NAG Fortran Library Routine Document

F04AXF

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

F04AXF calculates the approximate solution of a set of real sparse linear equations with a single right-hand side, $Ax = b$ or $A^{\mathrm{T}}x = b$, where A has been factorized by F01BRF or F01BSF.

2 Specification

```
SUBROUTINE F04AXF (N, A, LICN, ICN, IKEEP, RHS, W, MTYPE, IDISP, RESID)
INTEGER             N, LICN, ICN(LICN), IKEEP(5*N), MTYPE, IDISP(2)
double precision    A(LICN), RHS(N), W(N), RESID
```

3 Description

To solve a system of real linear equations $Ax = b$ or $A^{\mathrm{T}}x = b$, where A is a general sparse matrix, A must first be factorized by F01BRF or F01BSF. F04AXF then computes x by block forward or backward substitution using simple forward and backward substitution within each diagonal block.

The method is fully described in Duff (1977).

A more recent method is available through solver routine F11MFF and factorization routine F11MEF .

4 References

Duff I S (1977) MA28 – a set of Fortran subroutines for sparse unsymmetric linear equations *AERE Report R8730* HMSO

5 Parameters

1: N – INTEGER *Input*

On entry: n, the order of the matrix A.

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

2: A(LICN) – ***double precision*** array *Input*

On entry: the non-zero elements in the factorization of the matrix A, as returned by F01BRF or F01BSF.

3: LICN – INTEGER *Input*

On entry: the dimension of the arrays A and ICN as declared in the (sub)program from which F04AXF is called.

4: ICN(LICN) – INTEGER array *Communication Array*

On entry: the column indices of the non-zero elements of the factorization, as returned by F01BRF or F01BSF.

5: IKEEP($5 \times \mathrm{N}$) – INTEGER array *Input*

IKEEP provides, on entry, indexing information about the factorization, as returned by F01BRF or F01BSF. Used as internal workspace prior to being restored and hence is unchanged.

6: RHS(N) – ***double precision*** array *Input/Output*

On entry: the right-hand side vector b.

On exit: is overwritten by the solution vector x.

7: W(N) – ***double precision*** array *Workspace*

8: MTYPE – INTEGER *Input*

On entry: MTYPE specifies the task to be performed.

MTYPE $= 1$

Solve $Ax = b$.

MTYPE $\neq 1$

Solve $A^{\mathrm{T}}x = b$.

9: IDISP(2) – INTEGER array *Communication Array*

On entry: the values returned in IDISP by F01BRF.

10: RESID – ***double precision*** *Output*

On exit: the value of the maximum residual, $\max\left(\left|b_i - \sum_j a_{ij}x_j\right|\right)$, over all the unsatisfied equations, in case F01BRF or F01BSF has been used to factorize a singular or rectangular matrix.

6 Error Indicators and Warnings

None.

7 Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. Since F04AXF is always used with either F01BRF or F01BSF, you are recommended to set GROW = .TRUE. on entry to these routines and to examine the value of W(1) on exit (see F01BRF and F01BSF). For a detailed error analysis see page 17 of Duff (1977).

If storage for the original matrix is available then the error can be estimated by calculating the residual

$$r = b - Ax \quad \left(\text{or } b - A^{\mathrm{T}}x\right)$$

and calling F04AXF again to find a correction δ for x by solving

$$A\delta = r \quad \left(\text{or } A^{\mathrm{T}}\delta = r\right).$$

8 Further Comments

If the factorized form contains τ non-zeros (IDISP(2) $= \tau$) then the time taken is very approximately 2τ times longer than the inner loop of full matrix code. Some advantage is taken of zeros in the right-hand side when solving $A^{\mathrm{T}}x = b$ (MTYPE $\neq 1$).

9 Example

This example solves the set of linear equations $Ax = b$ where

$$A = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ -2 & 0 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 & 2 & -3 \\ -1 & -1 & 0 & 0 & 0 & 6 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 15 \\ 12 \\ 18 \\ 3 \\ -6 \\ 0 \end{pmatrix}.$$

The non-zero elements of A and indexing information are read in by the program, as described in the document for F01BRF.

9.1 Program Text

```
*     F04AXF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX, NZMAX, LICN, LIRN
      PARAMETER        (NMAX=20,NZMAX=50,LICN=3*NZMAX,LIRN=3*NZMAX/2)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION RESID, U
      INTEGER          I, IFAIL, MTYPE, N, NZ
      LOGICAL          GROW, LBLOCK
*     .. Local Arrays ..
      DOUBLE PRECISION A(LICN), RHS(NMAX), W(NMAX)
      INTEGER          ICN(LICN), IDISP(10), IKEEP(NMAX,5), IRN(LIRN),
     +                 IW(NMAX,8)
      LOGICAL          ABORT(4)
*     .. External Subroutines ..
      EXTERNAL         F01BRF, F04AXF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04AXF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NZ
      WRITE (NOUT,*)
      IF (N.GT.0 .AND. N.LE.NMAX .AND. NZ.GT.0 .AND. NZ.LE.NZMAX) THEN
         READ (NIN,*) (A(I),IRN(I),ICN(I),I=1,NZ)
         U = 0.1D0
         LBLOCK = .TRUE.
         GROW = .TRUE.
         ABORT(1) = .TRUE.
         ABORT(2) = .TRUE.
         ABORT(3) = .FALSE.
         ABORT(4) = .TRUE.
         IFAIL = 110
*
*        Decomposition of sparse matrix
         CALL F01BRF(N,NZ,A,LICN,IRN,LIRN,ICN,U,IKEEP,IW,W,LBLOCK,GROW,
     +               ABORT,IDISP,IFAIL)
*
         IF (GROW) THEN
            WRITE (NOUT,*) 'On exit from F01BRF'
            WRITE (NOUT,99998) 'Value of W(1) = ', W(1)
         END IF
         READ (NIN,*) (RHS(I),I=1,N)
         MTYPE = 1
*
*        Approximate solution of sparse linear equations
         CALL F04AXF(N,A,LICN,ICN,IKEEP,RHS,W,MTYPE,IDISP,RESID)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'On exit from F04AXF'
         WRITE (NOUT,*) ' Solution'
```

```
         WRITE (NOUT,99997) (RHS(I),I=1,N)
      ELSE
         WRITE (NOUT,99999) 'N or NZ is out of range: N = ', N,
     +     '  NZ = ', NZ
      END IF
      STOP
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,A,F9.4)
99997 FORMAT (1X,F9.4)
      END
```

9.2 Program Data

```
F04AXF Example Program Data
  6 15
   5.0  1  1   2.0  2  2  -1.0  2  3   2.0  2  4   3.0  3  3
  -2.0  4  1   1.0  4  4   1.0  4  5  -1.0  5  1  -1.0  5  4
   2.0  5  5  -3.0  5  6  -1.0  6  1  -1.0  6  2   6.0  6  6
  15.0  12.0  18.0   3.0  -6.0   0.0
```

9.3 Program Results

```
 F04AXF Example Program Results

 On exit from F01BRF
 Value of W(1) =   18.0000

 On exit from F04AXF
  Solution
    3.0000
    3.0000
    6.0000
    6.0000
    3.0000
    1.0000
```