République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Djilali Liabès
Faculté des Sciences Exactes Master1 WIC-RSSI-ISI
Semestre 01 Modélisation et Simulation
TP N:3

SIMULATIONS DES NOMBRES ALÉATOIRES

Un générateur de nombres pseudo-aléatoires (PRNG), également appelé pseudorandom number generator en anglais, est un algorithme conçu pour produire une séquence de nombres présentant certaines propriétés de hasard. Cependant, il est important de noter que les sorties de ces générateurs ne sont pas réellement aléatoires ; elles se rapprochent simplement des propriétés idéales que l'on retrouve dans des sources complètement aléatoires. John von Neumann souligna cette réalité avec la remarque suivante : "Quiconque envisage des méthodes arithmétiques pour produire des nombres aléatoires est, bien sûr, en train de commettre un péché." Les véritables nombres aléatoires peuvent être obtenus à l'aide de dispositifs exploitant certaines propriétés physiques.

L'objectif de ce TP est d'introduire les principales méthodes de génération de nombres pseudo-aléatoires.

Générateurs Middle square

- crée par John von Neumann pour générer une séquence de nombres aléatoires à n chiffres. Elle prend le carré de la graine pour en faire un nombre sur $2*n$ bits. Si ce n'est pas déjà sur $2*n$ bits, ajoutez des zéros au début pour en faire un nombre sur $2*n$ bits. Après avoir obtenu un nombre sur $2*n$ bits, prenez les n chiffres du milieu. Nous avons alors un nouveau nombre aléatoire, et le processus se répète.
- A partir de la suite n_i, on obtient la suite $R_i = \frac{n_i}{10000}$.
- R_i est une suite de nombres aléatoires uniformes sur [0, 1].

Exemple:

i	n_i	n_i^2	Middle
1	11	0121	12
2	12	0144	14
3	14	0196	19
4	19	0361	36
5	36	1296	29
6	29	0841	84
7	84	7056	05
8	05	0025	02
9	02	0004	00
10	00	0000	00

Générateurs à congruence linéaire

Il s'agit de l'algorithme le plus utilisé pour produire des nombres aléatoires depuis qu'il a été inventé en 1948 par D. H. Lehmer. C'est la suite :

$$X_{n+1} = (a * X_n + c) \bmod m$$

avec a (multiplicateur), c (incrément), X_0 (germe), et m le module. Si on désire produire toujours la même séquence (ce qui est pratique à des fins de tests), on rentre toujours la même valeur de X_0. Dans tous les cas, les nombres de la suite sont compris entre 0 et $m-1$.

Remarque:

- $X_i \in \{0, 1, ..., m-1\}$.
- $R_i = \frac{X_i}{m}$, $R_i \in [0, (m-1)/m]$ sont les nombres pseudos-aléatoires.

Exercice 1 – Écrire sous R le code qui gènère la suite R_i pour un générateur Middle square (prenez n = 2, et n = 4).

Exercice 2 –

Ecrire sous R la fonction définie par $f(x) = 326x \bmod 332$. $\forall x = 0...30000$. **Indication:** le modulo est représenté par %% sous R.

Tracer le graphe de la fonction.

Exercice 3 –

Écrire une fonction en R pour implémenter un Générateur À Congruences Linéaires (GCL).

Testez l'algorithme avec les valeurs suivantes. Que pensez-vous de la période du générateur pour chaque cas.

1- $m = 16$, $a = 13$, $c = 7$, $X_0 = 4$.

2- $m = 16$, $a = 19$, $c = 0$, $X_0 = 5$(impair).

3- $m = 7$, $a = 3$, $c = 0$, $X_0 = 6$.

Pour $m = (2^{31}) - 1$, $a = 7^5$, $c = 0$. Comparer la distribution de ces n = 1000000 nombres avec la distribution uniforme sur [0,1].