NAG Library Function Document

nag_bessel_k_nu (s18efc)

1 Purpose

nag_bessel_k_nu (s18efc) returns the value of the modified Bessel function $K_{\nu/4}(x)$ for real $x > 0$.

2 Specification

```c
#include <nag.h>
#include <nags.h>
double nag_bessel_k_nu (double x, Integer nu, NagError *fail)
```

3 Description

nag_bessel_k_nu (s18efc) evaluates an approximation to the modified Bessel function of the second kind $K_{\nu/4}(x)$, where the order $\nu = -3, -2, -1, 1, 2$ or $3$ and $x$ is real and positive. For negative orders the formula

$$K_{-\nu/4}(x) = K_{\nu/4}(x)$$

is used.

4 References


5 Arguments

1: $x$ – double

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

   Constraint: $x > 0.0$.

2: $\nu$ – Integer

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

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

3: $\text{fail}$ – NagError

   The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

**NE_INT**

   On entry, $\nu = (\text{value})$.

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

**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 contact NAG for assistance.
NE_OVERFLOW_LIKELY
The evaluation has been abandoned due to the likelihood of overflow. The result is returned as zero.

NE_REAL
On entry, $x = \langle\text{value}\rangle$.
Constraint: $x > 0.0$.

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

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

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

7 Accuracy
All constants in the underlying function 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 function, the actual number of correct digits is limited, in general, by $p - s$, where $s \approx \max(1, \left|\log_{10} x\right|)$ 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.

8 Parallelism and Performance
Not applicable.

9 Further Comments
None.

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

10.1 Program Text
/* nag_bessel_k_nu (s18efc) Example Program. */
* Copyright 2014 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)
\begin{verbatim}
{
    Integer exit_status = 0, nu;
    NagError fail;
    double x, y;
    INIT_FAIL(fail);
    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n");
#else
    scanf("%*[\n");
#endif
    printf("nag_bessel_k_nu (s18efc) Example Program Results
");
    printf(" x nu y\n");
#ifdef _WIN32
    while (scanf_s("%lf %"NAG_IFMT"%*[\n]", &x, &nu) != EOF)
#else
    while (scanf("%lf %"NAG_IFMT"%*[\n]", &x, &nu) != EOF)
#endif
    {
        /* nag_bessel_k_nu (s18efc).
           * Modified Bessel function K_{nu/4}(x)
           */
        y = nag_bessel_k_nu(x, nu, &fail);
        if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_bessel_k_nu (s18efc).\n\n%s\n", fail.message);
            exit_status = 1;
            goto END;
        }
        printf("%4.1f %6"NAG_IFMT" %13.4e\n", x, nu, y);
    }
END:
    return exit_status;
}
\end{verbatim}

10.2 Program Data

\texttt{nag\_bessel\_k\_nu (s18efc) Example Program Data}
3.9 -3
1.4 -2
0.2 -1
6.7 1
0.5 2
2.3 3 : Values of x and nu

10.3 Program Results

\texttt{nag\_bessel\_k\_nu (s18efc) Example Program Results}
\begin{tabular}{ccc}
   x & nu & y \\
3.9 & -3 & 1.3315e-02 \\
1.4 & -2 & 2.6121e-01 \\
0.2 & -1 & 1.1892e-04 \\
6.7 & 1 & 5.8826e-04 \\
0.5 & 2 & 1.0750e+00 \\
2.3 & 3 & 8.7724e-02 \\
\end{tabular}