brouillon :

Un algorithme est une suite d’instructions prédéfinies. C’est donc toujours déterministe. On aura donc toujours le même résultat en sortie. Afin de générer des nombres aléatoires, on utilise des algorithmes dont les résultats en sortie sont suffisamment imprévisibles pour considérer qu’ils sont aléatoires. On appelle ces nombres des nombres pseudo aléatoires, d’où le nom « Pseudo Random Number Generator » (PRNG).

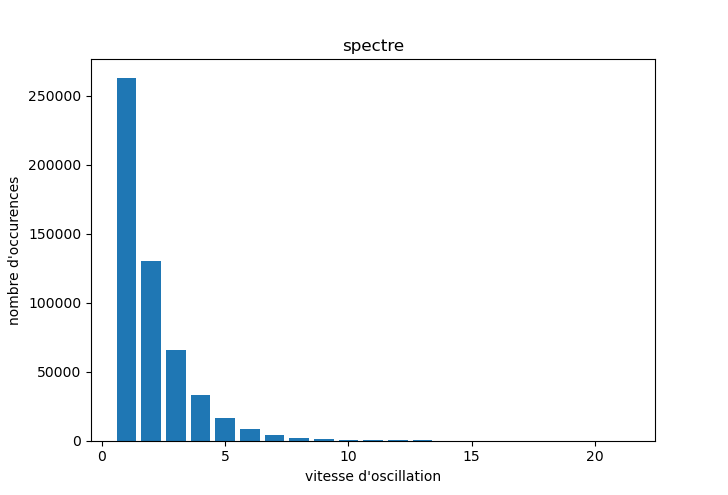
En python, la bibliothèque random se base sur le Mersenne Twister, un générateur développé par Makoto Matsumoto et Takuji Nishimura en 1997, et plus particulièrement la variante MT19937. Elle a notamment l’avantage d’avoir une période de 219937-1 et d’etre plus rapide que la plupart des PRNG de même qualité. Cependant, il présente plusieurs défauts, notamment celui de la complexité linéaire du LFSR , révélé par l’algorithme TestU01.

Aujourd’hui, on a surtout besoin de l’aléatoire afin de chiffrer correctement les données. Le problème des nombres pseudo aléatoires, c’est qu’il est relativement facile de les prédire, car ce sont des algorithmes déterministes qui les produisent. Par exemple pour MT19937, l’algorithme se base sur les 624 derniers nombres générés pour produire le suivant. Si quelqu’un récupère ces 624 nombres, il serait en mesure de prédire la suite de la génération.

Reslultats : fréquence d’apparition de 1/0 autour de 50%

Vitesse d’oscillation moyenne entre les 1 et les 0 : 2 bits

Spectre :



**Partie 1 : PRNG**

La génération de nombres aléatoires par ordinateur se fait grâce à des algorithmes, comme par exemple la méthode de Von Neumann ou encore l’algorithme Mersenne Twister. Cependant, un algorithme est une suite d’opérations prédéfinies. Il est donc déterministe et son résultat l’est aussi, ce qui est en opposition avec la génération de nombres aléatoires puisqu’il faudrait que le résultat soit complètement imprévisible. On a donc nommé les nombres issus de ces générateurs des nombres pseudo-aléatoires, d’où le nom Pseudo Random Number Generator (PRNG).

Afin de tester la qualité d’un PRNG, nous avons programmé en python un programme qui nous donne une séquence de bits, assez longue pour avoir des résultats significatifs. La bibliothèque random s’appuie sur l’algorithme de Mersenne Twister, qui a été développé par Makoto Matsumoto et Takuji Nishimura en 1997, et plus particulièrement la variante MT19937. Elle a notamment l’avantage d’avoir une période de 219937-1 et d’être plus rapide que la plupart des PRNG de même qualité. Cependant, il présente plusieurs défauts, comme celui de la complexité linéaire ou encore de ne pas avoir passé avec succès l’algorithme de test TestU01.

Afin de tester la qualité de Twister Mersenne, nous avons donc mesuré la fréquence d’apparition, la vitesse d’oscillation et le spectre de plusieurs séquences. Nous avons aussi mesuré la plus grande séquence contenue au moins deux fois dans le programme, mais compte tenu de la durée d’exécution, nous avons réduit la taille de la séquence pour ce paramètre.

TO DO : tableau avec les valeurs+figures

Aujourd’hui, l’aléatoire est de plus en plus nécessaire afin de chiffrer correctement les données. Le problème des nombres pseudo aléatoires, c’est qu’il est relativement facile de les prédire, car ce sont des algorithmes déterministes qui les produisent. Par exemple pour MT19937, l’algorithme se base sur les 624 derniers nombres générés pour produire le suivant. Si quelqu’un récupère ces 624 nombres, il serait en mesure de prédire la suite de la génération.