nag_complex_lu (f03ahc)

1. Purpose

nag_complex_lu (f03ahc) computes an LU factorization of a complex matrix, with partial pivoting, and evaluates the determinant.

2. Specification

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

void nag_complex_lu(Integer n, Complex a[], Integer tda, Integer pivot[],
     Complex *det, Integer *dete, NagError *fail)
```

3. Description

This function computes an LU factorization of a complex 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 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_complex_lu is called.
Constraint: $\mathbf{tda} \geq \mathbf{n}$.

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

det
dete
Output: the determinant of A is given by $(\mathbf{det.re} + i\mathbf{det.im}) \times 2.0^{\mathbf{dete}}$. It is given in this form to avoid overflow and 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_complex_lin_eqn_mult_rhs (f04adc)
nag_complex_lu_solve_mult_rhs (f04akc)

8. Example

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

$$\begin{pmatrix} 2 & 1+2i & 2+10i \\ 1+i & 1+3i & -5+14i \\ 1+i & 5i & -7+20i \end{pmatrix}.$$

8.1. Program Text

```
/* nag_complex_lu(f03ahc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1A revised, (Oct 1990).
 */

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

main()
{
#define NMAX 5
#define TDA NMAX

  Integer pivot[NMAX];
  Complex a[NMAX][TDA], det;
  Integer i, j, n, dete;
  static NagError fail;

  Vprintf("f03ahc Example Program Results\n");
  Vscanf("%*[^\n]"); /* Skip heading in data file */
  fail.print = TRUE;

  if (scanf("%ld",&n)!=EOF)
    if (n > 0 && n <= NMAX)
      {
        for (i = 0; i < n; i++)
          for (j = 0; j < n; j++)
            Vscanf(" ( %lf , %lf ) ", &a[i][j].re, &a[i][j].im);
        f03ahc(n, (Complex *)a, (Integer)TDA, pivot, &det, &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("(%7.3f, %7.3f) ", a[i][j].re, a[i][j].im);
                  Vprintf("\n");
                }
              Vprintf("\nArray pivot\n");
              for (i=0; i<n; i++)
                Vprintf("%5ld",pivot[i]);
              Vprintf("\n\ndet.re = %7.4f, det.im = %7.4f, dete = %2ld.\n",
                      det.re, det.im, dete);
              det.re = ldexp(det.re, (int)dete);
              det.im = ldexp(det.im, (int)dete);
              Vprintf("\nValue of determinant = (%7.4f, %7.4f)\n", det.re, det.im)
            }
        }
      else
        {
          Vfprintf(stderr, "Error: n is out of range: n = %5ld\n", n);
          exit(EXIT_FAILURE);
        }
    exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
f03ahc Example Program Data
  3
 (2.0, 0.0)   (1.0, 2.0)   (2.0,10.0)
 (1.0, 1.0)   (1.0, 3.0)  (-5.0,14.0)
 (1.0, 1.0)   (0.0, 5.0)  (-7.0,20.0)
```

8.3. Program Results

```
f03ahc Example Program Results
Array a after factorization
(  2.000,   0.000) (  0.500,   1.000) (  1.000,   5.000)
(  1.000,   1.000) (  0.500,   3.500) (  3.800,   1.400)
(  1.000,   1.000) (  1.500,   1.500) ( -4.600,   0.200)

Array pivot
    1    3    3

det.re =  0.0234, det.im =  0.1250, dete =  8.

Value of determinant = ( 6.0000, 32.0000)
```