F11JBF – NAG Fortran Library Routine Document

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

F11JBF solves a system of linear equations involving the incomplete Cholesky preconditioning matrix generated by F11JAF.

2 Specification

```
      SUBROUTINE F11JBF(N, A, LA, IROW, ICOL, IPIV, ISTR, CHECK, Y, X,
     1                  IFAIL)
      INTEGER           N, LA, IROW(LA), ICOL(LA), IPIV(N), ISTR(N+1),
     1                  IFAIL
      real              A(LA), Y(N), X(N)
      CHARACTER*1       CHECK
```

3 Description

This routine solves a system of linear equations

$$Mx = y$$

involving the preconditioning matrix $M = PLDL^TP^T$, corresponding to an incomplete Cholesky decomposition of a sparse symmetric matrix stored in symmetric coordinate storage (SCS) format (see Section 2.1.2 of the Chapter Introduction), as generated by F11JAF.

In the above decomposition L is a lower triangular sparse matrix with unit diagonal, D is a diagonal matrix and P is a permutation matrix. L and D are supplied to F11JBF through the matrix

$$C = L + D^{-1} - I$$

which is a lower triangular N by N sparse matrix, stored in SCS format, as returned by F11JAF. The permutation matrix P is returned from F11JAF via the array IPIV.

It is envisaged that a common use of F11JBF will be to carry out the preconditioning step required in the application of F11GBF to sparse symmetric linear systems. F11JBF is used for this purpose by the black-box routine F11JCF.

F11JBF may also be used in combination with F11JAF to solve a sparse symmetric positive-definite system of linear equations directly (see Section 8.4 of the document for F11JAF). This use of F11JBF is demonstrated in Section 9.

4 References

None.

5 Parameters

1: N — INTEGER *Input*

On entry: n, the order of the matrix M. This **must** be the same value as was supplied in the preceding call to F11JAF.

Constraint: $N \geq 1$.

2: A(LA) — ***real*** array *Input*

On entry: the values returned in array A by a previous call to F11JAF.

3: LA — INTEGER *Input*

On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F11JBF is called. This **must** be the same value as was supplied in the preceding call to F11JAF.

4: IROW(LA) — INTEGER array *Input*
5: ICOL(LA) — INTEGER array *Input*
6: IPIV(N) — INTEGER array *Input*
7: ISTR(N+1) — INTEGER array *Input*

On entry: the values returned in arrays IROW, ICOL, IPIV and ISTR by a previous call to F11JAF.

8: CHECK — CHARACTER*1 *Input*

On entry: specifies whether or not the input data should be checked:

if CHECK = ’C’, checks are carried out on the values of N, IROW, ICOL, IPIV and ISTR;
if CHECK = ’N’, none of these checks are carried out.

See also Section 8.2.

Constraint: CHECK = ’C’ or ’N’.

9: Y(N) — ***real*** array *Input*

On entry: the right-hand side vector y.

10: X(N) — ***real*** array *Output*

On exit: the solution vector x.

11: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Errors 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 detected by the routine:

IFAIL = 1

On entry, CHECK $\neq$ ’C’ or ’N’.

IFAIL = 2

On entry, N $<$ 1.

IFAIL = 3

On entry, the SCS representation of the preconditioning matrix M is invalid. Further details are given in the error message. Check that the call to F11JBF has been preceded by a valid call to F11JAF and that the arrays A, IROW, ICOL, IPIV and ISTR have not been corrupted between the two calls.

7 Accuracy

The computed solution x is the exact solution of a perturbed system of equations $(M+\delta M)x=y$, where

$$|\delta M| \leq c(n)\epsilon P|L||D||L^T|P^T,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

8 Further Comments

8.1 Timing

The time taken for a call to F11JBF is proportional to the value of NNZC returned from F11JAF.

8.2 Use of CHECK

It is expected that a common use of F11JBF will be to carry out the preconditioning step required in the application of F11GBF to sparse symmetric linear systems. In this situation F11JBF is likely to be called many times with the same matrix M. In the interests of both reliability and efficiency, you are recommended to set CHECK to 'C' for the first of such calls, and to 'N' for all subsequent calls.

9 Example

This example program reads in a symmetric positive-definite sparse matrix A and a vector y. It then calls F11JAF, with LFILL $= -1$ and DTOL $= 0.0$, to compute the **complete** Cholesky decomposition of A:

$$A = PLDL^TP^T.$$

Finally it calls F11JBF to solve the system

$$PLDL^TP^Tx = y.$$

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.

```
*     F11JBF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LA, LIWORK
      PARAMETER        (NMAX=1000,LA=10000,LIWORK=2*LA+7*NMAX+1)
*     .. Local Scalars ..
      real             DSCALE, DTOL
      INTEGER          I, IFAIL, LFILL, N, NNZ, NNZC, NPIVM
      CHARACTER        CHECK, MIC, PSTRAT
*     .. Local Arrays ..
      real             A(LA), X(NMAX), Y(NMAX)
      INTEGER          ICOL(LA), IPIV(NMAX), IROW(LA), ISTR(NMAX+1),
     +                 IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         F11JAF, F11JBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11JBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*
*     Read order of matrix and number of non-zero entries
*
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
         READ (NIN,*) NNZ
*
*     Read the matrix A
*
```

```
         DO 20 I = 1, NNZ
            READ (NIN,*) A(I), IROW(I), ICOL(I)
   20    CONTINUE
*
*     Read the vector y
*
         READ (NIN,*) (Y(I),I=1,N)
*
*     Calculate Cholesky factorization
*
         LFILL = -1
         DTOL = 0.0e0
         MIC = 'N'
         DSCALE = 0.0e0
         PSTRAT = 'M'
         IFAIL = 0
*
         CALL F11JAF(N,NNZ,A,LA,IROW,ICOL,LFILL,DTOL,MIC,DSCALE,PSTRAT,
     +               IPIV,ISTR,NNZC,NPIVM,IWORK,LIWORK,IFAIL)
*
*     Check the output value of NPIVM
*
         IF (NPIVM.NE.0) THEN
*
            WRITE (NOUT,*) 'Factorization is not complete'
*
         ELSE
*                   T T
*     Solve P L D L P x = y
*
            CHECK = 'C'
*
            CALL F11JBF(N,A,LA,IROW,ICOL,IPIV,ISTR,CHECK,Y,X,IFAIL)
*
*     Output results
*
            WRITE (NOUT,*) ' Solution of linear system'
            DO 40 I = 1, N
               WRITE (NOUT,99999) X(I)
   40       CONTINUE
*
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,e16.4)
      END
```

9.2 Program Data

```
F11JBF Example Program Data
  9                   N
 23                   NNZ
  4.   1    1
 -1.   2    1
  6.   2    2
  1.   3    2
  2.   3    3
  3.   4    4
  2.   5    1
  4.   5    5
  1.   6    3
  2.   6    4
  6.   6    6
 -4.   7    2
  1.   7    5
 -1.   7    6
  6.   7    7
 -1.   8    4
 -1.   8    6
  3.   8    8
  1.   9    1
  1.   9    5
 -1.   9    6
  1.   9    8
  4.   9    9           A(I), IROW(I), ICOL(I), I=1,...,NNZ
  4.10 -2.94  1.41
  2.53  4.35  1.29
  5.01  0.52  4.57     Y(I), I=1,...,N
```

9.3 Program Results

```
F11JBF Example Program Results
 Solution of linear system
      0.7000E+00
      0.1600E+00
      0.5200E+00
      0.7700E+00
      0.2800E+00
      0.2100E+00
      0.9300E+00
      0.2000E+00
      0.9000E+00
```