nag_elliptic_integral_rd (s21bcc)

1. Purpose

nag_elliptic_integral_rd (s21bcc) returns a value of the symmetrised elliptic integral of the second kind.

2. Specification

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

double nag_elliptical_integral_rd(double x, double y, double z, NagError *fail)
```

3. Description

This function calculates an approximate value for the integral

$$R_D(x,y,z) = \frac{3}{2}\int_0^\infty \frac{dt}{\sqrt{(t+x)(t+y)(t+z)^3}}$$

where x, $y \geq 0$, at most one of x and y is zero, and $z > 0$.

The basic algorithm, which is due to Carlson (1978) and Carlson (1988), is to reduce the arguments recursively towards their mean by the rule:

$$\begin{array}{rcl}
x_0 & = & x, y_0 = y, z_0 = z \\
\mu_n & = & (x_n + y_n + 3z_n)/5 \\
X_n & = & 1 - x_n/\mu_n \\
Y_n & = & 1 - y_n/\mu_n \\
Z_n & = & 1 - z_n/\mu_n \\
\lambda_n & = & \sqrt{x_n y_n} + \sqrt{y_n z_n} + \sqrt{z_n x_n} \\
x_{n+1} & = & (x_n + \lambda_n)/4 \\
y_{n+1} & = & (y_n + \lambda_n)/4 \\
z_{n+1} & = & (z_n + \lambda_n)/4
\end{array}$$

For n sufficiently large,

$$\epsilon_n = \max(|X_n|, |Y_n|, |Z_n|) \sim 1/4^n$$

and the function may be approximated adequately by a 5th-order power series

$$\begin{array}{rcl}
R_D(x,y,z) & = & 3\sum_{m=0}^{n-1} \frac{4^{-m}}{(z_m+\lambda_n)\sqrt{z_m}} \\
 & + & \frac{4^{-n}}{\sqrt{\mu_n^3}}\left(1 + \frac{3}{7}S_n^{(2)} + \frac{1}{3}S_n^{(3)} + \frac{3}{22}(S_n^{(2)})^2 + \frac{3}{11}S_n^{(4)} + \frac{3}{13}S_n^{(2)}S_n^{(3)} + \frac{3}{13}S_n^{(5)}\right),
\end{array}$$

where $S_n^{(m)} = \left(X_n^m + Y_n^m + 3Z_n^m\right)/2m$.

The truncation error in this expansion is bounded by $3\epsilon_n^6/\sqrt{(1-\epsilon_n)^3}$ and the recursive process is terminated when this quantity is negligible compared with the ***machine precision***.

The function may fail either because it has been called with arguments outside the domain of definition, or with arguments so extreme that there is an unavoidable danger of setting underflow or overflow.

Note: $R_D(x,x,x) = x^{-3/2}$, so there exists a region of extreme arguments for which the function value is not representable.

4. Parameters

x
y
z

Input: the arguments x, y and z of the function.
Constraint: **x**, **y** ≥ 0.0, **z** > 0.0 and only one of **x** and **y** may be zero.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_REAL_ARG_LT

On entry, **x** must not be less than 0.0: **x** = ⟨*value*⟩.
On entry, **y** must not be less than 0.0: **y** = ⟨*value*⟩.
The function is undefined. On entry, ⟨*parameters*⟩ must not be less than ⟨*value*⟩: ⟨*parameters*⟩ = ⟨*value*⟩.
On entry, either **z** is too close to zero or both **x** and **y** are too close to zero: there is a danger of setting overflow.

NE_REAL_ARG_EQ

On entry, **x+y** must not be equal to 0.0: **x+y** = ⟨*value*⟩.
Both **x** and **y** are zero and the function is undefined.

NE_REAL_ARG_LE

On entry, **z** must not be less than or equal to 0.0: **z** = ⟨*value*⟩.
The function is undefined.

NE_REAL_ARG_GE

On entry, **x** must not be greater than or equal to ⟨*value*⟩: **x** = ⟨*value*⟩.
On entry, **y** must not be greater than or equal to ⟨*value*⟩: **y** = ⟨*value*⟩.
On entry, **z** must not be greater than or equal to ⟨*value*⟩: **z** = ⟨*value*⟩.
There is a danger of setting underflow and the function returns zero.

6. Further Comments

Symmetrised elliptic integrals returned by functions nag_elliptic_integral_rd, nag_elliptic_integral_rc (s21bac), nag_elliptic_integral_rf (s21bbc) and nag_elliptic_integral_rj (s21bdc) can be related to the more traditional canonical forms (see Abramowitz and Stegun (1968)), as described in the Chapter Introduction.

6.1. 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***.

6.2. References

Abramowitz M and Stegun I A (1968) *Handbook of Mathematical Functions* Dover Publications, New York ch 17.
Carlson B C (1978) *Computing Elliptic Integrals by Duplication* Department of Physics, Iowa State University (Preprint).
Carlson B C (1988) A Table of Elliptic Integrals of the Third Kind *Math. Comp.* **51** 267–280.

7. See Also

nag_elliptic_integral_rc (s21bac)
nag_elliptic_integral_rf (s21bbc)
nag_elliptic_integral_rj (s21bdc)

8. Example

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

8.1. Program Text

```
/* nag_elliptic_integral_rd(s21bcc) Example Program
 *
```

```
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

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

main()
{
  double rd, x, y, z;
  Integer ix, iy;

  Vprintf("s21bcc Example Program Results\n");
  Vprintf ("      x      y      z       s21bcc  \n");
  for ( ix=1; ix<=3; ix++)
    {
      x = ix*0.5;
      for ( iy=ix; iy<=3; iy++)
        {
          y = iy*0.5;
          z = 1.0;
          rd = s21bcc(x, y, z, NAGERR_DEFAULT);
          Vprintf (" %7.2f%7.2f%7.2f%12.4f\n", x, y, z, rd);
        }
    }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

None.

8.3. Program Results

```
s21bcc Example Program Results
      x      y      z       s21bcc
    0.50   0.50   1.00      1.4787
    0.50   1.00   1.00      1.2108
    0.50   1.50   1.00      1.0611
    1.00   1.00   1.00      1.0000
    1.00   1.50   1.00      0.8805
    1.50   1.50   1.00      0.7775
```