NAG Fortran Library Routine Document

F04BBF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F04BBF computes the solution to a real system of linear equations $AX = B$, where A is an n by n band matrix, with k_l sub-diagonals and k_u super-diagonals, 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 Specification

```
      SUBROUTINE F04BBF (N, KL, KU, NRHS, AB, LDAB, IPIV, B, LDB, RCOND,
     1                   ERRBND, IFAIL)
      INTEGER            N, KL, KU, NRHS, LDAB, IPIV(*), LDB, IFAIL
      double precision   AB(LDAB,*), B(LDB,*), RCOND, ERRBND
```

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 sub-diagonals, and U is upper triangular with $(k_l + k_u)$ super-diagonals. 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

1: N – INTEGER *Input*

On entry: the number of linear equations n, i.e., the order of the matrix A.

Constraint: $N \geq 0$.

2: KL – INTEGER *Input*

On entry: the number of sub-diagonals k_l, within the band of A.

Constraint: $KL \geq 0$.

3: KU – INTEGER *Input*

On entry: the number of super-diagonals k_u, within the band of A.

Constraint: $KU \geq 0$.

4: NRHS – INTEGER *Input*

On entry: the number of right-hand sides r, i.e., the number of columns of the matrix B.

Constraint: $NRHS \geq 0$.

5: AB(LDAB,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array AB must be at least $\max(1, \mathrm{N})$.

On entry: the n by n coefficient matrix A in band storage, in rows $\mathrm{KL}+1$ to $2 \times \mathrm{KL}+\mathrm{KU}+1$; rows 1 to KL of the array need not be set. The jth column of A is stored in the jth column of the array AB as follows:

$$\mathrm{AB}(k_l + k_u + 1 + i - j, j) = a_{ij} \quad \text{for } \max(1, j - k_u) \leq i \leq \min(n, j + k_l)$$

See Section 8 below for further details.

On exit: if IFAIL ≥ 0, details of the factorization $A = PLU$; U is stored as an upper triangular band matrix with $\mathrm{KL}+\mathrm{KU}$ super-diagonals in rows 1 to $\mathrm{KL}+\mathrm{KU}+1$, and the multipliers used during the factorization are stored in rows $\mathrm{KL}+\mathrm{KU}+2$ to $2 \times \mathrm{KL}+\mathrm{KU}+1$.

6: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F04BBF is called.

Constraint: $\mathrm{LDAB} \geq 2 \times \mathrm{KL}+\mathrm{KU}+1$.

7: IPIV($*$) – INTEGER array *Output*

Note: the dimension of the array IPIV must be at least $\max(1, \mathrm{N})$.

On exit: 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) $= i$ indicates a row interchange was not required.

8: B(LDB,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, \mathrm{NRHS})$. 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 $\mathrm{LDB} = \max(1, \mathrm{N})$.

On entry: the n by r matrix of right-hand sides B.

On exit: if IFAIL $= 0$ or $\mathrm{N}+1$, the n by r solution matrix X.

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04BBF is called.

Constraint: $\mathrm{LDB} \geq \max(1, \mathrm{N})$.

10: RCOND – ***double precision*** *Output*

On exit: if IFAIL ≥ 0, an estimate of the reciprocal of the condition number of the matrix A, computed as $\mathrm{RCOND} = 1/\left(\|A\|_1 \|A^{-1}\|_1\right)$.

11: ERRBND – ***double precision*** *Output*

On exit: if IFAIL $= 0$ or $\mathrm{N}+1$, an estimate of the forward error bound for a computed solution $\hat{x}$, such that $\|\hat{x} - x\|_1 / \|x\|_1 \leq \mathrm{ERRBND}$, 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.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

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 precision*** 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 $\kappa(A) = \left\|A^{-1}\right\|_1\|A\|_1$, the condition number of A with respect to the solution of the linear equations. F04BBF 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 routine. Array elements marked $+$ need not be set, but are defined on exit from the routine 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 F04BBF is F04CBF.

9 Example

To solve the equations

$$AX = B,$$

where A is the band matrix

$$A = \begin{pmatrix} -0.23 & 2.54 & -3.66 & 0 \\ -6.98 & 2.46 & -2.73 & -2.13 \\ 0 & 2.56 & 2.46 & 4.07 \\ 0 & 0 & -4.78 & -3.82 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 4.42 & -36.01 \\ 27.13 & -31.67 \\ -6.14 & -1.16 \\ 10.50 & -25.82 \end{pmatrix}.$$

An estimate of the condition number of A and an approximate error bound for the computed solutions are also printed.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F04BBF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KLMAX, KUMAX, NRHSMX
      PARAMETER        (NMAX=8,KLMAX=4,KUMAX=4,NRHSMX=2)
      INTEGER          LDAB, LDB
      PARAMETER        (LDAB=2*KLMAX+KUMAX+1,LDB=NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION ERRBND, RCOND
      INTEGER          I, IERR, IFAIL, J, K, KL, KU, N, NRHS
*     .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,NMAX), B(LDB,NRHSMX)
      INTEGER          IPIV(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04BBF, X04CAF, X04CEF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04BBF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KL, KU, NRHS
```

```
      IF (N.LE.NMAX .AND. KL.LE.KLMAX .AND. KU.LE.KUMAX .AND. NRHS.LE.
     +    NRHSMX) THEN
*
*        Read A and B from data file
*
         K = KL + KU + 1
         READ (NIN,*) ((AB(K+I-J,J),J=MAX(I-KL,1),MIN(I+KU,N)),I=1,N)
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
*        Solve the equations AX = B for X
*
         IFAIL = -1
         CALL F04BBF(N,KL,KU,NRHS,AB,LDAB,IPIV,B,LDB,RCOND,ERRBND,IFAIL)
*
         IF (IFAIL.EQ.0) THEN
*
*           Print solution, estimate of condition number and approximate
*           error bound
*
            IERR = 0
            CALL X04CAF('General',' ',N,NRHS,B,LDB,'Solution',IERR)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Estimate of condition number'
            WRITE (NOUT,99999) 1.0D0/RCOND
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Estimate of error bound for computed solutions'
            WRITE (NOUT,99999) ERRBND
         ELSE IF (IFAIL.EQ.N+1) THEN
*
*           Matrix A is numerically singular.  Print estimate of
*           reciprocal of condition number and solution
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Estimate of reciprocal of condition number'
            WRITE (NOUT,99999) RCOND
*
            WRITE (NOUT,*)
            IERR = 0
            CALL X04CAF('General',' ',N,NRHS,B,LDB,'Solution',IERR)
*
         ELSE IF (IFAIL.GT.0 .AND. IFAIL.LE.N) THEN
*
*           The upper triangular matrix U is exactly singular.  Print
*           details of factorization
*
            WRITE (NOUT,*)
            IERR = 0
            CALL X04CEF(N,N,KL,KL+KU,AB,LDAB,'Details of factorization',
     +                  IERR)
*
*           Print pivot indices
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Pivot indices'
            WRITE (NOUT,99998) (IPIV(I),I=1,N)
         END IF
      ELSE
         WRITE (NOUT,*)
     +     'One or more of NMAX, KLMAX, KUMAX or NRHSMX is too small'
      END IF
      STOP
*
99999 FORMAT (6X,1P,E9.1)
99998 FORMAT ((1X,7I11))
      END
```

9.2 Program Data

```
F04BBF Example Program Data

  4      1      2      2    :Values of N, KL, KU and NRHS

 -0.23   2.54  -3.66
 -6.98   2.46  -2.73  -2.13
         2.56   2.46   4.07
               -4.78  -3.82 :End of matrix A

  4.42 -36.01
 27.13 -31.67
 -6.14  -1.16
 10.50 -25.82               :End of matrix B
```

9.3 Program Results

```
 F04BBF Example Program Results

 Solution
             1          2
 1     -2.0000     1.0000
 2      3.0000    -4.0000
 3      1.0000     7.0000
 4     -4.0000    -2.0000

 Estimate of condition number
        5.6E+01

 Estimate of error bound for computed solutions
        6.3E-15
```