NAG Library Function Document

nag_dgttrf (f07cdc)

1 Purpose

nag_dgttrf (f07cdc) computes the \(LU\) factorization of a real \(n\) by \(n\) tridiagonal matrix \(A\).

2 Specification

```c
#include <nag.h>
#include <nagf07.h>
void nag_dgttrf (Integer n, double dl[], double d[], double du[],
                 double du2[], Integer ipiv[], NagError *fail)
```

3 Description

nag_dgttrf (f07cdc) uses Gaussian elimination with partial pivoting and row interchanges to factorize the matrix \(A\) as

\[ A = PLU, \]

where \(P\) is a permutation matrix, \(L\) is unit lower triangular with at most one nonzero subdiagonal element in each column, and \(U\) is an upper triangular band matrix, with two superdiagonals.

4 References


5 Arguments

1: \(n\) – Integer

\textit{Input}

On entry: \(n\), the order of the matrix \(A\).

Constraint: \(n \geq 0\).

2: \(dl[\dim]\) – double

\textit{Input/Output}

\textit{Note}: the dimension, \(\dim\), of the array \(dl\) must be at least \(\max(1, n - 1)\).

On entry: must contain the \((n - 1)\) subdiagonal elements of the matrix \(A\).

On exit: is overwritten by the \((n - 1)\) multipliers that define the matrix \(L\) of the \(LU\) factorization of \(A\).

3: \(d[\dim]\) – double

\textit{Input/Output}

\textit{Note}: the dimension, \(\dim\), of the array \(d\) must be at least \(\max(1, n)\).

On entry: must contain the \(n\) diagonal elements of the matrix \(A\).

On exit: is overwritten by the \(n\) diagonal elements of the upper triangular matrix \(U\) from the \(LU\) factorization of \(A\).

4: \(du[\dim]\) – double

\textit{Input/Output}

\textit{Note}: the dimension, \(\dim\), of the array \(du\) must be at least \(\max(1, n - 1)\).
On entry: must contain the \((n - 1)\) superdiagonal elements of the matrix \(A\).

On exit: is overwritten by the \((n - 1)\) elements of the first superdiagonal of \(U\).

5: \(\textbf{du2}[n - 2]\) – double  
Output  
On exit: contains the \((n - 2)\) elements of the second superdiagonal of \(U\).

6: \(\textbf{ipiv}[n]\) – Integer  
Output  
On exit: contains the \(n\) pivot indices that define the permutation matrix \(P\). At the \(i\)th step, row \(i\) of the matrix was interchanged with row \(\text{ipiv}[i - 1]\). \text{ipiv}[i - 1]\) will always be either \(i\) or \((i + 1)\), \text{ipiv}[i - 1] = i\) indicating that a row interchange was not performed.

7: \(\textbf{fail}\) – 
\text{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, \(n = \langle\text{value}\rangle\).  
Constraint: \(n \geq 0\).

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 \(\langle\text{value}\rangle\) 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 factorization satisfies an equation of the form  
\[ A + E = PLU, \]
where  
\[ \|E\|_\infty = O(\epsilon)\|A\|_\infty \]
and \(\epsilon\) is the \textit{machine precision}. 

Following the use of this function, nag_dgtrfs (f07cec) can be used to solve systems of equations $AX = B$ or $A^TX = B$, and nag_dgtcon (f07cge) can be used to estimate the condition number of $A$.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The total number of floating-point operations required to factorize the matrix $A$ is proportional to $n$. The complex analogue of this function is nag_zgttrf (f07crc).

10 Example

This example factorizes the tridiagonal matrix $A$ given by

$$
A = \begin{pmatrix}
3.0 & 2.1 & 0 & 0 & 0 \\
3.4 & 2.3 & -1.0 & 0 & 0 \\
0 & 3.6 & -5.0 & 1.9 & 0 \\
0 & 0 & 7.0 & -0.9 & 8.0 \\
0 & 0 & 0 & -6.0 & 7.1
\end{pmatrix}.
$$

10.1 Program Text

/* nag_dgtrfs (f07cdc) Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 23, 2011. */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>

int main(void)
{
    /* Scalars */
    Integer exit_status = 0, i, n;

    /* Arrays */
    double *d = 0, *dl = 0, *du = 0, *du2 = 0;
    Integer *ipiv = 0;

    /* Nag Types */
    NagError fail;

    INIT_FAIL(fail);

    printf("nag_dgtrfs (f07cdc) Example Program Results\n\n");

    /* Skip heading in data file */
    #ifdef _WIN32
        scanf_s("%*[\n]");
    #else
        scanf("%*[\n]");
    #endif

    #ifdef _WIN32
        scanf_s("%NAG_IFMT"%*[\n], &n);
    #else
        scanf("%NAG_IFMT"%*[\n], &n);
    #endif

    if (n < 0)
printf("Invalid n\n\n");
exit_status = 1;
goto END;
}

/* Allocate memory */
if (!((d = NAG_ALLOC(n, double))) ||
    (!d1 = NAG_ALLOC(n-1, double)) ||
    (!du = NAG_ALLOC(n-1, double)) ||
    (!du2 = NAG_ALLOC(n-2, double)) ||
    (!ipiv = NAG_ALLOC(n, Integer)) )
{
    printf("Allocation failure\n");
    exit_status = -1;
goto END;
}

/* Read the tridiagonal matrix A from data file */
#ifdef _WIN32
    for (i = 0; i < n-1; ++i) scanf_s("%lf", &du[i]);
#else
    for (i = 0; i < n-1; ++i) scanf("%lf", &du[i]);
#endif
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
#ifdef _WIN32
    for (i = 0; i < n; ++i) scanf_s("%lf", &d[i]);
#else
    for (i = 0; i < n; ++i) scanf("%lf", &d[i]);
#endif
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
#ifdef _WIN32
    for (i = 0; i < n-1; ++i) scanf_s("%lf", &dl[i]);
#else
    for (i = 0; i < n-1; ++i) scanf("%lf", &dl[i]);
#endif
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif

/* Factorize the tridiagonal matrix A using nag_dgttrf (f07cdc). */
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_dgttrf (f07cdc). %s\n", fail.message);
    exit_status = 1;
goto END;
}

/* Print details of the factorization */
printf("Details of factorization\n");
for (i = 0; i < n - 2; ++i) printf("%8.4f\n", du2[i], i%8 == 7?'\n': "");
printf("%8.4f%8.4f%8.4f\n", d[i], du[i], i%8 == 7?'\n': "");
printf("%8.4f\n", du[i], i%8 == 7?'\n': "");
printf("%8.4f\n", d[i], i%8 == 7?'\n': "");

printf("%8.4f%8.4f%8.4f\n", d[i], du[i], i%8 == 7?'\n': "");
printf("%8.4f\n", du[i], i%8 == 7?'\n': "");
printf("%8.4f%8.4f%8.4f\n", d[i], du[i], i%8 == 7?'\n': "");
printf("%8.4f\n", du[i], i%8 == 7?'\n': "");
printf("%8.4f%8.4f%8.4f\n", d[i], du[i], i%8 == 7?'\n': "");
printf("%8.4f%8.4f%8.4f\n", d[i], du[i], i%8 == 7?'\n': "");
END:
NAG_FREE(d);
NAG_FREE(dl);
NAG_FREE(du);
NAG_FREE(du2);
NAG_FREE(ipiv);
return exit_status;
}

10.2 Program Data

nag_dgttrf (f07cdc) Example Program Data
5 : n
  2.1 -1.0 1.9 8.0
  3.0 2.3 -5.0 -0.9 7.1
  3.4 3.6 7.0 -6.0 : matrix A

10.3 Program Results

nag_dgttrf (f07cdc) Example Program Results

Details of factorization

Second super-diagonal of U
-1.0000 1.9000 8.0000

First super-diagonal of U
2.3000 -5.0000 -0.9000 7.1000

Main diagonal of U
3.4000 3.6000 7.0000 -6.0000 -1.0154

Multipliers
0.8824 0.0196 0.1401 -0.0148

Vector of interchanges
  2   3   4   5   5

---