# NAG Library Function Document

# nag_dpotrf (f07fdc)

## 1 Purpose

nag_dpotrf (f07fdc) computes the Cholesky factorization of a real symmetric positive definite matrix.

## 2 Specification

```
#include <nag.h>
#include <nagf07.h>
void nag_dpotrf (Nag_OrderType order, Nag_UploType uplo, Integer n,
     double a[], Integer pda, NagError *fail)
```

## 3 Description

nag_dpotrf (f07fdc) forms the Cholesky factorization of a real symmetric positive definite matrix $A$ either as $A = U^{\mathrm{T}}U$ if **uplo** = Nag_Upper or $A = LL^{\mathrm{T}}$ if **uplo** = Nag_Lower, where $U$ is an upper triangular matrix and $L$ is lower triangular.

## 4 References

Demmel J W (1989) On floating-point errors in Cholesky *LAPACK Working Note No. 14* University of Tennessee, Knoxville

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

## 5 Arguments

1: **order** – Nag_OrderType *Input*

*On entry*: the **order** argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order** = Nag_RowMajor. See Section 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

*Constraint*: **order** = Nag_RowMajor or Nag_ColMajor.

2: **uplo** – Nag_UploType *Input*

*On entry*: specifies whether the upper or lower triangular part of $A$ is stored and how $A$ is to be factorized.

**uplo** = Nag_Upper
The upper triangular part of $A$ is stored and $A$ is factorized as $U^{\mathrm{T}}U$, where $U$ is upper triangular.

**uplo** = Nag_Lower
The lower triangular part of $A$ is stored and $A$ is factorized as $LL^{\mathrm{T}}$, where $L$ is lower triangular.

*Constraint*: **uplo** = Nag_Upper or Nag_Lower.

3: **n** – Integer *Input*

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

*Constraint*: $\mathbf{n} \geq 0$.

4: **a**[$dim$] – double *Input/Output*

**Note**: the dimension, *dim*, of the array **a** must be at least $\max(1, \mathbf{pda} \times \mathbf{n})$.

*On entry*: the $n$ by $n$ symmetric positive definite matrix $A$.

If **order** = 'Nag_ColMajor', $A_{ij}$ is stored in $\mathbf{a}[(j-1) \times \mathbf{pda} + i - 1]$.

If **order** = 'Nag_RowMajor', $A_{ij}$ is stored in $\mathbf{a}[(i-1) \times \mathbf{pda} + j - 1]$.

If **uplo** = 'Nag_Upper', the upper triangular part of $A$ must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'Nag_Lower', the lower triangular part of $A$ must be stored and the elements of the array above the diagonal are not referenced.

*On exit*: the upper or lower triangle of $A$ is overwritten by the Cholesky factor $U$ or $L$ as specified by **uplo**.

5: **pda** – Integer *Input*

*On entry*: the stride separating row or column elements (depending on the value of **order**) of the matrix $A$ in the array **a**.

*Constraint*: $\mathbf{pda} \geq \max(1, \mathbf{n})$.

6: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

**NE_BAD_PARAM**

On entry, argument ⟨*value*⟩ had an illegal value.

**NE_INT**

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

On entry, $\mathbf{pda} = \langle value \rangle$.
Constraint: $\mathbf{pda} > 0$.

**NE_INT_2**

On entry, $\mathbf{pda} = \langle value \rangle$ and $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{pda} \geq \max(1, \mathbf{n})$.

**NE_INTERNAL_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

**NE_POS_DEF**

The leading minor of order ⟨*value*⟩ is not positive definite and the factorization could not be completed. Hence $A$ itself is not positive definite. This may indicate an error in forming the matrix $A$. To factorize a symmetric matrix which is not positive definite, call nag_dsytrf (f07mdc) instead.

## 7 Accuracy

If **uplo** = Nag_Upper, the computed factor $U$ is the exact factor of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon\left|U^{\mathrm{T}}\right||U|,$$

$c(n)$ is a modest linear function of $n$, and $\epsilon$ is the ***machine precision***. If **uplo** = Nag_Lower, a similar statement holds for the computed factor $L$. It follows that $\left|e_{ij}\right| \leq c(n)\epsilon\sqrt{a_{ii}a_{jj}}$.

## 8 Parallelism and Performance

nag_dpotrf (f07fdc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

nag_dpotrf (f07fdc) 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 Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

The total number of floating-point operations is approximately $\frac{1}{3}n^3$.

A call to nag_dpotrf (f07fdc) may be followed by calls to the functions:

nag_dpotrs (f07fec) to solve $AX = B$;

nag_dpocon (f07fgc) to estimate the condition number of $A$;

nag_dpotri (f07fjc) to compute the inverse of $A$.

The complex analogue of this function is nag_zpotrf (f07frc).

## 10 Example

This example computes the Cholesky factorization of the matrix $A$, where

$$A = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.18 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.18 & 0.34 & 1.18 \end{pmatrix}.$$

### 10.1 Program Text

```
/* nag_dpotrf (f07fdc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>
#include <nagx04.h>

int main(void)
{
  /* Scalars */
  Integer        i, j, n, pda;
  Integer        exit_status = 0;
  Nag_UploType   uplo;
  Nag_MatrixType matrix;
```

```
  NagError       fail;
  Nag_OrderType  order;
  /* Arrays */
  char           nag_enum_arg[40];
  double         *a = 0;

#ifdef NAG_COLUMN_MAJOR
#define A(I, J) a[(J-1)*pda + I - 1]
  order = Nag_ColMajor;
#else
#define A(I, J) a[(I-1)*pda + J - 1]
  order = Nag_RowMajor;
#endif

  INIT_FAIL(fail);

  printf("nag_dpotrf (f07fdc) Example Program Results\n\n");

  /* Skip heading in data file */
  scanf("%*[^\n] ");
  scanf("%ld%*[^\n] ", &n);
#ifdef NAG_COLUMN_MAJOR
  pda = n;
#else
  pda = n;
#endif
  /* Allocate memory */
  if (!(a = NAG_ALLOC(n * n, double)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }


  /* Read A from data file */
  scanf(" %39s%*[^\n] ", nag_enum_arg);
  /* nag_enum_name_to_value (x04nac).
   * Converts NAG enum member name to value
   */
  uplo = (Nag_UploType) nag_enum_name_to_value(nag_enum_arg);

  if (uplo == Nag_Upper)
    {
      matrix = Nag_UpperMatrix;
      for (i = 1; i <= n; ++i)
        {
          for (j = i; j <= n; ++j)
            scanf("%lf", &A(i, j));
        }
      scanf("%*[^\n] ");
    }
  else
    {
      matrix = Nag_LowerMatrix;
      for (i = 1; i <= n; ++i)
        {
          for (j = 1; j <= i; ++j)
            scanf("%lf", &A(i, j));
        }
      scanf("%*[^\n] ");
    }

  /* Factorize A */
  /* nag_dpotrf (f07fdc).
   * Cholesky factorization of real symmetric
   * positive-definite matrix
   */
  nag_dpotrf(order, uplo, n, a, pda, &fail);
  if (fail.code != NE_NOERROR)
    {
```

```
      printf("Error from nag_dpotrf (f07fdc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }

  /* Print factor */
  /* nag_gen_real_mat_print (x04cac).
   * Print real general matrix (easy-to-use)
   */
  fflush(stdout);
  nag_gen_real_mat_print(order, matrix, Nag_NonUnitDiag, n, n, a, pda,
                         "Factor", 0, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_gen_real_mat_print (x04cac).\n%s\n",
              fail.message);
      exit_status = 1;
      goto END;
    }

 END:
  NAG_FREE(a);

  return exit_status;
}
```

## 10.2 Program Data

```
nag_dpotrf (f07fdc) Example Program Data
  4                           :Value of n
  Nag_Lower                   :Value of uplo
  4.16
 -3.12   5.03
  0.56  -0.83   0.76
 -0.10   1.18   0.34   1.18   :End of matrix A
```

## 10.3 Program Results

```
nag_dpotrf (f07fdc) Example Program Results

 Factor
            1          2          3          4
 1      2.0396
 2     -1.5297     1.6401
 3      0.2746    -0.2500     0.7887
 4     -0.0490     0.6737     0.6617     0.5347
```