

1 Exemple 1

1.1 Question de cours

Énoncer l'inégalité de Cauchy-Schwarz dans $\mathbb{R}^n$.

1.2 Exercice préparé

On considère d'une part deux urnes A et B et d'autre part N boules numérotées de 1 à N .

On répartit initialement les boules entre les deux urnes. Puis on effectue une série illimitée d'étapes selon le protocole suivant : à chaque étape, on choisit aléatoirement et avec équiprobabilité un nombre entre 1 et N et on transfère la boule portant le numéro correspondant dans l'urne où elle n'était pas. On note X_0 la variable aléatoire égale au nombre de boules présentes initialement dans l'urne A et pour tout entier naturel n non nul, on note X_n la variable aléatoire égale au nombre de boules présentes dans l'urne A après n étapes.

1. On suppose qu'à l'issue d'une certaine étape, l'urne A contient k boules (avec $0 \leq k \leq N$). Quelle est alors la probabilité qu'à l'issue de l'étape suivante l'urne A contienne une boule de plus ? une boule de moins ?
2. Écrire une fonction Python nommée `etape` qui reçoit en argument le nombre N de boules et le nombre k de boules présentes dans l'urne A à l'issue d'une étape, qui simule l'étape suivante et renvoie le nombre de boules présentes dans l'urne A à l'issue de cette nouvelle étape.
3. Dans toute cette question, on étudie le cas particulier où $N = 3$.

Soient $M = \begin{pmatrix} 0 & 1/3 & 0 & 0 \\ 1 & 0 & 2/3 & 0 \\ 0 & 2/3 & 0 & 1 \\ 0 & 0 & 1/3 & 0 \end{pmatrix}$, et pour tout $n \in \mathbb{N}$, $U_n = \begin{pmatrix} P(X_n = 0) \\ P(X_n = 1) \\ P(X_n = 2) \\ P(X_n = 3) \end{pmatrix}$.

- (a) Démontrer que pour tout entier naturel n , $U_{n+1} = MU_n$.
- (b) Démontrer que 1 est valeur propre de M et donner une base du sous-espace propre associé.
- (c) On suppose dans cette question que X_0 suit la loi binomiale de paramètres 3 et $\frac{1}{2}$. Donner la loi de X_n pour tout entier naturel n .

A partir de maintenant et jusqu'à la fin de l'exercice, X_0 est la variable aléatoire certaine égale à 0, c'est-à-dire que l'urne A est initialement vide. On s'intéresse à la variable aléatoire D égale au nombre d'étapes nécessaires pour revenir à cet état initial.

4. Écrire une fonction Python nommée `EspD` qui prend en argument une valeur de N et qui permet d'estimer la valeur de l'espérance de D correspondante.
En utilisant cette fonction pour plusieurs valeurs de N , conjecturer une expression de l'espérance de D en fonction de N .

Dans tout le reste de l'exercice, on revient au cas où $N = 3$.

5. Calculer $P(D = 2)$ et $P(D = 4)$. Justifier que D ne peut prendre que des valeurs paires.

6. (a) Démontrer que : $\forall k \geq 2, (D = 2k) = \left(\bigcap_{i=1}^{k-1} [X_{2i} = 2] \right) \cap [X_{2k} = 0]$

- (b) Déterminer la loi de D .

7. Montrer que D admet une espérance que l'on déterminera. Vérifier la cohérence du résultat obtenu avec la conjecture.

1.3 Exercice non préparé

1. Écrire une fonction en langage Python nommée `premier_groupe(c)` qui prend en entrée une chaîne de caractères c , et renvoie le couple (k, a) où a est le premier caractère de c et k est le nombre de fois où a est présent consécutivement au début de c .

Par convention, `premier_groupe("")` renverra $(0, "")$.

Exemple :

```
premier_groupe("111221");  
(3, '1')
```

On considère la suite de chaînes de caractères $(c_n)_{n \in \mathbf{N}}$ définie récursivement de la manière suivante. Le terme initial est $c_0 = "1"$, puis chaque terme est obtenu en “lisant” le précédent.

Par exemple :

- 1 se lit “un 1” $\rightarrow 11$
- 11 se lit “deux 1” $\rightarrow 21$
- 21 se lit “un 2, un 1” $\rightarrow 1211$

Les premiers termes de la suite sont donc :

$$1 \rightarrow 11 \rightarrow 21 \rightarrow 1211 \rightarrow 111221 \rightarrow \dots$$

2. En utilisant la fonction précédente, écrire une fonction en langage Python nommée `suivant(c)` qui renvoie le terme suivant de la suite.

Exemple :

```
suivant("21");  
"1211"
```

2 Exemple 2

2.1 Question de cours

Donner une caractérisation de la propriété: « α est une racine du polynôme P ».

2.2 Exercice préparé

1

1. Soit f la fonction définie sur $]0, +\infty[$ par $\forall t \in]0, +\infty[, f(t) = t^{-\frac{1}{4}}$.
 - (a) Écrire un script en langage Python permettant de tracer la fonction f sur $]0, 01; 10[$.
 - (b) Dresser le tableau de variation de f en précisant les limites aux bornes.
 - (c) Montrer que la fonction $\varphi : x \mapsto \int_x^{x+1} f(t) dt$ est décroissante sur $]0, +\infty[$. En déduire:

$$\forall k \in \mathbb{N} \setminus \{0, 1\}, \frac{\varphi(2k) + \varphi(2k-1)}{2} \leq \varphi(2k-1) \leq \frac{\varphi(2k-1) + \varphi(2k-2)}{2}$$

- (d) En déduire que pour tout entier naturel k supérieur ou égal à 2,

$$\frac{(2k+1)^{3/4} - (2k-1)^{3/4}}{2} \leq (2k)^{3/4} - (2k-1)^{3/4} \leq \frac{(2k)^{3/4} - (2k-2)^{3/4}}{2}$$

Vérifier que cet encadrement est encore valable pour $k = 1$.

On définit les suites $(s_n)_{n \geq 1}$ et $(u_n)_{n \geq 1}$ par $\forall n \in \mathbb{N}^*, s_n = \sum_{k=1}^n (-1)^k k^{3/4}, u_n = \frac{s_{2n}}{(2n)^{3/4}}$.

2. (a) La série $\sum_{k \geq 1} (-1)^k k^{3/4}$ est-elle absolument convergente? convergente?
 - (b) Que dire quant à la convergence de la suite (s_n) ?
3. (a) Écrire une fonction en langage Python nommée `Suite_s` qui prend en entrée un entier naturel non nul n et qui renvoie la liste des n premiers termes de la suite $(s_n)_{n \geq 1}$.
 - (b) En déduire une fonction en langage Python nommée `Trace_u` qui prend en entrée un entier naturel non nul n et qui trace les n premiers termes de la suite $(u_n)_{n \geq 1}$.
 - (c) Que peut on conjecturer sur la suite (u_n) ?
4. (a) Montrer que $\forall n \in \mathbb{N}^*, u_n = \frac{1}{(2n)^{3/4}} \sum_{k=1}^n \left((2k)^{3/4} - (2k-1)^{3/4} \right)$.
 - (b) Montrer que $\forall n \in \mathbb{N}^*, \frac{(2n+1)^{3/4} - 1}{2(2n)^{3/4}} \leq u_n \leq \frac{1}{2}$.
 - (c) Que dire quant à la convergence de la suite (u_n) ?
5. (a) Montrer que pour tout $n \in \mathbb{N}^*, \frac{s_{2n+1}}{(2n+1)^{3/4}} = \left(\frac{2n}{2n+1} \right)^{3/4} u_n - 1$.
 - (b) En déduire que $\lim_{n \rightarrow +\infty} \frac{s_{2n+1}}{(2n+1)^{3/4}} = -\frac{1}{2}$.
 - (c) Déterminer un équivalent de s_n quand $n \rightarrow +\infty$.

2.3 Exercice non préparé

1. Écrire une fonction en langage Python nommée `booléens` prenant en argument un entier naturel n supérieur ou égal à 2 et renvoyant la liste booléenne de longueur $n+1$ `[False, False, True, ..., True]` (seuls les deux premiers éléments sont `False`).
2. Écrire une fonction `modif` prenant en argument une liste de booléens `L` et un entier naturel s . Cette fonction
 - recherche i , premier indice supérieur ou égal à s d'apparition de `True` dans `L` (on supposera qu'un tel indice existe);
 - modifie la liste `L` en passant à `False` tous les éléments de `L` d'indice ki avec $k \in \mathbb{N}$ et $k \geq 2$;
 - renvoie $i + 1$.

Par exemple, quand `L = [False,False,True,True,True,True,True,True]` et $s = 2$, on a $i = 2$, la liste devient `L = [False,False,True,True,False,True,False,True]` et la valeur renvoyée est 3.

3 Exemple 3

3.1 Question de cours

Que dire de la somme de deux variables aléatoires indépendantes suivant une loi de Poisson?

3.2 Exercice préparé

Soit $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ la fonction définie par $\varphi : x \mapsto (x+1)e^{-x}$.

1. (a) À l'aide de Python, tracer la courbe de φ sur l'intervalle $[-2; 4]$.
(b) En déduire le tableau de variations et les limites de φ .
(c) Déterminer graphiquement, pour chaque réel k , le nombre d'antécédents de k par φ .
2. (a) Résoudre l'équation différentielle (E) $\forall x \in \mathbb{R}, y'(x) - 2y(x) = xe^x$.
(b) L'ensemble des solutions de (E) est-il un sous-espace vectoriel de l'ensemble des fonctions définies sur $\mathbb{R}$?

On s'intéresse maintenant à l'équation différentielle

$$(F) \quad \forall x \in \mathbb{R}, \quad y'(x) + 2y(x) + xe^x y(x)^2 = 0.$$

3. Déterminer les fonctions constantes solutions de (F).
4. Soit P un polynôme non nul de $\mathbb{R}[X]$. Supposons que $x \mapsto P(x)e^{-x}$ est une solution de (F).
(a) Montrer que pour tout $x \in \mathbb{R}$, $P'(x) + P(x) + xP(x)^2 = 0$.
(b) En utilisant la limite en $+\infty$, en déduire une absurdité. Qu'en conclut-on?
5. On admet dans cette question que (F) admet une solution f qui ne s'annule pas sur $\mathbb{R}$.
L'ensemble des solutions de (F) est-il un sous-espace vectoriel de l'ensemble des fonctions de classe $\mathcal{C}^1$ sur $\mathbb{R}$?
6. Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une solution de (F) telle que $f(x) \neq 0$ pour tout $x \in \mathbb{R}$.
Pour tout $x \in \mathbb{R}$, on pose $g(x) = \frac{1}{f(x)}$.
(a) Montrer que g est solution d'une équation différentielle linéaire que l'on déterminera.
(b) En déduire qu'il existe un réel $k \in]1; +\infty[$ tel que

$$\forall x \in \mathbb{R}, \quad f(x) = \frac{1}{ke^{2x} - xe^x - e^x}$$

7. Pour tout entier $k \geq 2$, on pose $h_k : x \mapsto ke^{2x} - xe^x - e^x$.
Dans l'espace vectoriel des fonctions de $\mathbb{R}$ dans $\mathbb{R}$,
(a) la famille $(h_2; h_3; h_4)$ est-elle libre?
(b) la famille $\left(\frac{1}{h_2}; \frac{1}{h_3}; \frac{1}{h_4}\right)$ est-elle libre?

3.3 Exercice non préparé

1. En python, la fonction `ord`, qui prend en argument un caractère, renvoie un nombre entier qui est le code de ce caractère. Par exemple, `ord('A')` renvoie 65, `ord('T')` renvoie 84, `ord('C')` renvoie 67, `ord('G')` renvoie 71, ...

Écrire une fonction `score`, qui prend pour paramètre une chaîne de caractères `motif`, et renvoie la somme des codes de ses caractères.

Par exemple, `code('TCG')` doit renvoyer 222.

2. Écrire une fonction `liste_scores` qui prend en paramètres une chaîne de caractères `texte` et un entier naturel `p` et qui renvoie la liste des scores des sous-chaînes de texte de longueur `p`.

Par exemple, pour le texte 'TGCAAGCATT' et `p=3`, la fonction renverra la liste composée du score de 'TGC': 222, puis le score de 'GCA': 203 , puis le score de 'CAA': 197 et ainsi de suite...

4 Exemple 4

4.1 Question de cours

Donner la définition d'une famille libre finie d'un espace vectoriel E .

4.2 Exercice préparé

Pour tout entier $n \in \mathbb{N}^*$ et tout entier $k \in \llbracket 0, n \rrbracket$, on note $B_{n,k}$ le polynôme de $\mathbb{R}_n[X]$ défini par :

$$B_{n,k} = \binom{n}{k} X^k (1 - X)^{n-k}$$

1. (a) Déterminer $B_{3,1}$.
(b) Étudier sur $\mathbb{R}$ la fonction $f : x \mapsto B_{3,1}(x)$ et dresser son tableau de variations.
2. En déduire, suivant les valeurs de $m \in \mathbb{R}$, le nombre de solutions sur $\mathbb{R}$ de l'équation $B_{3,1}(x) = m$.

On note $C_n = (1, X, \dots, X^n)$ la base canonique de $\mathbb{R}_n[X]$, et on considère T_n l'application définie sur $\mathbb{R}_n[X]$ et à valeurs dans $\mathbb{R}[X]$ telle que

$$\forall P \in \mathbb{R}_n[X], T_n(P) = \sum_{k=0}^n P\left(\frac{k}{n}\right) B_{n,k}$$

3. (a) Montrer que pour $n \in \mathbb{N}^*$ et tout entier $k \in \llbracket 0, n \rrbracket$, $B_{n,k} = \sum_{i=k}^n \binom{n}{k} \binom{n-k}{n-i} (-1)^{i-k} X^i$
(b) Programmer en Python une fonction `binom` qui prend en entrée deux entiers n et k et qui renvoie $\binom{n}{k}$.
(c) En Python, on décide d'encoder un polynôme en listant ses coefficients par ordre de degré croissant. Par exemple, on encode $P = -2X^3 + X + 2$ par `p=[2,1,0,-2]`.
Programmer une fonction Python qui prend en entrée deux entiers n et k et qui renvoie renvoie la liste modélisant $B_{n,k}$.
4. Dans cette question uniquement, on choisit $n = 2$.
 - (a) Déterminer la matrice K_2 de la famille $(B_{2,0}, B_{2,1}, B_{2,2})$ dans la base C_2 .
 - (b) En déduire que la famille $(B_{2,0}, B_{2,1}, B_{2,2})$ est une base de $\mathbb{R}_2[X]$.
 - (c) Calculer $T_2(1)$, $T_2(X)$ et $T_2(X^2)$, et en déduire la matrice H_2 de T_2 dans la base C_2 .
 - (d) Préciser les valeurs propres et les sous-espaces propres de T_2 .
5. On revient au cas général où n est un entier supérieur ou égal à 1 .
 - (a) Montrer que la famille $(B_{n,0}, B_{n,1}, \dots, B_{n,n})$ est une base de $\mathbb{R}_n[X]$.
 - (b) Montrer que l'application T_n est un endomorphisme bijectif de $\mathbb{R}_n[X]$.
 - (c) Calculer $T_n(1)$ et $T_n(X)$.
 - (d) Montrer que pour tout $k \in \llbracket 0, n \rrbracket$, le degré du polynôme $T_n(X^k)$ est égal à k . Pour établir ce résultat, on pourra utiliser la propriété suivante que l'on ne demande pas de démontrer :

$$\forall k \in \llbracket 0, n-1 \rrbracket, T_n(X^{k+1}) = \frac{1}{n} X(1-X) \left(T_n(X^k)\right)' + X \left(T_n(X^k)\right)$$

où $\left(T_n(X^k)\right)'$ est le polynôme dérivé de $T_n(X^k)$.

- (e) L'endomorphisme T_n est-il diagonalisable ?

4.3 Exercice non préparé

Un vote a été organisé dans une population $\{P_0, P_1, \dots, P_{n-1}\}$ de n personnes, chaque personne devant classer les m candidats $C_0, C_1, \dots, C_{m-1}$ dans l'ordre de ses préférences. Un bulletin de vote est une liste indiquant les préférences d'un votant: par exemple, quand $m = 4$, le bulletin de vote $B = [1, 0, 3, 2]$ signifie que la personne préfère C_1 , puis C_0 , puis C_3 et enfin C_2 .

1. Écrire une fonction **pref** qui prend pour paramètres un bulletin de vote B et deux entiers a et b distincts de $\{0, \dots, m-1\}$ et qui renvoie **True** si C_a est classé avant C_b dans le bulletin B et **False** sinon.

Ainsi, les instructions **pref**(`[2,1,3,0],1,2`) et **pref**(`[2,1,3,0],2,3`) renverront respectivement **False** et **True**.

Le résultat du vote est donné sous forme d'une liste L de longueur n : la case $L[i]$ contient le bulletin de vote de P_i . Quand $n = 6$ et $m = 4$, on peut avoir par exemple:

$$L = [\underbrace{[1, 3, 2, 0]}_{\text{vote de } P_0}, \underbrace{[2, 1, 3, 0]}_{\text{vote de } P_1}, \underbrace{[0, 1, 3, 2]}_{\text{vote de } P_2}, \underbrace{[3, 1, 2, 0]}_{\text{vote de } P_3}, \underbrace{[2, 1, 3, 0]}_{\text{vote de } P_4}, \underbrace{[2, 0, 1, 3]}_{\text{vote de } P_5}]$$

2. Si a et b sont deux éléments distincts de $\{0, 1, \dots, m-1\}$, on dit que le candidat C_a bat le candidat C_b si C_a est classé avant C_b dans au moins la moitié des bulletins de la liste L .
Par exemple, avec la liste L ci-dessus, C_3 bat C_0 , C_1 bat C_3 et C_2 bat C_0 .
Écrire une fonction **duel** qui prend pour paramètres une liste L de bulletins et deux entiers a et b , et qui renvoie **True** si C_a bat C_b et **False** sinon.

5 Exemple 5

5.1 Question de cours

Donner la définition d'un système complet d'événements.

5.2 Exercice préparé

1. On considère la matrice $A \in \mathcal{M}_3(\mathbb{R})$ définie par :

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1/8 & 0 & 1 \end{pmatrix}$$

- (a) Montrer que, pour tout réel λ , on a : λ est valeur propre de $A \iff \lambda^3 - \lambda^2 + \frac{1}{8} = 0$.
(b) Montrer que $1/2$ est valeur propre de A .
(c) Montrer que A admet deux autres valeurs propres a et b vérifiant :

$$a + b = \frac{1}{2} \quad \text{et} \quad ab = \frac{-1}{4}$$

- (d) La matrice A est-elle diagonalisable ?

On dispose d'une pièce équilibrée, c'est-à-dire qu'à chaque lancer, on obtient Pile ou Face avec probabilité $1/2$, indépendamment des autres lancers.

On procède à une succession (infinie) de lancers avec cette pièce. Lorsqu'on obtient deux Pile consécutifs dans cette succession de lancers, on dit que l'on a réalisé un *doublé*.

Par exemple, trois Piles consécutifs donnent lieu à deux doublés.

On note pour tout $n \geq 2$:

- A_n l'événement « on obtient pour la première fois un doublé sur les lancers $n - 1$ et n ».
- B_n l'événement « on obtient au moins un doublé sur les n premiers lancers de la pièce ».
- B l'événement « on obtient au moins un doublé ».

On pose $u_1 = 0$ et pour tout $n \geq 2$, $u_n = P(A_n)$.

2. (a) Écrire, en Python, une fonction `simul` prenant en entrée un entier naturel n , simulant n lancers de la pièce et renvoyant alors le nombre de doublés réalisés sur ces n lancers.
(b) Conjecturer à l'aide du programme précédent $\lim_{n \rightarrow +\infty} P(B_n)$.

3. (a) Justifier que pour $n \geq 1$, $P(B_n) = \sum_{k=1}^n u_k$.

- (b) Montrer alors que pour $n \geq 1$, $u_{n+3} = \left(1 - \sum_{k=1}^n u_k\right) \times \frac{1}{8}$.

- (c) En déduire que pour $n \geq 1$, $u_{n+3} = u_{n+2} - \frac{1}{8}u_n$

- (d) On note toujours a et b les deux valeurs propres de A différentes de $1/2$ définies en 1(c).

Montrer par récurrence que pour tout $n \geq 1$: $u_n = \frac{a^{n-1} - b^{n-1}}{4(a - b)}$

- (e) En déduire la limite de $P(B_n)$ et montrer que $P(B) = 1$.

4. On note $X = n$ si A_n se réalise, et $X = 0$ si B n'est pas réalisé.
La variable aléatoire X ainsi obtenue admet-elle une espérance ?

5.3 Exercice non préparé

Un brin d'ADN est composé d'une séquence de nucléotides. Ces nucléotides font partie d'une famille de quatre nucléotides possibles, que l'on note avec leur initiale A, T, C et G. On représente un brin d'ADN comme une liste dont les éléments sont des lettres appartenant à la liste $N = ['A', 'T', 'C', 'G']$.

1. Écrire une fonction `brin_adn` prenant en entrée un entier n et renvoyant une liste de longueur n dont les éléments sont des lettres tirées uniformément dans la liste N .

La molécule d'ADN est composée de deux brins enroulés l'un autour de l'autre, et dont les nucléotides se correspondent de la manière suivante : un nucléotide A est apparié à un nucléotide T , et un nucléotide C est apparié à un nucléotide G .

2. Écrire une fonction `brin_complementaire` prenant en entrée une liste représentant un brin d'ADN, et renvoyant la liste représentant le brin auquel il est apparié. Par exemple, on devra avoir :
`brin_complementaire(['C', 'A', 'T']) = ['G', 'T', 'A']`.