NAG C Library Function Document

nag_bessel_i_nu (s18eec)

1 Purpose

nag_bessel_i_nu (s18eec) returns the value of the modified Bessel function $I_{\nu/4}(x)$ for real $x > 0$.

2 Specification

double nag_bessel_i_nu (double x, Integer nu, NagError *fail)

3 Description

This routine evaluates an approximation to the modified Bessel function of the first kind $I_{\nu/4}(x)$, where the order $\nu = -3, -2, -1, 1, 2$ or 3 and $x$ is real and positive. For positive orders it may also be called with $x = 0$, since $I_{\nu/4}(0) = 0$ when $\nu > 0$. For negative orders the formula

$$I_{-\nu/4}(x) = I_{\nu/4}(x) + \frac{\nu}{2\pi} \sin(\nu x) K_{\nu/4}(x)$$

is used.

4 Parameters

1:  x – double

   *Input*

   On entry: the argument $x$ of the function.

   Constraints:
   
   $x > 0.0$ when nu $< 0$,
   
   $x \geq 0.0$ when nu $> 0$.

2:  nu – Integer

   *Input*

   On entry: the argument $\nu$ of the function.

   Constraint: $1 \leq \text{abs}(\text{nu}) \leq 3$.

3:  fail – NagError *

   *Input/Output*

   The NAG error parameter (see the Essential Introduction).

5 Error Indicators and Warnings

**NE_REAL_INT**

   On entry, x = <value>, nu = <value>.

   Constraint: $x > 0.0$ when nu $< 0$.

   On entry, x = <value>, nu = <value>.

   Constraint: $x \geq 0.0$ when nu $> 0$.

**NE_INT**

   On entry, nu = <value>.

   Constraint: $1 \leq \text{abs}(\text{nu}) \leq 3$.
NE_OVERFLOWLIKELY
The evaluation has been abandoned due to the likelihood of overflow. The result is returned as zero.

NW_SOME_PRECISION_LOSS
The evaluation has been completed but some precision has been lost.

NE_TOTAL_PRECISION_LOSS
The evaluation has been abandoned due to total loss of precision. The result is returned as zero.

NE_TERMINATION_FAILURE
The evaluation has been abandoned due to failure to satisfy the termination condition. The result is returned as zero.

NE_INTERNAL_ERROR
An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

6 Further Comments
6.1 Accuracy
All constants in the underlying functions are specified to approximately 18 digits of precision. If \( t \) denotes the number of digits of precision in the floating-point arithmetic being used, then clearly the maximum number of correct digits in the results obtained is limited by \( p = \min(t, 18) \). Because of errors in argument reduction when computing elementary functions inside the underlying functions, the actual number of correct digits is limited, in general, by \( p - s \), where \( s \approx \max(1, |\log_{10} x|) \) represents the number of digits lost due to the argument reduction. Thus the larger the value of \( x \), the less the precision in the result.

6.2 References

7 See Also
None.

8 Example
The example program reads values of the arguments \( x \) and \( \nu \) from a file, evaluates the function and prints the results.

8.1 Program Text
/* nag_bessel_i_nu (s18ec) Example Program. */
* *
* Copyright 2000 Numerical Algorithms Group.
* *
* NAG C Library
* *
* Mark 6, 2000.
*/

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

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

    INIT_FAIL(fail);
    Vprintf("s18ec Example Program Results\n\n");

    Vprintf(" x   nu   y\n");

    /* Skip heading in data file */
    Vscanf("%*[\n"];
    while (scanf("%lf %ld%*[\n", &x, &nu) != EOF)
    {
        y = s18ec(x, nu, &fail);
        if (fail.code == NE_NOERROR)
            Vprintf("%4.1f %6d %12.4e\n", x, nu, y);
        else
        {
            Vprintf("Error from s18ec.\n", fail.message);
            exit_status = 1;
            goto END;
        }
    }
END:
    return exit_status;
} /* main */

8.2 Program Data

s18ec Example Program Data
  3.9  -3
  1.4  -2
  8.2  -1
  6.7   1
  0.5   2
  2.3   3 : Values of x and nu

8.3 Program Results

s18ec Example Program Results

<table>
<thead>
<tr>
<th>x</th>
<th>nu</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>-3</td>
<td>9.5207e+00</td>
</tr>
<tr>
<td>1.4</td>
<td>-2</td>
<td>1.4504e+00</td>
</tr>
<tr>
<td>8.2</td>
<td>-1</td>
<td>5.1349e+02</td>
</tr>
<tr>
<td>6.7</td>
<td>1</td>
<td>1.2714e+02</td>
</tr>
<tr>
<td>0.5</td>
<td>2</td>
<td>5.8799e-01</td>
</tr>
<tr>
<td>2.3</td>
<td>3</td>
<td>2.3687e+00</td>
</tr>
</tbody>
</table>