TP 7 : MÉTHODE DE MONTE-CARLO

Exercice 1. (Loi de Rayleigh) Soit a un réel > 0 et soit X une variable aléatoire suivant *la loi de Rayleigh de paramètre* a, c'est-à-dire admettant une densité donnée par :

$$f_a(x) = \begin{cases} 0 & \text{si} \quad x \leq 0 \\ \dfrac{x}{a^2} \exp\left(-\dfrac{x^2}{2a^2}\right) & \text{si} \quad x > 0 \end{cases}.$$

On admet que, si $V \hookrightarrow \mathcal{U}(]0,1])$, *alors* $Y = a\sqrt{-2\ln(V)}$ *suit la loi de Rayleigh de paramètre* a.

(1) Ecrire une fonction en Python notée `rayleigh` qui, étant donné un réel $a > 0$ et un entier $n \geq 1$, réalise et affiche n simulations indépendantes de la variable aléatoire X.

(2) Dans cette question, on considère la fonction suivante :

```
def exoray(n):
    x=rayleigh(1,n)
    y=0
    for i in range(1,n+1):
        if x[i]>1:
            y=y+1
    q=y/n
    return q
```

(a) Trouver la loi d'une variable aléatoire dont la valeur de y est, en fin de boucle, une simulation.

(b) De quel nombre peut-on s'attendre que q soit proche pour n assez grand et pourquoi?

Exercice 2. A l'aide de la méthode de Monte-Carlo, écrire une liste de commandes permettant de donner une valeur approchée de la somme S, et ce dans chacun des cas suivants :

$$(1)\ S = \sum_{k=0}^{+\infty} \frac{\ln(1+k)}{k!}, \qquad (2)\ S = \sum_{k=0}^{+\infty} \frac{\sqrt{k^2+1}}{3^k}, \qquad (3)\ S = \sum_{k=0}^{+\infty} \frac{1}{2^k k! \sqrt{k^2+1}}.$$

Exercice 3. A l'aide de la méthode de Monte-Carlo, écrire une liste de commandes permettant de donner une valeur approchée de l'intégrale I, et ce dans chacun des cas suivants :

$$(1)\ I = \int_0^1 \frac{\ln(1+x)}{1+x^2} dx, \qquad (2)\ I = \int_0^{+\infty} e^{-x} \sin(x) dx, \qquad (3)\ I = \int_0^{+\infty} e^{-\frac{x^2}{2}} \cos(x) dx.$$

Exercice 4. A l'aide de la méthode de Monte-Carlo, écrire une liste de commandes permettant de donner une valeur approchée de la probabilité p, et ce dans chacun des cas suivants :

(1) $p = P(X - Y \leq 4)$, $X \hookrightarrow \mathcal{U}([1,2])$, $Y \hookrightarrow \mathcal{U}([1,3])$ (X, Y sont indépendantes).

(2) $p = P(XY \leq 10)$, $X \hookrightarrow \mathcal{G}(\frac{1}{2})$, $Y \hookrightarrow \mathcal{P}(3)$ (X, Y sont indépendantes).

(3) $p = P(X^2 + Y^2 + Z^2 \leq 3)$, $X, Y, Z \hookrightarrow \mathcal{N}(1,4)$ (X, Y, Z sont indépendantes).

Exercice 5. Dans cet exercice, on considère la fonction suivante :

```
import numpy as np
import numpy.random as rd

def mont(n):
    x=rd.exponential(1,n)
    y=np.mean(x**2)
    return y
```

(1) Montrer que y contient une valeur approchée d'une intégrale (que l'on explicitera) pour n assez grand.

(2) Quelle est la valeur exacte de cette intégrale? Justifier.

Exercice 6. Dans cet exercice, on se propose de simuler une variable normale de paramètres m, σ^2. Pour ce faire, on considère une suite (X_n) de variables aléatoires indépendantes et suivant la loi uniforme sur $[0, 1]$.

(1) Justifier que $\frac{1}{2}(X_1 + ... + X_{48} - 24)$ fournit une bonne approximation de la loi normale centrée réduite.
(2) A l'aide uniquement de la commande `rd.random`, écrire une fonction en Python qui simule une variable normale centrée réduite.
(3) En déduire une fonction en Python qui, à partir de $(m, \sigma) \in \mathbb{R} \times \mathbb{R}_+^*$, simule la loi $\mathcal{N}(m, \sigma^2)$.

Exercice 7. Dans cet exercice, on considère une suite (X_n) de variables aléatoires indépendantes et suivant toutes la loi de Bernoulli de paramètre p, et l'on pose pour tout $n \in \mathbb{N}^*$:

$$\overline{X_n} = \frac{X_1 + ... + X_n}{n}.$$

(1) (a) Ecrire une fonction en Python notée `interconf1` qui, étant donnés un entier $n \geq 50$ et deux réels $p, \alpha \in]0, 1[$, réalise une simulation de la variable aléatoire $\overline{X_n}$, puis qui calcule et affiche les bornes d'un intervalle de confiance I_n de p au niveau de confiance $1 - \alpha$, calculé à l'aide de l'inégalité de Bienaymé-Tchebychev.
(b) En déduire une fonction en Python notée `frequence1` qui, étant donnés un entier $m \geq 100$ et un réel $p \in]0, 1[$, calcule les bornes de m intervalles de confiance I_{50} de p au niveau de confiance 95% obtenu à l'aide de la fonction `interconf1`, puis qui détermine et affiche la fréquence d'appartenance de p à ces intervalles de confiance.
(c) Après plusieurs exécutions de la fonction `frequence1` pour $p = 0.5$ et $m = 200$, on constate que la fréquence d'appartenance de p est toujours égale à 1. Que peut-on en déduire?

(2) (a) Ecrire une fonction en Python notée `interconf2` qui, étant donnés un entier $n \geq 50$ et deux réels $p, \alpha \in]0, 1[$, réalise une simulation de $\overline{X_n}$, puis qui calcule et affiche les bornes d'un intervalle de confiance asymptotique de p au niveau de confiance $1 - \alpha$.
(b) En déduire une fonction en Python notée `frequence2` qui, étant donnés un entier $m \geq 100$ et un réel $p \in]0, 1[$, calcule les bornes de m intervalles de confiance asymptotique I_{50} de p au niveau de confiance 95% obtenu à l'aide de la fonction `interconf2`, puis qui détermine et affiche la fréquence d'appartenance de p à ces intervalles de confiance.
(c) Après 4 exécutions de la fonction `frequence2` pour $p = 0.5$ et $m = 200$, on obtient les valeurs 0.955, 0.97, 0.965 et 0.96. Que peut-on en déduire?