NAG C Library Function Document

nag_rngs_triangular (g05lhc)

1 Purpose

nag_rngs_triangular (g05lhc) generates a vector of pseudo-random numbers from a triangular distribution with arguments $x_{\min}$, $x_{\max}$ and x_{med}.

2 Specification

```
#include <nag.h>
#include <nagg05.h>
void nag_rngs_triangular (double xmin, double xmax, double xmed, Integer n,
     double x[], Integer igen, Integer iseed[], NagError *fail)
```

3 Description

The triangular distribution has a PDF (probability density function) that is triangular in profile. The base of the triangle ranges from $x = x_{\min}$ to $x = x_{\max}$ and the PDF has a maximum value of $\dfrac{2}{x_{\max} - x_{\min}}$ at $x = x_{\text{med}}$. If $x_{\min} = x_{\text{med}} = x_{\max}$ then $x = x_{\text{med}}$ with probability 1; otherwise the triangular distribution has PDF:

$$f(x) = \frac{x - x_{\min}}{x_{\text{med}} - x_{\min}} \times \frac{2}{x_{\max} - x_{\min}} \quad \text{if } x_{\min} < x \leq x_{\text{med}},$$

$$f(x) = \frac{x_{\max} - x}{x_{\max} - x_{\text{med}}} \times \frac{2}{x_{\max} - x_{\min}} \quad \text{if } x_{\text{med}} < x \leq x_{\max},$$

$$f(x) = 0 \quad \text{otherwise.}$$

One of the initialization 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_triangular (g05lhc).

4 References

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

5 Arguments

1: **xmin** – double *Input*

2: **xmax** – double *Input*

On entry: the end points $x_{\min}$ and $x_{\max}$ of the uniform distribution.

Constraint: **xmin** $\leq$ **xmax**.

3: **xmed** – double *Input*

On entry: the median of the distribution x_{med} (also the location of the vertex of the triangular distribution at which the PDF reaches a maximum).

Constraint: **xmin** $\leq$ **xmed** $\leq$ **xmax**.

4: **n** – Integer *Input*

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

Constraint: $\mathbf{n} \geq 0$.

5: **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 triangular distribution.

6: **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 initialization by a prior call to one of the functions nag_rngs_init_repeatable (g05kbc) or nag_rngs_init_nonrepeatable (g05kcc).

7: **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.

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

The NAG error argument (see Section 2.6 of the Essential Introduction).

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

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

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.

NE_REAL_2

On entry, $\mathbf{xmed} = \langle value \rangle$, $\mathbf{xmax} = \langle value \rangle$.
Constraint: $\mathbf{xmed} \leq \mathbf{xmax}$.

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

On entry, $\mathbf{xmin} = \langle value \rangle$, $\mathbf{xmax} = \langle value \rangle$.
Constraint: $\mathbf{xmin} \leq \mathbf{xmax}$.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

The example program prints five pseudo-random numbers from a triangular distribution with arguments $x_{\min} = -1.0$, $x_{\max} = 1.0$ and $x_{\text{med}} = 0.5$, generated by a single call to nag_rngs_triangular (g05lhc), after initialization by nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

```
/* nag_rngs_triangular (g05lhc) 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  igen, j, m;
  Integer  exit_status=0;
  NagError fail;

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

  INIT_FAIL(fail);
  Vprintf("nag_rngs_triangular (g05lhc) 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;
  /* nag_rngs_init_repeatable (g05kbc).
   * Initialize seeds of a given generator for random number
   * generating functions (that pass seeds explicitly) to give
   * a repeatable sequence
   */
  nag_rngs_init_repeatable(&igen, iseed);

  /* nag_rngs_triangular (g05lhc).
   * Generates a vector of random numbers from a triangular
   * distribution, seeds and generator number passed
   * explicitly
   */
  nag_rngs_triangular(-1.0, 1.0, 0.5, m, x, igen, iseed, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from nag_rngs_triangular (g05lhc).\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

```
nag_rngs_triangular (g05lhc) Example Program Results

   -0.4823
    0.7786
    0.1042
    0.4932
    0.7759
```