NAG Library Function Document

nag_real_cholesky (f03aec)

1 Purpose

nag_real_cholesky (f03aec) computes a Cholesky factorization of a real symmetric positive definite matrix, and evaluates the determinant.

2 Specification

```
#include <nag.h>
#include <nagf03.h>
void nag_real_cholesky (Integer n, double a[], Integer tda, double p[],
     double *detf, Integer *dete, NagError *fail)
```

3 Description

nag_real_cholesky (f03aec) computes the Cholesky factorization of a real symmetric positive definite matrix $A = LL^{\mathrm{T}}$ where L is lower triangular. The determinant is the product of the squares of the diagonal elements of L.

4 References

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

5 Arguments

1: **n** – Integer *Input*

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

Constraint: $\mathbf{n} \geq 1$.

2: **a**[**n** × **tda**] – double *Input/Output*

Note: the (i, j)th element of the matrix A is stored in $\mathbf{a}[(i-1) \times \mathbf{tda} + j - 1]$.

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 sub-diagonal elements of the lower triangular matrix L. The upper triangle of A is unchanged.

3: **tda** – Integer *Input*

On entry: the stride separating matrix column elements in the array **a**.

Constraint: $\mathbf{tda} \geq \mathbf{n}$.

4: **p**[**n**] – double *Output*

On exit: the reciprocals of the diagonal elements of L.

5: **detf** – double * *Output*
6: **dete** – Integer * *Output*

On exit: the determinant of A is given by **detf** $\times 2.0^{\mathbf{dete}}$. It is given in this form to avoid overflow or underflow.

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

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

6 Error Indicators and Warnings

NE_2_INT_ARG_LT

On entry, **tda** $= \langle value \rangle$ while **n** $= \langle value \rangle$. These arguments must satisfy **tda** $\geq$ **n**.

NE_INT_ARG_LT

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 1.

NE_NOT_POS_DEF

The matrix is not positive definite, possibly due to rounding errors. The factorization could not be completed. **detf** and **dete** are set to zero.

7 Accuracy

The accuracy of the determinant depends on the conditioning of the original matrix. For a detailed error analysis see page 25 of Wilkinson and Reinsch (1971).

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by nag_real_cholesky (f03aec) is approximately proportional to n^3.

10 Example

To compute a Cholesky factorization and calculate the determinant of the real symmetric positive definite matrix

$$\begin{pmatrix} 6 & 7 & 6 & 5 \\ 7 & 11 & 8 & 7 \\ 6 & 8 & 11 & 9 \\ 5 & 7 & 9 & 11 \end{pmatrix}.$$

10.1 Program Text

```
/* nag_real_cholesky (f03aec) Example Program.
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 * Mark 8 revised, 2004.
 */

#include <nag.h>
#include <math.h>
```

```
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagf03.h>

#define A(I, J) a[(I) *tda + J]

int main(void)
{
  Integer  dete, exit_status = 0, i, j, n, tda;
  NagError fail;
  double   *a = 0, determ, detf, *p = 0;

  INIT_FAIL(fail);

  printf("nag_real_cholesky (f03aec) Example Program Results\n");
  /* Skip heading in data file */
  scanf("%*[^\n]");
  scanf("%" NAG_IFMT "\n", &n);
  if (n >= 1)
    {
      if (!(a = NAG_ALLOC(n*n, double)) ||
          !(p = NAG_ALLOC(n, double)))
        {
          printf("Allocation failure\n");
          exit_status = -1;
          goto END;
        }
      tda = n;
    }
  else
    {
      printf("Invalid n.\n");
      exit_status = 1;
      return exit_status;
    }
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      scanf("%lf", &A(i, j));
  /* nag_real_cholesky (f03aec).
   * LL^T factorization and determinant of real symmetric
   * positive-definite matrix
   */
  nag_real_cholesky(n, a, tda, p, &detf, &dete, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_real_cholesky (f03aec).\n%s\n",
              fail.message);
      exit_status = 1;
      goto END;
    }
  printf("Array A after factorization\n");
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      printf("%9.4f%s", A(i, j), (j%8 == 7 || j == n-1)?"\n":" ");
  printf("\nArray p\n");
  for (i = 0; i < n; i++)
    printf("%9.4f%s", p[i], (i%8 == 7 || i == n-1)?"\n":" ");
  printf("\ndetf = %9.4f           dete = %2" NAG_IFMT "\n\n", detf, dete);
  determ = detf*pow(2.0, (double) dete);
  printf("Value of determinant = %9.4f\n", determ);
 END:
  NAG_FREE(a);
  NAG_FREE(p);
  return exit_status;
}
```

10.2 Program Data

```
nag_real_cholesky (f03aec) Example Program Data
  4
    6    7    6    5
    7   11    8    7
    6    8   11    9
    5    7    9   11
```

10.3 Program Results

```
nag_real_cholesky (f03aec) Example Program Results
Array A after factorization
   6.0000    7.0000    6.0000    5.0000
   2.8577   11.0000    8.0000    7.0000
   2.4495    0.5941   11.0000    9.0000
   2.0412    0.6931    1.6645   11.0000

Array p
   0.4082    0.5941    0.4639    0.5283

detf =    0.0691           dete = 12

Value of determinant =  283.0000
```