Révisions 6

Sujet ECRICOME

Exercice 1 (ECRICOME 2023)

Partie 1

On considère la matrice $A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$. On note f l'endomorphisme de $\mathbb{R}^4$ représenté par la matrice A dans la base canonique $\mathcal{C} = (e_1, e_2, e_3, e_4)$ de $\mathbb{R}^4$.

1. On considère les vecteurs suivants de $\mathbb{R}^4$:

$$u_1 = (-1, 1, 0, 1), \quad u_2 = (0, -1, 1, 0), \quad u_3 = (0, 1, 1, 0) \quad u_4 = (1, 0, 0, 1).$$

On note $\mathcal{B} = (u_1, u_2, u_3, u_4)$.

   (a) Montrer que $\mathcal{B}$ est une base de $\mathbb{R}^4$.

   (b) Déterminer la matrice représentative de f dans la base $\mathcal{B}$.

   (c) En déduire une matrice P de $\mathscr{M}_4(\mathbb{R})$ inversible et une matrice T de $\mathscr{M}_4(\mathbb{R})$ triangulaire telles que $A = PTP^{-1}$.

2. (a) Calculer A^2, A^3, puis vérifier que $A^3 = 4A^2 - 4A$.

   (b) Montrer par récurrence que, pour tout entier naturel n non nul, il existe deux réels a_n et b_n tels que

$$A^n = a_n A^2 + b_n A$$

   vérifiant, pour tout entier naturel n non nul, $a_{n+1} = 4a_n + b_n$ et $b_{n+1} = -4a_n$.

3. (a) Montrer que, pour tout entier naturel n non nul,

$$a_{n+2} = 4a_{n+1} - 4a_n.$$

   (b) Déterminer, pour tout entier naturel n non nul, une expression de a_n en fonction de n.

   (c) En déduire, pour tout entier naturel n non nul, une expression de b_n en fonction de n.

4. Montrer que, pour tout entier naturel n non nul,

$$A^n = \begin{pmatrix} 2^{n-1} & 0 & 0 & 2^{n-1} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ (n+1)2^{n-2} & 2^{n-1} & 2^{n-1} & (n-1)2^{n-2} \\ 2^{n-1} & 0 & 0 & 2^{n-1} \end{pmatrix}$$

Partie 2

Soient p un entier naturel non nul et G un graphe non pondéré orienté à p sommets. On note s_0, $s_1, \ldots, s_{p-1}$ les sommets de G.

5. (a) Rappeler la définition de la matrice d'adjacence du graphe G.

   (b) Soient n un entier naturel non nul, i un entier de $[\![1, p]\!]$ et j un entier de $[\![1, p]\!]$.
   Rappeler sans justification l'interprétation du coefficient situé à la ligne i et à la colonne j dans la matrice M^n, où M est la matrice d'adjacence du graphe G.

6. Dans cette question uniquement, on suppose que $p = 4$ et que la matrice d'adjacence du graphe G est la matrice

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

étudiée dans la partie 1.

(a) Représenter les sommets et les arêtes du graphe G sous forme d'un diagramme.

(b) Le graphe G est-il connexe ? Justifier votre réponse.

(c) Soit n un entier naturel non nul.
Déterminer le nombre de chemins de longueur n menant du sommet s_3 au sommet s_0.

7. Dans cette question et les suivantes, on revient au cas général décrit au début de la partie 2.

Soit s un sommet de G. On dit que le sommet t est un voisin de s quand (s, t) est une arête du graphe.

Comme le graphe est orienté, si t est un voisin de s, alors s n'est pas forcément un voisin de t.

On appelle liste d'adjacence du graphe G, une liste de p sous-listes telle que, pour tout entier k de $[\![0, p-1]\!]$, la sous-liste située à la position k contient tous les numéros des sommets voisins de s_k.

Par exemple, la liste d'adjacence du graphe étudié à la question 6 est :

```
L = [[0,3], [0,1,2], [0,1,2], [0,3]]
```

Écrire une fonction en langage Python, nommée `matrice_vers_ligne`, prenant en entrée la matrice d'adjacence `A` d'un graphe G (définie sous forme de listes de listes) et renvoyant la liste d'adjacence de G.

8. On cherche à écrire une fonction en langage Python permettant d'obtenir la longueur du plus court chemin menant d'un sommet de départ s_i à chaque sommet du graphe G.

On souhaite pour cela appliquer un algorithme faisant intervenir les variables suivantes :

- Une liste `distances` à p éléments, où l'élément situé à la position k sera égal, à la fin de l'algorithme, à la longueur du plus court chemin menant du sommet de départ s_i au sommet s_k.
- Une liste `a_explorer` contenant tous les sommets restant à traiter.
- Une liste `marques` contenant tous les sommets déjà traités.

Nous donnons ci-dessous la description de l'algorithme :

- Initialisation des trois listes décrites ci-dessus :
  - Initialement, chaque élément de la liste `distances` est égal à p, à l'exception du sommet s_i, auquel on affecte la distance 0.
  - La liste `marques` ne contient initialement que le numéro du sommet de départ s_i.
  - La liste `a_explorer` ne contient initialement que le numéro du sommet de départ s_i.
- Tant que la liste `a_explorer` n'est pas vide, on répète les opérations suivantes :
  - Nommer `s` le premier sommet de la liste `a_explorer`, et le retirer de cette liste.
  - Pour chaque voisin `v` du sommet `s` : si `v` n'est pas dans la liste `marques`, on l'ajoute à la fin de la liste `a_explorer`, et on lui affecte une distance égale à `distances[s]+1`.

(a) On considère le graphe orienté G étudié à la question 6.
Donner la valeur de la liste `distances` à l'issue de l'exécution de l'algorithme décrit ci-dessus, lorsqu'on l'applique au graphe G en choisissant s_1 comme sommet de départ.

(b) Recopier et compléter la fonction suivante, prenant en entrée la liste d'adjcacence `L` du graphe G et le numéro `i0` du sommet de départ s_i, et renvoyant la liste `distances` après exécution de l'algorithme décrit ci-dessus.

```
def parcours(L, i0):
    p = len(L)
    distances = .....
    distances[i0] = 0
    a_explorer = .....
    marques = .....
    while ..... :
        s = .....
        .....
        for v in ..... :
            if v not in marques :
                marques.append(v)
                .....
                .....
    return distances
```

(c) Modifier la fonction précédente pour qu'elle renvoie la liste de tous les sommets s pour lesquels il existe un chemin menant du sommet de départ s_i au sommet s.

Exercice 2 (ECRICOME 2020)

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

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

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 C^1 sur $\mathbb{R}_+$ et :
$$\forall x \geq 0,\ f_n'(x) = \frac{x^{2n}-1}{x+1}.$$

2. Étudier les variations de f_n.

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

   En déduire que f_n est convexe sur $\mathbb{R}_+$.

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

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

   (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 que :

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

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 : $\forall n \in \mathbb{N}^*, \forall x \geq \dfrac{2n+2}{2n+1}$, $f_{n+1}(x) \geq f_n(x)$.
   (b) En déduire : $\forall n \in \mathbb{N}^*$, $f_{n+1}(x_n) \geq 0$.
   (c) Montrer alors que la suite (x_n) est décroissante, puis qu'elle est convergente.

9. (a) Démontrer que pour tout entier $n \geq 1$: $-\ln(2) \leq f_n(1) \leq 0$.
   (b) À l'aide de l'inégalité démontrée à la question 4.(b) de la partie A, montrer alors :

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

   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 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 (ECRICOME 2022)

On dispose de trois urnes U_1, U_2 et U_3, et d'une infinité de jetons numérotés 1, 2, 3, 4, ...
On répartit un par un les jetons dans les urnes : pour chaque jeton, on choisit au hasard et avec équiprobabilité une des trois urnes dans laquelle on place le jeton. Le placement de chaque jeton est indépendant de tous les autres jetons, et la capacité des urnes en nombre de jetons n'est pas limitée.
Pour tout entier naturel n non nul, on note X_n (respectivement Y_n, Z_n) le nombre de jetons présents dans l'urne 1 (respectivement l'urne 2, l'urne 3) après avoir réparti les n premiers jetons.

Partie I

Pour tout entier naturel n non nul, on note V_n l'événement : "Après la répartition des n premiers jetons, au moins une urne reste vide".

1. Soit $n \in \mathbb{N}^*$.

   (a) Justifier que X_n, Y_n et Z_n suivent la même loi binomiale dont on précisera les paramètres.

   (b) Expliciter $P\big(X_n = 0\big)$ et $P\big(X_n = n\big)$.

   (c) Justifier : $(Y_n = 0) \cap (Z_n = 0) = (X_n = n)$.

   (d) Exprimer l'événement V_n à l'aide des événements $(X_n = 0)$, $(Y_n = 0)$ et $(Z_n = 0)$.

   (e) En déduire que : $P(V_n) = 3\left(\dfrac{2}{3}\right)^n - 3\left(\dfrac{1}{3}\right)^n$.

2. On note V l'événement : "Au moins l'une des trois urnes reste toujours vide".

   Exprimer l'événement V à l'aide des événements V_n, puis démontrer que $P(V) = 0$.

3. Soit T la variable aléatoire égale au nombre de jetons nécessaires pour que, pour la première fois, chaque urne contienne au moins un jeton.

   (a) On rappelle qu'en `Python` la commande `rd.randint(a, b)` renvoie un nombre aléatoire dans l'intervalle $[\![a, b-1]\!]$.

   Compléter la fonction `Python` ci-dessous pour qu'elle simule le placement des jetons jusqu'au moment où chaque urne contient au moins un jeton, et pour qu'elle renvoie la valeur prise par la variable aléatoire T.

```
def T():
    X = 0
    Y = 0
    Z = 0
    n = 0
    liste = [X, Y, Z]
    while ..... :
        i = rd.randint(0, 3) # choix d'un entier entre 0 et 2
        liste[i] = .....
        n = n+1
    return( ..... )
```

   (b) Écrire un script `Python` qui simule 10 000 fois la variable aléatoire T et qui renvoie une valeur approchée de son espérance (en supposant que cette espérance existe).

4. Déterminer $T(\Omega)$.

5. Démontrer que : $\forall n \in T(\Omega)$, $P\big(T = n\big) = P(V_{n-1}) - P(V_n)$.

6. Démontrer que la variable aléatoire T admet une espérance, et calculer cette espérance.

Partie II

Pour tout entier naturel n non nul, on note W_n la variable aléatoire égale au nombre d'urne(s) encore vide(s) après le placement des n premiers jetons.

7. (a) Donner la loi du couple (X_2, W_2).

   (b) En déduire la loi de W_2, et calculer son espérance.

   (c) Calculer la covariance de X_2 et W_2.

   (d) Les variables aléatoires X_2 et W_2 sont-elles indépendantes ?

Soit n un entier naturel supérieur ou égal à 3.

8. Déterminer $W_n(\Omega)$.

9. Pour $i \in [\![1,3]\!]$, on note $W_{n,i}$ la variable aléatoire égale à 1 si l'urne i est encore vide après le placement des n premiers jetons, et qui vaut 0 sinon.

   (a) Démontrer : $\forall i \in [\![1,3]\!]$, $E(W_{n,i}) = \left(\frac{2}{3}\right)^n$.

   (b) Exprimer la variable aléatoire W_n en fonction des variables aléatoires $W_{n,1}$, $W_{n,2}$ et $W_{n,3}$.

   (c) Exprimer alors $E(W_n)$ en fonction de n.

10. Démontrer : $P\Big((X_n = n) \cap (W_n = 2)\Big) = \left(\frac{1}{3}\right)^n$.

    Pour $k \in [\![1, n-1]\!]$, quelle est la valeur de $P\Big((X_n = k) \cap (W_n = 2)\Big)$?

11. Démontrer : $\forall k \in [\![1, n-1]\!]$, $P\Big((X_n = k) \cap (W_n = 1)\Big) = \frac{2}{3^n}\binom{n}{k}$.

    Que vaut $P\Big((X_n = n) \cap (W_n = 1)\Big)$?

12. Démontrer :

$$E(X_n\, W_n) = 2nP\Big((X_n = n) \cap (W_n = 2)\Big) + \sum_{k=1}^{n-1} kP\Big((X_n = k) \cap (W_n = 1)\Big)$$

13. Démontrer alors : $E(X_n\, W_n) = n\left(\frac{2}{3}\right)^n$. Puis calculer la covariance de X_n et W_n.

14. Interpréter le résultat obtenu à la question précédente.
