NAG Toolbox

nag_specfun_airy_bi_real_vector (s17av)

1 Purpose

nag_specfun_airy_bi_real_vector (s17av) returns an array of values of the Airy function, $\mathrm{Bi}(x)$.

2 Syntax

```
[f, ivalid, ifail] = nag_specfun_airy_bi_real_vector(x, 'n', n)
[f, ivalid, ifail] = s17av(x, 'n', n)
```

3 Description

nag_specfun_airy_bi_real_vector (s17av) evaluates an approximation to the Airy function $\mathrm{Bi}(x_i)$ for an array of arguments x_i, for $i = 1, 2, \ldots, n$. It is based on a number of Chebyshev expansions.

For $x < -5$,

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

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 $-5 \leq x \leq 0$,

$$\mathrm{Bi}(x) = \sqrt{3}(f(t) + xg(t)),$$

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

For $0 < x < 4.5$,

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

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

For $4.5 \leq x < 9$,

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

where v is an expansion in $t = 4x/9 - 3$.

For $x \geq 9$,

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

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

For $|x| <$ ***machine precision***, the result is set directly to $\mathrm{Bi}(0)$. 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 ϵ is the ***machine precision***.

For large positive arguments, there is a danger of causing overflow since Bi grows in an essentially exponential manner, so the function must fail.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

5 Parameters

5.1 Compulsory Input Parameters

1: **x**(**n**) – REAL (KIND=nag_wp) array

The argument x_i of the function, for $i = 1, 2, \ldots,$ **n**.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **x**.

n, the number of points.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **f**(**n**) – REAL (KIND=nag_wp) array

$\mathrm{Bi}(x_i)$, the function values.

2: **ivalid**(**n**) – INTEGER array

ivalid(i) contains the error code for x_i, for $i = 1, 2, \ldots,$ **n**.

ivalid$(i) = 0$
No error.

ivalid$(i) = 1$
x_i is too large and positive. **f**(i) contains zero. The threshold value is the same as for **ifail** $= 1$ in nag_specfun_airy_bi_real (s17ah), as defined in the Users' Note for your implementation.

ivalid$(i) = 2$
x_i is too large and negative. **f**(i) contains zero. The threshold value is the same as for **ifail** $= 2$ in nag_specfun_airy_bi_real (s17ah), as defined in the Users' Note for your implementation.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

On entry, at least one value of **x** was invalid.
Check **ivalid** for more information.

ifail $= 2$

Constraint: $\mathbf{n} \geq 0$.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

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, ϵ, are related in principle to the relative error in the argument, δ, by

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

In practice, approximate equality is the best that can be expected. When δ, ϵ 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 $\left(\frac{E}{\delta}\right) \sim \frac{|x|^{5/4}}{\sqrt{\pi}}$.

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 Further Comments

None.

9 Example

This example reads values of **x** from a file, evaluates the function at each value of x_i and prints the results.

9.1 Program Text

```
function s17av_example

fprintf('s17av example results\n\n');

x = [-10; -1; 0; 1; 5; 10; 20];

[f, ivalid, ifail] = s17av(x);

fprintf('     x            Bi(x)   ivalid\n');
for i=1:numel(x)
  fprintf('%12.3e%12.3e%5d\n', x(i), f(i), ivalid(i));
end
```

9.2 Program Results

```
s17av example results

     x            Bi(x)   ivalid
  -1.000e+01  -3.147e-01    0
  -1.000e+00   1.040e-01    0
   0.000e+00   6.149e-01    0
   1.000e+00   1.207e+00    0
   5.000e+00   6.578e+02    0
   1.000e+01   4.556e+08    0
   2.000e+01   2.104e+25    0
```