ECRICOME 2020

Exercice 1

Dans cet exercice, on désigne par $\mathscr{M}_3(\mathbb{R})$ l'ensemble des matrices réelles carrées d'ordre 3, et on note I_3 la matrice identité de $\mathscr{M}_3(\mathbb{R})$.

Soit a un réel ; on pose $M = \begin{pmatrix} 2 & a-1 & -1 \\ 1-a & a & a-1 \\ 1 & a-1 & 0 \end{pmatrix}$.

Partie A : Étude du cas où $a = 1$

Dans toute cette partie, on suppose que $a = 1$.

1. Expliciter la matrice M, puis calculer $(M - I_3)^2$.

2. En déduire l'unique valeur propre possible de M.

3. La matrice M est-elle inversible ? La matrice M est-elle diagonalisable ?

Partie B : Étude du cas où $a = 0$

Dans cette partie, on suppose que $a = 0$.

4. Démontrer que 1 est une valeur propre de M, et donner une base et la dimension du sous-espace propre associé.

5. Démontrer que M n'est pas inversible.

6. En utilisant les deux questions précédentes, déterminer l'ensemble des valeurs propres de M, et la dimension des sous-espaces propres associés. La matrice M est-elle diagonalisable ?

Partie C : Étude du cas où a est différent de 0 et de 1

Dans cette partie, on suppose que a est différent de 0 et de 1.
On pose $E = \mathbb{R}^3$, et $\mathscr{B} = (e_1, e_2, e_3)$ la base canonique de E.
Soit f l'endomorphisme de E dont la matrice représentative dans la base $\mathscr{B}$ est la matrice M.
Soit $u = (1,1,1)$, $v = (1,0,1)$ et $w = (1,1,0)$.

7. Démontrer que la famille $\mathscr{B}' = (u, v, w)$ est une base de E.

8. Calculer $f(u)$, $f(v)$.

9. Calculer $f(w)$ et trouver deux réels α et β tels que $f(w) = \alpha v + \beta w$.

10. Déterminer la matrice représentative de f dans la base $\mathscr{B}'$, que l'on notera T.

11. En déduire l'ensemble des valeurs propres de M, et la dimension des sous-espaces propres associés. La matrice M est-elle diagonalisable ?

Exercice 2

Pour tout entier naturel non nul, on définit la fonction f_n sur $\mathbb{R}_+$ par :

$$\forall x \geqslant 0,\ f_n(x) = \int_0^x \frac{t^{2n}-1}{t+1}\, dt$$

Partie A : Étude de la fonction f_n

Dans cette partie, on fixe un entier naturel n non nul.

1. Démontrer que la fonction f_n est de classe $\mathcal{C}^1$ sur $\mathbb{R}_+$ et :

$$\forall x \geqslant 0,\ f_n'(x) = \frac{x^{2n}-1}{x+1}$$

2. Étudier les variations de f_n.

3. Démontrer que f_n est de classe $\mathcal{C}^2$ sur $\mathbb{R}_+$, et calculer sa dérivée seconde.
En déduire que f_n est convexe sur $\mathbb{R}_+$.

4. ***a)*** Démontrer : $\forall t \geqslant 1,\ t^{2n}-1 \ \geqslant\ n\,(t^2-1)$.

b) Montrer alors : $\forall x \geqslant 1,\ f_n(x) \ \geqslant\ f_n(1) + \dfrac{n}{2}\,(x-1)^2$.

c) En déduire la limite de $f_n(x)$ lorsque x tend vers $+\infty$.

5. Calculer $f_n(0)$, puis démontrer : $f_n(1) < 0$.

6. Démontrer que l'équation $f_n(x) = 0$ admet une unique solution strictement positive, et que cette solution est strictement supérieure à 1.
On note x_n cette solution.

Partie B : Étude d'une suite implicite

On étudie dans cette partie le comportement de la suite (x_n), où pour tout entier naturel n non nul, x_n est l'unique solution strictement positive de l'équation : $f_n(x) = 0$.
On admettra :

$$\forall n \in \mathbb{N}^*,\ x_n \geqslant \frac{2n+2}{2n+1}$$

7. Soit $x \in \mathbb{R}_+$. Démontrer :

$$\forall n \in \mathbb{N}^*,\ f_{n+1}(x) - f_n(x) \ =\ x^{2n+1}\ \left(\frac{x}{2n+2} - \frac{1}{2n+1}\right)$$

8. ***a)*** Montrer : $\forall n \in \mathbb{N}^*$, $\forall x \geqslant \dfrac{2n+2}{2n+1}$, $f_{n+1}(x) \geqslant f_n(x)$.

b) En déduire : $\forall n \in \mathbb{N}^*$, $f_{n+1}(x_n) \geqslant 0$.

c) Montrer alors que la suite (x_n) est décroissante, puis qu'elle est convergente.

9. ***a)*** Démontrer que pour tout entier $n \geqslant 1$: $-\ln(2) \leqslant f_n(1) \leqslant 0$.

b) À l'aide de l'inégalité démontrée à la question ***4.b)*** de la partie ***A***, montrer alors :

$$\forall n \in \mathbb{N}^*,\ 0 \leqslant x_n - 1 \leqslant \sqrt{\frac{2\ \ln(2)}{n}}$$

Quelle est la limite de x_n lorsque n tend vers $+\infty$?

Partie C : Étude d'une fonction de deux variables

Dans cette partie, on fixe à nouveau un entier naturel n non nul.
L'objectif de cette partie est d'étudier la fonction G_n définie sur $\mathbb{R}_+^* \times \mathbb{R}_+^*$ par :

$$G_n : (x, y) \mapsto f_n(x) \times f_n(y)$$

10. Justifier que la fonction G_n est de classe $\mathcal{C}^2$ sur $\mathbb{R}_+^* \times \mathbb{R}_+^*$ et calculer ses dérivées partielles premières.

11. Déterminer l'ensemble des points critiques de G_n.

12. Calculer la matrice hessienne de G_n au point (x_n, x_n) puis au point $(1, 1)$.

13. La fonction G_n admet-elle un extremum local en (x_n, x_n) ? Si oui, donner la nature de cet extremum.

14. La fonction G_n admet-elle un extremum local en $(1, 1)$? Si oui, donner la nature de cet extremum.

Exercice 3

Soit a un réel strictement positif.

1. Pour tout entier n supérieur ou égal à 2, on pose :

$$I_n(a) \ = \ \int_a^{+\infty} \frac{1}{t^n} \, dt$$

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 d'en-tête `def simulX(a, m, n)` d'arguments $a \in \mathbb{R}$, $m \in \mathbb{N}^*$ et $n \in \mathbb{N}^*$, 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 pourra utiliser l'instruction `rand(m, n)` du module `numpy.random` qui 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 indépendante.
On pourra aussi utiliser l'instruction `pow(c,d)` qui renvoie c^d pour $c > 0$ et $d \in R$.

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.a)*** 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)
for k in range(N):
    if .......... :
        s1 = s1 + 1
        if X[0,k] > 6*a:
            .........
if s1 > 0:
    print(.......)
```

On cherche dans la suite de l'exercice à estimer le paramètre a.
Soit n un entier naturel non nul, et $X_1, \ldots, X_n$ 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) Calculer son risque quadratique et vérifier que celui-ci vaut $\dfrac{a^2}{3n}$.

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) Calculer le risque quadratique de $\lambda_n W_n$ et vérifier que celui-ci vaut $\dfrac{a^2}{3n\,(3n-2)}$.

7. On rappelle que :

- si `A` est une matrice Python, l'instruction `A[i, :]` renvoie la $i^{\text{ème}}$ ligne de la matrice `A`, les lignes et les colonnes de `A` étant numérotées à partir de l'indice zéro.
- si `A` est une matrice Python, l'instruction `sum(A)` renvoie la somme des coefficients de `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'une matrice ligne à m éléments :

```
import numpy as np
def simulV(a,m,n):
    X=simulX(a,m,n)
    V=np.zeros(shape=(1,m))
    for k ..........:
        V[0,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 :

```
import matplotlib.pyplot as plt
W=simulW(...,...,...)
V=simulV(...,...,...)
abscisse=np.array([[k for k in range(...,...)]])
plt.plot(abscisse,...,"x")
plt.plot(abscisse,...,"+")
```

On justifiera la réponse pour les deux dernière lignes.