NAG Toolbox

nag_specfun_arccosh (s11ac)

1 Purpose

nag_specfun_arccosh (s11ac) returns the value of the inverse hyperbolic cosine, $\operatorname{arccosh} x$, via the function name. The result is in the principal positive branch.

2 Syntax

```
[result, ifail] = nag_specfun_arccosh(x)
[result, ifail] = s11ac(x)
```

3 Description

nag_specfun_arccosh (s11ac) calculates an approximate value for the inverse hyperbolic cosine, $\operatorname{arccosh} x$. It is based on the relation

$$\operatorname{arccosh} x = \ln\left(x + \sqrt{x^2 - 1}\right).$$

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 $\sqrt{x}$ to within the accuracy of the machine and hence we can guard against premature overflow and, without loss of accuracy, calculate

$$\operatorname{arccosh} x = \ln 2 + \ln x.$$

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **x** – REAL (KIND=nag_wp)

The argument x of the function.

Constraint: $\mathbf{x} \geq 1.0$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **result**

The result of the function.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

The function has been called with an argument less than 1.0, for which $\operatorname{arccosh} x$ is not defined. The result returned is zero.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

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

$$|\epsilon| \simeq \left| \frac{x}{\sqrt{x^2-1}\operatorname{arccosh} x} \times \delta \right|.$$

That is the relative error in the argument is amplified by a factor at least $\dfrac{x}{\sqrt{x^2-1}\operatorname{arccosh} x}$ in the result. The equality should apply if δ is greater than the ***machine precision*** (δ due to data errors etc.) but if δ is simply a result of round-off in the machine representation it is possible that an extra figure may be lost in internal calculation and round-off. The behaviour of the amplification factor is shown in the following graph:

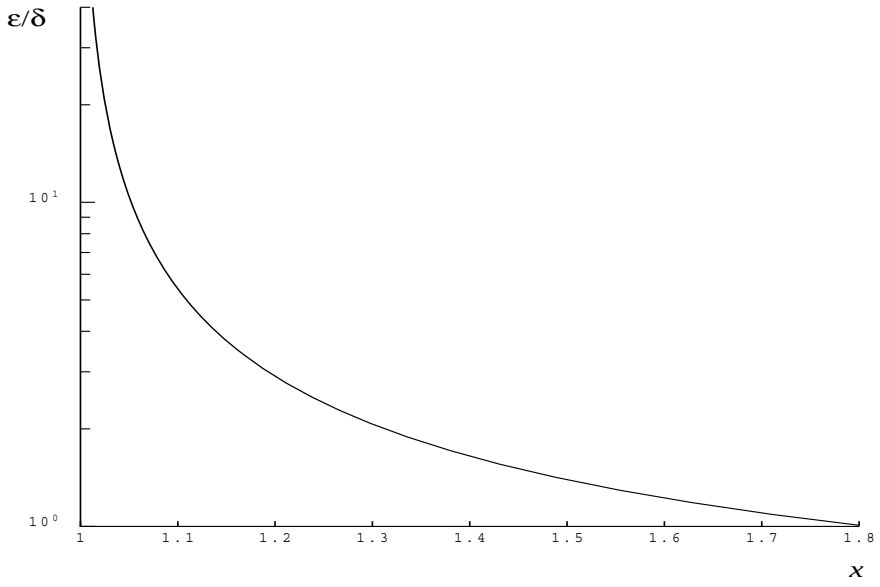

Figure 1

It should be noted that for $x > 2$ the factor is always less than 1.0. For large x we have the absolute error E in the result, in principle, given by

$$E \sim \delta.$$

This means that eventually accuracy is limited by ***machine precision***. More significantly for x close to 1, $x - 1 \sim \delta$, the above analysis becomes inapplicable due to the fact that both function and argument

are bounded, $x \geq 1$, $\operatorname{arccosh} x \geq 0$. In this region we have

$$E \sim \sqrt{\delta}.$$

That is, there will be approximately half as many decimal places correct in the result as there were correct figures in the argument.

8 Further Comments

None.

9 Example

This example 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

```
     function s11ac_example

fprintf('s11ac example results\n\n');

x = [1   2     5     10];
n = size(x,2);
result = x;

for j=1:n
  [result(j), ifail] = s11ac(x(j));
end

disp('      x        arccosh(x)');
fprintf('%12.3e%12.3e\n',[x; result]);
```

9.2 Program Results

```
     s11ac example results

      x        arccosh(x)
   1.000e+00   0.000e+00
   2.000e+00   1.317e+00
   5.000e+00   2.292e+00
   1.000e+01   2.993e+00
```