NAG Library Function Document

nag_elliptic_integral_rc (s21bac)

1 Purpose

nag_elliptic_integral_rc (s21bac) returns a value of an elementary integral, which occurs as a degenerate case of an elliptic integral of the first kind.

2 Specification

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

```
double nag_elliptic_integral_rc (double x, double y, NagError *fail)
```

3 Description

nag_elliptic_integral_rc (s21bac) calculates an approximate value for the integral

$$R_C(x,y) = \frac{1}{2}\int_0^\infty \frac{dt}{(t+y).\sqrt{t+x}}$$

where $x \geq 0$ and $y \neq 0$.

This function, which is related to the logarithm or inverse hyperbolic functions for $y < x$ and to inverse circular functions if $x < y$, arises as a degenerate form of the elliptic integral of the first kind. If $y < 0$, the result computed is the Cauchy principal value of the integral.

The basic algorithm, which is due to Carlson (1979) and Carlson (1988), is to reduce the arguments recursively towards their mean by the system:

$$\begin{aligned} x_0 &= x & y_0 &= y \\ \mu_n &= (x_n + 2y_n)/3, & S_n &= (y_n - x_n)/3\mu_n \\ & & \lambda_n &= y_n + 2\sqrt{x_n y_n} \\ x_{n+1} &= (x_n + \lambda_n)/4, & y_{n+1} &= (y_n + \lambda_n)/4. \end{aligned}$$

The quantity $|S_n|$ for $n = 0, 1, 2, 3, \ldots$ decreases with increasing n, eventually $|S_n| \sim 1/4^n$. For small enough S_n the required function value can be approximated by the first few terms of the Taylor series about the mean. That is

$$R_C(x,y) = \left(1 + \frac{3S_n^2}{10} + \frac{S_n^3}{7} + \frac{3S_n^4}{8} + \frac{9S_n^5}{22}\right) / \sqrt{\mu_n}.$$

The truncation error involved in using this approximation is bounded by $16|S_n|^6/(1 - 2|S_n|)$ and the recursive process is stopped when S_n is small enough for this truncation error to be negligible compared to the ***machine precision***.

Within the domain of definition, the function value is itself representable for all representable values of its arguments. However, for values of the arguments near the extremes the above algorithm must be modified so as to avoid causing underflows or overflows in intermediate steps. In extreme regions arguments are prescaled away from the extremes and compensating scaling of the result is done before returning to the calling program.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

Carlson B C (1979) Computing elliptic integrals by duplication *Numerische Mathematik* **33** 1–16

Carlson B C (1988) A table of elliptic integrals of the third kind *Math. Comput.* **51** 267–280

5 Arguments

1: **x** – double *Input*
2: **y** – double *Input*

On entry: the arguments x and y of the function, respectively.

Constraint: $\mathbf{x} \geq 0.0$ and $\mathbf{y} \neq 0.0$.

3: **fail** – NagError * *Input/Output*

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

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

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.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_REAL_ARG_EQ

On entry, $\mathbf{y} = 0.0$.
Constraint: $\mathbf{y} \neq 0.0$.
The function is undefined and returns zero.

NE_REAL_ARG_LT

On entry, $\mathbf{x} = \langle value \rangle$.
Constraint: $\mathbf{x} \geq 0.0$.
The function is undefined.

7 Accuracy

In principle the function is capable of producing full ***machine precision***. However round-off errors in internal arithmetic will result in slight loss of accuracy. This loss should never be excessive as the algorithm does not involve any significant amplification of round-off error. It is reasonable to assume that the result is accurate to within a small multiple of the ***machine precision***.

8 Parallelism and Performance

Not applicable.

9 Further Comments

You should consult the s Chapter Introduction which shows the relationship of this function to the classical definitions of the elliptic integrals.

10 Example

This example simply generates a small set of nonextreme arguments which are used with the function to produce the table of low accuracy results.

10.1 Program Text

```
/* nag_elliptic_integral_rc (s21bac) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

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

int main(void)
{
  Integer  exit_status = 0;
  double   rc, x, y;
  Integer  ix;
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_elliptic_integral_rc (s21bac) Example Program Results\n");
  printf("     x      y       nag_elliptic_integral_rc (s21bac)  \n");
  for (ix = 1; ix <= 3; ix++)
    {
      x = ix*0.5;
      y = 1.0;
      /* nag_elliptic_integral_rc (s21bac).
       * Degenerate symmetrised elliptic integral of 1st kind
       * R_C(xy)
       */
      rc = nag_elliptic_integral_rc(x, y, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_elliptic_integral_rc (s21bac).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%7.2f%7.2f%12.4f\n", x, y, rc);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_elliptic_integral_rc (s21bac) Example Program Results
     x      y       nag_elliptic_integral_rc (s21bac)
   0.50   1.00      1.1107
   1.00   1.00      1.0000
   1.50   1.00      0.9312
```