nag_random_continuous_uniform (g05cac)

1. Purpose

nag_random_continuous_uniform (g05cac) returns a pseudo-random number taken from a uniform distribution between 0 and 1.

2. Specification

```
#include <nag.h>
#include <nagg05.h>

double nag_random_continuous_uniform(void)
```

3. Description

This function returns the next pseudo-random number from the basic uniform (0,1) generator.

The basic generator uses a multiplicative congruential algorithm

$$b_{i+1} = 13^{13} \times b_i \bmod 2^{59}.$$

The integer b_{i+1} is divided by 2^{59} to yield a real value y, which is guaranteed to satisfy

$$0 < y < 1.$$

The value of b_i is saved internally in the code. The initial value b_0 is set by default to $123456789 \times (2^{32}+1)$, but the sequence may be re-initialized by a call to nag_random_init_repeatable (g05cbc) for a repeatable sequence, or nag_random_init_nonrepeatable (g05ccc) for a non-repeatable sequence.

4. Parameters

None.

5. Error Indications and Warnings

None.

6. Further Comments

The period of the basic generator is 2^{57}.

Its performance has been analysed by the Spectral Test (see Knuth 1981, Section 3.3.4), yielding the following results in the notation of Knuth.

n	ν_n	Upper bound for ν_n
2	3.44×10^8	4.08×10^8
3	4.29×10^5	5.88×10^5
4	1.72×10^4	2.32×10^4
5	1.92×10^3	3.33×10^3
6	593	939
7	198	380
8	108	197
9	67	120

The right-hand column gives an upper bound for the values of ν_n attainable by any multiplicative congruential generator working modulo 2^{59}.

An informal interpretation of the quantities ν_n is that consecutive n-tuples are statistically uncorrelated to an accuracy of $1/\nu_n$. This is a theoretical result; in practice the degree of randomness is usually much greater than the above figures might support. More details are given in Knuth (1981), and in the references cited therein.

Note that the achievable statistical independence drops rapidly as the number of dimensions increases. This is a property of all multiplicative congruential generators and is the reason why very long periods are needed even for samples of only a few random numbers.

6.1. Accuracy

Not applicable.

6.2. References

Knuth D E (1981) *The Art of Computer Programming (Vol 2)* (2nd Edn) Addison-Wesley.

7. See Also

nag_random_init_repeatable (g05cbc)
nag_random_init_nonrepeatable (g05ccc)

8. Example

The example program prints the first five pseudo-random numbers from a uniform distribution between 0 and 1, generated by nag_random_continuous_uniform after initialisation by nag_random_init_repeatable (g05cbc).

8.1. Program Text

```
/* nag_random_continuous_uniform(g05cac) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg05.h>

main()
{
  Integer i;
  Integer seed = 0;

  Vprintf("g05cac Example Program Results\n");
  g05cbc(seed);
  for (i=1; i<=5; i++)
    Vprintf("%10.4f\n",g05cac());
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

None.

8.3. Program Results

```
g05cac Example Program Results
    0.7951
    0.2257
    0.3713
    0.2250
    0.8787
```