NAG Fortran Library Routine Document

S17ALF

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

S17ALF determines the leading N zeros of one of the Bessel functions $J_\alpha(x)$, $Y_\alpha(x)$, $J'_\alpha(x)$ or $Y'_\alpha(x)$ for real x and non-negative α.

2 Specification

```
SUBROUTINE S17ALF(A, N, MODE, REL, X, IFAIL)
INTEGER           N, MODE, IFAIL
real              A, REL, X(N)
```

3 Description

This routine attempts to find the leading N zeros of one of the Bessel functions $J_\alpha(x)$, $Y_\alpha(x)$, $J'_\alpha(x)$ or $Y'_\alpha(x)$, where x is real. When α is real, these functions each have an infinite number of real zeros, all of which are simple with the possible exception of $x = 0$. If $\alpha \geq 0$, the nth positive zero is denoted by $j_{\alpha,n}, j'_{\alpha,n}, y_{\alpha,n}$ and $y'_{\alpha,n}$, respectively, for $n = 1, 2, \ldots, N$, except that $x = 0$ is counted as the first zero of $J'_\alpha(x)$ when $\alpha = 0$. Since $J'_0(x) = -J_1(x)$, it therefore follows that $j'_{0,1} = 0$ and $j'_{0,n} = -j_{1,n-1}$ for $n = 2, 3, \ldots, N-1$. Further details can be found in Section 9.5 of Abramowitz and Stegun (1972).

S17ALF is based on Algol 60 procedures given by Temme (1979). Initial approximations to the zeros are computed from asymptotic expansions. These are then improved by higher-order Newton iteration making use of the differential equation for the Bessel functions.

4 References

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

Temme N M (1979) An algorithm with Algol 60 program for the computation of the zeros of ordinary Bessel functions and those of their derivatives *J. Comput. Phys.* **32** 270–279

Temme N M (1976) On the numerical evaluation of the ordinary Bessel function of the second kind *J. Comput. Phys.* **21** 343–350

5 Parameters

1: A – ***real*** *Input*

On entry: the order α of the function.

Constraint: $0.0 \leq \text{A} \leq 100000.0$.

2: N – INTEGER *Input*

On entry: the number N of zeros required.

Constraint: $\text{N} \geq 1$.

3: MODE – INTEGER *Input*

On entry: specifies the form of the function whose zeros are required as follows:

if MODE $= 1$, then the zeros of $J_\alpha(x)$ are required;

if MODE $= 2$, then the zeros of $Y_\alpha(x)$ are required;

if MODE $= 3$, then the zeros of $J'_\alpha(x)$ are required;

if MODE $= 4$, then the zeros of $Y'_\alpha(x)$ are required.

Constraint: $1 \leq \text{MODE} \leq 4$.

4: REL – ***real*** *Input*

On entry: the relative accuracy to which the zeros are required.

Suggested value: the square root of the ***machine precision***.

Constraint: REL > 0.0.

5: X(N) – ***real*** array *Output*

On exit: the N required zeros of the function specified by MODE.

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, A < 0.0,
or A > 100000.0,
or N ≤ 0,
or MODE < 1,
or MODE > 4,
or REL ≤ 0.0.

7 Accuracy

If the value of REL is set to 10^{-d}, then the required zeros should have approximately d correct significant digits.

8 Further Comments

None.

9 Example

To determine the leading five positive zeros of the Bessel function $J_0(x)$.

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.

```
*     S17ALF Example Program Text.
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=100)
      real             ZERO
      PARAMETER        (ZERO=0.0e+0)
*     .. Local Scalars ..
      real             A, REL
      INTEGER          I, IFAIL, MODE, N
*     .. Local Arrays ..
      real             X(NMAX)
*     .. External Functions ..
      real             X02AJF
      EXTERNAL         X02AJF
*     .. External Subroutines ..
      EXTERNAL         S17ALF
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'S17ALF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      WRITE (NOUT,*)
      REL = SQRT(X02AJF())
      READ (NIN,*) A, N, MODE
      IFAIL = 0
*
      CALL S17ALF(A,N,MODE,REL,X,IFAIL)
*
      WRITE (NOUT,*) '   A   N   MODE   IFAIL          REL'
      WRITE (NOUT,*) '                            (machine-dependent)'
      WRITE (NOUT,*)
      WRITE (NOUT,99999) A, N, MODE, IFAIL, REL
      WRITE (NOUT,*)
      IF (MODE.EQ.1) THEN
         WRITE (NOUT,*) 'Leading N positive zeros of J'
      ELSE IF (MODE.EQ.2) THEN
         WRITE (NOUT,*) 'Leading N positive zeros of Y'
      ELSE IF (MODE.EQ.3) THEN
         IF (A.EQ.ZERO) THEN
            WRITE (NOUT,*) 'Leading N non-negative zeros of J'''
         ELSE
            WRITE (NOUT,*) 'Leading N positive zeros of J'''
         END IF
      ELSE IF (MODE.EQ.4) THEN
         WRITE (NOUT,*) 'Leading N positive zeros of Y'''
      END IF
      WRITE (NOUT,*)
      DO 20 I = 1, N
         WRITE (NOUT,99998) X(I)
   20 CONTINUE
      WRITE (NOUT,*)
*
      STOP
*
99999 FORMAT (1X,F4.1,I4,I7,I8,6X,1P,E9.1)
99998 FORMAT (' x = ',1P,E12.4)
      END
```

9.2 Program Data

```
S17ALF Example Program Data
 0.0  5  1 : Values of A, N and MODE
```

9.3 Program Results

```
 S17ALF Example Program Results

    A   N   MODE   IFAIL          REL
                          (machine-dependent)

  0.0   5      1       0        1.1E-08

 Leading N positive zeros of J

 x =   2.4048E+00
 x =   5.5201E+00
 x =   8.6537E+00
 x =   1.1792E+01
 x =   1.4931E+01
```