NAG Library Routine Document

S17DGF

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

S17DGF returns the value of the Airy function $\mathrm{Ai}(z)$ or its derivative $\mathrm{Ai}'(z)$ for complex z, with an option for exponential scaling.

2 Specification

```
SUBROUTINE S17DGF (DERIV, Z, SCAL, AI, NZ, IFAIL)

INTEGER                NZ, IFAIL
COMPLEX (KIND=nag_wp)  Z, AI
CHARACTER(1)           DERIV, SCAL
```

3 Description

S17DGF returns a value for the Airy function $\mathrm{Ai}(z)$ or its derivative $\mathrm{Ai}'(z)$, where z is complex, $-\pi < \arg z \leq \pi$. Optionally, the value is scaled by the factor $e^{2z\sqrt{z}/3}$.

The routine is derived from the routine CAIRY in Amos (1986). It is based on the relations $\mathrm{Ai}(z) = \frac{\sqrt{z}K_{1/3}(w)}{\pi\sqrt{3}}$, and $\mathrm{Ai}'(z) = \frac{-zK_{2/3}(w)}{\pi\sqrt{3}}$, where K_ν is the modified Bessel function and $w = 2z\sqrt{z}/3$.

For very large $|z|$, argument reduction will cause total loss of accuracy, and so no computation is performed. For slightly smaller $|z|$, the computation is performed but results are accurate to less than half of ***machine precision***. If $\mathrm{Re}(w)$ is too large, and the unscaled function is required, there is a risk of overflow and so no computation is performed. In all the above cases, a warning is given by the routine.

4 References

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

Amos D E (1986) Algorithm 644: A portable package for Bessel functions of a complex argument and non-negative order *ACM Trans. Math. Software* **12** 265–273

5 Parameters

1: DERIV – CHARACTER(1) *Input*

On entry: specifies whether the function or its derivative is required.

If DERIV = 'F', $\mathrm{Ai}(z)$ is returned.

If DERIV = 'D', $\mathrm{Ai}'(z)$ is returned.

Constraint: DERIV = 'F' or 'D'.

2: Z – COMPLEX (KIND=nag_wp) *Input*

On entry: the argument z of the function.

3: SCAL – CHARACTER(1) *Input*

On entry: the scaling option.

SCAL = 'U'
The result is returned unscaled.

SCAL = 'S'
The result is returned scaled by the factor $e^{2z\sqrt{z}/3}$.

Constraint: SCAL = 'U' or 'S'.

4: AI – COMPLEX (KIND=nag_wp) *Output*

On exit: the required function or derivative value.

5: NZ – INTEGER *Output*

On exit: indicates whether or not AI is set to zero due to underflow. This can only occur when SCAL = 'U'.

NZ = 0
AI is not set to zero.

NZ = 1
AI is set to zero.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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

On entry, DERIV $\neq$ 'F' or 'D'.
or SCAL $\neq$ 'U' or 'S'.

IFAIL = 2

No computation has been performed due to the likelihood of overflow, because $\mathrm{Re}(w)$ is too large, where $w = 2\mathrm{Z}\sqrt{\mathrm{Z}}/3$ – how large depends on Z and the overflow threshold of the machine. This error exit can only occur when SCAL = 'U'.

IFAIL = 3

The computation has been performed, but the errors due to argument reduction in elementary functions make it likely that the result returned by S17DGF is accurate to less than half of ***machine precision***. This error exit may occur if abs(Z) is greater than a machine-dependent threshold value (given in the Users' Note for your implementation).

IFAIL = 4

No computation has been performed because the errors due to argument reduction in elementary functions mean that all precision in the result returned by S17DGF would be lost. This error exit may occur if abs(Z) is greater than a machine-dependent threshold value (given in the Users' Note for your implementation).

IFAIL = 5

No result is returned because the algorithm termination condition has not been met. This may occur because the parameters supplied to S17DGF would have caused overflow or underflow.

7 Accuracy

All constants in S17DGF are given to approximately 18 digits of precision. Calling the number of digits of precision in the floating point arithmetic being used t, then clearly the maximum number of correct digits in the results obtained is limited by $p = \min(t, 18)$. Because of errors in argument reduction when computing elementary functions inside S17DGF, the actual number of correct digits is limited, in general, by $p - s$, where $s \approx \max(1, |\log_{10}|z||)$ represents the number of digits lost due to the argument reduction. Thus the larger the value of $|z|$, the less the precision in the result.

Empirical tests with modest values of z, checking relations between Airy functions $\mathrm{Ai}(z)$, $\mathrm{Ai}'(z)$, $\mathrm{Bi}(z)$ and $\mathrm{Bi}'(z)$, have shown errors limited to the least significant 3 – 4 digits of precision.

8 Further Comments

Note that if the function is required to operate on a real argument only, then it may be much cheaper to call S17AGF or S17AJF.

9 Example

This example prints a caption and then proceeds to read sets of data from the input data stream. The first datum is a value for the parameter DERIV, the second is a complex value for the argument, Z, and the third is a character value to set the parameter SCAL. The program calls the routine and prints the results. The process is repeated until the end of the input data stream is encountered.

9.1 Program Text

```
    Program s17dgfe

!     S17DGF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s17dgf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Complex (Kind=nag_wp)            :: ai, z
      Integer                          :: ifail, ioerr, nz
      Character (1)                    :: deriv, scal
!     .. Executable Statements ..
      Write (nout,*) 'S17DGF Example Program Results'

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

      Write (nout,*)
      Write (nout,*) 'DERIV          Z         SCAL           AI          NZ'
      Write (nout,*)
```

```
data: Do
        Read (nin,*,Iostat=ioerr) deriv, z, scal

        If (ioerr<0) Then
          Exit data
        End If

        ifail = 0
        Call s17dgf(deriv,z,scal,ai,nz,ifail)

        Write (nout,99999) deriv, z, scal, ai, nz
      End Do data

99999 Format (3X,A,'   (',F8.4,',',F8.4,')   ',A,'   (',F8.4,',',F8.4,')',I4)
    End Program s17dgfe
```

9.2 Program Data

```
S17DGF Example Program Data
 'F'   ( 0.3,  0.4)    'U'
 'F'   ( 0.2,  0.0)    'U'
 'F'   ( 1.1, -6.6)    'U'
 'F'   ( 1.1, -6.6)    'S'
 'D'   (-1.0,  0.0)    'U'
```

9.3 Program Results

```
 S17DGF Example Program Results

 DERIV           Z          SCAL           AI           NZ

   F   (  0.3000,  0.4000)   U   (  0.2716, -0.1002)   0
   F   (  0.2000,  0.0000)   U   (  0.3037,  0.0000)   0
   F   (  1.1000, -6.6000)   U   (-43.6632,-47.9030)   0
   F   (  1.1000, -6.6000)   S   (  0.1655,  0.0597)   0
   D   ( -1.0000,  0.0000)   U   ( -0.0102,  0.0000)   0
```