Annales des Concours

MP
Mathématiques · Informatique
2019

Sous la coordination de

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

Florian Metzger
docteur en mathématiques
ancien élève de l'École Normale Supérieure (Paris-Saclay)

Benjamin Monmege
enseignant-chercheur à l'université
ancien élève de l'École Normale Supérieure (Paris-Saclay)

Par

Virgile Andreani
ENS Ulm

William Aufort
professeur en CPGE

Guillaume Batog
professeur en CPGE

Loïc Devilliers
professeur en CPGE

Corentin Fierobe
ENS Lyon

Quentin Guilmant
ENS Lyon

Théo Lenoir
ENS Ulm

Olivier Levillain
enseignant-chercheur en école d'ingénieurs

Florian Metzger
docteur en mathématiques

David Michel
ENS Rennes

Angèle Niclas
ENS Lyon

Cyril Ravat
professeur en CPGE

Bertrand Wiel
professeur en CPGE

Sommaire thématique de mathématiques

2015 – 2019

	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	Dénombrement et probabilités
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		•••	•••	•	••	•				••		•		

Sommaire

Mines-Ponts

Polytechnique-ENS

Formulaires

SESSION 2019 **MPMA102**

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES 1

Lundi 29 avril : 14 h - 18 h

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.

Les calculatrices sont interdites

Le sujet est composé de deux exercices et d'un problème, tous indépendants.

EXERCICE I

On admet que $\displaystyle\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ et on pose, pour $t \in]0,+\infty[$, $f(t) = \dfrac{te^{-t}}{1-e^{-t}}$.

Q1. Justifier que la fonction *f* est intégrable sur $]0,+\infty[$ puis, à l'aide d'un théorème d'intégration terme à terme, calculer l'intégrale $\displaystyle\int_0^{+\infty} \frac{t}{e^t-1}\,\mathrm{d}t$.

EXERCICE II

Si X est une variable aléatoire à valeurs dans $\mathbb{N}$ de loi de probabilité donnée par : $\forall n \in \mathbb{N}$, $p_n = P(X = n)$, la fonction génératrice de X est $G_X(t) = E(t^X) = \displaystyle\sum_{n=0}^{+\infty} p_n t^n$.

Q2. Démontrer que l'intervalle $]-1,1[$ est inclus dans l'ensemble de définition de la fonction G_X .

Soient X_1 et X_2 deux variables aléatoires indépendantes à valeurs dans $\mathbb{N}$.
On pose $S = X_1 + X_2$, démontrer que pour tout $t \in]-1, 1[$, $G_S(t) = G_{X_1}(t).G_{X_2}(t)$ par deux méthodes : l'une utilisant le produit de Cauchy de deux séries entières et l'autre utilisant uniquement la définition : $G_X(t) = E(t^X)$.

On généralise ce résultat, que l'on pourra utiliser dans la question suivante, à *n* variables aléatoires mutuellement indépendantes à valeurs dans $\mathbb{N}$ (on ne demande pas de preuve de cette récurrence).

Q3. Un sac contient quatre boules : une boule numérotée 0, deux boules numérotées 1 et une boule numérotée 2.
On effectue *n* tirages d'une boule avec remise et on note S_n la somme des numéros tirés.
Déterminer pour tout $t \in]-1, 1[$, $G_{S_n}(t)$ et en déduire la loi de S_n .

PROBLÈME

Introduction

Dans ce sujet une série de fonctions L_a est une série de fonctions $\sum_{n\geq 1} a_n \dfrac{x^n}{1-x^n}$ où $(a_n)_{n\geq 1}$ est une suite de réels telle que la série entière $\sum_{n\geq 1} a_n x^n$ soit de rayon 1.

Partie I - Propriétés

Soit une série de fonctions L_a : $\sum_{n\geq 1} a_n \dfrac{x^n}{1-x^n}$.

Q4. Si $x \in \left]-1,1\right[$, donner un équivalent de $1-x^n$ pour n au voisinage de $+\infty$.

Démontrer que pour tout $x \in \left]-1,1\right[$, la série $\sum_{n\geq 1} a_n \dfrac{x^n}{1-x^n}$ converge absolument.

<u>Remarque</u> : la série L_a peut parfois converger en dehors de l'intervalle $\left]-1,1\right[$. Donner un exemple de suite $(a_n)_{n\geq 1}$ telle que la série L_a converge en au moins un x_0 n'appartenant pas à l'intervalle $\left]-1,1\right[$.

Q5. Démontrer que la série de fonctions $\sum_{n\geq 1} a_n \dfrac{x^n}{1-x^n}$ converge uniformément sur tout segment $[-b,b]$ inclus dans l'intervalle $\left]-1,1\right[$.

Q6. On pose, pour tout $x \in \left]-1,1\right[$, $f(x) = \sum_{n=1}^{+\infty} a_n \dfrac{x^n}{1-x^n}$.

Justifier que la fonction f est continue sur l'intervalle $\left]-1,1\right[$ et démontrer ensuite que la fonction f est de classe C^1 sur l'intervalle $\left]-1,1\right[$. Donner la valeur de $f'(0)$.

Q7. Expression sous forme de série entière

On note $A = \mathbb{N}^* \times \mathbb{N}^*$.

Lorsque $(u_{n,p})_{(n,p)\in A}$ est une famille sommable de réels, justifier que

$$\sum_{n=1}^{+\infty}\left(\sum_{p=1}^{+\infty} u_{n,p}\right) = \sum_{n=1}^{+\infty}\left(\sum_{(k,p)\in I_n} u_{k,p}\right), \text{ où } I_n = \{(k,p)\in A, kp = n\}.$$

Démontrer que pour tout $x \in \left]-1,1\right[$, la famille $(a_n x^{np})_{(n,p)\in A}$ est sommable.

En déduire que pour tout $x \in \left]-1,1\right[$, $\sum_{n=1}^{+\infty} a_n \dfrac{x^n}{1-x^n} = \sum_{n=1}^{+\infty} b_n x^n$ où $b_n = \sum_{d|n} a_d$

($d|n$ signifiant d divise n).

CCINP Maths 1 MP 2019 — Corrigé

Ce corrigé est proposé par Angèle Niclas (ENS Lyon) ; il a été relu par Thierry Limoges (professeur en CPGE) et Benoit Chevalier (ENS Ulm).

Ce sujet d'analyse est composé de deux exercices indépendants et d'un problème rédigé en deux parties. Dans ce problème, on s'intéresse à l'étude d'une suite particulière de fonctions, définie en utilisant des séries entières. La première partie reste entièrement théorique tandis que la deuxième offre des applications concrètes.

- Le premier exercice s'intéresse à une application du théorème d'intégration terme à terme pour calculer la valeur d'une intégrale généralisée. Il demande dans un premier temps de vérifier que cette intégrale généralisée existe, puis de justifier très rigoureusement l'interversion série/intégrale nécessaire au calcul.
- Le deuxième exercice, consacré aux probabilités, porte sur une application des fonctions génératrices. La première question demande de démontrer un résultat très classique sur le comportement de la fonction génératrice d'une somme de variables aléatoires. Dans la deuxième question, on applique ce résultat pour montrer comment il est possible de retrouver la loi d'une variable aléatoire en calculant sa fonction génératrice.
- La première partie du problème est très théorique et sert à démontrer des résultats utilisés dans la seconde partie. Elle demande une bonne maîtrise des différentes notions de convergence des séries (simple, uniforme et normale) et des connaissances sur les séries entières. On utilisera également à la fin le théorème de sommation par parties.
- Dans la deuxième partie du problème, on s'attaque à des applications de la première partie. La difficulté est croissante, en commençant par une application basique. On utilise ensuite le théorème de la double limite et le critère spécial des séries alternées pour obtenir des équivalents de fonctions et la valeur d'une série numérique.

Le sujet est plutôt court, avec uniquement 12 questions, et demande donc un soin particulier dans la rédaction et dans les arguments avancés pour justifier les résultats. Beaucoup de démarches sont réutilisées plusieurs fois, mais demandent souvent des justifications pointues. Les connaissances nécessaires sont concentrées presque uniquement sur le programme de deuxième année, et balayent une grande partie des méthodes apprises en lien avec l'analyse. Le premier exercice donne l'occasion de bien travailler les intégrales généralisées et l'interversion intégrale/série. Le problème est une excellente révision des chapitres sur les séries et suites de fonctions.

Indications

Exercice I

1 Chercher un équivalent de f en 0 puis une comparaison au voisinage de $+\infty$. Exprimer ensuite f sous la forme d'une série en se servant du développement en série entière de $x \mapsto 1/(1-x)$.

Exercice II

2 Utiliser le fait que si deux variables aléatoires X_1 et X_2 sont indépendantes alors pour toute fonction f, $f(X_1)$ et $f(X_2)$ sont aussi indépendantes.

3 Utiliser la question 2 en remarquant que les tirages successifs X_k sont indépendants et de même loi.

Problème

4 Utiliser le fait que $\sum a_n x^n$ est de rayon de convergence 1. Pour la convergence hors de l'intervalle, on pourra chercher une série a_n telle que L_a converge en $x_0 = 2$.

5 Prouver la convergence normale de la série plutôt que sa convergence uniforme.

6 Utiliser le résultat de la question 5, puis le théorème de dérivation d'une suite de fonctions. Pour établir la convergence uniforme des ${f_n}'$, on pourra s'inspirer de la démarche de la question 5.

7 Utiliser le théorème de sommation par paquets puis un raisonnement analogue à celui de la question 5 pour la sommabilité des $(a_n x^{np})_{(n,p)\in A}$.

8 Utiliser le résultat de la question 7.

9 Remarquer que $1 \leqslant a_n \leqslant n$ pour trouver le rayon de convergence de $\sum a_n x^n$. Pour la dernière partie de la question, exprimer $\sum n x^n$ comme la dérivée d'une série entière connue.

10 Utiliser le critère spécial des séries alternées pour montrer la convergence uniforme de la série.

11 Reprendre le raisonnement de la question 5 pour montrer la convergence uniforme de $f(x)/x$.

12 Utiliser le critère spécial des séries alternées pour montrer la convergence uniforme de la série. Pour obtenir une majoration uniforme, on pourra remarquer que

$$\forall k \in [\![\, 0\,; n-1 \,]\!] \quad x^k \geqslant x^n$$

Exercice I

$\boxed{1}$ Soit $t \in \left]\,0\,;+\infty\,\right[$, alors $\mathrm{e}^{-t} < 1$ et le dénominateur de f ne s'annule pas. La fonction $t \mapsto 1/(1-\mathrm{e}^{-t})$ est donc continue sur $\left]\,0\,;+\infty\,\right[$ et par produit de fonctions continues, f est également continue sur $\left]\,0\,;+\infty\,\right[$. Elle est ainsi intégrable sur tout segment contenu dans $\left]\,0\,;+\infty\,\right[$. Identifions le comportement de f aux bornes de l'intervalle.

- Comportement en 0^+ : le développement limité de $t \mapsto \mathrm{e}^{-t}$ en 0^+ est

$$\mathrm{e}^{-t} \underset{t\to 0^+}{\sim} 1-t$$

d'où
$$1-\mathrm{e}^{-t} \underset{t\to 0^+}{\sim} t \qquad \text{et} \qquad t\,\mathrm{e}^{-t} \underset{t\to 0^+}{\sim} t$$

Par division des deux équivalents, on obtient immédiatement que

$$f(t) \underset{t\to 0^+}{\sim} 1 \qquad \text{d'où} \qquad f(t) \xrightarrow[t\to 0^+]{} 1$$

La fonction f peut être prolongée par continuité en 0^+ et en particulier elle est intégrable sur tout segment $\left]\,0\,;\mathrm{A}\,\right[$ où $\mathrm{A} \in \mathbb{R}$.

- Comportement en $+\infty$: comme $1-\mathrm{e}^{-t} \xrightarrow[t\to+\infty]{} 1$, on sait que

$$f(t) = \frac{t\,\mathrm{e}^{-t}}{1-\mathrm{e}^{-t}} \underset{t\to+\infty}{\sim} t\,\mathrm{e}^{-t}$$

Or par croissance comparée,

$$t\,\mathrm{e}^{-t} = \underset{t\to+\infty}{\mathrm{o}}\left(\frac{1}{t^2}\right)$$

Puisque la fonction $t \mapsto 1/t^2$ est intégrable sur $\left]\,1\,;+\infty\,\right[$, par comparaison entre intégrales, f est également intégrable en $+\infty$.

Ainsi, $\boxed{\text{La fonction } f \text{ est intégrable sur } \left]\,0\,;+\infty\,\right[.}$

On remarque ensuite que

$$\forall t \in \left]\,0\,;+\infty\,\right[\qquad f(t) = \frac{t\,\mathrm{e}^{-t}}{1-\mathrm{e}^{-t}} = \frac{t}{\mathrm{e}^{t}-1}$$

On cherche donc à calculer l'intégrale de f, qui est bien définie car f est intégrable sur $\left]\,0\,;+\infty\,\right[$. L'énoncé recommande d'utiliser un théorème d'intégration terme à terme ce qui pousse à écrire f sous la forme d'une série. Utilisons le développement usuel en série entière

$$\forall x \in \left]\,-1\,;1\,\right[\qquad \frac{1}{1-x} = \sum_{n=0}^{+\infty} x^n$$

Comme $0 < \mathrm{e}^{-t} < 1$ si $t \in \left]\,0\,;+\infty\,\right[$, on en déduit

$$\forall t \in \left]\,0\,;+\infty\,\right[\qquad f(t) = t\,\mathrm{e}^{-t}\sum_{n=0}^{+\infty}(\mathrm{e}^{-t})^n = \sum_{n=0}^{+\infty} t\,\mathrm{e}^{-(n+1)t} = \sum_{n=1}^{+\infty} t\,\mathrm{e}^{-nt}$$

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

$$f_n(t) = t\,\mathrm{e}^{-nt} \qquad \text{et} \qquad \mathrm{I}_n = \int_0^{+\infty} f_n(t)\,\mathrm{d}t$$

Notons que pour tout $t \in \left]0\,;+\infty\right[$, $f_n(t) > 0$. Afin de vérifier les hypothèses du théorème d'intégration terme à terme, on va calculer la valeur de I_n grâce à une intégration par parties. Pour cela :

- la fonction $t \mapsto t$ est de classe $\mathscr{C}^1$ sur $\mathbb{R}^+$, de dérivée $t \mapsto 1$;
- la fonction $t \mapsto \mathrm{e}^{-nt}$ est la dérivée de $t \mapsto -\mathrm{e}^{-nt}/n$, aussi de classe $\mathscr{C}^1$ sur $\mathbb{R}^+$.

Ainsi,
$$\begin{aligned}\mathrm{I}_n &= \int_0^{+\infty} t\mathrm{e}^{-nt}\,\mathrm{d}t = \left[-\frac{\mathrm{e}^{-nt}}{n}t\right]_0^{+\infty} + \frac{1}{n}\int_0^{+\infty} \mathrm{e}^{-nt}\,\mathrm{d}t \\ &= 0 + \frac{1}{n}\left[-\frac{\mathrm{e}^{-nt}}{n}\right]_0^{+\infty} \\ \mathrm{I}_n &= \frac{1}{n^2}\end{aligned}$$

Vérifions les hypothèses du théorème d'intégration terme à terme :

- la fonction f_n est continue sur $\left]0\,;+\infty\right[$;
- la fonction $t \mapsto \sum\limits_{n=1}^{+\infty} f_n(t) = f(t)$ est continue sur $\left]0\,;+\infty\right[$;
- D'après l'énoncé, la série suivante converge :

$$\sum_{n\geqslant 1}\int_0^{+\infty} |f_n(t)|\,\mathrm{d}t = \sum_{n\geqslant 1}\int_0^{+\infty} f_n(t)\,\mathrm{d}t = \sum_{n\geqslant 1}\mathrm{I}_n = \sum_{n\geqslant 1}\frac{1}{n^2}$$

On conclut alors que

$$\int_0^{+\infty} f(t)\,\mathrm{d}t = \int_0^{+\infty}\sum_{n=1}^{+\infty} f_n(t)\,\mathrm{d}t = \sum_{n=1}^{+\infty}\int_0^{+\infty} f_n(t)\,\mathrm{d}t = \sum_{n=1}^{+\infty}\mathrm{I}_n = \frac{\pi^2}{6}$$

puis que
$$\boxed{\int_0^{+\infty} \frac{t}{\mathrm{e}^{t}-1}\,\mathrm{d}t = \frac{\pi^2}{6}}$$

Exercice 2

$\boxed{2}$ Comme $p_n = \mathrm{P}(\mathrm{X} = n)$, il vient que $|p_n| \leqslant 1$ et si $t \in \left]-1\,;1\right[$ alors $|p_n t^n| \leqslant |t|^n$ qui est le terme général d'une série géométrique convergente car $|t| < 1$. Ainsi $\mathrm{G_X}(t)$ converge absolument pour $t \in \left]-1\,;1\right[$ donc elle converge pour $t \in \left]-1\,;1\right[$.

$$\boxed{\text{L'intervalle } \left]-1\,;1\right[\text{ est inclus dans l'ensemble de définition de } \mathrm{G_X}.}$$

Soient X_1 et X_2 définies dans l'énoncé. Commençons par montrer l'affirmation demandée de manière directe. Soit $t \in \left]-1\,;1\right[$, on définit $f_t(x) = t^x$. Comme X_1 et X_2 sont indépendantes, $f_t(\mathrm{X}_1)$ et $f_t(\mathrm{X}_2)$ le sont également. On obtient alors en utilisant la notation de l'énoncé $\mathrm{S} = \mathrm{X}_1 + \mathrm{X}_2$ que

$$\mathrm{G_S}(t) = \mathrm{E}(t^{\mathrm{X}_1+\mathrm{X}_2}) = \mathrm{E}(f_t(\mathrm{X}_1)f_t(\mathrm{X}_2)) = \mathrm{E}(f_t(\mathrm{X}_1))\mathrm{E}(f_t(\mathrm{X}_2)) = \mathrm{G}_{\mathrm{X}_1}(t)\mathrm{G}_{\mathrm{X}_2}(t)$$

CCINP Maths 2 MP 2019 — Corrigé

Ce corrigé est proposé par Théo Lenoir (ENS Ulm) ; il a été relu par Bertrand Wiel (enseignant en CPGE) et Florian Metzger (docteur en mathématiques).

L'épreuve se compose de deux exercices et d'un problème, tous indépendants.

- Le premier exercice concerne l'informatique pour tous et consiste à coder un algorithme retournant la décomposition d'un entier en facteurs premiers.
- Le second exercice porte sur les espaces euclidiens. On y étudie quelques propriétés des endormorphismes antisymétriques et des endomorphismes normaux, c'est-à-dire qui commutent avec leur adjoint : l'adjoint d'un endomorphisme u est l'endomorphisme associé à ${}^t\,\mathrm{Mat}\ u$.
- Le problème a pour thème la réduction et plus particulièrement les matrices semblables. Tout d'abord, on regarde des propriétés invariantes par similitude et on étudie quelques cas particuliers. On démontre également un critère de diagonalisabilité pour les endomorphismes de rang 1. On prouve également des conditions suffisantes pour que deux matrices soient semblables dans $\mathscr{M}_n(\mathbb{R})$ dans des cas particuliers. Enfin, on étudie une propriété reliant polynômes minimaux, polynômes caractéristiques et similitude, qui se révèle vraie en dimensions 2 et 3, mais fausse en dimension 4.

Le sujet est plutôt court, assez facile et comporte quelques questions proches du cours. Le premier exercice permet de revoir efficacement le programme d'informatique. Le problème constitue une bonne révision du cours sur la réduction même si on y déplore un certain manque d'unité et d'originalité.

Indications

Exercice I

2 Se servir de la fonction `estPremier`.

4 Exploiter le fait que si p divise n, $v_p(n) = v_p(n/p) + 1$.

5 Utiliser la fonction `liste_premiers` pour obtenir la liste des nombres premiers entre 1 et n puis utiliser la fonction `valuation_p_adique` pour chaque nombre de la liste.

Exercice II

6 Si la dimension n de E est supérieure à 2, noter $(e_1, \ldots, e_n)$ une base orthonormée de E et considérer v l'endomorphisme qui à e_1 associe e_2, à e_2 associe $-e_1$, et à e_i associe 0 pour $i \geqslant 3$.

7 Pour iii $\Longrightarrow$ ii, penser à la formule de polarisation. Pour ii $\Longrightarrow$ i, calculer, pour (x, y) dans E^2, $\langle x \mid v \circ u(y) \rangle$.

Problème

10 Pour la première méthode, regarder la matrice de u dans la base (e_2, e_1, e_3). Pour la seconde méthode, calculer les polynômes caractéristiques de A et B.

11 Montrer que le noyau de u est de dimension $n - 1$ et regarder la matrice de u dans une base adaptée.

12 Trouver un polynôme de degré 2 qui annule u.

13 Considérer la matrice $\mathrm{M} = \begin{pmatrix} \mathrm{i} & 1 \\ 1 & -\mathrm{i} \end{pmatrix}$.

14 Vérifier que ${}^t\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ et ${}^t\begin{pmatrix} 1 & -1 & 1 & -1 \end{pmatrix}$ sont des vecteurs propres de A.

15 Calculer PAP^{-1} pour $\mathrm{P} = \begin{pmatrix} 1 & 0 \\ 0 & b/a \end{pmatrix}$.

16 Montrer que PB = AP et regarder les parties réelles et imaginaires des coefficients.

17 Utiliser la formule explicite du déterminant. Que vaut $\det(\mathrm{R} + \mathrm{iS})$?

18 Utiliser le résultat de la question 17.

19 Montrer que A est diagonalisable et que B a pour polynôme caractéristique $\mathrm{X}^3 + \mathrm{X}$.

20 Regarder le cas où le polynôme caractéristique est scindé à racines simples dans lequel on pourra utiliser le résultat de la question 18, puis le cas où il est de la forme $(\mathrm{X} - \lambda)^2$. Montrer que λ est réel et penser à trigonaliser A et B. Utiliser ensuite le résultat de la question 15 pour le cas où aucune des deux matrices n'est une homothétie.

21 Chercher deux matrices triangulaires supérieures M vérifiant $\mathrm{M}^2 = 0$, une de rang 1 et une de rang 2.

Exercice I

$\boxed{1}$ On commence par traiter le cas $n = 1$: 1 n'est pas premier. Ensuite on vérifie comme indiqué par l'énoncé si n est divisible par un entier entre 2 et $\lfloor\sqrt{n}\rfloor$. Si ce n'est pas le cas, n est premier donc on renvoie `True`.

```
def estPremier(n):
    if n == 1:
        return False
    for i in range (2, int(sqrt(n))+1):
        if n % i == 0:
            return False
    return True
```

$\boxed{2}$ Utilisons le résultat de la question précédente : on crée une liste vide, puis on parcourt les entiers entre 1 et n en ajoutant à la liste tous ceux qui sont premiers.

```
def liste_premiers(n):
    L = []
    for i in range (1, n+1):
        if estPremier(i):
            L.append(i)
    return L
```

$\boxed{3}$ On utilise la méthode donnée par l'énoncé : tant que n est divisible par p on effectue la division et on compte le nombre de divisions effectuées.

```
def valuation_p_adique (n, p):
    valuation = 0
    while n%p == 0:
        valuation += 1
        n = n//p
    return valuation
```

$\boxed{4}$ Implémentons l'algorithme suivant : si p ne divise pas n, on renvoie 0. Sinon la valuation p-adique de n vaut 1 plus celle de n/p. On renvoie donc $1 + val(n/p, p)$.

```
def val(n, p):
    if n%p != 0:
        return 0
    return 1 + val(n//p, p)
```

$\boxed{5}$ On utilise les fonctions précédentes. On crée la liste des nombres premiers entre 1 et n. Ensuite pour tout p dans cette liste on calcule la valuation p-adique de n, et, si elle est non nulle, on ajoute la liste $[p, v_p(n)]$ à la liste résultat.

```
def decomposition_facteurs_premiers(n):
    L = liste_premiers(n)
    T = []
    for p in L:
        a = valuation_p_adique(n, p)
        if a != 0:
            T.append([p, a])
    return T
```

Exercice II

6 Soit u un endomorphisme vérifiant $\langle u(x), x\rangle = 0$ pour tout x dans E. Si la dimension de E est 0 ou 1 alors u est l'endomorphisme nul. En effet si E est de dimension 0, alors tout endomorphisme de u est nul. Si E est de dimension 1, il existe une base orthonormée de E qu'on note (e_1). Il existe $\lambda \in \mathbb{R}$ tel que $u(e_1) = \lambda e_1$. Ainsi,

$$0 = \langle u(e_1), e_1\rangle = \lambda\langle e_1, e_1\rangle = \lambda$$

On obtient $u(e_1) = 0$, donc u est nul.

Par contre si la dimension de E est supérieure ou égale à 2, le résultat n'est plus vrai. Notons $\mathrm{B} = (e_1, \ldots, e_n)$ une base orthonormée de E. Considérons l'endomorphisme v défini par

$$\begin{aligned} v(e_1) &= e_2 \\ v(e_2) &= -e_1 \end{aligned}$$

et pour tout $i \in [\![\, 3\,; n \,]\!]$ $\qquad v(e_i) = 0$

Soit x dans E, en décomposant x dans la base B sous la forme $x_1e_1 + \cdots + x_ne_n$ avec $(x_1, \ldots, x_n) \in \mathbb{R}^n$ on obtient par bilinéarité du produit scalaire :

$$\begin{aligned} \langle v(x), x\rangle &= \left\langle v\left(\sum_{i=1}^n x_ie_i\right), \sum_{j=1}^n x_je_j\right\rangle \\ &= \left\langle \sum_{i=1}^n x_iv(e_i), \sum_{j=1}^n x_je_j\right\rangle \\ &= \sum_{j=1}^n \langle x_1e_2 - x_2e_1, x_je_j\rangle \\ &= \sum_{j=1}^n x_1x_j\langle e_2, e_j\rangle - \sum_{j=1}^n x_2x_j\langle e_1, e_j\rangle \\ &= x_1x_2 - x_1x_2 \\ \langle v(x), x\rangle &= 0 \end{aligned}$$

Comme la famille (e_i) est une base, $v(e_1) = e_2 \neq 0$ est non nul donc l'endomorphisme v est non nul.

$$\boxed{\text{L'endomorphisme } u \text{ n'est pas nécessairement nul.}}$$

> Les endomorphismes u de E qui vérifient pour tout x dans E l'égalité $\langle u(x), x\rangle = 0$ sont exactement les endomorphismes dont la matrice dans une base orthonormale est antisymétrique.
>
> En effet notons $(e_1, \ldots, e_n)$ une base orthonormale de E. Soit u un endomorphisme de E vérifiant pour tout x dans E $\langle u(x), x\rangle = 0$. On pose M la matrice de u dans la base $(e_1, \ldots, e_n)$. Soit $(x, y) \in \mathrm{E}^2$. Par bilinéarité du produit scalaire,
>
> $$\begin{aligned} 0 &= \langle u(x+y), x+y\rangle \\ &= \langle u(x), x\rangle + \langle u(x), y\rangle + \langle u(y), x\rangle + \langle u(y), y\rangle \\ 0 &= \langle u(y), x\rangle + \langle u(x), y\rangle \end{aligned}$$
>
> En particulier pour tout $(i, j) \in \{1, \ldots, n\}^2$:
>
> $$\mathrm{M}_{i,j} = \langle u(e_j), e_i\rangle = -\langle u(e_i), e_j\rangle = -\mathrm{M}_{i,j}$$
>
> La matrice M est bien antisymétrique.

Centrale Maths 1 MP 2019 — Corrigé

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

Ce sujet d'algèbre linéaire propose d'étudier une certaine classe d'endomorphismes, les *endomorphismes cycliques*, ainsi que leurs équivalents matriciels, les très classiques *matrices compagnons*. Les questions autour de ces objets visent notamment à redémontrer certains résultats puissants, comme le théorème de Cayley-Hamilton ou la décomposition de Frobenius d'un espace vectoriel associée à un endomorphisme. Elles nécessitent des connaissances solides en algèbre linéaire, notamment sur la réduction et sur les polynômes d'endomorphismes.

- La partie I analyse les liens entre endomorphismes cycliques et matrices compagnons, présentant au passage des propriétés sur leurs polynômes minimaux et caractéristiques. Elle fait redémontrer le théorème de Cayley-Hamilton en utilisant ces objets.
- La partie II complète les résultats de la partie I afin d'obtenir une condition nécessaire et suffisante pour qu'un endomorphisme quelconque soit cyclique : lorsque E est un $\mathbb{C}$-espace vectoriel, $f \in \mathscr{L}(\mathrm{E})$ est cyclique si et seulement si la famille $(\mathrm{Id}\,, f, \dots, f^{n-1})$ est libre, où n est la dimension de l'espace E.
- La partie III propose une démonstration du théorème de décomposition de Frobenius d'un espace associée à un endomorphisme : *pour un endomorphisme noté f, il existe des sous-espaces vectoriels $\mathrm{E}_1, \dots, \mathrm{E}_r$ de E stables par f, tels que*
  - $\mathrm{E} = \mathrm{E}_1 \oplus \cdots \oplus \mathrm{E}_r$ *;*
  - *f induit sur chaque E_i un endomorphisme cyclique ;*
  - *le polynôme minimal de f sur E_{i+1} divise celui de f sur E_i.*

  On s'en sert pour montrer des propriétés sur le commutant d'un endomorphisme en lien avec le caractère cyclique de celui-ci.
- La partie IV étudie les endomorphismes orthocycliques, un cas particulier d'endomorphisme cyclique sur un espace euclidien.

La principale difficulté de ce sujet réside dans sa longueur. De nombreuses questions sont très classiques et devaient être résolues rapidement afin de se concentrer sur les quelques questions plus délicates.

Indications

Partie I

1 Comparer χ_{M} et $\chi_{{}^t\mathrm{M}}$.

3 On pourra faire une opération élémentaire judicieuse sur la première ligne, ou développer par rapport à la dernière colonne.

4 Pour un vecteur X, calculer ${}^t\mathrm{C}_{\mathrm{Q}}\,\mathrm{X}$. Si, de plus, X est un vecteur propre de ${}^t\mathrm{C}_{\mathrm{Q}}$ pour la valeur propre λ, montrer qu'il est lié au vecteur $(\lambda^i)_{0\leqslant i\leqslant n-1}$.

5 Pour le sens direct, décomposer $f(e_n)$ dans la base $\mathcal{B}$.

6 Si f est diagonalisable, considérer la décomposition en sous-espaces propres de la matrice ${}^t\mathrm{C}_{\mathrm{Q}}$, où C_{Q} est une matrice compagnon représentant f.

7 Remarquer que si Q est un polynôme de degré d tel que $\mathrm{Q}(f) = 0$, alors la famille $(\mathrm{Id}\,, f, \ldots, f^{d-1})$ est liée.

8 Considérer le plus grand entier p tel que la famille $(x, f(x), \ldots, f^{p-1}(x))$ est libre, en justifiant son existence.

10 Utiliser les résultats des questions 8, 9 et 3.

11 Montrer que $\chi_f(f)(x) = 0$ pour tout vecteur x non nul de E.

Partie II

12 Montrer que si x_0 est tel que $f^{n-1}(x_0) \neq 0$, alors la famille $(x_0, \ldots, f^{n-1}(x_0))$ est libre.

13 Utiliser le lemme des noyaux.

16 On pourra utiliser la question 12 pour montrer que $\nu_k \leqslant m_k$ pour tout k. Raisonner ensuite par l'absurde en supposant qu'il existe un k tel que $\nu_k < m_k$ pour trouver un polynôme non nul de degré $< n$ qui annule f.

17 Prouver que $\nu_k = \dim \mathrm{F}_k$ pour tout k.

18 Utiliser le fait que $u_1 \in \mathrm{F}_1, \ldots, u_{m_1+\cdots+m_{p-1}+1} \in \mathrm{F}_p$, que $\mathrm{F}_1, \ldots, \mathrm{F}_p$ sont stables par f, et la décomposition $\mathrm{E} = \mathrm{F}_1 \oplus \cdots \oplus \mathrm{F}_p$.

Partie III

22 On pourra montrer que $g = \sum\limits_{k=0}^{n-1} \lambda_k f^k$.

24 Vérifier que, sans perte de généralité, on peut supposer qu'il existe un vecteur $x_1 \in \mathrm{F}_1 \smallsetminus \cup_{j\neq 1}\mathrm{F}_j$, et un vecteur $x_2 \in \mathrm{F}_2 \smallsetminus \cup_{j\neq 2}\mathrm{F}_j$, puis considérer les vecteurs $x_1 + tx_2$ pour $t \in \mathbb{R}$.

25 Vérifier que $\cup_{x\in\mathrm{E}\smallsetminus\{0\}} \operatorname{Ker} \pi_{f,x} = \mathrm{E}$ et que cette réunion est en fait finie.

29 Pour $x \in \mathrm{E}_1$, relier $\Psi(x)$ aux coordonnées de x dans une base de E_1 à déterminer.

30 Montrer que $\mathrm{F} = \operatorname{Ker} \Psi$ et appliquer le théorème du rang.

31 Envisager une récurrence sur la dimension de E.

32 Utiliser le résultat de la question 31 pour exhiber une famille libre de cardinal n d'endomorphismes qui commutent avec f.

33 Étudier la dimension de ces deux espaces.

Partie IV

34 Utiliser le théorème de réduction des isométries.

35 Si C_Q est une matrice compagnon représentant f dans une base orthonormale, ses colonnes sont orthonormées pour le produit scalaire canonique.

36 Pour cette question classique, on pourra d'abord montrer que f se triangularise dans une certaine base, puis utiliser Gram-Schmidt.

37 Pour le sens indirect, on pourra essayer d'adapter la base trouvée à la question 36, afin de transformer la matrice triangulaire correspondante en une matrice de la forme

$$\begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

I. Matrices compagnons et endomorphismes cycliques

1 On constate que par symétrie de XI_n et linéarité de la transposée

$$\mathrm{XI}_n - {}^t\mathrm{M} = {}^t(\mathrm{XI}_n) - {}^t\mathrm{M} = {}^t(\mathrm{XI}_n - \mathrm{M})$$

Ainsi, $$\chi_{{}^t\mathrm{M}} = \det {}^t(\mathrm{XI}_n - \mathrm{M})$$

Comme le déterminant d'une matrice est le même que celui de sa transposée

$$\chi_{{}^t\mathrm{M}} = \det {}^t(\mathrm{XI}_n - \mathrm{M}) = \det(\mathrm{XI}_n - \mathrm{M}) = \chi_\mathrm{M}$$

En particulier les polynômes $\chi_{{}^t\mathrm{M}}$ et χ_M ont les mêmes racines. Or les valeurs propres d'une matrice sont les racines de son polynôme caractéristique. Ainsi,

$$\boxed{\text{Les matrices M et } {}^t\mathrm{M} \text{ ont le même spectre.}}$$

2 Prouvons d'abord que pour toute matrice $\mathrm{A} \in \mathscr{M}_n(\mathbb{K})$ diagonalisable, ${}^t\mathrm{A}$ est aussi diagonalisable. Il suffira alors d'appliquer le résultat à $\mathrm{A} = \mathrm{M}$, puis à $\mathrm{A} = {}^t\mathrm{M}$ en utilisant le fait que ${}^t({}^t\mathrm{M}) = \mathrm{M}$, pour obtenir l'équivalence voulue.

Soit $\mathrm{A} \in \mathscr{M}_n(\mathbb{K})$ diagonalisable. Il existe une matrice inversible P et une matrice diagonale D telles que $\mathrm{A} = \mathrm{P}^{-1}\mathrm{DP}$. En transposant cette égalité, on obtient

$$ {}^t\mathrm{A} = {}^t(\mathrm{P}^{-1}\mathrm{DP}) = {}^t\mathrm{P}\,{}^t\mathrm{D}\,{}^t(\mathrm{P}^{-1})$$

Or la matrice $\mathrm{Q} = {}^t(\mathrm{P}^{-1})$ est inversible, d'inverse $\mathrm{Q}^{-1} = {}^t\mathrm{P}$. Puisque D est diagonale, ${}^t\mathrm{D} = \mathrm{D}$ et ainsi

$$ {}^t\mathrm{A} = \mathrm{Q}^{-1}\,{}^t\mathrm{D}\,\mathrm{Q} = \mathrm{Q}^{-1}\mathrm{DQ}$$

Ceci permet de conclure que ${}^t\mathrm{A}$ est diagonalisable. Compte tenu de la remarque initiale,

$$\boxed{\text{La matrice } {}^t\mathrm{M} \text{ est diagonalisable si et seulement si M est diagonalisable.}}$$

3 On peut écrire

$$\det(\mathrm{XI}_n - \mathrm{C_Q}) = \begin{vmatrix} \mathrm{X} & 0 & \cdots & \cdots & 0 & a_0 \\ -1 & \mathrm{X} & 0 & \cdots & 0 & a_1 \\ 0 & -1 & \ddots & & \vdots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & -1 & \mathrm{X} & a_{n-2} \\ 0 & \cdots & \cdots & 0 & -1 & \mathrm{X} + a_{n-1} \end{vmatrix}$$

On effectue alors l'opération suivante sur la première ligne

$$\mathrm{L}_1 \leftarrow \mathrm{L}_1 + \mathrm{XL}_2 + \mathrm{X}^2\mathrm{L}_3 + \ldots + \mathrm{X}^{n-1}\mathrm{L}_n$$

de sorte que le déterminant devient

$$\det(\mathrm{XI}_n - \mathrm{C_Q}) = \begin{vmatrix} 0 & 0 & \cdots & \cdots & 0 & \mathrm{Q(X)} \\ -1 & \mathrm{X} & \cdots & \cdots & 0 & a_1 \\ 0 & -1 & \ddots & & \vdots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & -1 & \mathrm{X} & a_{n-2} \\ 0 & \cdots & \cdots & 0 & -1 & \mathrm{X} + a_{n-1} \end{vmatrix}$$

Centrale Maths 2 MP 2019 — Corrigé

Ce corrigé est proposé par David Michel (ENS Rennes) ; il a été relu par Rémi Pellerin (ENS Lyon) et Florian Metzger (docteur en mathématiques).

Ce sujet traite de l'écriture binaire et du développement dyadique pour des nombres réels de $[\,0\,;1\,]$ ou des variables aléatoires réelles à valeurs dans $[\,0\,;1\,]$. Il est composé de cinq parties.

- Dans la première, on étudie la suite des fonctions caractéristiques associées à une suite de variables aléatoires. On démontre qu'elle converge simplement et on s'interroge sur sa convergence uniforme sur $\mathbb{R}$.
- Dans la deuxième partie, on s'intéresse à l'écriture binaire des entiers naturels et son lien avec la décomposition dyadique des réels $x \in [\,0\,;1\,[$ pour lesquels cette décomposition est finie (c'est-à-dire des réels dyadiques de $[\,0\,;1\,[$).
- Dans la troisième partie, on démontre que la donnée d'un n-uplet de variables aléatoires indépendantes suivant la loi de Bernoulli est équivalente à la donnée d'une variable aléatoire suivant la loi uniforme sur l'ensemble des réels de $[\,0\,;1\,[$ ayant une décomposition dyadique de longueur n.
- Dans la quatrième partie, on étudie le comportement asymptotique d'une telle variable aléatoire X_n. Plus précisément, on montre qu'elle « converge en loi » vers une variable aléatoire suivant la loi uniforme sur $[\,0\,;1\,]$, c'est-à-dire que pour toute fonction f continue (et bornée) sur $[\,0\,;1\,]$,
$$\mathbb{E}(f(\mathrm{X}_n)) \xrightarrow[n\to\infty]{} \mathbb{E}(f(\mathrm{X}))$$
où X suit la loi uniforme sur $[\,0\,;1\,]$.
- Dans la cinquième et dernière partie, on prouve que $\mathcal{P}(\mathbb{N})$ n'est pas dénombrable, mais qu'il est en bijection avec $\{0\,;1\}^{\mathbb{N}}$. Cela permet de conclure que l'intervalle $[\,0\,;1\,[$ n'est pas dénombrable puisqu'on en explicite en fin de sujet une bijection avec $\{0\,;1\}^{\mathbb{N}}$.

Ce sujet est long et, si les premières questions sont relativement faciles, les choses deviennent techniques dès la deuxième partie. Signalons que l'avant-dernière question du sujet est extrêmement difficile et demande beaucoup de recul et d'initiatives pour parvenir à la résoudre. Cela reste néanmoins un sujet riche de par les thématiques qu'il aborde (dénombrement, probabilités, suites et séries de fonctions, intégration), et par les résultats intéressants qui y sont démontrés. Face à un pareil sujet, il faut s'accrocher et se dire que l'on peut en venir à bout !

Indications

Partie I

1 Exprimer $\Phi_{X_n}(t)$ comme le produit de variables aléatoires indépendantes et utiliser la propriété de l'espérance rappelée dans l'énoncé. On rappelle que

$$\forall x \in \mathbb{R} \qquad \cos x = \frac{e^{i\,x} + e^{-i\,x}}{2}$$

2 Montrer que la suite $(\sin(t/2^n)\Phi_{X_n}(t))_{n\in\mathbb{N}^*}$ est géométrique. On rappelle que, pour tout réel x, $\sin(2x) = 2\cos(x)\sin(x)$.

6 Écrire cos comme somme d'exponentielles complexes et utiliser la parité de Φ_{X_n}.

7 Pour chaque $n \in \mathbb{N}^*$, déterminer un réel t_n tel que $\varphi_n(t_n) = -1$.

Partie II

10 Procéder par récurrence sur n et utiliser une division euclidienne.

11 Considérer les cardinaux de $\{0\,;1\}^n$ et $[\![\, 0\,; 2^n - 1\,]\!]$.

13 On rappelle que $\qquad \forall x \in \mathbb{R} \qquad \lfloor x \rfloor \leqslant x < \lfloor x \rfloor + 1$

14 Utiliser la définition de $d_j(x)$ pour faire apparaître une somme télescopique.

15 Montrer que $-1 < d_j(x) < 2$ en utilisant le fait que

$$\forall x \in \mathbb{R} \qquad x - 1 < \lfloor x \rfloor \leqslant x$$

16 Utiliser les questions 9 et 10.

17 Se servir des questions 11 et 16.

18 Faire une distinction de cas selon que $n > k$ ou $n \leqslant k$.

Partie III

20 Utiliser la question 17 pour déterminer la loi de Y_n avant de calculer F_n.

21 Remarquer que, pour tout $y \in D_n$, $\{Y_n < y\} = \{Y_n \leqslant y\} \smallsetminus \{Y_n = y\}$ et utiliser le résultat de la question 20 ainsi que la loi de Y_n.

23 Poser $V = (V_1, \dots, V_n) = {\Psi_n}^{-1}(X_n)$ puis montrer que V suit la loi uniforme sur $\{0\,;1\}^n$.

Partie IV

24 Remarquer que, pour tout $n \in \mathbb{N}^*$,

$$\{Y_{n+1} \leqslant x\} \subset \{Y_n \leqslant x\} \qquad \text{et} \qquad \{Y_{n+1} < x\} \subset \{Y_n < x\}$$

27 Utiliser la question 13 pour obtenir un encadrement de $F_n(x)$.

28 Distinguer les cas selon que I est de la forme $[\,a\,;b\,]$, $[\,a\,;b\,[$, $]\,a\,;b\,]$ ou $]\,a\,;b\,[$.

29 Reconnaître une somme de Riemann.

30 Considérer, pour tout $k \in [\![\,1\,;n\,]\!]$, $U_k = (\varepsilon_k + 1)/2$.

31 Utiliser la question 29 pour calculer l'intégrale de deux façons différentes. On aura besoin du théorème de transfert ainsi que du théorème de convergence dominée.

Partie V

33 Considérer un antécédent de A par f et déterminer s'il appartient à A.

35 Utiliser la question 14.

36 Remarquer que

$$\left\{\begin{array}{l}\Psi((x_n)) \in \mathrm{D} \cup \{1\} \\ (x_n) \text{ est stationnaire à } 1\end{array}\right. \qquad\Longleftrightarrow\qquad 0 < \Lambda((x_n)) \leqslant 1/2$$

$$\text{et} \qquad \left\{\begin{array}{l}\Psi((x_n)) \in \mathrm{D}^* \\ (x_n) \text{ est stationnaire à } 0\end{array}\right. \qquad\Longleftrightarrow\qquad 1/2 < \Lambda((x_n)) < 1$$

Avant d'étudier l'injectivité de Λ, on pourra démontrer que les éléments de l'ensemble $[\,0\,;1\,] \smallsetminus \mathrm{D}^*$ ont exactement un antécédent par Ψ, et que les éléments de D^* en ont exactement deux, à savoir une suite stationnaire à 0 et une suite stationnaire à 1.

37 Utiliser les questions 34 et 36.

I. Fonction caractéristique

1 Soient $n \in \mathbb{N}^*$ et $t \in \mathbb{R}$. Par définition,

$$\Phi_{\mathrm{X}_n}(t) = \mathbb{E}\left(\exp\left(\mathrm{i}\,t\sum_{k=1}^{n}\frac{\varepsilon_k}{2^k}\right)\right) = \mathbb{E}\left(\prod_{k=1}^{n}\exp\left(\frac{\mathrm{i}\,t\,\varepsilon_k}{2^k}\right)\right)$$

Les variables aléatoires $(\varepsilon_k)_{k\geqslant 1}$ étant mutuellement indépendantes, il en va de même des variables aléatoires $(f_k(\varepsilon_k))_{k\geqslant 1}$ où, pour tout $k \in \mathbb{N}^*$, on a posé

$$f_k : x \in \mathbb{R} \mapsto \exp\left(\frac{\mathrm{i}\,t\,x}{2^k}\right)$$

De plus, pour tout $k \in [\![\, 1\,;n\,]\!]$, l'image de Ω par la variable aléatoire $f_k(\varepsilon_k)$ est finie (de cardinal 2). On peut donc appliquer le résultat admis en introduction de l'énoncé et le théorème de transfert. On obtient :

$$\begin{aligned}\Phi_{\mathrm{X}_n}(t) &= \prod_{k=1}^{n}\mathbb{E}\left(\exp\left(\frac{\mathrm{i}\,t\,\varepsilon_k}{2^k}\right)\right)\\ &= \prod_{k=1}^{n}\left(\frac{1}{2}\exp\left(\frac{\mathrm{i}\,t}{2^k}\right)+\frac{1}{2}\exp\left(\frac{-\mathrm{i}\,t}{2^k}\right)\right)\end{aligned}$$

Ainsi,
$$\boxed{\forall n \in \mathbb{N}^* \quad \forall t \in \mathbb{R} \qquad \Phi_{\mathrm{X}_n}(t) = \prod_{k=1}^{n}\cos\left(\frac{t}{2^k}\right)}$$

> Le résultat admis par l'énoncé se démontre par récurrence sur le nombre de variables aléatoires à l'aide de la propriété suivante : si X et Y sont deux variables aléatoires indépendantes d'espérances finies, alors
> $$\mathbb{E}(\mathrm{XY}) = \mathbb{E}(\mathrm{X})\mathbb{E}(\mathrm{Y})$$

2 Soit $t \in \mathbb{R}$. Montrons par récurrence que la propriété

$$\mathscr{P}(n) : \qquad \sin\left(\frac{t}{2^n}\right)\Phi_{\mathrm{X}_n}(t) = \frac{\sin t}{2^n}$$

est vraie pour tout $n \in \mathbb{N}^*$.

- <u>$\mathscr{P}(1)$</u> est vraie car, d'après la question 1, $\Phi_{\mathrm{X}_1}(t) = \cos(t/2)$ et
$$\sin\left(\frac{t}{2}\right)\cos\left(\frac{t}{2}\right) = \frac{\sin t}{2}$$
- <u>$\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)$</u> : soit $n \in \mathbb{N}^*$ tel que $\mathscr{P}(n)$ est vraie. D'après la question 1, en isolant le facteur d'indice n du produit, on a :
$$\begin{aligned}\sin\left(\frac{t}{2^{n+1}}\right)\Phi_{\mathrm{X}_{n+1}}(t) &= \sin\left(\frac{t}{2^{n+1}}\right)\cos\left(\frac{t}{2^{n+1}}\right)\Phi_{\mathrm{X}_n}(t)\\ &= \frac{1}{2}\sin\left(\frac{t}{2^n}\right)\Phi_{\mathrm{X}_n}(t)\end{aligned}$$
$$\sin\left(\frac{t}{2^{n+1}}\right)\Phi_{\mathrm{X}_{n+1}}(t) = \frac{\sin t}{2^{n+1}} \qquad \text{(d'après } \mathscr{P}(n))$$
- <u>Conclusion</u> : $$\boxed{\forall n \in \mathbb{N}^* \quad \forall t \in \mathbb{R} \qquad \sin\left(\frac{t}{2^n}\right)\Phi_{\mathrm{X}_n}(t) = \frac{\sin t}{2^n}}$$

Centrale Informatique MP-PC-PSI 2019 — Corrigé

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

Ce sujet s'intéresse à l'élasticité de l'ADN, plus particulièrement à la relation qui existe entre une force exercée sur un brin d'ADN et l'allongement de ce brin. Cette relation permettrait d'avancer dans la compréhension de processus biologiques, comme la réplication de l'ADN.

- La première partie comporte seulement trois questions d'implémentation de fonctions utilitaires, dont deux sont explicitement au programme d'informatique pour tous.
- La deuxième partie introduit le protocole expérimental permettant d'évaluer l'intensité de la force de traction exercée artificiellement sur un brin d'ADN auquel on a accroché une bille aimantée, ainsi que l'allongement du brin. Cette partie est principalement consacrée à des algorithmes de traitement d'images (représentées par des tableaux Numpy à deux dimensions).
- Une fois les mesures effectuées, on souhaite maintenant les faire correspondre à un modèle mathématique. La troisième partie étudie le « modèle du ver », qui donne la relation entre force de traction et allongement sous la forme d'une formule mathématique faisant intervenir des caractéristiques du brin. Cette partie propose de déterminer ces caractéristiques à partir des observations effectuées et d'une fonction prédéfinie en Python, puis explore les bases de l'algorithme de minimisation sous-jacent pour des problèmes à une ou deux dimensions.
- Enfin, la dernière partie propose un second modèle, probabiliste cette fois, permettant de simuler l'allongement d'un brin d'ADN soumis à une force donnée en paramètre.

Les parties 2, 3 et 4 sont indépendantes, et le sujet couvre une grande proportion du programme d'informatique pour tous (listes et tableaux Numpy, représentation des nombres, ingénierie numérique et récursivité). La plupart des questions sont abordables dès la première année. Enfin, dans chaque partie la dernière question est plus délicate, sans être vraiment difficile.

Indications

Partie I

2 Attention à ne pas utiliser la fonction `moyenne` n fois.

3 L'utilisation d'une fonction récursive semble adaptée.

Partie II

5 Pour arrondir à l'entier le plus proche, on pourra utiliser la fonction `round`.

7 Commencer par calculer le barycentre des positions de la liste, puis dans un deuxième temps la moyenne des déplacements quadratiques. Ne pas oublier d'utiliser correctement le paramètre `t`.

8 L'énoncé suggère d'utiliser des fonctions intermédiaires. On peut commencer par déterminer le rayon extérieur du plus grand anneau en fonction de la position du centre de la bille. Puis pour chaque pixel de l'image, on cherchera à exprimer le numéro de l'anneau dans lequel il se trouve.

Partie III

11 Dans la fonction `curve_fit`, la fonction `f` ne doit pas avoir la température `T` comme argument. Il faut également extraire `xdata` et `ydata` à partir du tableau à deux colonnes `Fz`. Attention à l'ordre.

12 Le nombre de chiffres significatifs dépend uniquement de la mantisse.

13 Pour $h = 10^{-16}$, utiliser la question 12 pour savoir comment seront représentés les nombres $1 + h$ et $1 - h$ dans ce codage.

15 Le premier argument de `derive` est une fonction, il faut donc être précautionneux si on veut utiliser `derive` deux fois de suite.

16 Il faut appliquer la méthode de Newton à ϕ' et non à ϕ.

17 Le système se déduira directement des équations des plans tangents considérés.

18 Deux stratégies sont possibles : utiliser les applications partielles, ou effectuer le calcul à la main, avec des formules semblables à celle utilisée à la question 14.

19 On pourra séparer de la fonction `min_local_2D` le calcul de la matrice $\mathrm{J}(x, y)$. Pour ce calcul, commencer par définir les fonctions g_x et g_y, puis réutiliser la fonction `grad`.

Partie IV

20 Générer des nombres aléatoires dans $[\,0\,;1\,[$, puis utiliser une transformation affine pour obtenir des angles dans l'intervalle $[\,-\pi\,;\pi\,[$.

22 La question demande de créer une nouvelle conformation. Attention également aux indices manipulés.

24 Utiliser une liste comme une file pour conserver uniquement les 500 derniers allongements. On pourra commencer par initialiser les 500 premiers allongements, puis utiliser une boucle `while` pour la phase de convergence.

I. Fonctions utilitaires

$\boxed{1}$ Il suffit d'additionner tous les éléments de la séquence à l'aide d'une boucle, sans oublier de diviser par la longueur de la séquence pour obtenir la moyenne.

```
def moyenne(X):
    somme = 0
    for i in range(len(X)):
        somme += X[i]
    return somme/len(X)
```

> On peut aussi parcourir la séquence sans utiliser d'indice avec un itérateur :
>
> ```
> def moyenne(X):
>     somme = 0
>     for x in X:
>         somme += x
>     return somme/len(X)
> ```

$\boxed{2}$ On utilise la formule de König-Huygens rappelée dans l'énoncé : on calcule d'abord la moyenne des carrés comme précédemment, à laquelle on soustrait le carré de la moyenne.

```
def variance(X):
    s = 0
    for x in X:
        s += x**2
    return s/len(X) - moyenne(X)**2
```

> Une autre stratégie consiste à utiliser la définition de la variance. Attention dans ce cas à utiliser la fonction `moyenne` une seule fois avant la boucle, afin d'éviter une complexité en $O(n^2)$.
>
> ```
> def variance(X):
>     s = 0
>     m = moyenne(X)
>     for x in X:
>         s += (x - m)**2
>     return s/len(X)
> ```

$\boxed{3}$ Pour pouvoir explorer l'ensemble des éléments de la séquence imbriquée, le plus simple est d'écrire une fonction `somme` récursive. Si l'argument `M` de la fonction `somme` est un composant élémentaire, c'est-à-dire un nombre réel (que l'on peut détecter avec `isinstance(M, numbers.Real)`), on peut le renvoyer directement. Dans le cas contraire, `M` est une séquence imbriquée : on calcule alors la somme des éléments qui la compose à l'aide d'une boucle et de la fonction `somme` appliquée à chacun de ces éléments.

> L'énoncé ne suppose pas que le module `numbers` a été importé, ce qui est pourtant nécessaire pour pouvoir utiliser `numbers.Real`.

```
def somme(M):
    if isinstance(M, numbers.Real):
        return M
    else:
        s = 0
        for x in M:
            s += somme(x)
        return s
```

> Dans le code précédent, le `else` est facultatif, du fait de la présence du `return` dans le cas où `M` est un nombre réel. On pourrait donc l'enlever, ainsi que le niveau d'indentation induit dans tout le bloc suivant, ce qui peut rendre le code plus agréable à lire.

II. Mesures expérimentales

$\boxed{4}$ On commence par initialiser le tableau demandé à l'aide de la fonction `np.zeros`. Puis on parcourt le tableau initial à l'aide d'une double boucle pour mettre à jour les pixels dont la valeur est strictement inférieure au seuil, les autres étant déjà associés à la valeur 0 dans l'image seuillée.

```
def seuillage(A, seuil):
    S = np.zeros(A.shape, int)
    for i in range(A.shape[0]):
        for j in range(A.shape[1]):
            if A[i, j] < seuil:
                S[i, j] = 1
    return S
```

> Pour récupérer les dimensions du tableau en argument, on a utilisé ici l'attribut `A.shape`, qui était donné en annexe. Dans ce type d'épreuve, il est primordial de bien lire l'annexe avant de commencer le sujet, pour identifier notamment des fonctions pouvant être utiles.
>
> Une autre possibilité pour extraire les dimensions consiste à utiliser la fonction `len`. En effet, `len(A)` renvoie le nombre de lignes du tableau `A` (identique donc à `A.shape[0]`), et `len(A[0])` renvoie le nombre de colonnes de `A` (identique à `A.shape[1]`).
>
> Enfin, une dernière solution consiste à utiliser le fait que la plupart des opérations usuelles (arithmétiques et logiques) sur les tableaux Numpy agissent élément par élément. On pourrait par exemple écrire :
>
> ```
> def seuillage(A, seuil):
>     return 1 * (A < seuil)
> ```
>
> En effet, `A < seuil` renvoie un tableau Numpy de mêmes dimensions que `A`, contenant les booléens résultats des comparaisons des éléments de `A` avec `seuil`. Puis, le produit `1 * (A < seuil)` permet de renvoyer un tableau dans lequel chacun des booléens précédents a été multiplié par `1`, ce qui a pour effet de transformer les `True` en 1 et les `False` en 0. La fonction répond donc bien à la question posée.

Centrale Informatique optionnelle MP 2019
Corrigé

Ce corrigé est proposé par Olivier Levillain (enseignant-chercheur en école d'ingénieurs) ; il a été relu par Guillaume Batog (professeur en CPGE) et par Benjamin Monmege (enseignant-chercheur à l'université).

Le sujet traite principalement du code binaire de Gray. Ce code décrit un algorithme pour énumérer les éléments de l'intervalle $[\![\, 0\,; 2^n - 1\,]\!]$ de manière à ce que chaque transition ne modifie qu'un seul bit de la représentation binaire. Cette propriété peut s'avérer intéressante lorsque modifier un bit est une opération coûteuse (actionnement d'un interrupteur, courant électrique nécessaire). Le sujet est organisé en deux parties :

- Dans la première partie, il est question d'écrire du code pour énumérer les n-uplets, tout d'abord de manière naïve, puis en utilisant le code de Gray. L'énoncé propose ensuite d'étudier quelques propriétés de ce code, en particulier la relation entre k et le k-ième élément du code de Gray.
- La seconde partie traite des combinaisons de p éléments pris dans un ensemble de cardinal n. Il est proposé de développer du code pour énumérer ces combinaisons de plusieurs manières, et d'établir quelques propriétés liées à la décomposition combinatoire de degré p d'un entier. Les dernières questions de la seconde partie consistent à étudier le problème du sac à dos en réutilisant des résultats sur le code de Gray et sur les combinaisons de p éléments parmi n.

L'énoncé comporte 29 questions de difficultés très variées. Certaines questions sur les combinaisons sont assez peu guidées et nécessitent de l'intuition pour les résoudre. Les questions de programmation sont assez nombreuses et assez abordables. Globalement, ce sujet couvre plusieurs thèmes classiques pour ce type d'épreuve. Il s'agit donc d'un bon entraînement pour les concours, une fois les quelques coquilles de l'énoncé corrigées (voir les indications).

Indications

Partie I

1 L'énoncé indique qu'il faut modifier la liste reçue en argument, mais il ne faut pas prendre cette consigne littéralement, puisque les listes sont immuables en Caml. Plutôt écrire une fonction de type `int list -> int list * bool`.

6 La question n'est pas très claire, mais une réponse simple vient naturellement si on remarque que `monte` et `descend` sont essentiellement la même fonction.

8 Cette question peut se résoudre par récurrence. Il faudra utiliser la relation entre la représentation binaire de $2^n - 1 - r$ et celle de r.

10 On peut montrer que, pour tout $n \geqslant 0$, g est une bijection de l'intervalle $[\![\, 0\,;2^n{-}1\,]\!]$ dans lui-même. Pour cela, pour un y donné, on pourra exhiber un antécédent de y par g.

Partie II

16 La recherche de la combinaison suivante peut se faire via deux mécanismes : soit on incrémente la dernière composante du tableau, soit on est obligé d'incrémenter une autre composante. Pour ce deuxième cas, contrairement à ce que dit l'énoncé, il faut en réalité s'intéresser au plus *grand* indice j tel que $c_{j+1} > c_j + 1$.

18 Il semble plus aisé pour résoudre cette question de calculer le nombre de combinaisons affichées *après* la combinaison $c_0 \cdots c_{p-1}$. En effet, ce nombre peut s'exprimer comme une somme de combinaisons, ce qui sera utilisé dans la question 19.

19 On utilisera la somme trouvée à la question 18.

20 Il faut commencer par prouver le lemme proposé à l'aide de propriétés classiques sur les coefficients binomiaux.

21 On suppose qu'il existe deux décompositions et on s'intéresse au plus grand indice k tel que n_k diffère de n'_k. On cherche ensuite à établir une inégalité en utilisant la question précédente, ce qui devrait mener à une contradiction.

23 Pour évaluer la complexité de l'algorithme proposé, on peut considérer que la masse cumulée et la valeur cumulée sont systématiquement calculées, même si une optimisation évidente est possible. On obtient alors la complexité dans le pire cas.

25 Ajouter un paramètre p aux fonctions, et assurer la contrainte suivante : les listes produites doivent contenir exactement p éléments à 1.

27 Cette question se traite par récurrence sur n. Penser à utiliser les données de l'énoncé sur n et p pour bien démarrer la récurrence.

28 Contrairement à ce que dit l'énoncé, la propriété à admettre est qu'il existe un unique entier j tel que $g^{-1}(a') - g^{-1}(a) \equiv 2^j \bmod 2^n$ (on inverse a et a'). Avec cette propriété, il suffit d'essayer tous les a' possibles à partir de a. Pour cela, on utilisera g, g^{-1}, l'ajout de 2^j et on testera que le candidat contient bien p bits à 1.

I. Code binaire de Gray

1 La fonction demandée nécessite dans un premier temps de parcourir la liste jusqu'à son dernier élément. Si cet élément vaut 0, on met l'élément à 1 et on reconstruit le reste de la liste à l'identique. Si l'élément final vaut 1, on le met à 0 et on propage la retenue, c'est-à-dire qu'on va ajouter 1 à l'élément immédiatement au-dessus.

La propagation de la retenue se fait en renvoyant `false`, alors que `true` signifie qu'il n'y a plus rien à faire. Cette propagation s'arrête lorsqu'on rencontre un 0. Si on ne rencontre aucun 0, l'argument passé était le dernier n-uplet dans l'ordre lexicographique, et la fonction renvoie `false`, comme demandé.

La fonction `suivant` est du type `int list -> int list * bool` et peut être implémentée de la manière suivante :

```
let rec suivant t = match t with
  | [] -> [], false   (* cas impossible puisque n > 0 *)
  | [0] -> [1], true
  | [1] -> [0], false
  | b::bs -> let nouv_bs, retenue = suivant bs in
             match b, retenue with
             | _, true -> b::nouv_bs, true
             | 0, false -> 1::nouv_bs, true
             | 1, false -> 0::nouv_bs, false;;
```

> L'énoncé demande d'implémenter une fonction *transformant* un n-uplet, mais indique que ces n-uplets doivent être implémentés à l'aide de listes, qui sont par nature immuables. Au sens strict, la fonction proposée ne transforme pas la liste passée en argument, mais en reconstruit une nouvelle. Une solution alternative aurait été d'utiliser des références pour stocker les valeurs (le type d'un n-uplet aurait alors été `int ref list`), mais cela n'est pas cohérent avec l'énoncé de la question suivante qui utilise le type `int list`.

2 Comme le suggère l'énoncé, il est pratique d'avoir une fonction `affiche_nuplet` pour afficher les éléments d'un n-uplet. On peut l'écrire de façon récursive sur la liste en ajoutant un saut de ligne à la fin :

```
let rec affiche_nuplet l = match l with
  | [] -> print_newline ()
  | x::xs -> print_int x; affiche_nuplet xs;;
```

Ensuite, la fonction `affiche_nuplets`, de type `int -> unit`, prend en argument la taille des n-uplets considérés. Elle construit le n-uplet uniquement composé de 0. Puis, la fonction parcourt l'ensemble des n-uplets à l'aide de la fonction `suivant` de la question précédente, en affichant chaque valeur. La fonction s'arrête lorsque `suivant` signale que le dernier élément a été atteint.

```
let affiche_nuplets n =
  let rec zero k =
    if k > 0 then 0::(zero (k-1)) else []
  in
  let rec parcours x =
    affiche_nuplet x;
    let s, valide = suivant x in
    if valide then parcours s
  in parcours (zero n);;
```

3 Écrivons la fonction `ajout` de manière récursive. Elle est de type `int -> int list list -> int list list`.

```
let rec ajout a l = match l with
  | [] -> []
  | x::xs -> (a::x)::(ajout a xs);;
```

> En réalité, la fonction ci-dessus a un type plus général que celui annoncé, puisque rien n'impose que les éléments des listes considérées soient des entiers. Le type inféré par Caml sera `'a -> 'a list list -> 'a list list`.

4 Ces deux fonctions s'écrivent naturellement de manière récursive, à partir de la définition du code de Gray. Elles ont pour type `int -> int list list`.

```
let rec monte n =
  if n <= 0
  then [[]]
  else (ajout 0 (monte (n-1))) @ (ajout 1 (descend (n-1)))
and descend n =
  if n <= 0
  then [[]]
  else (ajout 1 (monte (n-1))) @ (ajout 0 (descend (n-1)));;
```

5 Chaque appel à `monte` ou à `descend` avec un argument `n+1` strictement positif mène à un appel à chaque fonction avec l'argument `n`. En notant C_n la complexité de `monte` (ou de `descend`, par symétrie) en nombre d'appels pour une taille n, on obtient donc :

$$\mathrm{C}_{n+1} = 2 + 2\mathrm{C}_n$$

En considérant la suite $u_n = 2 + \mathrm{C}_n$, on a, pour tout $n \geqslant 0$,

$$\begin{aligned} u_{n+1} &= 2 + \mathrm{C}_{n+1} \\ &= 4 + 2\mathrm{C}_n \\ &= 2(2 + 2\mathrm{C}_n) \\ u_{n+1} &= 2u_n \end{aligned}$$

Sachant que C_0 vaut 0, (u_n) est une suite géométrique de raison 2 et de valeur initiale $u_0 = 2$. Ainsi, pour tout n,

$$u_n = 2^n$$

d'où

$$\mathrm{C}_n = 2^n - 2$$

$$\boxed{\text{La complexité des fonctions décrites est donc exponentielle en le nombre d'appels récursifs.}}$$

6 On peut remarquer que `descend` et `monte` calculent essentiellement la même chose, et qu'il suffit d'inverser la liste renvoyée par une de ces fonctions pour trouver le résultat de l'autre fonction. Il est donc inutile de définir les deux fonctions. Le code suivant conserve `monte` et remplace l'appel à `descend (n-1)` par un appel à `List.rev` sur le résultat de `monte (n-1)`.

Mines Maths 1 MP 2019 — Corrigé

Ce corrigé est proposé par Quentin Guilmant (ENS Lyon) ; il a été relu par Philippe Bouafia (professeur agrégé en école d'ingénieurs) et Florian Metzger (docteur en mathématiques).

Ce sujet d'analyse a pour objectif de déterminer les équivalents en $+\infty$ d'une certaine famille de fonctions développables en série entière avec un rayon de convergence infini. L'une d'entre elles peut être reliée à une solution particulière de l'équation d'Airy, c'est-à-dire l'équation différentielle $x'' = tx$.

- La première partie étudie la famille des fonctions $\mathrm{S}_{r,p}$ définies comme des sommes de séries entières par

$$\forall x \in \mathbb{R} \qquad \mathrm{S}_{r,p}(x) = \sum_{n=0}^{+\infty} \left(\frac{(pn)^r}{(pn)!}\right) t^{pn}$$

  Il s'agit d'un travail préliminaire pour trouver un équivalent en l'infini à toutes ces fonctions. Pour cela, on montre que, si $r > 0$, déterminer un équivalent uniquement pour les fonctions définies avec $p = 1$ suffit à en trouver un pour toutes les autres.

- La deuxième partie est composée de deux sous-parties. La première s'intéresse aux espérances de variables aléatoires bien choisies afin de conclure à un équivalent pour les $(\mathrm{S}_{r,1})_{r>0}$. Cela permet donc d'en obtenir un pour tous les $(\mathrm{S}_{r,p})_{p\in\mathbb{N}^*,r>0}$. La deuxième s'attelle à ramener tous les autres cas à celui-ci par l'utilisation du lemme de comparaison asymptotique des séries entières, qui est admis. Ce lemme permet de dire que les sommes de deux séries entières de rayon de convergence $+\infty$ sont équivalentes au bord de leur domaine de définition, c'est-à-dire en $+\infty$, si leurs termes généraux sont équivalents et s'ils ne changent pas de signe.

- Enfin, la troisième et dernière partie propose de trouver une solution sous forme de série entière à l'équation d'Airy. On remarquera qu'une telle solution ressemble à une des fonctions étudiées jusque-là, $\mathrm{S}_{-1/6,2}$. On peut alors conclure à un équivalent pour la solution de l'équation d'Airy.

Dans cette épreuve, la maîtrise des cours sur les équivalents, la formule de Stirling, les séries entières, les variables aléatoires, l'espérance et les équations différentielles linéaires est essentielle. Tout l'enjeu est en effet de produire des calculs techniques et des estimations précises. Plus précisément, le sujet traite principalement de séries entières mais propose, afin de déterminer les équivalents, de passer par l'étude de variables aléatoires bien choisies et de leurs espérances. Le sujet tente clairement d'évaluer les candidats sur la maîtrise des équivalents et des méthodes usuelles de décomposition de variables aléatoires en plusieurs autres. Même si la dernière partie traite des équations différentielles, peu de connaissances sont demandées sur celles-ci et le sujet se concentre vraiment sur les deux thèmes mentionnés ci-dessus.

Indications

Partie A

1 Penser à la règle de d'Alembert.

2 Étudier le quotient $u_n(x)/u_{n-1}(x)$.

3 Penser à des développements limités.

4 Ici la somme est composée $n+1$ termes, en particulier un nombre fini. Il suffit, pour conclure, d'écrire la définition de suites équivalentes.

5 Utiliser la question 4 pour tout entier naturel n non nul, puis utiliser la définition de la partie entière pour conclure.

6 On pourra exprimer $(z-1)\mathrm{S}_{r,1}(zx)$ sous la forme de la somme d'une série entière en z et remarquer que la série $\sum (u_{n-1}(x) - u_n(x))$ converge vers 0. Pour conclure, ajouter ou soustraire judicieusement cette dernière.

7 Se rappeler la valeur d'une somme de racines de l'unité.

Partie B

8 Utiliser l'inégalité de Bienaymé-Tchebychev.

10 Exprimer simplement l'espérance de la variable aléatoire positive $\mathrm{Y}_{\mathrm{N},x}$ pour prouver que la série converge et l'exprimer à l'aide d'une probabilité sur la variable X_x.

11 Penser à une certaine famille de polynômes de degrés échelonnés suggérée par l'expression de $\mathrm{Y}_{\mathrm{N},x}$.

12 Exprimer ${\mathrm{Z}_x}^r$ à l'aide de trois autres variables aléatoires dont A_x et B_x.

13 Écrire la définition d'équivalent.

Partie C

14 Faire un développement asymptotique de $v_n - v_{n-1}$ et le comparer au terme général d'une série convergente de référence. En déduire que $(v_n)_{n\in\mathbb{N}}$ converge puis que $(\mathrm{e}^{v_n})_{n\in\mathbb{N}^*}$ converge elle aussi.

16 Procéder par identification pour obtenir des formules de récurrence liant les quantités a_n et a_{n+3}. Conclure en résolvant la récurrence.

17 Penser à la formule de Stirling.

18 Remarquer que, si $f(u) \underset{u\to+\infty}{\sim} g(u)$, alors on a aussi $f(u^3) \underset{u\to+\infty}{\sim} g(u^3)$.

1 Pour justifier que la série entière $\sum_{n\geqslant 1} ((pn)^r/(pn)!)\, z^n$ a pour rayon de convergence $+\infty$, on peut utiliser la règle de d'Alembert. Notons pour n un entier naturel non nul :

$$a_n = \frac{(pn)^r}{(pn)!}$$

Par suite,

$$\begin{aligned}\left|\frac{a_{n+1}}{a_n}\right| &= \left|\frac{n+1}{n}\right|^r \left|\frac{1}{(pn+1)\cdots(p(n+1))}\right| \\ &\leqslant \left(\frac{n+1}{n}\right)^r \frac{1}{pn+1} \qquad \text{car } p>0\end{aligned}$$

Finalement,

$$\lim_{n\to+\infty} \left|\frac{a_{n+1}}{a_n}\right| = 0$$

La règle de d'Alembert indique alors que le rayon de convergence de cette série entière est $+\infty$. Notons f la somme de la série. La quantité $f(z)$ est donc définie pour tout nombre complexe z. En particulier, pour tout complexe p,

$$\forall z \in \mathbb{C} \qquad f(z^p) = \sum_{n=1}^{+\infty} \frac{(pn)^r}{(pn)!} z^{pn} = \mathrm{S}_{r,p}(z)$$

Ainsi

$$\boxed{\text{La série } \sum_{n\geqslant 1} \frac{(pn)^r}{(pn)!} z^{pn} \text{ a pour rayon de convergence } +\infty.}$$

A. Équivalence entre $(\mathbf{H}_{r,p})$ et $(\mathbf{H}_{r,1})$ lorsque $r > 0$

2 Soient $x>0$ et $r>0$ des réels. La fonction φ_x définie sur $[\,1\,;+\infty\,[$ par

$$\forall t \in [\,1\,;+\infty\,[\qquad \varphi_x(t) = t^{1-r}(t-1)^r - x$$

est continue sur $[\,1\,;+\infty\,[$ et dérivable sur $]\,1\,;+\infty\,[$.

> Elle est dérivable sur $[\,1\,;+\infty\,[$ dès lors que $r \geqslant 1$.

Étudions son signe, ses zéros et ses variations. Avec la formule du cours, on calcule

$$\forall t \in \,]\,1\,;+\infty\,[\qquad \varphi_x{}'(t) = (1-r)\left(\frac{t-1}{t}\right)^r + r\left(\frac{t-1}{t}\right)^{r-1}$$

En particulier,

$$\forall t \in \,]\,1\,;+\infty\,[\qquad \varphi_x{}'(t) > 0$$

ce qui signifie que la fonction $\varphi_x(t)$ est strictement croissante sur $]\,1\,;+\infty\,[$, et par continuité en 1, elle est strictement croissante sur $[\,1\,;+\infty\,[$. De plus,

$$\varphi_x(1) = -x < 0$$

Enfin, comme x est fixé et $t-1 \sim t$ lorsque t tend vers $+\infty$, on a

$$\varphi_x(t) \underset{t\to+\infty}{\sim} t^{1-r}(t-1)^r \underset{t\to+\infty}{\sim} t^{1-r}t^r \underset{t\to+\infty}{\sim} t$$

Il s'ensuit que

$$\lim_{t\to+\infty} \varphi_x(t) = +\infty$$

Le théorème des valeurs intermédiaires assure donc l'existence d'un réel $t_x \in \,]\,1\,;+\infty\,[$ tel que $\varphi_x(t_x) = 0$ et permet donc de construire le tableau de variations et de signes suivant :

	1	t_x	$+\infty$
${\varphi_x}'$		$+$	
φ_x	$-x$	$\nearrow$ 0 $\nearrow$	$+\infty$
φ_x	$-$	0	$+$

Par croissance stricte de φ_x, on conclut

$$\boxed{\text{Il existe un unique } t_x \in \,]\,1\,;+\infty\,[\text{ tel que } \varphi_x(t_x) = 0.}$$

Remarquons que, pour $n \geqslant 1$ un entier, $u_n(x) \neq 0$. Ainsi, pour $n \geqslant 2$, on a

$$\begin{aligned}
\frac{u_n(x)}{u_{n-1}(x)} &= \left(\frac{n}{n-1}\right)^r \frac{1}{n} x \\
&= \frac{x}{n^{1-r}(n-1)^r} \\
\frac{u_n(x)}{u_{n-1}(x)} &= \frac{x}{\varphi_x(n) + x}
\end{aligned}$$

En utilisant l'étude précédente du signe de φ_x, il s'ensuit l'équivalence

$$\forall n \geqslant 1 \qquad u_n \geqslant u_{n-1} \Longleftrightarrow \varphi_x(n) \leqslant 0 \Longleftrightarrow n \leqslant t_x \Longleftrightarrow n \leqslant \lfloor t_x \rfloor$$

La dernière équivalence provenant du fait que n est un entier. Ainsi

$$\boxed{\text{La suite } (u_n(x))_{0 \leqslant n \leqslant \lfloor t_x \rfloor} \text{ est croissante et la suite } (u_n(x))_{n \geqslant \lfloor t_x \rfloor} \text{ est décroissante.}}$$

3 Soit $\alpha \in \mathbb{R}$. Remarquons que, pour $x \geqslant 1 - \alpha$, la quantité $\varphi_x(x + \alpha)$ a un sens. On peut donc bien en étudier la limite en $+\infty$. Déterminons-la à l'aide d'un développement limité. Plus précisément, nous utilisons

$$(1 + u)^r = 1 + ru + \underset{u \to 0}{\mathrm{o}}\,(u)$$

Ainsi,

$$\begin{aligned}
\varphi_x(x + \alpha) &= (x + \alpha)^{1-r}\,(x + \alpha - 1)^r - x \\
&= (x + \alpha)\left(1 - \frac{1}{x + \alpha}\right)^r - x \\
&= (x + \alpha)\left(1 - \frac{r}{x + \alpha} + \underset{x \to +\infty}{\mathrm{o}}\left(\frac{1}{x + \alpha}\right)\right) - x \\
&= x + \alpha - r + \underset{x \to +\infty}{\mathrm{o}}(1) - x \\
\varphi_x(x + \alpha) &= \alpha - r + \underset{x \to +\infty}{\mathrm{o}}(1)
\end{aligned}$$

Par conséquent,

$$\boxed{\lim_{x \to +\infty} \varphi_x(x + \alpha) = \alpha - r}$$

Mines Maths 2 MP 2019 — Corrigé

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

L'objet de ce problème est la détermination d'une suite de majorants du rayon spectral ρ_n de la matrice de Hilbert $(1/(j+k+1))_{0\leqslant j,k\leqslant n-1}$ et l'étude du comportement asymptotique de cette suite. Il s'agit d'un résultat publié en 2005 par Peter Otte. Il établit une relation entre la matrice de Hilbert et un opérateur intégral à noyau positif K_n auquel on applique un principe de min-max pour obtenir l'égalité

$$\rho_n = \inf_{\varphi\in\mathscr{A}} \sup_{x\in\,]\,0\,;\,1\,[} \frac{1}{\varphi(x)}\int_0^1 \mathrm{K}_n(tx)\varphi(t)\,\mathrm{d}t$$

où $\mathscr{A}$ est un ensemble de fonctions.

- La première partie est consacrée à l'étude, à l'aide d'outils d'algèbre linéaire et euclidienne, du sous-espace propre associé à ρ_n. On montre qu'il est de dimension 1 et engendré par un vecteur à coordonnées strictement positives.
- La seconde établit la majoration $\rho_n \leqslant \pi$ à l'aide de relations entre les coefficients de la matrice de Hilbert et l'intégration de fonctions polynomiales.
- La troisième fait le lien entre la matrice de Hilbert et l'opérateur intégral puis établit l'expression de ρ_n comme borne inférieure sur l'ensemble $\mathscr{A}$.
- La dernière partie a pour objectif de calculer cette expression pour une sous-famille particulière de fonctions à l'aide d'intégrales à paramètre et d'en déduire une suite de majorants et son comportement asymptotique.

Le thème du sujet est intéressant et permet d'obtenir un résultat non trivial en recourant à de nombreuses méthodes d'analyse et d'algèbre.

L'épreuve possède cependant plusieurs défauts très regrettables pour un sujet de concours. Le plus important étant celui d'utiliser à la première question une notion hors programme (matrice définie positive) sans en donner la définition. En outre, elle n'est pas assez progressive. La question 2 comporte une implication difficile pour les candidats qui n'ont pas rencontré l'étude de la matrice de Hilbert ou du théorème de Perron-Frobenius. Enfin la dernière partie est une suite de calculs complexes. L'énoncé comporte dans cette partie des erreurs qui n'ont certainement pas permis aux candidats d'aborder la dernière question, elle aussi d'un niveau très élevé.

C'est un sujet dont les trois premières parties méritent d'être travaillées, pour les méthodes employées et les résultats autour de la matrice de Hilbert, mais il n'intéressera comme outil de révision et d'entraînement que les tout meilleurs candidats.

Indications

Partie A

2 Pour l'implication réciproque, appliquer le théorème spectral afin de décomposer une colonne X dans une base orthonormée puis majorer la quantité ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X}$ par $\rho_n\|\mathrm{X}\|$. Se servir alors de l'inégalité de Cauchy-Schwarz pour discuter le cas d'égalité.

3 Utiliser la question 2 pour déduire que $|\mathrm{X}_0| \in \mathscr{V}$.

4 Exploiter le fait que les coefficients de H_n sont strictement positifs.

5 Construire une combinaison linéaire de deux vecteurs propres de $\mathscr{V}$ dont un coefficient est nul et en déduire qu'ils sont proportionnels.

Partie B

6 Commencer par calculer $\displaystyle\int_0^{\pi} \mathrm{P}\left(\mathrm{e}^{\,\mathrm{i}\,\theta}\right) \mathrm{d}\theta$ pour des polynômes élémentaires X^k.

7 Développer $|\widetilde{\mathrm{X}}(\mathrm{e}^{\,\mathrm{i}\,\theta})|^2$ pour faire apparaître une somme de termes constants et une somme de termes périodiques dont l'intégrale sur $[\,0\,;\pi\,]$ est nulle.

8 Construire à partir des coordonnées d'un vecteur propre X_0 de H_n une colonne X_0' de taille $n+1$ telle que ${}^t\mathrm{X}_0'\,\mathrm{H}_{n+1}\mathrm{X}_0' = \rho_n\|\mathrm{X}_0'\|^2$.

Partie C

10 Identifier les coefficients de $(\mathrm{H}_n)_{j,k}$ à l'intégrale sur $[\,0\,;1\,]$ de la fonction $t \longmapsto t^{j+k}$.

11 Exprimer la fonction $\rho_n\widetilde{\mathrm{X}_0}/\varphi$ à l'aide de l'opérateur intégrale T_n puis la majorer en faisant apparaître

$$\sup_{t\in]\,0\,;1\,[} \frac{\widetilde{\mathrm{X}_0}(t)}{\varphi(t)}$$

dont on prouvera l'existence. Utiliser la définition de la borne supérieure.

Partie D

12 Appliquer le théorème de dérivation des intégrales à paramètre.

13 Intégrer $\mathrm{J}_n(x)$ par parties en utilisant la primitive $t \longmapsto t-1$.

16 Exploiter l'équation différentielle dont φ est solution et utiliser le formulaire fourni par l'énoncé pour exprimer le coefficient c_n à l'aide de la fonction Γ.

18 Dans le calcul de l'intégrale en question 11, utiliser le fait que $\mathrm{K}_n(tx)$ est la somme des premiers termes d'une suite géométrique de raison tx.

19 Choisir la valeur $\alpha = 1/2$ et utiliser le changement de variable $u = \sqrt{t}$.

20 Utiliser la formule de Stirling, puis chercher un développement limité à l'ordre 1 et au voisinage de 0 de la fonction $x \longmapsto \mathrm{Arcsin}\,(1-x^2)$.

A. Une propriété de Perron-Frobenius

1 La notion de matrice définie positive est explicitement hors programme et aurait dû être définie dans l'énoncé. Si on fixe une base $(e_1, e_2, \ldots, e_n)$ d'un espace réel de dimension n, les matrices symétriques réelles définies positives de taille n sont exactement les matrices dans cette base des produits scalaires, la bijection entre la matrice S et le produit scalaire $(\cdot \mid \cdot)$ étant réalisée par la relation $\mathrm{S} = ((e_i \mid e_j))_{1 \leqslant i,j \leqslant n}$.

Notons X la matrice des coordonnées dans la base $(e_1, e_2, \ldots, e_n)$ d'un vecteur de l'espace euclidien muni de ce produit scalaire, le réel ${}^t\mathrm{X}\,\mathrm{S}\mathrm{X}$ s'identifie alors au carré de la norme euclidienne associée.

$$\begin{aligned} {}^t\mathrm{X}\,\mathrm{S}\mathrm{X} &= \sum_{i=1}^{n} \mathrm{X}_i \left(\sum_{j=1}^{n} \mathrm{S}_{i,j} \mathrm{X}_j \right) \\ &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \mathrm{X}_i \mathrm{X}_j (e_i \mid e_j) \right) \qquad (\text{comme } \mathrm{S}_{i,j} = (e_i|e_j)) \\ {}^t\mathrm{X}\,\mathrm{S}\mathrm{X} &= \left(\sum_{i=1}^{n} \mathrm{X}_i e_i \,\middle|\, \sum_{j=1}^{n} \mathrm{X}_j e_j \right) \end{aligned}$$

La matrice H_n est réelle et symétrique puisque pour tous j, k de $[\![\, 0\,; n-1 \,]\!]$

$$h_{j,k}^{(n)} = \frac{1}{j+k+1} = h_{k,j}^{(n)}$$

Montrons son caractère définie positive, c'est-à-dire que pour toute matrice colonne $\mathrm{X} \in \mathscr{M}_{n,1}(\mathbb{R})$ non nulle, ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} > 0$. Soit $\mathrm{X} \in \mathscr{M}_{n,1}(\mathbb{R})$, développons

$$\begin{aligned} {}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} &= \sum_{j=0}^{n-1} x_j \,(\mathrm{H}_n\mathrm{X})_{j,1} \\ &= \sum_{j=0}^{n-1} x_j \sum_{k=0}^{n-1} \frac{1}{j+k+1} x_k \\ &= \sum_{j=0}^{n-1} x_j \sum_{k=0}^{n-1} \left(\int_0^1 t^{j+k}\, \mathrm{d}t \right) x_k \\ &= \int_0^1 \sum_{j=0}^{n-1} x_j t^j \sum_{k=0}^{n-1} x_k t^k \,\mathrm{d}t \qquad (\text{linéarité de l'intégrale}) \\ &= \int_0^1 \widetilde{\mathrm{X}}(t) \widetilde{\mathrm{X}}(t)\, \mathrm{d}t \\ {}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} &= \int_0^1 \widetilde{\mathrm{X}}(t)^2\, \mathrm{d}t \end{aligned}$$

Ceci fournit ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} \geqslant 0$.

La fonction polynomiale $\widetilde{\mathrm{X}}^2$ est continue et positive donc si

$$\int_0^1 \left(\widetilde{\mathrm{X}}(t) \right)^2 \mathrm{d}t = 0$$

alors $\widetilde{\mathrm{X}}^2$ est nulle sur $[\,0\,;1\,]$. Son polynôme associé, possédant une infinité de racines, est nul d'où $\mathrm{X} = 0$. Réciproquement si $\mathrm{X} = 0$, on a bien ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} = 0$. Ainsi,

$$\boxed{\text{La matrice } \mathrm{H}_n \text{ est symétrique réelle et définie positive.}}$$

2 Si $\mathrm{X} \in \mathscr{V}$ alors $\mathrm{H}_n\mathrm{X} = \rho_n\mathrm{X}$ d'où

$$ {}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} = \rho_n\,{}^t\mathrm{X}\,\mathrm{X} = \rho_n\|\mathrm{X}\|^2 $$

Comme H_n est symétrique et réelle, elle possède des valeurs propres réelles et, d'après le théorème spectral, il existe une base orthonormée de vecteurs propres de H_n. Notons $(\mathrm{X}_i)_{1\leqslant i\leqslant n}$ une telle base, associée aux valeurs propres $(\lambda_i)_{1\leqslant i\leqslant n}$, non nécessairement distinctes. Ces valeurs propres sont strictement positives. En effet, si on applique le calcul de la question 1 à chaque $\mathrm{X}_i \neq 0$ on trouve

$$ 0 < {}^t\mathrm{X}_i\,\mathrm{H}_n\mathrm{X}_i = {}^t\mathrm{X}_i(\lambda_i\mathrm{X}_i) = \lambda_i\|\mathrm{X}_i\|^2 $$

Soit $\mathrm{X} \in \mathscr{M}_{n,1}(\mathbb{R})$ un vecteur quelconque décomposé dans la base orthonormée $(\mathrm{X}_i)_{1\leqslant i\leqslant n}$ sous la forme $\mathrm{X} = \sum\limits_{i=1}^{n} a_i\mathrm{X}_i$. D'après le théorème de Pythagore,

$$ \|\mathrm{X}\|^2 = \sum_{i=1}^{n} {a_i}^2 $$

d'où

$$ \begin{aligned} \|\mathrm{H}_n\mathrm{X}\|^2 &= \left\|\mathrm{H}_n\left(\sum_{i=1}^{n} a_i\mathrm{X}_i\right)\right\|^2 \\ &= \left\|\sum_{i=1}^{n} a_i\lambda_i\mathrm{X}_i\right\|^2 \\ &= \sum_{i=1}^{n} {a_i}^2{\lambda_i}^2 && \text{(théorème de Pythagore)} \\ &\leqslant \sum_{i=1}^{n} {a_i}^2{\rho_n}^2 && (\text{car } |\lambda_i| = \lambda_i \leqslant \rho_n) \\ \|\mathrm{H}_n\mathrm{X}\|^2 &\leqslant \|\mathrm{X}\|^2{\rho_n}^2 \end{aligned} $$

Donc pour tout $\mathrm{X} \in \mathscr{M}_{n,1}(\mathbb{R})$, $\quad \|\mathrm{H}_n\mathrm{X}\| \leqslant \rho_n\|\mathrm{X}\|$

Supposons alors ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} = \rho_n\|\mathrm{X}\|^2$. Utilisons l'inégalité de Cauchy-Schwarz et la majoration précédente

$$ \rho_n\|\mathrm{X}\|^2 = {}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} \leqslant \|\mathrm{X}\|\|\mathrm{H}_n\mathrm{X}\| \leqslant \rho_n\|\mathrm{X}\|^2 $$

Tous les termes sont donc égaux et, par le cas d'égalité dans l'inégalité de Cauchy-Schwarz, la famille de vecteurs $(\mathrm{X}, \mathrm{H}_n\mathrm{X})$ est liée. Si $\mathrm{X} = 0$, alors $\mathrm{X} \in \mathscr{V}$. Sinon il existe $\lambda \in \mathbb{R}$ tel que $\mathrm{H}_n\mathrm{X} = \lambda\mathrm{X}$. L'hypothèse ${}^t\mathrm{X}\,\mathrm{H}_n\mathrm{X} = \rho\|\mathrm{X}\|^2$ donne alors $\lambda^2 = {\rho_n}^2$, car $\mathrm{X} \neq 0$ et donc $\lambda = \rho_n$ puisque les valeurs propres de H_n sont positives. Par conséquent $\mathrm{X} \in \mathscr{V}$ également. On a montré

$$ \boxed{\mathrm{X} \in \mathscr{V} \text{ si et seulement si } {}^t\mathrm{X}\mathrm{H}_n\mathrm{X} = \rho_n\|\mathrm{X}\|^2} $$

3 D'après l'inégalité triangulaire, pour tout $t \in [\,0\,;1\,]$,

$$ \left|\widetilde{\mathrm{X}_0}(t)\right| = \left|\sum_{k=0}^{n-1} x_k t^k\right| \leqslant \sum_{k=0}^{n-1} |x_k|\,t^k = \widetilde{|\mathrm{X}_0|}(t) $$

En reprenant l'expression obtenue à la question 1 et par croissance de l'intégrale et de la fonction carré sur $\mathbb{R}_+$, il vient

$$ \begin{aligned} {}^t\mathrm{X}_0\,\mathrm{H}_n\mathrm{X}_0 &= \int_0^1 \widetilde{\mathrm{X}_0}(t)^2\,\mathrm{d}t \\ &\leqslant \int_0^1 \widetilde{|\mathrm{X}_0|}(t)^2\,\mathrm{d}t \end{aligned} $$

d'où

$$ \boxed{{}^t\mathrm{X}_0\,\mathrm{H}_n\mathrm{X}_0 \leqslant {}^t|\mathrm{X}_0|\,\mathrm{H}_n\,|\mathrm{X}_0|} $$

Mines Informatique MP-PC-PSI 2019 — Corrigé

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

Ce sujet aborde plusieurs problèmes numériques liés aux nombres premiers, notamment leur génération et leur comptage. Il est relativement complet, bien posé, et couvre une grande partie du programme.

- La première partie est peu liée au reste et demande l'écriture et la compréhension de fonctions simples. Elle aborde également la représentation des nombres réels en machine.
- La deuxième partie est consacrée à la génération de nombres premiers. Elle débute par la méthode classique du crible d'Ératosthène. L'algorithme est donné, il faut simplement l'implémenter et réfléchir à sa complexité et aux contraintes de mémoire. Une méthode plus efficace pour les grands nombres, qui s'appuie sur le test probabiliste de primalité de Fermat, est ensuite décrite et doit également être implémentée.
- La troisième partie aborde le problème de la répartition des nombres premiers, et plus particulièrement le calcul de la fonction $\pi(n)$, qui renvoie le nombre de nombres premiers dans l'intervalle $[\![1\,; n]\!]$. On peut la calculer directement avec le crible d'Ératosthène vu dans la partie précédente, mais cette approche souffre évidemment des mêmes défauts. On utilise alors l'équivalence des fonctions π et li (logarithme intégral) à l'infini, en approchant $\pi(n)$ par $\mathrm{li}(n)$. Le calcul de cette fonction nécessite l'intégration de l'inverse du logarithme, que l'on réalise ici par la méthode des rectangles. Le calcul est délicat à cause de la singularité en 1. Enfin, on exploite une identité mathématique reliant li à une série que l'on peut calculer avec une plus grande précision numérique.
- La dernière partie, très courte, porte sur les bases de données.

Il n'y a pas lieu de se laisser intimider par le thème, l'énoncé paraît plus mathématique qu'il n'est en réalité !

Indications

Partie I

3 Rappel : / est l'opérateur associé à la division flottante en Python.

4 Attention, x est sobrement défini par le sujet comme un « nombre » ; donc pas forcément entier. Faire attention au cas $0 < x < 1$. On peut utiliser une récurrence pour répondre rigoureusement à la question.

Partie II

8 Veiller aux erreurs « off-by-one » (erreurs de décalage) possibles entre l'entier `i` et la liste `liste_bool`.

12 Ne pas oublier d'importer les modules nécessaires à l'utilisation des fonctions demandées.

13 On a le droit de définir une fonction annexe pour le test de primalité de Fermat, qui est répété quatre fois.

14 Attention à ne pas appeler la fonction `erato_iter` plus que nécessaire.

Partie III

19 Question de cours. Faire en sorte de minimiser les imprécisions numériques, voir question 5.

22 Observer que suffisamment proche de 1, les rectangles à gauche et à droite s'annulent quasiment deux à deux en cascade, sauf un de chaque côté. C'est leur somme qui forme la différence absolue observée.

24 Attention à la confusion entre x et $\ln x$ dont l'utilisation par le sujet peut être troublante.

26 On pourra faire appel aux commandes `COUNT` (qui compte le nombre d'enregistrements) et `AVG` (qui calcule leur moyenne). Pour la deuxième question, la commande `MINUS` (ou `EXCEPT`) s'avère utile.

I. Préliminaires

1 Afin d'utiliser les fonctions d'un module, il faut l'importer. Si l'on n'a besoin que de quelques fonctions du module, et qu'il n'y a pas de risque de confusion avec des fonctions d'autres modules, on peut les importer spécifiquement dans l'espace de noms global du programme :

```
from math import log, sqrt, floor, ceil
print(log(0.5))
```

> Cette réponse est celle que le sujet semble attendre, mais on aurait également pu importer tout le module d'un coup, en faisant attention par la suite à appeler les fonctions dans l'espace de noms du module :
>
> ```
> import math
> print(math.log(0.5))
> ```
>
> On peut également importer toutes les fonctions du module dans l'espace de noms global avec `from math import *`, mais c'est déconseillé parce qu'on risque de masquer des fonctions déjà existantes.

2 On teste la condition au sein de l'instruction `return` et on renvoie directement sa valeur :

```
def sont_proches(x, y):
    atol = 1e-5
    rtol = 1e-8
    return abs(x - y) <= atol + abs(y) * rtol
```

> Attention à ne pas proposer la version suivante :
>
> ```
> def sont_proches(x, y):
>     atol = 1e-5
>     rtol = 1e-8
>     if abs(x - y) <= atol + abs(y) * rtol:
>         return True
>     else:
>         return False
> ```
>
> qui est un « anti-pattern » (un motif à ne pas reproduire). En effet, la condition est inutile puisque l'inégalité renvoie déjà les mêmes booléens, et on n'est même pas à l'abri de se tromper de sens en les recopiant.

3 En déroulant mécaniquement l'exécution du programme, on peut démontrer les égalités suivantes :

$$\begin{aligned}
\texttt{mystere(1001, 10)} &= \texttt{1 + mystere(100.1, 10)}\\
&= \texttt{1 + (1 + mystere(10.01, 10))}\\
&= \texttt{1 + (1 + (1 + mystere(1.001, 10)))}\\
\texttt{mystere(1001, 10)} &= \texttt{1 + (1 + (1 + 0))}
\end{aligned}$$

soit $\boxed{\texttt{mystere(1001, 10)} = 3}$

4 L'exemple précédent montre que chaque appel à la fonction ajoute 1 au résultat tout en divisant le premier paramètre par b. Considérons $x \geqslant 1$. On a donc

$$\text{mystere}(x, b) = 1 + \text{mystere}\left(\frac{x}{b}, b\right)$$

ou encore
$$\text{mystere}(x, b) = k + \text{mystere}\left(\frac{x}{b^k}, b\right)$$

jusqu'à ce que $x/b^k < b$, cas de base où la fonction renvoie 0 et la récursion s'arrête. Le résultat est donc l'entier k tel que

$$\frac{x}{b^k} < b \leqslant \frac{x}{b^{k-1}}$$

où la première inégalité représente la condition d'arrêt et la seconde le test de l'avant-dernier appel. On a donc

$$\frac{x}{b} < b^k \leqslant x$$

L'astuce consiste à prendre le logarithme en base b pour obtenir

$$\log_b x - 1 < k \leqslant \log_b x$$

où l'on reconnaît $k = \lfloor \log_b x \rfloor$.

Il reste à traiter la situation $0 < x < 1$. Dans ce cas, `mystere(x, b)` est nul, ce qui ne correspond plus à la partie entière du logarithme de x. On peut soit regrouper les cas avec un maximum, ce qui affaiblit un peu la formulation mathématique :

$$\boxed{\forall x > 0 \quad \texttt{mystere(x, b)} = \max(0, \lfloor \log_b(x) \rfloor)}$$

soit définir la fonction par morceaux :

$$\boxed{\forall x > 0 \quad \texttt{mystere(x, b)} = \begin{cases} 0 & \text{si } 0 < x < 1 \\ \lfloor \log_b(x) \rfloor & \text{sinon} \end{cases}}$$

Une démonstration par récurrence permettrait une réponse plus rigoureuse. Pour cela, fixons un b arbitraire. On va démontrer par récurrence sur n que l'hypothèse

$$\mathscr{P}(n): \quad \text{« } \forall x \in [\![\, b^n \,;\, b^{n+1} - 1 \,]\!] \quad \texttt{mystere(x, b)} = \lfloor \log_b x \rfloor \text{ »}$$

est vraie pour tout n entier naturel.

- $\underline{\mathscr{P}(0)}$: le cas de base est $1 \leqslant x < b$. Dans ces conditions, on a

  $$\texttt{mystere(x, b)} = 0$$

  La fonction $\log_b$ est strictement monotone croissante, avec $\log_b(1) = 0$ et $\log_b(b) = 1$, donc on a bien $\lfloor \log_b x \rfloor = 0$.

- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$: supposons que l'hypothèse est vraie pour n. Alors, pour un x tel que $b^{n+1} \leqslant x < b^{n+2}$, on a

  $$\begin{aligned} \texttt{mystere(x, b)} &= 1 + \texttt{mystere(x / b, b)} \quad \left(\text{où } b^n \leqslant \frac{x}{b} < b^{n+1}\right) \\ &= 1 + \lfloor \log_b(x/b) \rfloor \qquad (\text{par } \mathscr{P}(\text{n})) \\ &= 1 + \lfloor \log_b(x) - 1 \rfloor \\ &= 1 + \lfloor \log_b(x) \rfloor - 1 \\ \texttt{mystere(x, b)} &= \lfloor \log_b(x) \rfloor \end{aligned}$$

Mines Informatique optionnelle MP 2019 — Corrigé

Ce corrigé est proposé par Guillaume Batog (professeur en CPGE) ; il a été relu par Olivier Levillain (enseignant-chercheur en école d'ingénieurs) et Benjamin Monmege (enseignant-chercheur à l'université).

Le sujet porte sur les automates déterministes avec un alphabet à deux lettres. L'objectif est de construire un automate possédant un minimum d'états et acceptant un langage rationnel donné. La technique employée est élaborée à partir de morphismes d'automates et de parcours en profondeur de graphes.

- La première partie est rapide, il s'agit de déterminer des langages acceptés par des automates sur quelques exemples.
- La deuxième partie construit la partie accessible d'un automate à partir d'un parcours en profondeur de son graphe sous-jacent. Les trois questions de cette partie sont peu guidées, discriminantes et fatidiques en cas de méconnaissance du cours. Il ne faut pas hésiter à soigner cette partie, d'autant plus qu'elle comporte quelques-unes des rares questions de programmation du sujet.
- La troisième partie définit la notion de morphisme entre automates. Après quelques exemples, on établit des propriétés théoriques – composition, morphisme inverse, langage accepté – avant de coder un test d'existence de morphisme. Il faut réfléchir soi-même à l'algorithme, qui demande une bonne dose d'initiative et d'inspiration sur le parcours en profondeur précédent.
- La quatrième partie est longue et répétitive. On construit et manipule les morphismes de façon théorique. Il est possible de traiter les trois questions de programmation indépendamment du reste, notamment la construction d'un automate produit.
- La cinquième partie exploite les résultats précédents pour démontrer l'existence et l'unicité (à un isomorphisme près) d'un automate minimal. Sa construction s'appuie sur une technique de fusions d'états, modélisés par des morphismes. Après quelques manipulations sur un exemple, on passe à l'algorithmique et la programmation dans les deux dernières questions. Elles nécessitent énormément de recul, sachant que l'énoncé ne propose à nouveau aucun algorithme.

Difficile en trois heures de venir à bout de ce sujet, qui a le mérite d'être complet et bien construit (excepté pour la dernière question). Les exemples et l'ordre des questions sont bien choisis pour assimiler les nouvelles notions. Le programmeur chevronné sera en revanche déçu. En effet, les quelques questions de code sont l'aboutissement d'une réflexion à la fois théorique et algorithmique.

Cette épreuve de haut niveau visait à départager les meilleurs. Toutefois, pour ceux moins à l'aise avec la théorie, traiter les questions 1 à 15, puis 18, 19 et 27 en trois heures est tout à fait profitable (et honorable !) pour réviser les automates.

Indications

Partie 2

6 Parcourir récursivement la liste en argument à l'aide d'une fonction auxiliaire récursive dont un paramètre retient le nombre d'éléments rencontrés.

7 Coder les états visités à l'aide d'un tableau de booléens, puis écrire une fonction auxiliaire récursive `parcours` telle que `parcours q acc` réalise le parcours en profondeur à partir du sommet `q` étant donné la liste `acc` des sommets accessibles déjà visités. Dans le cadre des automates du sujet, tout état q ne possède que deux voisins $\delta(q, a)$ et $\delta(q, b)$. Ainsi, au maximum deux appels récursifs sont suffisants à chaque étape.

8 Gérer les numéros d'états est le plus délicat. On renumérote les états du parcours en profondeur grâce à la fonction `numero` de la question 6. Réciproquement, construire un tableau `decode` permettant de trouver le numéro d'état de l'automate initial à partir de celui de la partie accessible.

Partie 3

9 Avoir l'intuition d'un morphisme d'automates comme un processus de fusion d'états et de transitions ayant même origine, destination et étiquette.

11 Raisonner par l'absurde. Les conditions (2) et (4) fixent les images d'un morphisme de $\mathcal{A}_1$ vers $\mathcal{A}_2$ s'il existe. Toutefois, la condition (3) n'est pas satisfaite.

13 Montrer l'égalité des langages des deux automates par double inclusion. Pour tout mot $\omega = \sigma_1\sigma_2\cdots\sigma_\ell$ dans le langage d'un automate $\mathcal{A}$, considérer une exécution de ce mot sur $\mathcal{A}$:

$$q_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\sigma_2} \cdots q_\ell \xrightarrow{\sigma_\ell} q_{\ell+1} \qquad \text{avec} \qquad q_i \in \mathrm{Q}_{\mathcal{A}} \qquad q_1 = i_{\mathcal{A}} \qquad \text{et} \qquad q_{\ell+1} \in \mathrm{F}_{\mathcal{A}}$$

puis en déduire une exécution de ω sur l'autre automate.

14 Justifier que φ est injective. Pour montrer que φ^{-1} est un morphisme, vérifier les quatre conditions de morphisme.

16 Montrer que tout état de l'automate $\mathcal{B}$ possède un antécédent par φ par récurrence sur la distance n de cet état à l'état initial.

17 S'inspirer du parcours en profondeur de la question 7 où le tableau `visites` est remplacé par le tableau `phi` résultat, -1 signifiant que l'état correspondant n'a pas encore été visité. La condition (3) permet de construire les images de φ des successeurs d'un état q dont on connaît $\varphi(q)$. À chaque étape du parcours, tester l'absence d'incohérence dans la construction de φ vis-à-vis des conditions de morphisme (3) et (4).

Partie 4

19 Coder l'état (q_1, q_2) par l'entier $q_1 + n_1 q_2$ avec n_1 le nombre d'états du premier automate. Il suffit de parcourir un à un les états de l'automate produit pour construire sa fonction de transition et ses états finals.

20 Pour toute exécution d'un mot ω sur l'automate produit $\mathcal{A} \times \mathcal{A}'$, construire deux exécutions de ω sur chacun des automates $\mathcal{A}$ et $\mathcal{A}'$.

21 Les morphismes considérés sont les projections $(x, y) \mapsto x$ et $(x, y) \mapsto y$.

22 Les trois propriétés d'une relation d'équivalence sont la réflexivité, la symétrie et la transitivité.

23 Considérons la séquence des $\delta_{\mathcal{B}}(q_j, \sigma)$ pour $0 \leqslant j < k$, où $q_0, q_1, ..., q_k$ désigne une séquence résultant de la condition $p \equiv q$.

25 L'automate $\mathcal{C}$ est « l'image » de l'automate $\mathcal{B}$ par le morphisme $\eta : q \longmapsto [q]$.

26 Suivre les flèches dans le diagramme de début de section 4.2 du sujet.

27 Utiliser un dictionnaire `dico` de type `(int * int) list` contenant des couples de la forme `(cle,obj)` avec `cle` un entier présent dans le tableau initial et `obj` le renommage choisi pour cet entier.

28 Déterminer les composantes connexes du graphe où deux sommets p et q sont voisins si et seulement si $\varphi(p) = \varphi(q)$ ou $\psi(p) = \psi(q)$.

Partie 5

29 Employer les questions 21 et 26.

30 D'après le raisonnement de la question 29, utiliser l'automate de la question 18.

31 Justifier que les deux morphismes de la question 29 sont des isomorphismes.

32 Reprendre la construction de morphisme de la question 31.

34 Ne pas oublier la contrainte imposée par la condition (4) de morphisme.

35 Après fusion, justifier qu'on obtient bien un automate ayant un minimum d'états.

36 Réaliser un parcours en profondeur sur le graphe P, qu'on ne construira pas, en s'inspirant de la question 7 où le tableau `visites` est remplacé par la matrice attendue en sortie. Commencer le parcours à partir de la liste de tous les sommets de $\mathrm{F} \times \overline{\mathrm{F}} \cup \overline{\mathrm{F}} \times \mathrm{F}$. Mettre à jour la matrice `visites` au cours de l'initialisation de cette liste.

37 Une coquille s'est glissée dans l'énoncé de la question 36. On doit considérer plutôt un graphe P dont les arcs vont de $\big(\delta(p,\sigma), \delta(q,\sigma)\big)$ vers (p,q) pour tous $\sigma \in \{a,b\}$ et $p, q \in \mathrm{Q}$. Reprendre alors la construction de la matrice `m` de la question 36, qui encode la relation d'équivalence suivante sur les états de $\mathcal{A}$:

$$\forall\, p,q \in \mathrm{Q} \qquad p \sim q \quad \Longleftrightarrow \quad \texttt{m.(p).(q) = false}$$

L'automate minimal s'obtient en fusionnant les ensembles d'états appartenant à une même classe d'équivalence. On construit d'abord le morphisme φ qui, à tout état, associe le numéro de sa classe d'équivalence, puis l'automate image de $\mathcal{A}$ par φ en s'inspirant de la question 8.

1. Premiers exemples

$\boxed{1}$ L'automate $\mathcal{A}_1$ accepte le langage L_1 des **mots de longueur impaire**.

> En effet, toute exécution de l'automate $\mathcal{A}_1$ est de la forme
>
> $$A \xrightarrow{\sigma_1} B \xrightarrow{\sigma_2} A \xrightarrow{\sigma_3} B \cdots \xrightarrow{\sigma_{2k}} A \xrightarrow{\sigma_{2k+1}} B$$
>
> où les lettres σ_i peuvent valoir indifféremment a ou b.

$\boxed{2}$ L'automate $\mathcal{A}_2$ accepte le langage L_2 des **mots contenant un nombre impair de lettres** b.

> Quel que soit l'état dans lequel on se trouve, l'automate $\mathcal{A}_2$ peut lire un nombre quelconque de lettre a, et ce en restant toujours dans le même état courant. Ainsi, il n'existe aucune contrainte sur le nombre ou la position des lettres a dans le langage L_2. En enlevant de $\mathcal{A}_2$ les transitions étiquetées par a (non contraignantes), on retrouve l'automate $\mathcal{A}_1$ avec la seule lettre b.

$\boxed{3}$ Le langage L_1 est dénoté par l'expression

$$\boxed{(a+b)\cdot\left[(a+b)^2\right]^*}$$

$\boxed{4}$ Le langage L_2 est dénoté par l'expression

$$\boxed{a^*\cdot b\cdot\left[(a^*\cdot b)^2\right]^*\cdot a^*}$$

> Cette expression rationnelle est composée de trois parties traduisant les étapes d'un cheminement dans l'automate :
>
> 1. On arrive une première fois dans l'état D : $a^*\cdot b$.
> 2. On revient possiblement sur l'état D via un détour par l'état C : $(a^*\cdot b)^2$. On applique l'étoile de Kleene à cette expression pour répéter le processus (éventuellement zéro fois).
> 3. On reste à l'état D, auquel cas on ne peut que boucler sur l'état D final : a^*.

$\boxed{5}$ Le code suivant est de type `automate` et représente l'automate $\mathcal{A}_2$.

```
let n = 2
and delta = [| 0,1 ; 1,0 |]
and f = [| false; true |] in
n,delta,f;;
```

> Les parenthèses d'un n-uplet sont facultatives si aucune ambiguïté d'écriture ne se présente. Rappelons que l'état initial C est codé par 0 d'après l'énoncé.

X/ENS Maths A MP 2019 — Corrigé

Ce corrigé est proposé par Florian Metzger (docteur en mathématiques) ; il a été relu par Loïc Devilliers (professeur en CPGE) et Céline Chevalier (enseignant-chercheur à l'université).

Ce sujet traite de plusieurs résultats de théorie algébrique des nombres, et plus spécifiquement des nombres complexes *algébriques* (annulant un polynôme non nul de $\mathbb{Q}[X]$) et des *entiers algébriques* (annulant un polynôme unitaire de $\mathbb{Z}[X]$). La classe des nombres de Salem est en particulier abordée dans les deux dernières parties. L'étude des propriétés algébriques des nombres de Salem permet en particulier de comprendre des *dynamiques* (itérées de certaines fonctions).

- La première partie est l'occasion d'obtenir plusieurs résultats généraux sur les nombres et les entiers algébriques ; en particulier, on y établit que le polynôme minimal d'un nombre algébrique est irréductible sur $\mathbb{Q}$ et que celui d'un entier algébrique est dans $\mathbb{Z}[X]$. Une étude spécifique est menée sur des nombres de *degré* 2, le degré étant défini comme celui du polynôme minimal.
- Dans la deuxième partie, l'étude est centrée sur le n-ième polynôme cyclotomique Φ_n, dont les racines complexes sont les racines n-ième de l'unité *primitives*, c'est-à-dire d'ordre n dans $\mathbb{U}_n$. Des formules de récurrence ainsi que l'explicitation de Φ_n pour certaines valeurs de n y sont établies, $\Phi_n(0)$ et $\Phi_n(1)$ sont calculés, avant de prouver que $\Phi_n \in \mathbb{Z}[X]$. Après avoir établi que si les racines complexes d'un polynôme unitaire de $\mathbb{Z}[X]$ irréductible sur $\mathbb{Q}$ sont de module 1, ce sont en fait des racines de l'unité, on montre à la fin de cette partie l'irréductibilité de Φ_n qui se trouve être le polynôme minimal de toute racine n-ième primitive.
- Dans la troisième partie sont abordés les polynômes réciproques, dont les coefficients sont symétriques par rapport à la moitié du degré, afin d'étudier les *conjugués* d'un nombre algébrique α, c'est-à-dire les racines complexes de son polynôme minimal distinctes de α. On étudie finalement la classe $\mathcal{S}$ des entiers algébriques dits de Salem qui sont les entiers algébriques $\alpha > 1$ dont tous les conjugués autres que $1/\alpha$ sont de module 1 : ils sont nécessairement de degré pair supérieur à 4.
- La quatrième et dernière partie est dédiée à l'étude de la sous-famille $\mathcal{T}$ des entiers algébriques $\alpha \in \mathcal{S}$ de degré minimal, à savoir 4. On montre que $\mathcal{T}$ est de cardinal infini en exhibant une suite à termes construits comme racines d'une certaine famille de polynômes de $\mathbb{Z}[X]$ indexés par n. Enfin, la difficile dernière question permet d'établir que $\mathcal{T}$ admet un minimum et de le calculer. Ce résultat a un intérêt particulier car le résultat établi pour le degré 4 n'a pas encore été généralisé pour la classe $\mathcal{S}$ toute entière.

Si la première partie comporte beaucoup de questions classiques, les choses se corsent dans la deuxième partie où certaines questions demandent technique et réflexion poussées avec des raisonnements arithmétiques parfois fins et non standards. Les thématiques du cours abordées sont principalement les séries entières, les racines de l'unité et l'arithmétique dans $\mathbb{Z}$ ou $\mathbb{Q}[X]$. La dernière partie requiert plusieurs manipulations analytiques assez élémentaires, avant une dernière question de synthèse très pointue.

Indications

Partie 1

1 Écrire I_α comme le noyau d'un morphisme d'anneaux.

2 Utiliser un polynôme annulant α.

3.a Supposons que $\Pi_\alpha = \mathrm{UV}$ avec U, V non constants.

3.b Remarquer que l'hypothèse donne $\mathrm{P} \in \mathrm{I}(z)$.

4.a Mettre à profit une relation de Bézout.

4.b Si Π_α a une racine double, alors celle-ci annule sa dérivée.

5.a Écrire $\alpha = p/q$ sous forme de fraction irréductible avec $p \in \mathbb{Z}$ et $q \in \mathbb{N}^*$ puis montrer que $q \mid p^n$ en utilisant $\mathrm{P} \in \mathbb{Z}[\mathrm{X}]$ annulant α.

5.b Montrer que les racines complexes de Π_α sont des entiers algébriques. Utiliser les relations coefficients-racines pour une factorisation de Π_α dans $\mathbb{C}[\mathrm{X}]$ en lien avec le théorème admis.

6.a Justifier que $\Pi_\alpha = \mathrm{X}^2 + p\mathrm{X} + q$ avec p, q entiers et en déduire les valeurs possibles de p et q en écrivant $\alpha = e^{\,i\theta}$.

6.b Déterminer le polynôme minimal de $\alpha = (3+4i)/5$ puis remarquer qu'une racine de l'unité est nécessairement un entier algébrique.

Partie 2

7 Considérer une partition des racines n-ième de l'unité donnée par le théorème de Lagrange sur l'ordre d'un élément dans un groupe.

8.a Caractériser la racines primitives n-ième de la forme $e^{\,i\frac{2k\pi}{n}}$ et se rappeler que pour tout polynôme Y

$$\mathrm{Y}^m - 1 = (\mathrm{Y} - 1)(\mathrm{Y}^{m-1} + \cdots + \mathrm{Y} + 1)$$

8.b Utiliser les questions 7 et 8.

9.a Faire une récurrence forte sur n en utilisant les questions 7 et 8.

9.b Examiner les cas donnés par des résultats de la question 11.b pour n de 1 à 6, puis $n = p^k$ avec p premier. Enfin, faire une récurrence forte sur n en notant que

$$\Phi_n = (\mathrm{X}^n - 1) / \prod_{d \mid n, d < n} \Phi_d$$

10 Montrer que $\mathrm{X}^n - 1 = \Phi_n \mathrm{P}$ avec $\mathrm{P} \in \mathbb{Z}[\mathrm{X}]$.

11.a Majorer $|a_k|$ indépendamment de k en remarquant que n est fixé.

11.b Utiliser la forme factorisée de P pour écrire la fraction rationnelle P'/P de façon avantageuse.

11.c Écrire les deux membres de l'égalité précédente comme somme de série entière puis identifier leurs coefficients.

12.a Justifier que $(a_p, \ldots, a_{p+n}) \in [\![-n\,; n]\!]^{n+1}$ pour tout entier p.

12.b Se rappeler de la définition de a_{j+l} et a_{k+l}.

12.c Justifier que P et P′ sont premiers entre eux, puis que l'on peut trouver un polynôme $\mathrm{F} \in \mathbb{C}_{n-1}[\mathrm{X}]$ tel que $\mathrm{F}(z_i) = \delta_{i,j}$ pour tout $i \in [\![1\,; n]\!]$.

13.a Prouver que $p \mid \dbinom{p}{k}$ pour $k \in [\![1\,; p-1]\!]$.

13.b Se souvenir de la question 5.b puis appliquer le résultat de la question 13.a à Π_z écrit sous forme développée en pensant à Fermat.

13.c Utiliser le rappel sur l'anneau des entiers algébriques.

14.a Écrire P′ sous forme développée et sous forme factorisée pour exprimer le produit $\prod_{i=1}^{n} \mathrm{P}'(z_i)$.

14.b Remarquer que $z^p \in \mathbb{U}_n$ et écrire Π_z sous forme d'un produit de $\mathrm{X} - z$ pour z dans un sous-ensemble de $\mathbb{U}_n$ ne contenant pas z^p avant d'utiliser le résultat de la question 14.a.

14.c Si $\widetilde{p}$ est un nombre premier ne divisant pas n, prouver que $\Pi_z(z^{p\widetilde{p}}) = 0$. En conclure que $\Pi_z(z^k) = 0$ pour tout k premier avec n.

Partie 3

15.a Écrire $\mathrm{X}^d\,\mathrm{P}(1/\mathrm{X})$ en fonction d'une forme développée de P.

15.b Appliquer le résultat démontré en question 15.a à une factorisation du polynôme P dans $\mathbb{C}[\mathrm{X}]$.

16 Écrire $x = \mathrm{e}^{\,\mathrm{i}\theta}$ et remarquer que Π_x est en particulier à coefficients réels. Prouver ensuite que $\mathrm{X}^{\deg x}\,\Pi_x(1/\mathrm{X})$ est un polynôme annulateur de x.

17.a Utiliser $\Pi_\alpha = \Pi_\gamma$ et lui appliquer le résultat de la question 16.

17.b Prouver que si $\gamma^n = 1$ alors $\alpha^n = 1$.

17.c Noter que Π_α est réciproque et montrer que l'existence de $z \in \mathrm{C}(\alpha) \smallsetminus \{\alpha\}$ de module strictement inférieur à 1 aboutit à une contradiction.

18 Se souvenir du résultat de la question 15.b pour montrer que $\deg \Pi_\alpha$ est pair si $\alpha \in \mathcal{S}$. Trouver ensuite 4 racines deux à deux distinctes de Π_α à l'aide des résultats des questions 16 et 17.

Partie 4

19 Raisonner par l'absurde en supposant que $x = p/q \in \mathbb{Q}$ écrit sous forme irréductible est racine de P_n. Puis calculer $\mathrm{P}_n(1)$.

20 Remarquer que P_n est réciproque.

21 Mettre à profit les relations coefficients-racines pour P_n.

22 Remarquer que s_n, t_n sont racines d'un polynôme de degré 2 qu'on étudiera. Pour la suite, considérer l'application $f : x \mapsto x + 1/x$ sur $\mathbb{R}^*$ puis $\mathbb{C}^* \smallsetminus \mathbb{R}$.

23.a Justifier que s_n, t_n ne sont rationnels qu'à condition d'être entiers et examiner les résultats des questions 21 et 22.

23.b Les racines de P_n sont connues. Considérer sa décomposition sur $\mathbb{C}$ et en déduire des candidats pour celle sur $\mathbb{Q}$. Les questions 22 et 23.a permettent de conclure. En déduire que $\mathrm{P}_n = \Pi_{\alpha_n}$.

23.c Utiliser l'expression de s_n pour en calculer un développement asymptotique quand $n \to +\infty$, puis penser aux égalités de la question 21.

24 Justifier que l'ensemble $\mathcal{T}$ est non vide et minoré, puis prouver que toute suite minimisante tendant vers $\operatorname{Inf} \mathcal{T}$ ne comporte qu'un nombre fini de termes. Pour déterminer la valeur de $\operatorname{Min} \mathcal{T}$, décrire par CNS le polynôme minimal de $\alpha \in \mathcal{T}$ en reproduisant le même type de raisonnement qu'aux questions 21, 22 et 23. On pourra utiliser le lien entre les polynômes $\mathrm{Q} = \mathrm{X}^2 - a\mathrm{X} + b - 2$ quand $\Pi_\alpha = \mathrm{X}^4 - a\mathrm{X}^3 + b\mathrm{X}^2 - a\mathrm{X} + 1$ ainsi que l'expression explicite des racines du polynôme Q.

> Dans tout le corrigé, si $n \in \mathbb{N}^*$, on note $\mathbb{U}_n$ le groupe des racines n-ième de l'unité.

Partie 1

1 Comme α est algébrique, on a $\mathrm{I}(\alpha) \neq \{0\}$. De plus, par définition, $\mathrm{I}(\alpha) = \mathrm{Ker}\ \varphi_\alpha$ où φ_α est l'application

$$\varphi_\alpha : \begin{cases} \mathbb{Q}[\mathrm{X}] \longrightarrow \mathbb{C} \\ \quad \mathrm{P} \quad \longmapsto \mathrm{P}(\alpha) \end{cases}$$

laquelle est un morphisme d'anneaux d'après le cours. Ce dernier assure ainsi que

$$\boxed{\text{L'ensemble } \mathrm{I}(\alpha) \text{ est un idéal non réduit à } \{0\} \text{ de l'anneau } \mathbb{Q}[\mathrm{X}].}$$

> Dans toute la suite du corrigé, on note $\deg \alpha$ le degré de α, c'est-à-dire le degré du polynôme unitaire engendrant l'idéal $\mathrm{I}(\alpha)$.
>
> Rappelons aussi que l'idéal $\mathrm{I}(\alpha)$ est bien engendré par un unique polynôme unitaire car l'anneau $\mathbb{Q}[\mathrm{X}]$ est principal ($\mathbb{Q}$ est un corps).

2 Si $\alpha \in \mathbb{Q}$, alors $\mathrm{P} = \mathrm{X} - \alpha$ est un polynôme de $\mathbb{Q}[\mathrm{X}]$ annulant α et par conséquent on a $\deg \alpha = 1$.

Si $\deg \alpha = 1$, alors comme Π_α est unitaire, ce polynôme est de la forme $\mathrm{X} - a$ avec $a \in \mathbb{Q}$. Mais $\Pi_\alpha(\alpha) = 0$ ce qui implique que $a = \alpha$ donc $\alpha \in \mathbb{Q}$. Par double implication, il vient

$$\boxed{\deg \alpha = 1 \iff \alpha \in \mathbb{Q}}$$

3.a Supposons que Π_α n'est pas irréductible sur $\mathbb{Q}$. Soit $\Pi_\alpha = \mathrm{UV}$ une factorisation de Π_α dans $\mathbb{Q}[\mathrm{X}]$ avec U, V non constants, ce qui implique en particulier que U et V sont différents de Π_α.

La condition $\Pi_\alpha(\alpha) = 0$ assure que $\mathrm{U}(\alpha) = 0$ ou $\mathrm{V}(\alpha) = 0$ car $\mathbb{Q}$ est un corps. Quitte à échanger U et V on peut supposer que $\mathrm{U}(\alpha) = 0$, soit $\mathrm{U} \in \mathrm{I}(\alpha)$. On obtient alors que $\Pi_\alpha \mid \mathrm{U}$, d'où $\deg \Pi_\alpha \leqslant \deg \mathrm{U}$. Mais $\Pi_\alpha = \mathrm{UV}$ donc $\deg \mathrm{U} \leqslant \deg \Pi_\alpha$ et la divisibilité juste mentionnée assure que Π_α et U sont proportionnels donc V est constant, ce qui est une contradiction. Par suite,

$$\boxed{\text{Le polynôme } \Pi_\alpha \text{ est irréductible sur } \mathbb{Q}.}$$

3.b Soient $\mathrm{P} \in \mathbb{Q}[\mathrm{X}]$ unitaire irréductible sur $\mathbb{Q}$ et z racine complexe de P. On a donc $\mathrm{P} \in \mathrm{I}(z)$ et $\Pi_z \mid \mathrm{P}$. Comme P et Π_z sont irréductibles unitaires non constants, il vient que $\Pi_z = \mathrm{P}$. Ainsi,

$$\boxed{\text{Si } \mathrm{P} \in \mathbb{Q}[\mathrm{X}] \text{ unitaire irréductible sur } \mathbb{Q} \text{ s'annule en } z \in \mathbb{C}, \text{ alors } \mathrm{P} = \Pi_z.}$$

X Maths B MP 2019 — Corrigé

Ce corrigé est proposé par Loïc Devilliers (professeur en CPGE) ; il a été relu par Hugues Zuber (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

Ce sujet propose une généralisation de l'étude qualitative des équations différentielles. En effet, la recherche moderne dans le domaine des équations différentielles introduit des notions généralisant la notion de dérivabilité vue et étudiée au lycée ou dans les études supérieures. Ainsi, il est possible de transformer la notion d'équation différentielle en une équation fonctionnelle où la dérivée ne figure plus. Ces outils arrivent parfois à montrer l'existence ou l'unicité des solutions là où le théorème de Cauchy-Lipschitz est impuissant.

- Dans la première partie, on étudie une équation différentielle d'ordre 1 non linéaire (car une valeur absolue y est présente). On montre qu'elle n'a pas de solution de classe $\mathscr{C}^1$ (cas de non existence) sur son domaine de définition. En revanche, elle admet deux solutions si on relâche la contrainte en 0 (cas de non unicité). Cette partie est plutôt facile, mais devait être rédigée avec rigueur.
- La deuxième partie propose l'étude des sous-différentiels et des sur-différentiels d'une fonction. Ce sont des ensembles qui généralisent la notion de dérivée de cette fonction. On fait donc le lien entre ces nouvelles notions et la notion classique de dérivée. Lorsque les fonctions ne sont pas dérivables, le sujet propose de réécrire les sous-différentiels et sur-différentiels avec une définition utilisant les limites ainsi que les bornes supérieures et inférieures. Enfin, lorsque la fonction est concave, il existe une description encore plus simple des sur-différentiels.
- L'équation différentielle du début du sujet peut être remplacée par une équation fonctionnelle. Pour cela, la dérivée est remplacée par les sous/sur-différentiels étudiés dans la partie précédente. Dans la troisième partie, cette équation fonctionnelle est scindée en deux inéquations fonctionnelles que l'on étudie. On établit une inégalité entre deux fonctions vérifiant chacune l'une des deux inéquations fonctionnelles.
- Dans la dernière partie, un cas particulier de l'équation fonctionnelle est étudié. On se ramène ainsi à l'équation différentielle étudiée dans la première partie. On montre l'unicité de la solution en utilisant les résultats de la partie précédente.

C'est un sujet long et technique ; la rédaction de certaines questions est longue et pénible si on veut la faire proprement. De plus, quelques questions demandent de prendre beaucoup d'initiatives. Une des difficultés majeures est qu'il faut savoir appréhender de nouvelles notions, comme les sur/sous-différentiels ou les sur/sous-solutions, dans un temps court. Une excellente maîtrise des limites avec les quantificateurs, des bornes supérieures et du maniement des inégalités est indispensable pour bien traiter ce sujet.

En revanche, ce problème propose une démarche très intéressante car utilisée en recherche mathématique : généraliser la notion de dérivabilité. Cette idée est l'un des fondements de la théorie des distributions et de l'optimisation convexe.

Indications

Partie I

1a Si u' a un signe constant au voisinage de y simplifier la valeur absolue et dériver à nouveau l'équation.

1b Étudier le taux d'accroissement de u'.

2 Résoudre $u'' = u$ avec les conditions initiales.

Partie II

4 Par double inclusion : pour la première inclusion, utiliser le fait que si une fonction f a un extremum en un point x_0 intérieur à son domaine de définition, alors $f'(x_0) = 0$. Pour la seconde inclusion, poser $\varphi = u$.

5a Soit ψ_1 tel que $u - \psi_1$ ait un minimum local en x_0. Modifier ψ_1 pour que sa valeur en x_0 corresponde à ce qui est demandé.

5b Comparer les taux d'accroissement en x_0 de φ_1 et φ_2 en séparant bien les cas $x > x_0$ et $x < x_0$. Puis appliquer le théorème d'encadrement entre les taux d'accroissement de u, φ_1 et φ_2 en x_0.

6a Étudier $u - \varphi_{x_0,r}$ en particulier ses limites aux extrémités de $\mathrm{I}_{x_0}(r)$ pour montrer que sa valeur maximum est atteinte sur $\mathrm{I}_{x_0}(r)$.

7a Écrire la définition de $p \in \mathrm{D}^+u(x_0)$ avec une fonction φ puis utiliser le développement limité à l'ordre 1 de φ en x_0.

7b Comme (**3**) est supposée vraie, il existe au moins un $\varepsilon_0 > 0$ tel que la borne supérieure dans (**3**) existe bien. Quand ε varie, comparer ces bornes supérieures. Ne pas hésiter à introduire des notations supplémentaires.

7c Étudier la continuité à droite et à gauche de φ en tout point de $\mathbb{R}_+^*$.

7d Entre r et $2r$, minorer φ.

7e Utiliser l'inégalité démontrée en 7*b* et ce l'on sait sur ρ pour construire une fonction ψ tel que $u - \psi$ ait un maximum local en x_0.

9 Montrer que $\mathrm{D}^+u(x_0)$ est convexe et appliquer la caractérisation séquentielle des fermés.

10c Utiliser le résultat de 10b ainsi que la décroissance des pentes montrée en 10a.

Partie III

12 Appliquer le théorème de Heine à u et v et utiliser la continuité de ω en 0.

13 Comparer $\Phi_\eta(x_\eta, y_\eta)$ à $\Phi_\eta(x_0, x_0)$.

14b Supposer que $|x_\eta| = 1$, puis minorer/majorer $|x_\eta - y_\eta|$ en utilisant ce qui précède pour trouver une contradiction.

14c Écrire que Φ_η a un maximum en (x_η, y_η) pour trouver φ une fonction dont la dérivée en x_η va satisfaire la définition de $\mathrm{D}^+u(x_\eta)$.

Partie IV

16d Montrer que $0 \in \mathrm{D}^-u_1(0)$. En déduire que u_1 n'est pas sur-solution de (**1**).

16e Considérer une autre sur-solution et sous-solution et appliquer le résultat de la question 15.

18a Soit u une solution de (**5**), résoudre u sur des intervalles sur lesquels u' a un signe constant. Puis considérer un point sur lequel u' change de signe après avoir prouvé qu'un tel point existe.

Partie I

$\boxed{\textbf{1a}}$ Comme u est de classe $\mathscr{C}^1$ sur $[-1\,;1]$, l'application $x \mapsto |u'(x)| = -u(x)$ l'est également. Soit $y \in [-1\,;1]$ tel que $u'(y) \neq 0$. Il y a deux cas :

- Si $u(y) > 0$, alors, par continuité de u en y, il existe un voisinage J de y tel que pour tout $x \in \mathrm{J} \cap \mathrm{I}$, $u(x) > 0$. Ainsi, pour tout $x \in \mathrm{J} \cap \mathrm{I}$, $u(x) + u'(x) = 0$. Par conséquent, $u' = -u$ sur J et comme u est $\mathscr{C}^1$ sur J, on en déduit que u' l'est également. Ainsi, $u \in \mathscr{C}^2(\mathrm{J} \cap \mathrm{I})$, prouvant que u est de classe $\mathscr{C}^2$ au voisinage de y. En dérivant la relation $u' = -u$ sur $\mathrm{J} \cap \mathrm{I}$, on obtient $u'' = -u'$. En particulier,
$$u''(y) = -u'(y) = -|u'(y)|$$
- De même, si $u(y) < 0$, alors, par continuité de u en y, il existe un voisinage J de y tel que pour tout $x \in \mathrm{J} \cap \mathrm{I}$, $u(x) < 0$. Ainsi, $u' = u$ sur J et comme u est de classe $\mathscr{C}^1$, sur J, on en déduit que u' l'est également. Par conséquent, u est de classe $\mathscr{C}^2$ au voisinage de y et $u'' = u'$. En particulier, $u''(y) = u'(y) = -|u'(y)|$.

$$\boxed{\begin{array}{l}\text{La fonction } x \mapsto |u'(x)| \text{ est } \mathscr{C}^1([-1\,;1]). \text{ De plus, pour tout } y \in [-1\,;1],\\ \text{si } u'(y) \neq 0, \text{ alors } u \text{ est } \mathscr{C}^2 \text{ sur un voisinage de } y \text{ et } u''(y) = -|u'(y)|.\end{array}}$$

$\boxed{\textbf{1b}}$ Soit $y \in [-1\,;1]$. Supposons $u'(y) = 0$. En particulier, $u(y) = -|u'(y)| = 0$. Calculons la valeur absolue du taux d'accroissement de u' en y. Soit $x \in [-1\,;1] \smallsetminus \{y\}$:
$$\left|\frac{u'(x) - u'(y)}{x - y}\right| = \left|\frac{|u'(x)|}{x - y}\right| = \left|-\frac{u(x)}{x - y}\right| = \left|\frac{u(x) - u(y)}{x - y}\right|$$
Or comme u est dérivable en y de dérivée $u'(y) = 0$, on en déduit que
$$\lim_{x \to y} \frac{u(x) - u(y)}{x - y} = 0$$
La valeur absolue étant continue, on en déduit que
$$\lim_{x \to y} \left|\frac{u'(x) - u'(y)}{x - y}\right| = \lim_{x \to y} \left|\frac{u(x) - u(y)}{x - y}\right| = 0$$
Ceci prouve que u' est dérivable en y et que $u''(y) = 0$. En conclusion
$$\boxed{\text{Si } u'(y) = 0 \text{ alors la fonction } u' \text{ est dérivable en } y \text{ et } u''(y) = 0.}$$

$\boxed{\textbf{2}}$ Soit $y \in [-1\,;1]$. Alors

- si $u'(y) \neq 0$, alors en utilisant la question 1a, $u''(y) = -|u'(y)| = u(y)$;
- si $u'(y) = 0$, alors d'après la question 1b, $u''(y) = 0 = -|u'(y)| = u(y)$.

Dans tous les cas, $u''(y) = u(y)$, d'où $u'' = u$. La fonction $u = u''$ étant continue, il vient que u est de classe $\mathscr{C}^2$ sur $[-1\,;1]$. Finalement,
$$\boxed{\text{La fonction } u \text{ est de classe } \mathscr{C}^2 \text{ sur } [-1\,;1] \text{ et vérifie } u'' = u.}$$

Comme $u'' = u$, on peut affirmer qu'il existe $(\mathrm{A}, \mathrm{B}) \in \mathbb{R}^2$ tel que
$$\forall x \in [-1\,;1] \quad u(x) = \mathrm{A}e^{x} + \mathrm{B}e^{-x}$$
En utilisant les conditions initiales et finales de u , il vient
$$u(1) = \mathrm{A}e^{1} + \mathrm{B}e^{-1} = -1 \qquad \text{et} \qquad u(-1) = \mathrm{A}e^{-1} + \mathrm{B}e^{1} = -1$$
Par différence, $\qquad u(1) - u(-1) = 2\mathrm{A}\,\mathrm{sh}\,(1) - 2\mathrm{B}\,\mathrm{sh}\,(1) = 0$

Comme $\text{sh}\,(1) \neq 0$, il vient $\text{B} = -\text{A}$, ainsi $u = 2\text{A}\,\text{ch}$. En particulier, $u(0) = 2\text{A}$ et $u'(0) = 0$. En utilisant la relation $u(0) + |u'(0)| = 0$, il en découle $2\text{A} + 0 = 0$ puis $\text{A} = 0$. Dès lors, u est la fonction nulle. Ceci contredit la condition $u(1) = -1$. Pour conclure

$$\boxed{\text{Il n'existe pas de fonction } u \in \mathscr{C}^1([-1\,;1]) \text{ vérifiant } (\mathbf{2}).}$$

$\boxed{\mathbf{3}}$ Montrons que u_0 et u_1 vérifient les propriétés demandées :

- Les fonctions valeur absolue et exponentielle étant continues sur $\mathbb{R}$, on en déduit, par composition, que u_0 et u_1 sont continues sur $[-1\,;1]$.
- $u_0(-1) = u_0(1) = -1$ et $u_1(-1) = u_1(1) = -1$.
- Pour tout $x \in]\,0\,;1]$, $u_0(x) = -\text{e}^{-1+x}$ et $u_1(x) = -\text{e}^{1-x}$. De même, pour tout $x \in [-1\,;0\,[$, $u_0(x) = -\text{e}^{-1-x}$ et $u_1(x) = -\text{e}^{1+x}$. On en déduit par composition que u_0 et u_1 sont dérivables sur $[-1\,;1] \smallsetminus \{0\}$ avec

$$\begin{cases} {u_0}'(x) = -\text{e}^{-1+x} & \text{et} \quad {u_1}'(x) = \text{e}^{1-x} \quad \text{si } x \in]\,0\,;1] \\ {u_0}'(x) = \text{e}^{-1-x} & \text{et} \quad {u_1}'(x) = -\text{e}^{1+x} \quad \text{si } x \in [-1\,;0\,[\end{cases}$$

$$\text{d'où,} \quad u_0(x) + |{u_0}'(x)| = \begin{cases} -\text{e}^{-1+x} + \text{e}^{-1+x} = 0 & \text{si } x \in]\,0\,;1] \\ -\text{e}^{-1-x} + \text{e}^{-1-x} = 0 & \text{si } x \in [-1\,;0\,[\end{cases}$$

$$\text{et,} \quad u_1(x) + |{u_1}'(x)| = \begin{cases} -\text{e}^{1-x} + \text{e}^{1-x} = 0 & \text{si } x \in]\,0\,;1] \\ -\text{e}^{1+x} + \text{e}^{1+x} = 0 & \text{si } x \in [-1\,;0\,[\end{cases}$$

On peut en conclure que

$$\boxed{\begin{array}{l}\text{Les fonctions } u_0 \text{ et } u_1 \text{ sont des fonctions de } \mathscr{C}^0([-1\,;1]) \text{ vérifiant pour} \\ \text{tout } x \in [-1\,;1] \smallsetminus \{0\},\ u(x) + |u'(x)| = 0 \text{ ainsi que } u(-1) = u(1) = -1.\end{array}}$$

Partie II

$\boxed{\mathbf{4}}$ Supposons u de classe $\mathscr{C}^1$ au voisinage de x_0. Procédons par double inclusion.

Soit $p \in \text{D}^+u(x_0)$ (respectivement $p \in \text{D}^-u(x_0)$), alors il existe φ une fonction de classe $\mathscr{C}^1$ au voisinage de x_0 avec $\varphi'(x_0) = p$ et telle que $u - \varphi$ admet un maximum local (respectivement un minimum local) en x_0. Ainsi x_0 est un extremum local de $u - \varphi$. De plus, $u - \varphi$ est dérivable en x_0 comme différence de fonctions dérivables en x_0. Notons, de plus, que x_0 est un point intérieur à l'ensemble de définition de $u - \varphi$. Donc $(u - \varphi)'(x_0) = 0$. Par conséquent

$$p = \varphi'(x_0) = u'(x_0) \in \{u'(x_0)\}$$

Par suite, $\text{D}^+u(x_0) \subset \{u'(x_0)\}$ (respectivement $\text{D}^-u(x_0) \subset \{u'(x_0)\}$).

Réciproquement, montrons que le réel $u'(x_0)$ appartient à $\text{D}^+u(x_0)$ et à $\text{D}^-(x_0)$. Posons $\varphi = u$. Dès lors, φ est de classe $\mathscr{C}^1$ au voisinage de x_0, et $u - \varphi$ étant la fonction nulle, x_0 est bien un maximum local et un minimum local de $u - \varphi$. Ainsi,

$$u'(x_0) = \varphi'(x_0) \in \text{D}^+u(x_0) \qquad \text{et} \qquad u'(x_0) = \varphi'(x_0) \in \text{D}^-u(x_0)$$

Ceci montre que $\{u'(x_0)\} \subset \text{D}^+u(x_0)$ et $\{u'(x_0)\} \subset \text{D}^-u(x_0)$.

Par double inclusion, il y a alors égalité entre ces trois ensembles. En conclusion,

$$\boxed{\text{Si } u \text{ est } \mathscr{C}^1 \text{ sur un voisinage de } x_0, \text{ alors } \text{D}^+u(x_0) = \text{D}^-u(x_0) = \{u'(x_0)\}.}$$

X/ENS Informatique B MP-PC-PSI 2019 — Corrigé

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

Ce sujet a pour objectif la réalisation de fonctions qui pourraient se trouver à la base d'un jeu de stratégie, dans lequel un barreau vertical de k cases colorées descend, guidé par le joueur, comme dans le célèbre jeu Tetris. Lorsqu'il touche le sol, les alignements de cases de même couleur sont supprimés et des points sont marqués. Si l'interface graphique et les actions du joueur ne sont pas abordées, le sujet traite néanmoins un grand nombre de problématiques telles que l'initialisation du jeu, les déplacements, la détection d'alignements, les suppressions et le comptage des points. L'étude est divisée en cinq parties de tailles variées.

- La première partie étudie l'initialisation de la grille et son affichage en deux questions plutôt abordables.
- On traite ensuite l'apparition du barreau et ses différents mouvements : descente case par case ou plus rapide, déplacement latéral, permutation des cases. Les questions sont d'un niveau intermédiaire et contiennent peu de pièges. Elles demandent avant tout une manipulation correcte de la liste de listes qui modélise la grille du jeu.
- La troisième partie, plus longue que les autres, constitue le cœur de l'épreuve et nous fait inspecter une grille où un barreau vient de toucher le sol. Il faut détecter les alignements de couleurs, les supprimer, faire descendre les cases qui se trouvent alors suspendues et ainsi de suite. Les fonctions à produire sont plus longues et plus élaborées.
- La quatrième partie n'est constituée que de deux questions, traitant une variante bien plus complexe de détection des alignements. Ces questions sont plus difficiles car moins directives. La deuxième en particulier requiert l'analyse assez fine d'un code de plus de cinquante lignes, peu envisageable en moins de dix minutes.
- Enfin, la gestion d'une base de données contenant les scores de plusieurs parties et de plusieurs joueurs est abordée. L'ensemble des quatre questions permet d'évaluer correctement les candidats sur l'écriture de requêtes SQL.

Ce sujet est progressif et intéressant. Le contexte du jeu de grille est séduisant, même si la référence à Tetris ne fonctionne peut-être pas aussi bien avec les élèves qu'avec leurs professeurs, qui d'ailleurs peuvent être déçus du lien finalement quelque peu éloigné avec le jeu original. Si les questions 14 et 15 ne sont accessibles qu'aux meilleurs candidats dans le temps imparti, les autres construisent un sujet bien équilibré et classant, accessible (hors partie 4) dès la première année. Les chapitres d'ingénierie numérique ne sont pas abordés.

Indications

Partie I

2 Le parcours de la grille doit se faire par deux boucles `for` imbriquées, dans le bon sens : la boucle la plus interne permet de décrire une ligne.

Partie II

3 Il faut tester chaque colonne séparément sur les `k` cases les plus élevées, mais il est inutile de tester obligatoirement toutes les colonnes. Un `return` dans une boucle `for` permet une sortie anticipée.

4 Afin de ne réaliser la modification que si la case sous le barreau existe et est vide, il est possible de placer la boucle de déplacement à l'intérieur d'un test `if`.

5 Si une des cases de destination n'est pas vide, l'algorithme doit s'arrêter sans réaliser le déplacement. L'instruction `return None` est alors toute indiquée.

6 Il faut remplacer les cases de haut en bas, en faisant très attention à traiter toutes les cases du barreau et seulement celles-là.

7 On doit commencer par parcourir la grille vers le bas à partir de `(x,y)` pour déterminer la distance de déplacement du barreau, et le déplacer alors en une seule boucle. La procédure `deplacerBarreau` n'est pas utilisable.

Partie III

9 Il faut, lors de l'unique parcours autorisé de `rangee`, compter le nombre de cases adjacentes de même couleur. À chaque changement de couleur, on peut remettre à zéro un compteur après avoir modifié `marking` sur le bon nombre de cases. Attention à ne pas oublier l'éventuel dernier alignement.

10 Utiliser la fonction `detecteAlignement` en construisant la bonne rangée avant.

11 Il faut appliquer consécutivement la fonction `scoreRangee` sur toutes les rangées pouvant exister dans la grille. Il est donc important de les prévoir correctement, en s'aidant de schémas. Attention à ne pas traiter deux fois la même rangée.

12 Plusieurs techniques sont possibles. On peut par exemple parcourir chaque colonne de bas en haut en gardant dans une variable supplémentaire la position de la case de destination des déplacements.

Partie IV

14 Pour que la fonction puisse s'exécuter récursivement sans revenir sans cesse sur les mêmes cases, il faut modifier `grille` en effaçant les cases déjà parcourues. Les codes donnés dans l'énoncé aux figures 9 et 10 ne font pas partie de cette question.

15 La principale difficulté est dans la fonction `exploreHorizontal`. Il faut essayer de comprendre son fonctionnement global et d'en déduire quel genre de disposition de cases pourrait conduire à en oublier.

Partie V

17 Il faut compter le nombre de scores meilleurs que `s`.

18 Il s'agit ici d'obtenir la valeur maximale d'un champ.

19 Sélectionner les joueurs ayant un meilleur score que le joueur concerné nécessite l'insertion d'une sous-requête dans la clause `WHERE` et une agrégation de résultats. Le comptage des joueurs doit s'effectuer dans un deuxième temps, à l'aide d'une deuxième agrégation.

I. Initialisation et affichage de l'aire de jeu

1 La fonction `creerGrille` doit créer une par une chaque case de la grille et les assembler dans les deux directions.

```
def creerGrille(largeur,hauteur):
    grille = []
    for i in range(largeur):
        colonne = []
        for j in range(hauteur):
            colonne.append(VIDE)
        grille.append(colonne)
    return grille
```

> D'après l'énoncé, `VIDE` est une variable globale, déjà définie. Il ne faut donc pas l'entourer de guillemets, ni réaliser d'affectation de cette variable.
>
> Attention à ne pas recopier `largeur` fois la même colonne. En effet, les colonnes étant des listes, créer une unique colonne et la copier plusieurs fois conduirait à créer des colonnes aux valeurs toujours identiques. Par exemple,
>
> ```
> colonne = [0,0,0]
> grille = []
> for i in range(4):
>     grille.append(colonne)
> grille[2,1] = 1
> ```
>
> construit une `grille` égale à `[[0,1,0],[0,1,0],[0,1,0],[0,1,0]]`.
>
> Créer chaque colonne en une ligne à l'aide de la syntaxe `liste*entier` permet de réaliser un code correct plus compact :
>
> ```
> def creerGrille(largeur,hauteur):
>     grille = []
>     for i in range(largeur):
>         grille.append([VIDE]*hauteur)
>     return grille
> ```
>
> L'énoncé rappelle en préambule comment définir en compréhension une liste contenant n occurrences d'une valeur k. Ce type de syntaxe permet en effet d'écrire un code bien plus compact avec
>
> ```
> def creerGrille(l,h):
>     return [ [ VIDE for j in range(h) ] for i in range(l) ]
> ```

2 La procédure `afficheGrille` doit lire chaque ligne en partant de la plus « haute », c'est-à-dire la fin de chaque colonne.

```
def afficherGrille(grille):
    largeur = len(grille)
    hauteur = len(grille[0])
    for j in range(hauteur-1,-1,-1):
        for i in range(largeur):
            if grille[i][j] == VIDE:
                afficheBlanc()
            else:
                afficheCouleur(grille[i][j])
        nouvelleLigne()
```

> La procédure `nouvelleLigne` peut aussi être positionnée en début de première boucle. Les indications de l'énoncé sont insuffisantes pour choisir la meilleure position, mais celle adoptée ici correspond à une écriture classique des procédures `afficheBlanc`, `afficheCouleur` et `nouvelleLigne` de la forme
>
> ```
> def afficheBlanc():
>     print(' ',end='')
> def nouvelleLigne():
>     print()
> ```
>
> où `nouvelleLigne` crée ainsi un retour à la ligne, inutile en début d'affichage mais nécessaire à la fin.

II. Création et mouvement du barreau

$\boxed{3}$ La fonction `grilleLibre` permet de déterminer si `k` cases verticalement adjacentes sont vides en haut de grille. Dès que l'on obtient un tel alignement, il est donc possible d'arrêter la recherche.

```
def grilleLibre(grille,k):
    largeur = len(grille)
    hauteur = len(grille[0])
    for i in range(largeur):
        # Marqueur indiquant si la colonne a assez de cases libres
        placelibre = True
        # Si une case est prise, on change le marqueur
        for j in range(hauteur-k, hauteur):
            if grille[i][j] != VIDE:
                placelibre = False
        # Si on a trouvé une colonne disponible, on s'arrête
        if placelibre:
            return True
    # Si on arrive ici, il n'y a aucune colonne disponible
    return False
```

Dans le pire des cas, on inspecte les `k` plus hautes cases de chaque colonne de la grille. L'inspection d'une case, contenue dans la deuxième boucle, est de complexité constante. La complexité totale est donc

$$\boxed{\mathrm{O}(k \times \texttt{largeur})}$$

> L'hypothèse de non-tassement de la grille semble étonnante, la grille étant naturellement toujours tassée lorsqu'un barreau apparaît. On peut imaginer qu'il s'agit simplement de s'assurer que la seule réponse correcte soit de complexité $\mathrm{O}(k \times \texttt{largeur})$.