NAG Library Routine Document

F01ADF

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

F01ADF calculates the approximate inverse of a real symmetric positive definite matrix, using a Cholesky factorization.

2 Specification

```
SUBROUTINE F01ADF (N, A, LDA, IFAIL)
INTEGER            N, LDA, IFAIL
REAL (KIND=nag_wp) A(LDA,*)
```

3 Description

To compute the inverse X of a real symmetric positive definite matrix A, F01ADF first computes a Cholesky factorization of A as $A = LL^{\mathrm{T}}$, where L is lower triangular. It then computes L^{-1} and finally forms X as the product $L^{-\mathrm{T}}L^{-1}$.

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 A.

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

2: $\mathrm{A}(\mathrm{LDA},*)$ – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1,\mathrm{N})$.

On entry: the upper triangle of the n by n positive definite symmetric matrix A. The elements of the array below the diagonal need not be set.

On exit: the lower triangle of the inverse matrix X is stored in the elements of the array below the diagonal, in rows 2 to $n+1$; x_{ij} is stored in $\mathrm{A}(i+1,j)$ for $i \geq j$. The upper triangle of the original matrix is unchanged.

3: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F01ADF is called.

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

4: 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: $\mathrm{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

The matrix A is not positive definite, possibly due to rounding errors.

$\mathrm{IFAIL} = 2$

On entry, $\mathrm{N} < 0$,
or $\mathrm{LDA} < \mathrm{N} + 1$.

$\mathrm{IFAIL} = -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

$\mathrm{IFAIL} = -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

$\mathrm{IFAIL} = -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The accuracy of the computed inverse depends on the conditioning of the original matrix. For a detailed error analysis see page 39 of Wilkinson and Reinsch (1971).

8 Parallelism and Performance

F01ADF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F01ADF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken by F01ADF is approximately proportional to n^3. F01ADF calls routines F07FDF (DPOTRF) and F07FJF (DPOTRI) from LAPACK.

10 Example

This example finds the inverse of the 4 by 4 matrix:

$$\begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix}.$$

10.1 Program Text

```
    Program f01adfe

!     F01ADF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f01adf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F01ADF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n + 1
      Allocate (a(lda,n))
      Read (nin,*)(a(i,1:n),i=1,n)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f01adf(n,a,lda,ifail)

!     Print the result matrix A
      Call x04caf('L','B',lda,n,a,lda,'Lower triangle of inverse',ifail)

    End Program f01adfe
```

10.2 Program Data

```
F01ADF Example Program Data
   4                   : n
   5.   7.   6.   5.
   7.  10.   8.   7.
   6.   8.  10.   9.
   5.   7.   9.  10.   : a
```

10.3 Program Results

```
 F01ADF Example Program Results

 Lower triangle of inverse
            1          2          3          4
 1
 2    68.0000
 3   -41.0000    25.0000
 4   -17.0000    10.0000     5.0000
 5    10.0000    -6.0000    -3.0000     2.0000
```