Module 3.1: nag_inv_hyp_fun
Inverse Hyperbolic Functions

nag_inv_hyp_fun contains procedures for approximating the inverse hyperbolic functions arctanh, arcsinh and arccosh with real arguments.

Contents

Procedure: nag_arctanh

1 Description

`nag_arctanh` calculates an approximate value for the inverse hyperbolic tangent, $\operatorname{arctanh} x$, where x is real (Abramowitz and Stegun [1], Chapter 4.6).

2 Usage

```
USE nag_inv_hyp_fun
[value =] nag_arctanh(x  [, optional arguments])
```

The function result is a scalar, of type real(kind=*wp*), containing $\operatorname{arctanh} x$.

3 Arguments

3.1 Mandatory Argument

x — real(kind=*wp*), intent(in)

Input: the argument x of the function.

Constraints: $|\mathrm{x}| < 1.0$.

3.2 Optional Argument

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document `nag_error_handling` (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to `nag_set_error` before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Algorithmic Detail

- For $x^2 \le \frac{1}{2}$, the procedure uses a Chebyshev expansion of the form

$$\operatorname{arctanh} x = x \times y(t) = x \sum_{r=0}{}' a_r T_r(t)$$

where $-\dfrac{1}{\sqrt{2}} \le x \le \dfrac{1}{\sqrt{2}}, \quad -1 \le t \le 1, \quad \text{and} \quad t = 4x^2 - 1.$

- 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 procedure fails as $\operatorname{arctanh} x$ is undefined.

6.2 Accuracy

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

$$|\varepsilon| \simeq |\theta\delta|, \quad \text{where} \quad \theta = \frac{x}{(1-x^2)\operatorname{arctanh} x}.$$

That is, the relative error in the argument, x, is amplified by at least a factor θ in the result. The equality should hold if δ is greater than EPSILON(1.0_*wp*) (i.e., if δ is 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 Figure 1.

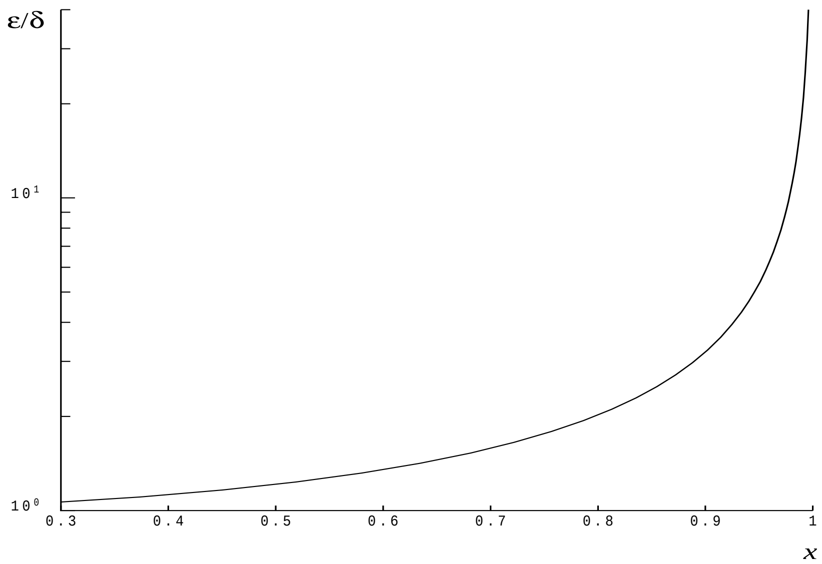

Figure 1: The error amplification factor θ.

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 arctanh is tending to infinity we have $\varepsilon \sim 1/\ln\delta$ 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.

Procedure: nag_arcsinh

1 Description

nag_arcsinh calculates an approximate value for the inverse hyperbolic sine, $\operatorname{arcsinh} x$, where x is real (Abramowitz and Stegun [1], Chapter 4.6).

2 Usage

```
USE nag_inv_hyp_fun
[value =] nag_arcsinh(x)
```

The function result is a scalar, of type real(kind=*wp*), containing $\operatorname{arcsinh} x$.

3 Arguments

3.1 Mandatory Argument

x — real(kind=*wp*), intent(in)

Input: the argument x of the function.

4 Error Codes

None.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Algorithmic Detail

- For $|x| \leq 1$ the procedure uses a Chebyshev expansion of the form

$$\operatorname{arcsinh} x = x \times y(t) = x \sum_{r=0}{}' c_r T_r(t), \quad \text{where } t = 2x^2 - 1.$$

- 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

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

6.2 Accuracy

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

$$|\varepsilon| \simeq |\theta\delta|, \quad \text{where} \quad \theta = \frac{x}{\sqrt{1+x^2}\,\operatorname{arcsinh} x}.$$

That is, the relative error in the argument, x, is amplified by a factor at least θ, in the result.

The equality should hold if δ is greater than EPSILON(1.0_*wp*) (i.e., if δ is 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 Figure 2.

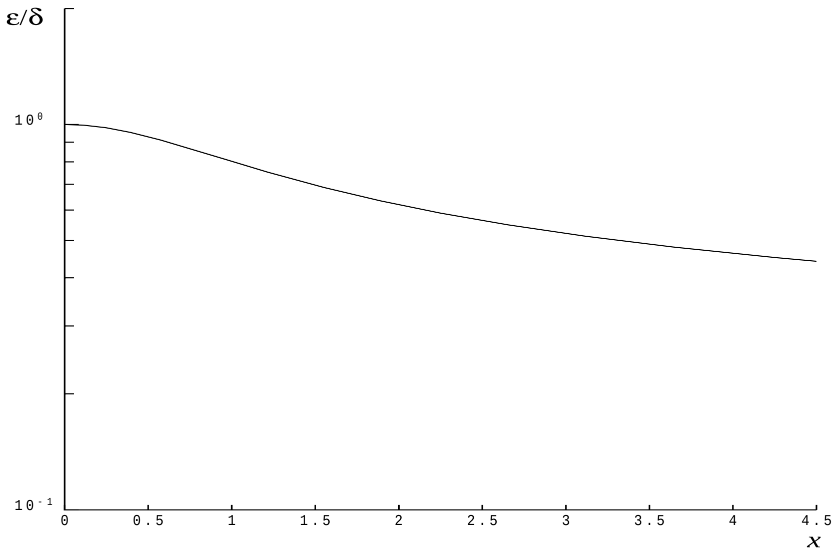

Figure 2: The error amplification factor θ.

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 EPSILON(1.0_*wp*).

Procedure: nag_arccosh

1 Description

`nag_arccosh` calculates an approximate value for the inverse hyperbolic cosine, $\operatorname{arccosh} x$, where x is real (Abramowitz and Stegun [1], Chapter 4.6).

2 Usage

`USE nag_inv_hyp_fun`

`[`*value* `=] nag_arccosh(x [,` *optional arguments*`])`

The function result is a scalar, of type real(kind=*wp*), containing $\operatorname{arccosh} x$.

3 Arguments

3.1 Mandatory Argument

x — real(kind=*wp*), intent(in)

Input: the argument x of the function.

Constraints: `x` ≥ 1.0.

3.2 Optional Argument

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document `nag_error_handling` (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to `nag_set_error` before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Algorithmic Detail

The result is based on the relation

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

where the square root is taken to be positive.

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

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

6.2 Accuracy

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

$$|\varepsilon| \simeq |\theta\delta|, \quad \text{where} \quad \theta = \frac{x}{\sqrt{x^2-1}\,\operatorname{arccosh} x}.$$

That is, the relative error in the argument is amplified by a factor at least θ in the result. The equality should apply if δ is greater than EPSILON(1.0_*wp*) (i.e., if δ is 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 Figure 3.

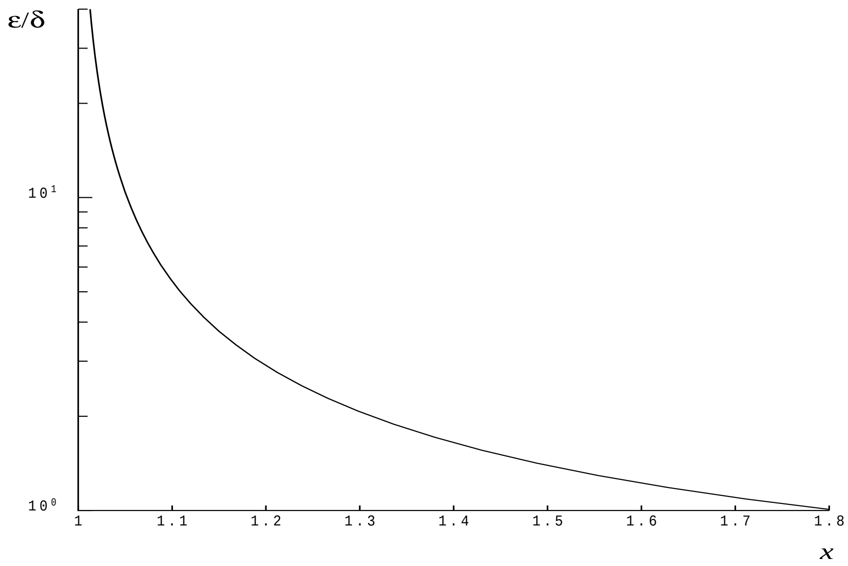

Figure 3: The error amplification factor θ.

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 EPSILON(1.0_*wp*). 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.

Example 1: Evaluation of the Inverse Hyperbolic Functions

This example program evaluates the functions `nag_arctanh`, `nag_arcsinh`, and `nag_arccosh` at a set of real values of the argument `x`.

1 Program Text

Note. The listing of the example program presented below is double precision. Single precision users are referred to Section 5.2 of the Essential Introduction for further information.

```
PROGRAM nag_inv_hyp_fun_ex01

  ! Example Program Text for nag_inv_hyp_fun
  ! NAG fl90, Release 3. NAG Copyright 1997.

  ! .. Use Statements ..
  USE nag_examples_io, ONLY : nag_std_out
  USE nag_inv_hyp_fun, ONLY : nag_arctanh, nag_arcsinh, nag_arccosh
  ! .. Implicit None Statement ..
  IMPLICIT NONE
  ! .. Intrinsic Functions ..
  INTRINSIC KIND
  ! .. Parameters ..
  INTEGER, PARAMETER :: n = 4
  INTEGER, PARAMETER :: wp = KIND(1.0D0)
  ! .. Local Scalars ..
  INTEGER :: i
  REAL (wp) :: y
  ! .. Local Arrays ..
  REAL (wp) :: x(n)
  ! .. Executable Statements ..

  WRITE (nag_std_out,*) 'Example Program Results for nag_inv_hyp_fun_ex01'

  WRITE (nag_std_out,*)
  WRITE (nag_std_out,*) '      x        arctanh(x)'
  x = (/ -0.5_wp, 0.0_wp, 0.5_wp, -0.9999_wp/)

  DO i = 1, n

    y = nag_arctanh(x(i))

    WRITE (nag_std_out,'(1X,1P,2E12.3)') x(i), y
  END DO

  WRITE (nag_std_out,*)
  WRITE (nag_std_out,*) '      x        arcsinh(x)'
  x = (/ -2.0_wp, -0.5_wp, 1.0_wp, 6.0_wp/)

  DO i = 1, n

    y = nag_arcsinh(x(i))

    WRITE (nag_std_out,'(1X,1P,2E12.3)') x(i), y
  END DO

  WRITE (nag_std_out,*)
  WRITE (nag_std_out,*) '      x        arccosh(x)'
  x = (/ 1.0_wp, 2.0_wp, 5.0_wp, 10.0_wp/)

  DO i = 1, n
```

```
      y = nag_arccosh(x(i))

      WRITE (nag_std_out,'(1X,1P,2E12.3)') x(i), y
    END DO

  END PROGRAM nag_inv_hyp_fun_ex01
```

2 Program Data

None.

3 Program Results

```
 Example Program Results for nag_inv_hyp_fun_ex01

      x        arctanh(x)
  -5.000E-01  -5.493E-01
   0.000E+00   0.000E+00
   5.000E-01   5.493E-01
  -9.999E-01  -4.952E+00

      x        arcsinh(x)
  -2.000E+00  -1.444E+00
  -5.000E-01  -4.812E-01
   1.000E+00   8.814E-01
   6.000E+00   2.492E+00

      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
```

Additional Examples

Not all example programs supplied with NAG *fl*90 appear in full in this module document. The following additional examples, associated with this module, are available.

`nag_inv_hyp_fun_ex02`

Evaluation of an approximation to the inverse hyperbolic sine.

`nag_inv_hyp_fun_ex03`

Evaluation of an approximation to the inverse hyperbolic cosine.

`nag_inv_hyp_fun_ex04`

Evaluation of an approximation to the inverse hyperbolic tangent.

References

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