NAG Library Function Document

nag_dgbtrf (f07bdc)

1 Purpose

nag_dgbtrf (f07bdc) computes the LU factorization of a real $m$ by $n$ band matrix.

2 Specification

```c
#include <nag.h>
#include <nagf07.h>

void nag_dgbtrf (Nag_OrderType order, Integer m, Integer n, Integer kl,
                 Integer ku, double ab[], Integer pdab, Integer ipiv[], NagError *fail)
```

3 Description

nag_dgbtrf (f07bdc) forms the LU factorization of a real $m$ by $n$ band matrix $A$ using partial pivoting, with row interchanges. Usually $m = n$, and then, if $A$ has $k_l$ nonzero subdiagonals and $k_u$ nonzero superdiagonals, the factorization has the form $A = PLU$, where $P$ is a permutation matrix, $L$ is a lower triangular matrix with unit diagonal elements and at most $k_l$ nonzero elements in each column, and $U$ is an upper triangular band matrix with $k_l + k_u$ superdiagonals.

Note that $L$ is not a band matrix, but the nonzero elements of $L$ can be stored in the same space as the subdiagonal elements of $A$. $U$ is a band matrix but with $k_l$ additional superdiagonals compared with $A$. These additional superdiagonals are created by the row interchanges.

4 References


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: m – Integer

   **Input**

   *On entry*: $m$, the number of rows of the matrix $A$.

   *Constraint*: $m \geq 0$.

3: n – Integer

   **Input**

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

   *Constraint*: $n \geq 0$. 
4: kl – Integer

*Input*

On entry: \( k_l \), the number of subdiagonals within the band of the matrix \( A \).

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

5: ku – Integer

*Input*

On entry: \( k_u \), the number of superdiagonals within the band of the matrix \( A \).

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

6: ab[\text{dim}] – double

*Input/Output*

*Note:* the dimension, \( \text{dim} \), of the array \( ab \) must be at least

\[
\max(1, pdab \times n) \quad \text{when} \quad \text{order} = \text{Nag\_ColMajor}; \\
\max(1, m \times pdab) \quad \text{when} \quad \text{order} = \text{Nag\_RowMajor}.
\]

On entry: the \( m \) by \( n \) matrix \( A \).

This is stored as a notional two-dimensional array with row elements or column elements stored contiguously. The storage of elements \( A_{ij} \), for row \( i = 1, \ldots, m \) and column \( j = \max(1, i - k_l), \ldots, \min(n, i + k_u) \), depends on the \text{order} argument as follows:

- if \( \text{order} = \text{Nag\_ColMajor} \), \( A_{ij} \) is stored as \( ab[(j - 1) \times pdab + kl + ku + i - j] \);
- if \( \text{order} = \text{Nag\_RowMajor} \), \( A_{ij} \) is stored as \( ab[(i - 1) \times pdab + kl + j - i] \).

See Section 9 in nag_dgbsv (f07bac) for further details.

On exit: \( ab \) is overwritten by details of the factorization.

The elements, \( u_{ij} \), of the upper triangular band factor \( U \) with \( k_l + k_u \) super-diagonals, and the multipliers, \( l_{ij} \), used to form the lower triangular factor \( L \) are stored. The elements \( u_{ij} \), for \( i = 1, \ldots, m \) and \( j = i, \ldots, \min(n, i + k_l + k_u) \), and \( l_{ij} \), for \( i = 1, \ldots, m \) and \( j = \max(1, i - k_l), \ldots, i \), are stored where \( A_{ij} \) is stored on entry.

7: pdab – Integer

*Input*

On entry: the stride separating row or column elements (depending on the value of \text{order}) of the matrix \( A \) in the array \( ab \).

*Constraint:* \( pdab \geq 2 \times kl + ku + 1 \).

8: ipiv[\text{min}(m, n)] – Integer

*Output*

On exit: the pivot indices that define the permutation matrix. At the \( i \)th step, if \( \text{ipiv}[i - 1] > i \) then row \( i \) of the matrix \( A \) was interchanged with row \( \text{ipiv}[i - 1] \), for \( i = 1, 2, \ldots, \min(m, n) \). \( \text{ipiv}[i - 1] \leq i \) indicates that, at the \( i \)th step, a row interchange was not required.

9: 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 \( \langle \text{value} \rangle \) had an illegal value.
NE_INT
On entry, $kl = \langle value \rangle$.
Constraint: $kl \geq 0$.
On entry, $ku = \langle value \rangle$.
Constraint: $ku \geq 0$.
On entry, $m = \langle value \rangle$.
Constraint: $m \geq 0$.
On entry, $n = \langle value \rangle$.
Constraint: $n \geq 0$.
On entry, $pdab = \langle value \rangle$.
Constraint: $pdab > 0$.

NE_INT_3
On entry, $pdab = \langle value \rangle$, $kl = \langle value \rangle$ and $ku = \langle value \rangle$.
Constraint: $pdab \geq 2 \times kl + ku + 1$.

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_SINGULAR
Element $\langle value \rangle$ of the diagonal is exactly zero. The factorization has been completed, but the
factor $U$ is exactly singular, and division by zero will occur if it is used to solve a system of
equations.

7 Accuracy
The computed factors $L$ and $U$ are the exact factors of a perturbed matrix $A + E$, where
$$|E| \leq c(k)\epsilon \|L\|\|U\|,$$
c($k$) is a modest linear function of $k = k_l + k_u + 1$, and $\epsilon$ is the machine precision. This assumes
$k \ll \min(m,n)$.

8 Parallelism and Performance
nag_dgbtrf (f07bdc) is threaded by NAG for parallel execution in multithreaded implementations of the
NAG Library.
nag_dgbtrf (f07bdc) 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 varies between approximately $2nkl(k_u + 1)$ and $2nkl(k_l + k_u + 1)$, depending on the interchanges, assuming $m = n \gg k_l$ and $n \gg k_u$.

A call to nag_dgbtrf (f07bdc) may be followed by calls to the functions:

- nag_dgbtrs (f07bec) to solve $AX = B$ or $A^T X = B$;
- nag_dgbcon (f07bgc) to estimate the condition number of $A$.

The complex analogue of this function is nag_zgbtrf (f07brc).

10 Example

This example computes the $LU$ factorization of the matrix $A$, where

$$A = \begin{pmatrix} -0.23 & 2.54 & -3.66 & 0.00 \\ -6.98 & 2.46 & -2.73 & -2.13 \\ 0.00 & 2.56 & 2.46 & 4.07 \\ 0.00 & 0.00 & -4.78 & -3.82 \end{pmatrix}.$$ 

Here $A$ is treated as a band matrix with one subdiagonal and two superdiagonals.

10.1 Program Text

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

int main(void) {
    /* Scalars */
    Integer i, ipiv_len, j, kl, ku, m, n, pdab;
    Integer exit_status = 0;
    NagError fail;
    Nag_OrderType order;

    /* Arrays */
    double *ab = 0;
    Integer *ipiv = 0;

    INIT_FAIL(fail);
    printf("nag_dgbtrf (f07bdc) Example Program Results\n\n");
    /* Skip heading in data file */
```
#ifdef _WIN32
    scanf_s("%*[\n] ");
#else
    scanf("%*[\n] ");
#endif
#ifdef _WIN32
    scanf("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n] ", &m, &n, &kl, &ku);
#else
    scanf("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n] ", &m, &n, &kl, &ku);
#endif
ipiv_len = MIN(m, n);
pdab = 2*kl + ku + 1;
/* Allocate memory */
if (!(ab = NAG_ALLOC((2*kl+ku+1) * MAX(m, n), double)) ||
    !(ipiv = NAG_ALLOC(ipiv_len, Integer)))
{
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
}
/* Read A from data file */
for (i = 1; i <= m; ++i)
{
    for (j = MAX(i-kl, 1); j <= MIN(i+ku, n); ++j)
    #ifdef _WIN32
        scanf_s("%lf", &AB(i, j));
    #else
        scanf("%lf", &AB(i, j));
    #endif
}
/* Factorize A */
/* nag_dgbtrf (f07bdc). */
/* LU factorization of real m by n band matrix */
/* f07bdc.5 */
nag_dgbtrf(order, m, n, kl, ku, ab, pdab, ipiv, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dgbtrf (f07bdc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}
/* Print details of factorization */
/* nag_band_real_mat_print (x04cec). */
/* Print real packed banded matrix (easy-to-use) */
fflush(stdout);
nag_band_real_mat_print(order, m, n, kl, kl+ku, ab, pdab,
    "Details of factorization", 0, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_band_real_mat_print (x04cec).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}
/* Print pivot indices */
printf("\nipiv\n");
for (i = 1; i <= MIN(m, n); ++i)
    printf("%10"NAG_IFMT"%s", ipiv[i-1], i%7 == 0?"\n":" ");
printf("\n");
END:
NAG_FREE(ab);
NAG_FREE(ipiv);
    return exit_status;
}

10.2 Program Data

nag_dgbtrf (f07bdc) Example Program Data
4 4 1 2 :Values of M, N, KL and KU
-0.23 2.54 -3.66
-6.98 2.46 -2.73 -2.13
  2.56 2.46 4.07
-4.78 -3.82 :End of matrix AB

10.3 Program Results

nag_dgbtrf (f07bdc) Example Program Results

Details of factorization

<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>-6.9800</td>
<td>2.4600</td>
<td>-2.7300</td>
<td>-2.1300</td>
</tr>
<tr>
<td>2</td>
<td>0.0330</td>
<td>2.5600</td>
<td>2.4600</td>
<td>4.0700</td>
</tr>
<tr>
<td>3</td>
<td>0.9605</td>
<td>-5.9329</td>
<td>-3.8391</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.8057</td>
<td>-0.7269</td>
<td></td>
</tr>
</tbody>
</table>

ipiv

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>