

# NAG Fortran Library Routine Document

## F01MCF

**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

F01MCF computes the Cholesky factorization of a real symmetric positive-definite variable-bandwidth matrix.

### 2 Specification

```
SUBROUTINE F01MCF(N, A, LAL, NROW, AL, D, IFAIL)
INTEGER          N, LAL, NROW(N), IFAIL
real           A(LAL), AL(LAL), D(N)
```

### 3 Description

This routine determines the unit lower triangular matrix  $L$  and the diagonal matrix  $D$  in the Cholesky factorization  $A = LDL^T$  of a symmetric positive-definite variable-bandwidth matrix  $A$  of order  $n$ . (Such a matrix is sometimes called a 'sky-line' matrix.)

The matrix  $A$  is represented by the elements lying within the **envelope** of its lower triangular part, that is, between the first non-zero of each row and the diagonal (see Section 9 for an example). The **width**  $NROW(i)$  of the  $i$ th row is the number of elements between the first non-zero element and the element on the diagonal, inclusive. Although, of course, any matrix possesses an envelope as defined, this routine is primarily intended for the factorization of symmetric positive-definite matrices with an **average** bandwidth which is small compared with  $n$  (also see Section 8).

The method is based on the property that during Cholesky factorization there is no fill-in outside the envelope.

The determination of  $L$  and  $D$  is normally the first of two steps in the solution of the system of equations  $Ax = b$ . The remaining step, viz. the solution of  $LDL^T x = b$ , may be carried out using F04MCF.

### 4 References

Jennings A (1966) A compact storage scheme for the solution of symmetric linear simultaneous equations *Comput. J.* **9** 281–285

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  $A$ .

*Constraint:*  $N \geq 1$ .

2: A(LAL) – **real** array *Input*

*On entry:* the elements within the envelope of the lower triangle of the positive-definite symmetric matrix  $A$ , taken in row by row order. The following code assigns the matrix elements within the envelope to the correct elements of the array:

```

      K = 0
      DO 20 I = 1, N
        DO 10 J = I-NROW(I)+1, I
          K = K + 1
          A(K) = matrix (I,J)
        10 CONTINUE
      20 CONTINUE

```

See also Section 8.

- 3: LAL – INTEGER *Input/Output*  
*On entry:* the smaller of the dimensions of the arrays A and AL as declared in the calling (sub)program from which F01MCF is called.  
*On exit:* LAL is used as internal workspace prior to being restored and hence is unchanged.  
*Constraint:*  $LAL \geq NROW(1) + NROW(2) + \dots + NROW(n)$ .
- 4: NROW(N) – INTEGER array *Input*  
*On entry:*  $NROW(i)$  must contain the width of row  $i$  of the matrix  $A$ , i.e., the number of elements between the first (leftmost) non-zero element and the element on the diagonal, inclusive.  
*Constraint:*  $1 \leq NROW(i) \leq i$ , for  $i = 1, 2, \dots, n$ .
- 5: AL(LAL) – *real* array *Output*  
*On exit:* the elements within the envelope of the lower triangular matrix  $L$ , taken in row by row order. The envelope of  $L$  is identical to that of the lower triangle of  $A$ . The unit diagonal elements of  $L$  are stored explicitly. See also Section 8.
- 6: D(N) – *real* array *Output*  
*On exit:* the diagonal elements of the diagonal matrix  $D$ . Note that the determinant of  $A$  is equal to the product of these diagonal elements. If the value of the determinant is required it should not be determined by forming the product explicitly, because of the possibility of overflow or underflow. The logarithm of the determinant may safely be formed from the sum of the logarithms of the diagonal elements.
- 7: IFAIL – INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.  
*On exit:* IFAIL = 0 unless the routine detects an error (see Section 6).  
 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, for users 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.**

## 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  $LAL < NROW(1) + NROW(2) + \dots + NROW(N)$ .



## 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.

```

*      F01MCF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
INTEGER          NMAX, LALMAX
PARAMETER       (NMAX=8,LALMAX=36)
INTEGER          NIN, NOUT
PARAMETER       (NIN=5,NOUT=6)
*      .. Local Scalars ..
INTEGER          I, IFAIL, K, K1, K2, LAL, N
*      .. Local Arrays ..
real           A(LALMAX), AL(LALMAX), D(NMAX)
INTEGER          NROW(NMAX)
*      .. External Subroutines ..
EXTERNAL        F01MCF
*      .. Executable Statements ..
WRITE (NOUT,*) 'F01MCF Example Program Results'
*      Skip heading in data file
READ (NIN,*)
READ (NIN,*) N
IF (N.GT.0 .AND. N.LE.NMAX) THEN
  READ (NIN,*) (NROW(I),I=1,N)
  K2 = 0
  DO 20 I = 1, N
    K1 = K2 + 1
    K2 = K2 + NROW(I)
    READ (NIN,*) (A(K),K=K1,K2)
20  CONTINUE
  LAL = K2
  IF (LAL.LE.LALMAX) THEN
    IFAIL = 1
*
    CALL F01MCF(N,A,LAL,NROW,AL,D,IFAIL)
*
    WRITE (NOUT,*)
    IF (IFAIL.EQ.0) THEN
      WRITE (NOUT,*)
      +      ' I      D(I)   Row I of unit lower triangle'
      WRITE (NOUT,*)
      K2 = 0
      DO 40 I = 1, N
        K1 = K2 + 1
        K2 = K2 + NROW(I)
        WRITE (NOUT,99999) I, D(I), (AL(K),K=K1,K2)
40     CONTINUE
      ELSE
        WRITE (NOUT,99998) 'F01MCF fails with IFAIL =', IFAIL
      END IF
    END IF
  END IF
  STOP
*
99999 FORMAT (1X,I3,7F8.3)
99998 FORMAT (1X,A,I3)
END

```

## 9.2 Program Data

F01MCF Example Program Data

```
6
1 2 2 1 5 3
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
```

## 9.3 Program Results

F01MCF Example Program Results

I	D(I)	Row I of unit lower triangle				
1	1.000	1.000				
2	1.000	2.000	1.000			
3	4.000	3.000	1.000			
4	16.000	1.000				
5	1.000	5.000	4.000	1.500	0.500	1.000
6	16.000	1.500	5.000	1.000		

---