NAG Library Function Document

nag_airy_bi (s17ahc)

1 Purpose
nag_airy_bi (s17ahc) returns a value of the Airy function, Bi(x).

2 Specification
#include <nag.h>
#include <nags.h>
double nag_airy_bi (double x, NagError *fail)

3 Description
nag_airy_bi (s17ahc) evaluates an approximation to the Airy function Bi(x). It is based on a number of
Chebyshev expansions.
For x < -5,

\[ Bi(x) = \frac{a(t) \cos z + b(t) \sin z}{(-x)^{1/4}}, \]

where \( z = \frac{\pi}{4} + \frac{2}{3} \sqrt{-x^3} \) and a(t) and b(t) are expansions in the variable \( t = -2 \left( \frac{5}{x} \right)^3 - 1 \).

For -5 ≤ x ≤ 0,

\[ Bi(x) = \sqrt{3}(f(t) + xg(t)), \]

where f and g are expansions in \( t = -2 \left( \frac{x}{5} \right)^3 - 1 \).

For 0 < x < 4.5,

\[ Bi(x) = e^{11x/8}y(t), \]

where y is an expansion in \( t = 4x/9 - 1 \).

For 4.5 ≤ x < 9,

\[ Bi(x) = e^{5x/2}v(t), \]

where v is an expansion in \( t = 4x/9 - 3 \).

For x ≥ 9,

\[ Bi(x) = \frac{e^z u(t)}{x^{1/4}}, \]

where \( z = \frac{2}{3} \sqrt{x^3} \) and u is an expansion in \( t = 2 \left( \frac{18}{2 \epsilon} \right) - 1 \).

For \( |x| < \text{machine precision} \), the result is set directly to Bi(0). This both saves time and avoids possible
intermediate underflows.

For large negative arguments, it becomes impossible to calculate the phase of the oscillating function
with any accuracy so the function must fail. This occurs if \( x < -\left( \frac{3}{2 \epsilon} \right)^{2/3} \), where \( \epsilon \) is the machine
precision.
For large positive arguments, there is a danger of causing overflow since $B_i$ grows in an essentially exponential manner, so the function must fail.

4 References


5 Arguments

1: $x$ – double
   
   On entry: the argument $x$ of the function.

2: $fail$ – NagError*
   
   The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction 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 the Essential Introduction 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 the Essential Introduction for further information.

NE_REAL_ARG_GT

On entry, $x = \langle value\rangle$.
Constraint: $x \leq \langle value\rangle$.
$x$ is too large and positive. The function returns zero.

NE_REAL_ARG_LT

On entry, $x = \langle value\rangle$.
Constraint: $x \geq \langle value\rangle$.
$x$ is too large and negative. The function returns zero.

7 Accuracy

For negative arguments the function is oscillatory and hence absolute error is the appropriate measure. In the positive region the function is essentially exponential-like and here relative error is appropriate. The absolute error, $E$, and the relative error, $\epsilon$, are related in principle to the relative error in the argument, $\delta$, by

$$ E \simeq |x B_i'(x)| \delta, \quad \epsilon \simeq \left| \frac{x B_i'(x)}{B_i(x)} \right| \delta. $$

In practice, approximate equality is the best that can be expected. When $\delta$, $\epsilon$ or $E$ is of the order of the machine precision, the errors in the result will be somewhat larger.
For small $x$, errors are strongly damped and hence will be bounded essentially by the *machine precision*. For moderate to large negative $x$, the error behaviour is clearly oscillatory but the amplitude of the error grows like amplitude $\left(\frac{E}{\delta}\right) \sim |x|^{5/4} \sqrt{n}$.

However the phase error will be growing roughly as $\frac{2}{3}\sqrt{|x|^3}$ and hence all accuracy will be lost for large negative arguments. This is due to the impossibility of calculating sin and cos to any accuracy if $\frac{2}{3}\sqrt{|x|^3} > \frac{1}{\delta}$.

For large positive arguments, the relative error amplification is considerable:

$$\frac{\epsilon}{\delta} \sim \sqrt{x^3}.$$

This means a loss of roughly two decimal places accuracy for arguments in the region of 20. However very large arguments are not possible due to the danger of causing overflow and errors are therefore limited in practice.

8 Parallelism and Performance

Not applicable.

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_airy_bi (s17ahc) Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 2 revised, 1992. */

#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_airy_bi (s17ahc) Example Program Results\n");
    printf(" x y\n");
    #ifdef _WIN32

while (scanf_s("%lf", &x) != EOF)
#else
while (scanf("%lf", &x) != EOF)
#endif
{
    /* nag_airy_bi (s17ahc).
     * Airy function Bi(x)
     */
    y = nag_airy_bi(x, &fail);
    if (fail.code != NE_NOERROR)
    { 
        printf("Error from nag_airy_bi (s17ahc).\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_airy_bi (s17ahc) Example Program Data
    -10.0
    -1.0
    0.0
    1.0
    5.0
    10.0
    20.0

10.3 Program Results

nag_airy_bi (s17ahc) Example Program Results
x   y
-1.000e+01 -3.147e-01
-1.000e+00  1.040e-01
 0.000e+00  6.149e-01
 1.000e+00  1.207e+00
 5.000e+00  6.578e+02
 1.000e+01  4.556e+08
 2.000e+01  2.104e+25
Example Program
Returns a Value of the Airy Function $Bi(x)$