nag_real_lin_eqn (f04arc)

1. Purpose

nag_real_lin_eqn (f04arc) calculates the approximate solution of a set of real linear equations with a single right-hand side, using an LU factorization with partial pivoting.

2. Specification

```
#include <nag.h>
#include <nagf04.h>

void nag_real_lin_eqn(Integer n, double a[], Integer tda, double b[],
     double x[], NagError *fail)
```

3. Description

Given a set of linear equations, $Ax = b$, the function first computes an LU factorization of A with partial pivoting, $PA = LU$, where P is a permutation matrix, L is lower triangular and U is unit upper triangular. The approximate solution x is found by forward and backward substitution in $Ly = Pb$ and $Ux = y$, where b is the right-hand side.

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_lin_eqn is called.
Constraint: $\mathbf{tda} \geq \mathbf{n}$.

b[n]

Input: the right-hand side vector b.

x[n]

Output: the solution vector x.

fail

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

5. Error Indications and Warnings

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 $\mathbf{n} = \langle value \rangle$. These parameters must satisfy $\mathbf{tda} \geq \mathbf{n}$.

NE_SINGULAR

The matrix A is singular, possibly due to rounding errors.

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 computed solution 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)
nag_real_lu (f03afc)

8. Example

To solve the set of linear equations $Ax = b$ where

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

and

$$B = \begin{pmatrix} -359 \\ 281 \\ 85 \end{pmatrix}.$$

8.1. Program Text

```
/* nag_real_lin_eqn(f04arc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagf04.h>

#define NMAX 8
#define TDA NMAX

main()
{

  double   a[NMAX][TDA], b[NMAX], x[NMAX];
  Integer  i, j, n;

  Vprintf("f04arc Example Program Results\n");
  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  Vscanf("%ld",&n);
  if (n<1 || n>NMAX)
    {
      Vfprintf(stderr, "n is out of range: n = %5ld\n", n);
      exit(EXIT_FAILURE);
    }
  for (i=0; i<n; i++)
    for (j=0; j<n; j++)
      Vscanf("%lf",&a[i][j]);
  for (i=0; i<n; i++)
    Vscanf("%lf",&b[i]);
  f04arc(n,(double *)a, (Integer)TDA, b, x, NAGERR_DEFAULT);
  Vprintf("Solution\n");
  for (i=0; i<n; i++)
    Vprintf("%9.4f\n",x[i]);
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
f04arc Example Program Data
  3
   33   16   72
  -24  -10  -57
   -8   -4  -17
 -359  281   85
```

8.3. Program Results

```
f04arc Example Program Results
Solution
   1.0000
  -2.0000
  -5.0000
```