Partiel MAO Probabilités et Statistiques (24 avril 2019, 16h15-18h15)

Le sujet comporte deux exercices indépendants. Un barème d'un total de 21 points + 2 points bonus vous est donné. Le nombre de points (tronqué à 20) sera la note finale.
Vous ne pouvez pas vous connecter à votre session personnelle, vous vous connectez à la session `mathens` *avec le mot de passe* `mathens`. *Tous vos fichiers devront être enregistrés dans le dossier "*`reMise_copie`*" présent sur le bureau. A la fin de la session, ce dossier comprendra (exclusivement) :*

— *le fichier Python contenant votre code source et portant le nom de votre numéro d'anonymat (se trouvant sur votre copie), par exemple "*`123456.py`*". Le code sera divisé en cellules Spyder correspondant aux questions des exercices (une cellule par question, avec le numéro de la question en commentaire).*

— *des enregistrements au format .png de tous les tracés. Les noms des fichiers feront référence au numéro de l'exercice et de la question, par exemple "*`exo_1_2.png`*". L'enregistrement se fait soit avec la commande* `plt.savefig`*, par exemple* `plt.savefig("exo_1_2.png")`*, soit directement depuis la fenêtre graphique avec le bouton correspondant.*

En aucun cas votre nom doit pouvoir être identifiable d'une façon ou d'une autre !

Exercice 1.— Méthode du rejet comparatif (5 points)

Consignes : il n'est pas autorisé d'utiliser dans cet exercice les fonctionnalités du module `scipy.stats`. *On utilisera pour la génération de v.a. de loi uniforme sur* $[0, 1]$ *des fonctions du module* `numpy`*, par exemple* `np.random.random_sample()`.

La loi du demi-cercle sur $[-1, 1]$ est la loi de densité

$$f(x) = \frac{2}{\pi}\sqrt{1 - x^2}, \quad x \in [-1, 1].$$

On souhaite simuler la loi du demi-cercle par la méthode du rejet comparatif à partir de la loi uniforme sur $[-1, 1]$.

1. (2P) Soit g la densité de la loi uniforme sur $[-1, 1]$. Expliciter l'algorithme qui simule la loi du demi-cercle par la méthode du rejet comparatif, à partir de la loi uniforme sur $[-1, 1]$.
   Consignes : i) on déterminera au préalable le plus petit $\lambda > 0$ *tel que* $f \leq \lambda g$ *et on utilisera ce* λ *dans l'algorithme, ii) on donnera le critère de rejet sous la forme "*$U > h(V)$*" pour* U *une v.a. uniforme sur* $[0, 1]$ *et* V *une v.a. uniforme sur* $[-1, 1]$*, indépendantes ; on simplifiera au maximum l'expression de* h*, iii) bien préciser ce que l'algorithme retourne*
2. (0.5P) Rappeler (sans preuve) la loi du nombre d'itérations que fait cet algorithme.
3. (2.5P) Simuler un échantillon de $M = 10\,000$ variables aléatoires suivant la loi du demi-cercle par la méthode du rejet comparatif. Tracer un histogramme et superposer la densité de la loi du demi-cercle. Enregistrer le tracé.

Exercice 2.— Une récurrence stochastique (16 points + 2 points bonus)

Soit $A_0, A_1, \ldots$ une suite de variables aléatoires positives et iid (indépendantes et identiquement distribuées). On définit une suite $Y_0, Y_1, \ldots$ par la récurrence suivante :

$$Y_0 = 1, \qquad Y_{n+1} = A_n Y_n + 1, \quad n \in \mathbb{N}.$$

On s'intéresse au comportement en temps long de cette suite.

Première partie théorique (4 points)

1. (1P) Montrer par récurrence que $Y_n = 1 + \sum_{k=1}^{n} A_{n-1} \cdots A_{n-k}$.
2. En déduire :
   (a) (0.5P) $Y_n \geq A_0 \cdots A_{n-1}$, $n \in \mathbb{N}$.
   (b) (0.5P) $Y_n \overset{\text{loi}}{=} 1 + \sum_{k=0}^{n-1} A_0 \cdots A_k$, $n \in \mathbb{N}$.

On supposera par la suite que $A_n = e^{\mu+\sigma\xi_n}$ avec $\xi_0, \xi_1, \ldots$ des v.a. indépendantes de loi normale centrée réduite, $\mu \in \mathbb{R}$ et $\sigma > 0$.

3. (1P) Soit $\mu > 0$. Montrer que $Y_n \to \infty$ presque sûrement.
4. (1P) Soit $\mu < 0$. Montrer que $Y \coloneqq 1 + \sum_{k=0}^{\infty} A_0 \cdots A_k < \infty$ presque sûrement. En déduire que Y_n converge en loi vers Y quand $n \to \infty$.

Deuxième partie expérimentale (12 points + 2 points bonus)

5. (1P) Définir une fonction Python `rec(mu,sigma,N)` qui retourne une réalisation de $(Y_0, \ldots, Y_N)$.
6. (1.5P) Générer 5 réalisations indépendantes de $(Y_0, \ldots, Y_N)$ pour $(\mu, \sigma, N) = (1, 1, 100)$. Pour chaque réalisation, tracer Y_n et $\log_{10} Y_n$ en fonction de n, en superposant les tracés des 5 réalisations (pour le tracé de $\log_{10} Y_n$, on peut utiliser la commande `plt.yscale('log')`). Faire de même pour $(\mu, \sigma, N) = (-1, 1, 100)$ (au total, 4 tracés que l'on pourra arranger dans une grille 2×2). Enregistrer les tracés.
7. (2P) Commenter les tracés (5 à 10 lignes). On pourra faire le lien avec des résultats théoriques ci-dessus (même si on n'a pas réussi à les démontrer).
8. (1.5P) Dans cet exercice, $(\mu, \sigma) = (-1, 1)$. On pose

$$N_n = \frac{1}{n+1} \sum_{k=0}^{n} \mathbf{1}_{Y_k \leq 2}.$$

   Tracer 30 réalisations de N_n en fonction de n et de $\log_{10} n$ (ou $\log_{10}(n+1)$ pour éviter des problèmes en $n = 0$) pour $0 \leq n \leq N = 10\,000$ (superposer les 30 tracés). Enregistrer le tracé.

9. (1P) Commenter ces tracés (5 lignes max.).
10. (1P) Définir une fonction Python `recM(mu,sigma,N,M)` qui retourne M réalisations de $(Y_0, \ldots, Y_N)$ dans un tableau de dimensions $(N + 1) \times M$ (on peut le faire en changeant seulement deux lignes de la fonction `rec`, mais toute solution sera valable).
11. (2P) On pose $(\mu, \sigma, N, M) = (-1, 1, 100, 1000)$. Générer M réalisations de $(Y_0, \ldots, Y_N)$ et tracer pour $n = 20, 50, 100$ les fonctions de répartitions empiriques et les histogrammes de Y_n et de $\log_{10} Y_n$ (donc 4 tracés en tout, en superposant ceux correspondant aux trois valeurs de n, penser à utiliser l'option `histtype='step'` pour l'histogramme). Enregistrer les tracés.
12. (2P) Commenter ces tracés (5 à 10 lignes). On pourra faire le lien avec des résultats théoriques ci-dessus (même si on n'a pas réussi à les démontrer).

Questions bonus (max. 2 points supplémentaires, barême plus strict)

— Etudier l'asymptotique de la queue $P(Y > x)$ quand $x \to \infty$ par le biais de la fonction de répartition empirique.

— Etudier l'asymptotique quand $n \to \infty$ de la fonction

$$\hat{F}_n(x) = \frac{1}{n+1} \sum_{k=0}^{n} \mathbf{1}_{Y_k \leq x}.$$

Corrigé

Exercice 1

1. La densité de la loi uniforme sur $[-1,1]$ est $g(x) = \frac{1}{2}\mathbf{1}_{[-1,1]}(x)$. On a

$$\lambda = \max_{x\in[-1,1]} \frac{f(x)}{g(x)} = \frac{4}{\pi} \max_{x\in[-1,1]} \sqrt{1-x^2} = \frac{4}{\pi}.$$

   Le critère de rejet est alors

$$U > h(V), \quad \text{avec } h(x) = \frac{f(x)}{\lambda g(x)} = \sqrt{1-x^2}.$$

   L'algorithme est alors le suivant :

   répète

   $U \leftarrow$ v.a. uniforme sur $[0,1]$

   $V \leftarrow$ v.a. uniforme sur $[-1,1]$

   tant que $U > \sqrt{1-V^2}$

   retourne V

2. Le nombre d'itérations de l'algorithme suit la loi géométrique de paramètre $p = 1/\lambda = \pi/4$.
3. voir code

Exercice 2

1. initialisation $(n = 0)$: évident, car $Y_0 = 1$.
   récurrence : on suppose l'égalité vraie pour n. Alors,

$$\begin{aligned} Y_{n+1} = A_n Y_n + 1 &= 1 + A_n \left(1 + \sum_{k=1}^{n} A_{n-1}\cdots A_{n-k}\right) \\ &= 1 + \sum_{k=0}^{n} A_n \cdots A_{n-k} = 1 + \sum_{k=1}^{n+1} A_{(n+1)-1}\cdots A_{(n+1)-k}. \end{aligned}$$

   CQFD.
2. (a) Puisque les A_n sont positifs, on minore par le dernier terme de la somme. Ceci donne

$$Y_n \geq A_0 \cdots A_{n-1}.$$

   (b) Puisque $(A_0,\ldots,A_{n-1}) \overset{\text{loi}}{=} (A_{n-1},\ldots,A_0)$, on a

$$Y_n = 1 + \sum_{k=1}^{n} A_{n-1}\cdots A_{n-k} \overset{\text{loi}}{=} 1 + \sum_{k=0}^{n-1} A_0 \cdots A_k.$$

3. Par la question 2.a) on a pour tout $n \in \mathbb{N}$,

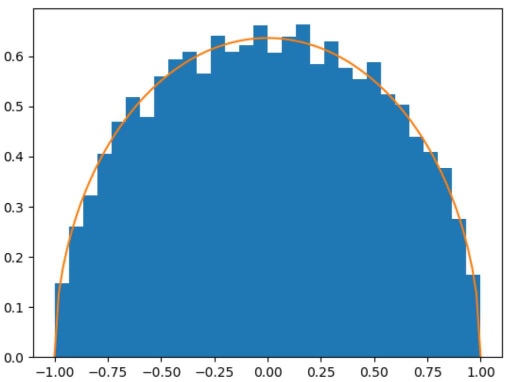
$$Y_n \geq A_0 \cdots A_{n-1} = \exp\left(\mu n + \sigma \sum_{k=0}^{n-1} \xi_k\right).$$

   Par la loi forte des grands nombres, $\frac{1}{n}\sum_{k=0}^{n-1} \xi_k \to 0$ p.s. quand $n \to \infty$. Par conséquent, presque sûrement, $Y_n \geq e^{\mu n/2}$ à partir d'un certain rang et donc $Y_n \to \infty$ p.s.

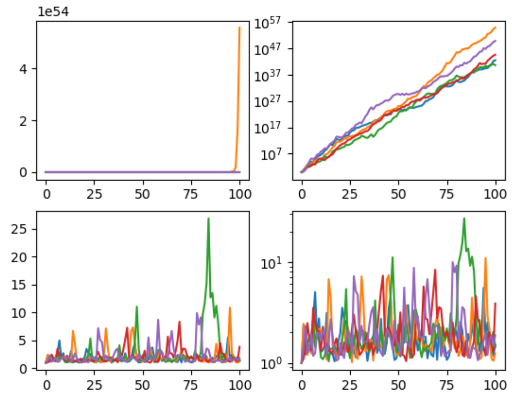
4. Par le même argument que la dernière question, on a $A_0 \cdots A_{n-1} \leq e^{\mu n/2}$ presque sûrement à partir d'un certain rang. Or, $\sum_n e^{\mu n/2} < \infty$ quand $\mu < 0$. De plus, les A_n sont positifs, donc $|A_0 \cdots A_{n-1}| = A_0 \cdots A_{n-1}$. La série converge donc absolument, presque sûrement. Puisque Y_n est égal en loi à la somme partielle $1 + \sum_{k=0}^{n-1} A_0 \cdots A_k$ par la question 2.b), ceci montre que Y_n converge vers Y en loi quand $n \to \infty$.
5. voir code
6. voir code
7. *Exemples d'observations :*
   — Quand $\mu = 1$, toutes les réalisations tendent vers l'infini, ce qui illustre la convergence p.s. vers l'infini de la question 3. Ceci est facile à voir sur le tracé de $\log_{10} Y_n$ en fonction de n : on voit bien que les tracés forment asymptotiquement une droite de pente positive. Ceci est en accord avec l'application de la loi des grands nombres dans la solution à la question 3. Le tracé de Y_n en fonction de n n'est pas très informatif, car il y a le tracé d'une seule réalisation qui domine les autres qui sont à une échelle plus petite.
   — Quand $\mu = -1$, Y_n ne converge pas presque sûrement mais reste proche de l'origine (de l'ordre de quelques dizaines au maximum) avec des fluctuations régulières. En particulier, Y_n ne tend pas vers l'infini. Ceci est en accord avec la convergence en loi de Y_n quand $n \to \infty$.
8. voir code
9. *Exemples d'observations :* Les tracés illustrent la convergence presque sûre de N_n quand $n \to \infty$, en effet, chacun des 30 tracés semble converger vers la même valeur (≈ 0.61). Le tracé en fonction de $\log_{10} n$ montre une convergence exponentielle en fonction de $\log_{10} n$, ce qui indique une convergence polynomiale en fonction de n.
10. voir code
11. voir code
12. *Exemples d'observations :* Les fonctions de répartition empiriques se ressemblent beaucoup et semblent approcher une fonction continue. Ceci illustre la convergence en loi de Y_n quand $n \to \infty$, montrée dans la question 4. La loi limite semble avoir une queue lourde : la valeur maximale des échantillons est de plusieurs centaines. Par conséquent, l'histogramme de Y_n est dégénérée : quasiment toutes les valeurs sont représentées par la première barre. Aussi, la fonction de répartition empirique est proche de 1 sur quasiment tout le tracé. En revanche, les tracés des fonctions de répartitions empiriques et de l'histogramme de $\log_{10} Y_n$ montrent clairement une limite en loi avec une densité continue. Ceci rappelle l'exercice sur la loi Pareto vu en cours. Ainsi, on peut conjecturer que Y admet une queue polynomiale.

Tracés

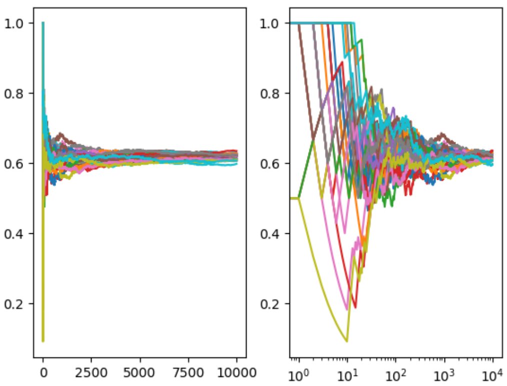
Exercice 1, question 3



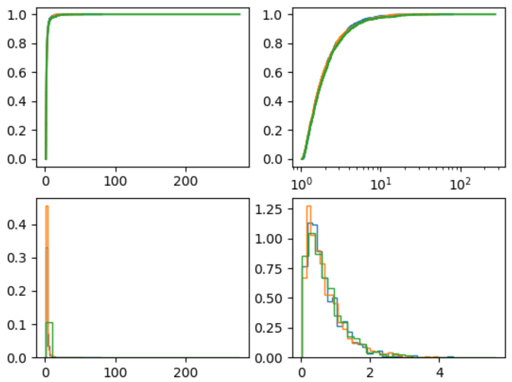
Exercice 2, question 6



Exercice 2, question 8



Exercice 2, question 11



Code

```python
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as scs

def TraceFctRepEmpirique(sample):
    n=len(sample)
    plt.step(np.sort(sample),np.arange(n)/n)

#%% Exo1 Q3

def simDemiCercle(n):
    X = np.zeros(n)
    for i in range(n):
        U,V = 2,0
        while U >= np.sqrt(1-V**2):
            U,V = np.random.random_sample(),2*np.random.random_sample()-1
        X[i] = V
    return X

def pdfDemiCercle(x):
    return 2/np.pi*np.sqrt(1-x**2)

n=10000
sample = simDemiCercle(n)
plt.figure()
X = np.linspace(-1,1,100)
plt.hist(sample, density=True, bins=30)
plt.plot(X,pdfDemiCercle(X))
plt.show()

#%% Exo2 Q5

def rec(mu,sigma,N):
    X = scs.norm.rvs(size=N)
    Y = np.zeros(N+1)
    Y[0]=1
    for n in range(N):
        Y[n+1] = np.exp(mu + sigma*X[n])*Y[n]+1
    return Y

#%% Exo2 Q6, Quelques tracés

N=100
plt.figure()

for i in range(5):
    Y = rec(1,1,N)
    plt.subplot(2,2,1)
    plt.plot(Y)
    plt.subplot(2,2,2)
    plt.plot(Y)
    plt.yscale('log')

    Y = rec(-1,1,N)
    plt.subplot(2,2,3)
    plt.plot(Y)
    plt.subplot(2,2,4)
    plt.plot(Y)
    plt.yscale('log')
plt.show()

#%% Exo2 Q8, Théorème ergodique

N = 10000
plt.figure()
for m in range(30):
    Y = rec(-1,1,N)
    N2 = np.cumsum(Y<=2)/np.arange(1,N+2)
    plt.subplot(1,2,1)
    plt.plot(N2)
    plt.subplot(1,2,2)
    plt.plot(N2)
    plt.xscale('log')
plt.show()

#%% Exo2 Q10

def recM(mu,sigma,N,M):
    X = scs.norm.rvs(size=(N,M))
    Y = np.zeros(shape = (N+1,M))
    Y[0]=1
    for n in range(N):
        Y[n+1] = np.exp(mu + sigma*X[n])*Y[n]+1
    return Y


#%% Exo2 Q11, Convergence en loi quand mu < 0

N = 100
M = 1000
Y = recM(-1,1,N,M)
plt.figure()
for n in [20,50,100]:
    plt.subplot(2,2,1)
    TraceFctRepEmpirique(Y[n])
    plt.subplot(2,2,2)
    TraceFctRepEmpirique(Y[n])
    plt.xscale('log')
    plt.subplot(2,2,3)
    plt.hist(Y[n],histtype='step',density=True,bins=30)
    plt.subplot(2,2,4)
    plt.hist(np.log(Y[n]),histtype='step',density=True,bins=30)
plt.show()

#%% Bonus: asympotique de la fct. de repartition quand mu < 0

N = 100
M = 1000
Y = recM(-1,1,N,M)
plt.figure()
plt.step(np.sort(Y[N]),np.arange(M,0,-1)/M)
plt.xscale('log')
plt.yscale('log')
plt.show()

#%% Bonus: comparaison de fct. de repartition empirique avec moyenne de Cé
    saro

N = 1000
M = 1000
Y = recM(-1,1,N,M)
plt.figure()
plt.subplot(1,2,1)
TraceFctRepEmpirique(Y[N])
TraceFctRepEmpirique(Y[:,M-1])
plt.xscale('log')
plt.subplot(1,2,2)
plt.hist(np.log(Y[N]),histtype='step',density=True,bins=20)
plt.hist(np.log(Y[:,M-1]),histtype='step',density=True,bins=20)
plt.show()

#%% Bonus: double moyenne (sur n et sur m) des fonctions de répartition
    empiriques
N = 1000
M = 1000
Y = recM(-1,1,N,M)
Yf = Y[100:,:].flatten()
plt.figure()
plt.subplot(2,2,1)
TraceFctRepEmpirique(Yf)
plt.subplot(2,2,2)
TraceFctRepEmpirique(Yf)
plt.xscale('log')
plt.subplot(2,2,3)
plt.hist(Yf,histtype='step',density=True,bins=30)
plt.subplot(2,2,4)
plt.hist(np.log(Yf),histtype='step',density=True,bins=30)
plt.show()

plt.figure()
plt.step(np.sort(Yf),np.arange(len(Yf),0,-1)/len(Yf))
plt.xscale('log')
plt.yscale('log')
plt.show()
```