DS8 (version A)

Exercice 1

On note $f : \mathbb{R} \to \mathbb{R}$ l'application de classe C^2 sur $\mathbb{R}$, définie, pour tout $x \in \mathbb{R}$, par :

$$f(x) = x - \ln(1 + x^2)$$

et $\mathcal{C}$ la courbe représentative de f dans un repère orthonormé.
On donne la valeur approchée : $\ln(2) \approx 0,69$.

Partie I : Étude de f et tracé de $\mathcal{C}$

1. ***a)*** Calculer, pour tout $x \in \mathbb{R}, f'(x)$.

b) En déduire le sens de variation de f.

c) Calculer, pour tout $x \in \mathbb{R}$, $f''(x)$.

2. Déterminer la limite de f en $-\infty$ et la limite de f en $+\infty$.

3. Montrer que $\mathcal{C}$ admet deux points d'inflexion dont on déterminera les coordonnées.

4. Tracer $\mathcal{C}$. On précisera la tangente à $\mathcal{C}$ en l'origine et en chacun des points d'inflexion.

Partie II : Étude d'une fonction de deux variables réelles associée à f

On considère l'application $F : \mathbb{R}^2 \to \mathbb{R}$, définie, pour tout $(x, y) \in \mathbb{R}^2$, par :

$$F(x, y) = f(x + y) - f(x) - f(y)$$

5. Montrer que F est de classe $\mathcal{C}^1$ sur $\mathbb{R}^2$ et exprimer, pour tout $(x, y) \in \mathbb{R}^2$, les dérivées partielles premières de F en (x, y), à l'aide de f', x et y.

6. ***a)*** Montrer que, pour tout $(x, y) \in \mathbb{R}^2$,

$$\nabla(f)(x,y) = 0_{\mathscr{M}_{2,1}(\mathbb{R})} \iff \left\{ \begin{array}{l} f'(x) = f'(y) \\ f'(x+y) = f'(x) \end{array} \right. \iff \left\{ \begin{array}{l} x(1+y^2) = y(1+x^2) \\ (x+y)(1+x^2) = x(1+(x+y)^2) \end{array} \right.$$

b) Soit $(x, y) \in \mathbb{R}^2$. Compléter la formule suivante : $x + xy^2 - y - yx^2 = (x - y)(\dots)$.

c) En déduire que F possède exactement trois points critiques : $(0, 0)$, $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ et $(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$.

7. ***a)*** Calculer les dérivées partielles secondes de F.

b) Est-ce que F admet un extremum local en $(0, 0)$?

c) Montrer que F admet un extremum local en $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$. Préciser sa nature.

d) En déduire la nature du point critique $(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$.

Exercice 2

Soit a un réel strictement positif.

1. Pour tout entier n supérieur ou égal à 2, on pose : $I_n(a) = \displaystyle\int_a^{+\infty} \frac{1}{t^n}\, dt$.

Montrer que l'intégrale $I_n(a)$ converge et vaut $\dfrac{1}{(n-1)\, a^{n-1}}$.

2. Soit f la fonction définie sur $\mathbb{R}$ par :

$$f : t \mapsto \begin{cases} 0 & \text{si } t < a \\ \dfrac{3\,a^3}{t^4} & \text{si } t \geqslant a \end{cases}$$

a) Démontrer que f est bien une densité de probabilté. Soit X une variable aléatoire admettant f pour densité.

b) Donner la fonction de répartition de X.

c) Démontrer que X admet une espérance et calculer cette espérance.

d) Démontrer que X admet une variance et que celle-ci vaut $\dfrac{3\,a^2}{4}$.

3. Soit U une variable aléatoire suivant la loi uniforme sur $]0,1]$. On pose : $Y = \dfrac{a}{U^{\frac{1}{3}}}$.

a) Déterminer $Y(\Omega)$.

b) Déterminer la fonction de répartition de Y et vérifier que Y et X suivent la même loi.

c) Écrire une fonction **Python** nommée `simulX(a, m, n)` prenant en argument un réel a strictement positif et deux entiers naturels m et n non nuls, qui renvoie une matrice à m lignes et n colonnes dont chaque coefficient est un réel choisi de façon aléatoire en suivant la loi de X. Ces réels seront choisis de façon indépendante. On rappelle que si m et n sont des entiers naturels non nuls, l'instruction `rd.random([m,n])` renvoie une matrice à m lignes et n colonnes dont chaque coefficient suit la loi uniforme sur $[0,1[$, ces coefficients étant choisis de façon indépendantes.

4. ***a)*** Calculer $\mathbb{P}(\,[X > 2a]\,)$.

b) Calculer $\mathbb{P}_{[X>2a]}(\,[X > 6a]\,)$.

c) On suppose que la fonction **Python** de la question ***3.c)*** a été programmée correctement. Compléter le script ci-dessous afin qu'il renvoie une valeur permettant de vérifier le résultat de la question précédente.

```
a = 10
N = 100000
s1 = 0
s2 = 0
X = simulX(a, 1, N)[0]
for k in range(N):
    if ______________:
        s1 = s1 + 1
        if X[k] > 6*a:
            _______________
if s1 > 0:
    print(_________)
```

On cherche dans la suite de l'exercice à estimer le paramètre a.
On dit qu'un estimateur T_n de a est *sans biais* si $\mathbb{E}(T_n) = a$.
Soit n un entier naturel non nul, et X_1, …, X_n n variables aléatoires indépendantes et suivant toutes la même loi que X.

5. On pose : $V_n = \dfrac{2}{3n} \sum\limits_{k=1}^{n} X_k$.

a) Montrer que V_n est un estimateur sans biais pour le paramètre a.

b) Montrer que $\mathbb{V}(V_n) = \dfrac{a^2}{3n}$.

6. On pose : $W_n = \min(X_1, \ldots, X_n)$.

a) Déterminer la fonction de répartition de W_n et vérifier que W_n est bien une variable aléatoire à densité.

b) Montrer que W_n admet pour densité la fonction f_n définie sur $\mathbb{R}$ par :

$$f_n : t \mapsto \begin{cases} 0 & \text{si } t < a \\ \dfrac{3na^{3n}}{t^{3n+1}} & \text{si } t \geqslant a \end{cases}$$

c) Démontrer que W_n admet une espérance et calculer cette espérance.
Déterminer alors l'unique réel λ_n dépendant de n tel que $\lambda_n W_n$ est un estimateur sans biais pour le paramètre a.

d) Montrer que $\mathbb{V}(\lambda_n W_n) = \dfrac{a^2}{3n\,(3n-2)}$.

7. Comparer les variances des estimateurs V_n et $\lambda_n W_n$. Quel est le meilleur estimateur selon vous ?

8. On rappelle que si `A` est un tableau (ou un vecteur ligne) **Python**, l'instruction `sum(A)` renvoie la somme des coefficients du tableau `A`.

a) Compléter la fonction ci-dessous afin qu'elle réalise m simulations de la variable aléatoire V_n et renvoie les résultats obtenus sous forme d'un tableau à m éléments :

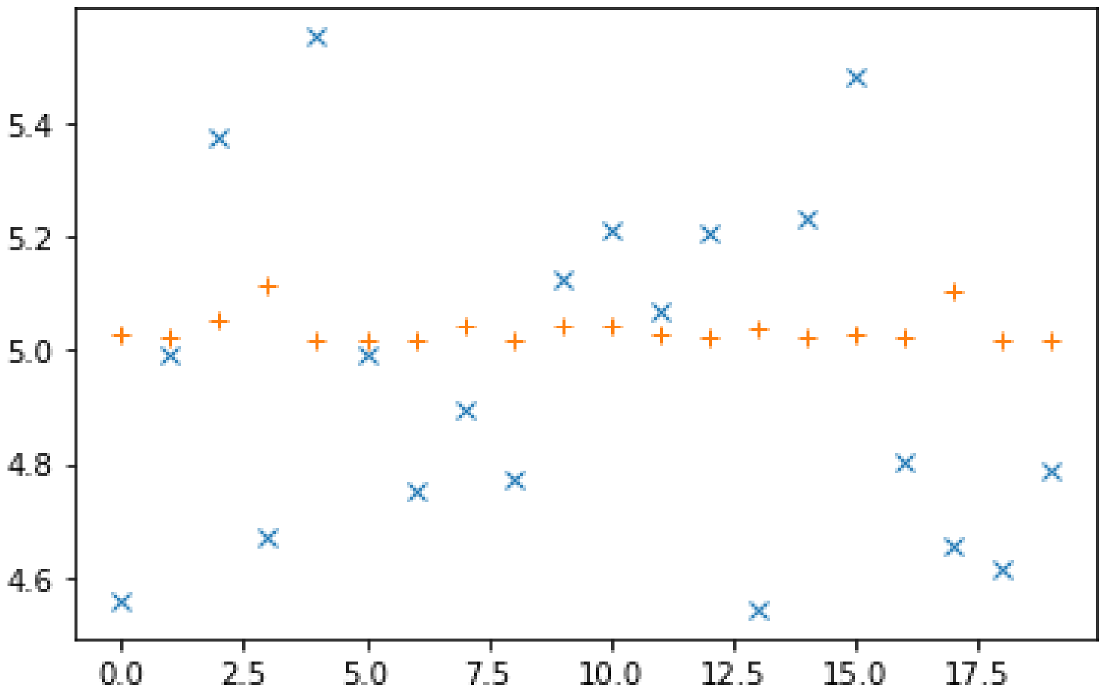
```
def simulV(a,m,n):
    X = simulX(a,m,n)
    V = np.zeros(m)
    for k in ____________:
        V[k] = ____________________________
    return V
```

Pour la suite, on prend $n = 100$ et on suppose que l'on dispose d'une fonction similaire `simulW` permettant d'obtenir m simulations de la variable aléatoire $\lambda_n W_n$.

b) Compléter les lignes ci-dessous pour écrire le script qui a permis d'obtenir le graphique présenté :

```
V = simulV(..., ..., ...)
W = simulW(..., ..., ...)
plt.plot(..., 'x')
plt.plot(..., '+')
```

On justifiera la réponse pour les deux dernière lignes à l'aide de la question 7.



Problème

Partie 1 : étude d'une variable aléatoire

Les sommets d'un carré sont numérotés 1, 2, 3, et 4 de telle façon que les côtés du carré relient le sommet 1 au sommet 2, le sommet 2 au sommet 3, le sommet 3 au sommet 4 et le sommet 4 au sommet 1.
Un mobile se déplace aléatoirement sur les sommets de ce carré selon le protocole suivant :

- Au départ, c'est à dire à l'instant 0, le mobile est sur le sommet 1.
- Lorsque le mobile est à un instant donné sur un sommet, il se déplace à l'instant suivant sur l'un quelconque des trois autres sommets, et ceci de façon équiprobable.

Pour tout $n \in \mathbb{N}$, on note X_n la variable aléatoire égale au numéro du sommet sur lequel se situe le mobile à l'instant n. D'après le premier des deux points précédents, on a donc $X_0 = 1$.

1. Représenter le graphe probabiliste associé à cette chaîne de Markov.

2. Donner la loi de X_1, ainsi que l'espérance $\mathbb{E}(X_1)$ de la variable X_1.
On admet pour la suite que la loi de X_2 est donnée par :

$$\mathbb{P}([X_2 = 1]) = \frac{1}{3}, \quad \mathbb{P}([X_2 = 2]) = \mathbb{P}([X_2 = 3]) = \mathbb{P}([X_2 = 4]) = \frac{2}{9}$$

3. Pour tout entier n supérieur ou égal à 2, donner, en justifiant, l'ensemble des valeurs prises par X_n.

4. ***a)*** Utiliser la formule des probabilités totales pour établir que, pour tout entier naturel n supérieur ou égal à 2, on a :

$$\mathbb{P}([X_{n+1} = 1]) = \frac{1}{3}\left(\mathbb{P}([X_n = 2]) + \mathbb{P}([X_n = 3]) + \mathbb{P}([X_n = 4])\right)$$

b) Vérifier que cette relation reste valable pour $n = 0$ et $n = 1$.

c) Justifier que, pour tout n de $\mathbb{N}$, on a $\mathbb{P}([X_n = 1]) + \mathbb{P}([X_n = 2]) + \mathbb{P}([X_n = 3]) + \mathbb{P}([X_n = 4]) = 1$ et en déduire l'égalité :

$$\forall n \in \mathbb{N},\ \mathbb{P}\left([X_{n+1} = 1]\right) = -\frac{1}{3}\, \mathbb{P}\left([X_n = 1]\right) + \frac{1}{3}$$

d) Établir alors que : $\forall n \in \mathbb{N},\ \mathbb{P}\left([X_n = 1]\right) = \dfrac{1}{4} + \dfrac{3}{4}\left(-\dfrac{1}{3}\right)^n$.

5. a) En procédant de la même façon qu'à la question précédente, montrer que l'on a :

$$\forall n \in \mathbb{N},\ \mathbb{P}\left([X_{n+1} = 2]\right) = \frac{1}{3}\ \left(\mathbb{P}\left([X_n = 1]\right) + \mathbb{P}\left([X_n = 3]\right) + \mathbb{P}\left([X_n = 4]\right)\right)$$

b) En déduire une relation entre $\mathbb{P}\left([X_{n+1} = 2]\right)$ et $\mathbb{P}\left([X_n = 2]\right)$.

c) Montrer enfin que : $\forall n \in \mathbb{N}, \mathbb{P}\left([X_n = 2]\right) = \dfrac{1}{4} - \dfrac{1}{4}\ \left(-\dfrac{1}{3}\right)^n$.

6. On admet que, pour tout entier naturel n, on a :

$$\mathbb{P}\left([X_{n+1} = 3]\right) = -\frac{1}{3}\, \mathbb{P}\left([X_n = 3]\right) + \frac{1}{3} \quad \text{et} \quad \mathbb{P}\left([X_{n+1} = 4]\right) = -\frac{1}{3}\, \mathbb{P}\left([X_n = 4]\right) + \frac{1}{3}$$

En déduire sans calcul que :

$$\forall n \in \mathbb{N},\ \mathbb{P}\left([X_n = 3]\right) = \mathbb{P}\left([X_n = 4]\right) = \frac{1}{4} - \frac{1}{4}\left(-\frac{1}{3}\right)^n$$

7. Déterminer, pour tout entier naturel n, l'espérance $\mathbb{E}(X_n)$ de la variable aléatoire X_n.

8. Montrer que la suite (X_n) converge en loi vers une variable aléatoire X dont on reconnaîtra la loi. En déduire un état stable de la chaîne de Markov.

Partie 2 : calcul des puissances d'une matrice A

Pour tout n de $\mathbb{N}$, on considère la matrice-ligne de $\mathscr{M}_{1,4}(\mathbb{R})$:

$$U_n = \begin{pmatrix} \mathbb{P}\left([X_n = 1]\right) & \mathbb{P}\left([X_n = 2]\right) & \mathbb{P}\left([X_n = 3]\right) & \mathbb{P}\left([X_n = 4]\right) \end{pmatrix}$$

U_n est le n^{e} état de la chaîne de Markov.

9. a) Montrer (grâce à certains résultats de la partie 1) que, si l'on pose $A = \dfrac{1}{3}\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$, on a :

$$\forall n \in \mathbb{N},\ U_{n+1} = U_n\ A$$

b) Établir par récurrence que : $\forall n \in \mathbb{N},\ U_n = U_0\ A^n$.

c) En déduire la première ligne de A^n.

d) Compléter le script suivant pour qu'il affiche le n^{e} état de la chaîne de Markov en utilisant la matrice A. On pourra utiliser les fonctions `al.matrix_power` et `np.dot`.

```
n = int(input('Rentrez la valeur de n :'))
A = (1/3) * ________________________________
U = _________________________
print(________________________________________)
```

10. Expliquer comment choisir la position du mobile au départ pour trouver les trois autres lignes de la matrice A^n, puis écrire ces trois lignes.

Partie 3 : une deuxième méthode de calcul des puissances de A

On considère les matrices I et J suivantes : $I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ et $J = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

11. Déterminer les réels a et b tels que $A = aI + bJ$.

12. ***a)*** Calculer J^2 puis établir que, pour tout entier naturel k non nul, on a : $J^k = 4^{k-1}J$.

b) À l'aide de la formule du binôme de Newton, en déduire, pour tout entier n non nul, l'expression de A^n comme combinaison linéaire de I et J.

c) Vérifier que l'expression trouvée reste valable pour $n = 0$.

Partie 4 : informatique

13. ***a)*** Compléter la fonction **Python** suivante pour qu'elle :

- prenne en argument la position `x` à un instant donné
- renvoie la position du mobile à l'instant suivant

On rappelle que `rd.randint(a,b)` simule un tirage uniforme dans $[\![a, b-1]\!]$.

```
def EtapeMarkov(x):
L = [1,2,3,4]
L.pop(x-1) # supprime le sommet x de la liste
i = ____________________
return L[i]
```

b) Compléter le script **Python** suivant pour qu'il affiche les 100 premières positions autres que celle d'origine du mobile dont le voyage est étudié dans ce problème, ainsi que le nombre n de fois où il est revenu sur le sommet numéroté 1 au cours de ses 100 premiers déplacements.

```
chaine = []
x = _____ # point de départ
for k in range(100):
    x = ____________________
    chaine.append(x)
print(chaine)
cpt = 0
for k in range(100):
    if ____________________:
        cpt = __________
print(cpt)
```

c) Après avoir exécuté cinq fois ce script, les réponses concernant le nombre de fois où le mobile est revenu sur le sommet 1 sont : $n = 23$, $n = 28$, $n = 23$, $n = 25$, $n = 26$.
En quoi est-ce normal ?