# NAG Library Function Document

# nag_bessel_k1_scaled (s18cdc)

## 1 Purpose

nag_bessel_k1_scaled (s18cdc) returns a value of the scaled modified Bessel function $e^x K_1(x)$.

## 2 Specification

```
#include <nag.h>
#include <nags.h>
double nag_bessel_k1_scaled (double x, NagError *fail)
```

## 3 Description

nag_bessel_k1_scaled (s18cdc) evaluates an approximation to $e^x K_1(x)$, where $K_1$ is a modified Bessel function of the second kind. The scaling factor $e^x$ removes most of the variation in $K_1(x)$.

The function uses the same Chebyshev expansions as nag_bessel_k1 (s18adc), which returns the unscaled value of $K_1(x)$.

## 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.

*Constraint*: **x** $> 0.0$.

2: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

## 6 Error Indicators and Warnings

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

**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.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

**NE_NO_LICENCE**

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

**NE_REAL_ARG_LE**

On entry, $\mathbf{x} = \langle value \rangle$.
Constraint: $\mathbf{x} > 0.0$.
$K_1$ is undefined and the function returns zero.

**NE_REAL_ARG_TOO_SMALL**

On entry, $\mathbf{x} = \langle value \rangle$.
Constraint: $\mathbf{x} > \langle value \rangle$.
The function returns the value of the function at the smallest permitted value of the argument.

## 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_k1_scaled (s18cdc) 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_k1_scaled (s18cdc) 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;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
```

```
  scanf("%*[^\n]");
#endif
  printf("nag_bessel_k1_scaled (s18cdc) 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_k1_scaled (s18cdc).
     * Scaled modified Bessel function exp(x) K_1(x)
     */
    y = nag_bessel_k1_scaled(x, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_bessel_k1_scaled (s18cdc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
    printf("%12.3e%12.3e\n", x, y);
  }

END:
  return exit_status;
}
```

## 10.2 Program Data

```
nag_bessel_k1_scaled (s18cdc) Example Program Data
             0.4
             0.6
             1.4
             2.5
             10.0
             1000.0
```

## 10.3 Program Results

```
nag_bessel_k1_scaled (s18cdc) Example Program Results
     x           y
   4.000e-01   3.259e+00
   6.000e-01   2.374e+00
   1.400e+00   1.301e+00
   2.500e+00   9.002e-01
   1.000e+01   4.108e-01
   1.000e+03   3.965e-02
```