NAG Library Function Document

nag_zpbstf (f08utc)

1 Purpose
nag_zpbstf (f08utc) computes a split Cholesky factorization of a complex Hermitian positive definite band matrix.

2 Specification

```c
#include <nag.h>
#include <nagf08.h>
void nag_zpbstf (Nag_OrderType order, Nag_UploType uplo, Integer n,
            Integer kb, Complex bb[], Integer pdbb, NagError *fail)
```

3 Description
nag_zpbstf (f08utc) computes a split Cholesky factorization of a complex Hermitian positive definite band matrix $B$. It is designed to be used in conjunction with nag_zhbgst (f08usc).

The factorization has the form $B = S^H S$, where $S$ is a band matrix of the same bandwidth as $B$ and the following structure: $S$ is upper triangular in the first $(n + k)/2$ rows, and transposed — hence, lower triangular — in the remaining rows. For example, if $n = 9$ and $k = 2$, then

$$
S = \begin{pmatrix}
S_{11} & S_{12} & S_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{22} & S_{23} & S_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{33} & S_{34} & S_{35} & 0 & 0 & 0 & 0 & 0 & 0 \\
S_{44} & S_{45} & S_{46} & S_{47} & 0 & 0 & 0 & 0 & 0 \\
S_{55} & S_{56} & S_{57} & S_{58} & S_{59} & 0 & 0 & 0 & 0 \\
S_{66} & S_{67} & S_{68} & S_{69} & S_{70} & S_{71} & 0 & 0 & 0 \\
S_{77} & S_{78} & S_{79} & S_{80} & S_{81} & S_{82} & S_{83} & 0 & 0 \\
S_{88} & S_{89} & S_{90} & S_{91} & S_{92} & S_{93} & S_{94} & S_{95} & 0 \\
S_{99} & S_{100} & S_{101} & S_{102} & S_{103} & S_{104} & S_{105} & S_{106} & S_{107} \\
\end{pmatrix}
$$

4 References
None.

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 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

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

2: `uplo` – Nag_UploType

*Input*

*On entry*: indicates whether the upper or lower triangular part of $B$ is stored.

`uplo = Nag_Upper`

The upper triangular part of $B$ is stored.
uplo = Nag_Lower
    The lower triangular part of B is stored.

Constraint: uplo = Nag_Upper or Nag_Lower.

3: n – Integer
    Input
    On entry: n, the order of the matrix B.
    Constraint: n ≥ 0.

4: kb – Integer
    Input
    On entry: if uplo = Nag_Upper, the number of superdiagonals, k_b, of the matrix B.
    If uplo = Nag_Lower, the number of subdiagonals, k_b, of the matrix B.
    Constraint: kb ≥ 0.

5: bb[dim] – Complex
    Input/Output
    Note: the dimension, dim, of the array bb must be at least max(1, pddb × n).
    On entry: this n by n Hermitian positive definite band matrix B.
    This is stored as a notional two-dimensional array with row elements or column elements stored contiguously. The storage of elements of $B_{ij}$, depends on the order and uplo arguments as follows:
    
    - if order = Nag_ColMajor and uplo = Nag_Upper,
      $B_{ij}$ is stored in $bb[k_b + i - j + (j - 1) × pddb]$, for $j = 1, \ldots, n$ and $i = \max(1, j - k_b), \ldots, j$;
    - if order = Nag_ColMajor and uplo = Nag_Lower,
      $B_{ij}$ is stored in $bb[i - j + (j - 1) × pddb]$, for $j = 1, \ldots, n$ and $i = j, \ldots, \min(n, j + k_b)$;
    - if order = Nag_RowMajor and uplo = Nag_Upper,
      $B_{ij}$ is stored in $bb[j - i + (i - 1) × pddb]$, for $i = 1, \ldots, n$ and $j = i, \ldots, \min(n, i + k_b)$;
    - if order = Nag_RowMajor and uplo = Nag_Lower,
      $B_{ij}$ is stored in $bb[k_b + j - i + (i - 1) × pddb]$, for $i = 1, \ldots, n$ and $j = \max(1, i - k_b), \ldots, i$.
    
    On exit: B is overwritten by the elements of its split Cholesky factor S.

6: pddb – Integer
    Input
    On entry: the stride separating row or column elements (depending on the value of order) of the matrix B in the array bb.
    Constraint: pddb ≥ kb + 1.

7: fail – NagError *
    Input/Output
    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 <value> had an illegal value.
NE_INT
On entry, \( kb = \langle \text{value} \rangle \).
Constraint: \( kb \geq 0 \).

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

On entry, \( pdbb = \langle \text{value} \rangle \).
Constraint: \( pdbb > 0 \).

NE_INT_2
On entry, \( pdbb = \langle \text{value} \rangle \) and \( kb = \langle \text{value} \rangle \).
Constraint: \( pdbb \geq kb + 1 \).

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_POS_DEF
The factorization could not be completed, because the updated element \( b(\langle \text{value} \rangle, \langle \text{value} \rangle) \) would be the square root of a negative number. Hence \( B \) is not positive definite. This may indicate an error in forming the matrix \( B \).

7 Accuracy
The computed factor \( S \) is the exact factor of a perturbed matrix \( (B + E) \), where
\[
|E| \leq c(k+1)\epsilon |S^H||S|,
\]
\( c(k+1) \) is a modest linear function of \( k+1 \), and \( \epsilon \) is the machine precision. It follows that
\[
|e_{ij}| \leq c(k+1)\epsilon \sqrt{(b_{ii}b_{jj})}.
\]

8 Parallelism and Performance
nag_zpbstf (f08utc) is not threaded by NAG in any implementation.

nag_zpbstf (f08utc) 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 \( 4n(k+1)^2 \), assuming \( n \gg k \).

A call to nag_zpbstf (f08utc) may be followed by a call to nag_zhbgst (f08usc) to solve the generalized eigenproblem \( Az = \lambda Bz \), where \( A \) and \( B \) are banded and \( B \) is positive definite.

The real analogue of this function is nag_dpbstf (f08ufc).
10 Example
See Section 10 in nag_zhbgst (f08usc).