nag_real_lu (f03afc)

1. Purpose

nag_real_lu (f03afc) computes an LU factorization of a real matrix, with partial pivoting, and evaluates the determinant.

2. Specification

```
#include <nag.h>
#include <nagf03.h>

void nag_real_lu(Integer n, double a[], Integer tda, Integer pivot[],
     double *detf, Integer *dete, NagError *fail)
```

3. Description

This function computes an LU factorization of a real matrix A with partial pivoting: $PA = LU$, where P is a permutation matrix, L is lower triangular and U is unit upper triangular. The determinant of A is the product of the diagonal elements of L with the correct sign determined by the row interchanges.

4. Parameters

n
Input: n, the order of the matrix A.
Constraint: $\mathbf{n} \geq 1$.

a[n][tda]
Input: the n by n matrix A.
Output: A is overwritten by the lower triangular matrix L and the off-diagonal elements of the upper triangular matrix U. The unit diagonal elements of U are not stored.

tda
Input: the second dimension of the array **a** as declared in the function from which nag_real_lu is called.
Constraint: $\mathbf{tda} \geq \mathbf{n}$.

pivot[n]
Output: $\mathbf{pivot}[i-1]$ gives the row index of the ith pivot.

detf
dete
Output: the determinant of A is given by $\mathbf{detf} \times 2.0^{\mathbf{dete}}$. It is given in this form to avoid overflow or underflow.

fail
The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_SINGULAR
The matrix A is singular, possibly due to rounding errors. The factorization could not be completed. **detf** and **dete** are set to zero.

NE_INT_ARG_LT
On entry, **n** must not be less than 1: $\mathbf{n} = \langle value \rangle$.

NE_2_INT_ARG_LT
On entry, $\mathbf{tda} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$. The parameters must satisfy $\mathbf{tda} \geq \mathbf{n}$.

NE_ALLOC_FAIL
Memory allocation failed.

6. Further Comments

The time taken by the function is approximately proportional to n^3.

6.1. Accuracy

The accuracy of the determinant depends on the conditioning of the original matrix. For a detailed error analysis, see Wilkinson and Reinsch (1971) p 107.

6.2. References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation (Vol II, Linear Algebra)* Springer-Verlag pp 93-110.

7. See Also

nag_real_lu_solve_mult_rhs (f04ajc)

8. Example

To compute the LU factorization with partial pivoting, and calculate the determinant, of the real matrix

$$\begin{pmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{pmatrix}.$$

8.1. Program Text

```
/* nag_real_lu(f03afc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <math.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagf03.h>

#define NMAX 8
#define TDA NMAX

main()
{
  double detf, two=2.0;
  Integer  i, dete, j, n;
  static NagError fail;
  double a[NMAX][TDA];
  Integer pivot[NMAX];

  Vprintf("f03afc Example Program Results\n");
  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  Vscanf("%ld",&n);

  if (n>0 && n<=NMAX)
    {
      for (i=0; i<n; i++)
        for (j=0; j<n; j++)
          Vscanf("%lf",&a[i][j]);
      fail.print = TRUE;
      f03afc(n,(double *)a,(Integer)TDA,pivot,&detf,&dete,&fail);
      if (fail.code!=NE_NOERROR)
        exit(EXIT_FAILURE);
      else
```

```
          {
            Vprintf("Array A after factorization\n");
            for (i=0; i<n; i++)
              for (j=0; j<n; j++)
                Vprintf("%9.4f%s",a[i][j],(j%8==7 || j==n-1) ? "\n" : " ");
            Vprintf("\nArray P\n");
            for (i=0; i<n; i++)
              Vprintf("%3ld%s",pivot[i],(i%8==7 || i==n-1) ? "\n" : " ");
            Vprintf("\ndetf = %9.4f  dete = %2ld\n", detf, dete);
            detf = detf * pow(two, (double)dete);
            Vprintf("\nValue of determinant = %9.4f\n", detf);
          }
      }
    exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
f03afc Example Program Data
  3
   33   16   72
  -24  -10  -57
   -8   -4  -17
```

8.3. Program Results

```
f03afc Example Program Results
Array A after factorization
  -8.0000    0.5000    2.1250
 -24.0000    2.0000   -3.0000
  33.0000   -0.5000    0.3750

Array P
  3   2   3

detf =    0.3750  dete =  4

Value of determinant =    6.0000
```