NAG Library Function Document

nag_dgetri (f07ajc)

1 Purpose
nag_dgetri (f07ajc) computes the inverse of a real matrix \( A \), where \( A \) has been factorized by nag_dgetrf (f07adc).

2 Specification

```c
#include <nag.h>
#include <nagf07.h>

void nag_dgetri (Nag_OrderType order, Integer n, double a[], Integer pda,
                 const Integer ipiv[], NagError *fail)
```

3 Description

nag_dgetri (f07ajc) is used to compute the inverse of a real matrix \( A \), the function must be preceded by a call to nag_dgetrf (f07adc), which computes the \( LU \) factorization of \( A \) as \( A = PLU \). The inverse of \( A \) is computed by forming \( U^{-1} \) and then solving the equation \( XPL = U^{-1} \) for \( X \).

4 References


5 Arguments

1: \( \text{order} \) \hspace{1cm} \text{Nag_OrderType} \hspace{1cm} \text{Input} \\
   \hspace{0.5cm} \text{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.} \\
   \hspace{0.5cm} \text{Constraint: order = Nag_RowMajor or Nag_ColMajor.}

2: \( \text{n} \) \hspace{1cm} \text{Integer} \hspace{1cm} \text{Input} \\
   \hspace{0.5cm} \text{On entry: n, the order of the matrix A.} \\
   \hspace{0.5cm} \text{Constraint: n \geq 0.}

3: \( \text{a[dim]} \) \hspace{1cm} \text{double} \hspace{1cm} \text{Input/Output} \\
   \hspace{0.5cm} \text{Note: the dimension, dim, of the array a must be at least max(1, pda \times n).} \\
   \hspace{0.5cm} \text{The (i,j)th element of the matrix A is stored in} \\
   \hspace{1cm} a[(j-1) \times pda + i - 1] \text{ when order = Nag_ColMajor; } \\
   \hspace{1cm} a[(i-1) \times pda + j - 1] \text{ when order = Nag_RowMajor.} \\
   \hspace{0.5cm} \text{On entry: the LU factorization of A, as returned by nag_dgetrf (f07adc).} \\
   \hspace{0.5cm} \text{On exit: the factorization is overwritten by the n by n matrix A^{-1}.}
4: \textbf{pda} -- Integer \hspace{1cm} \textit{Input}

\textit{On entry:} the stride separating row or column elements (depending on the value of \textbf{order}) in the array \textbf{a}.

\textit{Constraint:} \textbf{pda} \geq \max(1, n).

5: \textbf{ipiv}[$\text{dim}$] -- const Integer \hspace{1cm} \textit{Input}

\textit{Note:} the dimension, \textit{dim}, of the array \textbf{ipiv} must be at least \max(1, n).

\textit{On entry:} the pivot indices, as returned by nag_dgetrf (f07adc).

6: \textbf{fail} -- NagError * \hspace{1cm} \textit{Input/Output}

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

\section{Error Indicators and Warnings}

\textbf{NE_ALLOC_FAIL}

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

\textbf{NE_BAD_PARAM}

On entry, argument \textit{value} had an illegal value.

\textbf{NE_INT}

On entry, \textbf{n} = \textit{value}.

\textit{Constraint:} \textbf{n} \geq 0.

On entry, \textbf{pda} = \textit{value}.

\textit{Constraint:} \textbf{pda} > 0.

\textbf{NE_INT_2}

On entry, \textbf{pda} = \textit{value} and \textbf{n} = \textit{value}.

\textit{Constraint:} \textbf{pda} \geq \max(1, n).

\textbf{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 3.6.6 in the Essential Introduction for further information.

\textbf{NE_NO_LICENCE}

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

See Section 3.6.5 in the Essential Introduction for further information.

\textbf{NE_SINGULAR}

Element \textit{value} of the diagonal is zero. \textbf{U} is singular, and the inverse of \textbf{A} cannot be computed.

\section{Accuracy}

The computed inverse \textbf{X} satisfies a bound of the form:

\[|\textbf{X} \times \textbf{A} - \textbf{I}| \leq c(n)\epsilon|\textbf{X}||\textbf{P}| |\textbf{L}| |\textbf{U}|,

where \(c(n)\) is a modest linear function of \(n\), and \(\epsilon\) is the \textit{machine precision}.
Note that a similar bound for $|AX - I|$ cannot be guaranteed, although it is almost always satisfied. See Du Croz and Higham (1992).

8 Parallelism and Performance

nag_dgetri (f07ajc) is not threaded by NAG in any implementation.

nag_dgetri (f07ajc) 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 function. Please also 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 $\frac{4}{3} n^3$.

The complex analogue of this function is nag_zgetri (f07awc).

10 Example

This example computes the inverse of the matrix $A$, where

$$A = \begin{pmatrix} 1.80 & 2.88 & 2.05 & -0.89 \\ 5.25 & -2.95 & -0.95 & -3.80 \\ 1.58 & -2.69 & -2.90 & -1.04 \\ -1.11 & -0.66 & -0.59 & 0.80 \end{pmatrix}. $$

Here $A$ is nonsymmetric and must first be factorized by nag_dgetrf (f07adc).

10.1 Program Text

/* nag_dgetri (f07ajc) Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 7, 2001. */
/* Mark 7b revised, 2004. */

#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;
NagError fail;
Nag_OrderType order;
/* Arrays */
double *a = 0;
Integer *ipiv = 0;
#ifdef NAG_LOAD_FP
/* The following line is needed to force the Microsoft linker to load floating point support */
float force>Loading_of_ms_float_support = 0;
#endif /* NAG_LOAD_FP */
#ifdef NAG_COLUMN_MAJOR
#define A(I, J) a[(J-1)*pda + I - 1]
#else
#define A(I, J) a[(I-1)*pda + J - 1]
#endif

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

INIT_FAIL(fail);
printf("nag_dgetri (f07ajc) Example Program Results\n\n");

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

/* Allocate memory */
if (!(a = NAG_ALLOC(n * n, double)) ||
!(ipiv = NAG_ALLOC(n, Integer)))
{
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
}

/* Read A from data file */
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
}

/* Factorize A */
/* nag_dgetrf (f07adc).
 * LU factorization of real m by n matrix
 */
nag_dgetrf(order, n, n, a, pda, ipiv, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dgetrf (f07adc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Compute inverse of A */
/* nag_dgetri (f07ajc).
 * Inverse of real matrix, matrix already factorized by
 * nag_dgetrf (f07adc)
 */
nag_dgetri(order, n, a, pda, ipiv, &fail);
if (fail.code != NE_NOERROR)
printf("Error from nag_dgetri (f07ajc).\n%s\n", fail.message);
exit_status = 1;
goto END;
}

/* Print inverse */
/* nag_gen_real_mat_print (x04cac).
 * Print real general matrix (easy-to-use)
 */
fflush(stdout);
nag_gen_real_mat_print(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, n, a,
pda, "Inverse", 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);
NAG_FREE(ipiv);
return exit_status;

10.2 Program Data

nag_dgetri (f07ajc) Example Program Data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.80</td>
<td>2.88</td>
<td>2.05</td>
<td>-0.89</td>
</tr>
<tr>
<td>5.25</td>
<td>-2.95</td>
<td>-0.95</td>
<td>-3.80</td>
</tr>
<tr>
<td>1.58</td>
<td>-2.69</td>
<td>-2.90</td>
<td>-1.04</td>
</tr>
<tr>
<td>-1.11</td>
<td>-0.66</td>
<td>-0.59</td>
<td>0.80</td>
</tr>
</tbody>
</table>

:Value of N

:End of matrix A

10.3 Program Results

nag_dgetri (f07ajc) Example Program Results

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7720</td>
<td>0.5757</td>
<td>0.0843</td>
<td>4.8155</td>
</tr>
<tr>
<td>-0.1175</td>
<td>-0.4456</td>
<td>0.4114</td>
<td>-1.7126</td>
</tr>
<tr>
<td>0.1799</td>
<td>0.4527</td>
<td>-0.6676</td>
<td>1.4824</td>
</tr>
<tr>
<td>2.4944</td>
<td>0.7650</td>
<td>-0.0360</td>
<td>7.6119</td>
</tr>
</tbody>
</table>