

NAG Library Function Document

nag_bessel_i1_scaled (s18cfc)

1 Purpose

nag_bessel_i1_scaled (s18cfc) returns a value of the scaled modified Bessel function $e^{-|x|}I_1(x)$.

2 Specification

```
#include <nag.h>
#include <nags.h>
double nag_bessel_i1_scaled (double x)
```

3 Description

nag_bessel_i1_scaled (s18cfc) evaluates an approximation to $e^{-|x|}I_1(x)$, where I_1 is a modified Bessel function of the first kind. The scaling factor $e^{-|x|}$ removes most of the variation in $I_1(x)$.

The function uses the same Chebyshev expansions as nag_bessel_i1 (s18afc), which returns the unscaled value of $I_1(x)$.

4 References

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

5 Arguments

1: **x** – double *Input*
On entry: the argument x of the function.

6 Error Indicators and Warnings

None.

7 Accuracy

Relative errors in the argument are attenuated when propagated into the function value. When the accuracy of the argument is essentially limited by the *machine precision*, the accuracy of the function value will be similarly limited by at most a small multiple of the *machine precision*.

8 Parallelism and Performance

nag_bessel_i1_scaled (s18cfc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

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

10.1 Program Text

```

/* nag_bessel_il_scaled (s18cfc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
    Integer exit_status = 0;
    double x, y;

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
    printf("nag_bessel_il_scaled (s18cfc) Example Program Results\n");
    printf("      x              y\n");
#ifdef _WIN32
    while (scanf_s("%lf", &x) != EOF)
#else
    while (scanf("%lf", &x) != EOF)
#endif
    {
        /* nag_bessel_il_scaled (s18cfc).
         * Scaled modified Bessel function exp(-|x|) I_1(x)
         */
        y = nag_bessel_il_scaled(x);
        printf("%12.3e%12.3e\n", x, y);
    }

    return exit_status;
}

```

10.2 Program Data

```

nag_bessel_il_scaled (s18cfc) Example Program Data
      0.0
      0.5
      1.0
      3.0
      6.0
     10.0
    1000.0
     -1.0

```

10.3 Program Results

nag_bessel_i1_scaled (s18cfc) Example Program Results

x	y
0.000e+00	0.000e+00
5.000e-01	1.564e-01
1.000e+00	2.079e-01
3.000e+00	1.968e-01
6.000e+00	1.521e-01
1.000e+01	1.213e-01
1.000e+03	1.261e-02
-1.000e+00	-2.079e-01
