NAG C Library Function Document

nag_zhptri (f07pwc)

1 Purpose

nag_zhptri (f07pwc) computes the inverse of a complex Hermitian indefinite matrix \(A\), where \(A\) has been factorized by nag_zhptrf (f07prc), using packed storage.

2 Specification

```c
void nag_zhptri (Nag_OrderType order, Nag_UploType uplo, Integer n, Complex ap[],
                 const Integer ipiv[], NagError *fail)
```

3 Description

To compute the inverse of a complex Hermitian indefinite matrix \(A\), this function must be preceded by a call to nag_zhptrf (f07prc), which computes the Bunch–Kaufman factorization of \(A\) using packed storage.

If \(uplo = \text{Nag\_Upper}\), \(A = PUDU^H P^T\) and \(A^{-1}\) is computed by solving \(U^H P^T X P U = D^{-1}\) for \(X\).

If \(uplo = \text{Nag\_Lower}\), \(A = PLDL^H P^T\) and \(A^{-1}\) is computed by solving \(L^H P^T X P L = D^{-1}\) for \(X\).

4 References


5 Parameters

1: order – Nag_OrderType  
   *Input*
   
   *On entry:* the \(order\) parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by \(order = \text{Nag\_RowMajor}\). See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

   *Constraint:* \(order = \text{Nag\_RowMajor}\) or \(\text{Nag\_ColMajor}\).

2: uplo – Nag_UploType  
   *Input*
   
   *On entry:* indicates how \(A\) has been factorized as follows:
   
   if \(uplo = \text{Nag\_Upper}\), \(A = PUDU^H P^T\), where \(U\) is upper triangular;
   
   if \(uplo = \text{Nag\_Lower}\), \(A = PLDL^H P^T\), where \(L\) is lower triangular.

   *Constraint:* \(uplo = \text{Nag\_Upper}\) or \(\text{Nag\_Lower}\).

3: n – Integer  
   *Input*
   
   *On entry:* \(n\), the order of the matrix \(A\).

   *Constraint:* \(n \geq 0\).

4: ap[dim] – Complex  
   *Input/Output*
   
   *Note:* the dimension, \(dim\), of the array \(ap\) must be at least \(\max(1, n \times (n + 1)/2)\).

   *On entry:* details of the factorization of \(A\) stored in packed form, as returned by nag_zhptrf (f07prc).

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

   `[NP3645/7]`
5: \[ \text{ipiv}[\dim] - \text{const Integer} \quad \text{Input} \]

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

\textit{On entry:} details of the interchanges and the block structure of \( D \), as returned by \text{nag_zhptrf} (f07prc).

6: \[ \text{fail} - \text{NagError *} \quad \text{Output} \]

The NAG error parameter (see the Essential Introduction).

6 \textbf{Error Indicators and Warnings}

\textbf{NE_INT}

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

Constraint: \( n \geq 0 \).

\textbf{NE_SINGULAR}

The block diagonal matrix \( D \) is exactly singular.

\textbf{NE_ALLOC_FAIL}

Memory allocation failed.

\textbf{NE_BAD_PARAM}

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

\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 consult NAG for assistance.

7 \textbf{Accuracy}

The computed inverse \( X \) satisfies a bound of the form

\[
\text{if } \text{uplo} = \text{Nag_Upper}, \quad |DU^T P^T XPU - I| \leq c(n)\epsilon(|D| |U^T| |P| |X| |P| |U| + |D| |D^{-1}|); \\
\text{if } \text{uplo} = \text{Nag_Lower}, \quad |DL^T P^T XPL - I| \leq c(n)\epsilon(|D| |L^T| |P| |X| |P| |L| + |D| |D^{-1}|),
\]

\( c(n) \) is a modest linear function of \( n \), and \( \epsilon \) is the \textit{machine precision}.

8 \textbf{Further Comments}

The total number of real floating-point operations is approximately \( \frac{4}{3} n^3 \).

The real analogue of this function is \text{nag_dsptri} (f07pjc).

9 \textbf{Example}

To compute the inverse of the matrix \( A \), where

\[
A = \begin{pmatrix}
-1.36 + 0.00i & 1.58 + 0.90i & 2.21 - 0.21i & 3.91 + 1.50i \\
1.58 - 0.90i & -8.87 + 0.00i & -1.84 - 0.03i & -1.78 + 1.18i \\
2.21 + 0.21i & -1.84 + 0.03i & -4.63 + 0.00i & 0.11 + 0.11i \\
3.91 - 1.50i & -1.78 - 1.18i & 0.11 - 0.11i & -1.84 + 0.00i
\end{pmatrix}.
\]

Here \( A \) is Hermitian indefinite, stored in packed form, and must first be factorized by \text{nag_zhptrf} (f07prc).
9.1 Program Text

/* nag_zhptri (f07pwc) Example Program.  
 * Copyright 2001 Numerical Algorithms Group.  
 */
#include <stdio.h>
#include <nag.h>
#include <nagf07.h>
#include <nagx04.h>

int main(void)
{
    double ap_len, i, j, n;
    Integer exit_status=0;
    NagError fail;
    Nag_UploType uplo_enum;
    Nag_OrderType order;

    if ( !(ipiv = NAG_ALLOC(n, Integer)) ||
        !(ap = NAG_ALLOC(ap_len, Complex)) )
    {
        Vprintf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    if (uplo_enum == Nag_Upper)
    {
        for (i = 1; i <= n; ++i)
        {
            Vprintf("Unrecognised character for Nag_UploType type\n");
            exit_status = -1;
            goto END;
        }
    }
for (j = i; j <= n; ++j)
    Vscanf(" ( %lf , %lf )", &A_UPPER(i,j).re, &A_UPPER(i,j).im);
} 
Vscanf("%*[\n ] ");
} 
else 
{ 
    for (i = 1; i <= n; ++i)
    { 
        for (j = 1; j <= i; ++j)
            Vscanf(" ( %lf , %lf )", &A_LOWER(i,j).re, &A_LOWER(i,j).im);
        Vscanf("%*[\n ] ");
    }
/* Factorize A */
f07prc(order, uplo_enum, n, ap, ipiv, &fail); 
if (fail.code != NE_NOERROR) 
{ 
    Vprintf("Error from f07prc.
%s
", fail.message); 
    exit_status = 1; 
    goto END; 
} 
/* Compute inverse of A */
f07pwc(order, uplo_enum, n, ap, ipiv, &fail); 
if (fail.code != NE_NOERROR) 
{ 
    Vprintf("Error from f07pwc.
%s
", fail.message); 
    exit_status = 1; 
    goto END; 
} 
/* Print inverse */
x04ddc(order, uplo_enum, Nag_NonUnitDiag, n, ap, Nag_BracketForm, "%7.4f", "Inverse", Nag_IntegerLabels, 0, Nag_IntegerLabels, 0, 0, &fail); 
if (fail.code != NE_NOERROR) 
{ 
    Vprintf("Error from x04ddc.
%s
", fail.message); 
    exit_status = 1; 
    goto END; 
} 
END: 
if (ipiv) NAG_FREE(ipiv); 
if (ap) NAG_FREE(ap); 
return exit_status; }

9.2 Program Data

f07pwc Example Program Data
4 :Value of N
‘L’ :Value of UPLO
(-1.36, 0.00) (-1.58, -0.90) (-8.87, 0.00)
( 2.21, 0.21) (-1.84, 0.03) (-4.63, 0.00)
( 3.91, -1.50) (-1.78, -1.18) ( 0.11, -0.11) (-1.84, 0.00) :End of matrix A

9.3 Program Results

f07pwc Example Program Results

Inverse 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-0.0826, -0.0000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(-0.0335, 0.0440)</td>
<td>(-0.1408, 0.0000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( 0.0603, -0.0105)</td>
<td>( 0.0422, -0.0222)</td>
<td>(-0.2007, -0.0000)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( 0.2391, -0.0926)</td>
<td>( 0.0304, 0.0203)</td>
<td>( 0.0982, 0.0635)</td>
<td>( 0.0073, 0.0000)</td>
</tr>
</tbody>
</table>