NAG C Library Function Document

nag_rngs_geom (g05mbc)

1 Purpose

nag_rngs_geom (g05mbc) generates a vector of pseudo-random integers from the discrete geometric distribution with probability p of success at a trial.

2 Specification

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

3 Description

nag_rngs_geom (g05mbc) generates n integers x_I from a discrete geometric distribution, where the probability of $x_i = I$ (a first success after I trials) is

$$P(x_i = I) = p \times [(1-p)^{(I-1)}], \quad I = 1, 2, \ldots .$$

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_geom (g05mbc) with the same parameter value can then use this reference vector to generate further variates. If the search table is not used (as recommended for small values of p) then a direct transformation of uniform variates is used.

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_geom (g05mbc).

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_geom (g05mbc).

mode $= 2$

Set up reference vector and generate variates.

mode $= 3$

Generate variates without using the reference vector;

Constraint: $0 \leq$ **mode** ≤ 3.

2: **p** – double *Input*

On entry: the parameter p of the geometric distribution representing the probability of success at a single trial.

Constraint: ***machine precision*** $\leq$ **p** ≤ 1.0.

3: **n** – Integer *Input*

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

Constraint: **n** ≥ 1.

4: **x**[**n**] – Integer *Output*

On exit: the n pseudo-random numbers from the specified geometric 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 $6 + 42/\mathbf{p}$ 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** = ⟨*value*⟩.
Constraint: $0 \leq$ **mode** ≤ 3.

On entry, **n** = ⟨*value*⟩.
Constraint: **n** ≥ 1.

NE_DIM_INFEASIBLE

p is so small that the reference vector length would exceed integer range We recommend setting **mode** = 3. **p** = ⟨*value*⟩.

NE_PREV_CALL

p is not the same as when **r** was set up in a previous call. Previous value = ⟨*value*⟩, **p** = ⟨*value*⟩.

NE_REAL

On entry, **p** < ***machine precision*** or **p** > 1.0: **p** = ⟨*value*⟩.

NE_BAD_PARAM

On entry, parameter ⟨*value*⟩ 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

The time taken to set up the reference vector, if used, increases with the length of array **r**. However, if the reference vector is used, the time taken to generate numbers decreases as the space allotted to the index part of **r** increases. There is a point, depending on the distribution, where this improvement becomes very small and the recommended values for the length of array **r** in other functions are designed to approximate this point.

If **p** is very small then the storage requirements for the reference vector and the time taken to set up the reference vector becomes prohibitive. In this case it is recommended that the reference vector is not used. This is achieved by selecting **mode** = 3.

9 Example

The example program prints five pseudo-random integers from a geometric distribution with parameter $p = 0.001$, generated by a single call to nag_rngs_geom (g05mbc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

```
/* nag_rngs_geom(g05mbc) 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   p;
  Integer  igen, j, n, nr;
  Integer  exit_status=0;
  NagError fail;

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

  INIT_FAIL(fail);
  Vprintf("g05mbc Example Program Results\n\n");

  nr = 40000;
  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 P */
  p = 0.001;
  /* Initialise the seed to a repeatable sequence */
```

```
  iseed[0] = 1762543;
  iseed[1] = 9324783;
  iseed[2] = 423441;
  iseed[3] = 742355;
  /* igen identifies the stream. */
  igen = 1;
  g05kbc(&igen, iseed);

  /* Choose MODE = 3 because P is close to 0 */
  g05mbc(3, p, n, x, igen, iseed, r, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05mbc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  for (j = 0; j < n; ++j)
    {
      Vprintf("%12ld\n", x[j]);
    }
 END:
  if (r) NAG_FREE(r);
  if (x) NAG_FREE(x);
  return exit_status;
}
```

9.2 Program Data

None.

9.3 Program Results

```
g05mbc Example Program Results

        3699
         631
          29
        1228
         481
         341
        1394
         915
         332
         487
```