

NAG Library Function Document

nag_arctanh (s11aac)

1 Purpose

nag_arctanh (s11aac) returns the value of the inverse hyperbolic tangent, $\operatorname{arctanh} x$.

2 Specification

```
#include <nag.h>
#include <nags.h>
double nag_arctanh (double x, NagError *fail)
```

3 Description

nag_arctanh (s11aac) calculates an approximate value for the inverse hyperbolic tangent of its argument, $\operatorname{arctanh} x$.

For $x^2 \leq \frac{1}{2}$ it is based on the Chebyshev expansion

$$\operatorname{arctanh} x = x \times y(t) = x \sum_{r=0} a_r T_r(t)$$

where $-\frac{1}{\sqrt{2}} \leq x \leq \frac{1}{\sqrt{2}}$, $-1 \leq t \leq 1$, and $t = 4x^2 - 1$.

For $\frac{1}{2} < x^2 < 1$, it uses

$$\operatorname{arctanh} x = \frac{1}{2} \ln \left(\frac{1+x}{1-x} \right).$$

For $|x| \geq 1$, the function fails as $\operatorname{arctanh} x$ is undefined.

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: $|x| < 1.0$.
- 2: **fail** – NagError * *Input/Output*
The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

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.

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

See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

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

See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_REAL_ARG_GE

On entry, $x = \langle value \rangle$.

Constraint: $|x| < 1$.

The function has been called with an argument greater than or equal to 1.0 in magnitude, for which $\operatorname{arctanh}$ is not defined.

7 Accuracy

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

$$|\epsilon| \simeq \left| \frac{x}{(1-x^2) \operatorname{arctanh} x} \times \delta \right|.$$

That is, the relative error in the argument, x , is amplified by at least a factor $\frac{x}{(1-x^2) \operatorname{arctanh} x}$ 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 then 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:

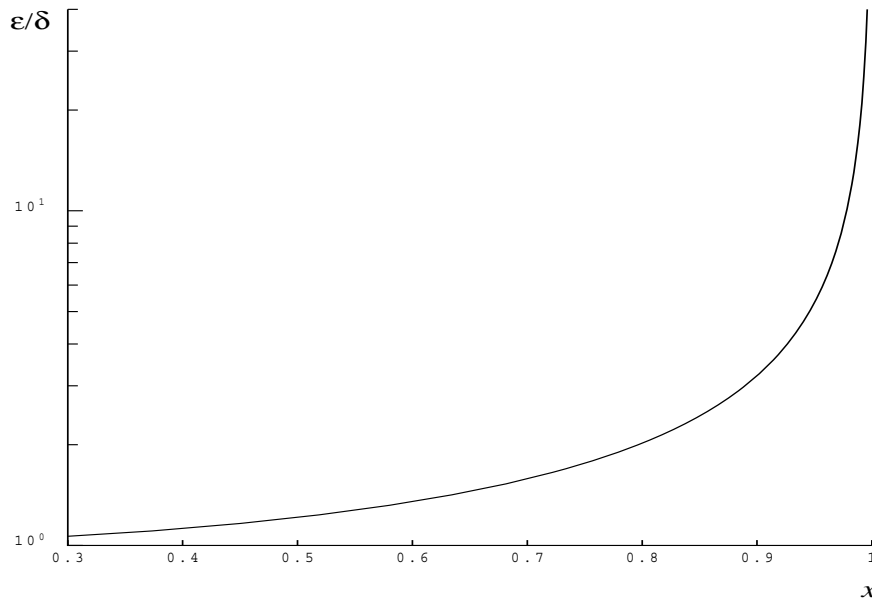


Figure 1

The factor is not significantly greater than one except for arguments close to $|x| = 1$. However in the region where $|x|$ is close to one, $1 - |x| \sim \delta$, the above analysis is inapplicable since x is bounded by definition, $|x| < 1$. In this region where $\operatorname{arctanh}$ is tending to infinity we have

$$\epsilon \sim 1/\ln \delta$$

which implies an obvious, unavoidable serious loss of accuracy near $|x| \sim 1$, e.g., if x and 1 agree to 6 significant figures, the result for $\operatorname{arctanh} x$ would be correct to at most about one figure.

8 Parallelism and Performance

`nag_arctanh` (s11aac) 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

```
/* nag_arctanh (s11aac) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#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 */
#ifdef _WIN32
scanf_s("%*[\n]");
#else
scanf("%*[\n]");
#endif
printf("nag_arctanh (s11aac) Example Program Results\n");
printf("      x              y\n");
#ifdef _WIN32
while (scanf_s("%lf", &x) != EOF)
#else
while (scanf("%lf", &x) != EOF)
#endif
{
/* nag_arctanh (s11aac).
 * Inverse hyperbolic tangent, arctanh x
 */
y = nag_arctanh(x, &fail);
if (fail.code != NE_NOERROR) {
printf("Error from nag_arctanh (s11aac).\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_arctanh (s11aac) Example Program Data
-0.5
 0.0
 0.5
-0.9999

```

10.3 Program Results

```

nag_arctanh (s11aac) Example Program Results
      x              y
-5.000e-01  -5.493e-01
 0.000e+00   0.000e+00
 5.000e-01   5.493e-01
-9.999e-01  -4.952e+00

```
