

NAG Library Routine Document

S17ASF

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

S17ASF returns an array of values of the Bessel function $J_0(x)$.

2 Specification

```
SUBROUTINE S17ASF (N, X, F, IVALID, IFAIL)
  INTEGER          N, IVALID(N), IFAIL
  REAL (KIND=nag_wp) X(N), F(N)
```

3 Description

S17ASF evaluates an approximation to the Bessel function of the first kind $J_0(x_i)$ for an array of arguments x_i , for $i = 1, 2, \dots, n$.

Note: $J_0(-x) = J_0(x)$, so the approximation need only consider $x \geq 0$.

The routine is based on three Chebyshev expansions:

For $0 < x \leq 8$,

$$J_0(x) = \sum_{r=0} a_r T_r(t), \quad \text{with } t = 2\left(\frac{x}{8}\right)^2 - 1.$$

For $x > 8$,

$$J_0(x) = \sqrt{\frac{2}{\pi x}} \left\{ P_0(x) \cos\left(x - \frac{\pi}{4}\right) - Q_0(x) \sin\left(x - \frac{\pi}{4}\right) \right\},$$

where $P_0(x) = \sum_{r=0} b_r T_r(t)$,

and $Q_0(x) = \frac{8}{x} \sum_{r=0} c_r T_r(t)$,

with $t = 2\left(\frac{8}{x}\right)^2 - 1$.

For x near zero, $J_0(x) \simeq 1$. This approximation is used when x is sufficiently small for the result to be correct to *machine precision*.

For very large x , it becomes impossible to provide results with any reasonable accuracy (see Section 7), hence the routine fails. Such arguments contain insufficient information to determine the phase of oscillation of $J_0(x)$; only the amplitude, $\sqrt{\frac{2}{\pi|x|}}$, can be determined and this is returned on softfailure.

The range for which this occurs is roughly related to *machine precision*; the routine will fail if $|x| \gtrsim 1/\textit{machine precision}$ (see the Users' Note for your implementation for details).

4 References

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

Clenshaw C W (1962) Chebyshev Series for Mathematical Functions *Mathematical tables* HMSO

5 Arguments

1: N – INTEGER *Input*

On entry: n , the number of points.

Constraint: $N \geq 0$.

2: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the argument x_i of the function, for $i = 1, 2, \dots, N$.

3: F(N) – REAL (KIND=nag_wp) array *Output*

On exit: $J_0(x_i)$, the function values.

4: IVALID(N) – INTEGER array *Output*

On exit: IVALID(i) contains the error code for x_i , for $i = 1, 2, \dots, N$.

IVALID(i) = 0

No error.

IVALID(i) = 1

On entry, x_i is too large. F(i) contains the amplitude of the J_0 oscillation, $\sqrt{\frac{2}{\pi|x_i|}}$.

5: 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, at least one value of X was invalid.

Check IVALID for more information.

IFAIL = 2

On entry, $N = \langle value \rangle$.

Constraint: $N \geq 0$.

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

Let δ be the relative error in the argument and E be the absolute error in the result. (Since $J_0(x)$ oscillates about zero, absolute error and not relative error is significant.)

If δ is somewhat larger than the *machine precision* (e.g., if δ is due to data errors etc.), then E and δ are approximately related by:

$$E \simeq |xJ_1(x)|\delta$$

(provided E is also within machine bounds). Figure 1 displays the behaviour of the amplification factor $|xJ_1(x)|$.

However, if δ is of the same order as *machine precision*, then rounding errors could make E slightly larger than the above relation predicts.

For very large x , the above relation ceases to apply. In this region, $J_0(x) \simeq \sqrt{\frac{2}{\pi|x|}} \cos\left(x - \frac{\pi}{4}\right)$. The amplitude $\sqrt{\frac{2}{\pi|x|}}$ can be calculated with reasonable accuracy for all x , but $\cos\left(x - \frac{\pi}{4}\right)$ cannot. If $x - \frac{\pi}{4}$ is written as $2N\pi + \theta$ where N is an integer and $0 \leq \theta < 2\pi$, then $\cos\left(x - \frac{\pi}{4}\right)$ is determined by θ only. If $x \gtrsim \delta^{-1}$, θ cannot be determined with any accuracy at all. Thus if x is greater than, or of the order of, the inverse of the *machine precision*, it is impossible to calculate the phase of $J_0(x)$ and the routine must fail.

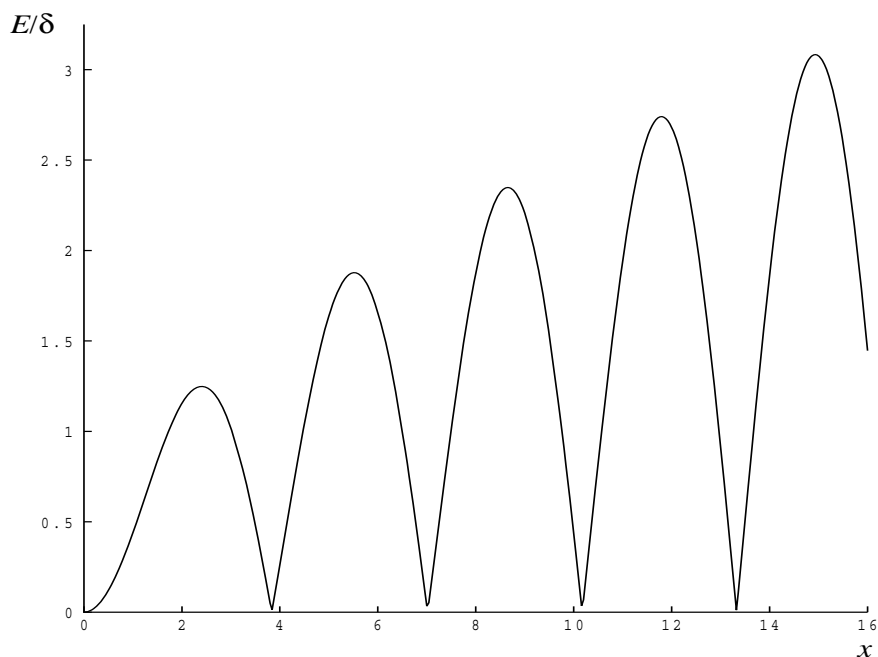


Figure 1

8 Parallelism and Performance

S17ASF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads values of X from a file, evaluates the function at each value of x_i and prints the results.

10.1 Program Text

```

Program s17asfe

!      S17ASF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, s17asf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Integer                    :: i, ifail, n
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: f(:), x(:)
Integer, Allocatable       :: ivalid(:)
!      .. Executable Statements ..
Write (nout,*) 'S17ASF Example Program Results'

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

Write (nout,*)
Write (nout,*) '      X      F      IVALID'
Write (nout,*)

Read (nin,*) n

Allocate (x(n),f(n),ivalid(n))

Read (nin,*) x(1:n)

ifail = 0
Call s17asf(n,x,f,ivalid,ifail)

Do i = 1, n
  Write (nout,99999) x(i), f(i), ivalid(i)
End Do

99999 Format (1X,1P,2E12.3,I5)
End Program s17asfe

```

10.2 Program Data

S17ASF Example Program Data

9

0.0 0.5 1.0 3.0 6.0 8.0 10.0 -1.0 1000.0

10.3 Program Results

S17ASF Example Program Results

X	F	INVALID
0.000E+00	1.000E+00	0
5.000E-01	9.385E-01	0
1.000E+00	7.652E-01	0
3.000E+00	-2.601E-01	0
6.000E+00	1.506E-01	0
8.000E+00	1.717E-01	0
1.000E+01	-2.459E-01	0
-1.000E+00	7.652E-01	0
1.000E+03	2.479E-02	0
