

NAG Library Function Document

nag_quartic_roots (c02alc)

1 Purpose

nag_quartic_roots (c02alc) determines the roots of a quartic equation with real coefficients.

2 Specification

```
#include <nag.h>
#include <nagc02.h>

void nag_quartic_roots (double e, double a, double b, double c, double d,
    double zeror[], double zeroi[], double errest[], NagError *fail)
```

3 Description

nag_quartic_roots (c02alc) attempts to find the roots of the quartic equation

$$ez^4 + az^3 + bz^2 + cz + d = 0,$$

where e , a , b , c and d are real coefficients with $e \neq 0$. The roots are located by finding the eigenvalues of the associated 4 by 4 (upper Hessenberg) companion matrix H given by

$$H = \begin{pmatrix} -d/e \\ -c/e \\ -b/e \\ -a/e \end{pmatrix}.$$

Further details can be found in Section 9.

To obtain the roots of a cubic equation, nag_cubic_roots (c02akc) 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: | e – double | <i>Input</i> |
| | <i>On entry:</i> e , the coefficient of z^4 . | |
| | <i>Constraint:</i> $e \neq 0.0$. | |
| 2: | a – double | <i>Input</i> |
| | <i>On entry:</i> a , the coefficient of z^3 . | |
| 3: | b – double | <i>Input</i> |
| | <i>On entry:</i> b , the coefficient of z^2 . | |
| 4: | c – double | <i>Input</i> |
| | <i>On entry:</i> c , the coefficient of z . | |

- 5: **d** – double *Input*
On entry: d , the constant coefficient.
- 6: **zeror**[4] – double *Output*
7: **zeroi**[4] – double *Output*
On exit: **zeror**[$i - 1$] and **zeroi**[$i - 1$] contain the real and imaginary parts, respectively, of the i th root.
- 8: **errest**[4] – double *Output*
On exit: **errest**[$i - 1$] contains an approximate error estimate for the i th root.
- 9: **fail** – NagError * *Input/Output*
The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

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, $e = 0.0$.
Constraint: $e \neq 0.0$.

7 Accuracy

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

8 Parallelism and Performance

nag_quartic_roots (c02alc) is not threaded in any implementation.

9 Further Comments

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

- (a) Form matrix 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 quartic equation

$$z^4 + 2z^3 + 6z^2 - 8z - 40 = 0.$$

10.1 Program Text

```

/* nag_quartic_roots (c02alc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * NAG C Library
 *
 * Mark 26, 2016.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagc02.h>

int main(void)
{
    double a, b, c, d, e;
    double *errest = 0, *zeroi = 0, *zeror = 0;
    Integer i;
    Integer exit_status = 0;
    NagError fail;

    INIT_FAIL(fail);

    printf("nag_quartic_roots (c02alc) Example Program Results\n\n");
    if (!(errest = NAG_ALLOC(4, double)) ||
        !(zeroi = NAG_ALLOC(4, double)) || !(zeror = NAG_ALLOC(4, double))
        )
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
#ifdef _WIN32
    scanf_s("%lf %lf %lf %lf %lf", &e, &a, &b, &c, &d);
#else
    scanf("%lf %lf %lf %lf %lf", &e, &a, &b, &c, &d);
#endif

    /* nag_quartic_roots (c02alc).
     * Zeros of a real quartic polynomial with real coefficients
     */
    nag_quartic_roots(e, a, b, c, d, zeror, zeroi, errest, &fail);
    if (fail.code == NE_NOERROR) {
        printf("Roots of quartic equation          Error estimates\n");
        printf("                                (machine-dependent)\n\n");
        for (i = 0; i <= 3; ++i) {
            printf("%s %10.5f %10.5f%s          %g\n", " z =",
                zeror[i], zeroi[i], "*i", errest[i]);
        }
    }
    else {
        printf("Error from nag_quartic_roots (c02alc).\n%s\n", fail.message);
    }
}

```

```
        exit_status = 1;
        goto END;
    }
END:
    NAG_FREE(errest);
    NAG_FREE(zeroi);
    NAG_FREE(zerror);
    return exit_status;
}
```

10.2 Program Data

nag_quartic_roots (c02alc) Example Program Data
1.0 2.0 6.0 -8.0 -40.0 : Values of e, a, b, c and d

10.3 Program Results

nag_quartic_roots (c02alc) Example Program Results

Roots of quartic equation		Error estimates (machine-dependent)
z =	2.00000 0.00000*i	3.39305e-15
z =	-2.00000 0.00000*i	5.29913e-15
z =	-1.00000 3.00000*i	4.54822e-15
z =	-1.00000 -3.00000*i	4.54822e-15
