S18ADF – NAG Fortran Library Routine Document

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

S18ADF returns the value of the modified Bessel Function $K_1(x)$, via the routine name.

2 Specification

```
real FUNCTION S18ADF(X, IFAIL)
INTEGER              IFAIL
real                 X
```

3 Description

This routine evaluates an approximation to the modified Bessel Function of the second kind $K_1(x)$.

Note. $K_1(x)$ is undefined for $x \leq 0$ and the routine will fail for such arguments.

The routine is based on five Chebyshev expansions:

For $0 < x \leq 1$,

$$K_1(x) = \frac{1}{x} + x \ln x \sum_{r=0}^{\prime} a_r T_r(t) - x \sum_{r=0}^{\prime} b_r T_r(t) \ , \text{ where } t = 2x^2 - 1;$$

For $1 < x \leq 2$,

$$K_1(x) = e^{-x} \sum_{r=0}^{\prime} c_r T_r(t) \ , \text{ where } t = 2x - 3;$$

For $2 < x \leq 4$,

$$K_1(x) = e^{-x} \sum_{r=0}^{\prime} d_r T_r(t) \ , \text{ where } t = x - 3;$$

For $x > 4$,

$$K_1(x) = \frac{e^{-x}}{\sqrt{x}} \sum_{r=0}^{\prime} e_r T_r(t) \ , \text{ where } t = \frac{9 - x}{1 + x}.$$

For x near zero, $K_1(x) \simeq \frac{1}{x}$. This approximation is used when x is sufficiently small for the result to be correct to ***machine precision***. For very small x on some machines, it is impossible to calculate $\frac{1}{x}$ without overflow and the routine must fail.

For large x, where there is a danger of underflow due to the smallness of K_1, the result is set exactly to zero.

4 References

[1] Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* Dover Publications (3rd Edition)

5 Parameters

1: X — ***real*** *Input*

On entry: the argument x of the function.

Constraint: X > 0.0.

2: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

X ≤ 0.0, K_1 is undefined. On soft failure the routine returns zero.

IFAIL = 2

X is too small, there is a danger of overflow. On soft failure the routine returns approximately the largest representable value.

7 Accuracy

Let δ and ϵ be the relative errors in the argument and result respectively.

If δ is somewhat larger than the ***machine precision*** (i.e., if δ is due to data errors etc.), then ϵ and δ are approximately related by:

$$\epsilon \simeq \left| \frac{xK_0(x) - K_1(x)}{K_1(x)} \right| \delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left| \frac{xK_0(x) - K_1(x)}{K_1(x)} \right|.$$

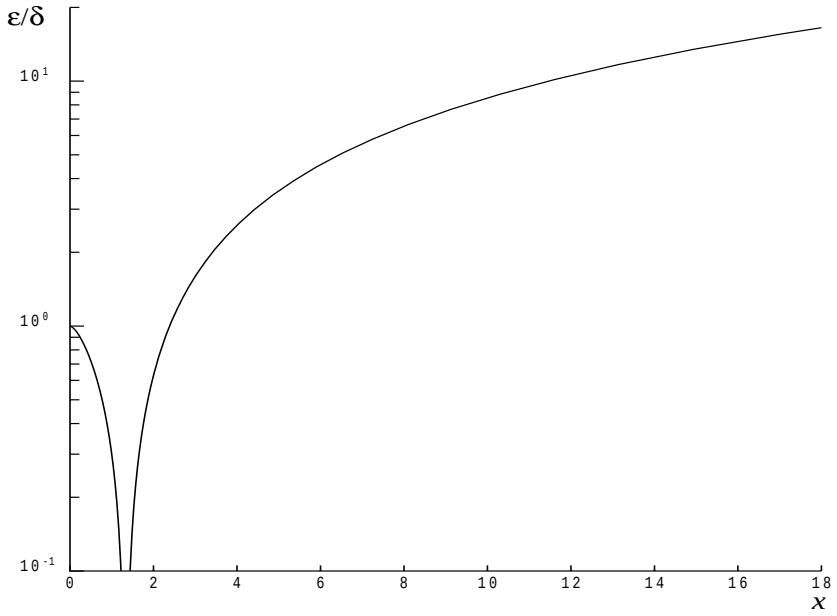

Figure 1

However if δ is of the same order as the ***machine precision***, then rounding errors could make ϵ slightly larger than the above relation predicts.

For small x, $\epsilon \simeq \delta$ and there is no amplification of errors.

For large x, $\epsilon \simeq x\delta$ and we have strong amplification of the relative error. Eventually K_1, which is asymptotically given by $\frac{e^{-x}}{\sqrt{x}}$, becomes so small that it cannot be calculated without underflow and hence the routine will return zero. Note that for large x the errors will be dominated by those of the Fortran intrinsic function EXP.

8 Further Comments

None.

9 Example

The example program reads values of the argument x from a file, evaluates the function at each value of x and prints the 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.

```
*     S18ADF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             X, Y
      INTEGER          IFAIL
*     .. External Functions ..
      real             S18ADF
      EXTERNAL         S18ADF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S18ADF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*) '     X           Y        IFAIL'
      WRITE (NOUT,*)
   20 READ (NIN,*,END=40) X
      IFAIL = 1
*
      Y = S18ADF(X,IFAIL)
*
      WRITE (NOUT,99999) X, Y, IFAIL
      GO TO 20
   40 STOP
*
99999 FORMAT (1X,1P,2e12.3,I7)
      END
```

9.2 Program Data

```
S18ADF Example Program Data
             0.0
             0.4
             0.6
             1.4
             1.6
             2.5
             3.5
             6.0
             8.0
             10.0
             -1.0
             1000.0
```

9.3 Program Results

```
 S18ADF Example Program Results

      X           Y        IFAIL

   0.000E+00   0.000E+00      1
   4.000E-01   2.184E+00      0
   6.000E-01   1.303E+00      0
   1.400E+00   3.208E-01      0
   1.600E+00   2.406E-01      0
   2.500E+00   7.389E-02      0
   3.500E+00   2.224E-02      0
   6.000E+00   1.344E-03      0
   8.000E+00   1.554E-04      0
   1.000E+01   1.865E-05      0
  -1.000E+00   0.000E+00      1
   1.000E+03   0.000E+00      0
```