NAG C Library Function Document

nag_rngs_logarithmic (g05mdc)

1 Purpose

nag_rngs_logarithmic (g05mdc) generates a vector of pseudo-random integers from the discrete logarithmic distribution with parameter a.

2 Specification

```
void nag_rngs_logarithmic (Integer mode, double a, Integer n, Integer x[],
    Integer igen, Integer iseed[], double r[], NagError *fail)
```

3 Description

nag_rngs_logarithmic (g05mdc) generates n integers x_i from a discrete logarithmic distribution, where the probability of $x_i = I$ is

$$P(x_i = I) = -\frac{(a^I)}{(I \times \log(1-a))} \quad I = 1, 2, \ldots,$$

where $0 < a < 1$.

The variates can be generated with or without using a search table and index. If a search table is used then it is stored with the index in a reference vector and subsequent calls to nag_rngs_logarithmic (g05mdc) with the same parameter value can then use this reference vector to generate further variates.

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_logarithmic (g05mdc).

4 References

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

5 Parameters

1: **mode** – Integer   *Input*

On entry: a code for selecting the operation to be performed by the function:

mode = 0

Set up reference vector only.

mode = 1

Generate variates using reference vector set up in a prior call to nag_rngs_logarithmic (g05mdc).

mode = 2

Set up reference vector and generate variates.

mode = 3

Generate variates without using the reference vector.

Constraint: $0 \le$ **mode** ≤ 3.

2:    **a** – double   *Input*

   On entry: the parameter a of the logarithmic distribution.

   Constraint: $0.0 < \mathbf{a} < 1.0$.

3:    **n** – Integer   *Input*

   On entry: the number, n, of pseudo-random numbers to be generated.

   Constraint: $\mathbf{n} \geq 1$.

4:    **x**[**n**] – Integer   *Output*

   On exit: the n pseudo-random numbers from the specified logarithmic 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).

6:    **iseed**[4] – Integer   *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:    **r**[dim] – double   *Input/Output*

   Note: the dimension, dim, of the array **r** must be at least $10 + \frac{40}{1-\mathbf{a}}$ when **mode** < 3 and at least 1 otherwise.

   On exit: the reference vector.

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

   The NAG error parameter (see the Essential Introduction).

6   Error Indicators and Warnings

NE_INT

   On entry, **mode** = $\langle value \rangle$.
   Constraint: $0 \leq \mathbf{mode} \leq 3$.

   On entry, **n** = $\langle value \rangle$.
   Constraint: $\mathbf{n} \geq 1$.

NE_DIM_INFEASIBLE

   a is such that the reference vector length would exceed integer range. We recommend setting **mode** = 3. **a** = $\langle value \rangle$.

NE_PREV_CALL

   a is not the same as when **r** was set up in a previous call. Previous value of **a** = $\langle value \rangle$, **a** = $\langle value \rangle$.

NE_REAL

   On entry, $\mathbf{a} \leq 0.0$ or $\mathbf{a} \geq 1.0$: **a** = $\langle value \rangle$.

NE_BAD_PARAM

   On entry, parameter $\langle value \rangle$ 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 consult NAG for assistance.

7   Accuracy

Not applicable.

8   Further Comments

None.

9   Example

The example program prints five pseudo-random integers from a logarithmic distribution with parameter $a = 0.999$, generated by a single call to nag_rngs_logarithmic (g05mdc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1   Program Text

```
/* nag_rngs_logarithmic(g05mdc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>

int main(void)
{
  /* Scalars */
  double  a;
  Integer i, igen, n, nr;
  Integer exit_status=0;
  NagError fail;

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

  INIT_FAIL(fail);
  Vprintf("g05mdc Example Program Results\n\n");

  nr = 1;
  n = 10;
  /* Allocate memory */
  if ( !(r = NAG_ALLOC(nr, double)) ||
       !(x = NAG_ALLOC(n, Integer)) )
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  /* Set the distribution parameter A */
  a = 0.9999;
  /* 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);

   /* Generate integers and store in X */
   /* Use MODE=3 because A > 0.95 */
   g05mdc(3, a, n, x, igen, iseed, r, &fail);
   if (fail.code != NE_NOERROR)
      {
        Vprintf("Error from g05mdc.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
      }
   for (i = 0; i < n; ++i)
      {
        Vprintf("%12ld\n", x[i]);
      }
 END:
   if (r) NAG_FREE(r);
   if (x) NAG_FREE(x);
   return exit_status;
}
```

9.2 Program Data

None.

9.3 Program Results

```
g05mdc Example Program Results

        262
         21
       8546
        737
          1
          1
         16
        197
         53
          3
```