Université de Pau et des Pays de l'Adour
Collège STEE
Algorithmique Mathématiques I et Python

TP-04: Suites pseudo-aléatoires

Exercice 1. Générateurs congruentiels linéaires

Les nombres pseudo aléatoires forment une suite dont chaque terme dépend du précédent, selon la formule:

$$y_{n+1} = (a \times y_n + c) \mod m,$$

où a est le multiplicateur, c l'incrément et m le module. Le terme initial y_0 est appelé la graine (seed en anglais). Les nombres générés (sauf éventuellement le premier) sont compris entre 0 et $m - 1$.

Dans cet exercice, nous allons voir comment introduire de l'aléatoire numériquement. On va dans un premier temps générer des nombres aléatoires dans $[0, 1[$.

1. Écrire une fonction `generateur_congruentiel_lineaire(a,c,m,y0,n)` qui retourne le $n$ème terme de la suite pseudo-aléatoire (y_n).

2. Écrire une fonction `valeurs(a,c,m,y0,n)` qui calcule les n premiers termes de la suite (y_n), étant donné y_0.

3. En utilisant la fonction `valeurs(a,c,m,y0,n)`, afficher les 10 premières valeurs de la suite pseudo-aléatoire (y_n) pour les valeurs

   - `(a,c,m,y0)`$= (25, 16, 256, 12)$: dans ce cas, tous les nombres générés sont pairs.
   - `(a,c,m,y0)`$= (25, 16, 256, 11)$: dans ce cas, tous les nombres générés sont impairs.
   - `(a,c,m,y0)`$= (25, 16, 256, 10)$: dans ce cas, la suite est stationnaire.

4. Adapter la fonction pour qu'elle retourne le $n$ème term du générateur congruentiels linéaire de module m, de multiplicateur a, d'incrément c et de graine y_0. La fonction doit retourner y_n/m.

   Puis, afficher les 10 premières valeurs du générateur congruentiels linéaire de module $m = 2^{31}$, de multiplicateur $a = 843314861$, d'incrément $c = 453816693$ et de graine $y_0 = 1$. Est-ce que cette suite semble aléatoire?

5. On souhaite maintenant tester la qualité de notre générateur de variables aléatoires. Pour cela, nous allons étudier deux tests: le test spectral qui consiste à tracer les couples $(y_{2i}, y_{2i+1})_i$ à l'intérieur du carré unité $[0, 1[\times[0, 1[$ et le test consistant à comparer la cumulative numérique à la fonction de répartition exacte.

   (a) Tracer les couples $(y_{2i}, y_{2i+1})_i$ pour $n = 10\ 000$, $m = 2^{31}$, $a = 843314861$, $c = 453816693$, $y_0 = 1$. Ce générateur satisfait-il le test spectral?

   Indication:

   - Importer le sous-module `pyplot` du module `matplotlib`.
   - Utiliser la fonction `scatter` du module `pyplot`. Cette fonction permet de tracer un nuage de points. Il est possible d'indiquer dans la fonction quelques paramètres optionnels de forme ou d'esthétisme. Par exemple,

vous pouvez préciser la taille des points et le type du symbole utilisé en ajoutant à la fonction `scatter` les options: `s = 1` (qui revient à la taille du symbole au carré, ici 1. Le défaut est 20), `marker = ´*´` (pour le type du symbole, ici une étoile).

(b) En utilisant la fonction `hist` de PYTHON, tracer l'histogramme de y où l'on prendra 10 classes. Comparer l'histogramme à la fonction densité d'une loi uniforme sur $[0,1[$.

Indication:

- La fonction `hist` prendra les options suivantes:
  `hist(y, bins=10,density=True,color="white",edgecolor="black")`
  où le paramètre `bins` permet de préciser soit le nombre de classes, soit leurs bornes. Ici, on précise le nombre de classes qui est de 10. Le paramètre `density= True` permet de tracer les fréquences plutôt que les nombres en ordonnée (somme vaut 1).

(c) Tracer les couples $(y_{2i}, y_{2i+1})_i$ pour $n = 10\,000$, $m = 2^{31} - 1$, $a = 31$, $c = 0$, $y_0 = 1$. Ce générateur satisfait-il le test spectral?

(d) Tracer l'histogramme de y où l'on prendra 10 classes. Comparer l'histogramme à la fonction densité d'une loi uniforme sur $[0,1[$.

6. Utiliser maintenant la fonction `random()` du module `random` pour générer un vecteur colonne constitué de 10 nombres aléatoires dans $[0,1[$.

   Lancez plusieurs fois la commande, que se passe-t-il?