NAG Library Function Document

nag_zpptri (f07gwc)

1 Purpose

nag_zpptri (f07gwc) computes the inverse of a complex Hermitian positive definite matrix $A$, where $A$ has been factorized by nag_zpptrf (f07grc), using packed storage.

2 Specification

```c
#include <nag.h>
#include <nagf07.h>
void nag_zpptri (Nag_OrderType order, Nag_UploType uplo, Integer n,
                Complex ap[], NagError *fail)
```

3 Description

nag_zpptri (f07gwc) is used to compute the inverse of a complex Hermitian positive definite matrix $A$, the function must be preceded by a call to nag_zpptrf (f07grc), which computes the Cholesky factorization of $A$, using packed storage.

If $\text{uplo} = \text{Nag_Upper}$, $A = U^H U$ and $A^{-1}$ is computed by first inverting $U$ and then forming $(U^{-1})U^{-H}$.

If $\text{uplo} = \text{Nag_Lower}$, $A = LL^H$ and $A^{-1}$ is computed by first inverting $L$ and then forming $L^{-H}(L^{-1})$.

4 References


5 Arguments

1: \textbf{order} – Nag_OrderType

\textit{Input}

On entry: the \textit{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 \textit{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: \textit{order} = Nag_RowMajor or Nag_ColMajor.

2: \textbf{uplo} – Nag_UploType

\textit{Input}

On entry: specifies how $A$ has been factorized.

\textit{uplo} = Nag_Upper

$A = U^H U$, where $U$ is upper triangular.

\textit{uplo} = Nag_Lower

$A = LL^H$, where $L$ is lower triangular.

Constraint: \textit{uplo} = Nag_Upper or Nag_Lower.

3: \textbf{n} – Integer

\textit{Input}

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

Constraint: $n \geq 0$. 

Mark 25
4: \[ \text{ap}[\text{dim}] - \text{Complex} \]

**Note:** the dimension, \( \text{dim} \), of the array \( \text{ap} \) must be at least \( \max(1, n \times (n + 1)/2) \).

*On entry:* the Cholesky factor of \( A \) stored in packed form, as returned by nag_zpptrf (\text{f07grc}).

*On exit:* the factorization is overwritten by the \( n \) by \( n \) matrix \( A^{-1} \).

The storage of elements \( A_{ij} \) depends on the \text{order} and \text{uplo} arguments as follows:

- if \( \text{order} = \text{Nag-ColMajor} \) and \( \text{uplo} = \text{Nag_Upper} \), \( A_{ij} \) is stored in \( \text{ap}[(j - 1) \times j/2 + i - 1] \), for \( i \leq j \);
- if \( \text{order} = \text{Nag-ColMajor} \) and \( \text{uplo} = \text{Nag_Lower} \), \( A_{ij} \) is stored in \( \text{ap}[(2n - j) \times (j - 1)/2 + i - 1] \), for \( i \geq j \);
- if \( \text{order} = \text{Nag_RowMajor} \) and \( \text{uplo} = \text{Nag_Upper} \), \( A_{ij} \) is stored in \( \text{ap}[(2n - i) \times (i - 1)/2 + j - 1] \), for \( i \leq j \);
- if \( \text{order} = \text{Nag_RowMajor} \) and \( \text{uplo} = \text{Nag_Lower} \), \( A_{ij} \) is stored in \( \text{ap}[(i - 1) \times i/2 + j - 1] \), for \( i \geq j \).

5: \[ \text{fail} - \text{NagError}^* \]

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

# 6 Error Indicators and Warnings

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**

On entry, argument \( \langle \text{value} \rangle \) had an illegal value.

**NE_INT**

On entry, \( n = \langle \text{value} \rangle \).

Constraint: \( n \geq 0 \).

**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.

**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.

**NE_SINGULAR**

Diagonal element \( \langle \text{value} \rangle \) of the Cholesky factor is zero; the Cholesky factor is singular and the inverse of \( A \) cannot be computed.

# 7 Accuracy

The computed inverse \( X \) satisfies

\[
\|XA - I\|_2 \leq c(n)\epsilon\kappa_2(A) \quad \text{and} \quad \|AX - I\|_2 \leq c(n)\epsilon\kappa_2(A),
\]

where \( c(n) \) is a modest function of \( n \), \( \epsilon \) is the **machine precision** and \( \kappa_2(A) \) is the condition number of \( A \) defined by
\[ \kappa_2(A) = \|A\|_2 \|A^{-1}\|_2. \]

8 Parallelism and Performance

nag_zpptri (f07gwc) is not threaded by NAG in any implementation.
nag_zpptri (f07gwc) 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 real floating-point operations is approximately \( \frac{8}{3}n^3 \).
The real analogue of this function is nag_dpptri (f07gjc).

10 Example

This example computes the inverse of the matrix \( A \), where

\[
A = \begin{pmatrix}
3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\
1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\
1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\
0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i
\end{pmatrix}.
\]

Here \( A \) is Hermitian positive definite, stored in packed form, and must first be factorized by nag_zpptrf (f07grc).

10.1 Program Text

/* nag_zpptri (f07gwc) Example Program. *
 * Copyright 2014 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 ap_len, i, j, n;
    Integer exit_status = 0;
    NagError fail;
    Nag_UploType uplo;
    Nag_OrderType order;
    /* Arrays */
    char nag_enum_arg[40];
    Complex *ap = 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 */


```c
#define NAG_COLUMN_MAJOR
#define A_UPPER(I, J) ap[J*(J-1)/2 + I - 1]
#define A_LOWER(I, J) ap[(2*n-J)*(J-1)/2 + I - 1]
order = Nag_ColMajor;
#else
#define A_LOWER(I, J) ap[I*(I-1)/2 + J - 1]
#define A_UPPER(I, J) ap[(2*n-i)*(i-1)/2 + J - 1]
order = Nag_RowMajor;
#endif

INIT_FAIL(fail);
printf("nag_zpptri (f07gwc) Example Program Results\n\n");
/* Skip heading in data file */
#ifndef WIN32
scanf_s('%*[\n] ');
#else
scanf('%*[\n] ');
#endif
#ifndef _WIN32
scanf("%NAG_IFMT%*[\n] ", &n);
#else
scanf("%NAG_IFMT%*[\n] ", &n);
#endif
ap_len = n * (n + 1)/2;
/* Allocate memory */
if (!(ap = NAG_ALLOC(ap_len, Complex)))
{
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
}
/* Read A from data file */
#ifndef _WIN32
scanf_s(" %39s%*[\n] ", nag_enum_arg, _countof(nag_enum_arg));
#else
scanf(" %39s%*[\n] ", nag_enum_arg);
#endif
/* 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)
{
    for (i = 1; i <= n; ++i)
    {
        for (j = i; j <= n; ++j)
#ifdef _WIN32
            scanf_s(" ( %lf , %lf )", &A_UPPER(i, j).re,
                &A_UPPER(i, j).im);
#else
            scanf(" ( %lf , %lf )", &A_UPPER(i, j).re,
                &A_UPPER(i, j).im);
#endif
    }
#else
    for (i = 1; i <= n; ++i)
    {
        for (j = 1; j <= i; ++j)
#ifdef _WIN32
            scanf_s("%*[\n] ");
#else
            scanf("%*[\n] ");
#endif
    }else
    {
        for (i = 1; i <= n; ++i)
        {
            for (j = 1; j <= i; ++j)
#ifdef _WIN32
            scanf_s("%*[\n] ");
#else
            scanf("%*[\n] ");
#endif
        }
    }
#endif
END:
```

NAG Library Manual

f07gwc.4

Mark 25
```c
scanf_s(" ( %lf , %lf )", &A_LOWER(i, j).re,
      &A_LOWER(i, j).im);
#endif
#endif
}
#else
scanf(" ( %lf , %lf )", &A_LOWER(i, j).re,
      &A_LOWER(i, j).im);
#endif
#define _WIN32
scanf_s("%*[\n ] ");
#else
scanf("%*[\n ] ");
#endif

/* Factorize A */
/* nag_zpptrf (f07grc). *
* Cholesky factorization of complex Hermitian *
* positive-definite matrix, packed storage *
*/
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_zpptrf (f07grc).
    %s
", fail.message);
    exit_status = 1;
    goto END;
}

/* Compute inverse of A */
/* nag_zpptri (f07gwc). *
* Inverse of complex Hermitian positive-definite matrix, *
* matrix already factorized by nag_zpptrf (f07grc), packed *
* storage *
*/
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_zpptri (f07gwc).
    %s
", fail.message);
    exit_status = 1;
    goto END;
}

/* Print inverse */
/* nag_pack_complx_mat_print_comp (x04ddc). *
* Print complex packed triangular matrix (comprehensive) *
*/
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_pack_complx_mat_print_comp (x04ddc).
    %s
", fail.message);
    exit_status = 1;
    goto END;
}
END:
NAG_FREE(ap);
return exit_status;

10.2 Program Data

nag_zpptri (f07gwc) Example Program Data

4 :Value of n
    Nag_Lower :Value of uplo
    (3.23, 0.00)
    (1.51, 1.92)
    (3.58, 0.00)
    (1.90,-0.84)
    (-0.23,-1.11)
    (4.09, 0.00)
    (0.42,-2.50)
    (-1.18,-1.37)
    (2.33, 0.14)
    (4.29, 0.00) :End of matrix A
```
### 10.3 Program Results

**nag_zpptri (f07gwc) Example Program Results**

<table>
<thead>
<tr>
<th>Inverse</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 5.4691, 0.0000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(-1.2624, -1.5491)</td>
<td>( 1.1024, 0.0000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(-2.9746, -0.9616)</td>
<td>( 0.8989, -0.5672)</td>
<td>( 2.1589, 0.0000)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( 1.1962, 2.9772)</td>
<td>(-0.9826, -0.2566)</td>
<td>(-1.3756, -1.4550)</td>
<td>( 2.2934, 0.0000)</td>
</tr>
</tbody>
</table>