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

This function 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.　Parameters

x

　　　　Input: the argument x of the function.
　　　　Constraint: $\mathbf{x} > 0.0$. If **x** is too close to zero, there is a danger of overflow, and a failure will occur.

fail

　　　　The NAG error parameter, see the Essential Introduction to the NAG C Library.

5.　Error Indications and Warnings

NE_REAL_ARG_LE

　　　　On entry, **x** must not be less than or equal to 0.0: $\mathbf{x} = \langle value \rangle$.

　　　　K_1 is undefined and the function returns zero.

NE_REAL_ARG_TOO_SMALL

　　　　On entry, **x** must be greater than $\langle value \rangle$: $\mathbf{x} = \langle value \rangle$.
　　　　The function returns the value of the function at the smallest permitted value of the argument.

6.　Further Comments

6.1.　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**.

6.2.　References

Abramowitz M and Stegun I A (1968) *Handbook of Mathematical Functions* Dover Publications, New York ch 9 p 374.

7.　See Also

　　nag_bessel_k1 (s18adc)
　　nag_bessel_k0_scaled (s18ccc)

8.　Example

The following program reads values of the argument x from a file, evaluates the function at each value of x and prints the results.

8.1. Program Text

```
/* nag_bessel_k1_scaled(s18cdc) Example Program
 *
 * Copyright 1991 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

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

main()
{
  double x, y;

  /* Skip heading in data file */
  Vscanf("%*[^\n]");
  Vprintf("s18cdc Example Program Results\n");
  Vprintf("      x             y\n");
  while (scanf("%lf", &x) != EOF)
    {
      y = s18cdc(x, NAGERR_DEFAULT);
      Vprintf("%12.3e%12.3e\n", x, y);
    }
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
s18cdc Example Program Data
         0.4
         0.6
         1.4
         2.5
        10.0
      1000.0
```

8.3. Program Results

```
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
```