TP N°5 : MONTE-CARLO

I Méthode de Monte Carlo

Pour approcher une intégrale, nous avons aussi dans le programme une méthode probabiliste : la méthode de Monte-Carlo .

<u>**Principe :**</u> D'une manière générale, la moyenne (espérance) d'une variable aléatoire peut-être numériquement approchée en réalisant des simulations de la variable et en calculant la moyenne empirique des tirages (loi faible des grands nombres) . Cette idée est la base des méthodes dites de Monte-Carlo (quartier de Monaco abritant le fameux casino qui a donné son nom à la technique car, avant l'ordinateur, les tirages de nombres aléatoires étaient réalisés par des moyens physiques : dés, roulettes, jeux de cartes etc..).

Ainsi pour approcher une intégrale, on peut utiliser une approximation de l'espérance d'une variable aléatoire.

II Un premier exemple

1 On cherche à approcher $J = \displaystyle\int_0^1 \mathrm{e}^{-\frac{t^2}{2}}\, \mathrm{d}t$.

1. Donner la loi de la variable aléatoire X telle que $\sqrt{2\pi}P(0 \leqslant X \leqslant 1) = J$.

2. Soit $(X_n)_{n\in\mathbb{N}^*}$ une suite de variables aléatoires indépendantes de même loi que X.
   On définit la suite de variables aléatoires $(Y_n)_{n\in\mathbb{N}^*}$ par :

$$\forall n \in \mathbb{N}^*, Y_n = \begin{cases} 1 & \text{si } X_n \in [0,1] \\ 0 & \text{sinon} \end{cases}$$

   (a) Donner la loi de la variable aléatoire Y_n et déterminer son espérance.

   (b) Démontrer que :

$$\forall \epsilon > 0, \lim_{n\to+\infty} P\left(\left|\frac{Y_1 + \cdots + Y_n}{n} - \frac{1}{\sqrt{2\pi}}J\right| \geqslant \epsilon\right) = 0.$$

   (c) Ecrire une fonction Python d'entête `approxbis(nbsim)` qui permet de renvoyer la moyenne de `nbsim` simulations des Y_n. Qu'a-t-on approché ? Comment modifier la fonction de sorte qu'elle fournisse une approximation de J ?

   (d) Déterminer une valeur approchée de J en utilisant la bibliothèque `scipy.stats`.

III Un deuxième exemple

2 Le but de l'exercice est de déterminer une valeur approchée de $J = \displaystyle\int_0^1 4\sqrt{1-t^2}\, \mathrm{d}t$ par une méthode probabiliste.
Soit $(U_n)_{n\in\mathbb{N}^*}$ une suite indépendante de variables aléatoires uniforme sur $[0,1]$.

Posons pour tout $t \in [0,1], g(t) = 4\sqrt{1-t^2}$

1. Calculer J en utilisant le changement de variable $t = \sin(x)$.
2. Déterminer $E(g(U))$ et $V(g(U))$.
3. Notons $M_n = \dfrac{1}{n}\sum\limits_{i=1}^{n} g(U_i)$, montrer que $\forall \epsilon > 0, \lim\limits_{n\to+\infty} P\left(|M_n - J| \geqslant \epsilon\right) = 0$.
4. Programmation :
   (a) Ecrire une fonction `fonc(x)` permettant de calculer et de renvoyer la valeur de $g(x)$
   (b) Ecrire une fonction `graph(n)` permettant de tracer sur un graphique les points de coordonnées $(k, M_k)_{1\leqslant k\leqslant n}$.
   (c) Ecrire une fonction `approx4(n)` qui permet de renvoyer une simulation de la variable M_n.
   En testant votre fonction pour $n = 10$, $n = 100$, $n = 1000$ puis $n = 10000$, donner une valeur approchée de J.