NAG Toolbox for MATLAB

f04bb

1 Purpose

f04bb computes the solution to a real 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. An estimate of the condition number of A and an error bound for the computed solution are also returned.

2 Syntax

```
[ab, ipiv, b, rcond, errbnd, ifail] = f04bb(kl, ku, ab, 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 the product of permutation matrices 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

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

5 Parameters

5.1 Compulsory Input Parameters

1: **kl – int32 scalar**

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

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

2: **ku – int32 scalar**

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

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

3: **ab(ldab,$*$) – double 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 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}}}}} $\mathbf{ab}(k_l + k_u + 1 + i - j, j)$ for $\max(1, j - k_u) \leq i \leq \min(n, j + k_l)$.

See Section 8 for further details.

4: **b(ldb,$*$) – double 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 second dimension of the array **ab**.

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

ldab, ldb

5.4 Output Parameters

1: **ab(ldab,$*$) – double 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 **ifail** ≥ 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 **ifail** ≥ 0, the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row $\mathbf{ipiv}(i)$. $\mathbf{ipiv}(i) = i$ indicates a row interchange was not required.

3: **b(ldb,$*$) – double 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.

4: **rcond – double scalar**

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

5: **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.

6: **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 integer allocatable memory required is **n**, and the double allocatable memory required is $3 \times$ **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. f04bb 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 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 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 complex analogue of f04bb is f04cb.

9 Example

```
kl = int32(1);
ku = int32(2);
ab = [0, 0, 0, 0;
     0, 0, -3.66, -2.13;
     0, 2.54, -2.73, 4.07;
     -0.23, 2.46, 2.46, -3.82;
     -6.98, 2.56, -4.78, 0];
b = [4.42, -36.01;
     27.13, -31.67;
     -6.14, -1.16;
     10.5, -25.82];
[abOut, ipiv, bOut, rcond, errbnd, ifail] = f04bb(kl, ku, ab, b)
```

```
abOut =
         0         0         0   -2.1300
         0         0   -2.7300    4.0700
         0    2.4600    2.4600   -3.8391
   -6.9800    2.5600   -5.9329   -0.7269
    0.0330    0.9605    0.8057         0
ipiv =
           2
           3
           3
           4
bOut =
   -2.0000    1.0000
    3.0000   -4.0000
    1.0000    7.0000
   -4.0000   -2.0000
rcond =
    0.0177
errbnd =
   6.2687e-15
ifail =
           0
```