NAG C Library Function Document

nag_rngs_poisson (g05mkc)

1 Purpose

nag_rngs_poisson (g05mkc) generates a vector of pseudo-random integers from the discrete Poisson distribution with mean $\lambda$.

2 Specification

```c
void nag_rngs_poisson (Integer mode, double lambda, Integer n, Integer x[],
                      Integer igen, Integer iseed[], double r[], NagError *fail)
```

3 Description

nag_rngs_poisson (g05mkc) generates $n$ integers $x_i$ from a discrete Poisson distribution with mean $\lambda$, where the probability of $x_i = I$ is

$$P(x_i = I) = \frac{\lambda^I \times e^{-\lambda}}{I!}, \quad I = 0, 1, \ldots,$$

where $0 \leq \lambda$.

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_poisson (g05mkc) with the same parameter values can then use this reference vector to generate further variates. The reference array is found using a recurrence relation if $\lambda$ is less than 50 and by Stirling’s formula otherwise.

One of the initialisation functions nag_rngs_init_repeateable (g05kbc) (for a repeatable sequence if computed sequentially) or nag_rngs_init_nonrepeateable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag_rngs_poisson (g05mkc).

4 References


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_poisson (g05mkc).

- mode = 2
  - Set up reference vector and generate variates.

- mode = 3
  - Generate variates without using the reference vector.

Constraint: $0 \leq \text{mode} \leq 3$. 

[NP3645/7] g05mkc.1
2: lambda – double
   Input
   On entry: the mean \( \lambda \) of the Poisson distribution.
   Constraint: \( \lambda \geq 0.0 \).

3: n – Integer
   Input
   On entry: the number, \( n \), of pseudo-random numbers to be generated.
   Constraint: \( n \geq 1 \).

4: x[n] – Integer
   Output
   On exit: the \( n \) pseudo-random numbers from the specified Poisson 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: r[dim] – double
   Input/Output
   Note: the dimension, \( dim \), of the array \( r \) must be at least \( 22 + 20 \times \sqrt{\lambda} \) when \( \text{mode} < 3 \) 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, \( \text{mode} = (\text{value}) \).
   Constraint: \( 0 \leq \text{mode} \leq 3 \).
   On entry, \( n = (\text{value}) \).
   Constraint: \( n \geq 1 \).

NE_DIM_INFEASIBLE
   lambda is so large that the reference vector length would exceed integer range. We recommend
   setting \( \text{mode} = 3 \). \( \lambda = (\text{value}) \).

NE_PREV_CALL
   lambda has changed since \( r \) was set up in a previous call. Previous value of \( \lambda = (\text{value}) \),
   \( \lambda = (\text{value}) \).

NE_REAL
   On entry, \( \lambda = (\text{value}) \).
   Constraint: \( \lambda \geq 0.0 \).
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
None.

9 Example
The example program prints 10 pseudo-random integers from a Poisson distribution with mean $\lambda = 20$, generated by a single call to nag_rngs_poisson (g05mkc), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text
/* nag_rngs_poisson(g05mkc) 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 lambda;
    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("g05mkc Example Program Results\n\n");
    nr = 120;
    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 LAMBDA */
    lambda = 20.0;
/* Initialise the seed to a repeatable sequence */
iseed[0] = 1762543;
iseed[1] = 9324763;
iseed[2] = 42344;
iseed[3] = 742355;
/* igen identifies the stream. */
igen = 1;
g05kbc(&igen, iseed);

/* Generate reference vector R */
g05mkc(0, lambda, n, x, igen, iseed, r, &fail);
if (fail.code != NE_NOERROR)
    { Vprintf("Error from g05mkc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
/* Generate integers and store in X */
g05mkc(1, lambda, n, x, igen, iseed, r, &fail);
if (fail.code != NE_NOERROR)
    { Vprintf("Error from g05mkc.\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

<table>
<thead>
<tr>
<th>g05mkc Example Program Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>26</td>
</tr>
</tbody>
</table>