

NAG Toolbox

nag_specfun_bessel_i1_real (s18af)

1 Purpose

nag_specfun_bessel_i1_real (s18af) returns a value for the modified Bessel function $I_1(x)$, via the function name.

2 Syntax

```
[result, ifail] = nag_specfun_bessel_i1_real(x)
[result, ifail] = s18af(x)
```

3 Description

nag_specfun_bessel_i1_real (s18af) evaluates an approximation to the modified Bessel function of the first kind $I_1(x)$.

Note: $I_1(-x) = -I_1(x)$, so the approximation need only consider $x \geq 0$.

The function is based on three Chebyshev expansions:

For $0 < x \leq 4$,

$$I_1(x) = x \sum_{r=0} a_r T_r(t), \quad \text{where } t = 2\left(\frac{x}{4}\right)^2 - 1;$$

For $4 < x \leq 12$,

$$I_1(x) = e^x \sum_{r=0} b_r T_r(t), \quad \text{where } t = \frac{x-8}{4};$$

For $x > 12$,

$$I_1(x) = \frac{e^x}{\sqrt{x}} \sum_{r=0} c_r T_r(t), \quad \text{where } t = 2\left(\frac{12}{x}\right) - 1.$$

For small x , $I_1(x) \simeq x$. 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 $I_1(x)$ cannot be represented without overflow.

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

x is too large. On softfailure the function returns the approximate value of $I_1(x)$ 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:

$$\epsilon \simeq \left| \frac{xI_0(x) - I_1(x)}{I_1(x)} \right| \delta.$$

Figure 1 shows the behaviour of the error amplification factor

$$\left| \frac{xI_0(x) - I_1(x)}{I_1(x)} \right|.$$

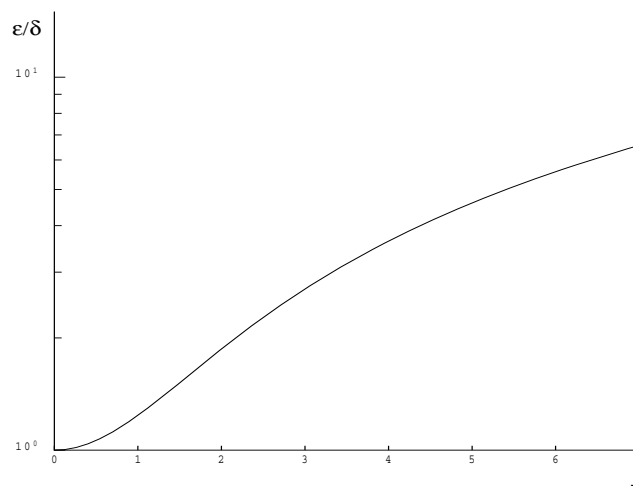


Figure 1

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

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 $I_1(x)$ 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 s18af_example

fprintf('s18af 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] = s18af(x(j));
end

disp('      x          I_1(x)');
fprintf('%12.3e%12.3e\n',[x; result]);

s18af_plot;

function s18af_plot
    x = [0:0.2:4];
    for j = 1:numel(x)
        [I1(j), ifail] = s18af(x(j));
    end

    fig1 = figure;
    plot(x,I1,'-r');
    xlabel('x');
    ylabel('I_1(x)');
    title('Bessel Function I_1(x)');
    axis([0 4 0 10]);

    % print(fig1,'-dpng','-r75','s18af_fig1.png');
    % print(fig1,'-deps','-r75','s18af_fig1.eps');
```

9.2 Program Results

```
s18af example results

      x          I_1(x)
0.000e+00    0.000e+00
5.000e-01    2.579e-01
1.000e+00    5.652e-01
3.000e+00    3.953e+00
6.000e+00    6.134e+01
```

```
8.000e+00  3.999e+02
1.000e+01  2.671e+03
1.500e+01  3.281e+05
2.000e+01  4.245e+07
-1.000e+00 -5.652e-01
```

