S17AGF – NAG Fortran Library Routine Document

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

S17AGF returns a value for the Airy function, $\mathrm{Ai}(x)$, via the routine name.

2 Specification

```
real FUNCTION S17AGF(X, IFAIL)
INTEGER              IFAIL
real                 X
```

3 Description

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

For $x < -5$,

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

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 $-5 \leq x \leq 0$,

$$\mathrm{Ai}(x) = f(t) - xg(t),$$

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

For $0 < x < 4.5$,

$$\mathrm{Ai}(x) = e^{-3x/2}y(t),$$

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

For $4.5 \leq x < 9$,

$$\mathrm{Ai}(x) = e^{-5x/2}u(t),$$

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

For $x \geq 9$,

$$\mathrm{Ai}(x) = \frac{e^{-z}v(t)}{x^{1/4}},$$

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

For $|x| <$ ***machine precision***, the result is set directly to $\mathrm{Ai}(0)$. This both saves time and guards against underflow in intermediate calculations.

For large negative arguments, it becomes impossible to calculate the phase of the oscillatory function with any precision and so the routine must fail. This occurs if $x < -\left(\frac{3}{2\epsilon}\right)^{2/3}$, 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

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

5 Parameters

1: X — ***real*** *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. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL $= 1$

X is too large and positive. On soft failure, the routine returns zero.

IFAIL $= 2$

X is too large and negative. On soft failure, the routine 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, ϵ, are related in principle to the relative error in the argument, δ, by

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

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, errors are strongly damped by the function and hence will be bounded by the ***machine precision***.

For moderate negative x, the error behaviour is oscillatory but the amplitude of the error grows like

$$\text{amplitude} \left(\frac{E}{\delta} \right) \sim \frac{|x|^{5/4}}{\sqrt{\pi}}.$$

However the phase error will be growing roughly like $\frac{2}{3}\sqrt{|x|^3}$ and hence all accuracy will be lost for large negative arguments 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 setting underflow and so the errors are limited in practice.

8 Further Comments

None.

9 Example

The example program reads values of the argument x from a file, evaluates the function at each value of x and prints the results.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     S17AGF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             X, Y
      INTEGER          IFAIL
*     .. External Functions ..
      real             S17AGF
      EXTERNAL         S17AGF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S17AGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*) '     X           Y        IFAIL'
      WRITE (NOUT,*)
   20 READ (NIN,*,END=40) X
      IFAIL = 1
*
      Y = S17AGF(X,IFAIL)
*
      WRITE (NOUT,99999) X, Y, IFAIL
      GO TO 20
   40 STOP
*
99999 FORMAT (1X,1P,2e12.3,I7)
      END
```

9.2 Program Data

```
S17AGF Example Program Data
             -10.0
             -1.0
             0.0
             1.0
             5.0
             10.0
             20.0
```

9.3 Program Results

```
S17AGF Example Program Results

     X           Y        IFAIL

  -1.000E+01   4.024E-02      0
  -1.000E+00   5.356E-01      0
   0.000E+00   3.550E-01      0
   1.000E+00   1.353E-01      0
   5.000E+00   1.083E-04      0
   1.000E+01   1.105E-10      0
   2.000E+01   1.692E-27      0
```