NAG Toolbox

nag_specfun_bessel_i0_real (s18ae)

1 Purpose

nag_specfun_bessel_i0_real (s18ae) returns the value of the modified Bessel function $I_0(x)$, via the function name.

2 Syntax

```
[result, ifail] = nag_specfun_bessel_i0_real(x)
[result, ifail] = s18ae(x)
```

3 Description

nag_specfun_bessel_i0_real (s18ae) evaluates an approximation to the modified Bessel function of the first kind $I_0(x)$.

Note: $I_0(-x) = I_0(x)$, so the approximation need only consider $x \geq 0$.

The function is based on three Chebyshev expansions:

For $0 < x \leq 4$,

$$I_0(x) = e^x \sum_{r=0} a_r T_r(t), \qquad \text{where } t = 2\left(\frac{x}{4}\right) - 1.$$

For $4 < x \leq 12$,

$$I_0(x) = e^x \sum_{r=0} b_r T_r(t), \qquad \text{where } t = \frac{x-8}{4}.$$

For $x > 12$,

$$I_0(x) = \frac{e^x}{\sqrt{x}} \sum_{r=0} c_r T_r(t), \qquad \text{where } t = 2\left(\frac{12}{x}\right) - 1.$$

For small x, $I_0(x) \simeq 1$. This approximation is used when x is sufficiently small for the result to be correct to ***machine precision***.

For large x, the function must fail because of the danger of overflow in calculating e^x.

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** – REAL (KIND=nag_wp)

The argument x of the function.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **result**

The result of the function.

2: **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 $= 1$

$|\mathbf{x}|$ is too large. On softfailure the function returns the approximate value of $I_0(x)$ at the nearest valid argument.

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:

$$\epsilon \simeq \left|\frac{xI_1(x)}{I_0(x)}\right|\delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left|\frac{xI_1(x)}{I_0(x)}\right|.$$

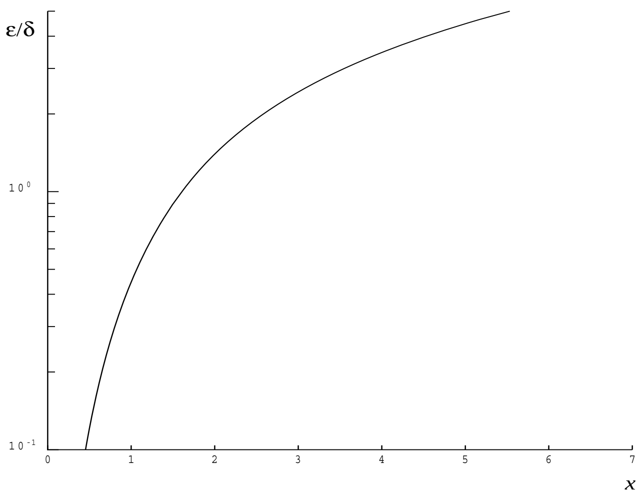

Figure 1

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

For small x the amplification factor is approximately $\frac{x^2}{2}$, which implies strong attenuation of the error, but in general ϵ can never be less than the ***machine precision***.

For large x, $\epsilon \simeq x\delta$ and we have strong amplification of errors. However the function must fail for quite moderate values of x, because $I_0(x)$ would overflow; hence in practice the loss of accuracy for large x is not excessive. Note that for large x the errors will be dominated by those of the standard function exp.

8 Further Comments

None.

9 Example

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

9.1 Program Text

```
    function s18ae_example

fprintf('s18ae example results\n\n');

x = [0    0.5    1    3    6    8    10    15    20   -1];
n = size(x,2);
result = x;

for j=1:n
  [result(j), ifail] = s18ae(x(j));
end

disp('      x           I_0(x)');
fprintf('%12.3e%12.3e\n',[x; result]);

s18ae_plot;

function s18ae_plot
  x = [0:0.2:4];
  for j = 1:numel(x)
    [I0(j), ifail] = s18ae(x(j));
  end

  fig1 = figure;
  plot(x,I0,'-r');
  xlabel('x');
  ylabel('I_0(x)');
  title('Bessel Function I_0(x)');
  axis([0 4 0 12]);

  % print(fig1,'-dpng','-r75','s18ae_fig1.png');
  % print(fig1,'-deps','-r75','s18ae_fig1.eps');
```

9.2 Program Results

```
s18ae example results

      x           I_0(x)
   0.000e+00   1.000e+00
   5.000e-01   1.063e+00
   1.000e+00   1.266e+00
   3.000e+00   4.881e+00
   6.000e+00   6.723e+01
```

```
 8.000e+00   4.276e+02
 1.000e+01   2.816e+03
 1.500e+01   3.396e+05
 2.000e+01   4.356e+07
-1.000e+00   1.266e+00
```