CPE Lyon 4ETI 2024-2025

Probabilités discrètes : TP1

Exercice 0

L'objectif de cet exercice est de se familiariser avec la commande `matplotlib.pyplot.hist`. Pour cela, télécharger sur e-campus le script python `tp1_ex0.ipynb`, l'exécuter et interpréter les résultats observés.

Exercice 1

Est-il plus ou moins avantageux, lorsqu'on joue aux dés, de parier :

- A : sur l'apparition d'au moins un 6 quand on lance 4 fois un dé
- B : sur l'apparition d'au moins un double-six, quand on lance 24 fois deux dés

Le chevalier de Méré (Antoine Gombaud : 1607-1684), qui était un grand joueur, avait remarqué que le premier mode de pari avait une probabilité supérieure à 1/2. Se laissant abuser par un soi-disant argument de proportionnalité, le chevalier considérait que le deuxième mode de pari avait la même probabilité de réussite que (A) en raisonnant ainsi : en lançant un dé, il y a 6 issues ; en lançant deux 2 dés, il y en a 36, soit 6 fois plus. Puisqu'il est avantageux de parier sur l'apparition d'au moins un 6 en lançant le dé 4 fois de suite, il doit être avantageux de miser sur l'apparition d'au moins un double-six en lançant deux dés 24 fois de suite. Malheureusement pour le chevalier, les règles des probabilités sont plus complexes, et c'est Blaise Pascal (1623-1662) qui calcula la vraie probabilité de (B), très légèrement inférieure à 1/2 : le deuxième jeu n'est pas avantageux !

Ecrire un programme simulant le jeu (A) et le jeu (B) et vérifier que le premier jeu est plus avantageux que le second, contrairement à l'intuition du chevalier de Méré… On comparera avec les résultats théoriques, et on observera comment les probabilités empiriques évoluent lorsqu'on fait varier le nombre de simulations.

Indications :

- Créer une fonction `lancer_de_6_faces(N)` qui retourne dans un tableau de dimension `N` le résultats de `N` lancers de dé à 6 faces (utiliser la fonction `numpy.random.randint`)
- Vérifier la validité de cette fonction sous la forme d'un histogramme en mettant en évidence une loi discrète uniforme sur $[\![1, 6]\!]$ (utiliser la fonction `matplotlib.pyplot.hist` , cf. exercice 0)

Exercice 2 : méthode de Monte-Carlo

1) Jeu de fléchettes

   On lance au hasard des fléchettes sur une cible carrée de côté 1 sur laquelle est représenté un quart de disque de rayon 1 et de centre l'origine. On suppose que l'origine se situe sur le coin en bas à gauche de la cible.

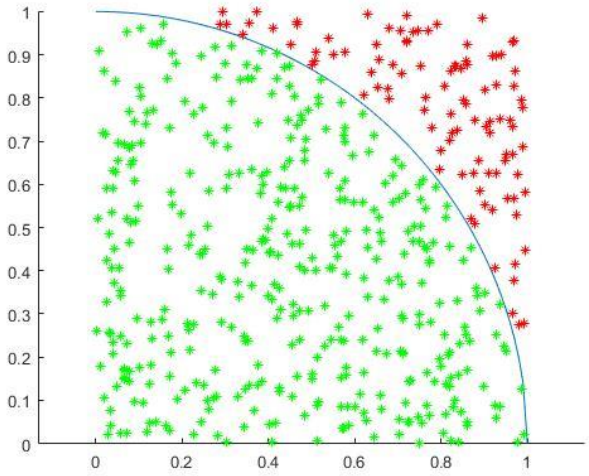
   On considère l'évènement A : « la fléchette arrive dans le quart de disque ».

   On note $p = P(A)$ la probabilité de l'évènement A et n le nombre de fléchettes lancées.

Concevoir un programme Python qui simule le lancer de fléchettes sur la cible et en déduire une approximation de π.

Le programme doit afficher :

- la figure suivante :



- les informations suivantes :

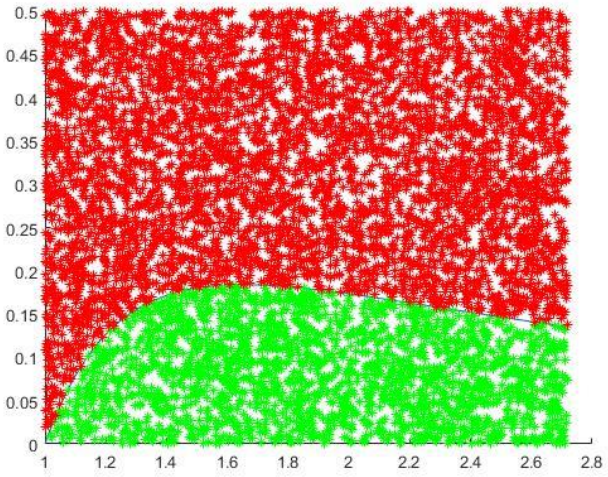
```
APPROXIMATION DE PI (METHODE DE MONTE-CARLO)
Nombre total de fléchettes : 500
Nombre de fléchettes dans le quadrant : 399
Valeur approximative de pi : 3.192000
```

2) La méthode précédente, basée sur un rapport d'aires, permet de calculer des intégrales. Par exemple, en remplaçant l'arc de cercle par la représentation graphique de la fonction $x \longmapsto \frac{\ln x}{x^2}$, appliquer ce principe au calcul suivant (attention il faudra aussi changer le carré unité par un rectangle judicieusement choisi) :

$$J = \int_1^e \frac{\ln x}{x^2}\, dx$$

Le programme doit afficher :

- la figure suivante :



- les informations suivantes :

```
VALEUR APPROXIMATIVE D'UNE INTEGRALE
Nombre total de fléchettes : 7000
Nombre de fléchettes sous la courbe : 2114
Valeur approximative de l'intégrale : 0.259461
Valeur exacte de l'intégrale : ????????
```

3) Faire une estimation de l'intégrale :

$$I = \iint_{\mathcal{D}} \frac{1}{(x+y)^3} dxdy$$

avec :

$$\mathcal{D} = \{(x,y) \in \mathbb{R}^2 \mid 1 \leq x \leq 3, \quad y \geq 2, \quad x + y \leq 5\}$$

a. Calculer « à la main » la valeur exacte de I (on dessinera dans un premier temps le domaine $\mathcal{D}$ puis le volume que représente l'intégrale I).

b. En utilisant la méthode de Monte-Carlo, donner une estimation de I.

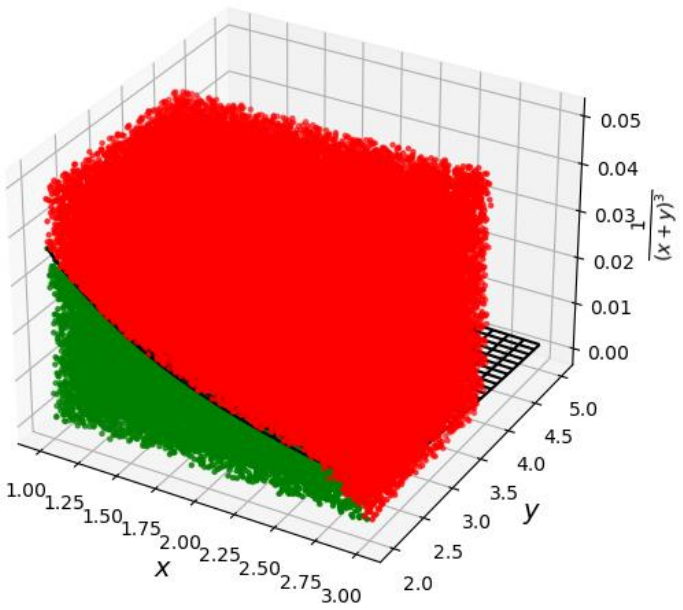
Indications :

- On prendra $N = 10000$ points
- Consulter l'aide en ligne de matplotlib pour la représentation de points (fléchettes) dans un repère en 3 dimensions
- Les lignes de code ci-dessous permettent d'afficher la surface $z = 1/(x+y)^3$

```
X, Y = np.meshgrid(np.arange(1, 3.1, 0.1), np.arange(2, 5.1, 0.1))
Z = 1 / (X + Y)**3
fig = plt.figure(figsize(5,5))
ax = fig.add_subplot(projection='3d')
ax.plot_wireframe(X, Y, Z, color='k')
ax.view_init(elev=60, azim=-60)
```

Le programme doit afficher :

- la figure suivante :



- les informations suivantes :

```
VALEUR APPROXIMATIVE D'UNE INTEGRALE DOUBLE
Nombre total de points : 100000
Nombre de points dans le quadrant : 13355
Valeur approximative de l'intégrale : 0.026710
Valeur exacte de l'intégrale : ????????
```