NAG C Library Function Document

nag_rngs_logistic (g05lnc)

1 Purpose

nag_rngs_logistic (g05lnc) generates a vector of pseudo-random numbers from a logistic distribution with mean $a$ and spread $b$.

2 Specification

```c
void nag_rngs_logistic (double a, double b, Integer n, double x[], Integer igen,
                       Integer iseed[], NagError *fail)
```

3 Description

The distribution has PDF (probability density function)

$$f(x) = \frac{e^{(x-a)/b}}{b(1 + e^{(x-a)/b})^2}.$$  

nag_rngs_logistic (g05lnc) returns the value

$$a + b \ln \left( \frac{y}{1 - y} \right),$$

where $y$ is a pseudo-random number uniformly distributed over $(0,1)$.

One of the initialisation functions nag_rngs_init_repeatable (g05kbc) (for a repeatable sequence if computed sequentially) or nag_rngs_init_nonrepeatable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag_rngs_logistic (g05lnc).

4 References


5 Parameters

1:  $a$ – double
   *Input*
   *On entry:* the mean, $a$, of the distribution.

2:  $b$ – double
   *Input*
   *On entry:* the spread, $b$, of the distribution, where ‘spread’ is $\frac{\sqrt{3}}{\pi} \times$ standard deviation.
   *Constraint:* $b > 0.0$.

3:  $n$ – Integer
   *Input*
   *On entry:* the number, $n$, of pseudo-random numbers to be generated.
   *Constraint:* $n \geq 0$.

4:  $x[\text{dim}]$ – double
   *Output*
   *Note:* the dimension, $\text{dim}$, of the array $x$ must be at least $\max(1, n)$.
   *On exit:* the $n$ pseudo-random numbers from the specified logistic distribution.
5: `igen` – Integer  
   *Input*
   
   On entry: must contain the identification number for the generator to be used to return a pseudo-
   random number and should remain unchanged following initialisation by a prior call to one of the
   functions `nag_rngs_init_repeatable` (g05kbc) or `nag_rngs_init_nonrepeatable` (g05kcc).

   *Input/Output*
   
   On entry: contains values which define the current state of the selected generator.
   
   On exit: contains updated values defining the new state of the selected generator.

7: `fail` – NagError *  
   *Input/Output*
   
   The NAG error parameter (see the Essential Introduction).

6  Error Indicators and Warnings

**NE_INT**

   On entry, `n = (value)`.
   
   Constraint: `n ≥ 0`.

**NE_REAL**

   On entry, `b = (value)`.
   
   Constraint: `b > 0.0`.

**NE_BAD_PARAM**

   On entry, parameter `h` 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 consult NAG for assistance.

7  Accuracy

Not applicable.

8  Further Comments

None.

9  Example

The example program prints the first five pseudo-random real numbers from a logistic distribution with
mean 1.0 and spread 2.0, generated by a single call to `nag_rngs_logistic` (g05lnc), after initialisation by
`nag_rngs_init_repeatable` (g05kbc).

9.1  Program Text

```c
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
```

/* nag_rngs_logistic(g05lnc) Example Program.  
 * Copyright 2001 Numerical Algorithms Group.  
 */
```c
#include <nagg05.h>

int main(void)
{
    /* Scalars */
    Integer igen, j, m;
    Integer exit_status=0;
    NagError fail;

    /* Arrays */
    double *x=0;
    Integer iseed[4];

    INIT_FAIL(fail);
    Vprintf("g05lnc Example Program Results\n\n");
    m=5;
    /* Allocate memory */
    if ( !(x = NAG_ALLOC(m, double)) )
    {
        Vprintf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    /* Initialise the seed to a repeatable sequence */
    iseed[0] = 1762543;
    iseed[1] = 9324783;
    iseed[2] = 42344;
    iseed[3] = 742355;
    /* igen identifies the stream. */
    igen = 1;
    g05kbc(&igen, iseed);
    g05lnc(1.0, 2.0, m, x, igen, iseed, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g05lnc.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }

    for (j = 0; j < m; ++j)
    {
        Vprintf("%10.4f\n", x[j]);
    }

    END:
    if (x) NAG_FREE(x);
    return exit_status;
}
```

9.2 Program Data
None.

9.3 Program Results

```
g05lnc Example Program Results

-3.6439
6.9299
0.2422
3.1253
6.8790
```