NAG Library Routine Document

S21BBF

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

S21BBF returns a value of the symmetrised elliptic integral of the first kind, via the function name.

2 Specification

```
FUNCTION S21BBF (X, Y, Z, IFAIL)
REAL (KIND=nag_wp) S21BBF

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

3 Description

S21BBF 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.)

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

$\epsilon_n=\max(|X_n|,|Y_n|,|Z_n|)$ 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 – REAL (KIND=nag_wp) *Input*
2: Y – REAL (KIND=nag_wp) *Input*
3: Z – REAL (KIND=nag_wp) *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: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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: 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 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:

IFAIL $= 1$

On entry, one or more of X, Y and Z is negative; the function is undefined.

IFAIL $= 2$

On entry, two or more of X, Y and Z are zero; the function is undefined. On softfailure, the routine returns zero.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

In principle S21BBF 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

S21BBF is not threaded in any implementation.

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 S21BAF.

10 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.

10.1 Program Text

```
    Program s21bbfe

!     S21BBF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s21bbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rf, x, y, z
      Integer                          :: ifail, ix
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'S21BBF Example Program Results'

      Write (nout,*)
      Write (nout,*) '    X      Y      Z        S21BBF'
      Write (nout,*)

data: Do ix = 1, 3
        x = real(ix,kind=nag_wp)*0.5E0_nag_wp
        y = real(ix+1,kind=nag_wp)*0.5E0_nag_wp
        z = real(ix+2,kind=nag_wp)*0.5E0_nag_wp

        ifail = -1
        rf = s21bbf(x,y,z,ifail)

        If (ifail<0) Then
          Exit data
        End If

        Write (nout,99999) x, y, z, rf
      End Do data

99999 Format (1X,3F7.2,F12.4)
    End Program s21bbfe
```

10.2 Program Data

None.

10.3 Program Results

```
 S21BBF Example Program Results

     X      Y      Z        S21BBF

    0.50   1.00   1.50      1.0281
    1.00   1.50   2.00      0.8260
    1.50   2.00   2.50      0.7116
```