NAG C Library Function Document

nag_rngs_normal (g05lac)

1   Purpose

nag_rngs_normal (g05lac) generates a vector of pseudo-random numbers taken from a Normal (Gaussian) distribution with mean μ and variance σ^2.

2   Specification

```
void nag_rngs_normal (double xmu, double var, Integer n, double x[], Integer igen,
    Integer iseed[], NagError *fail)
```

3   Description

The distribution has PDF (probability distribution function)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

nag_rngs_normal (g05lac) uses the Box–Muller method.

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_normal (g05lac).

4   References

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

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5   Parameters

1:    **xmu** – double   *Input*

   On entry: the mean, μ, of the distribution.

2:    **var** – double   *Input*

   On entry: the variance, σ^2, of the distribution.

   Constraint: **var** ≥ 0.0.

3:    **n** – Integer   *Input*

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

   Constraint: **n** ≥ 0.

4:    **x**$[dim]$ – double   *Output*

   Note: the dimension, dim, of the array **x** must be at least $\max(1, \mathbf{n})$.

   On exit: the n pseudo-random numbers from the specified Normal 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:   **fail** – NagError *   *Input/Output*

The NAG error parameter (see the Essential Introduction).

6   Error Indicators and Warnings

NE_INT

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

NE_REAL

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

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

The generated numbers conform to a Normal distribution with an accuracy of $\sqrt{\textbf{\textit{machine precision}}}$.

8   Further Comments

None.

9   Example

The example program prints 5 pseudo-random numbers from a Normal distribution with mean 1.0 and variance 1.5, generated by a single call to nag_rngs_normal (g05lac), after initialisation by nag_rngs_init_repeatable (g05kbc).

9.1   Program Text

```
/* nag_rngs_normal(g05lac) 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 */
  Integer i, igen, n;
  Integer exit_status=0;
  NagError fail;

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

  INIT_FAIL(fail);
  Vprintf("g05lac Example Program Results\n\n");

  n = 5;
  /* Allocate memory */
  if ( !(x = NAG_ALLOC(n, 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);
  g05lac(1.0, 1.5, n, x, igen, iseed, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05lac.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  for (i = 1; i <= n; ++i)
    {
      Vprintf("%10.4f\n",   x[i - 1]);
    }
 END:
  if (x) NAG_FREE(x);
  return exit_status;
}
```

9.2   Program Data

None.

9.3   Program Results

```
g05lac Example Program Results

    3.5652
    0.1837
    0.9297
   -0.6421
    0.6371
```