NAG Fortran 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, SCALE, AI, NZ, IFAIL)
INTEGER            NZ, IFAIL
complex            Z, AI
CHARACTER*1        DERIV, SCALE
```

3 Description

This subroutine 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 nonnegative 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', $Ai(z)$ is returned.

If DERIV = 'D', $Ai'(z)$ is returned.

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

2: Z – ***complex*** *Input*

On entry: the argument z of the function.

3: SCALE – CHARACTER*1 *Input*

On entry: the scaling option.

If SCALE = 'U', the result is returned unscaled.

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

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

4: AI – ***complex*** *Output*

On exit: the required function or derivative value.

5: NZ – INTEGER *Output*

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

If NZ = 0, AI is not set to zero.

If NZ = 1, AI is set to zero.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

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, for users 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.**

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 ≠ 'F' or 'D'.
or SCALE ≠ '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 SCALE = '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 subroutine 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

The example program 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 value for the parameter SCALE. 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

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.

```
*     S17DGF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      complex          AI, Z
      INTEGER          IFAIL, NZ
      CHARACTER*1      DERIV, SCALE
*     .. External Subroutines ..
      EXTERNAL         S17DGF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S17DGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +  'DERIV          Z         SCALE          AI          NZ  IFAIL'
      WRITE (NOUT,*)
   20 READ (NIN,*,END=40) DERIV, Z, SCALE
      IFAIL = 0
*
      CALL S17DGF(DERIV,Z,SCALE,AI,NZ,IFAIL)
*
      WRITE (NOUT,99999) DERIV, Z, SCALE, AI, NZ, IFAIL
      GO TO 20
   40 STOP
*
99999 FORMAT (3X,A,'   (',F8.4,',',F8.4,')   ',A,'   (',F8.4,',',F8.4,
     +       ')',I4,I5)
```

```
      END
```

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          SCALE           AI          NZ  IFAIL

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