NAG Library Routine Document

F04MCF

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

F04MCF computes the approximate solution of a system of real linear equations with multiple right-hand sides, $AX = B$, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorized by F01MCF. Related systems may also be solved.

2 Specification

```
SUBROUTINE F04MCF (N, AL, LAL, D, NROW, IR, B, LDB, ISELCT, X, LDX, IFAIL)

INTEGER             N, LAL, NROW(N), IR, LDB, ISELCT, LDX, IFAIL
REAL (KIND=nag_wp) AL(LAL), D(*), B(LDB,IR), X(LDX,IR)
```

3 Description

The normal use of this routine is the solution of the systems $AX = B$, following a call of F01MCF to determine the Cholesky factorization $A = LDL^{\mathrm{T}}$ of the symmetric positive definite variable-bandwidth matrix A.

However, the routine may be used to solve any one of the following systems of linear algebraic equations:

1. $LDL^{\mathrm{T}}X = B$ (usual system),
2. $LDX = B$ (lower triangular system),
3. $DL^{\mathrm{T}}X = B$ (upper triangular system),
4. $LL^{\mathrm{T}}X = B$
5. $LX = B$ (unit lower triangular system),
6. $L^{\mathrm{T}}X = B$ (unit upper triangular system).

L denotes a unit lower triangular variable-bandwidth matrix of order n, D a diagonal matrix of order n, and B a set of right-hand sides.

The matrix L is represented by the elements lying within its **envelope**, i.e., between the first nonzero of each row and the diagonal (see Section 9 for an example). The width NROW(i) of the ith row is the number of elements between the first nonzero element and the element on the diagonal inclusive.

4 References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag

5 Parameters

1: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 1$.

2: AL(LAL) – REAL (KIND=nag_wp) array *Input*

On entry: the elements within the envelope of the lower triangular matrix L, taken in row by row order, as returned by F01MCF. The unit diagonal elements of L must be stored explicitly.

3: LAL – INTEGER *Input*

On entry: the dimension of the array AL as declared in the (sub)program from which F04MCF is called.

Constraint: $\mathrm{LAL} \geq \mathrm{NROW}(1) + \mathrm{NROW}(2) + \ldots + \mathrm{NROW}(n)$.

4: D($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array D must be at least 1 if $\mathrm{ISELCT} \geq 4$, and at least N otherwise.

On entry: the diagonal elements of the diagonal matrix D. D is not referenced if $\mathrm{ISELCT} \geq 4$.

5: NROW(N) – INTEGER array *Input*

On entry: $\mathrm{NROW}(i)$ must contain the width of row i of L, i.e., the number of elements between the first (leftmost) nonzero element and the element on the diagonal, inclusive.

Constraint: $1 \leq \mathrm{NROW}(i) \leq i$.

6: IR – INTEGER *Input*

On entry: r, the number of right-hand sides.

Constraint: $\mathrm{IR} \geq 1$.

7: B(LDB,IR) – REAL (KIND=nag_wp) array *Input*

On entry: the n by r right-hand side matrix B. See also Section 8.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04MCF is called.

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

9: ISELCT – INTEGER *Input*

On entry: must specify the type of system to be solved, as follows:

$\mathrm{ISELCT} = 1$
Solve $LDL^{\mathrm{T}}X = B$.

$\mathrm{ISELCT} = 2$
Solve $LDX = B$.

$\mathrm{ISELCT} = 3$
Solve $DL^{\mathrm{T}}X = B$.

$\mathrm{ISELCT} = 4$
Solve $LL^{\mathrm{T}}X = B$.

$\mathrm{ISELCT} = 5$
Solve $LX = B$.

$\mathrm{ISELCT} = 6$
Solve $L^{\mathrm{T}}X = B$.

Constraint: $\mathrm{ISELCT} = 1$, 2, 3, 4, 5 or 6.

10: X(LDX,IR) – REAL (KIND=nag_wp) array *Output*

On exit: the n by r solution matrix X. See also Section 8.

11: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F04MCF is called.

Constraint: LDX $\geq$ N.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators 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 or warnings detected by the routine:

IFAIL $= 1$

On entry, N < 1,
or for some i, NROW(i) < 1 or NROW(i) $> i$,
or LAL $<$ NROW(1) $+$ NROW(2) $+ \cdots +$ NROW(N).

IFAIL $= 2$

On entry, IR < 1,
or LDB $<$ N,
or LDX $<$ N.

IFAIL $= 3$

On entry, ISELCT < 1,
or ISELCT > 6.

IFAIL $= 4$

The diagonal matrix D is singular, i.e., at least one of the elements of D is zero. This can only occur if ISELCT ≤ 3.

IFAIL $= 5$

At least one of the diagonal elements of L is not equal to unity.

7 Accuracy

The usual backward error analysis of the solution of triangular system applies: each computed solution vector is exact for slightly perturbed matrices L and D, as appropriate (see pages 25–27 and 54–55 of Wilkinson and Reinsch (1971)).

8 Further Comments

The time taken by F04MCF is approximately proportional to pr, where $p = \text{NROW}(1) + \text{NROW}(2) + \cdots + \text{NROW}(n)$.

Unless otherwise stated in the Users' Note for your implementation, the routine may be called with the same actual array supplied for the parameters B and X, in which case the solution matrix will overwrite the right-hand side matrix. However this is not standard Fortran and may not work in all implementations.

9 Example

This example solves the system of equations $AX = B$, where

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 5 & 0 \\ 2 & 5 & 3 & 0 & 14 & 0 \\ 0 & 3 & 13 & 0 & 18 & 0 \\ 0 & 0 & 0 & 16 & 8 & 24 \\ 5 & 14 & 18 & 8 & 55 & 17 \\ 0 & 0 & 0 & 24 & 17 & 77 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6 & -10 \\ 15 & -21 \\ 11 & -3 \\ 0 & 24 \\ 51 & -39 \\ 46 & 67 \end{pmatrix}$$

Here A is symmetric and positive definite and must first be factorized by F01MCF.

9.1 Program Text

```
    Program f04mcfe

!     F04MCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f01mcf, f04mcf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, ir, iselct, k1, k2, lal,   &
                                          ldb, ldx, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), al(:), b(:,:), d(:), x(:,:)
      Integer, Allocatable             :: nrow(:)
!     .. Executable Statements ..
      Write (nout,*) 'F04MCF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, ir
      ldb = n
      ldx = n
      Allocate (b(ldb,ir),d(n),x(ldx,ir),nrow(n))
      Read (nin,*) nrow(1:n)
      lal = 0
      Do i = 1, n
        lal = lal + nrow(i)
      End Do
      Allocate (a(lal),al(lal))
      k2 = 0
      Do i = 1, n
        k1 = k2 + 1
        k2 = k2 + nrow(i)
        Read (nin,*) a(k1:k2)
      End Do
      Read (nin,*)(b(i,1:ir),i=1,n)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
```

```
      ifail = 0
      Call f01mcf(n,a,lal,nrow,al,d,ifail)

      iselct = 1

      ifail = 0
      Call f04mcf(n,al,lal,d,nrow,ir,b,ldb,iselct,x,ldx,ifail)

      Write (nout,*) ' Solution'
      Do i = 1, n
        Write (nout,99999) x(i,1:ir)
      End Do

99999 Format (1X,8F9.3)
    End Program f04mcfe
```

9.2 Program Data

```
F04MCF Example Program Data
  6  2                            : n, ir
  1  2  2  1  5  3                : vector NROW
   1.0
   2.0   5.0
   3.0  13.0
  16.0
   5.0  14.0  18.0   8.0  55.0
  24.0  17.0  77.0                : vector A
   6.0 -10.0
  15.0 -21.0
  11.0  -3.0
   0.0  24.0
  51.0 -39.0
  46.0  67.0                      : matrix B
```

9.3 Program Results

```
 F04MCF Example Program Results

  Solution
    -3.000    4.000
     2.000   -2.000
    -1.000    3.000
    -2.000    1.000
     1.000   -2.000
     1.000    1.000
```