NAG Fortran 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 routine name.

2 Specification

```
real FUNCTION S21BBF(X, Y, Z, IFAIL)
INTEGER            IFAIL
real               X, Y, Z
```

3 Description

This routine 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 (1978) 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.)

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

$\epsilon_n = \max(|X_n|,|Y_n|,|Z_n|)$ and the function may be approximated adequately by a 5th 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 pre-scaled 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 (1978) Computing elliptic integrals by duplication *Preprint* Department of Physics, Iowa State University

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

5 Parameters

1: X – ***real*** *Input*
2: Y – ***real*** *Input*
3: Z – ***real*** *Input*

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

Constraint: X, Y, Z $\geq$ 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. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $=0$ unless the routine detects an error (see Section 6).

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, for users 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.**

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 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

Users 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.

9 Example

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

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     S21BBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             RF, X, Y, Z
      INTEGER          IFAIL, IX
*     .. External Functions ..
      real             S21BBF
      EXTERNAL         S21BBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S21BBF Example Program Results'
      WRITE (NOUT,*)
      WRITE (NOUT,*) '    X      Y      Z        S21BBF  IFAIL'
      WRITE (NOUT,*)
      DO 20 IX = 1, 3
         X = IX*0.5e0
         Y = (IX+1)*0.5e0
         Z = (IX+2)*0.5e0
         IFAIL = 1
*
         RF = S21BBF(X,Y,Z,IFAIL)
*
         WRITE (NOUT,99999) X, Y, Z, RF, IFAIL
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,3F7.2,F12.4,I5)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 S21BBF Example Program Results

     X      Y      Z        S21BBF  IFAIL

    0.50   1.00   1.50      1.0281    0
    1.00   1.50   2.00      0.8260    0
    1.50   2.00   2.50      0.7116    0
```