NAG Library Function Document

nag_incomplete_beta (s14ccc)

1 Purpose

nag_incomplete_beta (s14ccc) computes values for the incomplete beta function $I_x(a,b)$ and its complement $1 - I_x(a,b)$.

2 Specification

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

```
void nag_incomplete_beta (double a, double b, double x, double *w,
     double *w1, NagError *fail)
```

3 Description

nag_incomplete_beta (s14ccc) evaluates the incomplete beta function and its complement in the normalized form

$$\begin{aligned} I_x(a,b) &= \tfrac{1}{B(a,b)}\int_0^x t^{a-1}(1-t)^{b-1}dt \\ 1 - I_x(a,b) &= I_y(b,a), \text{ where } y = 1 - x, \end{aligned}$$

with

$0 \leq x \leq 1$,

$a \geq 0$ and $b \geq 0$,

and the beta function $B(a,b)$ is defined as $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ where $\Gamma(y)$ is the gamma function.

Several methods are used to evaluate the functions depending on the parameters a, b and x. The methods include Wise's asymptotic expansion (see Wise (1950)) when $a > b$, continued fraction derived by DiDonato and Morris (1992) when a, $b > 1$, and power series when $b \leq 1$ or $b \times x \leq 0.7$. When both a and b are large, specifically a, $b \geq 15$, the DiDonato and Morris (1992) asymptotic expansion is employed for greater efficiency.

Once either $I_x(a,b)$ or $I_y(b,a)$ is computed, the other is obtained by subtraction from 1. In order to avoid loss of relative precision in this subtraction, the smaller of $I_x(a,b)$ and $I_y(b,a)$ is computed first.

nag_incomplete_beta (s14ccc) is derived from BRATIO in DiDonato and Morris (1992).

4 References

DiDonato A R and Morris A H (1992) Algorithm 708: Significant digit computation of the incomplete beta function ratios *ACM Trans. Math. Software* **18** 360–373

Wise M E (1950) The incomplete beta function as a contour integral and a quickly converging series for its inverse *Biometrika* **37** 208–218

5 Arguments

1: **a** – double *Input*

On entry: the parameter a of the function.

Constraint: $\mathbf{a} \geq 0.0$.

2: **b** – double *Input*

On entry: the parameter b of the function.

Constraints:

$\mathbf{b} \geq 0.0$;
either $\mathbf{b} \neq 0.0$ or $\mathbf{a} \neq 0.0$.

3: **x** – double *Input*

On entry: x, upper limit of integration.

Constraints:

$0.0 \leq \mathbf{x} \leq 1.0$;
either $\mathbf{x} \neq 0.0$ or $\mathbf{a} \neq 0.0$;
either $1 - \mathbf{x} \neq 0.0$ or $\mathbf{b} \neq 0.0$.

4: **w** – double * *Output*

On exit: the value of the incomplete beta function $I_x(a,b)$ evaluated from zero to x.

5: **w1** – double * *Output*

On exit: the value of the complement of the incomplete beta function $1 - I_x(a,b)$, i.e., the incomplete beta function evaluated from x to one.

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

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

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_BAD_PARAM

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

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_REAL

On entry, $\mathbf{a} = \langle$*value*$\rangle$.
Constraint: $\mathbf{a} \geq 0.0$.

On entry, $\mathbf{b} = \langle$*value*$\rangle$.
Constraint: $\mathbf{b} \geq 0.0$.

On entry, $\mathbf{x} = \langle$*value*$\rangle$.
Constraint: $0.0 \leq \mathbf{x} \leq 1.0$.

NE_REAL_2

On entry, 1.0 − **x** and **b** were zero.
Constraint: 1.0 − **x** or **b** must be nonzero.

On entry, **a** and **b** were zero.
Constraint: **a** or **b** must be nonzero.

On entry, **x** and **a** were zero.
Constraint: **x** or **a** must be nonzero.

7 Accuracy

nag_incomplete_beta (s14ccc) is designed to maintain relative accuracy for all arguments. For very tiny results (of the order of ***machine precision*** or less) some relative accuracy may be lost – loss of three or four decimal places has been observed in experiments. For other arguments full relative accuracy may be expected.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example reads values of the arguments a and b from a file, evaluates the function and its complement for 10 different values of x and prints the results.

10.1 Program Text

```
/* nag_incomplete_beta (s14ccc) Example Program.
 *
 * Copyright 2011 Numerical Algorithms Group.
 */

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

int main(void)
{
  Integer  exit_status = 0;
  double   a, b, x, w, w1;
  NagError fail;

  INIT_FAIL(fail);

  printf("nag_incomplete_beta (s14ccc) Example Program Results\n");

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

  printf("%4s%7s%7s%13s%13s\n","a","b","x","Ix(a,b)","1-Ix(a,b)");
  while (scanf("%lf %lf %lf", &a, &b, &x) != EOF)
    {
      /* nag_incomplete_beta (s14ccc).
       * Incomplete beta function Ix(a,b) and its complement 1-Ix(a,b)
       */
      nag_incomplete_beta(a, b, x, &w, &w1, &fail);
      if (fail.code != NE_NOERROR)
        {
```

```
          printf("Error from nag_incomplete_beta (s14ccc).\n%s\n",
                 fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%6.2f%7.2f%7.2f%12.4e%12.4e\n", a, b, x, w, w1);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_incomplete_beta (s14ccc) Example Program Data
 5.3  10.1   0.01
 1.0   0.2   0.02
 0.2   1.0   0.04
 1.2   5.1   0.05
 7.0   2.5   0.08
 1.0   1.0   0.09
10.0   1.5   0.10
```

10.3 Program Results

```
nag_incomplete_beta (s14ccc) Example Program Results
   a      b      x      Ix(a,b)    1-Ix(a,b)
  5.30  10.10   0.01  6.4755e-08  1.0000e+00
  1.00   0.20   0.02  4.0324e-03  9.9597e-01
  0.20   1.00   0.04  5.2531e-01  4.7469e-01
  1.20   5.10   0.05  1.6101e-01  8.3899e-01
  7.00   2.50   0.08  3.3490e-07  1.0000e+00
  1.00   1.00   0.09  9.0000e-02  9.1000e-01
 10.00   1.50   0.10  3.5279e-10  1.0000e+00
```