

## NAG 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

```
#include <nag.h>
#include <nagc02.h>
void nag_cubic_roots (double u, double r, double s, double t, double zero[],
                     double zeroi[], double errest[], NagError *fail)
```

#### 3 Description

nag\_cubic\_roots (c02akc) 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<sup>2</sup>  $H$  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 9.

To obtain the roots of a quadratic equation, nag\_quartic\_roots (c02alc) can be used.

#### 4 References

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

#### 5 Arguments

- |   |              |
|---|--------------|
| 1: <b>u</b> – double                              | <i>Input</i> |
| <i>On entry:</i> $u$ , the coefficient of $z^3$ . |              |
| <i>Constraint:</i> $u \neq 0.0$ .                 |              |
| 2: <b>r</b> – double                              | <i>Input</i> |
| <i>On entry:</i> $r$ , the coefficient of $z^2$ . |              |
| 3: <b>s</b> – double                              | <i>Input</i> |
| <i>On entry:</i> $s$ , the coefficient of $z$ .   |              |
| 4: <b>t</b> – double                              | <i>Input</i> |
| <i>On entry:</i> $t$ , the constant coefficient.  |              |

- 5: **zeror**[3] – double *Output*
- 6: **zeroi**[3] – double *Output*  
*On exit:* **zeror**[ $i - 1$ ] and **zeroi**[ $i - 1$ ] contain the real and imaginary parts, respectively, of the  $i$ th root.
- 7: **errest**[3] – double *Output*  
*On exit:* **errest**[ $i - 1$ ] contains an approximate error estimate for the  $i$ th root.
- 8: **fail** – NagError \* *Input/Output*  
The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_C02\_NOT\_CONV

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

### NE\_C02\_OVERFLOW

The companion matrix  $H$  cannot be formed without overflow.

### 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.

### NE\_REAL

On entry,  $\mathbf{u} = 0.0$ .  
Constraint:  $\mathbf{u} \neq 0.0$ .

## 7 Accuracy

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

## 8 Parallelism and Performance

Not applicable.

## 9 Further Comments

The method used by the function consists of the following steps, which are performed by functions 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).

## 10 Example

To find the roots of the cubic equation

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

### 10.1 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);

    printf("nag_cubic_roots (c02akc) Example Program Results\n\n");

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

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

    /* nag_cubic_roots (c02akc).
     * Zeros of a cubic polynomial with real coefficients
     */
    nag_cubic_roots(u, r, s, t, zeror, zeroi, errest, &fail);
    if (fail.code == NE_NOERROR)
    {
        printf(
            "\n Roots of cubic equation           Error estimates\n");
        printf(
            "                                     (machine-dependent)\n\n");
        for (i = 0; i <= 2; ++i)
        {
            printf(" z = %10.5f %10.5f%s           %g\n",
                zeror[i], zeroi[i], " *i", errest[i]);
        }
    }
    else
    {
        printf("Error from nag_cubic_roots (c02akc).\n%s\n",

```

```
        fail.message);
    exit_status = 1;
    goto END;
}

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

## 10.2 Program Data

nag\_cubic\_roots (c02akc) Example Program Data  
1.0 3.0 9.0 -13.0 : Values of u, r, s and t

## 10.3 Program Results

nag\_cubic\_roots (c02akc) Example Program Results

Roots of cubic equation		Error estimates (machine-dependent)
z =	1.00000 0.00000*i	2.37922e-15
z =	-2.00000 3.00000*i	3.08789e-15
z =	-2.00000 -3.00000*i	3.08789e-15

---