NAG Toolbox

nag_specfun_bessel_y1_real_vector (s17ar)

1 Purpose

nag_specfun_bessel_y1_real_vector (s17ar) returns an array of values of the Bessel function $Y_1(x)$.

2 Syntax

```
[f, ivalid, ifail] = nag_specfun_bessel_y1_real_vector(x, 'n', n)
[f, ivalid, ifail] = s17ar(x, 'n', n)
```

3 Description

nag_specfun_bessel_y1_real_vector (s17ar) evaluates an approximation to the Bessel function of the second kind $Y_1(x_i)$ for an array of arguments x_i, for $i = 1, 2, \dots, n$.

Note: $Y_1(x)$ is undefined for $x \leq 0$ and the function will fail for such arguments.

The function is based on four Chebyshev expansions:

For $0 < x \leq 8$,

$$Y_1(x) = \frac{2}{\pi} \ln x \frac{x}{8} \sum_{r=0} a_r T_r(t) - \frac{2}{\pi x} + \frac{x}{8} \sum_{r=0} b_r T_r(t), \qquad \text{with } t = 2\left(\frac{x}{8}\right)^2 - 1.$$

For $x > 8$,

$$Y_1(x) = \sqrt{\frac{2}{\pi x}} \left\{ P_1(x) \sin\left(x - 3\frac{\pi}{4}\right) + Q_1(x) \cos\left(x - 3\frac{\pi}{4}\right) \right\}$$

where $P_1(x) = \sum_{r=0} c_r T_r(t)$,

and $Q_1(x) = \frac{8}{x} \sum_{r=0} d_r T_r(t)$, with $t = 2\left(\frac{8}{x}\right)^2 - 1$.

For x near zero, $Y_1(x) \simeq -\frac{2}{\pi x}$. This approximation is used when x is sufficiently small for the result to be correct to ***machine precision***. For extremely small x, there is a danger of overflow in calculating $-\frac{2}{\pi x}$ and for such arguments the function will fail.

For very large x, it becomes impossible to provide results with any reasonable accuracy (see Section 7), hence the function fails. Such arguments contain insufficient information to determine the phase of oscillation of $Y_1(x)$; 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 function will fail if $x \gtrsim 1/$***machine precision***.

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 Parameters

5.1 Compulsory Input Parameters

1: **x**(**n**) – REAL (KIND=nag_wp) array

The argument x_i of the function, for $i = 1, 2, \ldots, \mathbf{n}$.

Constraint: $\mathbf{x}(i) > 0.0$, for $i = 1, 2, \ldots, \mathbf{n}$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **x**.

n, the number of points.

Constraint: $\mathbf{n} \geq 0$.

5.3 Output Parameters

1: **f**(**n**) – REAL (KIND=nag_wp) array

$Y_1(x_i)$, the function values.

2: **ivalid**(**n**) – INTEGER array

$\mathbf{ivalid}(i)$ contains the error code for x_i, for $i = 1, 2, \ldots, \mathbf{n}$.

$\mathbf{ivalid}(i) = 0$
No error.

$\mathbf{ivalid}(i) = 1$

On entry, x_i is too large. $\mathbf{f}(i)$ contains the amplitude of the Y_1 oscillation, $\sqrt{\frac{2}{\pi x_i}}$.

$\mathbf{ivalid}(i) = 2$

On entry, $x_i \leq 0.0$, Y_1 is undefined. $\mathbf{f}(i)$ contains 0.0.

$\mathbf{ivalid}(i) = 3$
x_i is too close to zero, there is a danger of overflow. On softfailure, $\mathbf{f}(i)$ contains the value of $Y_1(x)$ at the smallest valid argument.

3: **ifail** – INTEGER

$\mathbf{ifail} = 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

$\mathbf{ifail} = 1$ (*warning*)

On entry, at least one value of **x** was invalid.
Check **ivalid** for more information.

$\mathbf{ifail} = 2$

Constraint: $\mathbf{n} \geq 0$.

$\mathbf{ifail} = -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Let δ be the relative error in the argument and E be the absolute error in the result. (Since $Y_1(x)$ oscillates about zero, absolute error and not relative error is significant, except for very small x.)

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:

$$E \simeq |xY_0(x) - Y_1(x)|\delta$$

(provided E is also within machine bounds). Figure 1 displays the behaviour of the amplification factor $|xY_0(x) - Y_1(x)|$.

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 small x, absolute error becomes large, but the relative error in the result is of the same order as δ.

For very large x, the above relation ceases to apply. In this region, $Y_1(x) \simeq \sqrt{\frac{2}{\pi x}} \sin\left(x - \frac{3\pi}{4}\right)$. The amplitude $\sqrt{\frac{2}{\pi x}}$ can be calculated with reasonable accuracy for all x, but $\sin\left(x - \frac{3\pi}{4}\right)$ cannot. If $x - \frac{3\pi}{4}$ is written as $2N\pi + \theta$ where N is an integer and $0 \leq \theta < 2\pi$, then $\sin\left(x - \frac{3\pi}{4}\right)$ is determined by θ only. If $x > \delta^{-1}$, 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 $Y_1(x)$ and the function must fail.

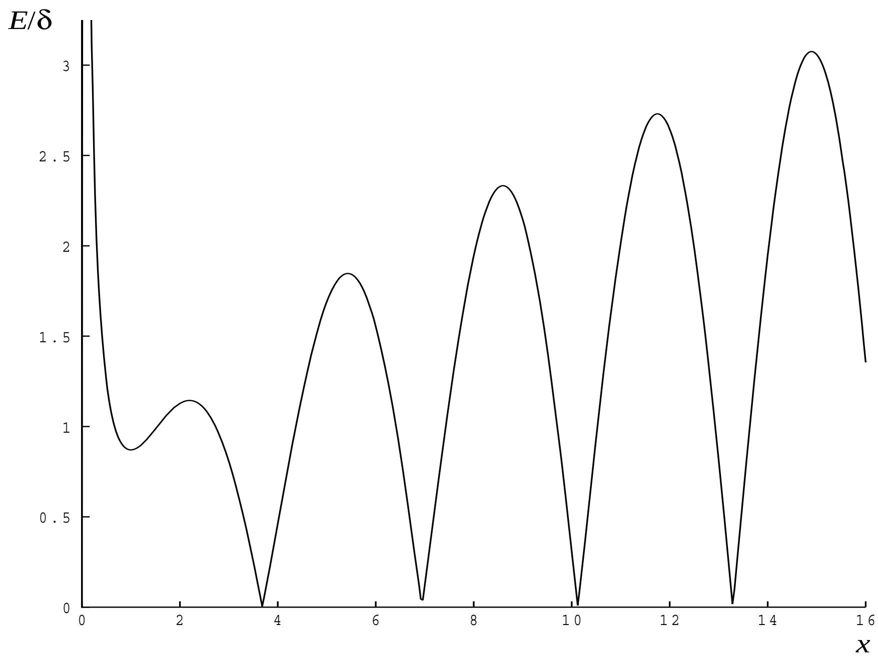

Figure 1

8 Further Comments

None.

9 Example

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

9.1 Program Text

```
function s17ar_example

fprintf('s17ar example results\n\n');

x = [0.5; 1; 3; 6; 8; 10; 1000];

[f, ivalid, ifail] = s17ar(x);

fprintf('     x           Y_1(x)   ivalid\n');
for i=1:numel(x)
  fprintf('%12.3e%12.3e%5d\n', x(i), f(i), ivalid(i));
end
```

9.2 Program Results

```
s17ar example results

     x           Y_1(x)   ivalid
   5.000e-01  -1.471e+00    0
   1.000e+00  -7.812e-01    0
   3.000e+00   3.247e-01    0
   6.000e+00  -1.750e-01    0
   8.000e+00  -1.581e-01    0
   1.000e+01   2.490e-01    0
   1.000e+03  -2.478e-02    0
```