NAG Library Routine Document

S21DAF

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

S21DAF returns the value of the general elliptic integral of the second kind $F(z, k', a, b)$ for a complex argument z, via the function name.

2 Specification

```
FUNCTION S21DAF (Z, AKP, A, B, IFAIL)
COMPLEX (KIND=nag_wp) S21DAF
INTEGER               IFAIL
REAL (KIND=nag_wp)    AKP, A, B
COMPLEX (KIND=nag_wp) Z
```

3 Description

S21DAF evaluates an approximation to the general elliptic integral of the second kind $F(z, k', a, b)$ given by

$$F(z, k', a, b) = \int_0^z \frac{a + b\zeta^2}{(1 + \zeta^2)\sqrt{(1 + \zeta^2)\left(1 + k'^2\zeta^2\right)}} d\zeta,$$

where a and b are real parameters, z is a complex argument whose real part is non-negative and k' is a real parameter (the *complementary modulus*). The evaluation of F is based on the Gauss transformation. Further details, in particular for the conformal mapping provided by F, can be found in Bulirsch (1960).

Special values include

$$F(z, k', 1, 1) = \int_0^z \frac{d\zeta}{\sqrt{(1 + \zeta^2)(1 + k'2\zeta^2)}},$$

or $F_1(z, k')$ (the *elliptic integral of the first kind*) and

$$F\left(z, k', 1, k'^2\right) = \int_0^z \frac{\sqrt{1 + k'^2\zeta^2}}{(1 + \zeta^2)\sqrt{1 + \zeta^2}} d\zeta,$$

or $F_2(z, k')$ (the *elliptic integral of the second kind*). Note that the values of $F_1(z, k')$ and $F_2(z, k')$ are equal to $\tan^{-1}(z)$ in the trivial case $k' = 1$.

S21DAF is derived from an Algol 60 procedure given by Bulirsch (1960). Constraints are placed on the values of z and k' in order to avoid the possibility of machine overflow.

4 References

Bulirsch R (1960) Numerical calculation of elliptic integrals and elliptic functions *Numer. Math.* **7** 76–90

5 Parameters

1: Z – COMPLEX (KIND=nag_wp) *Input*

On entry: the argument z of the function.

Constraints:

$0.0 \leq \text{Z}.\textbf{re} \leq \lambda$;
$\text{abs}(\text{Z}.\textbf{im}) \leq \lambda$, where $\lambda^6 = 1/\text{X02AMF}$.

2: AKP – REAL (KIND=nag_wp) *Input*

On entry: the argument k' of the function.

Constraint: $\text{abs}(\text{AKP}) \leq \lambda$.

3: A – REAL (KIND=nag_wp) *Input*

On entry: the argument a of the function.

4: B – REAL (KIND=nag_wp) *Input*

On entry: the argument b of the function.

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

6 Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{Re}(\text{Z}) < 0.0$,
or $\text{Re}(\text{Z}) > \lambda$,
or $|\text{Im}(\text{Z})| > \lambda$,
or $|\text{AKP}| > \lambda$, where $\lambda^6 = 1/\text{X02AMF}$.

$\text{IFAIL} = 2$

The iterative procedure used to evaluate the integral has failed to converge. The result is returned as zero.

7 Accuracy

In principle the routine is capable of achieving full relative precision in the computed values. However, the accuracy obtainable in practice depends on the accuracy of the standard elementary functions such as atan2 and log.

8 Further Comments

None.

9 Example

This example evaluates the elliptic integral of the first kind $F_1(z,k')$ given by

$$F_1(z,k') = \int_0^z \frac{d\zeta}{\sqrt{(1+\zeta^2)\left(1+k'^2\zeta^2\right)}},$$

where $z = 1.2 + 3.7i$ and $k' = 0.5$, and prints the results.

9.1 Program Text

```
    Program s21dafe

!     S21DAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s21daf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                 :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)              :: y, z
      Real (Kind=nag_wp)                 :: a, akp, b
      Integer                            :: ifail
!     .. Executable Statements ..
      Write (nout,*) 'S21DAF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Write (nout,*)
      Write (nout,*) '     Z          AKP     A', '      B                 Y'
      Write (nout,*)

      Read (nin,*) z, akp, a, b

      ifail = -1
      y = s21daf(z,akp,a,b,ifail)

      If (ifail>=0) Then
        Write (nout,99999) z, akp, a, b, y
      End If

99999 Format (1X,'(',F4.1,',',F4.1,' )',3F7.1,3X,'(',1P,E12.4,',',E12.4,' )')
    End Program s21dafe
```

9.2 Program Data

```
S21DAF Example Program Data
 (1.2, 3.7)  0.5  1.0  1.0 : Values of Z, AKP, A and B
```

9.3 Program Results

```
 S21DAF Example Program Results

      Z          AKP     A      B                 Y

 ( 1.2, 3.7 )    0.5    1.0    1.0    (  1.9713E+00,  5.0538E-01 )
```