

NAG C Library Function Document

nag_cubic_roots (c02akc)

1 Purpose

nag_cubic_roots (c02akc) determines the roots of a cubic equation with real coefficients.

2 Specification

```
void nag_cubic_roots (double u, double r, double s, double t, double zror[],
                     double zeroi[], double errest[], NagError *fail)
```

3 Description

nag_cubic_roots attempts to find the roots of the cubic equation

$$uz^3 + rz^2 + sz + t = 0,$$

where u, r, s and t are real coefficients with $u \neq 0$. The roots are located by finding the eigenvalues of the associated 3 by 3 (upper Hessenberg) companion matrix H given by

$$H = \begin{pmatrix} 0 & 0 & -t/u \\ 1 & 0 & -s/u \\ 0 & 1 & -r/u \end{pmatrix}.$$

Further details can be found in Section 6.

To obtain the roots of a quadratic equation, nag_quartic_roots (c02alc) can be used.

4 Parameters

- | | | |
|----|--|---------------|
| 1: | u – double | <i>Input</i> |
| | <i>On entry:</i> u , the coefficient of z^3 . | |
| | <i>Constraint:</i> $u \neq 0.0$. | |
| 2: | r – double | <i>Input</i> |
| | <i>On entry:</i> r , the coefficient of z^2 . | |
| 3: | s – double | <i>Input</i> |
| | <i>On entry:</i> s , the coefficient of z . | |
| 4: | t – double | <i>Input</i> |
| | <i>On entry:</i> t , the constant coefficient. | |
| 5: | zror[3] – double | <i>Output</i> |
| 6: | zeroi[3] – double | <i>Output</i> |
| | <i>On exit:</i> zror $[i - 1]$ and zeroi $[i - 1]$ contain the real and imaginary parts, respectively, of the i th root. | |
| 7: | errest[3] – double | <i>Output</i> |
| | <i>On exit:</i> errest $[i - 1]$ contains an approximate error estimate for the i th root. | |

8: **fail** – NagError *

Input/Output

The NAG error parameter (see the Essential Introduction).

5 Error Indicators and Warnings

NE_REAL

On entry, $u = 0.0$.

Constraint: $u \neq 0.0$.

NE_C02_OVERFLOW

The companion matrix H cannot be formed without overflow.

NE_C02_NOT_CONV

The iterative procedure used to determine the eigenvalues has failed to converge.

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.

6 Further Comments

The method used by the routine consists of the following steps, which are performed by routines from LAPACK.

- (a) Form H .
- (b) Apply a diagonal similarity transformation to H (to give H').
- (c) Calculate the eigenvalues and Schur factorization of H' .
- (d) Calculate the left and right eigenvectors of H' .
- (e) Estimate reciprocal condition numbers for all the eigenvalues of H' .
- (f) Calculate approximate error estimates for all the eigenvalues of H' (using the 1-norm).

6.1 Accuracy

If **fail.code** = **NE_NOERROR** on exit, then the i th computed root should have approximately $|\log_{10}(\text{errest}[i - 1])|$ correct significant digits.

6.2 References

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

7 See Also

nag_quartic_roots (c02alc)

8 Example

To find the roots of the cubic equation

$$z^3 + 3z^2 + 9z - 13 = 0.$$

8.1 Program Text

```

/* nag_cubic_roots (c02akc) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * NAG C Library
 *
 * Mark 6, 2000.
 */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagc02.h>

int main(void)
{
    double *errest=0, *zeroi=0, *zeror=0;
    double r, s, t, u;
    Integer i;
    Integer exit_status=0;
    NagError fail;

    INIT_FAIL(fail);
    Vprintf("c02akc Example Program Results\n\n");

    if
    (
        ! (errest = NAG_ALLOC(3, double)) ||
        ! (zeroi = NAG_ALLOC(3, double)) ||
        ! (zeror = NAG_ALLOC(3, double))
    )
    {
        Vprintf ("Allocation failure\n");
        exit_status=-1;
        goto END;
    }

    /* Skip heading in data file */
    Vscanf("%*[\n] ");
    Vscanf("%lf %lf %lf %lf ", &u, &r, &s, &t);

    c02akc (u, r, s, t, zeror, zeroi, errest, &fail);
    if (fail.code == NE_NOERROR)
    {
        Vprintf("\n Roots of cubic equation           Error estimates\n");
        Vprintf("                               (machine-dependent)\n\n");
        for (i = 0; i <= 2; ++i)
        {
            Vprintf(" z = %10.5f %10.5f%s           %g\n",
                    zeror[i], zeroi[i], "*i", errest[i]);
        }
    }
    else
    {
        Vprintf("Error from c02akc.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
}

```

```
    }  
  
END:  
  if (errest) NAG_FREE (errest);  
  if (zeroi) NAG_FREE (zeroi);  
  if (zeror) NAG_FREE (zeror);  
  return exit_status;  
}
```

8.2 Program Data

c02akc Example Program Data

1.0 3.0 9.0 -13.0 : Values of u, r, s and t

8.3 Program Results

c02akc Example Program Results

Roots of cubic equation		Error estimates (machine-dependent)
z =	1.00000 0.00000*i	2.37689e-15
z =	-2.00000 3.00000*i	3.08488e-15
z =	-2.00000 -3.00000*i	3.08488e-15
