NAG Toolbox

nag_specfun_bessel_k1_real_vector (s18ar)

1 Purpose

nag_specfun_bessel_k1_real_vector (s18ar) returns an array of values of the modified Bessel function $K_1(x)$.

2 Syntax

```
[f, ivalid, ifail] = nag_specfun_bessel_k1_real_vector(x, 'n', n)
[f, ivalid, ifail] = s18ar(x, 'n', n)
```

3 Description

nag_specfun_bessel_k1_real_vector (s18ar) evaluates an approximation to the modified Bessel function of the second kind $K_1(x_i)$ for an array of arguments x_i, for $i = 1, 2, \ldots, n$.

Note: $K_1(x)$ is undefined for $x \leq 0$ and the function will fail for such arguments.

The function is based on five Chebyshev expansions:

For $0 < x \leq 1$,

$$K_1(x) = \frac{1}{x} + x \ln x \sum_{r=0} a_r T_r(t) - x \sum_{r=0} b_r T_r(t), \qquad \text{where } t = 2x^2 - 1.$$

For $1 < x \leq 2$,

$$K_1(x) = e^{-x} \sum_{r=0} c_r T_r(t), \qquad \text{where } t = 2x - 3.$$

For $2 < x \leq 4$,

$$K_1(x) = e^{-x} \sum_{r=0} d_r T_r(t), \qquad \text{where } t = x - 3.$$

For $x > 4$,

$$K_1(x) = \frac{e^{-x}}{\sqrt{x}} \sum_{r=0} e_r T_r(t), \qquad \text{where } t = \frac{9 - x}{1 + x}.$$

For x near zero, $K_1(x) \simeq \frac{1}{x}$. This approximation is used when x is sufficiently small for the result to be correct to ***machine precision***. For very small x it is impossible to calculate $\frac{1}{x}$ without overflow and the function must fail.

For large x, where there is a danger of underflow due to the smallness of K_1, the result is set exactly to zero.

4 References

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

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: **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: **n** ≥ 0.

5.3 Output Parameters

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

$K_1(x_i)$, the function values.

2: **ivalid**(**n**) – INTEGER array

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

ivalid$(i) = 0$
No error.

ivalid$(i) = 1$
$x_i \leq 0.0$, $K_1(x_i)$ is undefined. **f**(i) contains 0.0.

ivalid$(i) = 2$
x_i is too small, there is a danger of overflow. **f**(i) contains zero. The threshold value is the same as for **ifail** $= 2$ in nag_specfun_bessel_k1_real (s18ad), as defined in the Users' Note for your implementation.

3: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

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

ifail $= 2$

Constraint: **n** ≥ 0.

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 and ϵ be the relative errors in the argument and result respectively.

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

$$\epsilon \simeq \left| \frac{xK_0(x) - K_1(x)}{K_1(x)} \right| \delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left| \frac{xK_0(x) - K_1(x)}{K_1(x)} \right|.$$

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

For small x, $\epsilon \simeq \delta$ and there is no amplification of errors.

For large x, $\epsilon \simeq x\delta$ and we have strong amplification of the relative error. Eventually K_1, which is asymptotically given by $\frac{e^{-x}}{\sqrt{x}}$, becomes so small that it cannot be calculated without underflow and hence the function will return zero. Note that for large x the errors will be dominated by those of the standard function exp.

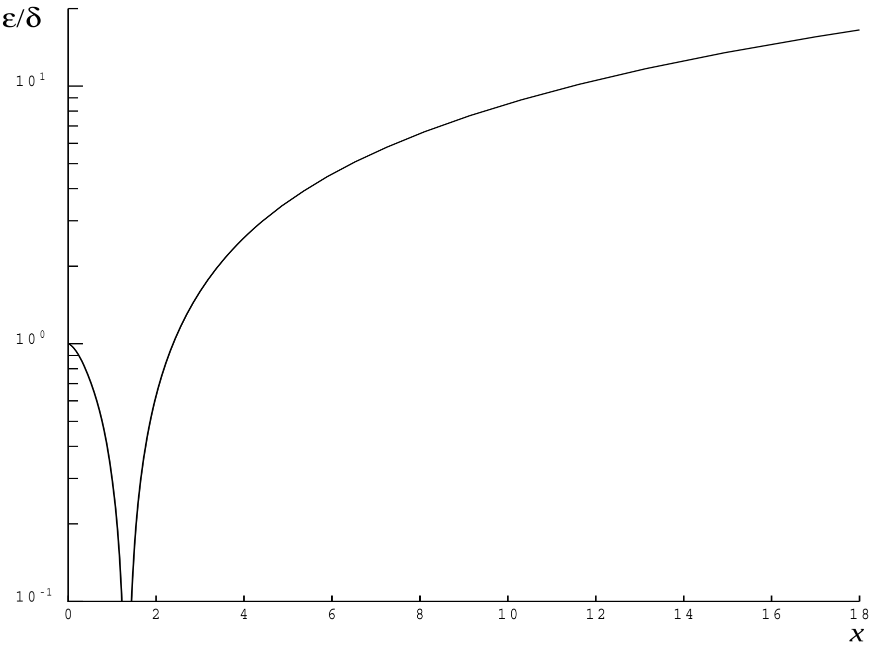

Figure 1

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 s18ar_example

fprintf('s18ar example results\n\n');

x = [0.4; 0.6; 1.4; 1.6; 2.5; 3.5; 6; 8; 10; 1000];

[f, ivalid, ifail] = s18ar(x);

fprintf('     x           K_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

```
s18ar example results

     x           K_1(x)   ivalid
   4.000e-01   2.184e+00    0
   6.000e-01   1.303e+00    0
   1.400e+00   3.208e-01    0
   1.600e+00   2.406e-01    0
   2.500e+00   7.389e-02    0
   3.500e+00   2.224e-02    0
   6.000e+00   1.344e-03    0
   8.000e+00   1.554e-04    0
   1.000e+01   1.865e-05    0
   1.000e+03   0.000e+00    0
```