

1 Purpose

f07bn computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n band matrix, with k_l subdiagonals and k_u superdiagonals, and X and B are n by r matrices.

2 Syntax

```
[ab, ipiv, b, info] = f07bn(kl, ku, ab, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07bn uses the LU decomposition with partial pivoting and row interchanges to factor A as $A = PLU$, where P is a permutation matrix, L is a product of permutation and unit lower triangular matrices with k_l subdiagonals, and U is upper triangular with $(k_l + k_u)$ superdiagonals. 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: **kl – int32 scalar**

k_l, the number of subdiagonals within the band of the matrix A.

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

2: **ku – int32 scalar**

k_u, the number of superdiagonals within the band of the matrix A.

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

3: **ab(ldab,$*$) – complex array**

The first dimension of the array **ab** must be at least $2 \times \mathbf{kl} + \mathbf{ku} + 1$

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

The n by n coefficient matrix A.

The matrix is stored in rows $k_l + 1$ to $2k_l + k_u + 1$; the first k_l rows need not be set, more precisely, the element A_{ij} must be stored in

{{{{{\it A}}}_{{{{\it i}{\it j}}}}}}**ab**$(k_l + k_u + 1 + i - j, j)$ for $\max(1, j - k_u) \leq i \leq \min(n, j + k_l)$.

See Section 8 for further details.

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

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 **ab**.

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

ldab, ldb

5.4 Output Parameters

1: **ab(ldab,$*$) – complex array**

The first dimension of the array **ab** must be at least $2 \times \mathbf{kl} + \mathbf{ku} + 1$

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

If **info** ≥ 0, **ab** contains details of the factorization.

The upper triangular band matrix U, with $k_l + k_u$ superdiagonals, is stored in rows 1 to $k_l + k_u + 1$ of the array, and the multipliers used to form the matrix L are stored in rows $k_l + k_u + 2$ to $2k_l + k_u + 1$.

2: **ipiv($*$) – int32 array**

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

If **info** ≥ 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**$(i) = i$ indicates a row interchange was not required.

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

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: **n**, 2: **kl**, 3: **ku**, 4: **nrhs_p**, 5: **ab**, 6: **ldab**, 7: **ipiv**, 8: **b**, 9: **ldb**, 10: **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. The factorization has been completed, but the factor U 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.

Following the use of this function, f07bu can be used to estimate the condition number of A and f07bv can be used to obtain approximate error bounds. Alternatives to f07bn, which return condition and error estimates directly are f04cb and f07bp.

8 Further Comments

The band storage scheme for the array **ab** is illustrated by the following example, when $n = 6$, $k_l = 1$, and $k_u = 2$. Storage of the band matrix A in the array **ab**:

$$\begin{array}{cccccc}
* & * & * & + & + & + \\
* & * & a_{13} & a_{24} & a_{35} & a_{46} \\
* & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\
a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\
a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & *
\end{array}$$

Array elements marked $*$ need not be set and are not referenced by the function. Array elements marked $+$ need not be set, but are defined on exit from the function and contain the elements u_{14}, u_{25} and u_{36}.

The total number of floating-point operations required to solve the equations $AX = B$ depends upon the pivoting required, but if $n \gg k_l + k_u$ then it is approximately bounded by $O(nk_l(k_l + k_u))$ for the factorization and $O(n(2k_l + k_u)r)$ for the solution following the factorization.

The real analogue of this function is f07ba.

9 Example

```
kl = int32(1);
ku = int32(2);
ab = [complex(0, 0), complex(0, 0), complex(0, 0), complex(0, 0);
      complex(0, 0), complex(0, 0), complex(0.97, -2.84), complex(0.59, -
0.48);
         complex(0,  +0),  complex(-2.05,  -0.85),  complex(-3.99,  +4.01),
complex(3.33, -1.04);
     complex(-1.65, +2.26), complex(-1.48, -1.75), complex(-1.06, +1.94),
complex(-0.46, -1.72);
        complex(0,  +6.3),  complex(-0.77,  +2.83),  complex(4.48,  -1.09),
complex(0, 0)];
b = [complex(-1.06, +21.5);
     complex(-22.72, -53.9);
     complex(28.24, -38.6);
     complex(-34.56, +16.73)];
[abOut, ipiv, bOut, info] = f07bn(kl, ku, ab, b)
```

```
abOut =
         0                    0                    0               0.5900 -
0.4800i
         0                    0              -3.9900 + 4.0100i   3.3300 -
1.0400i
         0              -1.4800 - 1.7500i   -1.0600 + 1.9400i   -1.7692 -
1.8587i
         0 + 6.3000i   -0.7700 + 2.8300i    4.9303 - 3.0086i    0.4338 +
0.1233i
   0.3587 + 0.2619i   0.2314 + 0.6358i   0.7604 + 0.2429i       0
ipiv =
           2
           3
           3
           4
bOut =
  -3.0000 + 2.0000i
   1.0000 - 7.0000i
  -5.0000 + 4.0000i
   6.0000 - 8.0000i
info =
           0
```