TP 1 – Générateur de nombres aléatoires

Les nombres « aléatoires » fournis par un ordinateur n'ont généralement rien d'aléatoire. Ils sont produits par un algorithme de calcul de façon parfaitement déterministe. Simplement, la liste des nombres obtenus ressemble à des nombres tirés au hasard. On parle donc plutôt de « nombres pseudo-aléatoires ». Parmi les premiers exemples de générateurs figurent les générateurs congruentiels linéaires que l'on va étudier ici.

Partie 1 - Principe d'un générateur congruentiel linéaire

On part d'un nombre u_0 (appelé « graine » ou « seed » en anglais) et on définit une suite (u_n) par $u_{n+1} = au_n + b\,[m]$ avec $0 \leqslant u_n < m$ pour tout $n \in \mathbb{N}$; a, b, m sont des entiers naturels, $m \neq 0$.
On définit enfin la suite (v_n) par $v_n = \dfrac{u_n}{m}$ pour tout $n \in \mathbb{N}$.

1. Justifier que v_n prend au plus m valeurs distinctes et que, pour tout $n \in \mathbb{N}$, $v_n \in [0; 1[$.
2. Justifier que la suite (v_n) est périodique.

Partie 2 - Exemple d'un mauvais générateur

Dans cette partie, on considère le cas où $a = 1365$, $b = 1$ et $m = 2048$. On choisit $u_0 = 1$.

1. L'algorithme Python ci-dessous permet de représenter graphiquement les 500 premiers termes de la suite (v_n).

```
import matplotlib.pyplot
x=range(500)
y=[]
u=1
for k in range(500):
  u=1365*u+1
  u=u%2048
  v=u/2048
  y.append(v)

pyplot.plot(y,linestyle='none',marker='x')
pyplot.show()
```

   (a) Expliquer chaque ligne de l'algorithme puis l'implémenter sur machine.

   (b) Le graphique vous semble-t-il conforme à ce que l'on peut penser obtenir pour une suite de nombres aléatoires ?

2. Représenter graphiquement v_{n+1} en fonction de v_n. Pourquoi ce graphique alerte-t-il sur la qualité de ce générateur de nombre ?

3. L'objectif de cette question est de justifier l'allure du graphique obtenue à la question précédente.
   (a) Montrer que $3u_{n+1} \equiv -u_n + 3\,[2048]$.
   (b) En déduire qu'il existe un entier k tel que $3u_{n+1} = -u_n + 3 + 2048k$ et montrer que $0 \leqslant k \leqslant 3$.
   (c) En déduire que les couples $(v_n; v_{n+1})$ appartiennent à l'une des droites d'équations :

$$y = -\frac{1}{3}x + \frac{3}{2048} + \frac{k}{3} \quad \text{pour} \quad k = 0, 1, 2, 3$$

Partie 3 - Exemple du générateur de base rand

Le générateur de base rand était utilisé dans les années 1990 et au début des années 2000 par le logiciel Scilab notamment. Il correspond au cas où $a = 843\,314\,861$, $b = 453\,816\,693$ et $m = 2^{31}$.

Représenter graphiquement v_{n+1} en fonction de v_n pour ce générateur. Pourquoi peut-on considérer que le générateur de base rand est meilleur que celui de la partie 2 ?

Histoire

Un autre exemple de mauvais générateur de nombres aléatoires est le RANDU qui fut longtemps implanté dans les ordinateurs ($a = 65539$, $b = 0$, $m = 2^{31}$). Dans ce cas, il est possible de montrer qu'il existe une relation entre v_{n+2}, v_{n+1} et v_n.

Au début des années 2000, les générateurs congruentiels linéaires ont progressivement été remplacés par des générateurs plus performants. Actuellement, le générateur *Mersenne Twister* qui a été développé en 1997 par **Makoto Matsumoto** et **Takuji Nishimura** est par exemple utilisé par les logiciels Scilab et Python.

Makoto Matsumoto