1 Purpose

nag_airy_bi_deriv (s17akc) returns a value for the derivative of the Airy function $Bi(x)$.

2 Specification

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

3 Description

nag_airy_bi_deriv (s17akc) calculates an approximate value for the derivative of the Airy function $Bi(x)$. It is based on a number of Chebyshev expansions.

For $x < -5$,

$$Bi'(x) = \sqrt{-x} \left[ -a(t) \sin \frac{b(t)}{\zeta} \cos z \right],$$

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

For $-5 \leq x \leq 0$,

$$Bi'(x) = \sqrt{3}(x^2 f(t) + g(t)),$$

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

For $0 < x < 4.5$,

$$Bi'(x) = e^{3x/2} y(t),$$

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

For $4.5 \leq x < 9$,

$$Bi'(x) = e^{21x/8} u(t),$$

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

For $x \geq 9$,

$$Bi'(x) = \sqrt{xe^z v(t)},$$

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

For $|x| < \text{the square of the machine precision}$, the result is set directly to $Bi'(0)$. This saves time and avoids possible underflows in calculation.

For large negative arguments, it becomes impossible to calculate a result for the oscillating function with any accuracy so the function must fail. This occurs for $x < -\left(\frac{\sqrt{\pi}}{\epsilon}\right)^{4/7}$, where $\epsilon$ is the machine precision.
For large positive arguments, where $B_i$ grows in an essentially exponential manner, there is a danger of overflow so the function must fail.

4 References


5 Arguments

1:  $x$ – double  
$\text{Input}$  
$On\ entry$: the argument $x$ of the function.

2:  fail – NagError*  
$\text{Input/Output}$  
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, $x = \langle\text{value}\rangle$.  
Constraint: $x \leq \langle\text{value}\rangle$.  
$x$ is too large and positive. The function returns zero.

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

7 Accuracy

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

$$E \approx |x^2 Bi(x)| \delta \quad \epsilon \approx \frac{|x^2 Bi(x)|}{Bi\prime(x)} \delta.$$
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 \( x \), positive or negative, errors are strongly attenuated by the function and hence will effectively be bounded by the [machine precision].

For moderate to large negative \( x \), the error is, like the function, oscillatory. However, the amplitude of the absolute error grows like \( \frac{|x|^{7/4}}{\sqrt{\pi}} \). Therefore it becomes impossible to calculate the function with any accuracy if \( |x|^{7/4} > \frac{\sqrt{\pi}}{\delta} \).

For large positive \( x \), the relative error amplification is considerable: \( \frac{\epsilon}{\delta} \sim x^{3} \). However, very large arguments are not possible due to the danger of overflow. Thus in practice the actual amplification that occurs is limited.

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_deriv (s17akc) 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_deriv (s17akc) 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_deriv (s17akc).
     * Derivative of the Airy function Bi(x)
     */
    y = nag_airy_bi_deriv(x, &fail);
    if (fail.code != NE_NOERROR)
    {
        printf("Error from nag_airy_bi_deriv (s17akc).\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_deriv (s17akc) Example Program Data
-10.0
-1.0
0.0
1.0
5.0
10.0
20.0

10.3 Program Results

nag_airy_bi_deriv (s17akc) Example Program Results
    x           y
-1.000e+01  1.194e-01
-1.000e+00  5.924e-01
 0.000e+00  4.483e-01
 1.000e+00  9.324e-01
 5.000e+00  1.436e+03
 1.000e+01  1.429e+09
 2.000e+01  9.382e+25
Example Program

Returns a Value for the Derivative of the Airy Function $B_i(x)$