NAG C Library Function Document

nag_zpbtrs (f07hsc)

1 Purpose

nag_zpbtrs (f07hsc) solves a complex Hermitian positive-definite band system of linear equations with multiple right-hand sides, $AX = B$, where $A$ has been factorized by nag_zpbtrf (f07hrc).

2 Specification

```c
void nag_zpbtrs (Nag_OrderType order, Nag_UploType uplo, Integer n, Integer kd, Integer nrhs, const Complex ab[], Integer pdab, Complex b[], Integer pdb, NagError *fail)
```

3 Description

To solve a complex Hermitian positive-definite band system of linear equations $AX = B$, this function must be preceded by a call to nag_zpbtrf (f07hrc) which computes the Cholesky factorization of $A$. The solution $X$ is computed by forward and backward substitution.

If $\text{uplo} = \text{Nag} \_ \text{Upper}$, $A = U^H U$, where $U$ is upper triangular; the solution $X$ is computed by solving $U^H Y = B$ and then $UX = Y$.

If $\text{uplo} = \text{Nag} \_ \text{Lower}$, $A = LL^H$, where $L$ is lower triangular; the solution $X$ is computed by solving $LY = B$ and then $L^H X = Y$.

4 References


5 Parameters

1: order – Nag_OrderType $\quad$ 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 = 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 = Nag_RowMajor or Nag_ColMajor.

2: uplo – Nag_UploType $\quad$ Input

*On entry*: indicates whether $A$ has been factorized as $U^H U$ or $LL^H$ as follows:

- if uplo = Nag_Upper, $A = U^H U$, where $U$ is upper triangular;
- if uplo = Nag_Lower, $A = LL^H$, where $L$ is lower triangular.

*Constraint*: uplo = Nag_Upper or Nag_Lower.

3: n – Integer $\quad$ Input

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

*Constraint*: $n \geq 0$. 
4: \textbf{kd} – Integer \hspace{1em} \textit{Input}

\textit{On entry:} $k$, the number of super-diagonals or sub-diagonals of the matrix $A$.
\textit{Constraint:} $\text{kd} \geq 0$.

5: \textbf{nrhs} – Integer \hspace{1em} \textit{Input}

\textit{On entry:} $r$, the number of right-hand sides.
\textit{Constraint:} $\text{nrhs} \geq 0$.

6: \textbf{ab}[$\text{dim}$] – const Complex \hspace{1em} \textit{Input}

\textit{Note:} the dimension, $\text{dim}$, of the array \textbf{ab} must be at least max($1$, pdab $\times$ n).
\textit{On entry:} the Cholesky factor of $A$, as returned by nag_zpbtrf (f07hrc).

7: \textbf{pdab} – Integer \hspace{1em} \textit{Input}

\textit{On entry:} the stride separating row or column elements (depending on the value of \texttt{order}) of the matrix in the array \textbf{ab}.
\textit{Constraint:} $\text{pdab} \geq \text{kd} + 1$.

8: \textbf{b}[$\text{dim}$] – Complex \hspace{1em} \textit{Input/Output}

\textit{Note:} the dimension, $\text{dim}$, of the array \textbf{b} must be at least max($1$, pdb $\times$ nrhs) when \texttt{order} = Nag_ColMajor and at least max($1$, pdb $\times$ n) when \texttt{order} = Nag_RowMajor.

If \texttt{order} = Nag_ColMajor, the $(i,j)$th element of the matrix $B$ is stored in $\textbf{b}[(j - 1) \times \text{pdb} + i - 1]$ and if \texttt{order} = Nag_RowMajor, the $(i,j)$th element of the matrix $B$ is stored in $\textbf{b}[(i - 1) \times \text{pdb} + j - 1]$.
\textit{On entry:} the $n$ by $r$ right-hand side matrix $B$.
\textit{On exit:} the $n$ by $r$ solution matrix $X$.

9: \textbf{pdb} – Integer \hspace{1em} \textit{Input}

\textit{On entry:} the stride separating matrix row or column elements (depending on the value of \texttt{order}) in the array \textbf{b}.
\textit{Constraints:}
\begin{align*}
\text{if} & \texttt{order} = \text{Nag_ColMajor}, \text{pdb} \geq \text{max}(1, \text{n}); \\
\text{if} & \texttt{order} = \text{Nag_RowMajor}, \text{pdb} \geq \text{max}(1, \text{nrhs}).
\end{align*}

10: \textbf{fail} – NagError * \hspace{1em} \textit{Output}

The NAG error parameter (see the Essential Introduction).

6 \hspace{1em} \textbf{Error Indicators and Warnings}

**NE_INT**

\textit{On entry,} $\text{n} = \langle\text{value}\rangle$.
\textit{Constraint:} $\text{n} \geq 0$.

\textit{On entry,} $\text{kd} = \langle\text{value}\rangle$.
\textit{Constraint:} $\text{kd} \geq 0$.

\textit{On entry,} $\text{nrhs} = \langle\text{value}\rangle$.
\textit{Constraint:} $\text{nrhs} \geq 0$.

\textit{On entry,} $\text{pdab} = \langle\text{value}\rangle$.
\textit{Constraint:} $\text{pdab} > 0$.

\textit{On entry,} $\text{pdb} = \langle\text{value}\rangle$.
\textit{Constraint:} $\text{pdb} > 0$. 
NE_INT_2

On entry, \( \text{pdab} = \langle \text{value} \rangle, \text{kd} = \langle \text{value} \rangle \).
Constraint: \( \text{pdab} \geq \text{kd} + 1 \).

On entry, \( \text{pdb} = \langle \text{value} \rangle, \text{n} = \langle \text{value} \rangle \).
Constraint: \( \text{pdb} \geq \max(1, \text{n}) \).

On entry, \( \text{pdb} = \langle \text{value} \rangle, \text{nrhs} = \langle \text{value} \rangle \).
Constraint: \( \text{pdb} \geq \max(1, \text{nrhs}) \).

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

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

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 Accuracy

For each right-hand side vector \( b \), the computed solution \( x \) is the exact solution of a perturbed system of
equations \( (A + E)x = b \), where

\[
\begin{align*}
\text{if} & \quad \text{uplo} = \text{Nag\_Upper}, \quad |E| \leq c(k + 1)\epsilon |U^H| |U|; \\
\text{if} & \quad \text{uplo} = \text{Nag\_Lower}, \quad |E| \leq c(k + 1)\epsilon |L^H| |L|,
\end{align*}
\]

\( c(k + 1) \) is a modest linear function of \( k + 1 \), and \( \epsilon \) is the machine precision.

If \( \hat{x} \) is the true solution, then the computed solution \( x \) satisfies a forward error bound of the form

\[
\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq c(k + 1) \text{cond}(A, x) \epsilon
\]

where \( \text{cond}(A, x) = \|\|A^{-1}\| |A||x\|_\infty / \|x\|_\infty \leq \text{cond}(A) = \|\|A^{-1}\| |A||\|_\infty \leq \kappa_\infty(A) \). Note that \( \text{cond}(A, x) \)

can be much smaller than \( \text{cond}(A) \).

Forward and backward error bounds can be computed by calling nag_zpbrfs (f07hvc), and an estimate for
\( \kappa_\infty(A) \) \( (= \kappa_1(A)) \) can be obtained by calling nag_zpbcon (f07huc).

8 Further Comments

The total number of real floating-point operations is approximately \( 16nk\tau \), assuming \( n \gg k \).

This function may be followed by a call to nag_zpbrfs (f07hvc) to refine the solution and return an error
estimate.

The real analogue of this function is nag_dpbtrs (f07hec).

9 Example

To solve the system of equations \( AX = B \), where

\[
A = \begin{pmatrix}
9.39 + 0.00i & 1.08 - 1.73i & 0.00 + 0.00i & 0.00 + 0.00i \\
1.08 + 1.73i & 1.69 + 0.00i & -0.04 + 0.29i & 0.00 + 0.00i \\
0.00 + 0.00i & -0.04 - 0.29i & 2.65 + 0.00i & -0.33 + 2.24i \\
0.00 + 0.00i & 0.00 + 0.00i & -0.33 - 2.24i & 2.17 + 0.00i
\end{pmatrix}
\]

and
Here $A$ is Hermitian positive-definite, and is treated as a band matrix, which must first be factorized by nag_zpbtrf (f07hrc).

### 9.1 Program Text

```c
/* nag_zpbtrs (f07hsc) Example Program. */
*
* Copyright 2001 Numerical Algorithms Group.
*
*/

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>
#include <nagx04.h>

int main(void) {
    /* Scalars */
    Integer i, j, k, kd, n, nrhs, pdab, pdb;
    Integer exit_status=0;
    Nag_UploType uplo_enum;
    NagError fail;
    Nag_OrderType order;

    /* Arrays */
    char uplo[2];
    Complex *ab=0, *b=0;

    #ifdef NAG_COLUMN_MAJOR
        #define AB_UPPER(I,J) ab[(J-1)*pdab + k + I - J - 1]
        #define AB_LOWER(I,J) ab[(J-1)*pdab + I - J]
        #define B(I,J) b[(J-1)*pdb + J - 1]
    #else
        #define AB_UPPER(I,J) ab[(I-1)*pdab + J - I]
        #define AB_LOWER(I,J) ab[(I-1)*pdab + k + J - I - 1]
        #define B(I,J) b[(I-1)*pdb + I - 1]
    #endif

    INIT_FAIL(fail);
    Vprintf("f07hsc Example Program Results\n\n");
    /* Skip heading in data file */
    Vscanf("%*[\n]");
    Vscanf("%ld%ld%ld%*[\n] ", &n, &kd, &nrhs);
    pdab = kd + 1;
    if ( !(ab = NAG_ALLOC((kd+1) * n, Complex)) ||
        !(b = NAG_ALLOC(n * nrhs, Complex)) )
    {
        Vprintf("Allocation failure\\n");
        exit_status = -1;
    }

    /* Allocate memory */
    if ( !((ab = NAG_ALLOC((kd+1) * n, Complex)) ||
        !((b = NAG_ALLOC(n * nrhs, Complex)) )
    {
        Vprintf("Allocation failure\\n");
        exit_status = -1;
    }
```

---

f07hsc

NAG C Library Manual

f07hsc.4

[NP3645/7]
/* Read A from data file */
Vscanf(" %ls \*[\n] ", uplo);
if (*{(unsigned char *)} uplo == 'L')
  uplo_enum = Nag_Lower;
else if (*{(unsigned char *)} uplo == 'U')
  uplo_enum = Nag_Upper;
else
{
  Vprintf("Unrecognised character for Nag_UploType type\n");
  exit_status = -1;
  goto END;
}
k = kd + 1;
if (uplo_enum == Nag_Upper)
{
  for (i = 1; i <= n; ++i)
  {
    for (j = i; j <= MIN(i+kd,n); ++j)
      Vscanf(" ( %lf , %lf )", &AB_UPPER(i,j).re, &AB_UPPER(i,j).im);
  }
  Vscanf("%*[\n] ");
}
else
{
  for (i = 1; i <= n; ++i)
  {
    for (j = MAX(1,i-kd); j <= i; ++j)
      Vscanf(" ( %lf , %lf )", &AB_LOWER(i,j).re, &AB_LOWER(i,j).im);
  }
  Vscanf("%*[\n] ");
}

/* Read B from data file */
for (i = 1; i <= n; ++i)
{
  for (j = 1; j <= nrhs; ++j)
    Vscanf(" ( %lf , %lf )", &B(i,j).re, &B(i,j).im);
}
Vscanf("%*[\n] ");

/* Factorize A */
f07hrc(order, uplo_enum, n, kd, ab, pdab, &fail);
if (fail.code != NE_NOERROR)
{
  Vprintf("Error from f07hrc.\n", fail.message);
  exit_status = 1;
  goto END;
}

/* Compute solution */
f07hsc(order, uplo_enum, n, kd, nrhs, ab, pdab, b, pdb, &fail);
if (fail.code != NE_NOERROR)
{
  Vprintf("Error from f07hsc.\n", fail.message);
  exit_status = 1;
  goto END;
}

/* Print solution */
x04dbc(order, Nag_GeneralMatrix, Nag_PackedDiag, n, nrhs, b, pdb,
        Nag_BracketForm, "%7.4f", "Solution(s)", Nag_IntegerLabels,
        0, Nag_IntegerLabels, 0, 80, 0, 0, &fail);
if (fail.code != NE_NOERROR)
{
  Vprintf("Error from x04dbc.\n", fail.message);
  exit_status = 1;
  goto END;
}

END:
if (ab) NAG_FREE(ab);
if (b) NAG_FREE(b);
return exit_status;
}

9.2 Program Data

f07hsc Example Program Data
4 1 2 :Values of N, KD and NRHS
‘L’ :Value of UPLO
   ( 9.39, 0.00)
   ( 1.08, 1.73) ( 1.69, 0.00)
   ( -0.04, -0.29) ( 2.65, 0.00)
   ( -0.33, -2.24) ( 2.17, 0.00) :End of matrix A
   (-12.42, 68.42) (54.30, -56.56)
   ( -9.93, 0.88) (18.32, 4.76)
   ( -27.30, -0.01) ( -4.40, 9.97)
   (  5.31, 23.63) ( 9.43, 1.41) :End of matrix B

9.3 Program Results

f07hsc Example Program Results

Solution(s)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-1.0000, 8.0000)</td>
<td>(5.0000, -6.0000)</td>
</tr>
<tr>
<td>2</td>
<td>(2.0000, -3.0000)</td>
<td>(2.0000, 3.0000)</td>
</tr>
<tr>
<td>3</td>
<td>(-4.0000, -5.0000)</td>
<td>(-8.0000, 4.0000)</td>
</tr>
<tr>
<td>4</td>
<td>(7.0000, 6.0000)</td>
<td>(-1.0000, -7.0000)</td>
</tr>
</tbody>
</table>