# NAG Fortran Library Routine Document

# F07BAF (DGBSV)

**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

F07BAF (DGBSV) 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.

## 2 Specification

```
SUBROUTINE F07BAF (N, KL, KU, NRHS, AB, LDAB, IPIV, B, LDB, INFO)
INTEGER             N, KL, KU, NRHS, LDAB, IPIV(*), LDB, INFO
double precision    AB(LDAB,*), B(LDB,*)
```

The routine may be called by its LAPACK name ***dgbsv***.

## 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 a product of permutation 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

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Parameters

1: N – INTEGER *Input*

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

*Constraint*: $\mathrm{N} \geq 0$.

2: KL – INTEGER *Input*

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

*Constraint*: $\mathrm{KL} \geq 0$.

3: KU – INTEGER *Input*

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

*Constraint*: $\mathrm{KU} \geq 0$.

4: NRHS – INTEGER *Input*

*On entry*: $r$, the number of right-hand sides, 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 $n$ coeffiencient 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 $j$th column of $A$ is stored in the $j$th column of the array AB as follows:

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

See Section 8 for further details.

*On exit*: if INFO $\geq 0$, 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 F07BAF (DGBSV) is called.

*Constraint*: 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 INFO $\geq 0$, the pivot indices that define the permutation matrix $P$; at the $i$th 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 $\max(1, \mathrm{NRHS})$.

*On entry*: the $n$ by $r$ right-hand side matrix $B$.

*On exit*: if INFO $= 0$, 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 F07BAF (DGBSV) is called.

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

10: INFO – INTEGER *Output*

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

## 6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO $< 0$

If INFO $= -i$, the $i$th argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

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 $\epsilon$ 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. See Section 4.4 of Anderson *et al.* (1999) for further details.

Following the use of F07BAF (DGBSV), F07BGF (DGBCON) can be used to estimate the condition number of $A$ and F07BHF (DGBRFS) can be used to obtain approximate error bounds. Alternatives to F07BAF (DGBSV), which return condition and error estimates directly are F04BBF and F07BBF (DGBSVX).

## 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 complex analogue of this routine is F07BNF (ZGBSV).

## 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

$$b = \begin{pmatrix} 4.42 \\ 27.13 \\ -6.14 \\ 10.50 \end{pmatrix}.$$

Details of the $LU$ factorization of $A$ are also output.

## 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.

```
*     F07BAF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KLMAX, KUMAX
      PARAMETER        (NMAX=8,KLMAX=4,KUMAX=4)
      INTEGER          LDAB
      PARAMETER        (LDAB=2*KLMAX+KUMAX+1)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, K, KL, KU, N
*     .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,NMAX), B(NMAX)
      INTEGER          IPIV(NMAX)
*     .. External Subroutines ..
      EXTERNAL         DGBSV, X04CEF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07BAF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KL, KU
      IF (N.LE.NMAX .AND. KL.LE.KLMAX .AND. KU.LE.KUMAX) THEN
*
*        Read the band matrix A and the right hand side 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),I=1,N)
*
*        Solve the equations Ax = b for x
*
         CALL DGBSV(N,KL,KU,1,AB,LDAB,IPIV,B,N,INFO)
*
         IF (INFO.EQ.0) THEN
*
*           Print solution
*
            WRITE (NOUT,*) 'Solution'
            WRITE (NOUT,99999) (B(I),I=1,N)
*
*           Print details of the factorization
*
            WRITE (NOUT,*)
            IFAIL = 0
            CALL X04CEF(N,N,KL,KL+KU,AB,LDAB,'Details of factorization',
     +                  IFAIL)
*
*           Print pivot indices'
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Pivot indices'
            WRITE (NOUT,99998) (IPIV(I),I=1,N)
```

```
         ELSE
            WRITE (NOUT,99997) 'The (', INFO, ',', INFO, ')',
     +        ' element of the factor U is zero'
         END IF
      ELSE
         WRITE (NOUT,*)
     +     'One or more of NMAX, KLMAX or KUMAX is too small'
      END IF
      STOP
*
99999 FORMAT ((3X,7F11.4))
99998 FORMAT ((3X,7I11))
99997 FORMAT (1X,A,I3,A,I3,A,A)
      END
```

## 9.2 Program Data

```
F07BAF Example Program Data

  4  1  2                     :Values of N, KL and KU

 -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  27.13  -6.14  10.50   :End of vector B
```

## 9.3 Program Results

```
 F07BAF Example Program Results

 Solution
       -2.0000     3.0000     1.0000    -4.0000

 Details of factorization
             1          2          3          4
 1     -6.9800     2.4600    -2.7300    -2.1300
 2      0.0330     2.5600     2.4600     4.0700
 3                 0.9605    -5.9329    -3.8391
 4                            0.8057    -0.7269

 Pivot indices
             2          3          3          4
```