

# NAG Library Function Document

## nag\_sinh (s10abc)

### 1 Purpose

nag\_sinh (s10abc) returns the value of the hyperbolic sine,  $\sinh x$ .

### 2 Specification

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

### 3 Description

nag\_sinh (s10abc) calculates an approximate value for the hyperbolic sine of its argument,  $\sinh x$ .

For  $|x| \leq 1$  it uses the Chebyshev expansion

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

where  $t = 2x^2 - 1$ .

For  $1 < |x| \leq E_1$ ,  $\sinh x = \frac{1}{2}(e^x - e^{-x})$

where  $E_1$  is a machine-dependent constant, details of which are given in the Users' Note for your implementation.

For  $|x| > E_1$ , the function fails owing to the danger of setting overflow in calculating  $e^x$ . The result returned for such calls is  $\sinh(\text{sign } x E_1)$ , i.e., it returns the result for the nearest valid argument.

### 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.
- 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\_GT**

On entry,  $x = \langle \text{value} \rangle$ .

Constraint:  $|x| \leq E_1$ .

The function has been called with an argument too large in absolute magnitude. There is a danger of overflow. The result returned is the value of  $\sinh x$  at the closest argument for which a valid call could be made.

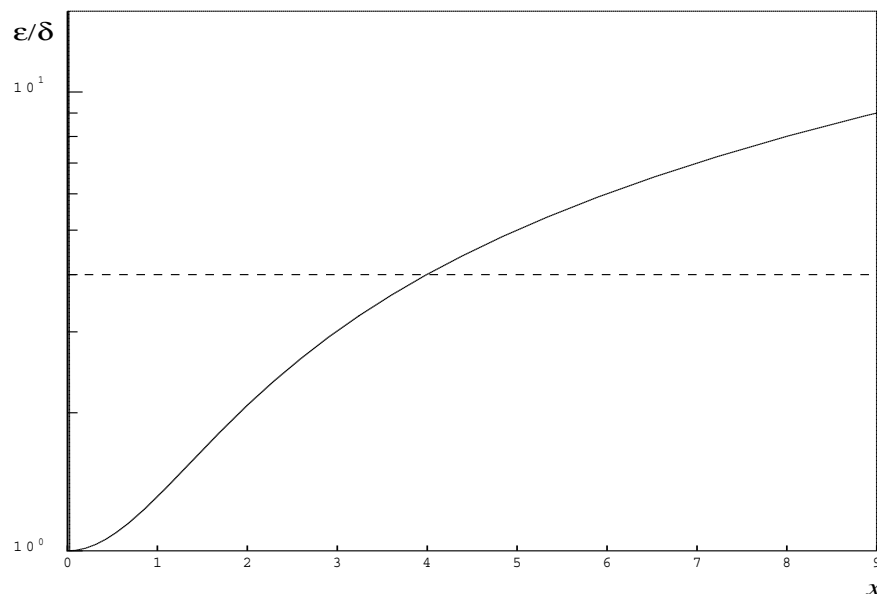
**7 Accuracy**

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

$$|\epsilon| \simeq |x \coth x \times \delta|.$$

That is the relative error in the argument,  $x$ , is amplified by a factor, approximately  $x \coth x$ . The equality should hold if  $\delta$  is greater than the *machine precision* ( $\delta$  is a result of data errors etc.) but, if  $\delta$  is simply a result of round-off in the machine representation of  $x$ , then it is possible that an extra figure may be lost in internal calculation round-off.

The behaviour of the error amplification factor can be seen in the following graph:



**Figure 1**

It should be noted that for  $|x| \geq 2$

$$\epsilon \sim x\delta = \Delta$$

where  $\Delta$  is the absolute error in the argument.

## 8 Parallelism and Performance

nag\_sinh (s10abc) 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_sinh (s10abc) 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_sinh (s10abc) Example Program Results\n");
    printf("      x              y\n");
#ifdef _WIN32
    while (scanf_s("%lf", &x) != EOF)
#else
    while (scanf("%lf", &x) != EOF)
#endif
    {
        /* nag_sinh (s10abc).
         * Hyperbolic sine, sinh x
         */
        y = nag_sinh(x, &fail);
        if (fail.code != NE_NOERROR) {
            printf("Error from nag_sinh (s10abc).\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_sinh (s10abc) Example Program Data
      -10.0
       -0.5
         0.0
         0.5
        25.0
```

## 10.3 Program Results

```
nag_sinh (s10abc) Example Program Results
      x              y
-1.000e+01  -1.101e+04
-5.000e-01  -5.211e-01
 0.000e+00   0.000e+00
 5.000e-01   5.211e-01
 2.500e+01   3.600e+10
```

---