DM 7

A rendre le Mercredi 27 Novembre

Exercice 1

Dans ce problème, la lettre n désigne un entier naturel non nul.
On note f_n la fonction définie sur $\mathbb{R}$ par : $f_n(x) = xe^{-n/x}$ si $x \neq 0$ et $f_n(0) = 0$.
On note $(\mathscr{C}_n)$ la courbe représentative de f_n dans un repère orthonormé $(O, \vec{i}, \vec{j})$.

1. (a) Montrer que f_n est continue à droite en 0.
   (b) Montrer que f_n est dérivable à droite en 0 et donner la valeur du nombre dérivé à droite en 0 de f_n.
2. (a) Montrer que f_n est dérivable sur $]-\infty, 0[$ et sur $]0, +\infty[$.
   Pour tout réel x non nul, calculer $f_n'(x)$ puis étudier son signe.
   (b) Calculer les limites de f_n en $+\infty$, $-\infty$ et 0^-, puis donner le tableau de variation de f_n.
3. (a) Rappeler le développement limité à l'ordre 2 de e^u lorsque u est au voisinage de 0.
   (b) En déduire que, lorsque x est au voisinage de $+\infty$ ou au voisinage de $-\infty$, on a :
   $$f_n(x) = x - n + \frac{n^2}{2x} + o\left(\frac{1}{x}\right).$$
   (c) En déduire qu'au voisinage de $+\infty$, ainsi qu'au voisinage de $-\infty$, $(\mathscr{C}_n)$ admet une asymptote oblique $(\mathscr{D}_n)$ dont on donnera une équation. Préciser la position relative de $(\mathscr{D}_n)$ et $(\mathscr{C}_n)$ aux voisinages de $+\infty$ et de $-\infty$.
   (d) Donner l'allure de la courbe $(\mathscr{C}_1)$.
4. (a) Montrer qu'il existe un unique réel, que l'on notera u_n, tel que $f_n(u_n) = 1$.
   (b) Vérifier que, pour tout n de $\mathbb{N}^*$, u_n est strictement supérieur à 1 et que u_n est solution de l'équation $x\ln(x) = n$.
   (c) Étudier la fonction g définie sur $[1, +\infty[$ par $g(x) = x\ln(x)$.
   En déduire, en utilisant la fonction g^{-1}, que $\lim\limits_{n\to+\infty} u_n = +\infty$.
   (d) Justifier la relation $\ln(u_n) + \ln(\ln(u_n)) = \ln(n)$, puis montrer que $\ln(u_n) \underset{n\to+\infty}{\sim} \ln(n)$.
   En déduire un équivalent de u_n lorsque n est au voisinage de $+\infty$.
5. (a) Montrer que la suite $(u_n)_{n\geq 1}$ est strictement croissante.
   (b) Montrer que : $f_n(u_{n+1}) = \exp(\dfrac{1}{u_{n+1}})$.
6. On pose $I_n = \displaystyle\int_{u_n}^{u_{n+1}} f_n(t)dt$.
   (a) Montrer que : $1 \leq \dfrac{I_n}{u_{n+1} - u_n} \leq \exp\left(\dfrac{1}{u_{n+1}}\right)$.
   (b) En déduire un équivalent de I_n lorsque n est au voisinage de $+\infty$.
   (c) Montrer alors que la série de terme général I_n est divergente.

Exercice 2

On lance indéfiniment une pièce équilibrée.
On s'intéresse au rang du lancer auquel on obtient pour la première fois deux Pile consécutifs.
On modélise cette expérience aléatoire par un espace probabilisé $(\Omega, \mathcal{A}, P)$. On note alors X la variable aléatoire égale au rang du lancer où, pour la première fois, on obtient deux Pile consécutifs.
Si on n'obtient jamais deux Pile consécutifs, on conviendra que X vaut -1.
Par exemple, si on obtient dans cet ordre : Pile, Face, Face, Pile, Pile, Pile, Face,... alors X prend la valeur 5.
Pour tout entier n supérieur ou égal 1, on pose les événements suivants :

- F_n : "Obtenir Face au n-ième lancer"
- P_n : "Obtenir Pile au n-ième lancer"

La suite $(P_n)_{n\geq 1}$ est donc une suite d'événements mutuellement indépendants.
Pour tout entier n supérieur ou égal à 2, on pose les événements suivants :

- U_n : "Au cours des n premiers lancers, on obtient au moins une fois la succession de deux piles consécutifs"
- $B_n = P_{n-1} \cap P_n$.

Enfin, pour tout entier n supérieur ou égal à 2, on note : $u_n = P(U_n)$ et $a_n = P(X = n)$.

Partie A

1. Exprimer les événements $[X = 2]$, $[X = 3]$ et $[X = 4]$ à l'aide de certains événements P_k et F_k. En déduire les valeurs de a_2, a_3 et a_4.

2. Montrer que, pour tout entier n supérieur ou égal à 2 : $u_n = \displaystyle\sum_{k=2}^{n} a_k$.

3. (a) Recopier et compléter la fonction `Python` ci-dessous afin qu'elle simule les lancers de la pièce jusqu'à l'obtention de deux Pile consécutifs, et qu'elle renvoie le nombre de lancers effectués.

```
def simulX():
    tirs = 0
    pile = 0
    while pile ..... :
        if rd.random()<1/2 :
            pile = pile+1
        else:
            pile = .....
        tirs = .....
    return(tirs)
```

   (b) Écrire une fonction `Python` d'entête `def moyenne(n)` qui simule `n` fois l'expérience ci-dessus et renvoie la moyenne des résultats obtenus.

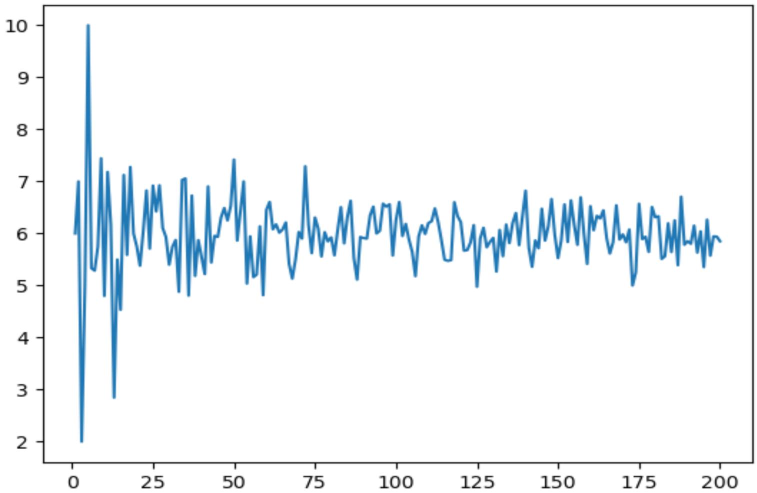
   (c) On calcule `moyenne(n)` pour chaque entier `n` de $[\![1, 200]\!]$ et on stocke les résultats dans un vecteur `U` puis on trace le graphe associé :

```
U = np.zeros(200)
for n in range(200):
    U[n] = moyenn(n+1)

```

```
N = np.arange(1,201)
plt.plot(N,U)
plt.show()
```

On obtient le résultat graphique suivant :



Que pouvez-vous conjecturer sur la variable aléatoire X ?

Partie B

4. (a) Montrer que pour tout entier n supérieur ou égal à 2 :

$$P\left(U_{n+1}\right)=P\left(U_n\right)+P\left(B_{n+1}\right)-P\left(U_n\cap B_{n+1}\right)$$

   (b) Montrer que pour tout entier n supérieur ou égal à 4 :

$$U_n\cap B_{n+1}=\left(U_{n-2}\cap F_{n-1}\cap P_n\cap P_{n+1}\right)\cup\left(P_{n-1}\cap P_n\cap P_{n+1}\right).$$

   (c) En déduire que pour tout entier n supérieur ou égal à 4 :

$$u_{n+1}=u_n+\frac{1}{8}\left(1-u_{n-2}\right).$$

5. Démontrer que la suite $(u_n)_{n\geq 4}$ est croissante, puis qu'elle converge vers 1.

6. En déduire que :

$$P\left(X=-1\right)=1-P\left(\bigcup_{n=2}^{+\infty}U_n\right)=0$$

Partie C : Étude de l'espérance de X.

Dans cette partie, on pose pour tout entier $n\geq 2$:

$$v_n=1-u_n \quad \text{et} \quad S_n=\sum_{k=2}^{n}kP(X=k).$$

7. Montrer que pour tout entier n supérieur ou égal à 4 :

$$v_n-v_{n+1}=\frac{1}{8}v_{n-2}$$

8. Justifier que pour tout entier n supérieur ou égal à 2 :

$$P(X=n+1)=v_n-v_{n+1}$$

9. Démontrer alors par récurrence que, pour tout entier n supérieur ou égal à 2 :

$$S_n = 6 - 8v_{n+2} - nv_n$$

10. En déduire que la suite $(S_n)_{n\geq 2}$ est croissante et majorée.

11. Montrer que X admet une espérance.

12. (a) Démontrer que la suite $(nv_n)_{n\geq 2}$ converge vers un réel λ.
    (b) Montrer que si λ est non nul, alors la série de terme général v_n est divergente.
    A l'aide de l'égalité démontrée à la question 7, obtenir une contradiction.
    (c) Donner alors la valeur de l'espérance de X.

Exercice 3

On considère 3 boîtes contenant chacune 4 jetons :

- La boîte 1 contient 3 jetons portant le numéro 1 et 1 jeton portant le numéro 3.
- La boîte 2 contient 2 jetons portant le numéro 1 et 2 jetons portant le numéro 3.
- La boîte 3 contient 2 jetons portant le numéro 2 et 2 jetons portant le numéro 3.

On effectue alors une suite de tirages selon les règles suivantes :

- Le premier tirage a lieu dans la boîte 1.
- Après chaque tirage dans une boîte, le jeton tiré est remis dans la même boîte.
- Pour tout entier naturel non nul n, le $(n+1)$-ième tirage a lieu dans la boîte du numéro du jeton tiré lors du n-ième tirage.

Pour tout $n \in \mathbb{N}^*$, on désigne par X_n la variable aléatoire égale au numéro de la boîte où on effectue le n-ième tirage. On a par exemple que $X_1 = 1$. On note $E(X_n)$ l'espérance de X_n.
On pose enfin $U_n = \big(P(X_n = 1) \quad P(X_n = 2) \quad P(X_n = 3)\big)$.

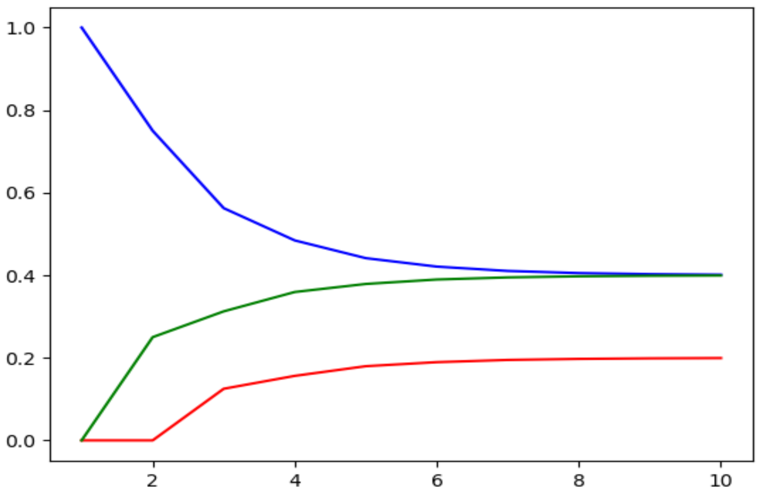
1. Loi de X_n :
   Soit n un entier naturel non nul.
   (a) Déterminer les probabilités $P_{(X_n=i)}(X_{n+1} = j)$ pour tout couple $(i, j) \in \{1, 2, 3\}^2$. Justifier.
   (b) Exprimer $P(X_{n+1} = 1)$ en fonction des probabilités $P(X_n = 1)$, $P(X_n = 2)$, $P(X_n = 3)$.
   Faire de même avec $P(X_{n+1} = 2)$ et $P(X_{n+1} = 3)$ en fonction de $P(X_n = 1)$, $P(X_n = 2)$, $P(X_n = 3)$.
   (c) Expliciter une matrice M telle que $U_{n+1} = U_nM$. Préciser la valeur de U_1.
   (d) Tracer le graphe probabiliste associé à M.
   (e) Compléter la fonction `loideX` suivante pour que, étant donné un entier naturel n non nul, elle permette d'obtenir la vecteur-ligne U_n :

```python
def loideX(n):
    U = .....
    M = .....
    for k in range(2,n+1):
        U = .....
    return(U)
```

(f) On considère les instructions suivantes :

```python
A = np.zeros((10,3))
for k in range(10) :
    A[k,:] = loideX(k+1)
N = np.arange(1,11)
plt.plot(N, A[:, 0], color='blue')
plt.plot(N, A[:, 1], color='red')
plt.plot(N, A[:, 3], color='green')
plt.show()
```

On exécute ces instructions depuis l'éditeur et on obtient le résultat suivant :



Que représente la variable `A` ? Comment interpréter le résultat graphique obtenu ?

2. Espérance de X_n :

(a) Compléter la fonction `esperancedeX` suivante pour que, étant donné un entier naturel n non nul, elle permette d'obtenir l'espérance de X_n :

```python
def esperancedeX(n):
    U = loideX(n)
    return( ..... )
```

(b) On considère les matrices colonnes L et J définies par $L = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ et $J = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

Déterminer deux réels α et β tels que $(*)$: $ML = \alpha L + \beta J$.

(c) En multipliant l'égalité $(*)$ par U_n à gauche, en déduire que : $E(X_{n+1}) = \alpha E(X_n) + \beta$.

(d) En déduire l'expression de $E(X_n)$ en fonction de n, puis $\lim\limits_{n \to +\infty} E(X_n)$.

3. Simulation de X_n :

(a) Compléter la fonction `boitesuivante(i)` qui simule un tirage dans la boite numéro i et retourne le numéro du jeton obtenu :

```python
def boitesuivante(i):
    j = i
    r = rd.random()
    if j == 1:
```

```python
        if r<1/4 :
            j = .....
    elif j == 2:
        if ..... :
            j = .....
        else:
            j = .....
    else:
        if .... :
            j = .....
    return(j)
```

(b) Compléter la fonction `simulX` suivante pour que, étant donné un entier naturel n non nul, elle permette de simuler l'expérience aléatoire et d'obtenir la valeur de X_n :

```python
def simulX(n):
    i = .....
    for k in range(n):
        i = .....
    return(i)
```

(c) On considère les instructions suivantes :

```python
L = []
for k in range(10000):
    L.append(simulX(10))

print(L.count(1)/10000)
print(L.count(2)/10000)
print(L.count(3)/10000)
```

On les exécute depuis l'éditeur et on obtient les résultats suivants dans la console :

```
0.4007
0.2029
0.3964
```

Que représente la variable `L` ? Interpréter les résultats obtenus.

(d) Que fait l'instruction `np.mean(L)` ? Lorsqu'on l'exécute dans la console, on obtient le résultat suivant :

```
1.9957
```

Interpréter ce résultat.