

NAG Library Routine Document

S11ABF

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

S11ABF returns the value of the inverse hyperbolic sine, $\operatorname{arcsinh} x$, via the function name.

2 Specification

```
FUNCTION S11ABF (X, IFAIL)
REAL (KIND=nag_wp) S11ABF
INTEGER          IFAIL
REAL (KIND=nag_wp) X
```

3 Description

S11ABF calculates an approximate value for the inverse hyperbolic sine of its argument, $\operatorname{arcsinh} x$.

For $|x| \leq 1$ it is based on the Chebyshev expansion

$$\operatorname{arcsinh} x = x \times y(t) = x \sum_{r=0} c_r T_r(t), \quad \text{where } t = 2x^2 - 1.$$

For $|x| > 1$ it uses the fact that

$$\operatorname{arcsinh} x = \operatorname{sign} x \times \ln \left(|x| + \sqrt{x^2 + 1} \right).$$

This form is used directly for $1 < |x| < 10^k$, where $k = n/2 + 1$, and the machine uses approximately n decimal place arithmetic.

For $|x| \geq 10^k$, $\sqrt{x^2 + 1}$ is equal to $|x|$ to within the accuracy of the machine and hence we can guard against premature overflow and, without loss of accuracy, calculate

$$\operatorname{arcsinh} x = \operatorname{sign} x \times (\ln 2 + \ln|x|).$$

4 References

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

5 Arguments

1: X – REAL (KIND=nag_wp) *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. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

None.

7 Accuracy

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

$$|\epsilon| \simeq \left| \frac{x}{\sqrt{1+x^2} \operatorname{arcsinh} x} \delta \right|.$$

That is, the relative error in the argument, x , is amplified by a factor at least $\frac{x}{\sqrt{1+x^2} \operatorname{arcsinh} x}$, in the result.

The equality should hold if δ is greater than the *machine precision* (δ due to data errors etc.) but if δ is simply due to round-off in the machine representation it is possible that an extra figure may be lost in internal calculation round-off.

The behaviour of the amplification factor is shown in the following graph:

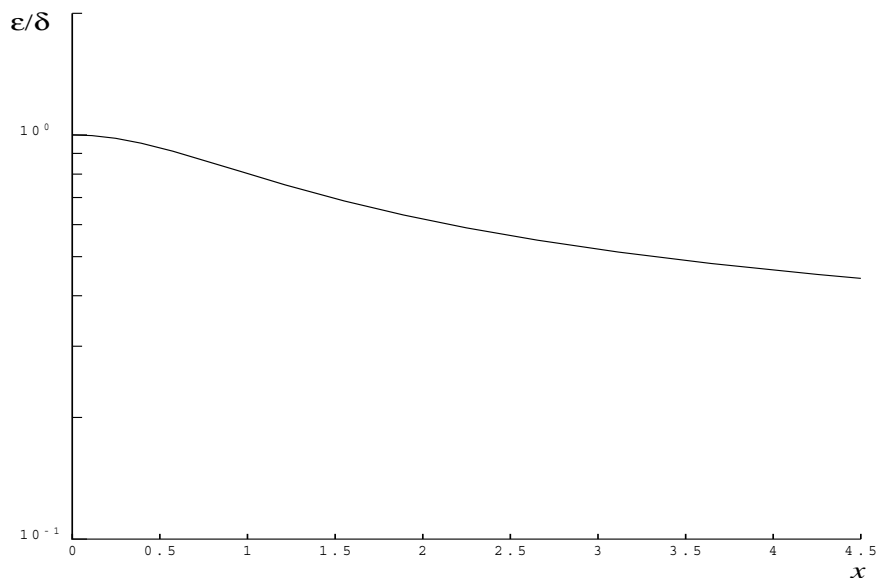


Figure 1

It should be noted that this factor is always less than or equal to one. For large x we have the absolute error in the result, E , in principle, given by

$$E \sim \delta.$$

This means that eventually accuracy is limited by *machine precision*.

8 Parallelism and Performance

S11ABF is not threaded in any implementation.

9 Further Comments

None.

10 Example

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

10.1 Program Text

```

Program s11abfe

!      S11ABF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
!      Use nag_library, Only: nag_wp, s11abf
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)         :: x, y
!      Integer                     :: ifail, ioerr
!
!      .. Executable Statements ..
!      Write (nout,*) 'S11ABF Example Program Results'

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

!
!      Write (nout,*)
!      Write (nout,*) '      X      Y'
!      Write (nout,*)

data: Do
!      Read (nin,*,Iostat=ioerr) x

!
!      If (ioerr<0) Then
!          Exit data
!      End If

!
!      ifail = 0
!      y = s11abf(x,ifail)

!      Write (nout,99999) x, y
!      End Do data

99999 Format (1X,1P,2E12.3)
End Program s11abfe

```

10.2 Program Data

```

S11ABF Example Program Data
      -2.0
      -0.5
      1.0
      6.0

```

10.3 Program Results

S11ABF Example Program Results

X	Y
-2.000E+00	-1.444E+00
-5.000E-01	-4.812E-01
1.000E+00	8.814E-01
6.000E+00	2.492E+00
