NAG Library Function Document

nag_cosh (s10acc)

1   Purpose

nag_cosh (s10acc) returns the value of the hyperbolic cosine, $\cosh x$.

2   Specification

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

3   Description

nag_cosh (s10acc) calculates an approximate value for the hyperbolic cosine, $\cosh x$.

For $|x| \leq E_1$, $\cosh x = \frac{1}{2}(e^x + e^{-x})$.

For $|x| > E_1$, the function fails owing to danger of setting overflow in calculating e^x. The result returned for such calls is $\cosh E_1$, i.e., it returns the result for the nearest valid argument. The value of machine-dependent constant E_1 may be given in the Users' Note for your implementation.

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, $\mathbf{x} = \langle value \rangle$.
> Constraint: $|\mathbf{x}| \leq E_1$.
> 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 $\cosh x$ at the nearest valid argument.
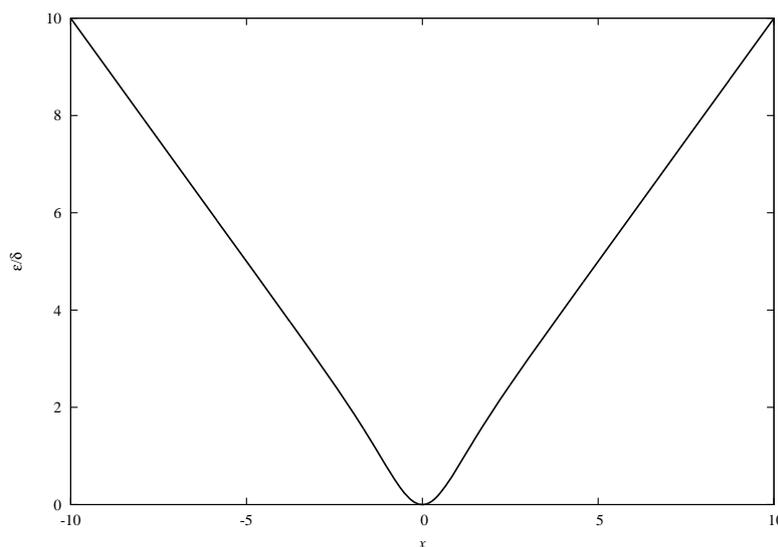
7   Accuracy

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

$$\epsilon \simeq x \tanh x \times \delta.$$

That is, the relative error in the argument, x, is amplified by a factor, at least $x \tanh x$. The equality should hold if δ is greater than the **machine precision** (δ is due to data errors etc.) but if δ 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 is shown by the following graph:

Figure 1

It should be noted that near $x = 0$ where this amplification factor tends to zero the accuracy will be limited eventually by the **machine precision**. Also for $|x| \geq 2$

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

where Δ is the absolute error in the argument x.

8   Parallelism and Performance

nag_cosh (s10acc) 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_cosh (s10acc) 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_cosh (s10acc) Example Program Results\n");
  printf("      x            y\n");
#ifdef _WIN32
  while (scanf_s("%lf", &x) != EOF)
#else
  while (scanf("%lf", &x) != EOF)
#endif
  {
    /* nag_cosh (s10acc).
     * Hyperbolic cosine, cosh x
     */
    y = nag_cosh(x, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_cosh (s10acc).\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_cosh (s10acc) Example Program Data
          -10.0
           -0.5
            0.0
            0.5
           25.0
```

10.3 Program Results

```
nag_cosh (s10acc) Example Program Results
     x          y
 -1.000e+01  1.101e+04
 -5.000e-01  1.128e+00
  0.000e+00  1.000e+00
  5.000e-01  1.128e+00
  2.500e+01  3.600e+10
```