NAG C Library Function Document

nag_rngs_uniform (g05lgc)

1 Purpose

nag_rngs_uniform (g05lgc) generates a vector of pseudo-random numbers uniformly distributed over the interval \([a, b]\).

2 Specification

```c
void nag_rngs_uniform (double a, double b, Integer n, double x[], Integer igen,
                      Integer iseed[], NagError *fail)
```

3 Description

If \(a = 0\) and \(b = 1\), nag_rngs_uniform (g05lgc) returns the next \(n\) values \(y_i\) from a uniform \((0,1)\) generator (see nag_rngs_basic (g05kac) for details).

For other values of \(a\) and \(b\), nag_rngs_uniform (g05lgc) applies the transformation

\[ x_i = a + (b - a) y_i, \]

The function ensures that the values \(x_i\) lie in the closed interval \([a, b]\).

If computing sequentially and using the same generator, nag_rngs_uniform (g05lgc) always generates exactly the same pseudo-random numbers as would \(n\) consecutive calls of nag_rngs_basic (g05kac) and on many machines is likely to be much faster.

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_uniform (g05lgc).

4 References


5 Parameters

1: \(a\) – double

Input

On entry: the end-points \(a\) and \(b\) of the uniform distribution.

Constraint: \(a \leq b\).

2: \(b\) – double

Input

On entry: the end-points \(a\) and \(b\) of the uniform distribution.

Constraint: \(a \leq b\).

3: \(n\) – Integer

Input

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

Constraint: \(n \geq 0\).

4: \(x[\text{dim}]\) – double

Output

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

On exit: the \(n\) pseudo-random numbers from the specified uniform distribution.
igen – Integer

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).


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.

fail – NagError *

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, n = ⟨value⟩.
Constraint: n ≥ 0.

NE_REAL_2

On entry, a = ⟨value⟩, b = ⟨value⟩.
Constraint: b ≥ a.

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 five pseudo-random numbers from a uniform distribution between −1.0 and
1.0, generated by a single call to nag_rngs_uniform (g05lgc), after initialisation by
nag_rngs_init_repeatable (g05kbc).

9.1 Program Text

/* nag_rngs_uniform(g05lgc) 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("g05lgc 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;
    g05kbc(&igen, iseed);
    g05lgc(-1.0, 1.0, m, x, igen, iseed, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g05lgc.\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

g05lgc Example Program Results

-0.8214
0.9019
-0.1872
0.4864
0.8995