NAG Library Routine Document

S21BAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S21BAF returns a value of an elementary integral, which occurs as a degenerate case of an elliptic integral of the first kind, via the function name.

2 Specification

```
FUNCTION S21BAF (X, Y, IFAIL)
REAL (KIND=nag_wp) S21BAF
INTEGER            IFAIL
REAL (KIND=nag_wp) X, Y
```

3 Description

S21BAF 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 Parameters

1: X – REAL (KIND=nag_wp) *Input*
2: Y – REAL (KIND=nag_wp) *Input*

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

Constraint: $\mathrm{X} \geq 0.0$ and $\mathrm{Y} \neq 0.0$.

3: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{X} < 0.0$; the function is undefined.

$\mathrm{IFAIL} = 2$

On entry, $\mathrm{Y} = 0.0$; the function is undefined.

On soft failure the routine returns zero.

7 Accuracy

In principle the routine 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 Further Comments

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

9 Example

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

9.1 Program Text

```
    Program s21bafe

!     S21BAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s21baf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rc, x, y
      Integer                          :: ifail, ix
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'S21BAF Example Program Results'

      Write (nout,*)
      Write (nout,*) '    X      Y        S21BAF'
      Write (nout,*)

data: Do ix = 1, 3
        x = real(ix,kind=nag_wp)*0.5E0_nag_wp
        y = 1.0E0_nag_wp

        ifail = -1
        rc = s21baf(x,y,ifail)

        If (ifail<0) Then
          Exit data
        End If

        Write (nout,99999) x, y, rc
      End Do data

99999 Format (1X,2F7.2,F12.4)
    End Program s21bafe
```

9.2 Program Data

None.

9.3 Program Results

```
 S21BAF Example Program Results

     X      Y        S21BAF

    0.50   1.00      1.1107
    1.00   1.00      1.0000
    1.50   1.00      0.9312
```