NAG Library Routine Document

S17AJF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S17AJF returns a value of the derivative of the Airy function $\mathrm{Ai}(x)$, via the function name.

2 Specification

```
FUNCTION S17AJF (X, IFAIL)
REAL (KIND=nag_wp) S17AJF
INTEGER            IFAIL
REAL (KIND=nag_wp) X
```

3 Description

S17AJF evaluates an approximation to the derivative of the Airy function $\mathrm{Ai}(x)$. It is based on a number of Chebyshev expansions.

For $x < -5$,

$$\mathrm{Ai}'(x) = \sqrt[4]{-x}\left[a(t)\cos z + \frac{b(t)}{\zeta}\sin z\right],$$

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

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

$$\mathrm{Ai}'(x) = x^2 f(t) - g(t),$$

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

For $0 < x < 4.5$,

$$\mathrm{Ai}'(x) = e^{-11x/8} y(t),$$

where $y(t)$ is an expansion in $t = 4\left(\frac{x}{9}\right) - 1$.

For $4.5 \leq x < 9$,

$$\mathrm{Ai}'(x) = e^{-5x/2} v(t),$$

where $v(t)$ is an expansion in $t = 4\left(\frac{x}{9}\right) - 3$.

For $x \geq 9$,

$$\mathrm{Ai}'(x) = \sqrt[4]{x} e^{-z} u(t),$$

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

For $|x| <$ the square of the ***machine precision***, the result is set directly to $\mathrm{Ai}'(0)$. This both saves time and avoids possible intermediate underflows.

For large negative arguments, it becomes impossible to calculate a result for the oscillating function with any accuracy and so the routine must fail. This occurs for $x < -\left(\frac{\sqrt{\pi}}{\epsilon}\right)^{4/7}$, where ϵ is the ***machine precision***.

For large positive arguments, where Ai' decays in an essentially exponential manner, there is a danger of underflow so the routine must fail.

4 References

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

5 Arguments

1: X – REAL (KIND=nag_wp) *Input*

On entry: the argument x of the function.

2: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

X is too large and positive. On softfailure, the routine returns zero. (see the Users' Note for your implementation for details)

IFAIL $= 2$

X is too large and negative. On softfailure, the routine returns zero. See also the Users' Note for your implementation.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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 in character and here relative error is needed. The absolute error, E, and the relative error, ϵ, are related in principle to the relative error in the argument, δ, by

$$E \simeq \left|x^2\,\mathrm{Ai}(x)\right|\delta \quad \epsilon \simeq \left|\frac{x^2\,\mathrm{Ai}(x)}{\mathrm{Ai}'(x)}\right|\delta.$$

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, positive or negative, errors are strongly attenuated by the function and hence will be roughly bounded by the ***machine precision***.

For moderate to large negative x, the error, like the function, is oscillatory; however the amplitude of the error grows like

$$\frac{|x|^{7/4}}{\sqrt{\pi}}.$$

Therefore it becomes impossible to calculate the function with any accuracy if $|x|^{7/4} > \dfrac{\sqrt{\pi}}{\delta}$.

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

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

However, very large arguments are not possible due to the danger of underflow. Thus in practice error amplification is limited.

8 Parallelism and Performance

S17AJF 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

```
    Program s17ajfe

!     S17AJF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s17ajf
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: x, y
      Integer                          :: ifail, ioerr
!     .. Executable Statements ..
      Write (nout,*) 'S17AJF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Write (nout,*)
      Write (nout,*) '     X           Y'
      Write (nout,*)

data: Do
        Read (nin,*,Iostat=ioerr) x

        If (ioerr<0) Then
          Exit data
        End If

        ifail = -1
        y = s17ajf(x,ifail)

        If (ifail<0) Then
          Exit data
        End If

        Write (nout,99999) x, y
      End Do data

99999 Format (1X,1P,2E12.3)
    End Program s17ajfe
```

10.2 Program Data

```
S17AJF Example Program Data
             -10.0
             -1.0
             0.0
             1.0
             5.0
             10.0
             20.0
```

10.3 Program Results

```
 S17AJF Example Program Results

      X           Y

   -1.000E+01   9.963E-01
   -1.000E+00  -1.016E-02
    0.000E+00  -2.588E-01
    1.000E+00  -1.591E-01
    5.000E+00  -2.474E-04
    1.000E+01  -3.521E-10
    2.000E+01  -7.586E-27
```

Example Program
Returns a Value for the Derivative of the Airy Function *Ai*(*x*)