

# NAG Library Function Document

## nag\_cos\_integral (s13acc)

### 1 Purpose

nag\_cos\_integral (s13acc) returns the value of the cosine integral

$$\text{Ci}(x) = \gamma + \ln x + \int_0^x \frac{\cos u - 1}{u} du, \quad x > 0$$

where  $\gamma$  denotes Euler's constant.

### 2 Specification

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

### 3 Description

nag\_cos\_integral (s13acc) calculates an approximate value for  $\text{Ci}(x)$ .

For  $0 < x \leq 16$  it is based on the Chebyshev expansion

$$\text{Ci}(x) = \ln x + \sum_{r=0}^l a_r T_r(t), \quad t = 2 \left( \frac{x}{16} \right)^2 - 1.$$

For  $16 < x < x_{\text{hi}}$  where the value of  $x_{\text{hi}}$  is given in the Users' Note for your implementation,

$$\text{Ci}(x) = \frac{f(x) \sin x}{x} - \frac{g(x) \cos x}{x^2}$$

where  $f(x) = \sum_{r=0} f_r T_r(t)$  and  $g(x) = \sum_{r=0} g_r T_r(t)$ ,  $t = 2 \left( \frac{16}{x} \right)^2 - 1$ .

For  $x \geq x_{\text{hi}}$ ,  $\text{Ci}(x) = 0$  to within the accuracy possible (see Section 7).

### 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:*  $x > 0.0$ .
- 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 3.2.1.2 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 3.6.6 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 3.6.5 in How to Use the NAG Library and its Documentation for further information.

### NE\_REAL\_ARG\_LE

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

Constraint:  $x > 0.0$ .

The function has been called with an argument less than or equal to zero for which  $Ci(x)$  is not defined.

## 7 Accuracy

If  $E$  and  $\epsilon$  are the absolute and relative errors in the result and  $\delta$  is the relative error in the argument then in principle these are related by

$$|E| \simeq |\delta \cos x| \text{ and } |\epsilon| \simeq \left| \frac{\delta \cos x}{Ci(x)} \right|.$$

That is accuracy will be limited by *machine precision* near the origin and near the zeros of  $\cos x$ , but near the zeros of  $Ci(x)$  only absolute accuracy can be maintained.

The behaviour of this amplification is shown in Figure 1.

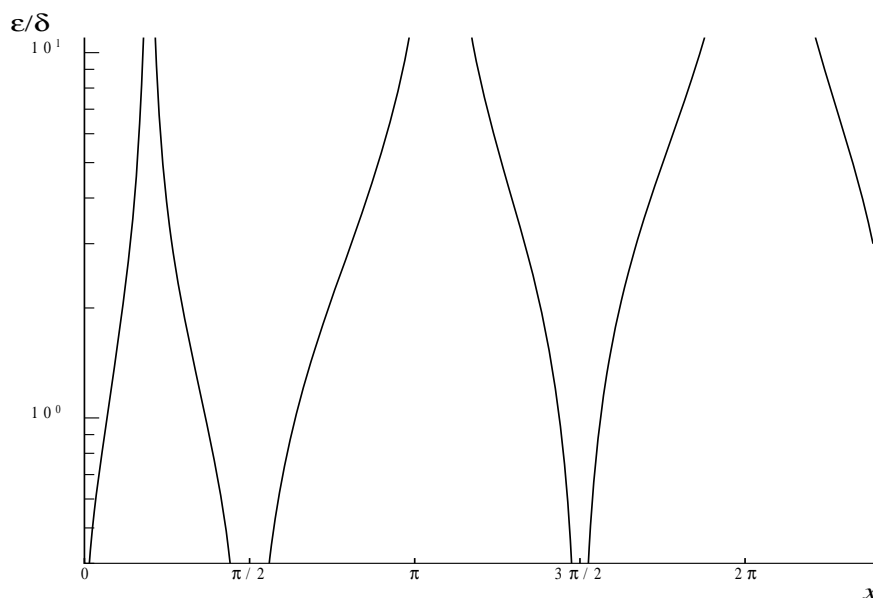


Figure 1

For large values of  $x$ ,  $\text{Ci}(x) \sim \frac{\sin x}{x}$  therefore  $\epsilon \sim \delta x \cot x$  and since  $\delta$  is limited by the finite precision of the machine it becomes impossible to return results which have any relative accuracy. That is, when  $x \geq 1/\delta$  we have that  $|\text{Ci}(x)| \leq 1/x \sim E$  and hence is not significantly different from zero.

Hence  $x_{\text{hi}}$  is chosen such that for values of  $x \geq x_{\text{hi}}$ ,  $\text{Ci}(x)$  in principle would have values less than the *machine precision* and so is essentially zero.

## 8 Parallelism and Performance

nag\_cos\_integral (s13acc) 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_cos_integral (s13acc) 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_cos_integral (s13acc) Example Program Results\n");
    printf("      x      y\n");
#ifdef _WIN32
    while (scanf_s("%lf", &x) != EOF)
#else
    while (scanf("%lf", &x) != EOF)
#endif
    {
        /* nag_cos_integral (s13acc).
         * Cosine integral Ci(x)
         */
        y = nag_cos_integral(x, &fail);
        if (fail.code != NE_NOERROR) {
            printf("Error from nag_cos_integral (s13acc).\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_cos_integral (s13acc) Example Program Data
    0.2
    0.4
    0.6
    0.8
    1.0
```

## 10.3 Program Results

```
nag_cos_integral (s13acc) Example Program Results
    x           y
2.000e-01  -1.042e+00
4.000e-01  -3.788e-01
6.000e-01  -2.227e-02
8.000e-01   1.983e-01
1.000e+00   3.374e-01
```

