NAG Fortran Library Routine Document

S18ACF

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

S18ACF returns the value of the modified Bessel Function $K_0(x)$, via the routine name.

2 Specification

```
real FUNCTION S18ACF(X, IFAIL)
INTEGER            IFAIL
real               X
```

3 Description

This routine evaluates an approximation to the modified Bessel Function of the second kind $K_0(x)$.

Note: $K_0(x)$ is undefined for $x \le 0$ and the routine will fail for such arguments.

The routine is based on five Chebyshev expansions:

For $0 < x \le 1$,

$$K_0(x) = -\ln x \sum_{r=0}{}' a_r T_r(t) + \sum_{r=0}{}' b_r T_r(t), \quad \text{where } t = 2x^2 - 1.$$

For $1 < x \le 2$,

$$K_0(x) = e^{-x} \sum_{r=0}{}' c_r T_r(t), \quad \text{where } t = 2x - 3.$$

For $2 < x \le 4$,

$$K_0(x) = e^{-x} \sum_{r=0}{}' d_r T_r(t), \quad \text{where } t = x - 3.$$

For $x > 4$,

$$K_0(x) = \frac{e^{-x}}{\sqrt{x}} \sum_{r=0}{}' e_r T_r(t), \text{ where } t = \frac{9-x}{1+x}.$$

For x near zero, $K_0(x) \simeq -\gamma - \ln\left(\frac{x}{2}\right)$, where γ denotes Euler's constant. This approximation is used when x is sufficiently small for the result to be correct to ***machine precision***.

For large x, where there is a danger of underflow due to the smallness of K_0, the result is set exactly to zero.

4 References

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

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. 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 $= 1$

$X \leq 0.0$, K_0 is undefined. On soft failure the routine returns zero.

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:

$$\epsilon \simeq \left|\frac{xK_1(x)}{K_0(x)}\right|\delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left|\frac{xK_1(x)}{K_0(x)}\right|.$$

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

For small x, the amplification factor is approximately $\left|\frac{1}{\ln x}\right|$, which implies strong attenuation of the error, but in general ϵ can never be less than the ***machine precision***.

For large x, $\epsilon \simeq x\delta$ and we have strong amplification of the relative error. Eventually K_0, 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.

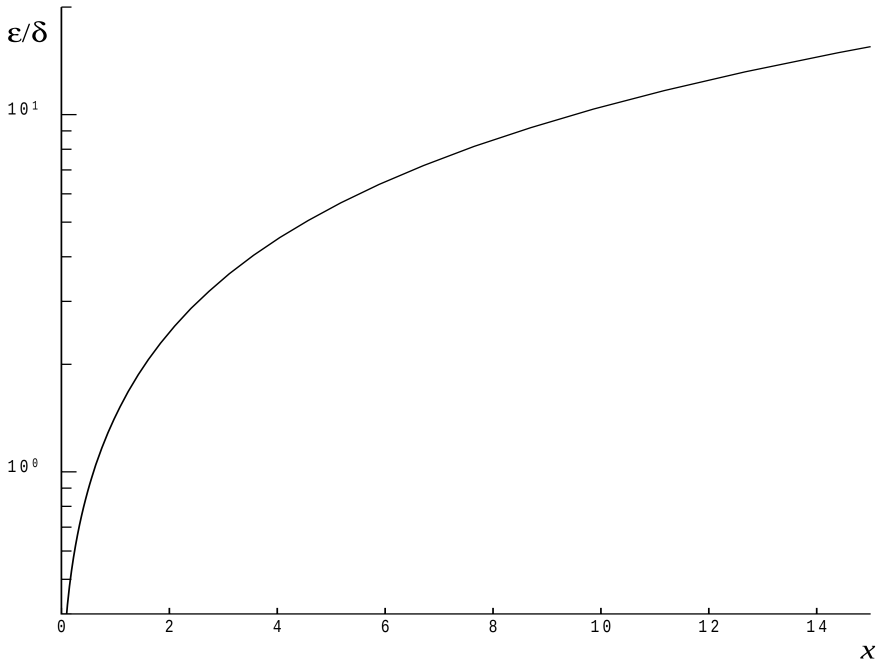

Figure 1

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.

```
*     S18ACF 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             S18ACF
      EXTERNAL         S18ACF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S18ACF 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 = S18ACF(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

```
S18ACF 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

```
 S18ACF Example Program Results

      X           Y        IFAIL

    0.000E+00   0.000E+00      1
    4.000E-01   1.115E+00      0
    6.000E-01   7.775E-01      0
    1.400E+00   2.437E-01      0
    1.600E+00   1.880E-01      0
    2.500E+00   6.235E-02      0
    3.500E+00   1.960E-02      0
    6.000E+00   1.244E-03      0
    8.000E+00   1.465E-04      0
    1.000E+01   1.778E-05      0
   -1.000E+00   0.000E+00      1
    1.000E+03   0.000E+00      0
```