S10ACF – 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

S10ACF returns the value of the hyperbolic cosine, $\cosh x$, via the routine name.

2 Specification

```
real FUNCTION S10ACF(X, IFAIL)
INTEGER              IFAIL
real                 X
```

3 Description

The routine calculates an approximate value for the hyperbolic cosine, $\cosh x$.

For $|x| \leq E_1$, $\quad \cosh x = \frac{1}{2}(e^x + e^{-x})$.

For $|x| > E_1$, the routine fails owing to danger of setting overflow in calculating e^x. The result returned for such calls is $\cosh E_1$, i.e., it returns the result for the nearest valid argument. The value of machine-dependent constant E_1 may be given in the Users' Note for your implementation.

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.

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

The routine has been called with an argument too large in absolute magnitude. There is a danger of overflow. The result returned is the value of $\cosh x$ at the nearest valid argument.

7 Accuracy

If δ and ϵ are the relative errors in the argument and result, respectively, then in principle

$$\epsilon \simeq x \tanh x \times \delta.$$

That is, the relative error in the argument, x, is amplified by a factor, at least $x \tanh x$. The equality should hold if δ is greater than the ***machine precision*** (δ is due to data errors etc.) but if δ is simply a result of round-off in the machine representation of x then it is possible that an extra figure may be lost in internal calculation round-off.

The behaviour of the error amplification factor is shown by the following graph:

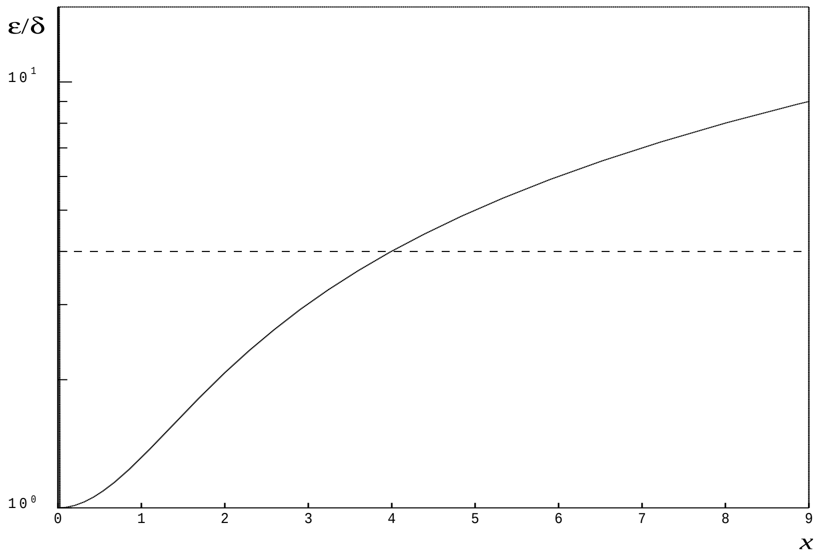

Figure 1

It should be noted that near $x = 0$ where this amplification factor tends to zero the accuracy will be limited eventually by the ***machine precision***. Also for $|x| \geq 2$

$$\epsilon \sim x\delta = \Delta$$

where Δ is the absolute error in the argument x.

8 Further Comments

None.

9 Example

The following 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.

```
*     S10ACF 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             S10ACF
      EXTERNAL         S10ACF
*     .. Executable Statements ..
```

```
      WRITE (NOUT,*) 'S10ACF 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 = S10ACF(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

```
S10ACF Example Program Data
           -10.0
            -0.5
             0.0
             0.5
            25.0
```

9.3 Program Results

```
 S10ACF Example Program Results

      X           Y        IFAIL

   -1.000E+01   1.101E+04      0
   -5.000E-01   1.128E+00      0
    0.000E+00   1.000E+00      0
    5.000E-01   1.128E+00      0
    2.500E+01   3.600E+10      0
```