NAG C 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
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 kl non-zero sub-diagonals and ku non-zero super-diagonals, 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 kl non-zero elements in each column, and U is an upper triangular band matrix with kl + ku super-diagonals.

Note that L is not a band matrix, but the non-zero elements of L can be stored in the same space as the sub-diagonal elements of A. U is a band matrix but with kl additional super-diagonals compared with A. These additional super-diagonals are created by the row interchanges.

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 = 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: `m` – Integer

   *Input*

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

   *Constraint:* `m >= 0`.

3: `n` – Integer

   *Input*

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

   *Constraint:* `n >= 0`.

4: `kl` – Integer

   *Input*

   *On entry:* kl, the number of sub-diagonals within the band of A.

   *Constraint:* `kl >= 0`.

[NP3645/7]
5:  **ku** – Integer

*Input*

*On entry:* \(k_u\), the number of super-diagonals within the band of \(A\).

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

6:  **ab[\text{dim}]** – double

*Input/Output*

*Note:* the dimension, \(\text{dim}\), of the array \(ab\) must be at least \(\max(1, pdab \times n)\) when \(\text{order} = \text{Nag\_ColMajor}\) and at least \(\max(1, pdab \times m)\) when \(\text{order} = \text{Nag\_RowMajor}\).

*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 \(i = 1, \ldots, m\) and \(j = \max(1, i - kl), \ldots, \min(n, i + ku)\), depends on the \text{order} parameter 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 \(ab[(i - 1) \times pdab + kl + j - i]\).

*On exit:* \(ab\) is overwritten by details of the factorization. The elements, \(u_{ij}\), of the upper triangular band factor \(U\) with \(kl + ku\) 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 \(j = i, \ldots, \min(n, i + kl + ku)\), and \(l_{ij}\), for \(i = 1, \ldots, m\) and \(j = \max(1, i - kl), \ldots, i\) are stored using the same storage scheme as as described for \(a_{ij}\) 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{dim}]** – Integer

*Output*

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

*On exit:* the pivot indices. Row \(i\) of the matrix \(A\) was interchanged with row \(ipiv[i - 1]\), for \(i = 1, 2, \ldots, \min(m, n)\).

9:  **fail** – NagError *

*Output*

The NAG error parameter (see the Essential Introduction).

### 6 Error Indicators and Warnings

**NE\_INT**

*On entry,* \(m = \langle value\rangle\).

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

*On entry,* \(n = \langle value\rangle\).

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

*On entry,* \(kl = \langle value\rangle\).

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

*On entry,* \(ku = \langle value\rangle\).

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

*On entry,* \(pdab = \langle value\rangle\).

*Constraint:* \(pdab > 0\).

**NE\_INT\_3**

*On entry,* \(pdab = \langle value\rangle, kl = \langle value\rangle, ku = \langle value\rangle\).

*Constraint:* \(pdab \geq 2 \times kl + ku + 1\).
The factor $U$ is exactly singular.

Memory allocation failed.

On entry, parameter (value) had an illegal value.

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.

The computed factors $L$ and $U$ are the exact factors of a perturbed matrix $A + E$, where

$$|E| \leq c(k)\epsilon|P| |L| |U|,$$

$c(k)$ is a modest linear function of $k = k_l + k_u + 1$, and $\epsilon$ is the *machine precision*. This assumes

$k \ll \min(m,n)$.

The total number of floating-point operations varies between approximately $2nk_l(k_u + 1)$ and $2nk_l(k_l + k_u + 1)$, depending on the interchanges, assuming $m = n \gg k_l$ and $n \gg k_u$.

A call to this function 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).

To compute 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 1 sub-diagonal and 2 super-diagonals.

/* nag_dgbtrf (f07bdc) 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, ipiv_len, j, kl, ku, m, n, pdab;
  Integer exit_status=0;
  NagError fail;
  Nag_OrderType order;
/* Arrays */
  double *ab=0;
  Integer *ipiv=0;
#ifdef NAG_COLUMN_MAJOR
#define AB(I,J) ab[(J-1)*pdab + kl + ku + I - J]
#else
#define AB(I,J) ab[(I-1)*pdab + kl + J - I]
#endif
order = Nag_ColMajor;
#else
order = Nag_RowMajor;
#endif
INIT_FAIL(fail);
Vprintf("f07bdc Example Program Results\n\n");
/* Skip heading in data file */
Vscanf("%*[\n ]");
Vscanf("%ld%ld%ld%ld%*[\n ]", &m, &n, &kl, &ku);
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)) )
{
  Vprintf("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)
    Vscanf("%lf", &AB(i,j));
}
Vscanf("%*[\n ]");
/* Factorize A */
f07bdc(order, m, n, kl, ku, ab, pdab, ipiv, &fail);
if (fail.code != NE_NOERROR)
{
  Vprintf("Error from f07bdc.\n\n", fail.message);
  exit_status = 1;
  goto END;
}
/* Print details of factorization */
x04cec(order, m, n, kl, kl+ku, ab, pdab,
       "Details of factorization", 0, &fail);
if (fail.code != NE_NOERROR)
{
  Vprintf("Error from x04cec.\n\n", fail.message);
  exit_status = 1;
  goto END;
}
/* Print pivot indices */
Vprintf("\n\n", "PIV");
for (i = 1; i <= MIN(m,n); ++i)
  Vprintf("%10ld%8s", ipiv[i-1], i%7==0 ?"\n": "");
Vprintf("\n\n");
END:
  if (ab) NAG_FREE(ab);
if (ipiv) NAG_FREE(ipiv);
return exit_status;
}

9.2 Program Data

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

9.3 Program Results

f07bdc Example Program Results

<table>
<thead>
<tr>
<th>Details of factorization</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>0.8057</td>
<td>-0.7269</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| IPIV | 2 | 3 | 3 | 4 |