

NAG Library Function Document

nag_bessel_i1_scaled_vector (s18ctc)

1 Purpose

nag_bessel_i1_scaled_vector (s18ctc) returns an array of values of the scaled modified Bessel function $e^{-|x|}I_1(x)$.

2 Specification

```
#include <nag.h>
#include <nags.h>

void nag_bessel_i1_scaled_vector (Integer n, const double x[], double f[],
    NagError *fail)
```

3 Description

nag_bessel_i1_scaled_vector (s18ctc) evaluates an approximation to $e^{-|x_i|}I_1(x_i)$, where I_1 is a modified Bessel function of the first kind for an array of arguments x_i , for $i = 1, 2, \dots, n$. The scaling factor $e^{-|x|}$ removes most of the variation in $I_1(x)$.

The function uses the same Chebyshev expansions as nag_bessel_i1_vector (s18atc), which returns an array of the unscaled values of $I_1(x)$.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

5 Arguments

- | | | |
|----|---|---------------------|
| 1: | n – Integer | <i>Input</i> |
| | <i>On entry:</i> n , the number of points. | |
| | <i>Constraint:</i> $n \geq 0$. | |
| 2: | x[n] – const double | <i>Input</i> |
| | <i>On entry:</i> the argument x_i of the function, for $i = 1, 2, \dots, n$. | |
| 3: | f[n] – double | <i>Output</i> |
| | <i>On exit:</i> $e^{- x_i }I_1(x_i)$, the function values. | |
| 4: | fail – NagError * | <i>Input/Output</i> |
| | The NAG error argument (see Section 3.6 in the Essential Introduction). | |

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, $n = \langle value \rangle$.
 Constraint: $n \geq 0$.

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.

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 x from a file, evaluates the function at each value of x_i and prints the results.

10.1 Program Text

```

/* nag_bessel_il_scaled_vector (s18ctc) Example Program.
 *
 * Copyright 2011, Numerical Algorithms Group.
 *
 * Mark 23 2011.
 */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
  Integer  exit_status = 0;
  Integer  i, n;
  double   *f = 0, *x = 0;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
  scanf("%*[\n]");

  printf("nag_bessel_il_scaled_vector (s18ctc) Example Program Results\n");
  printf("\n");
  printf("      x          f\n");
  printf("\n");
  scanf("%ld", &n);
  scanf("%*[\n]");

  /* Allocate memory */
  if (!(x = NAG_ALLOC(n, double)) ||
      !(f = NAG_ALLOC(n, double)))

```

```

    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    for (i=0; i<n; i++)
        scanf("%lf", &x[i]);
    scanf("%*[^\\n]");

    /* nag_bessel_il_scaled_vector (s18ctc).
     * scaled modified Bessel Function exp(-|x|) I1(x)
     */
    nag_bessel_il_scaled_vector(n, x, f, &fail);
    if (fail.code!=NE_NOERROR)
    {
        printf("Error from nag_bessel_il_scaled_vector (s18ctc).\n%s\n",
            fail.message);
        exit_status = 1;
        goto END;
    }

    for (i=0; i<n; i++)
        printf(" %11.3e %11.3e\n", x[i], f[i]);

    END:
    NAG_FREE(f);
    NAG_FREE(x);

    return exit_status;
}

```

10.2 Program Data

nag_bessel_il_scaled_vector (s18ctc) Example Program Data

8

0.0 0.5 1.0 3.0 6.0 10.0 1000.0 -1.0

10.3 Program Results

nag_bessel_il_scaled_vector (s18ctc) Example Program Results

x	f
0.000e+00	0.000e+00
5.000e-01	1.564e-01
1.000e+00	2.079e-01
3.000e+00	1.968e-01
6.000e+00	1.521e-01
1.000e+01	1.213e-01
1.000e+03	1.261e-02
-1.000e+00	-2.079e-01
