NAG Library Function Document

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

nag_arccosh (s11acc) 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 Arguments

1: **x** – double *Input*

On entry: the argument x of the function.

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

2: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

NE_REAL_ARG_LT

On entry, $\mathbf{x} = \langle value\rangle$.
Constraint: $\mathbf{x} \geq 1.0$.
The function has been called with an argument less than 1.0, for which $\operatorname{arccosh} x$ is not defined.

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

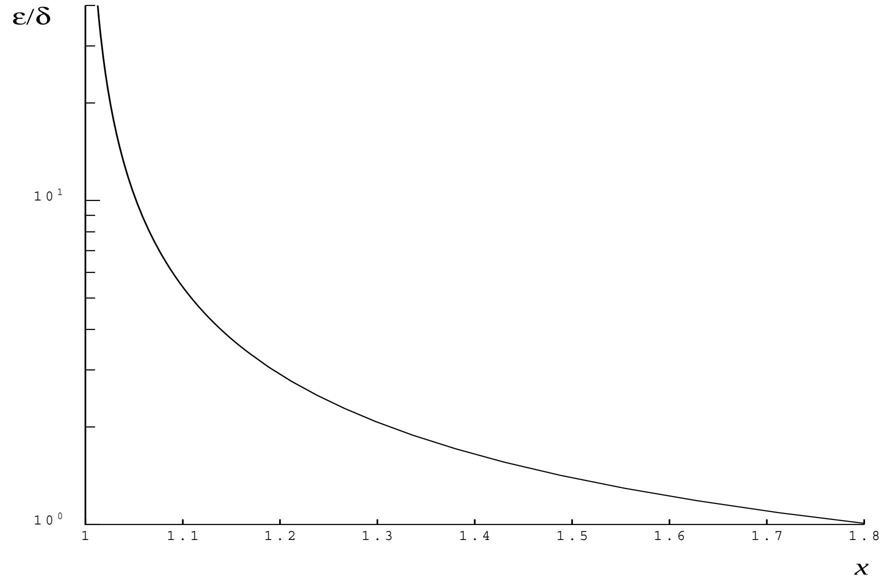

Figure 1

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 Parallelism and Performance

Not applicable.

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

```
/* 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>

int main(void)
{
  Integer  exit_status = 0;
  double   x, y;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
  scanf("%*[^\n]");
  printf("nag_arccosh (s11acc) Example Program Results\n");
  printf("     x           y\n");
  while (scanf("%lf", &x) != EOF)
    {
      /* nag_arccosh (s11acc).
       * Inverse hyperbolic cosine, arccosh x
       */
      y = nag_arccosh(x, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_arccosh (s11acc).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%12.3e%12.3e\n", x, y);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_arccosh (s11acc) Example Program Data
    1.00
    2.0
    5.0
   10.0
```

10.3 Program Results

```
nag_arccosh (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
```