

MP

Mathématiques · Informatique

2022

Sous la coordination de

William AUFORT
professeur en CPGE
ancien élève de l'École Normale Supérieure (Lyon)

Florian METZGER
professeur en CPGE
ancien élève de l'École Normale Supérieure (Paris-Saclay)

Vincent PUYHAUBERT
professeur en CPGE
ancien élève de l'École Normale Supérieure (Paris-Saclay)

Par

Virgile ANDREANI
ENS Ulm

Hicham ASSAKAF
ENS Paris-Saclay

William AUFORT
professeur en CPGE

Antoine BARRIER
ENS Paris-Saclay

Jean-Paul BONNET
professeur en CPGE

Christophe FISZKA
professeur en CPGE

Julie GAUTHIER
professeur agrégé

Florian METZGER
professeur en CPGE

Benjamin MONMEGE
enseignant-chercheur à l'université

Angèle NICLAS
ENS Lyon

Vincent PUYHAUBERT
professeur en CPGE

Cyril RAVAT
professeur en CPGE

Jean STARYNKÉVITCH
professeur en CPGE

Sommaire thématique de mathématiques

2015 – 2022

e3a MP Maths (2021+2022)		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
e3a PC Maths (2021+2022)			•	•	•			•	•			•					•	
e3a PSI Maths (2021+2022)		•	•	•				•		•		•	•				•	
CCINP MP Maths 1	•	•	•		•	•	•	•	•	•	•	•	•		•	•	•	•
CCINP MP Maths 2	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•	•
CCINP PC Maths		•	•	•	•	•		•	•	•		•	•	•	•	•	•	
CCINP PSI Maths			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
Centrale MP Maths 1	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•		
Centrale MP Maths 2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
Centrale PC Maths 1	•	•	•	•	•	•				•		•	•	•	•	•	•	
Centrale PC Maths 2	•	•	•		•		•	•	•	•	•	•				•		
Centrale PSI Maths 1		•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•
Centrale PSI Maths 2		•	•	•	•			•	•	•	•	•	•	•	•	•	•	
Mines MP Maths 1	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Mines MP Maths 2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
Mines PC Maths 1		•	•	•	•			•	•	•	•	•	•	•	•	•	•	
Mines PC Maths 2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
Mines PSI Maths 1			•	•			•	•	•	•	•	•	•	•		•	•	
Mines PSI Maths 2		•	•	•	•	•	•	•	•		•	•	•	•		•		
X/ENS MP Maths A	•	•	•	•	•	•	•			•	•							
X/ENS MP Maths B	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	
X/ENS PC Maths		•	•	•	•	•	•				•	•	•		•	•		
X/ENS PSI Maths		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		
	Structures algébriques et arithmétique	Polynômes	Algèbre linéaire générale	Réduction des endomorphismes	Produit scalaire et espaces euclidiens	Topologie des espaces vectoriels normés	Suites et séries numériques	Suites et séries de fonctions	Séries entières	Analyse réelle	Intégration	Équations différentielles	Fonctions de plusieurs variables	Dénombrément et probabilités	Informatique tronc commun			

Sommaire

		Énoncé	Corrigé
E3A			
Mathématiques 1	Intégrale à paramètre, espaces euclidiens et matrices circulantes. <i>intégration, variables aléatoires, espaces euclidiens, polynômes, réduction</i>	19	25
CONCOURS COMMUN INP			
Mathématiques 1	Intégrales de Fresnel. <i>Python, probabilités discrètes, intégration, séries de fonctions</i>	46	51
Mathématiques 2	Déterminant de Vandermonde et exponentielles de matrices. <i>algèbre linéaire, espaces vectoriels normés, séries de fonctions, réduction</i>	67	72
Informatique optionnelle	Automates augmentés, tas binomiaux et arbre de Calkin-Wilf. <i>arbres, files de priorité, automates, programmation OCaml</i>	87	95
CENTRALE-SUPÉLEC			
Mathématiques 1	Exemples de contraintes symplectiques linéaires. <i>algèbre linéaire, espace vectoriels normés, réduction des endomorphismes</i>	113	118
Mathématiques 2	Étude de la dérivation de sommes de séries de fonctions. <i>analyse réelle, polynômes, suites et séries numériques et de fonctions, probabilités</i>	139	143
Informatique commune	Modélisations autour de la Formule 1. <i>Python, SQL, algèbre linéaire, équations différentielles</i>	163	171
Informatique optionnelle	Algorithmes de Brzozowski et de Conway, automate des dérivées d'Antimirov. <i>automates et expressions, programmation OCaml, complexité, diviser-pour-régner</i>	187	196

MINES-PONTS

Mathématiques 1	Formule asymptotique de Hardy et Ramanujan. <i>séries numériques, intégration, suites et séries de fonctions, polynômes</i>	226	234
Mathématiques 2	Autour des exponentielles de matrices. <i>algèbre linéaire, calcul différentiel, espaces vectoriels normés, équations différentielles</i>	258	265
Informatique commune	Modélisation numérique d'un matériau magnétique. <i>programmation Python, SQL, algorithmes de listes, récursivité, piles, complexité</i>	285	298
Informatique optionnelle	Correction d'erreurs dans des mots. <i>programmation dynamique, arbres, automates, programmation OCaml</i>	308	318

POLYTECHNIQUE-ENS

Mathématiques A	Formes p -linéaires alternées. <i>algèbre linéaire, espaces euclidiens, topologie, analyse réelle</i>	340	347
Mathématiques B	Convergence uniforme de suites de probabilités. <i>arithmétique, probabilités, suites et séries de fonctions, analyse réelle</i>	369	376
Informatique commune	Spéleo-logique. <i>algorithmique, complexité, listes, fonctions récursives, programmation Python</i>	404	414

FORMULAIRES

Développements limités usuels en 0	437
Développements en série entière usuels	438
Dérivées usuelles	439
Primitives usuelles	440
Trigonométrie	442

SESSION 2022



MP8M

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES

Durée : 4 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.
- Ne pas utiliser de correcteur.
- Écrire le mot FIN à la fin de votre composition.

Les calculatrices sont interdites.

Le sujet est composé de quatre exercices indépendants.

EXERCICE 1

1. Pour tout réel x , on pose, lorsque cela est possible, $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$.
 - 1.1. Déterminer l'ensemble de définition Δ de Γ .
 - 1.2. Démontrer que pour tout réel x de Δ , $\Gamma(x+1) = x\Gamma(x)$.
 - 1.3. On admet que $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$.
Calculer $\Gamma\left(n + \frac{1}{2}\right)$ pour tout entier naturel n . On exprimera le résultat à l'aide de factorielles.
2. Pour tout entier naturel n , on pose $I_n = \int_0^{+\infty} t^{2n} \exp(-t^2) dt$.
 - 2.1. Justifier l'existence de I_n .
 - 2.2. En utilisant la question 1. calculer I_n .
3. Pour tout réel x , on pose, lorsque cela est possible, $H(x) = \int_0^{+\infty} \cos(xt) \exp(-t^2) dt$.
 - 3.1. Donner le développement en série entière de la fonction $\cos$ au voisinage de 0 et préciser son domaine de validité.
 - 3.2. Justifier que H est définie sur $\mathbb{R}$ et l'exprimer à l'aide de fonctions usuelles.
On citera les théorèmes utilisés en s'assurant que toutes leurs hypothèses sont bien vérifiées.
4. On se propose de retrouver le résultat établi à la question 3.2. par une autre méthode.
 - 4.1. Démontrer que H est de classe C^1 sur $\mathbb{R}$.
 - 4.2. Montrer que H est solution d'une équation différentielle linéaire du premier ordre.
 - 4.3. Retrouver l'expression de H obtenue à la question 3.2.

EXERCICE 2

1. Questions de cours

- 1.1. Soit f une fonction continue sur le segment $[a, b]$.

Donner, sans démonstration, la limite quand n tend vers l'infini de l'expression :

$$\frac{b-a}{n} \sum_{k=0}^{n-1} f\left(a + k \frac{b-a}{n}\right).$$

- 1.2. Soit $m \in \mathbb{N}$. Déterminer en fonction de m la valeur de $\lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{j=0}^{n-1} \left(\frac{j}{n}\right)^m$.

- 1.3. Soit n un entier non nul. Donner, sans démonstration, l'espérance d'une variable aléatoire qui suit la loi uniforme sur $\llbracket 1, n \rrbracket$.

* * * * *

Soient k et n deux éléments de $\mathbb{N}^*$. On dispose de k urnes contenant chacune n boules numérotées de 1 à n .

On tire une boule au hasard de chaque urne et on désigne par X_n la variable aléatoire égale au plus grand des numéros obtenus. On suppose que les tirages sont indépendants les uns des autres.

2. Donner l'ensemble J des valeurs prises par X_n .
3. Soit $j \in J$. Évaluer $\mathbb{P}(X_n \leq j)$ et prouver que l'on a : $\mathbb{P}(X_n = j) = \frac{j^k - (j-1)^k}{n^k}$.
4. Démontrer que l'espérance $\mathbb{E}(X_n)$ de la variable aléatoire X_n peut s'écrire :

$$\mathbb{E}(X_n) = \sum_{j=0}^{n-1} \mathbb{P}(X_n > j).$$

5. Calculer $\mathbb{E}(X_n)$ et en donner un équivalent lorsque n tend vers l'infini.
6. Lorsque $k = 1$, reconnaître la loi de X_n et vérifier la cohérence du résultat obtenu à la question précédente.

EXERCICE 3

Soit E un espace euclidien muni d'un produit scalaire $(\cdot | \cdot)$ dont la norme est notée $\| \cdot \|$.

1. Questions de cours

- 1.1. Soient x et y deux vecteurs de E . Démontrer l'inégalité de Cauchy-Schwarz : $|(x|y)| \leq \|x\| \|y\|$.
On pourra utiliser la fonction $t \mapsto \|x + ty\|^2$.
- 1.2. Démontrer qu'on a l'égalité $|(x|y)| = \|x\| \|y\|$ si, et seulement si, les vecteurs x et y sont colinéaires.
- 1.3. On considère $E = \mathcal{M}_{n,1}(\mathbb{R})$ muni de sa base canonique et du produit scalaire canonique $(X|Y) = X^T Y$.

Écrire cette inégalité pour $X = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ et $Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$.

* * * * *

Pour toute la suite de l'exercice, on identifie $\mathbb{R}^n$ et $\mathcal{M}_{n,1}(\mathbb{R})$

Partie 1

Soit n un entier naturel supérieur ou égal à 2.

On note $B = \{X \in \mathbb{R}^n, \|X\| \leq 1\}$.

On considère l'application F de $\mathbb{R}^n$ vers $\mathbb{R}$ définie par :

$$\forall X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n, \quad F(X) = \sum_{\substack{1 \leq i, j \leq n \\ i \neq j}} x_i x_j$$

Par exemple, pour $n = 3$, on a $F(X) = x_1x_2 + x_1x_3 + x_2x_1 + x_2x_3 + x_3x_1 + x_3x_2 = 2(x_1x_2 + x_1x_3 + x_2x_3)$.

2. Exprimer alors $F(X)$ à l'aide de $S_1(n) = \sum_{i=1}^n x_i$ et de $S_2(n) = \sum_{i=1}^n x_i^2$.
3. Montrer que F possède un maximum sur B que l'on notera M .
4. Montrer en utilisant la question 1. que $M = n - 1$.
5. Déterminer tous les $X \in \mathbb{R}^n$ tels que $F(X) = M$.

Partie 2

On note $\mathcal{B} = (e_1, \dots, e_n)$ la base canonique orthonormale pour le produit scalaire $(X|Y) = X^T Y$ de $\mathbb{R}^n$.

Pour tout couple de vecteurs (X, Y) de $\mathbb{R}^n$ décomposés dans la base $\mathcal{B}$: $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ et $Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$, on pose :

$$\varphi(X, Y) = \frac{1}{2} \sum_{\substack{1 \leq i, j \leq n \\ i \neq j}} (x_i y_j + x_j y_i).$$

Par exemple, pour $n = 3$, on a $\varphi(X, Y) = x_1y_2 + x_1y_3 + x_2y_1 + x_2y_3 + x_3y_1 + x_3y_2$.

6. Pour tout $X \in \mathbb{R}^n$ exprimer $F(X)$ à l'aide de φ .
7. Écrire la matrice $A = (a_{ij}) \in \mathcal{M}_n(\mathbb{R})$ définie pour tout $(i, j) \in \llbracket 1, n \rrbracket^2$ par $a_{ij} = \varphi(e_i, e_j)$.
8. Justifier l'existence d'une base orthonormale $\mathcal{U} = (u_1, u_2, \dots, u_n)$ constituée de vecteurs propres de la matrice A .
9. Vérifier que pour tout couple de vecteurs (X, Y) de $(\mathbb{R}^n)^2$, on a $\varphi(X, Y) = Y^T A X = X^T A Y$.
10. Soit J la matrice de $\mathcal{M}_n(\mathbb{R})$ dont tous les éléments sont égaux à 1.
 - 10.1. Déterminer les valeurs propres de la matrice J .
 - 10.2. En déduire une matrice diagonale Δ semblable à la matrice A .
11. Donner l'expression de $\varphi(X, Y)$ en fonction des coordonnées de X et Y dans la base $\mathcal{U}$.
12. Retrouver alors le résultat établi à la question 4.

e3a Mathématiques MP 2022 — Corrigé

Ce corrigé est proposé par Jean Starynkévitch (professeur en CPGE) ; il a été relu par Antoine Barrier (ENS Paris-Saclay) et Benjamin Monmege (enseignant-chercheur à l'université).

Cette épreuve est composée de quatre exercices indépendants.

- Dans le premier, il faut appliquer les résultats du cours sur les intégrales à paramètre. Après des questions sur la fonction Gamma, on y calcule par deux méthodes l'intégrale

$$\int_0^{+\infty} \cos(xt) e^{-t^2} dt$$

- L'enjeu de l'exercice 2 est le calcul de la loi du maximum de k variables aléatoires indépendantes suivant la loi uniforme sur $\llbracket 1 ; n \rrbracket$ et d'obtenir le comportement asymptotique de l'espérance lorsque n tend vers l'infini. N'utilisant que des probabilités prenant un nombre fini de valeurs, ainsi que le théorème des sommes de Riemann, cet énoncé est accessible en première année.
- L'exercice 3 commence par la démonstration de l'inégalité de Cauchy-Schwarz, puis étudie le maximum d'une certaine fonction quadratique sur l'espace euclidien $\mathbb{R}^n$. Il est centré sur les espaces euclidiens.
- Le dernier exercice calcule les puissance d'une matrice circulante particulière. Il utilise le cours de MP de réduction des endomorphismes, ainsi que les polynômes à coefficients complexes.

Ce sujet comporte des questions faciles, et d'autres qui demandent de la réflexion. Il permettait de classer efficacement les candidats. Il sera utile pour préparer les concours CCINP et e3a, et présente l'avantage d'être très majoritairement abordable bien avant la période des révisions de fin de seconde année.

INDICATIONS

Exercice 1

- I.1.1 Justifier la continuité de la fonction intégrée. Pour l'intégrabilité au voisinage de $+\infty$, remarquer que la fonction à intégrer est négligeable devant $1/t^2$. Pour l'intégrabilité au voisinage de 0, comparer à une intégrale de Riemann.
- I.1.2 Intégrer par parties.
- I.1.3 Utiliser à n reprises la question I.1.2.
- I.2.1 Mêmes indications qu'en question I.1.1.
- I.2.2 Effectuer le changement de variable $t = \sqrt{u}$.
- I.3.2 Appliquer le théorème d'intégration terme à terme.
- I.4.1 Appliquer le théorème de dérivabilité sous le signe intégrale.
- I.4.2 Intégrer par parties.

Exercice 2

- II.1.1 Reconnaître le théorème des sommes de Riemann.
- II.1.2 Appliquer la question précédente avec $(a, b) = (0, 1)$ et $f(t) = t^m$.
- II.3 Pour $1 \leq i \leq k$, introduire U_i la variable aléatoire représentant le résultat de l'urne i , et exprimer l'événement $\{X_n \leq j\}$ à l'aide des $(\{U_i \leq j\})_{1 \leq i \leq k}$.
- II.4 Remarquer que l'événement $\{X_n > j\}$ est la réunion des événements deux à deux disjoints $\{X_n = i\}$ pour $i \in \llbracket j + 1 ; n \rrbracket$.
- II.5 On doit trouver $E(X_n) = n - \sum_{j=1}^{n-1} \frac{j^k}{n^k}$.
- II.6 Penser à la question II.1.3.

Exercice 3

- III.1.1 Remarquer que f est une fonction polynomiale de degré 2 de signe constant, afin d'en déduire le signe de son discriminant.
- III.2 Calculer $S_1(n)^2$.
- III.3 C'est une application directe du théorème des bornes atteintes.
- III.4 Utiliser les questions III.1.3 et III.2.
- III.5 Il y a une erreur d'énoncé : remplacer « $X \in \mathbb{R}^n$ » par « $X \in B$ ».
- III.7 On trouve que A est la matrice à diagonale nulle dont les coefficients non diagonaux valent tous 1.
- III.8 Remarquer que A est symétrique.
- III.10.1 Appliquer le théorème du rang à la matrice J .

Exercice 4

- IV.1 Utiliser que $z\bar{z} = |z|^2$.
- IV.2 Se souvenir que ω est racine n -ième de l'unité et utiliser la question IV.1.
- IV.3 S_n est une somme de termes en progression géométrique.
- IV.4.1 Introduire le polynôme $Q = \sum_{k=0}^n X^k$ et remarquer que $Q' = P$.
- IV.4.2 Appliquer la question IV.4.1, ainsi que le fait que ω^k est une racine n -ième de l'unité.
- IV.4.3 Factoriser dans $\mathbb{C}[X]$ le polynôme $X^n - 1$ de deux manières : d'une part en le scindant (trouver toutes ses racines, et justifier qu'elles sont simples), d'autre part en factorisant indépendamment de ce qui précède $X^n - 1$ par $X - 1$.
- IV.5.2 Montrer que la famille $(F^j)_{0 \leq j \leq n-1}$ est une famille libre et génératrice de G_F . Pour le caractère générateur, en utilisant que $F^n = I_n$, montrer que pour tout entier $k \in \mathbb{N}$, $F^k \in \{F_j \mid 0 \leq j \leq n-1\}$.
- IV.5.3 Utiliser une division euclidienne et le résultat de la question IV.5.2.
- IV.5.4 Utiliser la question IV.5.3.
- IV.6.1 On trouve que A est une matrice circulante de première ligne $(1 \quad 2 \quad \cdots \quad n)$.
- IV.6.2 Utiliser la question IV.5.4.
- IV.7 Utiliser la question IV.6.2.
- IV.8 Trouver un polynôme annulateur de A n'ayant pas 0 pour racine, afin de l'écrire sous la forme $K(1 - XQ)$ pour $K \neq 0$.
- IV.9 Prouver $G_A \subset G_F$ et utiliser un argument de dimension.
- IV.10 Utiliser la question IV.4.1.
- IV.11 Utiliser les questions IV.8, IV.9, IV.5 dans un premier temps, puis la question IV.10, puis déterminer $\text{Tr}(A^{-1})$ avec la question IV.6.1. On trouve

$$A^{-1} = \frac{1}{n^2(n+1)} \left(-(n^2 + n - 2)I_n + (n^2 + n + 2)F + 2 \sum_{k=2}^{n-1} F^k \right)$$

EXERCICE 1

I.1.1 Fixons $x \in \mathbb{R}$. Remarquons que la fonction $g_x : t \mapsto t^{x-1}e^{-t}$ est continue par morceaux (car continue) et positive sur $\mathbb{R}_+^*$. Ainsi, l'intégrale définissant $\Gamma(x)$ est convergente si et seulement si la fonction g_x est intégrable sur $\mathbb{R}_+^*$. De plus,

- La fonction g_x est continue (et positive) sur $]0; +\infty[$, donc intégrable sur tout segment de $]0; +\infty[$.

• On a
$$g_x(t) \underset{t \rightarrow 0^+}{\sim} \frac{1}{t^{1-x}}$$

L'intégrale de la fonction positive $t \mapsto 1/t^{1-x}$, intégrale de Riemann, est convergente au voisinage à droite de 0 si, et seulement si $1-x < 1$, c'est-à-dire $x > 0$. Il s'ensuit par comparaison que g_x est intégrable au voisinage à droite de 0 si et seulement si $x > 0$.

- Par croissances comparées, on a

$$t^2 g_x(t) = t^{x+1} e^{-t} \xrightarrow[t \rightarrow +\infty]{} 0$$

Ainsi,
$$g_x(t) = \underset{t \rightarrow +\infty}{o} \left(\frac{1}{t^2} \right)$$

Par comparaison à une intégrale de Riemann au voisinage de $+\infty$, g_x est intégrable au voisinage de $+\infty$, ceci pour tout x réel.

On conclut que l'intégrale définissant $\Gamma(x)$ converge si et seulement si $x > 0$.

Le domaine de définition de Γ est $\Delta = \mathbb{R}_+^*$.

I.1.2 Soit $x \in \Delta$. Alors $x > 0$ donc $x+1 > 0$, et ainsi $\Gamma(x+1)$ existe. Observons alors que

- la fonction $t \mapsto t^x$ est de classe $\mathcal{C}^1$ sur $\mathbb{R}_+^*$, de dérivée $t \mapsto xt^{x-1}$;
 - la fonction $t \mapsto e^{-t}$ est dérivable sur $\mathbb{R}_+^*$, de dérivée $t \mapsto -e^{-t}$;
 - puisque $x > 0$, on a $t^x e^{-t} \xrightarrow[t \rightarrow 0^+]{} 0$ et, par croissance comparée, $t^x e^{-t} \xrightarrow[t \rightarrow +\infty]{} 0$.
- Par conséquent, $[-t^x e^{-t}]_0^{+\infty}$ existe et est nul.

Le théorème d'intégration par parties sur un intervalle ouvert entraîne que

$$\begin{aligned} \Gamma(x+1) &= \int_0^{+\infty} t^x e^{-t} dt \\ &= [-t^x e^{-t}]_0^{+\infty} - \int_0^{+\infty} xt^{x-1}(-e^{-t}) dt \\ \Gamma(x+1) &= x \Gamma(x) \end{aligned}$$

Ainsi,

$\forall x \in \Delta \quad \Gamma(x+1) = x \Gamma(x)$

CCINP Maths 1 MP 2022 — Corrigé

Ce corrigé est proposé par Christophe Fiszka (professeur en CPGE) ; il a été relu par Julie Gauthier (professeur agrégé) et Tristan Poullaouec (professeur en CPGE).

Le sujet est composé d'un exercice et d'un problème. L'exercice étudie deux stratégies de recherche d'un code à quatre chiffres. Le problème, composé de trois parties, propose l'étude et le calcul d'intégrales introduites par le physicien français Augustin Fresnel (1788-1827) dans son travail sur le caractère ondulatoire de la lumière.

- L'exercice commence par des résultats sur des lois de probabilité discrètes (uniforme et géométrique) et se termine sur quelques programmes Python assez rudimentaires.
- La première partie du problème permet de justifier la convergence de l'intégrale généralisée

$$\int_0^{+\infty} e^{it^2} dt$$

Cette partie se termine par deux questions d'informatique autour de l'approximation des intégrales par la méthode des rectangles.

- La deuxième partie utilise une intégrale à paramètre pour le calcul des intégrales de Fresnel

$$\int_0^{+\infty} e^{it^2} dt, \quad \int_0^{+\infty} \cos(t^2) dt \quad \text{et} \quad \int_0^{+\infty} \sin(t^2) dt$$

- Enfin, dans la troisième et dernière partie du problème, on s'intéresse à une série de fonctions en lien avec les intégrales étudiées dans les parties précédentes.

L'exercice, assez court, ne présente pas de réelle difficulté. Il permet notamment de tester ses connaissances sur les fondamentaux des probabilités et de l'informatique. Le problème, plus intéressant et relativement facile, permet d'évaluer sa maîtrise de l'étude des intégrales généralisées, des séries et des techniques de calcul. Ce sujet est donc finalement très abordable par son niveau et par sa longueur. Il pouvait être traité dans son intégralité le jour de l'épreuve.

INDICATIONS

Exercice

- 3 Justifier que X suit une loi uniforme. Préciser le support.
- 4 Reconnaître une loi géométrique sur $\mathbb{N}^*$.

Problème

Partie I

- 8 Prouver que H est impaire à l'aide du changement de variable $u = -t$.
- 10 Effectuer un changement de variable. Attention, la nouvelle expression de H est une intégrale généralisée en 0.
- 11 Utiliser une intégration par parties.
- 12 Justifier la convergence absolue de $\int_{2\pi}^{+\infty} \frac{e^{iu}}{u^{\frac{3}{2}}} du$ par comparaison. Puis utiliser la question précédente.
- 13 Pour une fonction $f : [a; b] \rightarrow \mathbb{R}$, l'approximation de l'intégrale de f sur le segment $[a; b]$ donnée par la méthode des rectangles est

$$\forall n \in \mathbb{N}^* \quad S_n(f) = \frac{b-a}{n} \sum_{k=0}^{n-1} f\left(a + k \frac{b-a}{n}\right)$$

Partie II

- 15 Montrer que $|e^z| = e^{\operatorname{Re} z}$ pour $z \in \mathbb{C}$.
- 18 Appliquer le théorème de dérivabilité sous le signe intégrale sur tout intervalle du type $[a; b]$ avec $0 < a < b$ pour vérifier l'hypothèse de domination.
- 20 Déterminer $\alpha \in \mathbb{C}$ tel que $\alpha^2 = i$. La décomposition en éléments simples des fractions rationnelles justifie l'existence de $a, b \in \mathbb{C}$ tels que

$$\frac{1}{X^2 - i} = \frac{a}{X - \alpha} + \frac{b}{X + \alpha}$$

Pour le calcul des intégrales, déterminer les primitives à l'aide de Arctan .

- 21 Utiliser le résultat de la question 19.

Partie III

- 22 Utiliser le critère de majoration des séries à termes positifs pour justifier la convergence absolue de la série $\sum (a_n - a_{n+1}) b_n$.
- 23 Le membre de gauche est une série géométrique de raison e^{ix} . Utiliser les formules d'Euler et le principe de factorisation par l'angle moitié pour obtenir le membre de droite.
- 24 Appliquer le résultat de la question 22 avec la suite $(b_n)_{n \in \mathbb{N}}$ définie par $b_0 = 0$ et

$$\forall n \geq 1 \quad b_n = \sum_{k=1}^n e^{ikx}$$

- 26 Réaliser le changement de variable $u = xt$.

EXERCICE

1 D'après le cours, la fonction génératrice G_X d'une variable aléatoire X à valeurs dans $\mathbb{N}^*$ est définie par la somme de la série

$$\sum_{k \geq 1} P(X = k) t^k$$

Le rayon de convergence de cette série entière, noté R , est supérieur ou égal à 1. De plus, si X suit une loi géométrique de paramètre p sur $\mathbb{N}^*$

$$X(\Omega) = \mathbb{N}^* \quad \text{et} \quad \forall k \in \mathbb{N}^* \quad P(X = k) = p(1-p)^{k-1}$$

Dans ce cas, on obtient pour $t \in]-R; R[$

$$\begin{aligned} G_X(t) &= \sum_{k=1}^{+\infty} p(1-p)^{k-1} t^k \\ &= pt \sum_{k=1}^{+\infty} ((1-p)t)^{k-1} \\ G_X(t) &= pt \sum_{i=0}^{+\infty} ((1-p)t)^i \quad (i = k-1) \end{aligned}$$

On reconnaît l'expression d'une somme géométrique. Il y a convergence si et seulement si $|(1-p)t| < 1$, et dans ce cas

$$G_X(t) = pt \times \frac{1}{1 - (1-p)t} = \frac{pt}{1 - (1-p)t}$$

Précisons que le rayon de convergence est alors $R = 1/(1-p) > 1$. La fonction G_X est donc dérivable sur $] -R; R[$: en particulier G_X est dérivable en 1. La variable aléatoire X admet de ce fait une espérance qui vaut

$$E(X) = G_X'(1)$$

Or, pour tout $t \in] -R; R[$

$$G_X'(t) = \frac{p \times (1 - (1-p)t) + pt \times (1-p)}{(1 - (1-p)t)^2} = \frac{p}{(1 - (1-p)t)^2}$$

Finalement,

$$\boxed{\text{La variable } X \text{ admet une espérance et } E(X) = \frac{1}{p} .}$$

2 Soit $\Omega = \llbracket 0; 9 \rrbracket^4$ l'ensemble des codes possibles. Comme chaque code a la même probabilité d'être choisi, on est dans une situation d'équiprobabilité. La probabilité de tomber sur le seul bon code en un essai est ainsi

$$p = \frac{1}{\text{Card}(\Omega)} = \frac{1}{10^4}$$

3 Posons $N = 10^4 = \text{Card}(\Omega)$. Supposons que M. Toutlemonde choisisse un ordre sur les N codes possibles qu'il va essayer. Toutes les positions pour le bon code dans la liste sont possibles et équiprobables. La variable aléatoire X désignant le rang du code correct, il vient

$$\forall k \in \llbracket 1; N \rrbracket \quad P(X = k) = \frac{1}{N} = 10^{-4}$$

Ainsi,

La variable aléatoire X suit la loi uniforme sur $\llbracket 1; 10^4 \rrbracket$.

De ce fait,

$$E(X) = \frac{1 + 10^4}{2} \simeq 5 \cdot 10^3$$

4 La variable aléatoire X renvoie le rang du premier succès lors de la répétition d'une infinité d'expériences de Bernoulli mutuellement indépendantes de paramètre $p = 10^{-4}$. On sait alors que X suit une loi géométrique de paramètre p sur $\mathbb{N}^*$. D'après le résultat de la question 1

$$E(X) = \frac{1}{p} = 10^4$$

Si M. Toutlemonde s'arrange pour ne pas recomposer le même code, l'espérance est divisée par 2. Le résultat reste toutefois très grand. Par exemple, si M. Toutlemonde met seulement 5 secondes pour taper un code, il mettra en moyenne près de 3 jours à trouver le bon code. Autant dire que M. Toutlemonde va surtout rester dehors !

5 Voici un programme possible :

```
code=4714
n=int(input("Taper un code à 4 chiffres : "))
k=1
while n!=code :
    k+=1
    n=int(input("Ressayer : "))
print("Vous avez trouvé le code en "+str(k)+" essais")
```

6 Donnons une fonction à l'aide d'une boucle inconditionnelle `for` :

```
def crypte(m) :

    if len(m)!=4 :
        print("Erreur, l'argument doit être une 4-liste")
        # Un message d'erreur si le code n'a pas le
        # bon format
    else :
        M=[]
        for x in m:
            Nbre=(5+x)%10
            M.append(Nbre)
        return M
```

Donnons une seconde solution sous la forme d'une liste en compréhension :

```
def crypte(m) :
    if len(m)!=4 :
        # Un message d'erreur si le code n'a pas le bon format
        print("Erreur, l'argument doit être une 4-liste")
    else :
        return [(5+x)%10 for x in m]
```

CCINP Maths 2 MP 2022 — Corrigé

Ce corrigé est proposé par Julie Gauthier (professeur agrégée) ; il a été relu par Florian Metzger (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

Ce sujet se compose de deux exercices et d'un problème. Le premier exercice relève du cours de MPSI : il traite du déterminant de Vandermonde. Dans le deuxième exercice ainsi que dans le problème, on s'intéresse aux exponentielles de matrices. L'objectif du problème est de montrer que l'exponentielle terme à terme d'une matrice symétrique positive est une matrice symétrique positive.

- Dans l'exercice 1, on démontre la formule du déterminant de Vandermonde que l'on utilise pour calculer un déterminant, puis pour énoncer un résultat sur les sommes des puissances k -ièmes de complexes distincts.
- Dans l'exercice 2, on utilise une norme d'algèbre sur l'ensemble des matrices de taille n pour montrer la convergence de la série définissant l'exponentielle d'une matrice. On montre ensuite que l'application ainsi définie est continue. Enfin, on montre que l'exponentielle est différentiable en 0.
- Dans la partie I du problème, on traite d'abord un exemple de matrice symétrique en trouvant une condition nécessaire et suffisante pour qu'elle soit positive. Ensuite, on détermine son exponentielle. Enfin, on travaille sur l'exponentielle de matrices symétriques positives dans le cas général.
- Dans la partie II du problème, on trouve une condition nécessaire et suffisante plus générale pour qu'une matrice symétrique soit positive. On introduit le produit de Hadamard et on démontre que l'ensemble des matrices symétriques positives est stable par cette opération. Finalement on prouve que, si une matrice est symétrique positive, alors son exponentielle terme à terme l'est aussi.

Ce sujet mêle à la fois des outils d'algèbre et d'analyse des programmes de première et de deuxième année. Cela en fait un bon sujet de révision. La plupart des questions sont très abordables.

INDICATIONS

Exercice 1

- 1 Pour la deuxième partie de la question, montrer que si $x_k = x_\ell$ avec $k < \ell$ à la fois le déterminant de Vandermonde et le produit sont nuls.
- 2 Pour la récurrence, l'initialisation provient du résultat de la question 1.
- 3 Utiliser la formule démontrée dans la question 2.
- 4 Raisonner par l'absurde et utiliser la formule démontrée dans la question 2.

Exercice 2

- 5 Par récurrence, majorer la norme de A^k par la puissance k -ième de la norme de A (pour $k > 0$) afin de montrer que la série converge absolument.
- 6 Montrer qu'il y a convergence normale sur tout compact.
- 7 La convergence de la série donnée dans l'énoncé permet d'écrire un développement limité en 0 de la fonction $A \mapsto e^A$.

Problème

- 8 Déterminer $\det(A - XI_3)$ et exhiber une condition nécessaire et suffisante pour que les trois racines de ce polynôme soient positives ou nulles.
- 9 Pour montrer que $e^A \in S_3^+$, utiliser la caractérisation obtenue à la question précédente.
- 10 Montrer que pour tout $k \in \mathbb{N}$, $A^k = PD^kP^{-1}$. Utiliser la continuité pour passer à la limite dans les sommes partielles. Enfin, montrer que $e^D \in S_n^+$.
- 11 Utiliser la caractérisation de la question 8.
- 12 Utiliser le fait que les matrices symétriques réelles sont diagonalisables en base orthonormée dans la deuxième partie de la question.
- 13 Pour le produit de deux matrices symétriques positives, trouver un contre-exemple en dimension 2 lorsque ces dernières ne commutent pas.
- 15 Pour montrer que $A * B \in S_n^+$, utiliser la caractérisation obtenue dans la question 12.
- 16 Raisonner terme à terme dans T_N .
- 17 Pour montrer que $E(A) \in S_n^+$, on utilise les questions 13, 15 et 16 ainsi que le caractère fermé de S_n^+ .

EXERCICE 1

1 Pour $(x_1, x_2) \in \mathbb{C}^2$,

$$V(x_1, x_2) = \begin{vmatrix} 1 & 1 \\ x_1 & x_2 \end{vmatrix}$$

$$V(x_1, x_2) = x_2 - x_1$$

Si $x_1, x_2, \dots, x_n$ ne sont pas deux à deux distincts alors il existe $(k, \ell) \in \llbracket 1; n \rrbracket^2$ tel que

$$k < \ell \quad \text{et} \quad x_k = x_\ell$$

Dans ce cas, les colonnes k et ℓ du déterminant $V(x_1, x_2, \dots, x_n)$ sont identiques donc, par caractère alterné du déterminant, il vient que

$$V(x_1, x_2, \dots, x_n) = 0$$

Par ailleurs, le facteur d'indice (k, ℓ) dans le produit

$$\prod_{1 \leq i < j \leq n} (x_j - x_i)$$

est nul. Par conséquent,

$$\prod_{1 \leq i < j \leq n} (x_j - x_i) = 0 = V(x_1, x_2, \dots, x_n)$$

Il suffit de démontrer le résultat pour des complexes x_i deux à deux distincts.

2 Soit $t \in \mathbb{R}$. Alors, en développant le déterminant de Vandermonde $P(t)$ par rapport à la dernière colonne, il existe $(d_0, d_1, \dots, d_{n-2}) \in \mathbb{C}^{n-1}$ tel que

$$\begin{aligned} P(t) &= \sum_{i=0}^{n-2} d_i t^i + (-1)^{n+n} V(x_1, x_2, \dots, x_{n-1}) t^{n-1} \\ &= \sum_{i=0}^{n-2} d_i t^i + V(x_1, x_2, \dots, x_{n-1}) t^{n-1} \end{aligned}$$

où, pour tout $i \in \llbracket 0; n-2 \rrbracket$, d_i représente le cofacteur d'indice (i, n) et est par conséquent indépendant de t . Ainsi,

La fonction P est polynomiale de degré au plus $n-1$, et le coefficient devant le terme de degré $n-1$ est $V(x_1, x_2, \dots, x_{n-1})$.

Montrons par récurrence que la propriété

$$\mathcal{P}(n) : \quad \forall (x_1, x_2, \dots, x_n) \in \mathbb{C}^n \quad V(x_1, x_2, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

est vraie pour tout $n \geq 2$.

- $\mathcal{P}(2)$ est vraie d'après la question 1.
- $\mathcal{P}(n-1) \implies \mathcal{P}(n)$: soit $n \geq 3$. On suppose que $\mathcal{P}(n-1)$ est vraie. On considère $(x_1, x_2, \dots, x_n) \in \mathbb{C}^n$. Si les x_i ne sont pas deux à deux distincts, on a montré dans la question 1 que la formule est vérifiée. Supposons maintenant que les x_i sont deux à deux distincts. Soit $i \in \llbracket 1; n-1 \rrbracket$. On a montré dans la question 1 que

$$P(x_i) = V(x_1, x_2, \dots, x_{n-1}, x_i) = 0$$

Ainsi, le polynôme P admet $n - 1$ racines distinctes. Par ailleurs, il est de degré au plus $n - 1$. Donc il existe $\alpha \in \mathbb{R}$ tel que

$$\forall t \in \mathbb{R} \quad P(t) = \alpha \prod_{i=1}^{n-1} (t - x_i)$$

En développant ce produit, on remarque que α est le coefficient devant t^{n-1} . Par identification des coefficients, vu ce qui a été prouvé au début de la question, il vient puisque $\mathcal{P}(n - 1)$ est vraie,

$$\alpha = V(x_1, x_2, \dots, x_{n-1}) = \prod_{1 \leq i < j \leq n-1} (x_j - x_i)$$

$$\begin{aligned} \text{Puis, } V(x_1, x_2, \dots, x_n) &= P(x_n) = \alpha \prod_{i=1}^{n-1} (x_n - x_i) \\ &= \prod_{1 \leq i < j \leq n-1} (x_j - x_i) \prod_{i=1}^{n-1} (x_n - x_i) \\ V(x_1, x_2, \dots, x_n) &= \prod_{1 \leq i < j \leq n} (x_j - x_i) \end{aligned}$$

ce qui prouve $\mathcal{P}(n)$.

- **Conclusion** : La propriété $\mathcal{P}(n)$ est vraie pour tout $n \geq 2$.

$$\forall n \geq 2 \quad \forall (x_1, x_2, \dots, x_n) \in \mathbb{C}^n \quad V(x_1, x_2, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

3 Le déterminant de A est égal à celui de sa transposée donc

$$\det A = \begin{vmatrix} 1 & 2 & \cdots & n \\ 1^2 & 2^2 & \cdots & n^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1^n & 2^n & \cdots & n^n \end{vmatrix}$$

Pour tout $i \in \llbracket 1; n \rrbracket$, factorisons la i -ième colonne par i . Il vient

$$\det A = n! \begin{vmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & n \\ 1^2 & 2^2 & \cdots & n^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{n-1} & 2^{n-1} & \cdots & n^{n-1} \end{vmatrix} = n! V(1, 2, \dots, n)$$

D'après la formule démontrée dans la question 2, on obtient

$$\begin{aligned} \det A &= n! \prod_{1 \leq i < j \leq n} (j - i) \\ &= n! \prod_{j=2}^n \prod_{i=1}^{j-1} (j - i) \\ &= n! \prod_{j=2}^n \prod_{k=1}^{j-1} k \quad (k = j - i) \\ &= n! \prod_{j=2}^n (j - 1)! \\ &= n! \prod_{j=1}^{n-1} j! \end{aligned}$$

Enfin,

$$\det A = \prod_{j=1}^n j!$$

CCINP Informatique optionnelle MP 2022

Corrigé

Ce corrigé est proposé par Jean Starynkévitch (professeur en CPGE) ; il a été relu par Vincent Puyhaubert (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

Cette épreuve est composée de trois exercices indépendants. Les chapitres utilisés sont à chaque fois différents.

- La partie I est un exercice de facture classique sur les automates. Y sont introduits les concepts d'automate augmenté (muni d'un sous-ensemble O d'états supplémentaire) et de calcul réussi à un seuil s , dans lequel par définition au moins un état de O figure parmi $s + 1$ états consécutifs. Il est d'abord demandé de manipuler les définitions sur des exemples simples, puis de montrer qu'on peut ramener la recherche du langage des mots reconnus au seuil s à la recherche du langage reconnu par un automate classique.
- Dans la partie II, on étudie la structure d'arbre binomial, puis celle de tas bino-mial, qui constitue une réalisation de la structure abstraite de file de priorité. L'implémentation est faite en Python.
- La partie III est davantage un exercice de mathématiques prétexte à quelques questions de programmation en OCaml. On introduit l'arbre infini de Calkin-Wilf et la suite diatomique de Stern, qui permettent de démontrer le beau théorème suivant : la suite $(v_n)_{n \in \mathbb{N}}$ définie par récurrence par

$$\begin{cases} v_0 = 0 \\ v_{n+1} = f(v_n) \end{cases} \quad \text{où} \quad f(x) = \frac{1}{1 + 2\lfloor x \rfloor - x}$$

est une bijection de $\mathbb{N}$ vers $\mathbb{Q}_+$. Cette partie porte intégralement sur le programme de première année.

Ce sujet, de longueur et de difficulté raisonnables, est bien adapté au concours. Il comporte toutefois peu de questions de programmation. Notons que, comme souvent dans les énoncés d'informatique, une partie des définitions du cours sont rappelées dans le sujet, avec parfois quelques différences par rapport aux usages. Il faut être capable de s'adapter à l'énoncé.

INDICATIONS

Partie I

- 1-3 Déterminer la forme des calculs réussis dans $\mathcal{A}$ avant d'en extraire dans chacun des 3 cas ceux qui satisfont la condition (iii).
- 4 Relire la définition inhabituelle figurant dans le sujet de calcul dans un automate. Que dire de la longueur d'un calcul réussi au seuil s sans passer par O ?
- 5 Remarquer que chacun des ensembles $\{q_0, q_1, \dots, q_s\}$, $\{q_{\ell-s}, q_{\ell-s+1}, \dots, q_{\ell}\}$ et $\{q_{i_j+1}, \dots, q_{i_j+s+1}\}$ contient au moins un élément de O .
- 6 Construire l'automate en prenant pour ensemble d'états les couples (q, s') tels que $q \in Q$ et $s' \in \llbracket 0; s \rrbracket$ représente le nombre d'états parcourus depuis le dernier état de O rencontré s'il en existe, ou depuis l'état initial sinon.

Partie II

- 8 Penser à comparer les racines, afin de préserver les invariants d'un arbre binomial écrites dans la définition 6.
- 10 Utiliser le résultat de la question 9.
- 12 Créer une fonction auxiliaire récursive `EstBinomialOrdre(a, k)` qui teste si a est un arbre binomial d'ordre k .
- 15 Montrer tout d'abord que le minimum d'un tas est présent en étiquette de la racine d'un des arbres binomiaux le composant. Pour l'estimation de la complexité, ne pas confondre k et $|T|$.
- 18 Introduire, lorsqu'il existe, le plus petit indice i tel que $s_i = 0$.

Partie III

- 23 L'algorithme d'Euclide fait l'affaire.
- 26 Raisonner par récurrence sur k .
- 29 Raisonner par récurrence sur la profondeur maximale de deux nœuds de même étiquette, constater que les conditions $n < d$ et $d < n$ caractérisent respectivement les fils gauches et les fils droits et utiliser la question 25.
- 31 Si k désigne le niveau du nœud v_i , compter le nombre de fils des nœuds à gauche de v_i , et le nombre de nœuds à droite de v_i .
- 34 Montrer par récurrence forte que $s_i \wedge s_{i+1} = 1$ et $v_i = s_i/s_{i+1}$ pour tout $i \in \mathbb{N}$.
- 35 Constater que i est pair et que $0 < v_i < 1$.
- 36 Utiliser la question 26.
- 37 Coder une fonction auxiliaire construisant le chemin en sens inverse.
- 38 Coder une fonction auxiliaire qui, à un nœud et une liste de directions, associe le descendant du nœud selon la liste.
- 39 Coder une fonction auxiliaire calculant le plus long préfixe commun à deux listes.
- 40 Observer que, si v_i et v_{i+1} ont deux pères distincts v_j et v_{ℓ} , alors $\ell = j + 1$, v_i est un fils droit et v_{i+1} un fils gauche.
- 41 Utiliser les résultats des questions 40 et 26.

I. AUTOMATES

1 Tout d'abord, observons que les calculs réussis dans l'automate (non augmenté) $\mathcal{A}$ sont exactement les calculs de la forme

$$(q'_0, \dots, q'_n) = (\underbrace{q_0, \dots, q_0}_{x+1 \text{ fois}}, q_1, \underbrace{q_2, \dots, q_2}_{y+1 \text{ fois}})$$

pour un certain couple $(x, y) \in \mathbb{N}^2$; le calcul précédent est alors d'étiquette $b^x a^2 b^y$.

Constatons également que les conditions (i) et (ii) figurant dans la définition 3 de calcul réussi dans un automate augmenté ne sont rien de plus que celles qui indiquent un calcul réussi dans l'automate non augmenté sous-jacent. En d'autres termes, les calculs réussis au seuil s dans l'automate augmenté sont exactement ceux réussis dans l'automate non augmenté vérifiant de plus la condition (iii).

Soit $(q'_0, \dots, q'_n) = (\underbrace{q_0, \dots, q_0}_{x+1 \text{ fois}}, q_1, \underbrace{q_2, \dots, q_2}_{y+1 \text{ fois}})$ un calcul réussi dans l'automate $\mathcal{A}$. Si le calcul est réussi au seuil 2 dans l'automate augmenté $(\mathcal{A}, \{q_0\})$, q_0 figure parmi les trois derniers états parcourus, donc $y = 0$. Réciproquement, lorsque $y = 0$, le calcul est bien réussi au seuil 2 dans $(\mathcal{A}, \{q_0\})$, or l'étiquette de ce calcul est $b^x a^2$.

$$\boxed{L_2(\mathcal{A}_{q_0}) = \{b^x a^2 \mid x \in \mathbb{N}\}}$$

2 Soit $(q'_0, \dots, q'_n) = (\underbrace{q_0, \dots, q_0}_{x+1 \text{ fois}}, q_1, \underbrace{q_2, \dots, q_2}_{y+1 \text{ fois}})$ un calcul réussi dans l'automate $\mathcal{A}$. Si ce calcul est réussi au seuil 2 dans l'automate augmenté $(\mathcal{A}, \{q_1\})$, l'état q_1 figure parmi les trois premiers états, donc $x \leq 1$, ainsi que parmi les trois derniers, ce qui entraîne $y \leq 1$. Réciproquement, si $(x, y) \in \{0, 1\}^2$, on a $n \leq 4$ et pour tout $i \in \llbracket 0; n-2 \rrbracket$, $q_1 \in \{q'_i, q'_{i+1}, q'_{i+2}\}$. Finalement,

$$\boxed{L_2(\mathcal{A}_{q_1}) = \{b^x a^2 b^y \mid (x, y) \in \{0, 1\}^2\} = \{aa, aab, baa, baab\}}$$

3 Lorsque le sous-ensemble d'états O est vide, la contrainte $\{q_i, \dots, q_{i+s}\} \cap O \neq \emptyset$ présente en fin de (iii) n'est satisfaite pour aucun indice i . Pour que la contrainte (iii) soit satisfaite, il faut et il suffit que l'ensemble des indices i sur lesquelles elle porte soit vide. C'est le cas si et seulement si $s > n$. Ainsi, pour tout automate $\mathcal{A}$, $L_s(\mathcal{A}_\emptyset)$ est le langage des mots reconnus par $\mathcal{A}$ qui sont de longueur strictement inférieure à s .

Dans le cas présent, où le langage reconnu par $\mathcal{A}$ est $\{b^x a^2 b^y \mid (x, y) \in \mathbb{N}^2\}$,

$$L_s(\mathcal{A}_\emptyset) = \{b^x a^2 b^y \mid (x, y) \in \mathbb{N}^2, \quad x + y + 2 < s\}$$

Le cardinal de $L_s(\mathcal{A}_\emptyset)$ est donc le nombre de couples $(x, y) \in \mathbb{N}^2$ tels que $x + y + 2 < s$. Par conséquent, si $s \leq 2$, on a $L_s(\mathcal{A}_\emptyset) = \emptyset$. Lorsque $s \geq 3$, en partitionnant selon la valeur de x , on a

$$|L_s(\mathcal{A}_\emptyset)| = \sum_{x=0}^{s-3} (s-2-x) = \sum_{p \leftarrow s-2-x}^{s-2} p = \frac{(s-2)(s-1)}{2}$$

Ainsi,

$$\boxed{\forall s \in \mathbb{N} \quad |L_s(\mathcal{A}_\emptyset)| = \begin{cases} 0 & \text{si } s \leq 2 \\ \frac{(s-2)(s-1)}{2} & \text{sinon} \end{cases}}$$

4

L'énoncé est mal formulé : il faut lire « majorant du nombre de calculs réussis ».

La définition du sujet de *calcul* dans un automate est inhabituelle, mais simplifie le comptage demandé : le sujet définit un calcul comme une « suite » finie $(q_0, q_1, \dots, q_n)$ d'états, tandis qu'usuellement, la donnée du calcul est la donnée d'une suite de transitions qui se recollent :

$$q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \dots \xrightarrow{u_n} q_n$$

La conséquence est qu'il peut y avoir plusieurs étiquettes à un même calcul dans l'énoncé, mais aussi que construire un calcul au sens de l'énoncé repose sur moins de paramètres : on n'a pas besoin de savoir quelle lettre est lue, mais uniquement l'existence d'une transition entre deux états.

Pour qu'un calcul $(q_0, q_1, \dots, q_n)$ soit réussi au seuil s sans passer par O , il est nécessaire que $n \leq s-1$. En effet, si $s \leq n$, la condition (iii) assure que $\{q_0, q_1, \dots, q_s\}$ intersecte O .

Un tel calcul de longueur n est donné par un état initial, un état final, et $n-1$ valeurs d'états intermédiaires, tous ces états n'étant pas dans O . Ainsi, pour $n \leq s-1$ fixé, il y a au plus $|I \setminus O| |Q \setminus O|^{n-1} |F \setminus O|$ calculs possibles si $n \geq 1$, et $|I \cap F \setminus O|$ si $n = 0$. Autrement dit,

En notant $p = |I \setminus O|$, $q = |Q \setminus O|$, $r = |F \setminus O|$ et $i = |I \cap F \setminus O|$, le nombre de calculs réussis au seuil s sans passer par O est majoré par

$$i + \sum_{n=1}^{s-1} p q^{n-1} r = i + \frac{pr(q^{s-1} - 1)}{q - 1}$$

En majorant brutalement p , q , r et i par $|Q|$,

Un majorant du nombre de calculs réussis au seuil s sans passer par O est

$$\sum_{n=0}^{s-1} |Q|^{n+1} = |Q| \frac{|Q|^s - 1}{|Q| - 1}$$

Si la définition d'un calcul comportait la donnée des lettres lues au cours des transitions, on aurait trouvé le majorant

$$\sum_{n=0}^{s-1} |\Sigma|^n |Q|^{n+1} = |Q| \frac{|Q|^s |\Sigma|^s - 1}{|Q||\Sigma| - 1}$$

5

Puisque, ou bien $\ell < s$, ou bien $\{q_0, q_1, \dots, q_s\} \cap O \neq \emptyset$, on a

$$i_0 \leq s$$

Si $\ell < s$, on a $0 \leq i_k \leq \ell$. Si $\ell \geq s$, on a $\{q_{\ell-s}, q_{\ell-s+1}, \dots, q_\ell\} \cap O \neq \emptyset$, donc

$$\ell - s \leq i_k \leq \ell$$

Soit $j \in \llbracket 0; k-1 \rrbracket$. Si $i_j + s + 1 \leq n$, on a $\{q_{i_j+1}, \dots, q_{i_j+s+1}\} \cap O \neq \emptyset$, donc $i_{j+1} \leq i_j + s + 1$. Cette inégalité est toujours valide lorsque $i_j + s + 1 > n$ du fait que $i_{j+1} \leq i_j + s + 1$. Ainsi,

$$\forall j \in \llbracket 0; k-1 \rrbracket \quad i_{j+1} - i_j \leq s + 1$$

Centrale Maths 1 MP 2022 — Corrigé

Ce corrigé est proposé par Angèle Niclas (ENS Lyon) ; il a été relu par David Michel (professeur en CPGE) et Tristan Poullaouec (professeur en CPGE).

Ce sujet introduit la notion d'espace vectoriel symplectique réel : c'est un espace vectoriel réel E de dimension finie muni d'une forme bilinéaire ω antisymétrique et non dégénérée. Un endomorphisme u d'un tel espace est alors dit symplectique s'il vérifie

$$\forall (x, y) \in E^2 \quad \omega(u(x), u(y)) = \omega(x, y)$$

Ce formalisme est utile en mécanique hamiltonienne et plus particulièrement en mécanique céleste.

L'énoncé choisit de se placer alternativement dans le cas général et dans le cas particulier de l'espace vectoriel $\mathbb{R}^{2m}$ muni de la forme symplectique

$$b_s : (x, y) \mapsto \langle x, j(y) \rangle$$

où j est l'endomorphisme canoniquement associé à la matrice

$$J = \begin{pmatrix} 0 & -I_m \\ I_m & 0 \end{pmatrix}$$

Le sujet propose au passage d'étudier les endomorphismes symplectiques de l'espace $(\mathbb{R}^n, b_s)$, qui ont un lien fort avec le groupe symplectique

$$\mathrm{Sp}_n(\mathbb{R}) = \{M \in \mathcal{M}_n(\mathbb{R}) \mid M^\top J M = J\} \quad \text{où} \quad n = 2m$$

Le problème est composé de quatre parties ; les deux dernières sont largement indépendantes.

- La partie I, très courte, propose de montrer deux résultats préliminaires sur les matrices. Ils seront utilisés dans les parties II et III.
- Dans la partie II, on propose d'étudier différents objets symplectiques et leurs propriétés. Après avoir défini les espaces symplectiques, et généralisé la notion d'orthogonalité à ces espaces, on introduit les endomorphismes symplectiques, que l'on caractérise dans le cas particulier de l'espace symplectique $(\mathbb{R}^n, b_s)$.
- La partie III a pour but de démontrer l'inclusion $\mathrm{Sp}_n(\mathbb{R}) \subset \mathrm{SL}_n(\mathbb{R})$ en utilisant deux méthodes différentes. La première repose sur l'introduction de la décomposition polaire des matrices symplectiques réelles. La deuxième méthode, plus générale, démontre que tout endomorphisme symplectique peut se décomposer à l'aide de transvections symplectiques.
- Finalement, la partie IV a pour objectif d'étudier les plongements symplectiques et de prouver le théorème de non-tassement linéaire, qui établit qu'il est possible de plonger de manière symplectique une boule de rayon R dans un cylindre de rayon R' si et seulement si $R \leq R'$.

Ce problème fait appel à très peu de résultats du programme hormis quelques propriétés d'algèbre linéaire et de décomposition matricielle. La difficulté majeure repose sur la compréhension et la manipulation des nouveaux concepts introduits par l'énoncé. Il nécessite d'avoir en tête les techniques de preuve habituellement utilisées en algèbre linéaire et de savoir les adapter dans le cadre moins conventionnel des espaces symplectiques.

INDICATIONS

Partie I

- 1 Penser à utiliser des éléments X et Y de la base canonique de $\mathcal{M}_{n,1}(\mathbb{R})$.
- 2 Utiliser le théorème spectral et le fait que $\langle M^\top MX, X \rangle = \langle MX, MX \rangle$.

Partie II

- 5 Considérer l'espace $F = \text{Vect}(x)$ où $x \in E \setminus \{0_E\}$ et utiliser le résultat de la question 3.
- 6 Utiliser la non-dégénérescence de ω pour montrer que d_ω est injective.
- 8 Appliquer le théorème du rang à $r_F \circ d_\omega$ en utilisant les résultats des questions 6 et 7.
- 9 Utiliser le résultat de la question 8 et la non-dégénérescence de ω .
- 11 Utiliser le résultat des questions 1 et 10 puis la non-dégénérescence de ω pour prouver que $\text{Ker}(\Omega) = \{0_{n,1}\}$.
- 12 Utiliser la question 11 et exhiber une contradiction si n est impair.
- 13 Pour montrer que b_s est non dégénérée, calculer $b_s(x, y)$ où y est un vecteur de la base canonique.
- 15 Utiliser le résultat de la question 1.
- 16 Remarquer que $J^{-1} = -J = J^\top$ puis penser à appliquer l'inverse des deux côtés de l'égalité $(M^\top)^{-1}JM^{-1} = J$.

Partie III

- 18 Utiliser le résultat de la question 17 et reconnaître la formule du déterminant d'une matrice carrée d'ordre 2.
- 20 Utiliser la décomposition de la question 19 et l'égalité $\det(U - iV) = \overline{\det(U + iV)}$.
- 21 Montrer que $\text{OSp}_n(\mathbb{R})$ est un fermé borné. Pour cela, contrôler la norme de chaque colonne de $M \in \text{OSp}_n(\mathbb{R})$ et considérer l'application $M \mapsto M^\top M - I_n$.
- 23 Utiliser les résultats des questions 20 et 22.
- 24 Utiliser les résultats des questions 15 et 16 pour montrer que s^2 est symplectique.
- 25 Penser aux résultats des questions 16 et 24.
- 26 Utiliser les résultats des questions 23 et 25 et le fait que $\det(M)^2 = 1$.
- 27 Se rappeler le résultat de la question 3, la bilinéarité et l'antisymétrie de ω .
- 28 Penser au résultat de la question 3 et la bilinéarité de ω .
- 29 Utiliser le résultat de la question 28 pour montrer successivement que $\det(\tau_a^\lambda) \geq 0$, puis que $\det(\tau_a^\lambda) \neq 0$.
- 30 Penser au résultat de la question 28.
- 31 Utiliser le résultat de la question 3.
- 32 Considérer des vecteurs z_1 et z_2 tels que $\omega(x, z_1) \neq 0$ et $\omega(y, z_2) \neq 0$, et les sommer éventuellement pour obtenir z .
- 33 Distinguer les cas selon la valeur de $\omega(x, y)$ et utiliser les résultats des questions 31 et 32.

- 34 Considérer un vecteur x tel que $\omega(x, e_1) \neq 0$, puis raisonner par l'absurde pour montrer que x et e_1 ne sont pas colinéaires en utilisant le résultat de la question 3.
- 35 Montrer que les conditions d'application du lemme prouvé en question 33 sont vérifiées.
- 36 Prouver que pour tous $x, y \in E$ satisfaisant $\omega(y - x, e_1) = 0$ et $\omega(x, y) \neq 0$, il existe $\lambda \in \mathbb{R}$ tel que $\tau_{y-x}^\lambda(x) = y$ et $\tau_{y-x}^\lambda(e_1) = e_1$ en utilisant le résultat de la question 31. Distinguer ensuite les cas selon la valeur de $\omega(\tilde{f}_1, f_1)$. Dans le cas où $\omega(\tilde{f}_1, f_1) = 0$, utiliser un vecteur intermédiaire $z = f_1 + e_1$.
- 38 Utiliser le résultat de la question 37.
- 39 Utiliser les résultats des questions 3, 8 et 9.
- 40 Procéder par récurrence sur m et utiliser les questions 30 et 34 à 39 pour montrer que tout endomorphisme symplectique d'un espace symplectique (E, ω) de dimension $2m$ peut s'écrire comme la composée d'au plus $4m$ transvections symplectiques de E .
- 41 Montrer que $\mathrm{Sp}_n(\mathbb{R})$ est étoilé par rapport à I_n en utilisant la décomposition donnée par le théorème prouvé en question 40 et les applications $t \mapsto \tau_a^{t\lambda}$.
- 42 Utiliser le théorème prouvé en question 40, le résultat de la question 29 et le fait que $\det(M)^2 = 1$ pour toute matrice $M \in \mathrm{Sp}_n(\mathbb{R})$.

Partie IV

- 43 Choisir une application u qui multiplie par r les coordonnées x_1 et y_1 et qui reste de déterminant 1.
- 44 Utiliser l'indication de l'énoncé même dans le cas $\lambda \in \mathbb{R}$.
- 45 Appliquer le résultat de la question 44 à toutes les valeurs propres de u pour majorer son déterminant.
- 46 Utiliser le résultat de la question 45.
- 47 Utiliser le fait que $\psi^\top \in \mathrm{Symp}_{b_s}(\mathbb{R}^n)$ puis l'inégalité de Cauchy-Schwarz.
- 48 Normaliser les vecteurs de la question 47 puis utiliser $\psi(B^{2m}(1)) \subset Z^{2m}(r)$.
- 49 Penser au résultat de la question 48.

I. PRÉLIMINAIRES

1 Soient $i, j \in \llbracket 1; n \rrbracket$, considérons les vecteurs $X, Y \in \mathcal{M}_{n,1}(\mathbb{R})$ définis par

$$\forall k \in \llbracket 1; n \rrbracket \quad X_k = \begin{cases} 1 & \text{si } k = i \\ 0 & \text{sinon} \end{cases}, \quad \text{et} \quad Y_k = \begin{cases} 1 & \text{si } k = j \\ 0 & \text{sinon} \end{cases}$$

Avec ce choix de vecteurs,

$$X^\top A Y = \sum_{k=1}^n \sum_{\ell=1}^n X_k A_{k,\ell} Y_\ell = X_i A_{i,j} Y_j = A_{i,j}$$

et de même, $X^\top B Y = B_{i,j}$. Comme $X^\top A Y = X^\top B Y$, on déduit que $A_{i,j} = B_{i,j}$. Ce résultat étant vrai pour tous i et j dans $\llbracket 1; n \rrbracket$, tous les coefficients des matrices A et B sont égaux, ce qui prouve que

$$\boxed{A = B}$$

2 Soit $M \in GL_n(\mathbb{R})$. La matrice $M^\top M$ est réelle et symétrique donc le théorème spectral permet d'affirmer que toutes ses valeurs propres sont réelles. Soit $\lambda \in \mathbb{R}$ une valeur propre de $M^\top M$ associée à un vecteur propre $X \in \mathcal{M}_{n,1}(\mathbb{R}) \setminus \{0_{n,1}\}$. Remarquons que

$$\langle M^\top M X, X \rangle = \langle \lambda X, X \rangle = \lambda \langle X, X \rangle = \lambda \|X\|^2$$

Comme $\langle \cdot, \cdot \rangle$ est le produit scalaire canonique de $\mathbb{R}^n$, on a

$$\forall Y_1, Y_2 \in \mathcal{M}_{n,1}(\mathbb{R}) \quad \langle Y_1, Y_2 \rangle = Y_1^\top Y_2$$

$$\text{Ainsi,} \quad \langle M^\top M X, X \rangle = (M^\top M X)^\top X = X^\top M^\top M X = \langle M X, M X \rangle = \|M X\|^2$$

Le vecteur X est non nul et M est inversible, donc $M X$ est non nul et

$$\lambda \|X\|^2 = \|M X\|^2 \quad \Rightarrow \quad \lambda = \frac{\|M X\|^2}{\|X\|^2} > 0$$

On a donc prouvé que

$$\boxed{\text{Toutes les valeurs propres de } M^\top M \text{ sont réelles et strictement positives.}}$$

Le théorème spectral permet également d'affirmer que la matrice $M^\top M$ est diagonalisable dans une base orthonormale. En notant $(\lambda_1, \dots, \lambda_n)$ les valeurs propres de $M^\top M$, il existe $P \in O_n(\mathbb{R})$ telle que

$$M^\top M = P \operatorname{diag}(\lambda_1, \dots, \lambda_n) P^{-1}$$

On a prouvé que $\lambda_i > 0$ pour tout $i \in \llbracket 1; n \rrbracket$. Définissons alors

$$S = P \operatorname{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) P^{-1}$$

de sorte que

$$S^2 = P \left(\operatorname{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n}) \right)^2 P^{-1} = P \operatorname{diag}(\lambda_1, \dots, \lambda_n) P^{-1} = M^\top M$$

Finalement, comme $P^{-1} = P^\top$, la matrice S est symétrique car

$$S^\top = (P^{-1})^\top \operatorname{diag}(\lambda_1, \dots, \lambda_n) P^\top = P \operatorname{diag}(\lambda_1, \dots, \lambda_n) P^{-1} = S$$

De ce fait, on a prouvé que

$$\boxed{\text{Il existe une matrice } S \text{ symétrique à valeurs propres strictement positives telle que } S^2 = M^\top M}$$

Centrale Maths 2 MP 2022 — Corrigé

Ce corrigé est proposé par Antoine Barrier (ENS Paris-Saclay) ; il a été relu par Hugo Koubbi (ENS Paris-Saclay) et Vincent Puyhaubert (professeur en CPGE).

Ce sujet porte sur la dérivation de sommes de séries de fonctions. D'abord dans un cadre d'analyse déterministe : l'objectif est notamment de montrer que la dérivation terme à terme de la série peut s'obtenir avec des hypothèses plus faibles que celles vues dans les théorèmes au programme. Ensuite dans un cadre probabiliste : on s'intéresse à des séries aléatoires de fonctions qui possèdent des propriétés similaires de dérivation terme à terme avec probabilité 1.

- Dans la partie I, on établit une inégalité d'interpolation permettant de majorer les dérivées d'une fonction f de classe $\mathcal{C}^K$ par une quantité qui est une combinaison linéaire de $\|f^{(K)}\|_\infty$ et de valeurs de f en des points fixés.
- La partie II utilise l'inégalité d'interpolation démontrée dans la première partie pour obtenir un résultat de dérivation terme à terme de la somme d'une série de fonctions sous des hypothèses moins fortes. Un exemple est étudié en fin de partie.
- La troisième partie introduit une série de fonctions aléatoires de la forme

$$\sum a_n X_n \quad \text{avec} \quad X_n \text{ suivant la loi uniforme sur } \{-1, 1\}$$

Sous l'hypothèse que la série $\sum a_n^2$ converge, on montre que la série aléatoire ci-dessus converge avec probabilité 1.

- Enfin, la dernière partie combine les études menées dans les parties II et III afin d'obtenir des conditions simples pour qu'une série aléatoire de fonctions soit de classe $\mathcal{C}^K$ et jouisse de propriétés de dérivation terme à terme avec probabilité 1.

Ce sujet d'analyse utilise de nombreux chapitres des programmes de MPSI et de MP : analyse de fonctions réelles, polynômes, séries de fonctions et probabilités. De nombreuses questions peuvent être traitées rapidement dès lors que l'on pense à appliquer le bon résultat du cours. Il constitue donc un très bon sujet d'entraînement pour assimiler ces résultats. Par ailleurs, des questions plus techniques requièrent de développer des raisonnements en plusieurs étapes, ce qui suppose une rédaction très soignée.

INDICATIONS

Partie I

- 1 Utiliser le théorème fondamental de l'analyse puis l'inégalité triangulaire.
- 3 Appliquer l'inégalité (I.1) à la fonction

$$g: \begin{cases} [0, 1] \longrightarrow \mathbb{R} \\ x \longmapsto f'(x) - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \end{cases}$$

avec un réel y tel que $g(y) = 0$.

- 7 Définir $L_j = \Psi^{-1}(\{e_j\})$ où e_j est le j -ième vecteur de la base canonique de $\mathbb{R}^K$.
- 8 Procéder par récurrence. Pour l'hérédité, appliquer le théorème de Rolle entre deux zéros consécutifs de $f^{(k)} - P^{(k)}$.
- 9 Appliquer l'inégalité des accroissements finis à la fonction $f^{(k)} - P^{(k)}$ entre un point de $[0; 1]$ et un de ses zéros.
- 10 En utilisant le résultat de la question 9 et l'égalité $P^{(K)} = 0$, établir que

$$\max_{0 \leq k \leq K-1} \|f^{(k)}\|_{\infty} \leq \|f^{(K)}\|_{\infty} + \max_{0 \leq k \leq K-1} \|P^{(k)}\|_{\infty}$$

et montrer que $\|P^{(k)}\|_{\infty} \leq C \sum_{\ell=1}^K |f(x_{\ell})|$ pour une certaine constante C définie à partir des normes des dérivées des polynômes $(L_j)_{j \in \llbracket 1; K \rrbracket}$.

Partie II

- 11 Utiliser l'inégalité d'interpolation (I.1) précédemment démontrée.
- 13 Utiliser le théorème de dérivation des séries de fonctions de classe $\mathcal{C}^K$.
- 14 Démontrer que l'équation différentielle linéaire $y''(x) = (-1)^n 2^{-nx^2}$ admet une unique solution satisfaisant $y(1) = 0$ et $y(2) = 0$.
- 15 Montrer que F satisfait les hypothèses (H1) et (H2) avec $K = 2$, $x_1 = 1$ et $x_2 = 2$ sur tout segment de $]0; +\infty[$.
- 17 Appliquer l'inégalité (I.3) à F , en s'assurant qu'elle reste valable sur l'intervalle $[1; 2]$.

Partie III

- 19 Montrer que $E(X_n) = 0$ et $V(X_n) = 1$ puis utiliser la linéarité de l'espérance, la bilinéarité de la variance, et la mutuelle indépendance des $(X_n)_{n \in \mathbb{N}}$.
- 20 Appliquer l'inégalité de Bienaymé-Tchebychev.
- 22 Penser à la formule des probabilités totales.
- 23 Pour $x = (x_n)_{n \in \llbracket \phi(j)+1; \phi(j+1) \rrbracket} \in \{-1, 1\}^{\phi(j+1)-\phi(j)}$, introduire l'évènement

$$E_x = \bigcap_{n=\phi(j)+1}^{\phi(j+1)} \{X_n = x_n\}$$

Dans un premier temps, pour montrer que l'application est à valeurs entières, écrire $\{|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}\} \cap B_{j,m}$ comme l'union des $(E_x)_{x \in \Theta_{\alpha}}$ pour un sous-ensemble Θ_{α} de $\{-1, 1\}^{\phi(j+1)-\phi(j)}$. Montrer ensuite la parité en établissant que $\text{Card}(\Theta_{\alpha}) = \text{Card}(\Theta_{-\alpha})$.

- 25 Réécrire le résultat de la question 24 sous forme d'inclusion d'évènements et utiliser la parité de l'application définie à la question 23.
- 26 Remarquer que $(\bigcup_{j \geq J} B_j)_{J \in \mathbb{N}}$ est une suite décroissante d'évènements.
- 27 Montrer que l'évènement considéré est le complémentaire de B.
- 28 Lorsque l'évènement de la question 27 se réalise, montrer que la série donnée en indication converge.
- 29 Supposer de nouveau que l'évènement de la question 27 se réalise et utiliser ses propriétés ainsi que la convergence de la suite des sommes partielles $(S_{\phi(j)})_{j \in \mathbb{N}}$ pour montrer la convergence de $(S_n)_{n \in \mathbb{N}}$.

Partie IV

- 31 Appliquer le résultat de la question 29 avec $(a_n)_{n \in \mathbb{N}} = (f_n(x_\ell))_{n \in \mathbb{N}}$.
- 32 Montrer que $(g_n = f_n - P_n)_{n \in \mathbb{N}}$ satisfait les hypothèses (H1) et (H2'). En appliquant la question 31 avec $(g_n)_{n \in \mathbb{N}}$, en déduire que sur l'évènement associé de probabilité 1, la suite $(h_n = X_n(f_n - P_n))_{n \in \mathbb{N}}$ satisfait les hypothèses (H1) et (H2) de la partie II.
- 33 En supposant que l'évènement de la question 31 se réalise, appliquer les résultats de la partie II avec $(\ell_n = X_n f_n)_{n \in \mathbb{N}}$.
- 34 Montrer le résultat pour $K = 2$ en vérifiant les hypothèses (H1) et (H2'), ou remarquer qu'il est vrai pour tout $K \in \mathbb{N}$ au prix de calculs plus longs.

I. INÉGALITÉS D'INTERPOLATION DES DÉRIVÉES

1 Soit $f \in \mathcal{C}^1([0;1])$. Fixons $x \in [0;1]$. Le théorème fondamental de l'analyse donne

$$f(x) = \int_{x_1}^x f'(t) dt + f(x_1)$$

et donc d'après l'inégalité triangulaire

$$|f(x)| \leq \left| \int_{x_1}^x f'(t) dt \right| + |f(x_1)|$$

Une fonction continue sur un intervalle fermé étant bornée, f' est bornée sur $[0;1]$, d'où par croissance de l'intégrale

$$\begin{aligned} \left| \int_{x_1}^x f'(t) dt \right| &\leq \int_{\min(x_1, x)}^{\max(x_1, x)} |f'(t)| dt \quad (\text{inégalité triangulaire}) \\ &\leq \int_{\min(x_1, x)}^{\max(x_1, x)} \|f'\|_{\infty} dt \\ &\leq \|f'\|_{\infty} \int_{\min(x_1, x)}^{\max(x_1, x)} 1 dt \end{aligned}$$

ce qui implique $\left| \int_{x_1}^x f'(t) dt \right| \leq \|f'\|_{\infty} \quad (x \in [0;1] \text{ et } x_1 \in [0;1])$

et donc

$$|f(x)| \leq \|f'\|_{\infty} + |f(x_1)|$$

Ceci étant vrai pour tout $x \in [0;1]$, on en déduit que $\|f\|_{\infty} \leq \|f'\|_{\infty} + |f(x_1)|$. En conclusion,

L'inégalité d'interpolation (I.2) est satisfaite avec $C = 1$:

$$\forall f \in \mathcal{C}^1([0;1]) \quad \|f\|_{\infty} \leq \|f'\|_{\infty} + |f(x_1)|$$

2 Soit $C \in]0;1[$. Soit f la fonction constante égale à 1 sur $[0;1]$. C'est une fonction de classe $\mathcal{C}^1$ sur $[0;1]$, de dérivée nulle, vérifiant

$$\|f\|_{\infty} = 1 > C = 0 + C = \|f'\|_{\infty} + C|f(x_1)|$$

On a donc trouvé $f \in \mathcal{C}^1([0;1])$ ne satisfaisant pas l'inégalité (I.2) avec C .

L'inégalité (I.2) est fausse pour $C < 1$.

On aurait pu prendre n'importe quelle fonction constante non nulle. Notons par ailleurs que la fonction déterminée ne dépend pas de C .

3 Dans la suite du corrigé, on se permet d'utiliser la norme infinie d'une fonction continue sur un intervalle borné sans justifier à chaque fois que cette norme est bien définie.

Soit $f \in \mathcal{C}^2([0;1])$. Définissons

$$g: \begin{cases} [0;1] \longrightarrow \mathbb{R} \\ x \longmapsto f'(x) - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \end{cases}$$

Centrale Informatique MP-PC-PSI 2022 — Corrigé

Ce corrigé est proposé par Virgile Andreani (ENS Ulm) ; il a été relu par William Aufort (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

Ce sujet traite de Formule 1. Il aborde deux façons de modéliser un circuit et étudie la gestion des résultats par une base de données relationnelle.

- La première partie présente une modélisation de circuit sous la forme d'une liste de caractères alphabétiques décrivant des lignes droites ou des virages à angle droit. Il faut écrire quelques fonctions vérifiant la validité d'un circuit, puis le tracer à l'aide de la bibliothèque graphique `turtle` dont les fonctions principales sont décrites dans l'annexe.
- La partie suivante propose une modélisation du circuit plus réaliste car elle prend en compte des virages en arc de cercle. Il faut à nouveau dessiner le circuit, d'abord avec `turtle`, ensuite au moyen de primitives de plus bas niveau.
- Dans la partie III, on modélise le temps de course. La première moitié est théorique et demande de résoudre les équations différentielles cinématiques. La seconde moitié utilise ces résultats pour calculer avec Python le temps minimal nécessaire pour finir la course.
- Enfin, la dernière partie pose quelques questions de SQL exploitant les données de six tables.

Ce sujet original est de difficulté très variable. Il associe des questions de géométrie et de physique à de la programmation, c'est en cela un bon sujet de modélisation. Malheureusement, certaines questions sont trop imprécises, et les réponses à apporter dépendent de la façon de les interpréter. Pareille latitude dans la réponse est commune dans les métiers d'ingénieur et de chercheur, mais n'est pas habituelle en prépa.

Les trois parties de Python sont très guidées mais d'une manière qui peut ne pas sembler naturelle. Le lien entre les questions n'est pas toujours évident, et leur enchaînement parfois déroutant.

Outre la répétition de tracés de circuits de Formule 1, ce sujet permet de s'entraîner au méta-jeu consistant à se mettre dans la tête des concepteurs de l'épreuve pour deviner les réponses attendues à des questions accidentellement floues, ou encore de s'attaquer à une base de données relationnelle impliquant un nombre de tables inhabituel.

INDICATIONS

- 4 On pourra se demander comment un demi-tour est modélisé en représentation minimale. Si cela aide, on peut faire l'hypothèse pour cette question que la représentation d'un circuit commence forcément par une ligne droite.
- 5 Maintenir à jour trois variables caractérisant la position et l'orientation de la voiture tout au long du circuit.
- 6 En conservant l'historique des positions atteintes pendant un parcours du circuit, on peut vérifier à chaque étape que c'est la première fois qu'on atteint ce point. Il est alors pertinent de se demander si l'on a besoin d'appeler les fonctions `contient_demi_tour1` et `est_fermé1`.
- 8 Utiliser la fonction `isinstance` donnée dans l'annexe pour vérifier le type de `e`. Attention à bien traiter tous les cas.
- 9 Il est conseillé de s'aider de dessins pour visualiser les conséquences des signes des arguments de la fonction `turtle.circle` telles que décrites par l'annexe.
- 11 Considérer le cas où le vecteur `pos` envoyé à la fonction fait un angle `dir` avec l'horizontale.
- 12 On pourra commencer par écrire les deux fonctions annexes `fin_de_virage` et `fin_de_ligne_droite` qui calculent respectivement la position de sortie d'une ligne droite ou d'un virage. Le dessin doit partir du centre de l'écran, dont la position peut être déterminée à l'aide des dimensions du tableau `s`, accessibles avec `s.shape`.
- 17 Dans ce cas, le segment est parcouru en deux parties : la première pendant laquelle le pilote accélère de v_1 jusqu'à $v_{\max}$, la seconde dans laquelle il freine de $v_{\max}$ jusqu'à v_2 .
- 18 Dans ce cas, le segment est parcouru en trois parties : les deux parties d'accélération et de freinage, séparées par une partie à vitesse constante $v_{\max}$.
- 20 Remarquer que l'on peut calculer la vitesse maximale d'entrée d'un élément si l'on connaît celle de l'élément suivant. En déduire un algorithme parcourant le circuit en sens inverse.
- 21 Commencer par gérer le cas où l'on soulève une exception `raise ValueError`. Gérer ensuite le cas où l'on peut accélérer durant toute la ligne droite. Les trois cas restants correspondent aux résultats des questions 17, 18 et 19.
- 22 Parcourir le circuit `c` et la liste de vitesses retournée par `vitesses_entrée_max` en même temps. Pour chaque élément, calculer le temps mis à le parcourir et retourner la somme de tous ces temps.
- 23 Ne pas chercher trop complexe : le corps de la fonction peut s'écrire en une ligne seulement avec un circuit bien choisi.
- 26 Utiliser les données numériques présentes dans l'introduction du problème.
- 27 Cette requête nécessite des données provenant de cinq tables. Pour calculer les différents temps des courses, additionner les temps des différents tours en question à l'aide des commandes `GROUP BY` et `SUM`.
- 29 Commencer par expliquer la sous-requête imbriquée.

I. MODÉLISATION SOMMAIRE D'UN CIRCUIT

1 Dans la représentation proposée, les virages "G" et "D" se faisant sur place, seule la lettre "A" représente des portions de route. On peut les compter à l'aide de la méthode `count` donnée à la fin de l'énoncé. Il faut ensuite multiplier leur nombre par leur longueur `d` pour obtenir la longueur totale de la route.

```
def longueur1(c: [str], d: int) -> int:
    return d * c.count("A")
```

Cette solution est la plus courte et la plus idiomatique, mais il est bien sûr possible de réimplémenter la méthode `count` manuellement, par exemple avec une boucle `for`.

```
def longueur1(c: [str], d: int) -> int:
    n = 0
    for e in c:
        if e == "A":
            n += 1
    return d * n
```

On notera que le sujet n'impose pas de recopier les annotations de type. On a fait le choix de le faire ici pour des raisons pédagogiques.

2 Comme on le voit à la ligne 5, la fonction `représentation_minimale` examine chaque élément de la liste `c` un par un. Quand ces éléments correspondent à des virages, elle incrémente ("G") ou décrémente ("D") la variable `nbg` modulo 4. Lorsqu'elle arrive sur une ligne droite, elle ajoute au résultat `res` l'élément de `virages` dont l'indice correspond à la valeur de `nbg`, remet cette variable à zéro et ajoute enfin la ligne droite à `res`. On en conclut que

À tout moment, la variable `nbg` contient le nombre de virages de 90° à gauche qu'il faut effectuer pour tourner d'un angle égal à celui résultant de la séquence de virages effectués depuis la dernière ligne droite.

Par suite,

L'expression proposée vaut `["A", "A", "D", "A"]`.

3 Comme son nom l'indique, la fonction `représentation_minimale` calcule un circuit de longueur minimale qui a la même représentation graphique que le circuit qu'on lui passe en argument.

L'énoncé est très ambigu quant aux contraintes sur les circuits. En effet, alors que la ligne 14 de la fonction montre clairement que des virages peuvent se trouver en fin de liste, l'énoncé ne précise pas s'il est autorisé pour la représentation d'un circuit de commencer par un virage. Le fait que cette contrainte n'est jamais explicitée rend difficile à évaluer s'il s'agit vraiment d'une hypothèse implicite de l'énoncé, ou d'une négligence qui rend accidentellement certaines questions beaucoup plus difficiles qu'elles ne devraient être.

Cette contrainte est importante car la fonction ne remplit pas son contrat dans le cas général. En effet, en faisant l'hypothèse raisonnable que deux circuits identiques à une rotation près sont équivalents, des virages en début

et fin d'un circuit fermé ne se simplifieraient pas, comme par exemple dans le circuit suivant : GAGAGAAGAGAD. Si les circuits ne peuvent commencer que par un "A", alors la fonction est correcte.

4 On peut détecter les demi-tours du circuit en cherchant deux lettres "G" consécutives dans sa représentation minimale. Le sujet ne précisant pas si la représentation donnée à la fonction est minimale ou non, on commence par calculer la représentation minimale au cas où. On regarde ensuite chaque paire d'éléments consécutifs à la recherche de deux lettres "G" successives.

```
def contient_demi_tour1(c: [str]) -> bool:
    c_minimale = représentation_minimale(c)
    # Vérification du circuit
    for i in range(len(c_minimale) - 1):
        if c_minimale[i:i+2] == ["G", "G"]:
            return True
    return False
```

Cette question est remarquablement peu claire, et sans doute beaucoup plus complexe à résoudre que les concepteurs du sujet ne le prévoyaient. En effet, son résultat dépend de plusieurs hypothèses qui ne sont qu'implicites. Doit-elle fonctionner pour tous les circuits, ouverts et fermés ? Dans le cas des circuits fermés, doit-on faire l'hypothèse qu'ils sont parcourus plusieurs fois ? Les circuits peuvent-ils commencer par un virage ? Chacun de ces cas implique des réponses différentes.

Essayons d'être bienveillant avec l'énoncé et de lire entre les lignes. Le fait que cette question suive immédiatement celle de la représentation minimale, et qu'une question impliquant des positions n'arrive qu'à la question 6, suggère que l'énoncé attend que l'on utilise seulement la représentation minimale. Par ailleurs, le fait que la question 5 demande d'écrire une fonction vérifiant si le circuit est fermé laisse à penser que la fonction attendue ici ne doit pas faire cette vérification. On a alors deux possibilités : soit il est attendu que la fonction `contient_demi_tour1` considère tous les circuits comme ouverts (et qu'elle a donc le droit de se tromper sur les circuits en boucle), soit qu'elle les considère tous comme fermés et parcourus plusieurs fois (et qu'elle a le droit de se tromper sur les circuits ouverts).

Le premier cas est le plus simple et était probablement celui qui était attendu (c'est la réponse que l'on donne dans ce corrigé). Mais c'est alors un exemple de mauvaise conception de sujet, puisque cette fonction n'est pas réutilisable à la question 6, qui doit être correcte pour les circuits en boucle.

Le deuxième cas, sûrement le plus naturel étant donnés les termes du sujet, est beaucoup plus complexe car il demande de gérer tous les cas particuliers impliquant des virages au début et à la fin du circuit (réfléchir par exemple à AGADADADAA, GAGADADADAAG ou encore GGAGAGAGAD). La solution la plus simple dans ce cas serait probablement de commencer par placer tous les virages initiaux à la fin du circuit, et de calculer la représentation minimale de ce nouveau circuit, équivalent au circuit donné mais où chaque demi-tour est bien équivalent à deux "G" consécutifs.

Bref, à question maladroite beaucoup de bonnes réponses, certaines faisant perdre plus de temps que d'autres. Dans ce genre de cas, aller au plus simple et briser le quatrième mur en expliquant en quelques lignes pourquoi l'énoncé est déroutant est généralement une bonne stratégie.

Centrale Informatique optionnelle MP 2022

Corrigé

Ce corrigé est proposé par Benjamin Monmege (enseignant-chercheur à l'université) ; il a été relu par Jean Starynkévitch (professeur en CPGE) et William Aufort (professeur en CPGE).

Ce sujet est entièrement concentré sur la partie « Motifs, automates et expressions » du programme d'option informatique. Il s'intéresse à trois algorithmes sur les automates finis et les expressions rationnelles.

- Dans la partie I, après quelques questions d'échauffement sur la notion de miroir d'un automate et sur les mots palindromes, l'algorithme de détermination d'un automate fini à l'aide de l'automate des parties est implémenté en OCaml. On étudie ensuite l'algorithme de Brzozowski qui permet de minimiser le nombre d'états d'un automate fini déterministe en appliquant deux fois les opérations de miroir et de détermination.
- Dans la partie suivante, on étudie les expressions rationnelles et on implémente l'algorithme de Conway permettant de prouver la réciproque du théorème de Kleene, à savoir que tout langage reconnu par un automate fini peut être décrit par une expression rationnelle. Pour cela, on emploie des matrices d'expressions rationnelles sur lesquelles on réalise des opérations de somme, de produit et d'étoile. L'algorithme de Conway utilise le paradigme « diviser pour régner » et sa complexité est analysée très finement.
- La partie III étudie le problème réciproque en proposant la construction d'un automate fini, non déterministe, à partir d'une expression rationnelle. Plutôt que la construction de l'automate de Glushkov au programme, le sujet propose la construction de l'automate d'Antimirov, qui utilise un mécanisme de dérivation formelle des expressions rationnelles. Cette dernière partie ne contient pas de question de programmation.

Ce sujet est très long et nécessite une bonne connaissance des automates et expressions rationnelles. Des questions manipulant des exemples alternent avec des questions de programmation, de complexité algorithmique et des questions plus théoriques, demandant de justifier la correction d'une construction à l'aide de récurrences. Le sujet reste parfaitement adapté au nouveau programme de l'option informatique en MP ou de l'informatique en MPI.

INDICATIONS

Partie I

- 2 Il suffit d'inverser états initiaux et finaux, et d'inverser le sens des flèches des transitions.
- 3 Après avoir formalisé la construction décrite dans l'indication précédente, on peut la justifier en montrant qu'il existe une bijection entre les chemins de l'automate A et ceux de l'automate $\tilde{A}$ dont les étiquettes sont le miroir l'une de l'autre.
- 4 Commencer par écrire une fonction récursive qui renverse chaque transition dans une liste de transitions.
- 6 Distinguer les cas d'un mot de longueur paire et de longueur impaire, en essayant de limiter le nombre de tests à effectuer à environ la moitié de la longueur du mot. Utiliser ensuite une boucle **while** ou une fonction récursive.
- 8 Raisonner par l'absurde en supposant l'existence d'un automate fini déterministe reconnaissant le langage $\text{Pal}(\Sigma) \cap \mathcal{L}(a^*ba^*)$. En notant n son nombre d'états, considérer le mot $w = a^nba^n$ et exhiber une répétition d'états parmi les $n + 1$ premiers états du chemin d'étiquette w .
- 9 Il suffit de modifier les états initiaux et finaux de l'automate A pour montrer que $L_{q,q'}$ est reconnaissable.
- 10 Raisonner par double inclusion.
- 11 Montrer que $D(\mathcal{L}(a^*b)) = \{a^n bba^n \mid n \in \mathbb{N}\}$ et $R(\mathcal{L}(a^*ba^*)) = \mathcal{L}(a^*b^*)$.
- 12 Montrer que $D(L)$ est parfois reconnaissable, et parfois ne l'est pas. Montrer, à l'aide du résultat des questions 3 et 9, que $R(L)$ est reconnaissable.
- 14 Une coquille s'est glissée dans l'énoncé puisque l'état initial de l'automate des parties est la partie I et non l'ensemble contenant une unique partie $\{I\}$.
- 16 Noter que l'automate des parties reconnaît le même langage que l'automate de départ, alors que l'automate $\tilde{A}$ reconnaît le miroir du langage reconnu par A .
- 17 Commencer par écrire une fonction récursive prenant en arguments une liste et un élément x et qui renvoie la liste de laquelle on a supprimé toutes les occurrences de x .
- 19 Commencer par utiliser le quotient de la division euclidienne de k par 2^q , puis tester la parité du résultat.
- 20 Il suffit de parcourir la liste, en utilisant les fonctions **numero** et **est_dans**, ainsi que le tableau **pow** pour chaque élément de la liste.
- 21 Utiliser la fonction **List.exists** ou écrire une fonction récursive qui parcourt la liste ℓ et utilise la fonction **est_dans** pour chaque élément de celle-ci.
- 22 Une fonction récursive fait à nouveau l'affaire, afin de parcourir la liste des transitions et faire appel, pour chaque transition, aux fonctions **etat_suivant** et **est_dans**.
- 24 S'inspirer d'un parcours en profondeur récursif. Ne pas hésiter à découper le code avec des fonctions auxiliaires, en particulier pour parcourir (et ainsi générer) l'ensemble des états de l'automate des parties (en particulier en utilisant les fonctions **etat_suivant** et **cherche**) tout en maintenant le nombre d'états générés jusqu'à et la liste de paires (k, v) décrite dans l'énoncé. Une autre fonction auxiliaire récursive permettra de repérer les états finaux dans la liste de paires générée préalablement, en utilisant la fonction **intersecte**.

- 26 À nouveau, la même coquille qu'en question 13 perturbe la lecture de l'énoncé : il faut lire $q \in \delta^*(I, u)$. La notation $\delta^*(I, u)$ décrit donc l'état X de $A_{\det}$ qu'on atteint après avoir lu le mot u . Pour répondre à la question, montrer d'abord que pour tout mot $u \in \Sigma^*$ et pour tout état $q \in \delta^*(I, u)$, il existe un état $q_0 \in I$ et une suite de transitions dans A de q_0 à q d'étiquette u . Conclure ensuite en utilisant le fait que $\tilde{A}$ est accessible.
- 27 Dans la question 26, on a montré un résultat plus fort : « si $q \in \delta^*(I, u)$, alors il existe un mot $w \in \Sigma^*$ tel que $uw \in L$ et $\tilde{A}$ possède un chemin de f à q d'étiquette w . » Quant à elle, la propriété (*) nécessite d'être corrigée, toujours à cause de la coquille de la question 13 : elle devient « si l'on prend deux mots u et v dans Σ^* tels que $u^{-1}L = v^{-1}L$, alors dans l'automate $A_{\det}$, $\delta^*(I, u) = \delta^*(I, v)$ ». La montrer en raisonnant par double inclusion et en utilisant le résultat élargi de la question 26.
- 28 Appliquer la question 27 à l'automate $\tilde{B}$, après avoir vérifié qu'il satisfait les hypothèses.

Partie II

- 31 On peut décider récursivement si le langage représenté par une expression rationnelle est vide ou non.
- 34 Écrire une fonction récursive qui applique la simplification d'abord pour les sous-expressions les plus profondes, puis utilise les fonctions `su`, `sc` ou `se` selon l'opérateur courant.
- 38 Justifier une inégalité de la forme

$$\forall n \geq 2 \quad C(n) = 2C(n-1) + O(n^2)$$

Faire apparaître un télescopage en divisant par 2^n puis sommer pour obtenir un majorant de $C(n)$ en $O(2^n)$.

- 39 Adapter la méthode de la question 38 après avoir établi une majoration de la forme

$$\forall n \geq 2 \quad C(n) \leq 4C(n/2) + O(n^3)$$

- 41 Seul l'algorithme étudié dans les questions 39 et 40 est à implémenter.
- 42 Utiliser le résultat de la question 9 et faire le lien entre le vecteur X et les états initiaux de A , ainsi qu'entre le vecteur Y et les états finaux.
- 43 Ne pas hésiter à découper le code en fonction auxiliaire récursive, permettant de construire la matrice de transitions, ainsi que les vecteurs X et Y . Finalement, utiliser les fonctions `produit` et `etoile`.

Partie III

- 45 Poursuivre le calcul des dérivées initié dans la question 44 pour générer l'ensemble d'états de l'automate d'Antimirov de E . Vous devriez trouver quatre états.
- 47 Commencer par considérer le cas d'un mot w réduit à une lettre. Utiliser ensuite la propriété admise par l'énoncé pour se ramener au cas où S est un singleton $\{E\}$. Prouver alors la propriété par récurrence sur la longueur du mot w . Dans la récurrence, on utilisera le résultat de la question 46.
- 48 Montrer la propriété par récurrence sur la longueur du mot w .
- 49 Conclure à l'aide des questions 47 et 48.
- 50 Procéder par induction sur la structure de l'expression rationnelle E (ou par récurrence sur le nombre d'opérations utilisées dans E). En déduire un majorant du nombre d'états de l'automate d'Antimirov.

I. MOTS ET AUTOMATES

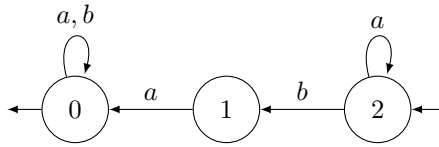
1 L'automate $\mathcal{A}_1$ est un automate non déterministe qui reconnaît tous les mots sur l'alphabet $\{a, b\}$ qui commencent par un nombre quelconque de lettres a et b , suivies du mot ab , puis d'éventuelles lettres a .

On peut décrire ce langage par l'expression rationnelle $(a + b)^* ab a^*$.

Son langage miroir consiste donc en l'ensemble des mots qui commencent par un nombre arbitraire de a , suivi du mot ba , puis d'un mot quelconque composé des lettres a et b .

$$L_1 = \{waba^n \mid w \in \{a, b\}^*, n \in \mathbb{N}\} \quad \text{et} \quad \tilde{L}_1 = \{a^n baw \mid w \in \{a, b\}^*, n \in \mathbb{N}\}$$

2 On obtient l'automate $\tilde{\mathcal{A}}_1$ ci-dessous qui reconnaît le langage miroir $\tilde{L}_1$ en inversant états initiaux et finaux et en inversant le sens des transitions :



Notons que cet automate est déterministe, alors que l'automate $\mathcal{A}_1$ est non déterministe.

3 Généralisons l'approche appliquée dans la question 2, en construisant l'automate miroir $\tilde{A} = (Q, I', F', T')$ de l'automate non déterministe $A = (Q, I, F, T)$ en définissant :

- l'ensemble $I' = F$ des états initiaux de $\tilde{A}$ comme les états finaux de A ;
- l'ensemble $F' = I$ des états finaux de $\tilde{A}$ comme les états initiaux de A ;
- l'ensemble $T' = \{(q', a, q) \mid (q, a, q') \in T\}$ de transitions de $\tilde{A}$ obtenu en inversant le sens des transitions de A .

Montrons que l'automate miroir $\tilde{A}$ reconnaît le langage miroir $\tilde{L}$. Pour cela, exhibons une bijection f entre les chemins c dans l'automate A d'un état de I à un état de F et les chemins c' dans l'automate $\tilde{A}$ d'un état de I' à un état de F' tel que si le chemin c est étiqueté par un mot w alors le chemin c' est étiqueté par le mot $\tilde{w}$. Pour tout chemin c dans A décrit par la suite de transitions

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \cdots q_{n-1} \xrightarrow{a_{n-1}} q_n$$

définissons la suite $f(c)$ par

$$q_n \xrightarrow{a_{n-1}} q_{n-1} \xrightarrow{a_{n-2}} \cdots q_1 \xrightarrow{a_0} q_0$$

Tout d'abord notons que si $w = a_0 a_1 \cdots a_{n-1}$ est l'étiquette du chemin c , alors le mot $a_{n-1} a_{n-2} \cdots a_0 = \tilde{w}$ est bien l'étiquette de la suite $f(c)$. Montrons ensuite que $f(c)$ est un chemin dans $\tilde{A}$:

- il commence dans l'état q_n qui est final dans A et donc initial dans $\tilde{A}$;
- il termine dans l'état q_0 qui est initial dans A donc final dans $\tilde{A}$;
- pour tout $i \in \llbracket 0; n-1 \rrbracket$, (q_i, a_i, q_{i+1}) est une transition de A , donc (q_{i+1}, a_i, q_i) est une transition de $\tilde{A}$.

On a donc bien toutes les propriétés assurant que $f(c)$ est un chemin de $\tilde{A}$ d'étiquette $\tilde{w}$.

Mines Maths 1 MP 2022 — Corrigé

Ce corrigé est proposé par Hicham Assakaf (ENS Paris-Saclay) ; il a été relu par Thierry Limoges (enseignant en CPGE) et Gilbert Monna (professeur honoraire en CPGE).

On appelle partition d'un entier $n \geq 1$ une décomposition de n en une somme d'entiers naturels non nuls. En 1918, Hardy et Ramanujan ont donné un développement asymptotique du nombre de partitions de n quand n tend vers $+\infty$. Le sujet propose de redémontrer ce résultat.

- La partie A est consacrée à la construction d'une fonction P que l'on utilisera dans tout le problème.
- La partie B définit le nombre p_n de partitions d'un entier n puis établit que le développement en série entière de P est $\sum p_n z^n$. On obtient finalement une expression explicite de p_n . La preuve de la formule de Hardy et Ramanujan se montrera en étudiant le comportement asymptotique des différentes composantes intervenant dans l'expression de p_n .
- La partie suivante fournit un majorant simple du module de l'intégrande intervenant dans l'expression de p_n . Ce résultat servira à prouver qu'une intégrale similaire à celle intervenant dans l'expression de p_n vérifie l'hypothèse de domination du théorème de convergence dominée.
- La partie D est indépendante des précédentes. Elle montre des résultats techniques qui seront utilisés dans les parties suivantes.
- La partie E utilise un cadre probabiliste pour démontrer certaines majorations. Les premières questions donnent une majoration de $E((X - E(X))^4)$ lorsque X suit une loi géométrique. Grâce au résultat du début de la partie, on obtient, sous certaines conditions, une nouvelle majoration de la distance entre la fonction caractéristique d'une variable aléatoire Y centrée et une gaussienne. Dans la partie suivante, on utilisera ce résultat pour montrer l'hypothèse de convergence simple du théorème de convergence dominée.
- La partie F introduit plusieurs notations utilisant les résultats des parties précédentes. Les premières questions montrent que les différentes hypothèses du théorème de convergence dominée sont vérifiées. On utilise ensuite ce théorème pour prouver qu'une intégrale ressemblant à celle intervenant dans l'expression de p_n converge vers une gaussienne.
- La dernière partie conclut en démontrant la formule de Hardy et Ramanujan.

Comme on le voit, le problème est très technique, et nécessite d'être au point sur les intégrales à paramètre, les séries de fonctions, ou encore les séries entières. Les manipulations des objets contenus dans ce problème sont délicates. Le sujet constitue un bon entraînement au calcul et à l'application des théorèmes d'analyse.

INDICATIONS

2 Dériver la fonction $t \mapsto (1 - tz)e^{L(tz)}$.

3 Remarquer que $-\ln(1 - |z|^n) = O(|z|^n)$ pour tout z dans D .

4 Montrer que si $(x_1, x_2, \dots, x_N) \in P_{n,N}$, alors pour tout $k \in \llbracket 1; n \rrbracket$, $x_k \leq n$. Justifier que l'application

$$\phi: \begin{cases} P_{n,N} & \longrightarrow P_{n,N+1} \\ (a_1, \dots, a_N) & \longmapsto (a_1, \dots, a_N, 0) \end{cases}$$

est injective. Montrer que la fonction ϕ est surjective si $N \geq \max(1, n)$.

5 Établir que

$$P_{n,N+1} = \bigcup_{m=0}^n \left\{ (a_1, \dots, a_N, m) \in \mathbb{N}^N \mid \sum_{k=1}^N ka_k = n - m(N+1) \right\}$$

En déduire que $p_{n,N+1} = \sum_{m=0}^{\lfloor \frac{n}{N+1} \rfloor} p_{n-m(N+1), N}$.

8 Expliquer que $\sum_{n=1}^{+\infty} \frac{x^n}{n} (1 - \cos(n\theta)) \geq x(1 - \cos \theta)$.

9 Distinguer les cas $(1 - x)^2 \leq x(1 - \cos \theta)$ et $(1 - x)^2 \geq x(1 - \cos \theta)$.

12 Remarquer que si $x > 0$, $\frac{e^{-x}}{1 - e^{-x}} = \sum_{k=1}^{+\infty} e^{-kx}$.

18 Utiliser la formule du binôme de Newton, et utiliser la question 17 en faisant apparaître $|E(X^k) - 1/p^k|$ pour $1 \leq k \leq 4$.

22 Raisonner par récurrence en remarquant que

$$\left(\prod_{k=1}^{n+1} z_k - \prod_{k=1}^{n+1} u_k \right) = z_{n+1} \left(\prod_{k=1}^n z_k - \prod_{k=1}^n u_k \right) + \prod_{k=1}^n u_k (z_{n+1} - u_{n+1})$$

25 Montrer que $\phi: \begin{cases} [-\pi; \pi] & \longrightarrow \mathbb{R} \\ \theta & \longmapsto \begin{cases} \frac{1 - \cos \theta}{\theta^2} & \text{si } \theta \neq 0 \\ 1/2 & \text{sinon} \end{cases} \end{cases}$ est continue

et en déduire qu'elle admet un minimum. Utiliser la question 9.

27 Montrer que l'on peut effectuer le changement de variable $v = u\sigma_{t_n}$, et faire apparaître l'intégrale de la question 26.

A. FONCTIONS L ET P

1 | Très souvent, pour prouver qu'une série converge ou qu'une fonction f est intégrable au voisinage de $+\infty$, montrer que le terme général est un $o(1/n^2)$ ou que $f(x) = \underset{x \rightarrow +\infty}{o}(1/x^2)$ est une méthode simple et efficace.

Soit $z \in D$. Alors $|z| < 1$, donc par croissances comparées,

$$\frac{|z|^n}{n} \cdot n^2 = n|z|^n \xrightarrow{n \rightarrow \infty} 0$$

Par conséquent, $\frac{|z|^n}{n} = o\left(\frac{1}{n^2}\right)$ et la série $\sum \frac{z^n}{n}$ converge absolument. En particulier,

La série $\sum z^n/n$ converge.

Si $z \in]-1; 1[$, alors $\sum_{n=1}^{+\infty} z^n/n$ est le développement en série entière de $-\ln(1-z)$, donc

$$\sum_{n=1}^{+\infty} \frac{z^n}{n} = -\ln(1-z)$$

2 Soit $z \in D$. Posons $f: \begin{cases} [0; 1] \longrightarrow \mathbb{R} \\ t \longmapsto L(tz) \end{cases}$

Montrons dans un premier temps que la fonction f est bien définie. Soit $t \in [0; 1]$. Alors $|tz| = t|z| \leq |z| < 1$, donc tz appartient à D , et $L(tz)$ existe d'après la question précédente. Montrons maintenant que f est dérivable sur $[0; 1]$. Posons,

$$\forall n \geq 1 \quad u_n: \begin{cases} [0; 1] \longrightarrow \mathbb{R} \\ t \longmapsto \frac{(tz)^n}{n} \end{cases}$$

Soit $n \geq 1$. Alors u_n est dérivable sur $[0; 1]$ et pour tout $t \in [0; 1]$, $u_n'(t) = z^n t^{n-1}$. Par conséquent, $\|u_n'\|_\infty = |z|^n$, et puisque $z \in D$,

$$\sum_{n=1}^{+\infty} \|u_n'\|_\infty < \infty$$

Ainsi la série de fonctions $\sum u_n'$ converge normalement donc uniformément, d'où

$$\sum_{n=1}^{+\infty} u_n' = \sum_{n=1}^{+\infty} z^n t^{n-1} = \sum_{n=0}^{+\infty} t^n z^{n+1} = z \sum_{n=0}^{+\infty} t^n z^n = \frac{z}{1-tz}$$

et donc La fonction f est dérivable et pour tout $t \in [0; 1]$, $f'(t) = \frac{z}{1-tz}$.

En posant $\psi: \begin{cases} [0; 1] \longrightarrow \mathbb{R} \\ t \longmapsto (1-tz)e^{f(t)} \end{cases}$

pour tout $t \in [0; 1]$, $\psi(t) = (1-tz)e^{f(t)}$, et la fonction ψ est dérivable sur $[0; 1]$. De plus, pour tout $t \in [0; 1]$,

$$\begin{aligned} \psi'(t) &= f'(t)(1-tz)e^{f(t)} - ze^{f(t)} \\ &= \frac{z}{1-tz}(1-tz)e^{f(t)} - ze^{f(t)} \\ &= ze^{f(t)} - ze^{f(t)} \\ \psi'(t) &= 0 \end{aligned}$$

Puisqu'on travaille sur l'intervalle $[0; 1]$, ce calcul prouve que

La fonction ψ est constante sur $[0; 1]$.

En particulier,

$$\psi(1) = \psi(0)$$

$$(1 - z)e^{L(z)} = e^{L(0)}$$

$$(1 - z)e^{L(z)} = 1$$

$$e^{L(z)} = \frac{1}{1 - z}$$

3 Soit $z \in D$. Puisque $|z| \in]-1; 1[$, la série $\sum |z|^n/n$ converge. En appliquant l'inégalité triangulaire,

$$\left| \sum_{n=1}^{+\infty} \frac{z^n}{n} \right| \leq \sum_{n=1}^{+\infty} \frac{|z|^n}{n}$$

d'où

$$|L(z)| \leq -\ln(1 - |z|)$$

Par conséquent, puisque $|z|^n \xrightarrow{n \rightarrow \infty} 0$,

$$|L(z^n)| \leq -\ln(1 - |z|^n) \underset{n \rightarrow +\infty}{\sim} |z|^n$$

Or, $\sum |z|^n$ converge, et en appliquant le théorème de comparaison des séries à termes positifs, $\sum -\ln(1 - |z|^n)$ converge. Ainsi, la série $\sum L(z^n)$ converge absolument, et on en déduit que

La série $\sum L(z^n)$ converge.

B. DÉVELOPPEMENT DE P EN SÉRIE ENTIÈRE

4 Soient $n \in \mathbb{N}$, $N \in \mathbb{N}^*$, $(a_1, \dots, a_N) \in P_{n,N}$, ainsi que $\ell \in \llbracket 1; N \rrbracket$. Alors

$$a_\ell \leq \ell a_\ell \leq \sum_{k=1}^N k a_k = n$$

En conclusion, $a_\ell \leq n$ pour tout $\ell \in \llbracket 1; N \rrbracket$ et $P_{n,N} \subset \llbracket 0; n \rrbracket^N$. Puisque $\llbracket 0; n \rrbracket^N$ est de cardinal $(n+1)^N$,

L'ensemble $P_{n,N}$ est de cardinal fini pour tout $N \in \mathbb{N}^*$.

Posons

$$\phi: \begin{cases} P_{n,N} & \longrightarrow P_{n,N+1} \\ (a_1, \dots, a_N) & \longmapsto (a_1, \dots, a_N, 0) \end{cases}$$

La fonction ϕ est bien définie. En effet, si $(a_1, \dots, a_N) \in P_{n,N}$, alors $\sum_{k=1}^N k a_k = n$ d'où

$$\sum_{k=1}^N k a_k + (N+1) \cdot 0 = n$$

Mines Maths 2 MP 2022 — Corrigé

Ce corrigé est proposé par Jean-Paul Bonnet (professeur en CPGE) ; il a été relu par Bertrand Wiel (professeur en CPGE) et Jean Starynkévitch (professeur en CPGE).

Le sujet s'intéresse aux fonctions $f_A : \mathbb{R} \rightarrow \mathrm{GL}_n(\mathbb{K}), t \mapsto e^{tA}$ où A est une matrice fixée. L'image de f_A définit alors un sous-groupe de $\mathrm{GL}_n(\mathbb{K})$, qui est un cas particulier de sous-groupe dit à 1 paramètre. L'énoncé s'en sert pour décrire l'espace tangent en l'identité d'un sous-groupe fermé de $\mathrm{GL}_n(\mathbb{K})$ appelé algèbre de Lie, puis pour exprimer la solution d'une équation différentielle vectorielle.

- Dans la partie I, on étudie les propriétés des sous-groupes à 1 paramètre. Il s'agit essentiellement de résultats sur les exponentielles de matrice qui seront exploités dans les deux parties suivantes.
- Ensuite, on démontre la formule de (Lie-)Trotter-Kato permettant de calculer l'exponentielle de la somme de deux matrices qui ne commutent pas. C'est un résultat technique dont l'usage sera fait plus tard.
- En troisième partie, on définit l'algèbre de Lie d'un sous-groupe fermé du groupe linéaire réel et on l'identifie à l'espace vectoriel tangent en l'identité à ce sous-groupe dans le cas particulier des groupes $\mathrm{SL}_n(\mathbb{R})$ et $\mathrm{O}_n(\mathbb{R})$.
- La dernière partie, largement indépendante des trois précédentes, traite du comportement asymptotique des solutions de l'équation différentielle vectorielle $X' = AX$. On y démontre que ce comportement dépend du signe des parties réelles des valeurs propres de A . Pour cela, on utilise une décomposition de l'espace vectoriel en lien avec le polynôme caractéristique de A .

Ce sujet mélange habilement algèbre, calcul différentiel et fonctions à valeurs vectorielles. Il utilise au fil de l'eau des résultats sur les espaces vectoriels normés sans en faire un sujet d'étude central. De nombreuses questions demandent une bonne part d'autonomie pour justifier rigoureusement chaque étape du raisonnement. Il demande d'établir plusieurs résultats classiques des écrits ou des oraux de concours, parmi lesquels $\|e^A\| \leq e^{\|A\|}$, $\det(e^A) = e^{\mathrm{Tr}(A)}$, une version faible de la décomposition de Dunford d'une matrice, etc. Il s'agit d'un bon sujet de synthèse pour travailler ces différentes notions.

INDICATIONS

Partie I

- 1 Par récurrence sur les sommes partielles.
- 2 Utiliser l'unicité de la solution à un problème de Cauchy. Remarquer ensuite que si $A = 0_n$, e^{tB} est inversible d'inverse e^{-tB} .
- 3 Penser à évaluer l'identité en $t = 0$.
- 5 Trigonaliser la matrice dans $\mathbb{C}$ et justifier que le résultat est également vrai dans $\mathbb{R}$.

Partie II

- 6 Exploiter la question 4.
- 7 Utiliser un développement limité à l'ordre 1 de la fonction h .
- 8 Remarquer que $e^{A+B} = Y_k^k$.

Partie III

- 9 Utiliser la question 5.
- 10 Justifier que $(e^{tM})^\top = e^{tM^\top}$ pour un sens et dériver $f_M f_{M^\top}$ pour la réciproque.
- 11 Utiliser le caractère fermé de G et l'inégalité de Trotter-Kato pour la stabilité de la somme.
- 12 Interpréter $e^{tA} B e^{-tA}$ comme un changement de base et utiliser le comportement de l'exponentielle par changement de base.
- 13 Justifier que u est dérivable en 0 puis utiliser la caractérisation séquentielle de la limite et le caractère fermé de G .
- 15 Travailler avec une matrice triangulaire et expliciter le calcul du déterminant dans ce cas.
- 17 Utiliser la question 16 pour $SL_n(\mathbb{R})$ et dériver $t \mapsto \gamma(t)\gamma(t)^\top$ en 0 pour $O_n(\mathbb{R})$.

Partie IV

- 18 Identifier le polynôme caractéristique de A puis effectuer une trigonalisation. Utiliser un binôme de Newton pour l'expression de T^k .
- 19 Utiliser un vecteur propre associé à la valeur propre de A de partie réelle α pour déterminer un vecteur propre de e^{tA} et se ramener à une exponentielle réelle.
- 20 Penser au théorème de décomposition des noyaux.
- 21 Il s'agit d'une version faible de la décomposition de Dunford.
- 22 Utiliser la question 21 et le fait que $f_A = f_D f_N$, que $\|f_A\| \leq \|f_D\| \|f_N\|$ et majorer séparément. Conclure en exploitant l'équivalence des normes avec $\|\cdot\|_\infty$ sur $\mathcal{M}_n(\mathbb{K})$.
- 24 Trigonaliser A et utiliser le fait que que l'on connaît la diagonale de l'exponentielle d'une matrice triangulaire pour effectuer un raisonnement par l'absurde.
- 25 Le théorème de décomposition des noyaux donne la somme directe. Construire des endomorphismes définis suivant la somme directe pour obtenir une décomposition de la forme $A = A_s + A_i + A_n$ telle que, si $X \in \mathbb{C}^n$ se décompose comme $X_s + X_n + X_i$ selon $E_s \oplus E_n \oplus E_i$, alors $e^{tA}X$ admet pour décomposition selon la même somme directe $e^{tA_s}X_s + e^{tA_n}X_n + e^{tA_i}X_i$.
- 26 Procéder par double inclusion en exploitant la démarche de la question 25 pour chacune d'elle. Pour la réciproque, introduire une matrice $A - \mu I_n$ pour une valeur bien choisie de μ .

I. QUESTIONS PRÉLIMINAIRES

1 On montre par récurrence sur $k \in \mathbb{N}$, la propriété

$$\mathcal{P}(k) : \quad AB^k = B^k A$$

- $\mathcal{P}(0)$ est vraie. En effet, $B^0 = I_n$ donc $AB^0 = AI_n = A = I_n A = B^0 A$.
- $\mathcal{P}(k) \implies \mathcal{P}(k+1)$: soit $k \in \mathbb{N}$, on suppose $\mathcal{P}(k)$.

$$\begin{aligned} AB^{k+1} &= AB^k B && \text{(associativité)} \\ &= B^k AB && \text{(hyp. de récurrence)} \\ &= B^k BA && \text{(commutativité)} \\ AB^{k+1} &= B^{k+1} A && \text{(associativité)} \end{aligned}$$

Donc $\mathcal{P}(k+1)$ est vraie.

- **Conclusion** : $\forall k \in \mathbb{N} \quad AB^k = B^k A$

Par bilinéarité du produit matriciel

$$\forall m \in \mathbb{N} \quad A \left(\sum_{k=0}^m \frac{B^k}{k!} \right) = \sum_{k=0}^m \frac{AB^k}{k!} = \sum_{k=0}^m \frac{B^k A}{k!} = \left(\sum_{k=0}^m \frac{B^k}{k!} \right) A$$

Comme le produit matriciel est une application bilinéaire entre $\mathcal{M}_n(\mathbb{K})^2$ et $\mathcal{M}_n(\mathbb{K})$ qui sont de dimensions finies, c'est une application continue. Ainsi, par passage à la limite quand $m \rightarrow +\infty$, le terme de gauche converge vers Ae^B alors que celui de droite converge vers $e^B A$ d'où,

$$AB = BA \implies Ae^B = e^B A$$

La suite de terme général $\sum_{k=0}^m B^k/k!$ est à valeurs dans le sous-espace vectoriel $\mathbb{K}[B]$ qui est dimension finie, donc fermé dans $\mathcal{M}_n(\mathbb{K})$. Par suite, la matrice e^B appartient à $\mathbb{K}[B]$ et il existe alors un polynôme $P(X) \in \mathbb{K}[X]$ tel que $e^B = P(B)$. La bilinéarité du produit matriciel et le résultat de récurrence ci-dessus entraînent alors que $AP(B) = P(B)A$.

2 D'après l'énoncé, f_A vérifie sur $\mathbb{R}$ l'équation différentielle $y' = Ay$ et $f_A(0) = I_n$ donc f_A est la solution du problème de Cauchy

$$\begin{cases} \forall t \in \mathbb{R} & y'(t) = Ay(t) \\ & y(0) = I_n \end{cases}$$

Remarquons maintenant que, pour tout $t \in \mathbb{R}$, $g(t) = f_{A+B}(t)f_{-B}(t)$. Les applications f_{A+B} et f_A étant de classe $\mathcal{C}^1$ sur $\mathbb{R}$ d'après l'énoncé, $g \in \mathcal{C}^1(\mathbb{R}, \mathcal{M}_n(\mathbb{K}))$ par les théorèmes usuels et, pour tout $t \in \mathbb{R}$, on a

$$\begin{aligned} g'(t) &= f'_{A+B}(t)f_{-B}(t) + f_{A+B}(t)f'_{-B}(t) \\ &= (A+B)f_{A+B}(t)f_{-B}(t) - f_{A+B}(t)Bf_{-B}(t) \end{aligned}$$

Comme A et B commutent, A et $A+B$ également. Ainsi, avec la question 1, pour tout $t \in \mathbb{R}$, les matrices B et $f_{A+B}(t)$ commutent aussi. Par suite, pour tout $t \in \mathbb{R}$,

$$\begin{aligned} g'(t) &= (A+B)f_{A+B}(t)f_{-B}(t) - Bf_{A+B}(t)f_{-B}(t) \\ &= (A+B)g(t) - Bg(t) \end{aligned}$$

$$g'(t) = Ag(t)$$

En outre, $g(0) = f_{A+B}(0)f_{-B}(0) = I_n \times I_n = I_n$. En conclusion,

La fonction g vérifie le même problème de Cauchy que f_A sur $\mathbb{R}$.

L'application $y \mapsto Ay$ étant un endomorphisme de $\mathcal{M}_n(\mathbb{K})$, d'après le théorème de Cauchy linéaire, ce problème de Cauchy admet une unique solution. Il en résulte que $f_A = g = f_{A+B}f_{-B}$.

Soit $t \in \mathbb{R}$. En répliquant ce qui précède dans le cas particulier où $A = 0_n$, on obtient que $f_B(t) \times f_{-B}(t) = f_{0_n}(t) = I_n$. On en déduit que $f_{-B}(t) \in GL_n(\mathbb{K})$ et que son inverse est $f_B(t)$. Il s'ensuit que

$$\forall t \in \mathbb{R} \quad e^{t(A+B)} = e^{tA}e^{tB}$$

3 Remarquons tout d'abord que, comme la fonction f_A vérifie sur $\mathbb{R}$ l'équation différentielle $y' = Ay$, f est $\mathcal{C}^1$ et vérifie $f_A' = Af_A$. Ainsi, on en déduit par la linéarité de $X \mapsto AX$, que f_A' est $\mathcal{C}^1$ sur $\mathbb{R}$, donc $f_A \in \mathcal{C}^2(\mathbb{R}, \mathcal{M}_n(\mathbb{K}))$ et

$$f_A'' = (Af_A)' = Af_A' = A(Af_A) = A^2f_A$$

On peut même prouver par récurrence qu'elle est de classe $\mathcal{C}^\infty$ sur $\mathbb{R}$ mais ce n'est pas nécessaire.

Cette remarque s'applique également à f_B et f_{A+B} . Ainsi, f_{A+B} et $f_A \times f_B$ sont toutes deux de classe $\mathcal{C}^2$ sur $\mathbb{R}$ et coïncident sur $\mathbb{R}$. On a alors d'une part,

$$(f_{A+B})'' = (A+B)^2f_{A+B}$$

d'autre part, comme f_A et f_B commutent, avec la formule de Leibniz, on a

$$\begin{aligned} (f_A f_B)'' &= f_A'' f_B + 2f_A' f_B' + f_A f_B'' \\ &= A^2 f_A f_B + 2A f_A B f_B + f_A B^2 f_B \end{aligned}$$

En évaluant cette identité en $t = 0$, avec $f_{A+B}(0) = f_A(0) = f_B(0) = I_n$, on obtient

$$\begin{aligned} (A+B)^2 &= A^2 + 2AB + B^2 \text{ donc } A^2 + AB + BA + B^2 = A^2 + 2AB + B^2 \\ \text{donc } BA &= AB \end{aligned}$$

En conclusion,

Les matrices A et B commutent.

4 Dans cette question, on utilisera les propriétés (N1) et (N2) vérifiées par la norme de l'énoncé. Montrons tout d'abord par récurrence que la propriété

$$\mathcal{P}(m) : \quad \|A^m\| \leq \|A\|^m$$

est vraie pour tout $m \in \mathbb{N}$.

- $\mathcal{P}(0)$ est vraie car $\|A^0\| = \|I_n\| = \|A\|^0$ puisque d'après la propriété (N1), on a $\|I_n\| = 1$.
- $\mathcal{P}(m) \implies \mathcal{P}(m+1)$: soit $m \in \mathbb{N}$. Supposons que $\mathcal{P}(m)$ est vraie.

$$\begin{aligned} \text{Alors,} \quad \|A^{m+1}\| &= \|A^m A\| \\ &\leq \|A^m\| \|A\| && \text{(d'après (N2))} \\ &\leq \|A\|^m \|A\| && (\mathcal{P}(m)) \\ \|A^{m+1}\| &\leq \|A\|^{m+1} \end{aligned}$$

- Conclusion : $\forall m \in \mathbb{N} \quad \|A^m\| \leq \|A\|^m$

Mines Informatique MP-PC-PSI 2022 — Corrigé

Ce corrigé est proposé par Cyril Ravat (professeur en CPGE) ; il a été relu par Virgile Andreani (ENS Ulm) et Benjamin Monmege (enseignant-chercheur à l'université).

Ce sujet aborde le thème du magnétisme des matériaux et la modélisation de son aspect aléatoire. Il contient une brève introduction décrivant la différence entre les matériaux paramagnétiques et ferromagnétiques, ainsi que l'importance de la température de Curie dans le comportement du matériau. Il est constitué de 4 parties indépendantes.

- Dans la partie I, on étudie l'évolution théorique de l'aimantation en fonction de la température. Cette évolution est définie à l'aide d'une relation non linéaire qu'il faut résoudre numériquement, ce qui donne l'occasion d'appliquer l'algorithme de recherche par dichotomie.
- La partie suivante fait intervenir des bases de données économiques (fournisseurs, matériaux, prix au kilogramme). Quatre questions de difficulté graduelle permettent de montrer ses compétences.
- La partie III, plus longue, étudie le modèle microscopique de l'aimantation à partir des sens *up* et *down* des spins présents à l'intérieur du matériau. L'échantillon est modélisé par un tableau à deux dimensions, où l'on cherche comment déterminer les éléments voisins notamment au bord de l'échantillon. Une recherche d'états typiques est réalisée par la méthode aléatoire de Monte-Carlo.
- Dans la dernière partie, il est question de reconnaissance de formes au sein des images produites précédemment. On réalise en particulier un marquage des domaines magnétiques dits de Weiss par la recherche des pixels voisins identiques. Une question aborde les fonctions récursives, une autre les piles.

Cette épreuve amène correctement un sujet complexe et le traite sous plusieurs aspects intéressants. Elle balaye l'ensemble du programme, avec des questions de difficulté progressive.

INDICATIONS

Partie I

- 3 Il faut reprendre l'algorithme de la recherche par dichotomie vu en cours et l'appliquer à cette fonction particulière qui possède un deuxième argument, constant.
- 4 La longueur de l'intervalle de recherche est divisée par 2 à chaque itération. Combien d'itérations sont réalisées avant d'arriver à la bonne précision ?
- 5 La construction de la liste est classique, réalisée à l'aide de la méthode `append`. Attention, deux cas sont possibles en fonction de la valeur de t .

Partie II

- 7 Des colonnes de deux tables différentes sont nécessaires, il faut faire une jointure.
- 8 On doit récupérer le prix minimum et s'en servir pour sélectionner les bonnes lignes de la table obtenue par jointure. L'utilisation de l'opérateur `ORDER BY` ne donne pas le bon résultat ici.
- 9 On peut générer la table contenant le prix moyen de tous les matériaux, puis y sélectionner uniquement les prix moyens voulus en imbriquant la requête précédente à l'intérieur d'une nouvelle requête.

Partie III

- 11 Il est possible d'adapter le code de la question précédente pour modifier les valeurs une ligne sur deux.
- 12 L'extraction des valeurs de s entre les indices a et b exclu se fait par `s[a:b]`.
- 13 Le plus simple est de traiter le cas général en premier, puis d'ajouter une modification de chaque voisin dans le cas où le spin se trouve sur un des bords de l'échantillon.
- 14 L'équation 3 de l'énoncé montre la nécessité d'une double boucle, sur l'ensemble des spins puis sur les voisins de chaque spin. On doit récupérer les voisins avec la fonction définie à la question 13.
- 15 Il faut retourner une valeur booléenne, vraie dans deux cas disjoints.
- 16 Quelle est la complexité de la fonction `energie` ?
- 17 Le tirage au hasard du spin à modifier est à réaliser avec la fonction `randrange`.
- 19 Comme à la question 18, la complexité du calcul de ΔE est essentiel.
- 20 Quelle est la différence d'énergie ΔE lorsqu'un spin change ? Peut-on simplifier le calcul, supprimer les sommes ?

Partie IV

- 22 La fonction `explorer_voisinage` regarde simplement chaque pixel voisin de i pour éventuellement en modifier la couleur et exécuter à nouveau la fonction sur ce pixel. On ne demande pas la fonction qui crée une nouvelle couleur.
- 23 L'utilisation d'une pile se fait à l'aide d'une boucle itérant tant que la pile est non vide. La pile doit être initialisée avec la valeur i .
- 24 Il faut exécuter la fonction précédente pour chaque pixel non encore traité, sans oublier d'incrémenter `num` à chaque fois.

I. TRANSITION PARAMAGNÉTIQUE/FERROMAGNÉTIQUE SANS CHAMP MAGNÉTIQUE EXTÉRIEUR

1 Pour importer uniquement quelques fonctions à partir de modules, il faut utiliser la syntaxe

```
from math import exp, tanh
from random import randrange, random
```

2 L'équation donnée doit être résolue avec la variable m et se met sous la forme

$$f(x, t) = x - \tanh\left(\frac{x}{t}\right) = 0$$

d'inconnue x .

```
def f(x, t):
    return x - tanh(x/t)
```

3 On utilise la recherche par dichotomie d'une solution à l'équation $f(x, t) = 0$ où la variable est x alors que t est un paramètre fixé. Dans cet algorithme, on conserve deux valeurs correspondant aux deux bornes de l'intervalle dans lequel se trouve la solution, initialement $[a, b]$. À chaque itération, on divise la largeur de l'intervalle par 2, car en appelant m le milieu de l'intervalle,

- soit $f(a, t)$ et $f(m, t)$ sont de signes opposés et cela signifie que la solution se trouve entre a et m , on peut donc affecter m à la variable b ;
- soit $f(a, t)$ et $f(m, t)$ sont de même signe et cela signifie que la solution ne se trouve pas entre a et m , on peut donc affecter m à la variable a .

On continue les itérations jusqu'à ce que la largeur de l'intervalle soit inférieure à 2ε pour obtenir un résultat final égal au milieu de l'intervalle. La solution réelle est nécessairement à une distance inférieure à ε de ce résultat.

```
def dichot(f, t, a, b, eps):
    while b - a > 2*eps:
        m = (a + b) / 2
        if f(a, t) * f(m, t) < 0:
            b = m
        else:
            a = m
    return (a + b) / 2
```

4 À chaque itération, la largeur de l'intervalle de recherche est divisée par 2 à l'aide d'un nombre d'opérations toujours identique et indépendant de a et b . En nommant n le nombre d'itérations, on peut dire que l'algorithme est de complexité linéaire en n . Au bout de n itérations, la largeur de l'intervalle de recherche est $(b - a) / 2^n$. Il faut ainsi, pour sortir de la boucle, que

$$\frac{b - a}{2^n} \leq 2\varepsilon$$

c'est-à-dire $\frac{b - a}{\varepsilon} \leq 2^{n+1}$

soit $n \geq \log_2\left(\frac{b - a}{\varepsilon}\right) - 1$

donc

La complexité de la fonction `dichot` est en $O\left(\log_2\left(\frac{b - a}{\varepsilon}\right)\right)$.

5 Il faut, pour chacune des 500 valeurs t uniformément réparties entre t_1 et t_2 , appliquer la fonction `dicho` si $t < 1$ (le matériau est alors ferromagnétique), ou simplement définir une aimantation nulle si $t \geq 1$ (le matériau est alors paramagnétique).

```
def construction_liste_m(t1, t2):
    m = []
    n = 500
    pas = (t2 - t1) / (n - 1)
    for i in range(n):
        t = t1 + pas * i
        if t < 1:
            m.append( dicho(f, t, 0.001, 1, 1e-6) )
        else:
            m.append(0)
    return m
```

La figure 1 de l'énoncé peut être obtenue avec le code

```
t = [ 1.5*t/500 for t in range(1,501) ]
m = construction_liste_m(t[0],t[-1])
plt.plot(t,m)
plt.grid()
plt.show()
```

II. RECHERCHE DANS UNE BASE DE DONNÉES DE MATÉRIAUX MAGNÉTIQUES

6 Pour obtenir le nom des matériaux ayant une température de Curie inférieure à 500 kelvins, il faut exécuter la sélection

```
SELECT nom FROM materiaux WHERE t_curie < 500;
```

7 Les noms de fournisseurs sont dans la table `fournisseurs` alors que les prix sont dans la table `prix`. Il faut donc réaliser une jointure avant la sélection, soit la requête

```
SELECT nom_fournisseur, prix_kg * 4.5
FROM fournisseurs JOIN prix ON id_fournisseur = id_four
WHERE id_mat = 8713;
```

8 Le prix minimal peut être obtenu par l'utilisation de la fonction d'agrégation `MIN` mais pas le fournisseur correspondant, car une telle fonction n'agit que sur une seule colonne du résultat. Ce n'est donc pas la projection (après l'opérateur `SELECT`) mais la sélection (après l'opération `WHERE`) qu'il faut modifier, en y incluant une condition sur le prix à l'aide d'une sous-requête, entre parenthèses.

```
SELECT nom_fournisseur, prix_kg * 4.5
FROM fournisseurs JOIN prix ON id_fournisseur = id_four
WHERE id_mat = 8713
AND prix_kg = (SELECT MIN(prix_kg) FROM prix WHERE id_mat = 8713);
```

Si l'on conserve la requête de la question 7 et que l'on classe les lignes à l'aide de `ORDER BY prix`, on ne peut alors pas savoir si plusieurs fournisseurs sont aussi compétitifs. Ce n'est donc pas une réponse valable.

Mines Informatique optionnelle MP 2022

Corrigé

Ce corrigé est proposé par Vincent Puyhaubert (professeur en CPGE) ; il a été relu par William Aufort (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

Ce problème comporte trois parties largement indépendantes.

- La première débute par l'étude d'une fonction `mystere` dont le code est donné. Il faut comprendre puis démontrer ce qu'elle fait. La suite introduit la notion de distance d'édition de Levenshtein entre deux mots. Elle se définit comme le nombre minimum d'opérations (insertion, suppression ou modification de symboles) pour transformer un mot en l'autre. Le calcul de cette distance pour deux mots donnés est une problématique classique que l'on résout efficacement par une méthode de programmation dynamique.
- La partie suivante introduit la notion de « trie » (qui se prononce comme le mot anglais *try*, et que l'on appelle « arbre préfixe » en français). Il s'agit d'un cas particulier d'arbre permettant de représenter efficacement des ensembles de mots. Afin de créer puis manipuler cette structure de données, l'énoncé fournit une implémentation concrète de dictionnaire dont il introduit les opérations au fur et à mesure. En fin de partie, l'énoncé introduit la notion de système de transitions, qui coïncide avec un automate non complet, potentiellement infini, dont tous les états sont acceptants. Elle s'achève sur la rédaction d'une fonction supprimant dans un trie les mots qui ne sont pas acceptés par un système de transitions fourni en argument.
- La troisième partie fait le lien entre les deux précédentes. On introduit la notion de script qui permet de décrire une suite d'opérations pour passer d'un mot ***b*** « brouillé » à un mot ***c*** « cible ». À l'aide de cet outil, on construit un système de transitions permettant de filtrer dans un trie les mots qui sont préfixes d'au moins un mot ***c*** à une distance k fixée d'un mot ***b*** également fixé.

Les thématiques abordées sont classiques et très intéressantes. Cependant, le sujet est d'un niveau particulièrement élevé. L'énoncé est plutôt avare en indications. La première partie présente très rapidement une question 6 qui non seulement est bloquante pour toute la suite, mais comporte pour ne rien arranger une coquille pas évidente à repérer. Les codes de la première partie sont assez longs à écrire ; ceux de la partie suivante sont très courts, mais d'une technicité assez élevée. Cette épreuve est donc à déconseiller si vous cherchez à vous exercer sur les méthodes de base du cours. Elle conviendra mieux aux candidats aguerris qui ont envie de se frotter aux difficultés.

INDICATIONS

Partie I

- 1 La première ligne du filtrage permet de deviner le type du retour de la fonction. La seconde ligne suffit ensuite pour conclure sur la nature de l'argument $\mathbf{z}$.
- 2 Prouver la terminaison à l'aide d'une récurrence sur la longueur de la liste.
- 3 Montrer par récurrence sur la longueur de la liste $\mathbf{z}$ que `mystere z` renvoie le plus petit élément de cette liste lorsqu'elle est non vide.
- 4 Il y a de nombreuses interprétations possibles de cette question mal posée. Ne pas s'y attarder.
- 5 Pour transformer un mot en le mot vide, il suffit de supprimer tous ses symboles.
- 6 L'énoncé commet une erreur en n'autorisant pas l'utilisation des quantités $d_{i,j+1}$ et $d_{i+1,j}$ dans la formule de récurrence. Justifier plutôt que pour tous entiers i, j appartenant à $\llbracket 0; m-1 \rrbracket \times \llbracket 0; n-1 \rrbracket$,

$$d_{i+1,j+1} = \begin{cases} d_{i,j} & \text{si } b_{i+1} = c_{j+1} \\ 1 + \min \{d_{i,j}, d_{i+1,j}, d_{i,j+1}\} & \text{sinon} \end{cases}$$

- 7 Commencer par créer un tableau de la longueur de la liste contenant un symbole arbitraire, puis utiliser une fonction auxiliaire récursive pour remplir ce tableau avec les symboles de la liste.
- 8 Créer une matrice d'entiers de taille $(m+1) \times (n+1)$ pour calculer les coefficients $(d_{i,j})_{i,j \in \llbracket 0; m \rrbracket \times \llbracket 0; n \rrbracket}$. Remplir ensuite ce tableau ligne par ligne ou colonne par colonne en utilisant les formules des questions 5 et 6.
- 9 Par la méthode précédente, chaque coefficient $d_{i,j}$ se calcule en temps constant à l'aide de ceux précédemment calculés.
- 10 Commencer par introduire une séquence optimale de modifications qui transforme $\mathbf{b}$ en $\mathbf{c}$. Définir les symboles $x_0, \dots, x_{r-1}$ comme étant les symboles du mot initial $\mathbf{b}$ qui ne subissent aucune modification. Justifier ensuite qu'entre deux de ces symboles consécutifs, il ne peut y avoir soit que des insertions, soit que des suppressions de symboles.

Partie II

- 11 Lister tous les mots obtenus en partant de la racine jusqu'à un symbole $\$$.
- 12 C'est une question de cours. On attend au minimum la notion d'arbre binaire de recherche, puis soit un tableau, soit une table de hachage, cette dernière n'étant pas exigible officiellement.
- 13 Le trie vide est un nœud contenant un dictionnaire vide. Le trie contenant le mot vide est un nœud contenant un dictionnaire ayant pour seule clé le symbole $\$$ associé à un dictionnaire vide.
- 14 Construire un nœud contenant un dictionnaire à une seule clé x , à laquelle on associe le dictionnaire représentant le mot vide.
- 15 Tester la présence du premier symbole du mot dans le dictionnaire associé au trie puis, si tel est le cas, lancer un appel récursif avec la queue de la liste représentant le mot et le trie associé à ce symbole.

- 16 L'implémentation est similaire à celle de la question précédente si ce n'est que lorsqu'un symbole du mot n'apparaît plus dans le dictionnaire, il faut créer une nouvelle branche à l'aide de `CharMap.add` et d'un appel récursif sur la queue et un trie vide.
- 17 Justifier qu'il n'y a qu'un seul exemplaire de `trie_vide` qui existe en fin de construction.
- 18 La fonction `filtre` a pour but d'appliquer récursivement la fonction `trie_trim` de manière à élaguer le trie `y`. Si ce faisant l'arbre devient vide, renvoyer une option `None`.
- 19 Commencer par majorer le nombre de mots à distance au plus 1 de `b`. On exprimera le résultat en fonction de la longueur n du mot `b`. En déduire un majorant du nombre de mots à distance k à l'aide d'une récurrence sur k .
- 20 Construire le système de transitions demandé à partir d'un automate déterministe complet qui reconnaît les mots contenant le facteur `ccmp`.
- 21 Utiliser une nouvelle fois `CharMap.filter_map` de manière à appliquer récursivement la fonction `trie_filter` en mettant à jour l'état `qchapeau` à chaque appel récursif.
- 22 Remarquer que la fonction `delta` est appliquée une fois pour chaque nœud du trie.

Partie III

- 23 Pour le sens indirect, remarquer qu'un k -script de `c'` vers `b'` permet d'obtenir une suite d'au plus k opérations qui transforme `c` en `b`. Pour la réciproque, introduire une nouvelle séquence optimale de modifications qui transforme `c` en `b`. Utiliser à nouveau les arguments de la question 6. Démontrer de plus que, sans perdre de généralité, on peut supposer que si on insère un symbole entre deux symboles consécutives de `c`, alors le second symbole n'est ni substitué, ni supprimé dans la séquence de modifications.
- 24 Distinguer les trois cas pour la valeur de τ_{j+1} . Pour chacun, déterminer le coût du script complété et les mots en lesquels ce script peut transformer `c'`.
- 25 Appliquer la même méthode qu'à la question précédente.
- 26 Utiliser un système de transitions dont l'ensemble des états est

$$Q = \{(\text{pref}_i(\mathbf{b\$}), \kappa) \mid i \leq m + 1, \quad \kappa \leq k\}$$

Définir ensuite trois familles de transitions en vous aidant des résultats des questions 24 et 25. Pour l'implémentation, il est préférable d'utiliser les suffixes de `b$` plutôt que ses préfixes.

- 27 L'adaptation porte essentiellement sur le type et le rôle de la variable `qchapeau`, ainsi que le calcul de `delta qchapeau`. On peut également maintenant se passer du type `option`.

I. UNE MESURE DES ERREURS DE SAISIE

1 À la lecture du code de la fonction `mystere`, on constate que

- l'argument `z` de la fonction est une liste, compte tenu du filtrage mis en place ;
- la fonction renvoie un entier dans le cas où `z` est une liste vide, et par conséquent dans tous les cas de figure ;
- lorsque la liste ne contient qu'un seul élément, c'est ce dernier qui est renvoyé, ce qui implique qu'il est de type `int`, et donc que `z` est du type `int list`.

Il s'ensuit que

La fonction `mystere` a pour signature `int list -> int`.

2 Pour tout entier n , on note $\mathcal{P}(n)$ la propriété suivante :

$\mathcal{P}(n)$: « La fonction `mystere` termine pour toute liste de longueur n avec n appels à la fonction `mystere` si n est non nul, et un seul sinon. »

Montrons que $\mathcal{P}(n)$ est vraie pour tout entier n .

On notera qu'ici, l'appel initial à la fonction `mystere` est pris en compte dans le comptage.

- $\mathcal{P}(0)$ et $\mathcal{P}(1)$ sont vraies car la fonction termine immédiatement sans appel récursif dans ces deux cas.
- $\mathcal{P}(n) \implies \mathcal{P}(n+1)$: Soit $n \geq 1$. Supposons $\mathcal{P}(n)$ vérifiée et considérons une liste `z` de longueur $n+1 \geq 2$. L'instruction `mystere z` effectue alors un appel récursif sur une liste de longueur n puisque, selon les notations du filtrage, `y` est de longueur $n-1$. Par hypothèse de récurrence, celui-ci termine en n appels. Par suite, `mystere z` termine en $n+1$ appels (en incluant l'appel initial).
- **Conclusion :** $\forall n \in \mathbb{N} \quad \mathcal{P}(n) \text{ est vraie}$

Ainsi,

Pour toute liste non vide `z` en entrée, `mystere z` termine en $|z|$ appels à `mystere`, où $|z|$ désigne la longueur de la liste. Elle termine en un seul appel pour une liste vide.

3 On peut compléter le commentaire de la ligne 2 de la manière suivante :

« La fonction `mystere` calcule le plus petit élément de la liste `z`. »

Cette affirmation n'est toutefois pas valable lorsque `z` est une liste vide. On peut donc proposer la propriété plus précise suivante :

« La fonction `mystere` calcule le plus petit élément de la liste `z` si celle-ci est non vide, et renvoie `max_int` sinon. »

Démonstrons cette propriété une nouvelle fois par récurrence sur la longueur de la liste. Pour tout entier $n \in \mathbb{N}^*$, notons

$\mathcal{P}(n)$: « La fonction `mystere` renvoie le plus petit élément de `z` pour toute liste de longueur n . »

et montrons que $\mathcal{P}(n)$ est vraie pour tout entier $n \geq 1$.

X/ENS Maths A MP 2022 — Corrigé

Ce corrigé est proposé par Florian Metzger (professeur en CPGE) ; il a été relu par Jean Starynkévitch (professeur en CPGE) et par Benoit Chevalier (ENS Ulm).

Ce sujet traite principalement des applications p -linéaires alternées définies sur un espace vectoriel de dimension finie d , sans que l'on suppose $d = p$, ce qui généralise en quelque sorte les notions de déterminant et de volume.

- Dans la partie I, on construit des familles orthonormées intéressantes pour des sous-espaces vectoriels, qui permettent de les caractériser en utilisant les déterminants des matrices de Gram définies par

$$\text{Gram}(u, v)_{i,j} = \langle u_i, v_j \rangle$$

- Dans la partie II, on s'intéresse à $\mathcal{A}_p(E, \mathbb{R})$ l'ensemble des applications p -linéaires alternées sur un espace vectoriel de dimension d . On y étudie plusieurs propriétés et exemples donnés par les matrices de Gram avant d'y définir la notion de p -volume d'une famille de vecteurs.
- La partie III construit, à l'aide d'une base orthonormée de E , un produit scalaire sur $\mathcal{A}_p(E, \mathbb{R})$, qui se révèle indépendant de la base orthonormée de E initialement choisie.
- Enfin, la partie IV étudie la p -ième grassmannienne de E , c'est-à-dire de l'ensemble des sous-espaces vectoriels de dimension p , orientés par le choix d'une base. On y introduit une injection de cette grassmannienne vers la sphère unité de l'espace euclidien $\mathcal{A}_p(\mathbb{R})$ afin de munir la grassmannienne d'une topologie naturelle, dont on finit par étudier la compacité ainsi que la connexité par arcs.

Ce sujet de difficulté inégale mais dans l'ensemble abordable fait principalement appel au programme de mathématiques de première année. Hormis un soupçon de topologie au tout début et à la fin, seule l'algèbre linéaire et les espaces euclidiens sont utilisés. Bien qu'intéressant par les résultats établis, il ne constitue pas vraiment un bon sujet pour réviser le programme en fin d'année.

INDICATIONS

Partie I

- 1.a La sphère unité de V et celle de V' sont compactes et $\langle \cdot, \cdot \rangle$ est bilinéaire en dimension finie.
- 2 Raisonner par récurrence sur k . Pourquoi suffit-il de montrer que $\langle u_k, u'_k \rangle \geq 1$? Se rappeler la définition de u et u' .
- 3.b La borne supérieure définissant u et u' étant un maximum, on pourra utiliser la condition nécessaire d'extrema d'une fonction réelle.
- 3.c Que vaut $(F + G)^\perp$ dans un espace euclidien ?
- 4.a Penser à l'inégalité de Cauchy-Schwarz et montrer que $\langle u_k, u'_k \rangle > 0$ pour tout entier k .
- 4.c Quid du cas d'égalité dans l'inégalité de Cauchy-Schwarz ?

Partie II

- 5.a Utiliser la formule sommatoire du déterminant.
- 6.a Se rappeler du résultat de la question 5.b.
- 7.a Utiliser la p -linéarité pour développer l'expression sous forme d'une somme à p indices.
- 7.b Raisonner par double implication et utiliser la question 7.a.
- 7.c Montrer que $\text{Gram}(e, e)$ est symétrique et que ses valeurs propres sont positives.
- 8.b Écrire $e_1 = \text{pr}(e_1) + h$ avec $h \in \text{Vect}(e_2, \dots, e_p)$ puis utiliser un calcul par blocs.
- 8.c Que dire si un vecteur et son projeté orthogonal ont la même norme ?
- 9.a Utiliser la question 8.a.

Partie III

- 10.a Décomposer des vecteurs $u_1, \dots, u_p$ dans une base orthonormée de E et utiliser la p -linéarité pour écrire $\omega(u_1, \dots, u_p)$ sous forme d'une somme à multi-indice. Le caractère alterné montre que la somme peut ne porter que sur un ensemble d'indices deux à deux distincts. Partitionner ces indices en utilisant $\mathcal{I}_p$ et les réécrire en utilisant des permutations.
- 10.b Que dire de matrices de Gram pour des familles orthonormées ? Pour la dimension, on pourra mettre en bijection $\mathcal{I}_p$ est un ensemble de combinaisons.
- 10.c Utiliser le déterminant.
- 11.a Penser à la question 10.a et à la symétrie de Ω_p .
- 12 Que dire d'une matrice de passage d'une base orthonormée à une autre ? Se rappeler le résultat de la question 10.a.
- 13.a Penser aux questions 4.c et 7.a et calculer de deux façons $\Omega_p(u)(u')$ pour les familles u, u' de la question 1.b.
- 13.b Montrer que, nécessairement, l'application Ψ est constante sur l'ensemble des bases orthonormées directes de (V, C) .
- 14.a Montrer que $\Psi(V, C) = \Omega_p(b)$ pour toute base orthonormée $b \in C$, c'est-à-dire directe pour l'orientation C .

- 14.b On est en dimension finie et il ne reste donc plus qu'à montrer que $\Psi(\widetilde{\text{Gr}}(p, E))$ est fermée. Utiliser $\Psi(V, C) = \Omega_p(b)$ pour tout $b \in C$ base orthonormée.
- 15 Le cas $p = d$ conduit à se poser la question de la connexité d'une paire dans un espace vectoriel normé.

Pour le cas $p \leq d-1$, utiliser les familles u, u' de la question 1.b et le fait que, si u est directe pour l'orientation C choisie sur V , alors on obtient $\Psi(V, C) = \Omega_p(u)$ et $\Psi(V, C) = -\Omega_p(u)$ sinon. Dans le deuxième cas, on pourra se demander comment relier un vecteur $u \in V$ à $-u$ en utilisant $V^\perp$.

Dans tout le corrigé, on notera $\mathbb{S}_1(V)$ la sphère unité d'un sous espace vectoriel V de E (pour la norme euclidienne).

Le caractère nommé alterné par l'énoncé est traditionnellement appelée antisymétrie : l'échange de deux vecteurs en argument de l'application p -linéaire change le signe de sa valeur. Cela équivaut au critère pour toute permutation donné par l'énoncé. En effet, l'ensemble S des permutations σ satisfaisant

$$\forall (x_1, \dots, x_p) \quad f(x_{\sigma(1)}, \dots, x_{\sigma(p)}) = \varepsilon(\sigma) f(x_1, \dots, x_p)$$

contient l'ensemble des transpositions pour la définition de l'antisymétrie donnée ci-dessus. Mais on peut vérifier facilement que cet ensemble est un sous-groupe de $\mathfrak{S}_p$, et ce dernier est engendré par les transpositions, donc l'ensemble S est bien $\mathfrak{S}_p$ tout entier (ce qui est la définition de l'énoncé).

Usuellement, on définit plutôt le caractère alterné par le fait que si une famille comporte deux vecteurs égaux, alors la valeur de l'application est nulle en cette famille. Ces deux notions sont équivalentes. Il est en effet facile d'obtenir le caractère alterné à partir de l'antisymétrie : l'échange de deux vecteurs égaux dans la famille ne la modifie pas, mais change le signe de la valeur calculée. On a alors

$$f(u_1, \dots, u_i, \dots, u_i, \dots, u_p) = -f(u_1, \dots, u_i, \dots, u_i, \dots, u_p)$$

et donc cette valeur est nulle puisque $\mathbb{R}$ est de caractéristique différente de 2. Réciproquement, en supposant le caractère alterné, on développe par linéarité pour les i -ième et j -ième variables :

$$\begin{aligned} 0 &= f(u_1, \dots, u + v, \dots, u + v, \dots, u_p) \\ &= f(u_1, \dots, u, \dots, u, \dots, u_p) + f(u_1, \dots, v, \dots, u, \dots, u_p) \\ &\quad + f(u_1, \dots, u, \dots, v, \dots, u_p) + f(u_1, \dots, v, \dots, v, \dots, u_p) \\ &= 0 + f(u_1, \dots, u, \dots, v, \dots, u_p) + f(u_1, \dots, v, \dots, u, \dots, u_p) + 0 \end{aligned}$$

ce qui donne le résultat.

PARTIE I

1.a L'application $\langle \cdot, \cdot \rangle : V \times V' \rightarrow \mathbb{R}$ est bilinéaire entre espaces de dimensions finies donc continue. Les parties $\mathbb{S}_1(V)$ et $\mathbb{S}_1(V')$ sont fermées et bornées dans un espace de dimension finie, donc sont compactes. Notons $K = \mathbb{S}_1(V) \times \mathbb{S}_1(V')$: c'est un compact par produit fini de compacts. D'après le théorème des bornes atteintes, $\langle \cdot, \cdot \rangle$ est bornée et atteint ses bornes sur K , ce qui établit bien que cette borne supérieure est en fait un maximum :

$$\exists (u_1, u'_1) \in V \times V' \quad \|u_1\| = \|u'_1\| = 1 \quad \text{et} \quad \sup_{(a, a') \in \mathbb{S}_1(V) \times \mathbb{S}_1(V')} \langle a, a' \rangle = \langle u_1, u'_1 \rangle$$

1.b Montrons par récurrence finie que la propriété

$\mathcal{P}(k)$: il existe deux familles orthonormées $u = (u_1, \dots, u_k) \in V^k$
et $u' = (u'_1, \dots, u'_k) \in (V')^k$ vérifiant

$$\langle u_k, u'_k \rangle = \sup \left\{ \langle a, a' \rangle \left| \begin{array}{l} a \in \mathbb{S}_1(V \cap \{u_1, \dots, u_{k-1}\}^\perp) \\ a' \in \mathbb{S}_1(V' \cap \{u'_1, \dots, u'_{k-1}\}^\perp) \end{array} \right. \right\}$$

est vraie pour tout $k \in \llbracket 1; p \rrbracket$.

X Maths B MP 2022 — Corrigé

Ce corrigé est proposé par Jean Starynkévitch (professeur en CPGE); il a été relu par Sélim Cornet (professeur en CPGE) et Vincent Puyhaubert (professeur en CPGE).

Le problème traite de la convergence uniforme de suites de probabilités sur un ensemble dénombrable, et en présente quelques applications.

- La première partie a pour premier objectif d'obtenir le *développement eulérien de la cotangente* :

$$\pi \cotan(\pi x) = \sum_{n=-\infty}^{+\infty} \frac{1}{x+n} \stackrel{\text{déf}}{=} \lim_{N \rightarrow +\infty} \sum_{n=-N}^N \frac{1}{x+n}$$

puis, en introduisant les nombres de Bernoulli, d'en déduire les valeurs de

$$\zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s}$$

pour les petites valeurs entières paires de s .

- La deuxième partie introduit la notion de suite de probabilités *tendue*, qui est, sans que l'énoncé ne le dise explicitement, un critère de compacité pour la norme de la convergence uniforme sur les ensembles de probabilités sur un ensemble dénombrable E : ici, les probabilités sur E sont des cas particuliers de fonctions bornées de $\mathcal{P}(E)$ vers $\mathbb{R}$.
- La troisième partie, indépendante de la précédente, vise à prouver que si X est une variable aléatoire et $\psi_n : \mathbb{N}^* \rightarrow \mathbb{N}^*$ l'application qui à un entier x associe la partie de la décomposition en facteurs premiers de x qui ne comporte que les facteurs de x inférieurs au n -ième nombre premier, alors

$$P(\psi_n(X) = x) \xrightarrow{n \rightarrow +\infty} P(X = x)$$

- La quatrième partie a pour objectif de montrer que, si s nombres sont tirés au hasard dans $\llbracket 1; n \rrbracket$, la probabilité $P_n(s)$ que ces nombres soient premiers entre eux dans leur ensemble vérifie

$$P_n(s) \xrightarrow{n \rightarrow +\infty} \frac{1}{\zeta(s)}$$

Si la première partie est classique pour des révisions d'analyse, le reste du sujet est d'une difficulté adaptée au concours, à l'exception de quelques questions de cours faciles qui figurent au début de certaines parties. Pour le travailler avec profit, il est souhaitable d'être bien au point en analyse élémentaire et en probabilités.

INDICATIONS

- 1a Prouver la convergence absolue.
- 1c, 1d Introduire les sommes partielles de la série définissant g .
- 2a Réduire au même dénominateur le membre de gauche, puis utiliser la formule $\cos a \sin b + \sin a \cos b = \sin(a + b)$ pour simplifier le numérateur.
- 2b Introduire les sommes partielles de la série définissant g .
- 3a Prouver que $f(x) - \frac{1}{x} \xrightarrow{x \rightarrow 0} 0$ et $g(x) - \frac{1}{x} \xrightarrow{x \rightarrow 0} 0$.
- 3b Utiliser l'équation fonctionnelle explicitée en question 2.
- 4 Prouver que $M \leq 0$, puis que $-M = \sup_{t \in [0; 1]} (-\tilde{D}(t)) \leq 0$.
- 5a Appliquer la question 4 au réel non entier $x/2\pi$, puis écrire le terme

$$\frac{(x/2\pi)^2}{(x/2\pi)^2 - n^2}$$

comme somme d'une série.

- 5b Se ramener à la question 5a en commençant le calcul du membre de gauche par l'égalité $e^{ix} - 1 = 2 \sin(x/2) e^{ix/2}$.
- 6 Constater que l'égalité est vérifiée lorsque z est imaginaire pur, et se servir de l'unicité du développement en série entière au voisinage de 0 d'une fonction fixée.
- 7b Utiliser encore une fois l'unicité du développement en série entière, ainsi que le produit de Cauchy.
- 11 Si $m \neq n$, que vaut $\|\delta_m - \delta_n\|$?
- 12a Construire de proche en proche $\varphi_1, \varphi_2, \dots, \varphi_k$ par récurrence forte.
- 12d Commencer par prouver que pour tout m et tout k , on a

$$\sum_{i=1}^m \mu_{\psi(k)}(x_i) \leq 1$$

- 14 Remarquer que $\mu_X(A) = \mathbb{E}(\mathbb{1}_{\{X \in A\}})$ et que pour tout $\omega \in \Omega$,
- $$\mathbb{1}_{\{X \in A\}}(\omega) \neq \mathbb{1}_{\{Y \in A\}}(\omega) \implies X(\omega) \neq Y(\omega)$$
- 15b Montrer l'inclusion d'événements $\{X_n \neq X\} \subset \{L \geq n\}$.
- 15c Utiliser la propriété de continuité décroissante de P avec la suite d'événements

$$(\{L \geq n\})_{n \in \mathbb{N}}$$

- 16 C'est une application des questions précédentes.
- 18 C'est l'une des questions les plus difficiles. Utiliser la question 9, et revenir aux ε en utilisant encore que la suite est tendue.
- 19 On trouve $P(r \mid X_n^{(i)}) = (1/n) \lfloor n/r \rfloor$.
- 20a On trouve $P(k \mid Z) = 1/k^s$.
- 20b Appliquer la question 18. Pour prouver que la suite est tendue, chercher F_ε de la forme $\llbracket 1; r \rrbracket$ et utiliser

$$\{Z_n^{(s)} = k\} \subset \{k \mid Z_n^{(s)}\} = \bigcap_{i=1}^s \{k \mid X_n^{(i)}\}$$

- 21 La question 20 est une modélisation judicieuse de cette dernière question.

I. PREMIÈRE PARTIE

1a Soit $x \in \mathbb{R} \setminus \mathbb{Z}$. Alors, lorsque n tend vers $+\infty$, on a

$$\frac{1}{x+n} + \frac{1}{x-n} = \frac{-2x}{n^2 - x^2} = O\left(\frac{1}{n^2}\right)$$

et $\sum 1/n^2$ est une série convergente à termes positifs. Il résulte du théorème de comparaison pour les séries à termes positifs que la série de terme général $1/(x+n) + 1/(x-n)$ est absolument convergente, donc convergente.

Pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$, la série $\sum_{n \geq 1} \left(\frac{1}{x+n} + \frac{1}{x-n} \right)$ converge.

Rappelons que l'étude de la convergence d'une série se fait la plupart du temps en étudiant uniquement son terme général; bien plus rarement en étudiant la suite des sommes partielles.

1b Commençons par remarquer que $\mathbb{R} \setminus \mathbb{Z}$ est symétrique par rapport à 0 (pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$, on a $-x \in \mathbb{R} \setminus \mathbb{Z}$) ce qui donne un sens à l'étude de la parité des trois fonctions f , g et D , définies sur $\mathbb{R} \setminus \mathbb{Z}$.

Montrons que f est impaire. Soit $x \in \mathbb{R} \setminus \mathbb{Z}$. Du fait que la fonction sinus est impaire et la fonction cosinus paire, on a

$$f(-x) = \frac{\pi \cos(\pi(-x))}{\sin(\pi(-x))} = \frac{\pi \cos(-\pi x)}{\sin(-\pi x)} = -\frac{\pi \cos(\pi x)}{\sin(\pi x)} = -f(x)$$

De fait, f est impaire.

Montrons que g est impaire. Pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$, on a

$$g(-x) = \frac{1}{-x} + \sum_{n=1}^{+\infty} \left(\frac{1}{-x+n} + \frac{1}{-x-n} \right) = -\frac{1}{x} + \sum_{n=1}^{+\infty} - \left(\frac{1}{x-n} + \frac{1}{x+n} \right) = g(-x)$$

La fonction g est impaire.

Puisque $D = f - g$ est une différence de deux fonctions impaires,

La fonction D est impaire.

1c Commençons par remarquer que pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$, on a $x+1 \in \mathbb{R} \setminus \mathbb{Z}$, ce qui permet d'étudier la 1-périodicité des fonctions f , g et D .

Montrons que f est 1-périodique. Rappelons que l'on a pour tout réel t :

$$\cos(t + \pi) = -\cos t \quad \text{et} \quad \sin(t + \pi) = -\sin t$$

Par conséquent, pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$,

$$f(x+1) = \frac{\pi \cos(\pi(x+1))}{\sin(\pi(x+1))} = \frac{\pi \cos(\pi x + \pi)}{\sin(\pi x + \pi)} = \frac{-\pi \cos(\pi x)}{-\sin(\pi x)} = f(x)$$

Ainsi, f est 1-périodique.

Montrons que g est 1-périodique. Pour $N \in \mathbb{N}$, définissons $g_N : \mathbb{R} \setminus \mathbb{Z} \rightarrow \mathbb{R}$ par

$$g_N(x) = \frac{1}{x} + \sum_{n=1}^N \left(\frac{1}{x+n} + \frac{1}{x-n} \right) = \sum_{n=-N}^N \frac{1}{x+n}$$

de sorte que g est, par définition, la limite simple de la suite de fonctions $(g_N)_{N \in \mathbb{N}}$. Pour tout $x \in \mathbb{R} \setminus \mathbb{Z}$ et tout $N \in \mathbb{N}$, on a

$$g_N(x+1) = \sum_{n=-N}^N \frac{1}{x+n+1} \underset{m \leftarrow n+1}{=} \sum_{m=-N+1}^{N+1} \frac{1}{x+m} = \frac{-1}{x-N} + g_N(x) + \frac{1}{x+N+1}$$

donc, en passant à la limite (à x fixé) lorsque N tend vers $+\infty$, il vient :

$$g(x+1) = g(x)$$

La fonction g est 1-périodique.

Puisque $D = f - g$ est une différence de deux fonctions 1-périodiques,

La fonction D est 1-périodique.

1d Montrons que g est continue sur $\mathbb{R} \setminus \mathbb{Z}$. Puisque g est 1-périodique, il suffit de montrer que g est continue sur $]0; 1[$.

- Pour tout N , g_N est continue sur $]0; 1[$ (en tant que fonction rationnelle à pôles entiers).
- Montrons que la série de fonctions $\sum (g_n - g_{n-1})$, dont la somme définit g , converge normalement sur $]0; 1[$. Pour $x \in]0; 1[$ et $n \geq 2$, on a

$$|g_n(x) - g_{n-1}(x)| = \left| \frac{1}{x+n} + \frac{1}{x-n} \right| = \frac{2x}{n^2 - x} \leq \frac{2}{n^2 - 1}$$

et le majorant ainsi obtenu est indépendant de x , et terme général d'une série convergente. Il en découle que la série de fonctions $\sum (g_n - g_{n-1})$ converge normalement, donc uniformément. Par télescopage, la suite $(g_N)_{N \in \mathbb{N}}$ converge uniformément sur $]0; 1[$.

Par conséquent, en tant que limite uniforme de fonctions continues sur $]0; 1[$, la fonction g est continue sur $]0; 1[$. Puisque g est aussi périodique de période 1,

La fonction g est continue sur $\mathbb{R} \setminus \mathbb{Z}$.

Il devrait vous paraître surprenant d'obtenir une convergence uniforme sur l'intervalle ouvert $]0; 1[$, alors qu'il y a des singularités en 0 et en 1 (g y admet des limites infinies) : si $(f_n)_{n \in \mathbb{N}}$ est une suite de fonctions qui converge uniformément sur $]0; 1[$ et telle que les suites $(f_n(0))_{n \in \mathbb{N}}$ et $(f_n(1))_{n \in \mathbb{N}}$ convergent, alors il y a convergence uniforme sur $[0; 1]$.

En règle générale, lorsqu'il est pertinent de prouver la continuité d'une fonction (définie comme la somme d'une série de fonctions) sur un intervalle non fermé I (ou non borné), on établit la convergence uniforme sur tout segment de I , et il n'y a pas convergence uniforme sur l'intervalle I .

L'argument qui fait de l'exemple de la fonction g une exception à la règle générale précitée est le fait que les singularités en 0 et en 1 sont dues chacune à un seul terme : $1/x$ en 0 et $1/(x-1)$ en 1. En retirant ces termes, on aura bien convergence uniforme sur l'intervalle fermé $[0; 1]$.

Par ailleurs f est continue sur $\mathbb{R} \setminus \mathbb{Z}$ par opérations usuelles sur les fonctions continues. Il s'ensuit que

D est continue sur $\mathbb{R} \setminus \mathbb{Z}$.

X/ENS Informatique B MP-PC-PSI 2022 — Corrigé

Ce corrigé est proposé par William Aufort (professeur en CPGE) ; il a été relu par Benjamin Monmege (enseignant-chercheur à l'université) et Vincent Puyhaubert (professeur en CPGE).

Ce sujet s'intéresse au remplissage d'une grotte lors d'une inondation. Plus précisément, étant donné une grotte et une source d'eau de débit constant, on souhaite déterminer la hauteur d'eau en tout point de la grotte à un instant donné. Le sujet est de difficulté globalement croissante et doit être abordé dans l'ordre, chaque partie reposant sur les notions précédemment introduites.

- Dans la première partie, on modélise une grotte par une liste de déplacements verticaux et horizontaux décrivant le profil de la grotte. Il s'agit principalement d'écrire des fonctions permettant de vérifier qu'une suite de déplacements correspond à des profils de grottes valides. Les questions de cette partie sont plutôt abordables.
- La deuxième partie s'intéresse au cas particulier du remplissage d'une vallée, qui est un type particulier de grotte dans lequel la source est située au fond de la vallée. Il apparaît notamment qu'en décomposant la vallée en rectangles de moins en moins profonds, on peut écrire un programme qui évalue la hauteur d'eau à tout instant. Cette partie comporte trois questions permettant d'y parvenir ; l'une d'elles est particulièrement délicate.
- Enfin, la partie III traite le cas plus général d'une grotte dite à ciel ouvert. On commence comme précédemment par décomposer la grotte en rectangles, mais cette fois ceux-ci sont liés par une structure arborescente longuement détaillée (trois pages d'énoncé!). Une fois cette structure arborescente déterminée, on peut en déduire l'ordre de remplissage des différents rectangles puis la hauteur d'eau dans chacun d'eux. Deux sous-parties s'intéressent respectivement au cas d'une source positionnée à l'entrée gauche de la grotte, et au cas - beaucoup plus difficile - d'une source arbitrairement positionnée.

Comme très souvent au concours X/ENS, nous avons affaire ici à un sujet exclusivement algorithmique, très bien écrit et détaillé. Il propose l'étude d'un problème intéressant (et même amusant), donnant lieu à l'implémentation d'algorithmes élaborés et à l'étude de leurs complexités. Certaines questions peuvent être traitées de façon simple et élégante en utilisant la récursivité ; il vaut donc mieux aborder ce sujet en deuxième année. Même s'il ne comporte que 15 questions, aucun des codes demandés ne contient moins de 5 lignes, et certains sont particulièrement longs et/ou difficiles. Il s'agit donc d'un bon sujet pour s'entraîner à écrire rapidement du code correct.

INDICATIONS

Partie I

- 1 Pour repérer les rebroussements, utiliser une boucle où chaque tour examine deux pas successifs dans la liste.
- 2 Dans une vallée, on doit d'abord trouver tous les déplacements B, et ensuite tous les déplacements H. On peut distinguer les deux phases en mettant à jour une variable lorsque l'on rencontre un déplacement H.
- 4 Appliquer la fonction `voisin` sur le dernier point ajouté dans la liste à renvoyer et le pas actuellement examiné.
- 5 Commencer par écrire une fonction testant si un élément est présent dans une liste donnée. L'utiliser ensuite pour repérer les points en double.

Partie II

- 6 Remarquer qu'il s'agit d'implémenter un algorithme de recherche de maximum dans une liste. Attention à bien renvoyer le point d'abscisse minimale parmi les plus profonds de la vallée.
- 7 Utiliser trois variables contenant la profondeur et les abscisses extrémales du plateau, et les mettre à jour en fonction du déplacement traité.
- 8 Question difficile de programmation. Commencer par repérer l'indice du fond de la vallée, puis parcourir les portions montante et descendante selon les profondeurs décroissantes en adaptant l'algorithme de fusion de deux listes triées. Mettre à jour les caractéristiques des rectangles à construire en fonction des profondeurs des deux plateaux comparés.
- 9 Parcourir les rectangles décomposant le volume intérieur de la vallée et les remplir entièrement du plus profond au moins profond tant que c'est possible. Attention à bien gérer le dernier rectangle partiellement rempli.

Partie III

- 10 Cette question est en apparence difficile, mais il s'agit uniquement de suivre à la lettre l'algorithme donné dans l'énoncé et de le traduire en Python. Introduire une variable supplémentaire correspondant au numéro du rectangle actuellement ouvert.
- 11 Remarquer que le remplissage d'un père ne peut se faire qu'après le remplissage de chacun de ses enfants, lesquels se remplissent de gauche à droite. En déduire une implémentation utilisant une fonction auxiliaire récursive `remplir(rect)` ajoutant, dans une liste définie au préalable, les numéros des rectangles.
- 12 Adapter le programme de la question 9 à l'ordre renvoyé par la fonction précédente.
- 14 Reprendre le principe de l'exploration récursive de la hiérarchie utilisé à la question 11.
- 15 On pourra commencer par traiter quelques cas particuliers, notamment les cas où l'eau ne remonte pas au dessus de la source, et celui où tous les rectangles ont une hauteur non nulle. En dehors de ces cas, l'un des rectangles ancêtres de la source sera partiellement rempli : détailler alors l'ordre de remplissage de ses enfants.

I. VALIDITÉ D'UN PROFIL

1 Suivons la définition d'un profil sans rebroussement donnée par l'énoncé. On commence par vérifier que le profil de la grotte ne commence pas par H et ne se termine pas par B. On utilise ensuite une boucle permettant de vérifier que deux pas consécutifs ne correspondent jamais à des déplacements contraires.

```
def est_sans_rebroussement(g):
    if g[0] == H or g[-1] == B:
        return False
    for i in range(len(g)-1):
        if (g[i],g[i+1]) == (G,D) or (g[i],g[i+1]) == (D,G):
            return False
        if (g[i],g[i+1]) == (H,B) or (g[i],g[i+1]) == (B,H):
            return False
    return True
```

On a choisi dans le code précédent d'écrire les quatre conditions à examiner à l'aide de deux tests pour plus de lisibilité. On aurait pu compacter ces deux tests en un seul test d'appartenance à la liste des couples de positions contraires, en utilisant l'opérateur `in` de Python. La boucle obtenue aurait alors été la suivante

```
for i in range(len(g)-1):
    if (g[i], g[i+1]) in [(G,D), (D,G), (H,B), (B,H)]:
        return False
```

Il est cependant difficile de savoir à la lecture du sujet si la syntaxe « `if x in L` » était autorisée mais il est également possible de reprogrammer rapidement une fonction réalisant ce test et de l'utiliser dans la suite du problème. C'est ce que nous ferons à la question 5, où il sera plus difficile de n'utiliser que les manipulations élémentaires de listes autorisées.

Remarquons enfin que le premier test effectué nécessite que la liste `g` soit non vide. Il est toutefois inutile de gérer le cas de la liste vide à part, car l'énoncé précise que les profils considérés contiennent toujours au moins un pas vers la droite, ils sont donc en particulier non vides.

2 La fonction commence par contrôler que le profil considéré est sans rebroussement à l'aide de la fonction précédente. Il reste à vérifier que le profil ne contient pas de déplacement gauche, et que tous les déplacements vers le haut sont situés après tous les déplacements vers le bas dans la liste. Pour vérifier cette seconde contrainte, on utilise une variable `direction` qui contiendra H dès que l'on aura rencontré un premier H : toutes les directions verticales suivantes devront alors être des H également.

```
def est_une_vallee(g):
    if not est_sans_rebroussement(g):
        return False
    direction = B
    for pas in g:
        if pas == G or (direction == H and pas == B):
            return False
        if pas == H:
            direction = H
    return True
```

Le profil d'une vallée n'est pas obligé de contenir au moins un déplacement vers le haut. Ceci peut paraître étonnant au premier abord (comment une telle grotte pourrait se remplir?), mais l'énoncé précise que le profil de la grotte est implicitement prolongée par des murs verticaux infinis à gauche et à droite.

- 3** Une disjonction de cas permet de trouver les coordonnées du voisin à renvoyer.

```
def voisin(x, y, d):
    if d == G:
        return (x-1, y)
    if d == D:
        return (x+1, y)
    if d == B:
        return (x, y+1)
    else:
        return (x, y-1)
```

- 4** On initialise la liste des coordonnées à renvoyer avec l'origine du profil, puis pour chaque déplacement de la liste `g`, on ajoute les coordonnées du point obtenu en utilisant la fonction `voisin` appliquée au dernier point enregistré.

```
def liste_des_points(g):
    coords = [(0, 0)]
    for pas in g:
        x, y = coords[len(coords)-1]
        coords.append(voisin(x, y, pas))
    return coords
```

- 5** Un profil décrit par une liste `g` est simple si et seulement si la liste générée par l'appel `liste_des_points(g)` ne contient aucun doublon. Pour vérifier cette condition, on peut reprendre le code de la fonction `liste_des_points` en vérifiant avant chaque insertion d'un point que celui-ci n'est pas déjà présent dans la liste. Pour cela, on programme au préalable une fonction `appartient` réalisant un tel test d'appartenance.

```
def appartient(elt, L):
    for x in L:
        if elt == x:
            return True
    return False
```

Dans le doute, il ne faut pas hésiter à définir rapidement une fonction plutôt qu'utiliser le raccourci « `if .. in ..` » qui n'est pas explicitement autorisé.

```
def est_simple(g):
    coords = [(0, 0)]
    for pas in g:
        x, y = coords[-1]
        suivant = voisin(x, y, pas)
        if appartient(suivant, coords):
            return False
        else:
            coords.append(suivant)
    return True
```