Mohamed Boudiaf University, M'Sila

Faculty of Mathematics and Computer Science.

Department of Mathematics, 3rd Year Bachelor's Degree in Applied Mathematics, Semester 6

Module : Simulation and Software Practice

Module Manager :

Dr. BOUHADJAR El Mountasar Billah

Chapter 1 : Random and Pseudo-Random Numbers

Academic Year : 2025-2026

Table des matières

Chapitre 1

Random and Pseudo-random Numbers

1.1 Introduction

In the modern world, randomness plays an essential role in solving problems ranging from simple games of chance to complex scientific simulations. Whether it is for predicting financial market trends, modeling meteorological phenomena, or encrypting sensitive data, the ability to generate unpredictable numbers is a cornerstone of computing and analysis.

However, true randomness, originating from natural phenomena or physical processes, remains difficult to exploit consistently. To meet this challenge, computational methods have been developed to produce sequences that mimic chance. These pseudo-random numbers, although generated by deterministic algorithms, possess statistical properties close to those of truly random sequences.

1.2 Explanation : Random Number Generators

A random number generator (RNG) is a mathematical or physical construction used to produce numbers without a sequential pattern or correlation, appearing thus random. Random number generators fall into two main categories :

— **True Random Number Generators (TRNG)**, based on physical or natural phenomena (for example, dice throwing, thermal noise, or radioactivity).

— **Pseudo-Random Number Generators (PRNG)**, which rely on deterministic algorithms to produce sequences of numbers that simulate randomness.

Historically, methods for obtaining random numbers used physical processes, such as games of chance (dice, lottery wheels) or draws. These methods allowed the construction of random number tables used in calculations and simulations. However, the sequences generated in this manner were limited in size and variety, which made them unsuitable for modern applications requiring large quantities of random numbers.

Today, most needs for random numbers are met by PRNGs, which generate sequences of numbers through mathematical algorithms. Although they are deterministic, these generators offer statistical properties close to those of true random sequences, making them ideal for simulations, cryptography, and other applications. The simplicity and efficiency of PRNGs make them an indispensable tool in modern computer science.

1.3 Generation of Pseudo-random Numbers

The generation of pseudo-random numbers is an essential step in many applications, including simulations, cryptography, and probabilistic models. Unlike true random numbers, obtained from physical phenomena such as thermal or quantum noise, pseudo-random numbers are generated by mathematical algorithms.

A pseudo-random number generator (PRNG) operates deterministically. This means it produces a sequence of numbers from an initial value called a *seed*, following a fixed rule or a predefined algorithm. Although these numbers are produced systematically, their statistical properties make them similar to true random numbers. These generators must meet two important criteria :

— **Uniformity** : The generated numbers must be uniformly distributed over the interval $[0, 1]$ or within a specified range.

— **Apparent independence** : Successive values must not show any perceptible correlation, even though they originate from a deterministic process.

PRNGs are widely used due to their speed, their ability to ge-

nerate large sequences, and their reproducibility, which is useful for repeated tests or simulations. However, they are not suitable for applications requiring true security, such as cryptography, because their sequences can be predicted if the algorithm or the seed is known.

General recurrence algorithm

1. Choose x_0 in $M := \{0, 1, \ldots, m-1\}$.
2. Set $x_{n+1} = f(x_n)$, for $n > 0$ where f is an adequate mapping from M to M.

1.3.1 The Fibonacci Method

This method is based on a well-known additive sequence :

$$X_n = (X_{n-1} + X_{n-2}) \mod m$$

It belongs to the class of **additive congruential generators**. More generally, with an initial set of k numbers $X_0, X_1, \ldots, X_{k-1}$, the sequence is defined for $n \geq k$ by :

$$X_n = (X_{n-1} + X_{n-k}) \mod m$$

Advantages

— Simple method to implement.
— Can generate pseudo-random sequences in certain contexts.

Disadvantages

— **Memory required :** It is necessary to store the last k generated values, which can become costly for high values of k.
— **Correlations for low k :** When k is small, consecutive numbers in the sequence show strong dependence, which decreases their pseudo-random quality.

1.3.2 Simple Congruential Method

The congruential method is a technique used to generate a sequence of pseudo-random numbers. This method relies on a simple and repetitive mathematical formula, which produces numbers that

appear random, although they are generated in an entirely deterministic manner.

The general formula : The sequence is defined by :

$$x_{n+1} = (a \cdot x_n + c) \mod m$$

With :
— x_0 : the seed (the first number chosen to start the sequence).
— a : the multiplier.
— c : the increment.
— m : the modulus, a positive integer that determines the range of the results.

If $c > 0$, the generator is called mixed ; otherwise ($c = 0$), it is called a multiplicative generator.

Operation :

1. Start with an initial number x_0, called the seed.
2. Using the formula above, calculate x_1, then x_2, and so on.
3. The generated values x_n range between 0 and $m - 1$.

Exemple 1.3.1. *If we choose $x_0 = 5$, $a = 3$, $c = 7$, and $m = 10$, the sequence is calculated as follows :*

$$x_1 = (3 \cdot 5 + 7) \mod 10 = 22 \mod 10 = 2$$

$$x_2 = (3 \cdot 2 + 7) \mod 10 = 13 \mod 10 = 3$$

$$x_3 = (3 \cdot 3 + 7) \mod 10 = 16 \mod 10 = 6$$

Thus, the obtained sequence is : $5, 2, 3, 6, \ldots$

Important Properties :

1. Uniformity : By dividing each x_n by m, we obtain a sequence of pseudo-random numbers between 0 and 1.
2. Period "p" : the number of possible values in the sequence.
3. If $p = m$, it is called a maximal period generator.

Exemple 1.3.2. *Let*

$$a = 4,\ c = 1,\ m = 9, \text{ and the initial seed } x_0 = 3.$$

Let us calculate the generation steps :

$$\begin{aligned}
x_1 &= (4 \cdot 3 + 1) \mod 9 = 13 \mod 9 = 4,\\
x_2 &= (4 \cdot 4 + 1) \mod 9 = 17 \mod 9 = 8,\\
x_3 &= (4 \cdot 8 + 1) \mod 9 = 33 \mod 9 = 6,\\
x_4 &= (4 \cdot 6 + 1) \mod 9 = 25 \mod 9 = 7,\\
x_5 &= (4 \cdot 7 + 1) \mod 9 = 29 \mod 9 = 2,\\
x_6 &= (4 \cdot 2 + 1) \mod 9 = 9 \mod 9 = 0,\\
x_7 &= (4 \cdot 0 + 1) \mod 9 = 1,\\
x_8 &= (4 \cdot 1 + 1) \mod 9 = 5,\\
x_9 &= (4 \cdot 5 + 1) \mod 9 = 21 \mod 9 = 3.
\end{aligned}$$

The generated sequence is :

$$\{3, 4, 8, 6, 7, 2, 0, 1, 5, 3\}.$$

Notice that the sequence restarts from $x_9 = 3$, which means the period is 9. Thus, this method produces a pseudo-random sequence with a maximal period of $M = 9$.

Remarque 1. *The congruential method is simple to implement and fast, but it has limitations. For example, if the parameters a, c, and m are not chosen carefully, the sequence may be biased or too short.*

Théorème 1.3.3. *(Hull-Dobell) : A linear congruential generator has a maximal period ($p = m$) if, and only if, the following three conditions are met :*

1) The integers c and m are coprime.
2) For every prime factor $\hat{p}$ of m, $a - 1$ is a multiple of $\hat{p}$.
3) If m is a multiple of 4, then $a - 1$ is also a multiple of 4.

Remarque 2. *The situation is more complicated for multiplicative generators ($c = 0$) because there is always the possibility of blocking (if $x_0 = 0$). In this case, the maximal period can be at most $m - 1$.*

Exemple 1.3.4. *Consider the following congruential generator :*

$$x_{n+1} = (31415821 x_n + 1) \mod 100000000$$

Verification Steps

1. ***Condition 1 :*** *Here,* $c = 1$ *and* $m = 100000000$*. The greatest common divisor is :*

$$GCD(1, 100000000) = 1$$

2. ***Condition 2 :*** *The prime factors of* $m = 100000000$ *are* 2 *and* 5*, since :*

$$m = 10^8 = 2^8 \cdot 5^8$$

Let us calculate $a - 1$ *:*

$$a - 1 = 31415821 - 1 = 31415820$$

It is clear that $a - 1$ *is indeed a multiple of 2 and 5.*

3. ***Condition 3 :*** *Here,* $m = 100000000$ *is a multiple of* 4*. Also,* $a - 1$ *is a multiple of 4.*

All conditions of the Hull-Dobell theorem are met. Therefore, the congruential generator has a maximal period of $p = 100000000$*.*

Assignment : Research and explain the following algorithms :

1. The Mersenne Twister algorithm (pseudo-random number generator).
2. The Berlekamp-Massey algorithm (for Linear Feedback Shift Registers).
3. The Reed-Solomon algorithm (for error correction).
4. The RSA algorithm (public-key cryptography).

Instructions : Provide a detailed explanation of each algorithm, including its operation, applications, and importance. Include examples or pseudocode where possible. Prepare a written report (5-10 pages) and an oral presentation (10-15 minutes).

1.4 Testing Pseudo-random Number Generators

The objective of conformity tests is to evaluate whether a data sample originates from a random variable following a given theoretical distribution law $\mathbb{F}_0(x)$. Suppose that $\mathbb{F}(x)$ is the empirical distribution function based on the sample. The goal is to verify the null hypothesis :

$$\mathbb{H}_0 : \mathbb{F}(x) = \mathbb{F}_0(x)$$

against the alternative hypothesis :

$$\mathbb{H}_1 : \mathbb{F}(x) \neq \mathbb{F}_0(x).$$

These hypotheses allow us to determine if the data are compatible with the proposed theoretical distribution.

1.4.1 Chi-squared Test χ^2

The aim is to test the uniformity of a generated sequence $\{u_1, u_2, \ldots, u_n\}$ over the interval $[0, 1]$, where each $u_i = \frac{x_i}{m}$. The main steps of this test are as follows :

1. **Partitioning of the interval** $[0, 1]$ **:** The interval $[0, 1]$ is divided into r sub-intervals $I_1, I_2, \ldots, I_r$, where each sub-interval is defined by :

   $$I_i = [c_{i-1}, c_i], \quad c_0 = 0, \quad c_r = 1.$$

   Generally, $r \approx \sqrt{n}$ is chosen for a better distribution.

2. **Observed counts :** Let n_i be the observed frequency of class I_i, defined as :

   $$n_i = \text{card}\{k : u_k \in I_i,\ k = 1, \ldots, n\}, \quad \forall i = 1, \ldots, r.$$

3. **Probabilities :** The probability is given by :

   $$p_i = \mathbb{F}(c_i) - \mathbb{F}(c_{i-1}),$$

   where $\mathbb{F}(x)$ is the cumulative distribution function (CDF) associated with the uniform distribution, thus $\mathbb{F}(x) = x\mathbf{1}_{[0,1[}(x) + \mathbf{1}_{[1,+\infty)}(x)$. This implies that $p_i = c_i - c_{i-1}$.

4. **Pearson's test statistic :** The test statistic is given by :

   $$k_n^2 = \sum_{i=1}^{r} \frac{(n_i - np_i)^2}{np_i},$$

   where np_i represents the theoretical frequency expected for class I_i.

5. **Asymptotic distribution :** Pearson proved that the random variable k_n^2 asymptotically follows a Chi-squared distribution with $(r-1)$ degrees of freedom under the null hypothesis $\mathbb{H}_0$ (the sequence is uniform).
6. **Decision rule :**
   — If $k_n^2 < \chi^2_{(r-1,\alpha)}$, the fit is accepted, meaning we accept $\mathbb{H}_0$.
   — If $k_n^2 \geq \chi^2_{(r-1,\alpha)}$, the fit is rejected.

Remarque 3. *(Case of equiprobable intervals) To test uniformity on* $[0,1]$ *with equiprobable intervals, the probabilities* p_i *are all equal to* $\frac{1}{r}$*, i.e.,* $p_1 = p_2 = \cdots = p_r = \frac{1}{r}$*. The intervals are then defined by :*

$$I_i = \left[\frac{i-1}{r}, \frac{i}{r}\right[, \quad \forall i = 1, \ldots, r.$$

Exemple 1.4.1. *1000 random numbers were sampled using a pseudo-random number generator. We test whether these numbers are uniformly distributed over* $[0,1]$*. The interval is divided into* $r = 10$ *sub-intervals of width* $h = 0.1$*. The observed frequencies* (n_i) *and the theoretical frequencies* (np_i) *are given in the following table :*

Interval	$[0,h)$	$[h,2h)$	$[2h,3h)$	$[3h,4h)$	$[4h,5h)$	$[5h,6h)$	$[6h,7h)$	$[7h,8h)$	$[8h,9h)$	$[9h,1)$
n_i	*99*	*94*	*95*	*108*	*108*	*88*	*111*	*92*	*111*	*94*
np_i	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*	*100*

By performing the calculations, we find :

$$k_n^2 = \sum_{i=1}^{r} \frac{(n_i - np_i)^2}{np_i} = \frac{676}{100} = 6.76.$$

The number of degrees of freedom is $r - 1 = 9$*. The critical value of* χ^2 *at the significance level* $\alpha = 0.05$ *for 9 degrees of freedom is :*

$$\chi^2_{9,0.05} = 16.92.$$

The decision rule is as follows :
— *If* $k_n^2 < \chi^2_{9,0.05}$*, we accept the null hypothesis* $\mathbb{H}_0$ *: the sequence is uniformly distributed over* $[0,1]$*.*
— *Otherwise, we reject* $\mathbb{H}_0$*.*
In this case, $k_n^2 = 6.76 < 16.92$*. Consequently, we accept* $\mathbb{H}_0$*, meaning that the series* $\{u_1, \ldots, u_n\}$ *is uniformly distributed over* $[0,1]$*.*

1.4.2 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test is non-parametric, which means it can be used without assuming a precise form of the data distribution. Here are the main steps to perform this test :

1. A sample of n observations is generated using the random number generator ;
2. The sample values are sorted in ascending order ;
3. The empirical distribution function $\mathbb{F}_n(x)$, constructed from the n observations, is compared to the theoretical cumulative distribution function $\mathbb{F}(x)$. The test is based on calculating the maximum deviation between the two functions, defined as :

$$D_n = \max_x |\mathbb{F}_n(x) - \mathbb{F}(x)| ,$$

where :

$$\mathbb{F}_n(x) = \frac{\text{number of observations } \leq x}{n}.$$

A significance level α is set, and the critical value $d(\alpha)$ is determined from the Kolmogorov-Smirnov tables. This threshold is chosen such that $P(D_n > d(\alpha)) = \alpha$, where α represents the risk of error.

Decision Rule

The decision is made according to the following criteria :

— If $D_n > d(\alpha)$, the null hypothesis $\mathbb{H}_0$ is rejected, meaning that the empirical distribution $\mathbb{F}_n(x)$ is significantly different from the theoretical distribution $\mathbb{F}(x)$.

— Otherwise, $\mathbb{H}_0$ is accepted, indicating that the data are compatible with the theoretical distribution.