

F11JCF – 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

F11JCF solves a real sparse symmetric system of linear equations, represented in symmetric coordinate storage format, using a conjugate gradient or Lanczos method, with incomplete Cholesky preconditioning.

2 Specification

```

SUBROUTINE F11JCF(METHOD, N, NNZ, A, LA, IROW, ICOL, IPIV, ISTR,
1      B, TOL, MAXITN, X, RNORM, ITN, WORK, LWORK, IFAIL)
      INTEGER      N, NNZ, LA, IROW(LA), ICOL(LA), IPIV(N),
1      ISTR(N+1), MAXITN, ITN, LWORK, IFAIL
      real          A(LA), B(N), TOL, X(N), RNORM, WORK(LWORK)
      CHARACTER*(*) METHOD
```

3 Description

This routine solves a real sparse symmetric linear system of equations:

$$Ax = b,$$

using a preconditioned conjugate gradient method [2], or a preconditioned Lanczos method based on the algorithm SYMMLQ [3]. The conjugate gradient method is more efficient if A is positive-definite, but may fail to converge for indefinite matrices. In this case the Lanczos method should be used instead. For further details see [1].

F11JCF uses the incomplete Cholesky factorization determined by F11JAF as the preconditioning matrix. A call to F11JCF must always be preceded by a call to F11JAF. Alternative preconditioners for the same storage scheme are available by calling F11JEF.

The matrix A , and the preconditioning matrix M , are represented in symmetric coordinate storage (SCS) format (see Section 2.1.2 of the Chapter Introduction) in the arrays A , $IROW$ and $ICOL$, as returned from F11JAF. The array A holds the non-zero entries in the lower triangular parts of these matrices, while $IROW$ and $ICOL$ hold the corresponding row and column indices.

4 References

- [1] Barrett R, Berry M, Chan T F, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C and van der Vorst H (1994) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* SIAM, Philadelphia
- [2] Meijerink J and van der Vorst H (1977) An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix *Math. Comput.* **31** 148–162
- [3] Paige C C and Saunders M A (1975) Solution of sparse indefinite systems of linear equations *SIAM J. Numer. Anal.* **12** 617–629
- [4] Salvini S A and Shaw G J (1995) An evaluation of new NAG Library solvers for large sparse symmetric linear systems *NAG Technical Report TR1/95*

5 Parameters

- 1:** METHOD — CHARACTER(*) *Input*
On entry: specifies the iterative method to be used. The possible choices are:
 'CG' Conjugate Gradient method;
 'SYMLQ' Lanczos method (SYMLQ).
Constraint: METHOD = 'CG' or 'SYMLQ'.
- 2:** N — INTEGER *Input*
On entry: n , the order of the matrix A . This **must** be the same value as was supplied in the preceding call to F11JAF.
Constraint: $N \geq 1$.
- 3:** NNZ — INTEGER *Input*
On entry: the number of non-zero elements in the lower triangular part of the matrix A . This **must** be the same value as was supplied in the preceding call to F11JAF.
Constraint: $1 \leq \text{NNZ} \leq N \times (N+1)/2$.
- 4:** A(LA) — *real* array *Input*
On entry: the values returned in array A by a previous call to F11JAF.
- 5:** LA — INTEGER *Input*
On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F11JCF is called. This **must** be the same value as was supplied in the preceding call to F11JAF.
Constraint: $\text{LA} \geq 2 \times \text{NNZ}$.
- 6:** IROW(LA) — INTEGER array *Input*
7: ICOL(LA) — INTEGER array *Input*
8: IPIV(N) — INTEGER array *Input*
9: ISTR(N+1) — INTEGER array *Input*
On entry: the values returned in arrays IROW, ICOL, IPIV and ISTR by a previous call to F11JAF.
- 10:** B(N) — *real* array *Input*
On entry: the right-hand side vector b .
- 11:** TOL — *real* *Input*
On entry: the required tolerance. Let x_k denote the approximate solution at iteration k , and r_k the corresponding residual. The algorithm is considered to have converged at iteration k if:

$$\|r_k\|_\infty \leq \tau \times (\|b\|_\infty + \|A\|_\infty \|x_k\|_\infty).$$
If $\text{TOL} \leq 0.0$, $\tau = \max(\sqrt{\epsilon}, \sqrt{n}\epsilon)$ is used, where ϵ is the **machine precision**. Otherwise $\tau = \max(\text{TOL}, 10\epsilon, \sqrt{n}\epsilon)$ is used.
Constraint: $\text{TOL} < 1.0$.
- 12:** MAXITN — INTEGER *Input*
On entry: the maximum number of iterations allowed.
Constraint: $\text{MAXITN} \geq 1$.

- 13:** X(N) — *real* array *Input/Output*
On entry: an initial approximation to the solution vector x .
On exit: an improved approximation to the solution vector x .
- 14:** RNORM — *real* *Output*
On exit: the final value of the residual norm $\|r_k\|_\infty$, where k is the output value of ITN.
- 15:** ITN — INTEGER *Output*
On exit: the number of iterations carried out.
- 16:** WORK(LWORK) — *real* array *Workspace*
17: LWORK — INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F11JCF is called.
Constraints:
 if METHOD = 'CG' then LWORK $\geq 6 \times N$;
 if METHOD = 'SYMMLQ' then LWORK $\geq 7 \times N$.
- 18:** 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, METHOD \neq 'CG' or 'SYMMLQ',
 or $N < 1$,
 or $NNZ < 1$,
 or $NNZ > N \times (N+1)/2$,
 or LA too small,
 or $TOL \geq 1.0$,
 or $MAXITN < 1$,
 or LWORK too small.

IFAIL = 2

On entry, the SCS representation of A is invalid. Further details are given in the error message. Check that the call to F11JCF has been preceded by a valid call to F11JAF, and that the arrays A, IROW, and ICOL have not been corrupted between the two calls.

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 F11JCF 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.

IFAIL = 4

The required accuracy could not be obtained. However, a reasonable accuracy has been obtained and further iterations could not improve the result.

IFAIL = 5

Required accuracy not obtained in MAXITN iterations.

IFAIL = 6

The preconditioner appears not to be positive-definite.

IFAIL = 7

The matrix of the coefficients appears not to be positive-definite (conjugate gradient method only).

IFAIL = 8

A serious error has occurred in an internal call to F11GAF, F11GBF or F11GCF. Check all subroutine calls and array sizes. Seek expert help.

7 Accuracy

On successful termination, the final residual $r_k = b - Ax_k$, where $k = \text{ITN}$, satisfies the termination criterion

$$\|r_k\|_\infty \leq \tau \times (\|b\|_\infty + \|A\|_\infty \|x_k\|_\infty).$$

The value of the final residual norm is returned in RNORM.

8 Further Comments

The time taken by F11JCF for each iteration is roughly proportional to the value of NNZC returned from the preceding call to F11JAF. One iteration with the Lanczos method (SYMMLQ) requires a slightly larger number of operations than one iteration with the conjugate gradient method.

The number of iterations required to achieve a prescribed accuracy cannot be easily determined a priori, as it can depend dramatically on the conditioning and spectrum of the preconditioned matrix of the coefficients $\bar{A} = M^{-1}A$.

Some illustrations of the application of F11JCF to linear systems arising from the discretization of two-dimensional elliptic partial differential equations, and to random-valued randomly structured symmetric positive-definite linear systems, can be found in [4].

9 Example

This example program solves a symmetric positive-definite system of equations using the conjugate gradient method, with incomplete Cholesky preconditioning.

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.

```
*      F11JCF Example Program Text
*      Mark 19 Revised. NAG Copyright 1999.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LA, LIWORK, LWORK
      PARAMETER        (NMAX=1000,LA=10000,LIWORK=2*LA+7*NMAX+1,
+                      LWORK=6*NMAX+120)
*      .. Local Scalars ..
```

```

      real          DSCALE, DTOL, RNORM, TOL
      INTEGER       I, IFAIL, ITN, LFILL, MAXITN, N, NNZ, NNZC, NPIVM
      CHARACTER     MIC, PSTRAT
      CHARACTER*6    METHOD
*    .. Local Arrays ..
      real          A(LA), B(NMAX), WORK(LWORK), X(NMAX)
      INTEGER       ICOL(LA), IPIV(NMAX), IROW(LA), ISTR(NMAX+1),
+                 IWORK(LIWORK)
*    .. External Subroutines ..
      EXTERNAL      F11JAF, F11JCF
*    .. Executable Statements ..
      WRITE (NOUT,*) 'F11JCF Example Program Results'
*    Skip heading in data file
      READ (NIN,*)
*
*    Read algorithmic parameters
*
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
        READ (NIN,*) NNZ
        READ (NIN,*) METHOD
        READ (NIN,*) LFILL, DTOL
        READ (NIN,*) MIC, DSCALE
        READ (NIN,*) PSTRAT
        READ (NIN,*) TOL, MAXITN
*
*    Read the matrix A
*
        DO 20 I = 1, NNZ
          READ (NIN,*) A(I), IROW(I), ICOL(I)
20      CONTINUE
*
*    Read right-hand side vector b and initial approximate solution x
*
        READ (NIN,*) (B(I),I=1,N)
        READ (NIN,*) (X(I),I=1,N)
*
*    Calculate incomplete Cholesky factorization
*
        IFAIL = 0
        CALL F11JAF(N,NNZ,A,LA,IROW,ICOL,LFILL,DTOL,MIC,DSCALE,PSTRAT,
+                 IPIV,ISTR,NNZC,NPIVM,IWORK,LIWORK,IFAIL)
*
*    Solve Ax = b using F11JCF
*
        CALL F11JCF(METHOD,N,NNZ,A,LA,IROW,ICOL,IPIV,ISTR,B,TOL,MAXITN,
+                 X,RNORM,ITN,WORK,LWORK,IFAIL)
*
        WRITE (NOUT,99999) 'Converged in', ITN, ' iterations'
        WRITE (NOUT,99998) 'Final residual norm =', RNORM
*
*    Output x
*
        DO 40 I = 1, N
          WRITE (NOUT,99997) X(I)
40      CONTINUE
      END IF
      STOP

```

```

*
99999 FORMAT (1X,A,I10,A)
99998 FORMAT (1X,A,1P,e16.3)
99997 FORMAT (1X,1P,e16.4)
      END

```

9.2 Program Data

F11JCF Example Program Data

7				N
16				NNZ
'CG'				METHOD
1 0.0				LFILL, DTOL
'N' 0.0				MIC, DSCALE
'M'				PSTRAT
1.0E-6 100				TOL, MAXITN
4.	1	1		
1.	2	1		
5.	2	2		
2.	3	3		
2.	4	2		
3.	4	4		
-1.	5	1		
1.	5	4		
4.	5	5		
1.	6	2		
-2.	6	5		
3.	6	6		
2.	7	1		
-1.	7	2		
-2.	7	3		
5.	7	7		A(I), IROW(I), ICOL(I), I=1,...,NNZ
15.	18.	-8.	21.	
11.	10.	29.		B(I), I=1,...,N
0.	0.	0.	0.	
0.	0.	0.		X(I), I=1,...,N

9.3 Program Results

F11JCF Example Program Results

```

Converged in      1 iterations
Final residual norm =      7.105E-15
1.0000E+00
2.0000E+00
3.0000E+00
4.0000E+00
5.0000E+00
6.0000E+00
7.0000E+00

```
