NAG Library Function Document

nag_bessel_k0_scaled (s18ccc)

1 Purpose
nag_bessel_k0_scaled (s18ccc) returns a value of the scaled modified Bessel function $e^x K_0(x)$.

2 Specification
#include <nag.h>
#include <nags.h>
double nag_bessel_k0_scaled (double x, NagError *fail)

3 Description
nag_bessel_k0_scaled (s18ccc) evaluates an approximation to $e^x K_0(x)$, where $K_0$ is a modified Bessel function of the second kind. The scaling factor $e^x$ removes most of the variation in $K_0(x)$.

The function uses the same Chebyshev expansions as nag_bessel_k0 (s18acc), which returns the unscaled value of $K_0(x)$.

4 References

5 Arguments
1: x – double

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

Constraint: $x > 0.0$.

2: fail – NagError *

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

6 Error Indicators and Warnings

NE_ALLOC_FAIL
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction 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 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.
On entry, \(x = \langle\text{value}\rangle\).
Constraint: \(x > 0.0\).

### 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

Not applicable.

### 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

```c
#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);

    printf("nag_bessel_k0_scaled (s18ccc) 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_k0_scaled (s18ccc).
         * Scaled modified Bessel function exp(x) K_0(x)
         *\*/
        y = nag_bessel_k0_scaled(x, &fail);
        if (fail.code != NE_NOERROR)
            s18ccc
    }
}
```

NAG Library Manual
s18ccc.2
Mark 25
```c
{  printf("Error from nag_bessel_k0_scaled (sl8ccc).\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_k0_scaled (sl8ccc) Example Program Data
  0.4
  0.6
  1.4
  2.5
  10.0
  1000.0

10.3 Program Results

nag_bessel_k0_scaled (sl8ccc) Example Program Results

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.000e-01</td>
<td>1.663e+00</td>
</tr>
<tr>
<td>6.000e-01</td>
<td>1.417e+00</td>
</tr>
<tr>
<td>1.400e+00</td>
<td>9.881e-01</td>
</tr>
<tr>
<td>2.500e+00</td>
<td>7.595e-01</td>
</tr>
<tr>
<td>1.000e+01</td>
<td>3.916e-01</td>
</tr>
<tr>
<td>1.000e+03</td>
<td>3.963e-02</td>
</tr>
</tbody>
</table>
```