NAG Library Function Document

nag_dpbstf (f08ufc)

1 Purpose

nag_dpbstf (f08ufc) computes a split Cholesky factorization of a real symmetric positive definite band matrix.

2 Specification

```
#include <nag.h>
#include <nagf08.h>
```

```
void nag_dpbstf (Nag_OrderType order, Nag_UploType uplo, Integer n,
     Integer kb, double bb[], Integer pdbb, NagError *fail)
```

3 Description

nag_dpbstf (f08ufc) computes a split Cholesky factorization of a real symmetric positive definite band matrix B. It is designed to be used in conjunction with nag_dsbgst (f08uec).

The factorization has the form $B = S^{\mathrm{T}}S$, 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

$$S = \begin{pmatrix} s_{11} & s_{12} & s_{13} & & & & & & \\ & s_{22} & s_{23} & s_{24} & & & & & \\ & & s_{33} & s_{34} & s_{35} & & & & \\ & & & s_{44} & s_{45} & & & & \\ & & & & s_{55} & & & & \\ & & & s_{64} & s_{65} & s_{66} & & & \\ & & & & s_{75} & s_{76} & s_{77} & & \\ & & & & & s_{86} & s_{87} & s_{88} & \\ & & & & & & s_{97} & s_{98} & s_{99} \end{pmatrix}.$$

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 2.3.1.3 in How to Use the NAG Library and its Documentation 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: $\mathbf{n} \geq 0$.

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: $\mathbf{kb} \geq 0$.

5: **bb**[*dim*] – double *Input/Output*

Note: the dimension, *dim*, of the array **bb** must be at least $\max(1, \mathbf{pdbb} \times \mathbf{n})$.

On entry: the n by n symmetric 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 $\mathbf{bb}[k_b + i - j + (j-1) \times \mathbf{pdbb}]$, 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 $\mathbf{bb}[i - j + (j-1) \times \mathbf{pdbb}]$, 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 $\mathbf{bb}[j - i + (i-1) \times \mathbf{pdbb}]$, 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 $\mathbf{bb}[k_b + j - i + (i-1) \times \mathbf{pdbb}]$, 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: **pdbb** – 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: $\mathbf{pdbb} \geq \mathbf{kb} + 1$.

7: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, **kb** $= \langle value \rangle$.
Constraint: **kb** ≥ 0.

On entry, **n** $= \langle value \rangle$.
Constraint: **n** ≥ 0.

On entry, **pdbb** $= \langle value \rangle$.
Constraint: **pdbb** > 0.

NE_INT_2

On entry, **pdbb** $= \langle value \rangle$ and **kb** $= \langle value \rangle$.
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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_POS_DEF

The factorization could not be completed, because the updated element $b(\langle value \rangle, \langle value \rangle)$ 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

$$|E| \leq c(k+1)\epsilon \left|S^{\mathrm{T}}\right| |S|,$$

$c(k+1)$ is a modest linear function of $k+1$, and ϵ is the ***machine precision***. It follows that $\left|e_{ij}\right| \leq c(k+1)\epsilon\sqrt{\left(b_{ii}b_{jj}\right)}$.

8 Parallelism and Performance

nag_dpbstf (f08ufc) 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 $n(k+1)^2$, assuming $n \gg k$.

A call to nag_dpbstf (f08ufc) may be followed by a call to nag_dsbgst (f08uec) to solve the generalized eigenproblem $Az = \lambda Bz$, where A and B are banded and B is positive definite.

The complex analogue of this function is nag_zpbstf (f08utc).

10 Example

See Section 10 in nag_dsbgst (f08uec).