

NAG 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 $A^T x = b$, 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)
REAL (KIND=nag_wp) A(LICN), RHS(N), W(N), RESID
```

3 Description

To solve a system of real linear equations $Ax = b$ or $A^T x = b$, 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 Arguments

- 1: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 2: A(LICN) – REAL (KIND=nag_wp) array *Input*
On entry: the nonzero 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 nonzero elements of the factorization, as returned by F01BRF or F01BSF.
- 5: IKEEP($5 \times N$) – 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) – REAL (KIND=nag_wp) array *Input/Output*
On entry: the right-hand side vector b .
On exit: RHS is overwritten by the solution vector x .
- 7: W(N) – REAL (KIND=nag_wp) array *Workspace*
- 8: MTYPE – INTEGER *Input*
On entry: MTYPE specifies the task to be performed.
 MTYPE = 1
 Solve $Ax = b$.
 MTYPE \neq 1
 Solve $A^T x = b$.
- 9: IDISP(2) – INTEGER array *Communication Array*
On entry: the values returned in IDISP by F01BRF.
- 10: RESID – REAL (KIND=nag_wp) *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

If an error is detected in an input argument F04AXF will act as if a soft noisy exit has been requested (see Section 3.4.4 in How to Use the NAG Library and its Documentation).

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

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

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

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

8 Parallelism and Performance

F04AXF is not threaded in any implementation.

9 Further Comments

If the factorized form contains τ nonzeros (`IDISP(2) = τ`) 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 $A^T x = b$ (MTYPE \neq 1).

10 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 nonzero elements of A and indexing information are read in by the program, as described in the document for F01BRF.

10.1 Program Text

```

Program f04axfe

!      F04AXF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: f01brf, f04axf, nag_wp
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: resid, u
Integer                     :: i, ifail, licn, lirn, mtype, n, nz
Logical                     :: grow, lblock
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:), rhs(:), w(:)
Integer, Allocatable        :: icn(:), ikeep(:, :), irn(:), iw(:, :)
Integer                     :: idisp(10)
Logical                     :: abort(4)
!      .. Executable Statements ..
Write (nout,*) 'F04AXF Example Program Results'
Write (nout,*)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n, nz
licn = 3*nz
lirn = 3*nz/2
Allocate (a(licn),rhs(n),w(n),icn(licn),ikeep(n,5),irn(lirn),iw(n,8))
Read (nin,*)(a(i),irn(i),icn(i),i=1,nz)
u = 0.1E0_nag_wp
lblock = .True.
grow = .True.
abort(1) = .True.
abort(2) = .True.
abort(3) = .False.
abort(4) = .True.

!      ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
ifail = 0
!      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,99999) 'Value of W(1) = ', w(1)
End If
Read (nin,*) rhs(1:n)
mtype = 1

```

```

!      Approximate solution of sparse linear equations
      Call f04axf(n,a,licn,icn,ikeep,rhs,w,mtype,ideisp,resid)

      Write (nout,*)
      Write (nout,*) 'On exit from F04AXF'
      Write (nout,*) ' Solution'
      Write (nout,99998) rhs(1:n)

99999 Format (1X,A,F9.4)
99998 Format (1X,F9.4)
      End Program f04axfe

```

10.2 Program Data

F04AXF Example Program Data

```

6 15                                     : n, nz
  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                                     : a
                                                                    : rhs

```

10.3 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

```
