

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

```
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 non-zero sub-diagonals and k_u 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 k_l non-zero elements in each column, and U is an upper triangular band matrix with $k_l + k_u$ 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 k_l additional super-diagonals compared with A . These additional super-diagonals are created by the row interchanges.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

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: $m \geq 0$.
- 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 sub-diagonals within the band of A .
Constraint: $kl \geq 0$.

- 5: **ku** – Integer *Input*
On entry: k_u , the number of super-diagonals within the band of A .
Constraint: $\mathbf{ku} \geq 0$.
- 6: **ab**[*dim*] – double *Input/Output*
Note: the dimension, *dim*, of the array **ab** must be at least $\max(1, \mathbf{pdab} \times \mathbf{n})$ when **order** = **Nag_ColMajor** and at least $\max(1, \mathbf{pdab} \times \mathbf{m})$ when **order** = **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 $i = 1, \dots, m$ and $j = \max(1, i - k_l), \dots, \min(n, i + k_u)$, depends on the **order** parameter as follows:
 if **order** = **Nag_ColMajor**, a_{ij} is stored as $\mathbf{ab}[(j - 1) \times \mathbf{pdab} + \mathbf{kl} + \mathbf{ku} + i - j]$;
 if **order** = **Nag_RowMajor**, a_{ij} is stored as $\mathbf{ab}[(i - 1) \times \mathbf{pdab} + \mathbf{kl} + j - i]$.
On exit: **ab** is overwritten by details of the factorization. The elements, u_{ij} , of the upper triangular band factor U with $k_l + k_u$ super-diagonals, and the multipliers, l_{ij} , used to form the lower triangular factor L are stored. The elements u_{ij} , for $i = 1, \dots, m$ and $j = i, \dots, \min(n, i + k_l + k_u)$, and l_{ij} , for $i = 1, \dots, m$ and $j = \max(1, i - k_l), \dots, i$ 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 **order**) of the matrix A in the array **ab**.
Constraint: $\mathbf{pdab} \geq 2 \times \mathbf{kl} + \mathbf{ku} + 1$.
- 8: **ipiv**[*dim*] – Integer *Output*
Note: the dimension, *dim*, of the array **ipiv** must be at least $\max(1, \min(\mathbf{m}, \mathbf{n}))$.
On exit: the pivot indices. Row i of the matrix A was interchanged with row **ipiv**[$i - 1$], for $i = 1, 2, \dots, \min(m, n)$.
- 9: **fail** – NagError * *Output*
 The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **m** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{m} \geq 0$.

On entry, **n** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{n} \geq 0$.

On entry, **kl** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{kl} \geq 0$.

On entry, **ku** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{ku} \geq 0$.

On entry, **pdab** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{pdab} > 0$.

NE_INT_3

On entry, **pdab** = $\langle \text{value} \rangle$, **kl** = $\langle \text{value} \rangle$, **ku** = $\langle \text{value} \rangle$.

Constraint: $\mathbf{pdab} \geq 2 \times \mathbf{kl} + \mathbf{ku} + 1$.

NE_SINGULAR

The factor U is exactly singular.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ 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

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 ϵ is the *machine precision*. This assumes $k \ll \min(m, n)$.

8 Further Comments

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).

9 Example

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.

9.1 Program Text

```
/* nag_dgbtrf (f07bdc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#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]
order = Nag_ColMajor;
#else
#define AB(I,J) ab[(I-1)*pdab + kl + J - I]
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%s\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%s\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Print pivot indices */
Vprintf("\n%s\n", "IPIV");
for (i = 1; i <= MIN(m,n); ++i)
    Vprintf("%10ld%s", ipiv[i-1], i%7==0 ? "\n": " ");
Vprintf("\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

Details of factorization

	1	2	3	4
1	-6.9800	2.4600	-2.7300	-2.1300
2	0.0330	2.5600	2.4600	4.0700
3		0.9605	-5.9329	-3.8391
4			0.8057	-0.7269

IPIV

2	3	3	4
---	---	---	---
