NAG Library Function Document

nag_det_real_sym (f03bfc)

1 Purpose

nag_det_real_sym (f03bfc) computes the determinant of a real n by n symmetric positive definite matrix A. nag_dpotrf (f07fdc) must be called first to supply the symmetric matrix A in Cholesky factorized form. The storage (upper or lower triangular) used by nag_dpotrf (f07fdc) is not relevant to nag_det_real_sym (f03bfc) since only the diagonal elements of the factorized A are referenced.

2 Specification

```
#include <nag.h>
#include <nagf03.h>
void nag_det_real_sym (Nag_OrderType order, Integer n, const double a[],
     Integer pda, double *d, Integer *id, NagError *fail)
```

3 Description

nag_det_real_sym (f03bfc) computes the determinant of a real n by n symmetric positive definite matrix A that has been factorized as $A = U^{\mathrm{T}}U$, where U is upper triangular, or $A = LL^{\mathrm{T}}$, where L is lower triangular. The determinant is the product of the squares of the diagonal elements of U or L. The Cholesky factorized form of the matrix must be supplied; this is returned by a call to nag_dpotrf (f07fdc).

4 References

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

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 2.3.1.3 in How to Use the NAG Library and its Documentation for a more detailed explanation of the use of this argument.

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

2: **n** – Integer *Input*

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

Constraint: **n** > 0.

3: **a**$[dim]$ – const double *Input*

Note: the dimension, *dim*, of the array **a** must be at least **pda** $\times$ **n**.

The (i, j)th element of the Cholesky factorization of the matrix A is stored in

a$[(j-1) \times$ **pda** $+ i - 1]$ when **order** = Nag_ColMajor;
a$[(i-1) \times$ **pda** $+ j - 1]$ when **order** = Nag_RowMajor.

On entry: the lower or upper triangle of the Cholesky factorized form of the n by n positive definite symmetric matrix A. Only the diagonal elements are referenced.

4: **pda** – Integer *Input*

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

Constraint: **pda** $\geq$ **n**.

5: **d** – double * *Output*

6: **id** – Integer * *Output*

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

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

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

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

NE_INT_2

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

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.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_MAT_NOT_POS_DEF

The matrix A is not positive definite.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

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

nag_det_real_sym (f03bfc) is not threaded in any implementation.

9 Further Comments

The time taken by nag_det_real_sym (f03bfc) is approximately proportional to n.

10 Example

This example computes a Cholesky factorization and calculates 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_det_real_sym (f03bfc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */
#include <math.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf03.h>
#include <nagf07.h>
#include <nagx04.h>

int main(void)
{
  /* Scalars */
  Integer exit_status = 0;
  Integer i, id, j, n, pda;
  double d;
  /* Arrays */
  char nag_enum_arg[40];
  double *a = 0;
  /* NAG types */
  NagError fail;
  Nag_UploType uplo;
  Nag_OrderType order;
  Nag_MatrixType matrix;
  Nag_DiagType diag = Nag_NonUnitDiag;

  printf("nag_det_real_sym (f03bfc) Example Program Results\n\n");

  /* Skip heading in data file  */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

#ifdef _WIN32
  scanf_s("%" NAG_IFMT "%*[^\n]", &n);
#else
  scanf("%" NAG_IFMT "%*[^\n]", &n);
#endif
  pda = n;
```

```
  if (!(a = NAG_ALLOC(n * n, double)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* Define matrix element A_ij in terms of elements of array a[k] */
#ifdef NAG_COLUMN_MAJOR
  order = Nag_ColMajor;
#define A(I, J) a[(J-1)*pda+(I-1)]
#else
  order = Nag_RowMajor;
#define A(J, I) a[(J-1)*pda+(I-1)]
#endif
  for (i = 1; i <= n; i++)
    for (j = 1; j <= n; j++)
#ifdef _WIN32
      scanf_s("%lf", &A(i, j));
#else
      scanf("%lf", &A(i, j));
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

#ifdef _WIN32
  scanf_s("%39s %*[^\n] ", nag_enum_arg, (unsigned)_countof(nag_enum_arg));
#else
  scanf("%39s %*[^\n] ", nag_enum_arg);
#endif
  uplo = (Nag_UploType) nag_enum_name_to_value(nag_enum_arg);
  if (uplo == Nag_Lower) {
    matrix = Nag_LowerMatrix;
  }
  else {
    matrix = Nag_UpperMatrix;
  }

  INIT_FAIL(fail);

  /* 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("%s\n", fail.message);
    exit_status = 1;
    goto END;
  }

  /* nag_gen_real_mat_print (x04cac)
   * Print real general matrix (easy-to-use)
   */
  fflush(stdout);
  nag_gen_real_mat_print(order, matrix, diag, n, n, a, pda,
                         "Array A after factorization", NULL, &fail);
  if (fail.code != NE_NOERROR) {
    printf("%s\n", fail.message);
    exit_status = 2;
    goto END;
  }

  /* nag_det_real_sym (f03bfc)
   * determinant of factorized real symmetric positive definite matrix
   */
  nag_det_real_sym(order, n, a, pda, &d, &id, &fail);
  if (fail.code != NE_NOERROR) {
    printf("%s\n", fail.message);
```

```
    exit_status = 3;
    goto END;
  }

  printf("\nd = %12.5f  id = %12" NAG_IFMT "\n", d, id);
  printf("Value of determinant = %12.5e\n", d * pow(2.0, id));

END:
  NAG_FREE(a);

  return exit_status;
}
```

10.2 Program Data

```
nag_det_real_sym (f03bfc) Example Program Data
 4                      : n
 6     7     6     5
 7    11     8     7
 6     8    11     9
 5     7     9    11   : a
 Nag_Lower             : uplo
```

10.3 Program Results

```
nag_det_real_sym (f03bfc) Example Program Results

 Array A after factorization
            1          2          3          4
 1      2.4495
 2      2.8577     1.6833
 3      2.4495     0.5941     2.1557
 4      2.0412     0.6931     1.6645     1.8927

d =      0.06909  id =           12
Value of determinant =  2.83000e+02
```