NAG Library Function Document

nag_elliptic_integral_rf (s21bbc)

1 Purpose

nag_elliptic_integral_rf (s21bbc) returns a value of the symmetrised elliptic integral of the first kind.

2 Specification

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

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

3 Description

nag_elliptic_integral_rf (s21bbc) calculates an approximation to the integral

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

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

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

$x_0 = \min(x,y,z), \quad z_0 = \max(x,y,z),$

$y_0 =$ remaining third intermediate value argument.

(This ordering, which is possible because of the symmetry of the function, is done for technical reasons related to the avoidance of overflow and underflow.)

$$\begin{array}{rcl}
\mu_n & = & (x_n + y_n + z_n)/3 \\
X_n & = & (1 - x_n)/\mu_n \\
Y_n & = & (1 - y_n)/\mu_n \\
Z_n & = & (1 - z_n)/\mu_n \\
\lambda_n & = & \sqrt{x_n y_n} + \sqrt{y_n z_n} + \sqrt{z_n x_n} \\
x_{n+1} & = & (x_n + \lambda_n)/4 \\
y_{n+1} & = & (y_n + \lambda_n)/4 \\
z_{n+1} & = & (z_n + \lambda_n)/4
\end{array}$$

$\epsilon_n = \max(|X_n|, |Y_n|, |Z_n|)$ and the function may be approximated adequately by a fifth order power series:

$$R_F(x,y,z) = \frac{1}{\sqrt{\mu_n}}\left(1 - \frac{E_2}{10} + \frac{E_2^2}{24} - \frac{3E_2E_3}{44} + \frac{E_3}{14}\right)$$

where $E_2 = X_nY_n + Y_nZ_n + Z_nX_n$, $E_3 = X_nY_nZ_n$.

The truncation error involved in using this approximation is bounded by $\epsilon_n^6/4(1 - \epsilon_n)$ and the recursive process is stopped when this truncation error is negligible compared with 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*

3: **z** – double *Input*

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

Constraint: **x**, **y**, **z** ≥ 0.0 and only one of **x**, **y** and **z** may be zero.

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

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

6 Error Indicators and Warnings

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_ARG_EQ

On entry, **x** $= \langle$*value*$\rangle$, **y** $= \langle$*value*$\rangle$ and **z** $= \langle$*value*$\rangle$.
Constraint: at most one of **x**, **y** and **z** is 0.0.
The function is undefined and returns zero.

NE_REAL_ARG_LT

On entry, **x** $= \langle$*value*$\rangle$, **y** $= \langle$*value*$\rangle$ and **z** $= \langle$*value*$\rangle$.
Constraint: **x** ≥ 0.0 and **y** ≥ 0.0 and **z** ≥ 0.0.
The function is undefined.

7 Accuracy

In principle nag_elliptic_integral_rf (s21bbc) 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.

If two arguments are equal, the function reduces to the elementary integral R_C, computed by nag_elliptic_integral_rc (s21bac).

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_rf (s21bbc) Example Program.
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 *
 * Mark 3 revised, 1994.
 */

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

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

  INIT_FAIL(fail);

  printf("nag_elliptic_integral_rf (s21bbc) Example Program Results\n");
  printf("     x      y      z       nag_elliptic_integral_rf\n");
  for (ix = 1; ix <= 3; ix++)
    {
      x = ix*0.5;
      y = (ix+1)*0.5;
      z = (ix+2)*0.5;
      /* nag_elliptic_integral_rf (s21bbc).
       * Symmetrised elliptic integral of 1st kind R_F(xyz)
       */
      rf = nag_elliptic_integral_rf(x, y, z, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_elliptic_integral_rf (s21bbc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%7.2f%7.2f%7.2f%12.4f\n", x, y, z, rf);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_elliptic_integral_rf (s21bbc) Example Program Results
     x      y      z       nag_elliptic_integral_rf
   0.50   1.00   1.50      1.0281
   1.00   1.50   2.00      0.8260
   1.50   2.00   2.50      0.7116
```