

## nag\_arccosh (s11acc)

### 1. Purpose

**nag\_arccosh (s11acc)** returns the value of the inverse hyperbolic cosine,  $\operatorname{arccosh} x$ . The result is in the principal positive branch.

### 2. Specification

```
#include <nag.h>
#include <nags.h>
```

```
double nag_arccosh(double x, NagError *fail)
```

### 3. Description

The function calculates an approximate value for the inverse hyperbolic cosine,  $\operatorname{arccosh} x$ . It is based on the relation

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

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. Parameters

**x**

Input: the argument  $x$  of the function.  
Constraint:  $x \geq 1.0$ .

**fail**

The NAG error parameter, see the Essential Introduction to the NAG C Library.

### 5. Error Indications and Warnings

**NE\_REAL\_ARG\_LT**

On entry, **x** must not be less than 1.0: **x** = *value*.  
 $\operatorname{arccosh} x$  is not defined and the result returned is zero.

### 6. Further Comments

#### 6.1. Accuracy

If  $\delta$  and  $\epsilon$  are the relative errors in the argument and result respectively, then in principle

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

That is the relative error in the argument is amplified by a factor at least

$$\frac{x}{\sqrt{x^2 - 1} \operatorname{arccosh} x}$$

in the result. The equality should apply if  $\delta$  is greater than the **machine precision** ( $\delta$  due to data error etc.), but if  $\delta$  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.

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.

## 6.2. References

Abramowitz M and Stegun I A (1968) *Handbook of Mathematical Functions* Dover Publications, New York ch 4.6 p 86.

## 7. See Also

None.

## 8. 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.

### 8.1. Program Text

```
/* nag_arccosh(s11acc) Example Program
 *
 * Copyright 1989 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

main()
{
    double x, y;

    Vprintf("s11acc Example Program Results\n");
    Vscanf("%*[^\\n]"); /* skip the first input line */
    Vprintf("      x      y\n");
    while (scanf("%lf", &x) != EOF)
    {
        y = s11acc(x, NAGERR_DEFAULT);
        Vprintf("%12.3e%12.3e\n", x, y);
    }
    exit(EXIT_SUCCESS);
}
```

### 8.2. Program Data

```
s11acc Example Program Data
1.00
2.0
5.0
10.0
```

**8.3. Program Results**

s11acc Example Program Results	
x	y
1.000e+00	0.000e+00
2.000e+00	1.317e+00
5.000e+00	2.292e+00
1.000e+01	2.993e+00

---