

NAG C Library Function Document

nag_1d_ratnl_eval (e01rbc)

1 Purpose

nag_1d_ratnl_eval (e01rbc) evaluates continued fractions of the form produced by nag_1d_ratnl_interp (e01rac).

2 Specification

```
void nag_1d_ratnl_eval (Integer m, const double a[], const double u[], double x,
    double *f, NagError *fail)
```

3 Description

nag_1d_ratnl_eval (e01rbc) evaluates the continued fraction

$$R(x) = a_1 + R_m(x)$$

where

$$R_i(x) = \frac{a_{m-i+2}(x - u_{m-i+1})}{1 + R_{i-1}(x)}, \quad \text{for } i = m, m-1, \dots, 2.$$

and

$$R_1(x) = 0$$

for a prescribed value of x . nag_1d_ratnl_eval (e01rbc) is intended to be used to evaluate the continued fraction representation (of an interpolatory rational function) produced by nag_1d_ratnl_interp (e01rac).

4 References

Graves–Morris P R and Hopkins T R (1981) Reliable rational interpolation *Numer. Math.* **36** 111–128

5 Parameters

- 1: **m** – Integer *Input*
On entry: m , the number of terms in the continued fraction.
Constraint: $m \geq 1$.
- 2: **a[m]** – const double *Input*
On entry: **a**[$j-1$] must be set to the value of the parameter a_j in the continued fraction, for $j = 1, 2, \dots, m$.
- 3: **u[m]** – const double *Input*
On entry: **u**[$j-1$] must be set to the value of the parameter u_j in the continued fraction, for $j = 1, 2, \dots, m-1$. (The element **u**[m] is not used).
- 4: **x** – double *Input*
On entry: the value of x at which the continued fraction is to be evaluated.
- 5: **f** – double * *Output*
On exit: the value of the continued fraction corresponding to the value of x .

6: **fail** – NagError *

Input/Output

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_POLE_PRESENT

x corresponds to a pole of $R(x)$, or is very close. $x = \langle value \rangle$.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ had an illegal value.

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.

7 Accuracy

See Section 7 of the document for nag_ld_ratnl_interp (e01rac).

8 Further Comments

The time taken by the function is approximately proportional to m .

9 Example

This example program reads in the parameters a_j and u_j of a continued fraction (as determined by the example for nag_ld_ratnl_interp (e01rac)) and evaluates the continued fraction at a point x .

9.1 Program Text

```
/* nag_ld_ratnl_eval (e01rbc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nage01.h>

int main(void)
{
    /* Scalars */
    double f, x;
    Integer exit_status, i, m;
    NagError fail;

    /* Arrays */
    double *a = 0, *u = 0;

    exit_status = 0;

    INIT_FAIL(fail);
    Vprintf("e01rbc Example Program Results\n");

    /* Skip heading in data file */
    Vscanf("%s[^\n] ");
    m = 4;
```

```

/* Allocate memory */
if ( !(a = NAG_ALLOC(m, double)) ||
     !(u = NAG_ALLOC(m, double)) )
{
    Vprintf("Allocation failure\n");
    exit_status = -1;
    goto END;
}

for (i = 1; i <= m; ++i)
    Vscanf("%lf", &a[i-1]);
Vscanf("%*[^\\n] ");

for (i = 1; i <= m - 1; ++i)
    Vscanf("%lf", &u[i-1]);
Vscanf("%*[^\\n] ");

Vscanf("%lf%*[^\\n] ", &x);

Vprintf("\n");
Vprintf("x = %11.4e\n", x);

e01rbc(m, a, u, x, &f, &fail);

Vprintf("\n");
Vprintf("The value of R(x) is %12.4e\n", f);

END:
if (a) NAG_FREE(a);
if (u) NAG_FREE(u);

return exit_status;
}

```

9.2 Program Data

```

e01rbc Example Program Data
4.000  1.000  0.750 -1.000
0.000  3.000  1.000
6.000

```

9.3 Program Results

```

e01rbc Example Program Results

x =  6.0000e+00

The value of R(x) is  1.7714e+01

```
