

NAG 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 (KIND=nag_wp) A, REL, X(N)
```

3 Description

S17ALF 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 n th positive zero is denoted by $j_{\alpha,n}$, $j'_{\alpha,n}$, $y_{\alpha,n}$ and $y'_{\alpha,n}$, respectively, for $n = 1, 2, \dots, N$, 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 $n = 2, 3, \dots, N - 1$. 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 (1976) On the numerical evaluation of the ordinary Bessel function of the second kind *J. Comput. Phys.* **21** 343–350

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

5 Arguments

- 1: A – REAL (KIND=nag_wp) *Input*
On entry: the order α of the function.
Constraint: $0.0 \leq A \leq 100000.0$.
- 2: N – INTEGER *Input*
On entry: the number N of zeros required.
Constraint: $N \geq 1$.

- 3: MODE – INTEGER *Input*
On entry: specifies the form of the function whose zeros are required.
MODE = 1
 The zeros of $J_\alpha(x)$ are required.
MODE = 2
 The zeros of $Y_\alpha(x)$ are required;
MODE = 3
 The zeros of $J'_\alpha(x)$ are required;
MODE = 4
 The zeros of $Y'_\alpha(x)$ are required.
Constraint: $1 \leq \text{MODE} \leq 4$.
- 4: REL – REAL (KIND=nag_wp) *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 (KIND=nag_wp) 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. 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

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

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

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

8 Parallelism and Performance

S17ALF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example determines the leading five positive zeros of the Bessel function $J_0(x)$.

10.1 Program Text

```

Program s17alf
!      S17ALF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
Use nag_library, Only: nag_wp, s17alf, x02ajf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: a, rel
Integer                     :: i, ifail, mode, n
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: x(:)
!      .. Intrinsic Procedures ..
Intrinsic                   :: sqrt
!      .. Executable Statements ..
Write (nout,*) 'S17ALF Example Program Results'

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

rel = sqrt(x02ajf())
Read (nin,*) a, n, mode
Allocate (x(n))

ifail = 0
Call s17alf(a,n,mode,rel,x,ifail)

Write (nout,*)
Write (nout,*) '  A  N  MODE          REL'
```

```

Write (nout,*) '                               (machine-dependent)'
Write (nout,*)
Write (nout,99999) a, n, mode, rel
Write (nout,*)

Select Case (mode)
Case (1)
  Write (nout,*) 'Leading N positive zeros of J'
Case (2)
  Write (nout,*) 'Leading N positive zeros of Y'
Case (3)

  If (a==0.OE0_nag_wp) Then
    Write (nout,*) 'Leading N non-negative zeros of J'''
  Else
    Write (nout,*) 'Leading N positive zeros of J'''
  End If

Case (4)
  Write (nout,*) 'Leading N positive zeros of Y'''
End Select

Write (nout,*)
Write (nout,*) 'X ='
Write (nout,99998)(x(i),i=1,n)
Write (nout,*)

99999 Format (1X,F4.1,I4,I7,4X,1P,E9.1)
99998 Format (1P,(E12.4))
End Program s17alfe

```

10.2 Program Data

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

10.3 Program Results

S17ALF Example Program Results

A	N	MODE	REL (machine-dependent)
0.0	5	1	1.1E-08

Leading N positive zeros of J

X =

```

2.4048E+00
5.5201E+00
8.6537E+00
1.1792E+01
1.4931E+01

```
