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 zeror[],
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 \( H \) 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

5 Arguments
1: \( u \) – double \hspace{1cm} Input
   \( On \ entry: u \), the coefficient of \( z^3 \).
   \( Constraint: u \neq 0.0 \).

2: \( r \) – double \hspace{1cm} Input
   \( On \ entry: r \), the coefficient of \( z^2 \).

3: \( s \) – double \hspace{1cm} Input
   \( On \ entry: s \), the coefficient of \( z \).

4: \( t \) – double \hspace{1cm} Input
   \( On \ entry: t \), the constant coefficient.
5:  \texttt{zeror[3]} – double  \hspace{1cm} \textit{Output}
6:  \texttt{zeroi[3]} – double  \hspace{1cm} \textit{Output}

\textit{On exit:} \texttt{zeror[i – 1]} and \texttt{zeroi[i – 1]} contain the real and imaginary parts, respectively, of the \textit{i}th root.

7:  \texttt{errest[3]} – double  \hspace{1cm} \textit{Output}

\textit{On exit:} \texttt{errest[i – 1]} contains an approximate error estimate for the \textit{i}th root.

8:  \texttt{fail} – NagError * \hspace{1cm} \textit{Input/Output}

The NAG error argument (see Section 3.6 in the Essential Introduction).

\section{Error Indicators and Warnings}

\textbf{NE\_C02\_NOT\_CONV}

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

\textbf{NE\_C02\_OVERFLOW}

The companion matrix \( H \) cannot be formed without overflow.

\textbf{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.

\textbf{NE\_REAL}

On entry, \( u = 0.0 \).
Constraint: \( u \neq 0.0 \).

\section{Accuracy}

If \( \text{fail.code} = \text{NE\_NOERROR} \) on exit, then the \textit{i}th computed root should have approximately \( \log_{10} (\texttt{errest[i – 1]}) \) correct significant digits.

\section{Parallelism and Performance}

Not applicable.

\section{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

```c
/* nag_cubic_roots (c02akc) Example Program. */
/* Copyright 2014 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 */
    #ifdef _WIN32
    scanf_s("%*[\n] ");
    #else
    scanf("%*[\n] ");
    #endif
    #ifdef _WIN32
    scanf_s("%lf %lf %lf %lf ", &u, &r, &s, &t);
    #else
    scanf("%lf %lf %lf %lf ", &u, &r, &s, &t);
    #endif

    /* 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\n");
        printf("n (machine-dependent)\n\n");
        for (i = 0; i <= 2; ++i)
            printf("%8.4f\n", zeror[i]);
        printf("%8.4f\n", zeroi[i]);
    }

    return 0;
}
```

Mark 25

"c02akc.3"


```c
{  
  printf(" z = %10.5f %10.5f%s %g\n",  
         zeron[i], zeroi[i], "i", errest[i]);  
}
}

else {
  printf("Error from nag_cubic_roots (c02akc).\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

<table>
<thead>
<tr>
<th>Roots of cubic equation</th>
<th>Error estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>z = 1.00000 0.00000*i</td>
<td></td>
</tr>
<tr>
<td>z = -2.00000 3.00000*i</td>
<td></td>
</tr>
<tr>
<td>z = -2.00000 -3.00000*i</td>
<td></td>
</tr>
<tr>
<td>2.37922e-15</td>
<td></td>
</tr>
<tr>
<td>3.08789e-15</td>
<td></td>
</tr>
<tr>
<td>3.08789e-15</td>
<td></td>
</tr>
</tbody>
</table>
```

---

C02AKC NAG Library Manual

C02AKC (last) Mark 25