

MP

Mathématiques · Informatique

2021

Sous la coordination de

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

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

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

Par

Virgile ANDREANI
ENS Ulm

Antoine BARRIER
ENS Paris-Saclay

Jean-Paul BONNET
professeur en CPGE

Sélim CORNET
professeur agrégé

Julien DUMONT
professeur en CPGE

Christophe FISZKA
professeur en CPGE

Titouan LECLERCQ
ENS Lyon

Pacôme LUTON
ENS Lyon

Florian METZGER
professeur en CPGE

Angèle NICLAS
ENS Lyon

Tristan POULLAOUEC
professeur en CPGE

Vincent PUYHAUBERT
professeur en CPGE

Cyril RAVAT
professeur en CPGE

Jean STARYNKÉVITCH
professeur en CPGE

Bertrand WIEL
professeur en CPGE

Sommaire thématique de mathématiques

2015 – 2021

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

Sommaire

		Énoncé	Corrigé
E3A			
Mathématiques 1	Quatre exercices indépendants. <i>suites et séries de fonctions, réduction des endomorphismes, probabilités, polynômes, intégration</i>	17	21
CONCOURS COMMUN INP			
Mathématiques 1	Interversion somme-intégrale. Recherche d'extremum. Un peu d'arithmétique avec la fonction zêta de Riemann. <i>séries, intégrales généralisées, programmation Python, familles sommables, arithmétique, variables aléatoires</i>	38	43
Mathématiques 2	Théorème de décomposition de Dunford. <i>réduction, nilpotence, diagonalisabilité</i>	57	61
Informatique optionnelle	Partitions non croisées, problème Horn-Sat et classes sylvestres. <i>dénombrement, logique, satisfiabilité, arbres binaires de recherche</i>	76	87
CENTRALE-SUPÉLEC			
Mathématiques 1	Loi du demi-cercle. <i>dénombrement, séries entières, topologie, probabilités, réduction</i>	108	114
Mathématiques 2	Inégalités de Bernstein. <i>trigonométrie, polynômes, intégration, topologie, endomorphismes</i>	140	144
Informatique commune	Lancer de rayons. <i>numpy, SQL, bases de données, images</i>	162	170
Informatique optionnelle	Génération aléatoire de pavages d'un échiquier. <i>arbres, graphes, programmation OCaml, complexité, tableaux, randomisation</i>	183	192

MINES-PONTS

Mathématiques 1	Théorème de De Moivre-Laplace. <i>suites et séries numériques, probabilités, intégration</i>	213	218
Mathématiques 2	Fonctions de matrices symétriques, continuité et convexité. <i>topologie, calculs matriciels, réduction, suites et séries de fonctions</i>	238	244
Informatique commune	Marchons, marchons, marchons... <i>programmation Python, bases de données, complexité</i>	263	274
Informatique optionnelle	Le jeu du solitaire. <i>parcours de graphe, opérations binaires sur les entiers, automates</i>	286	298

POLYTECHNIQUE-ENS

Mathématiques A	Sous-groupes finis de $GL_n(\mathbb{Z})$. <i>algèbre générale, algèbre linéaire, arithmétique, réduction, groupes, polynômes</i>	313	319
Mathématiques B	Autour de la loi zêta. <i>probabilités, arithmétique, intégration, polynômes, convexité, familles sommables</i>	339	345
Informatique commune	Gestion d'un allocateur dynamique de mémoire. <i>algorithmique, représentation binaire, programmation Python</i>	377	388

FORMULAIRES

Développements limités usuels en 0	401
Développements en série entière usuels	402
Dérivées usuelles	403
Primitives usuelles	404
Trigonométrie	406

SESSION 2021



MP8M

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES

Durée : 4 heures

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de quatre exercices indépendants.

Exercice 1

Dans tout l'exercice, I est le segment $[0, 1]$ et f la fonction définie sur I par : $x \mapsto \begin{cases} x^{-x} & \text{si } x \neq 0 \\ 1 & \text{si } x = 0 \end{cases}$.

On considère la suite de fonctions $(f_n)_{n \in \mathbb{N}}$ définies sur I par :

$$\bullet \forall x \in I, f_0(x) = 1$$

$$\bullet \forall n \in \mathbb{N}^*, \forall x \in I, f_n(x) = \begin{cases} 0 & \text{si } x = 0 \\ \frac{(-1)^n}{n!} (x \ln(x))^n & \text{sinon} \end{cases}$$

1. Montrer que f et toutes les fonctions f_n sont continues sur I .

2. On considère la série de fonctions $\sum_{n \geq 0} f_n$.

Démontrer que cette série de fonctions converge simplement sur I vers une fonction que l'on déterminera.

3. Étudier les variations de la fonction φ continue sur I , définie pour tout $t \in]0, 1]$ par $\varphi(t) = t \ln(t)$.

4. Représenter graphiquement la fonction φ sur I en précisant les tangentes aux bornes.

5. Démontrer que la série de fonctions $\sum_{n \geq 0} f_n$ converge normalement sur I .

6. On pose pour tout réel x et lorsque cela est possible $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$.

6.1. Déterminer l'ensemble de définition de la fonction Γ .

6.2. Soit $n \in \mathbb{N}$. Calculer $\Gamma(n+1)$.

7. Soit $n \in \mathbb{N}^*$. Calculer l'intégrale $J_n = \int_0^1 f_n(t) dt$.

On pourra effectuer le changement de variable $u = -\ln(t)$.

8. On pose $J = \int_0^1 f(t) dt$. Montrer que l'on a : $J = \sum_{n=1}^{+\infty} n^{-n}$.

9. Trouver un rang n_0 pour lequel la somme partielle d'ordre n_0 sera une valeur approchée de J à 10^{-6} près.

Exercice 2

Soient n un entier naturel supérieur ou égal à 2 et E un espace euclidien de dimension n dont le produit scalaire est noté $(\cdot)$ et la norme $\|\cdot\|$.

On note id_E l'endomorphisme identité de E et θ l'endomorphisme nul de E .

1. Soit f un endomorphisme symétrique de E que l'on suppose non inversible et non nul.

1.1. Citer le **théorème spectral**.

1.2. Montrer que 0 est valeur propre de f et que f admet au moins une valeur propre non nulle.

1.3. Montrer que les sous-espaces $\text{Ker}(f)$ et $\text{Im}(f)$ sont orthogonaux.

Sont-ils supplémentaires ? On justifiera la réponse.

On suppose désormais et jusqu'à la fin de l'exercice que f admet exactement $k+1$ valeurs propres deux à deux distinctes $(\lambda_j)_{j \in \llbracket 0, k \rrbracket}$ avec :

$$k \geq 1, \lambda_0 = 0 \text{ et } 0 < |\lambda_1| \leq \dots \leq |\lambda_k|.$$

Pour tout $j \in \llbracket 0, k \rrbracket$, on note E_j le sous-espace propre associé à la valeur propre λ_j et p_j le projecteur orthogonal sur E_j .

1.4. Montrer que $\text{id}_E = \sum_{j=0}^k p_j$.

1.5. Prouver que l'on a pour tout couple (i, j) de $\llbracket 0, k \rrbracket^2$ tels que $i \neq j$, $p_i \circ p_j = \theta$.

1.6. Démontrer que : $f = \sum_{j=0}^k \lambda_j p_j$.

1.7. Soit p le projecteur orthogonal sur $\text{Im}(f)$. Montrer que l'on a : $p = \sum_{j=1}^k p_j$.

On note alors f^I l'endomorphisme de E défini par : $f^I = \sum_{j=1}^k \frac{1}{\lambda_j} p_j$, appelé **inverse généralisé** de f .

2. Quelques propriétés de l'inverse généralisé

2.1. Montrer que l'on a : $f \circ f^I = p$.

En déduire que : $\forall (x, y) \in E^2, (f(x) = p(y) \iff x - f^I(y) \in \text{Ker}(f))$.

2.2. Soit y un vecteur de E .

Montrer que l'on a : $\forall x \in E, (\|f(x) - y\| = \inf_{z \in E} \|f(z) - y\| \iff x - f^I(y) \in \text{Ker}(f))$.

3. Application à un exemple

On prend E un espace euclidien de dimension 4 et $\mathcal{B} = (e_1, e_2, e_3, e_4)$ une base orthonormale de E .

Soit f l'endomorphisme de E dont la matrice dans $\mathcal{B}$ est : $A = \begin{pmatrix} 3 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 3 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$

3.1. Justifier que f est un endomorphisme symétrique, non nul et non inversible.

3.2. Montrer que 2 est valeur propre double de la matrice A .

3.3. En déduire que f admet exactement 3 valeurs propres : $\lambda_0 < \lambda_1 < \lambda_2$.

On note pour tout $j \in \llbracket 0, 2 \rrbracket$, M_j la matrice de p_j dans la base $\mathcal{B}$.

3.4. Justifier que l'on peut écrire A sous la forme : $A = 2M_1 + 4M_2$.

3.5. Montrer que E_2 est de dimension 1 et déterminer un vecteur v_2 de E_2 tel que $\|v_2\| = 1$.

3.6. Démontrer que : $\forall x \in E, p_2(x) = (x|v_2)v_2$.

3.7. Déterminer la matrice M_2 .

4. En déduire la matrice associée à f^I relativement à la base $\mathcal{B}$.

Exercice 3

Soient X et Y deux variables aléatoires indépendantes à valeurs dans $\mathbb{N}$ définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.

Pour $|t| < 1$, on définit les fonctions génératrices de X et de Y respectivement par :

- $G_X(t) = \frac{1}{2-t},$
- $G_Y(t) = 2 - \sqrt{2-t}.$

e3a Mathématiques MP 2021 — Corrigé

Ce corrigé est proposé par Tristan Poullaouec (professeur en CPGE) ; il a été relu par Sélim Cornet (professeur agrégé) et Gilbert Monna (professeur honoraire en CPGE).

Ce sujet est constitué de quatre exercices indépendants qui abordent les grands thèmes du programme.

- Au cours du premier exercice, on établit la convergence normale d'une série de fonctions vers la fonction $x \mapsto x^{-x}$ prolongée par continuité en 0, afin de calculer l'intégrale de cette fonction sur $[0; 1]$. On retrouve au passage la fonction Γ d'Euler, dont on détermine l'expression sur $\mathbb{N}^*$.
- On s'intéresse ensuite à l'inverse généralisé d'un endomorphisme symétrique f d'un espace euclidien, que l'on définit à l'aide des projecteurs orthogonaux sur les sous-espaces propres de f . L'étude débouche sur un exemple concret en dimension 4.
- Dans le troisième exercice, on étudie deux variables aléatoires entières qui sont caractérisées par leurs fonctions génératrices. On détermine leurs lois, espérances et variances, ainsi que celles de leur somme.
- Le dernier exercice est consacré à l'étude de formes linéaires définies à l'aide d'intégrales sur l'espace $\mathbb{R}_n[X]$.

Ce sujet ne contient pas de difficulté particulière : les questions sont bien détaillées, les résultats essentiels apparaissent explicitement dans l'énoncé et les quatre exercices restent très proches du cours (on retrouve d'ailleurs quelques grands classiques). Rien ne s'oppose donc à ce qu'un candidat bien préparé le traite dans le temps imparti.

INDICATIONS

Exercice 1

- 1.1 Ces fonctions étant clairement toutes continues sur l'intervalle $]0; 1]$, il suffit d'étudier leur continuité (à droite) en 0.
- 1.2.1 Utiliser le développement en série entière de la fonction exponentielle sur $\mathbb{R}$.
- 1.5.1 Se servir de la question 1.3 pour calculer $\|f_n\|_\infty$.
- 1.6.1 Étudier l'intégrabilité sur $]0; 1]$ et sur $[1; +\infty[$ en comparant à des intégrales de Riemann.
- 1.7 Pour $a \in]0; 1]$, calculer $\int_a^1 f_n(t) dt$ à l'aide du changement de variable proposé, puis faire tendre a vers 0.
- 1.8 Utiliser les résultats des questions 1.5 et 1.7.
- 1.9 On pourra majorer le reste de la série à l'aide du reste d'une série géométrique.

Exercice 2

- 2.1.2 Se servir de la diagonalisabilité de f pour le second point.
- 2.1.5 Expliciter les sous-espaces $\text{Im}(p_j)$ et $\text{Ker}(p_i)$.
- 2.1.6 Pour $x \in E$, calculer $f(x)$ à l'aide du résultat de la question 2.1.4.
- 2.1.7 Utiliser les questions 2.1.3 et 2.1.4.
- 2.2.1 Se servir des résultats des questions 2.1.6, 2.1.5 et 2.1.7.
- 2.3.2 On peut trouver des vecteurs propres de la matrice A à la main, en effectuant des combinaisons simples des colonnes.
- 2.3.4 Utiliser le résultat de la question 2.1.6.
- 2.3.5 Se servir de la question 2.3.3.
- 2.3.7 Penser au résultat précédent et à l'expression matricielle du produit scalaire.
- 2.4 On commencera par déterminer M_1 à l'aide de la question 2.3.4.

Exercice 3

- 3.1 On pourra exprimer G_X à l'aide de la fonction $f : t \mapsto 1/(1-t)$.
- 3.4 Se servir de la définition des fonctions génératrices.
- 3.5 Utiliser l'indépendance de X et Y , ainsi que les résultats de la question 3.4.
- 3.6.3 Faire le lien avec les dérivées successives de G_Y en 1.

Exercice 4

- 4.3.2 Déterminer l'image par ψ de la base canonique $\mathcal{B}_0 = (1, X, \dots, X^n)$.
- 4.3.3 On pourra montrer que les deux propriétés sont équivalentes à $\psi(P)(1) = 0$.
- 4.3.4 Trouver des antécédents par ψ des polynômes qui apparaissent à la question 4.3.3, et utiliser les résultats des questions 4.3.3 et 4.2.2.
- 4.4.2 Utiliser la formule de Taylor en 0 pour exprimer tout polynôme à l'aide des ψ_k .
- 4.4.3 Se servir des calculs effectués à la question précédente.

EXERCICE 1

1.1 La fonction f_0 étant constante, elle est évidemment continue sur I . Pour les autres, on va d'abord justifier qu'elles sont continues sur l'intervalle $]0; 1]$ avant d'étudier leur continuité (à droite) en 0.

- La fonction f a pour expression sur $]0; 1]$

$$f(x) = x^{-x} = \exp(-x \ln x)$$

si bien qu'elle est continue sur cet intervalle en tant que composée de fonctions usuelles continues sur leurs ensembles de définition. Étudions sa continuité en 0 : on sait que $\lim_{x \rightarrow 0^+} x \ln x = 0$ d'après les croissances comparées, et par conséquent

$$\lim_{x \rightarrow 0^+} f(x) = \exp(0) = 1 = f(0)$$

par continuité de l'exponentielle. La fonction f est bien continue en 0, donc

La fonction f est continue sur I .

- Soit maintenant $n \in \mathbb{N}^*$. Comme f , la fonction f_n est composée de fonctions usuelles continues, donc elle est continue sur $]0; 1]$. De plus, $\lim_{x \rightarrow 0^+} x \ln x = 0$ d'où par produit de limites

$$\lim_{x \rightarrow 0^+} f_n(x) = 0 = f_n(0)$$

ce qui prouve que f_n est continue en 0. De ce fait,

Pour tout $n \in \mathbb{N}$, la fonction f_n est continue sur I .

La continuité de ces fonctions va permettre de définir leurs intégrales sur l'intervalle $[0; 1]$ en fin d'exercice.

1.2 Puisque $f_n(0) = 0$ pour tout $n \geq 1$, la série numérique $\sum f_n(0)$ converge et

$$\sum_{n=0}^{+\infty} f_n(0) = f_0(0) + \sum_{n=1}^{+\infty} f_n(0) = 1 = f(0)$$

Soit $x \in]0; 1]$. La fonction exponentielle est développable en série entière sur $\mathbb{R}$ et

$$\forall y \in \mathbb{R} \quad e^y = \sum_{n=0}^{+\infty} \frac{y^n}{n!}$$

En particulier, on obtient pour $y = -x \ln x$

$$e^{-x \ln x} = f(x) = \sum_{n=0}^{+\infty} \frac{(-x \ln x)^n}{n!} = \sum_{n=0}^{+\infty} \frac{(-1)^n}{n!} (x \ln x)^n = \sum_{n=0}^{+\infty} f_n(x)$$

Ceci prouve que

La série de fonctions $\sum f_n$ converge simplement sur I vers la fonction f .

On ne peut pas dire que ce soit une grosse surprise, étant donnée la façon dont sont introduites les fonctions f et $(f_n)_{n \in \mathbb{N}}$.

1.3 Puisque la fonction φ est continue sur I , on a nécessairement

$$\varphi(0) = \lim_{t \rightarrow 0^+} \varphi(t) = \lim_{t \rightarrow 0^+} t \ln t = 0$$

| Il est quand même étonnant que cette valeur n'apparaisse pas dans l'énoncé. En outre, φ est dérivable sur $]0; 1]$ en tant que produit de fonctions dérivables et

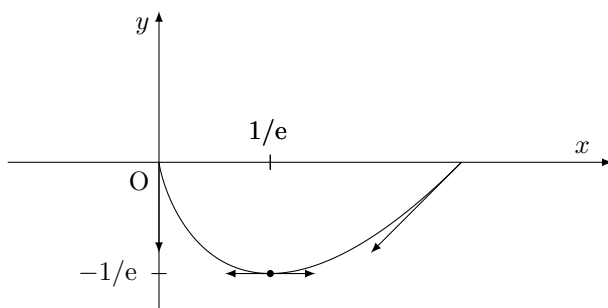
$$\forall t \in]0; 1] \quad \varphi'(t) = 1 \times \ln t + t \times \frac{1}{t} = \ln t + 1$$

Ainsi, $\forall t \in]0; 1] \quad \varphi'(t) > 0 \iff \ln t > -1 \iff t > 1/e$

par croissance de $\ln$, ce qui permet de dresser le tableau de variations de φ sur I :

t	0	$1/e$	1
$\varphi'(t)$		-	0
$\varphi(t)$	0	$\searrow$	$\nearrow$ 0
		$-1/e$	

1.4 Notons tout d'abord que $\lim_{t \rightarrow 0^+} \varphi'(t) = -\infty$, ce qui montre que la courbe de φ admet l'axe des ordonnées pour tangente au point d'abscisse 0. Par ailleurs, $\varphi'(1) = 1$ donc la tangente au point d'abscisse 1 a pour pente 1, et ainsi pour équation $y = x - 1$. Voici alors la courbe demandée :



1.5 On a $\|\varphi\|_\infty = 1/e$ d'après la question 1.3. Par définition, on a $f_n = (-1)^n \varphi^n / n!$ pour tout entier $n \in \mathbb{N}$ donc

$$\|f_n\|_\infty = \frac{(1/e)^n}{n!} = \frac{(e^{-1})^n}{n!}$$

Or, la série numérique $\sum (e^{-1})^n / n!$ converge puisque la fonction exponentielle est développable en série entière sur $\mathbb{R}$, si bien que

La série de fonctions $\sum f_n$ converge normalement sur I .

1.6.1 Soit $x \in \mathbb{R}$. Pour que le nombre $\Gamma(x)$ existe, il faut et il suffit que les intégrales données par $I_1 = \int_0^1 t^{x-1} e^{-t} dt$ et $I_2 = \int_1^{+\infty} t^{x-1} e^{-t} dt$ soient convergentes.

- Notons tout d'abord que I_1 est impropre en 0 et que l'on a $t^{x-1} e^{-t} \underset{t \rightarrow 0}{\sim} t^{x-1}$.

D'après le critère de Riemann, $\int_0^1 t^{x-1} dt$ converge si, et seulement si, $1 - x < 1$, ce qui est équivalent à $x > 0$. On en déduit, par comparaison de fonctions positives, que l'intégrale I_1 converge si, et seulement si, $x > 0$.

CCINP Maths 1 MP 2021 — Corrigé

Ce corrigé est proposé par Bertrand Wiel (professeur en CPGE) ; il a été relu par Rémi Pellerin (ENS Lyon) et Céline Chevalier (enseignant-chercheur à l'université).

L'épreuve est constituée de deux exercices d'analyse, proches dans la forme de ceux de la banque d'épreuves orales du concours, et d'un problème en trois parties autour de la fonction zêta de Riemann faisant intervenir des questions d'informatique, d'analyse, d'arithmétique et de probabilités.

- Le premier exercice s'intéresse au calcul d'une intégrale dont on montre qu'elle est égale à la somme d'une série. Il ne contient que deux questions mais fait appel à de nombreux résultats du cours sur les intégrales généralisées. Pour réussir cet exercice, il fallait bien maîtriser la méthode d'inversion série-intégrale et appliquer rigoureusement le théorème d'intégration terme à terme.
- Le second exercice, moins technique, demande d'établir par un argument classique de concavité l'inégalité arithmético-géométrique pour 3 réels strictement positifs afin d'établir ensuite qu'une certaine fonction à deux variables réelles admet un extremum local sur un ouvert. On recherche d'abord les éventuels points critiques de la fonction pour déterminer où l'extremum pourrait être atteint.
- La première partie du problème porte sur le programme de tronc commun d'informatique, elle est indépendante des suivantes. L'objectif est d'écrire une fonction itérative calculant les nombres de Bernoulli, après avoir écrit des fonctions auxiliaires pour calculer la factorielle et des coefficients binomiaux, cette dernière sous forme récursive. Les questions d'efficacité associées à l'utilisation de fonctions récursives ne sont pas soulevées. Seule la capacité à écrire une fonction simple en Python est évaluée. La dernière fonction demande de mettre en œuvre le principe de mémorisation.
- Dans la deuxième partie, on établit quelques propriétés de la fonction zêta à l'aide du cours sur les séries de fonctions et les familles sommables. Comme le premier exercice, elle exigeait de maîtriser les résultats du cours d'analyse de seconde année et les exercices classiques associés.
- La dernière partie démontre la formule du produit eulérien liant la fonction zêta à l'ensemble des nombres premiers, en utilisant le formalisme du cours de probabilités. Elle exploite des résultats de première année en arithmétique et utilise la notion de famille d'événements mutuellement indépendants.

Si le sujet comporte un nombre de questions plutôt faible, certaines d'entre elles nécessitent une résolution en plusieurs étapes. L'ensemble est conforme aux sujets récents de la filière MP du concours CCINP, c'est-à-dire d'une longueur raisonnable, proche du cours et de ses applications classiques. C'est un bon sujet d'entraînement pour l'écrit de ce concours mais aussi pour l'épreuve orale des concours Mines-Ponts et Centrale-Supélec.

INDICATIONS

Exercice I

- 1 Intégrer par parties.
- 2 Appliquer le théorème d'intégration terme à terme des séries de fonctions.

Exercice II

- 4 Montrer que la fonction f admet un minimum global en son unique point critique à l'aide de l'inégalité de la question 3.

Problème

- 7 Identifier les cas de base et le cas récursif pour écrire la fonction récursive.
- 8 Calculer les nombres de Bernoulli de b_0 à b_n (de bas en haut) en les stockant dans une liste.
- 10 Appliquer le théorème de dérivation terme à terme à la série de fonctions $\sum_{n \geq 1} f_n$.
- 11 Raisonner par l'absurde et appliquer le théorème de la double limite en 1.
- 12 Appliquer le théorème de la double limite en $+\infty$.
- 13 Procéder par comparaison série-intégrale. Pour $x > 1$, calculer la valeur de $I(x)$ pour établir l'équivalent.
- 14 Montrer que $A = \bigcup_{n \in \mathbb{N}^*} A_n$ puis justifier que

$$\forall x > 1 \quad \sum_{(a,b) \in A_n} \frac{1}{(ab)^x} = \frac{d_n}{n^x}$$

Conclure grâce à la réciproque du théorème de sommation par paquets.

- 16 Montrer l'implication directe par récurrence en établissant à l'aide du lemme de Gauss que le produit de deux diviseurs de N premiers entre eux est un diviseur de N .
- 17 Utiliser la question 16 pour montrer que l'intersection des éléments d'une sous-famille de $([X \in a_k \mathbb{N}^*])_{1 \leq k \leq n}$ est de la forme $[X \in b_1 b_2 \dots b_r \mathbb{N}^*]$ puis utiliser le résultat de la question 15.
- 18 Exprimer B_n comme l'intersection d'événements mutuellement indépendants.
- 19 Appliquer le théorème de continuité décroissante à la suite $(B_n)_{n \in \mathbb{N}^*}$.
- 20 Montrer que la suite $(\ln u_n)_{n \in \mathbb{N}^*}$ et la série $\sum 1/p_n$ sont de même nature. Utiliser l'équivalent de la question 13 pour établir la contradiction.

EXERCICE I

1 Notons f_k la fonction $t \mapsto t^{2k} \ln t$, définie sur $]0; 1]$. La fonction f_k est continue en tant que produit de fonctions continues et $\sqrt{t}f_k(t) \xrightarrow[t \rightarrow 0^+]{} 0$ par croissances comparées, ainsi

$$f_k(t) \underset{t \rightarrow 0^+}{=} o\left(\frac{1}{\sqrt{t}}\right)$$

Le critère de comparaison aux intégrales de Riemann établit que la fonction f_k est intégrable sur $]0; 1]$, donc

$$\boxed{\text{L'intégrale } I_k = \int_0^1 t^{2k} \ln t \, dt \text{ existe.}}$$

Ici la fonction de comparaison $t \mapsto 1/\sqrt{t}$ est intégrable, il n'est ainsi pas nécessaire d'utiliser le critère de comparaison des fonctions positives en écrivant

$$|f_k(t)| \underset{t \rightarrow 0^+}{=} o\left(\frac{1}{\sqrt{t}}\right)$$

qui d'ailleurs est strictement équivalent à l'écriture sans valeurs absolues puisque la définition de la relation de négligeabilité porte sur les valeurs absolues des fonctions. Le critère de comparaison aux intégrales de Riemann correspond à cette situation.

Par croissances comparées, on a la limite $t^{2k+1} \ln t \xrightarrow[t \rightarrow 0^+]{} 0$. Intégrons par parties, les fonctions $t \mapsto \ln t$ et $t \mapsto t^{2k+1}/(2k+1)$ étant de classe $\mathcal{C}^1$ sur $]0; 1]$,

$$\begin{aligned} I_k &= \int_0^1 t^{2k} \ln t \, dt \\ &= \left[\frac{t^{2k+1}}{2k+1} \ln t \right]_0^1 - \int_0^1 \frac{t^{2k+1}}{(2k+1)t} \, dt \\ &= 0 - \frac{1}{2k+1} \int_0^1 t^{2k} \, dt \end{aligned}$$

d'où

$$\boxed{I_k = \frac{-1}{(2k+1)^2}}$$

On peut appliquer le théorème d'intégration par parties aux intégrales sur un intervalle quelconque lorsque l'existence de deux des limites parmi les trois qu'elle décrit (les deux intégrales, le crochet) a été établie, la troisième limite se déduisant alors du théorème de limite d'une somme. On peut également intégrer par parties sur un segment $[\varepsilon; 1]$, où $\varepsilon \in]0; 1[$, puis justifier le passage à la limite quand $\varepsilon \rightarrow 0$.

2 La fonction f est continue sur $]0; 1[$ et

$$\sqrt{t}f(t) = \sqrt{t} \frac{\ln t}{t^2 - 1} \xrightarrow[t \rightarrow 0^+]{} 0$$

d'où

$$f(t) \underset{t \rightarrow 0^+}{=} o\left(\frac{1}{\sqrt{t}}\right)$$

Par comparaison aux intégrales de Riemann, la fonction f est donc intégrable au voisinage de 0.

Utilisons l'équivalent $\ln t \underset{t \rightarrow 1}{\sim} t - 1$ pour calculer la limite

$$f(t) = \frac{1}{t+1} \frac{\ln t}{t-1} \xrightarrow{t \rightarrow 1^-} \frac{1}{2}$$

La fonction f est donc prolongeable par continuité en 1, puis intégrable au voisinage de 1. On en déduit que

La fonction f est intégrable sur $]0; 1[$.

Appliquons le résultat du cours sur le développement en série entière sur $] -1; 1[$ de la fonction $u \mapsto 1/(1-u)$. On a

$$\forall t \in]0; 1[\quad f(t) = -\frac{\ln t}{1-t^2} = -\ln t \sum_{k=0}^{+\infty} (t^2)^k = -\sum_{k=0}^{+\infty} t^{2k} \ln t = -\sum_{k=0}^{+\infty} f_k(t)$$

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

- D'après la question 1, $(f_k)_{k \in \mathbb{N}}$ est une suite de fonctions continues intégrables sur $]0; 1[$.
- La série $\sum f_k$ converge simplement vers la fonction $(-f)$ continue sur $]0; 1[$.
- Pour tout $k \in \mathbb{N}$, la fonction $|f_k| = -f_k$ est intégrable sur $]0; 1[$ d'après la question 1. De plus,

$$\int_0^1 |f_k(t)| dt = \frac{1}{(2k+1)^2}$$

d'où, par comparaison avec une série de Riemann, la série $\sum \int_0^1 |f_k(t)| dt$ est convergente.

On en déduit, par application du théorème d'intégration terme à terme :

$$\int_0^1 (-f)(t) dt = \int_0^1 \sum_{k=0}^{+\infty} f_k(t) dt = \sum_{k=0}^{+\infty} \int_0^1 f_k(t) dt = \sum_{k=0}^{+\infty} I_k = \sum_{k=0}^{+\infty} \frac{-1}{(2k+1)^2}$$

Soit $p \in \mathbb{N}$, séparons les termes d'indices pairs et ceux d'indices impairs, on a :

$$\sum_{k=0}^p \frac{1}{(2k+1)^2} = \sum_{n=1}^{2p+1} \frac{1}{n^2} - \sum_{k=1}^p \frac{1}{(2k)^2}$$

Les séries étant convergentes, passons à la limite en utilisant la valeur rappelée dans l'énoncé,

$$\int_0^1 f(t) dt = \sum_{k=0}^{+\infty} \frac{1}{(2k+1)^2} = \sum_{n=1}^{+\infty} \frac{1}{n^2} - \sum_{k=1}^{+\infty} \frac{1}{(2k)^2} = \frac{\pi^2}{6} - \frac{1}{4} \frac{\pi^2}{6} = \frac{3}{4} \frac{\pi^2}{6}$$

d'où, $\int_0^1 f(t) dt = \frac{\pi^2}{8}$

L'énoncé de la question demande de justifier que f est intégrable puis de démontrer l'expression de l'intégrale comme somme d'une série. La réponse qui est proposée respecte cet ordre, en établissant dans un premier temps l'intégrabilité puis en calculant l'intégrale. Ce n'était pas nécessaire. En effet, le théorème d'intégration terme à terme ne s'appuie pas sur le fait que la fonction f soit intégrable, au contraire il en fournit une preuve. On aurait donc pu directement appliquer le théorème, les réponses attendues en étant, toutes deux, les conséquences.

CCINP Maths 2 MP 2021 — Corrigé

Ce corrigé est proposé par Jean-Paul Bonnet (professeur en CPGE) ; il a été relu par Christophe Fiszka (professeur en CPGE) et Tristan Poullaouec (professeur en CPGE).

Ce sujet traite de la réduction des endomorphismes et plus précisément de la décomposition de Dunford, c'est-à-dire l'écriture de tout endomorphisme u , scindé, sous la forme $u = d + n$ avec d diagonalisable et n nilpotente. Ce théorème est admis. Le sujet est composé d'un très court exercice et d'un problème en quatre parties. Ce dernier illustre sur des exemples cette décomposition, en donne des applications, puis en établit l'unicité.

- Dans l'exercice, on détermine l'orthogonal de l'espace vectoriel des matrices diagonales. C'est la seule question en lien avec les espaces euclidiens.
- Dans la partie I du problème, on nous propose d'étudier des exemples de la réduction de Dunford en dimensions 2 et 3, ainsi qu'une application de cette décomposition au calcul d'une exponentielle de matrice.
- Dans la partie suivante, deux preuves de l'existence de la décomposition de Dunford sont illustrées sur un exemple.
- Dans la partie III, on démontre l'unicité de la décomposition de Dunford. C'est l'occasion de démontrer des résultats classiques sur les endomorphismes nilpotents et sur la codiagonalisabilité des endomorphismes.
- Dans la dernière partie, on établit que l'application qui, à une matrice complexe, associe la partie diagonalisable de sa décomposition, n'est pas continue. Pour cela, on démontre et exploite la densité dans $\mathcal{M}_n(\mathbb{C})$ de l'ensemble des matrices complexes diagonalisables.

Pour traiter ce sujet, il fallait maîtriser les fondamentaux de l'algèbre linéaire, les résultats et les techniques portant sur la réduction, ainsi que les principales propriétés des endomorphismes nilpotents. Certaines questions font appel à des résultats de topologie des espaces vectoriels normés de dimension finie. Une question traite d'orthogonalité.

Si le sujet n'est pas original, il constitue un incontournable de la préparation et un excellent moyen de tester sa connaissance du cours et de ses applications directes. Il est par ailleurs relativement progressif et plutôt court, ce qui permet de le terminer dans le temps imparti.

INDICATIONS

- 1 Trouver une base $\mathcal{B}$ de $\mathcal{D}_n(\mathbb{R})$. Une matrice appartient à $\mathcal{D}_n(\mathbb{R})^\perp$ si, et seulement si, elle est orthogonale à toute matrice de $\mathcal{B}$.

Partie I

- 2 Penser à l'unicité du couple, la caractérisation d'une matrice trigonalisable et à l'hypothèse de commutativité des éléments du couple.
- 3 Choisir une matrice dont le polynôme caractéristique est irréductible dans $\mathbb{R}$.
- 4 Remarquer qu'il n'existe qu'une seule matrice diagonalisable D admettant χ_A comme polynôme caractéristique puis considérer $N = A - D$.
- 5 Remarquer que la série $\exp(N)$ ne comporte que deux termes non nuls.
- 6 Vérifier que le couple proposé a les bonnes propriétés. Conclure par unicité.

Partie II

- 7 Remarquer que le polynôme minimal est donné dans la question.
- 9 Séparer la matrice B en sa partie diagonale et sa partie triangulaire supérieure stricte. Utiliser les formules de changement de base.
- 10 Multiplier la décomposition en éléments simples par $(X - 1)(X - 2)^2$ pour obtenir la relation de Bézout demandée.
- 12 Utiliser la question 11 qui permet d'exprimer p et q comme polynômes en u .

Partie III

- 14 Montrer que $A - B$ est semblable à une matrice diagonale.
- 15 Utiliser astucieusement la formule du binôme de Newton.
- 17 Combiner les réponses des questions 14, 15 et 16.

Partie IV

- 18 Montrer que $\mathcal{D}$ n'est pas stable par addition. Une application linéaire en dimension finie est continue.
- 20 Utiliser les questions 2, 18 et 19 et procéder par l'absurde.

Exercice

1 Soit $i \in \llbracket 1; n \rrbracket$; on pose $E_{i,i}$ la matrice dont tous les coefficients sont nuls, sauf celui à l'intersection de la i -ième ligne et de la i -ième colonne qui vaut 1. Il est clair que $\mathcal{D}_n(\mathbb{R}) = \text{Vect} \{E_{j,j} \mid j \in \llbracket 1; n \rrbracket\}$ d'où

$$\mathcal{D}_n(\mathbb{R})^\perp = \{E_{i,i} \mid i \in \llbracket 1; n \rrbracket\}^\perp = \bigcap_{i=1}^n E_{i,i}^\perp$$

Soit $A = (a_{i,j})_{1 \leq i,j \leq n}$ une matrice de $\mathcal{M}_n(\mathbb{R})$ et $i \in \llbracket 1; n \rrbracket$, alors

$${}^t A E_{i,i} = \begin{pmatrix} 0 & \cdots & 0 & a_{i,1} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{i,n} & 0 & \cdots & 0 \end{pmatrix}$$

Par suite, $\langle A \mid E_{i,i} \rangle = a_{i,i}$, d'où

$$A \in E_{i,i}^\perp \iff \langle A \mid E_{i,i} \rangle = 0 \iff a_{i,i} = 0$$

Ainsi, une matrice A appartient à l'orthogonal de $\mathcal{D}_n(\mathbb{R})$ si, et seulement si, A possède une diagonale nulle. En conclusion,

$$\mathcal{D}_n(\mathbb{R})^\perp = \{(a_{i,j})_{1 \leq i,j \leq n} \in \mathcal{M}_n(\mathbb{R}) \mid \forall i \in \llbracket 1; n \rrbracket \quad a_{i,i} = 0\}$$

On notera que $\mathcal{D}_n(\mathbb{R})^\perp$ n'est pas égal au sous-espace vectoriel constitué des matrices de trace nulle. Il en est un sous-espace vectoriel strict.

Problème

I. QUELQUES EXEMPLES

2 On désigne par 0_n la matrice nulle de $\mathcal{M}_n(\mathbb{K})$. Si A est diagonalisable alors le couple $(A, 0_n)$ convient car

- (1) $A = A + 0_n$;
- (2) A est diagonalisable dans $\mathcal{M}_n(\mathbb{K})$;
- (3) 0_n est nilpotente;
- (4) $A \times 0_n = 0_n = 0_n \times A$;

donc les quatre propriétés sont vérifiées et, par unicité, c'est le seul couple possible.

La décomposition de Dunford d'une matrice A diagonalisable est $(A, 0_n)$.

En appliquant le même raisonnement et toujours par unicité du couple,

La décomposition de Dunford d'une matrice A nilpotente est $(0_n, A)$.

D'après le cours, une matrice est trigonalisable si, et seulement si, son polynôme caractéristique est scindé, ce qui est l'hypothèse du théorème de décomposition de Dunford. Par suite,

Une matrice trigonalisable admet une décomposition de Dunford.

Enfin, le couple proposé n'est pas la décomposition de Dunford de la matrice car il ne vérifie pas la quatrième propriété. En effet, d'une part

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

D'autre part,

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}$$

Le couple proposé n'est pas la décomposition de Dunford de la matrice $\begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$.

On pouvait aussi noter que, comme la matrice proposée A est triangulaire supérieure, 1 et 2 sont ses valeurs propres. Par suite, elle est diagonalisable donc sa décomposition de Dunford est le couple $(A, 0_2)$. Par unicité de la décomposition, la décomposition proposée n'est pas celle de Dunford.

3 Il suffit de contredire l'hypothèse du théorème, c'est-à-dire proposer une matrice dont le polynôme caractéristique n'est pas scindé. Par exemple, la matrice

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

admet $X^2 + 1$ comme polynôme caractéristique, qui est irréductible dans $\mathbb{R}$. Ainsi,

La matrice $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ n'admet pas de décomposition de Dunford dans $\mathcal{M}_2(\mathbb{R})$.

Plus généralement, les matrices de rotation d'un angle $\theta \neq k\pi$ ($k \in \mathbb{Z}$) fournissent également des contre-exemples.

4 En développant le déterminant par rapport à la colonne du milieu, on obtient

$$\chi_A(X) = \begin{vmatrix} X-3 & 0 & -8 \\ -3 & X+1 & -6 \\ 2 & 0 & X+5 \end{vmatrix} = (X+1) \begin{vmatrix} X-3 & -8 \\ 2 & X+5 \end{vmatrix} = (X+1)^3$$

Ainsi,

Le polynôme caractéristique est $\chi_A(X) = (X+1)^3$.

Le polynôme caractéristique de A est scindé donc la décomposition de Dunford de A , notée (D, N) , existe. Comme $\chi_D = \chi_A$ la matrice D recherchée admet -1 comme unique valeur propre. Or, toute matrice diagonalisable admettant une unique valeur propre est nécessairement une matrice scalaire. En conséquence, $D = -I_3$. On pose alors

$$N = A - D = \begin{pmatrix} 4 & 0 & 8 \\ 3 & 0 & 6 \\ -2 & 0 & -4 \end{pmatrix}$$

et on vérifie facilement que $N^2 = 0_3$. Finalement,

Le couple (D, N) est la décomposition de Dunford de A .

On note que D et N sont bien des polynômes en A .

CCINP Informatique optionnelle MP 2021

Corrigé

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

Cette épreuve est constituée de trois exercices indépendants touchant chacun à trois thématiques différentes du programme de prépa.

- Le premier exercice s'intéresse à la notion de partition emboîtée d'un sous-ensemble d'entiers. Après quelques questions élémentaires servant uniquement à vérifier la compréhension des définitions données, on utilise le langage Python pour implémenter des fonctions permettant de vérifier qu'une liste de listes représente dans un premier temps une partition d'un ensemble, puis une partition emboîtée. L'exercice se termine par des questions relatives au dénombrement de ces partitions.
- Le deuxième exercice porte sur la logique et la satisfiabilité. On définit une classe particulière de formules sous forme normale conjonctive qui porte le nom de formules de Horn. À l'aide d'une règle de simplification portant le nom de propagation unitaire, on met en place une méthode permettant de décider de manière efficace si une telle formule est satisfiable.
- Le dernier exercice a pour objet l'étude des classes sylvestres d'un arbre binaire de recherche. On définit dans un premier temps l'arbre binaire de recherche associé à un mot w , noté $\circ \leftarrow w$, comme l'arbre obtenu par insertion successive des lettres de w dans un arbre initialement vide. La classe sylvestre d'un arbre T est alors l'image réciproque de $\{T\}$ par l'application $w \mapsto (\circ \leftarrow w)$. Après l'écriture de quelques fonctions élémentaires OCaml, l'énoncé s'attache à donner une caractérisation simple, via une relation d'équivalence, de l'appartenance de deux mots à une même classe sylvestre. Il s'achève sur une nouvelle salve d'écriture de fonctions dont l'objectif est de calculer la classe d'un arbre donné en argument.

Il s'agit d'un excellent problème de révision pour les concours. Les trois exercices traitent de problématiques variées et le sujet est très bien écrit. La dernière partie est assez technique mais la décomposition en de nombreuses sous-questions la rend très abordable. Le seul reproche que l'on pourrait faire à cette épreuve porte sur son aspect très théorique, et son faible nombre de questions de programmation, sachant de plus que l'énoncé facilite tellement le travail que les codes à écrire font en moyenne cinq lignes.

INDICATIONS

Partie I

- 3 Pour gagner du temps, chercher les partitions qui ne sont pas non croisées (elles ne sont pas nombreuses).
- 4 La fonction **mystere** détermine si la liste donnée en argument représente une partition de l'ensemble des valeurs de **A**.
- 5 Chercher l'élément **i** à l'intérieur de chaque liste contenue dans **P**.
- 6 À l'intérieur de la quadruple boucle, il ne reste qu'à vérifier si le quadruplet (i, j, k, l) contredit la condition **non croisée**.
- 7 Construire une bijection entre les ensembles $\text{NC}(\llbracket n \rrbracket)$ et $\text{NCE}(\llbracket n + 1 \rrbracket)$.
- 8 Deux ensembles finis en bijection ont même cardinal et le cardinal d'une union disjointe (respectivement d'un produit cartésien) d'ensembles est la somme (respectivement le produit) des cardinaux des ensembles.
- 9 Calculer les valeurs $C_0, \dots, C_n$ en utilisant la programmation dynamique.

Partie II

- 12 On doit aboutir à $\Pi(P) = (x_3 \vee x_4)$.
- 13 Montrer que la propriété « **P** est une formule de Horn » est invariante par application de la règle de propagation unitaire.
- 14 Montrer que la propriété « **P** est satisfiable » est invariante par application de la règle de propagation unitaire.
- 15 On rappelle que $\llbracket () \rrbracket_I = F$ quelle que soit l'interprétation **I**.
- 16 La question est mal posée donc triviale. Montrer plutôt qu'on peut satisfaire **C** à l'aide d'une interprétation indépendante de **C**.
- 17 Utiliser l'indication de la question précédente.
- 18 Raisonner par double implication en utilisant l'équivalence de la question 14 et les résultats des questions 15 (par contraposée) et 17.

Partie III

- 19 Dessiner l'arbre au fur et à mesure au brouillon.
- 22 On doit retrouver un mot qui apparaît un peu partout dans cette partie.
- 23 Donner une implémentation récursive utilisant les opérateurs **::** et **@**.
- 24 Il s'agit de justifier que pour tout arbre **T**, on a $\circ \leftarrow w_T = T$. On pourra raisonner par récurrence sur la hauteur ou le nombre de nœuds de l'arbre.
- 25 Montrer que la relation est réflexive, transitive et symétrique. On remarquera que la relation « être S-adjacent à » est symétrique par définition.
- 26 Raisonner par récurrence sur le nombre de nœuds de l'arbre. Pour le cas général, on distinguera trois cas suivant la position relative de la racine *r* de l'arbre par rapport à *a* et *c*.
- 27 Justifier la propriété pour deux mots S-adjacents à l'aide de la question précédente.
- 28a Remarquer que deux mots S-adjacents ont la même longueur.

28b Remarquer que deux mots S-adjacents conservent la même première lettre.

28c Étant donnés $1 \leq i < j \leq n - 1$ tels que $a_j < r \leq a_i$, raisonner sur

$$X = \{s \in \llbracket i; j \rrbracket \mid a_s \geq r\}$$

28d Considérer cette fois $X = \{s \in \llbracket 1; n - 1 \rrbracket \mid a_s \geq r\}$

28e C'est la question la plus délicate du sujet. Supposer $N(a) \neq \emptyset$ et montrer qu'en permutant les lettres a_k et a_{k+1} où $(k, k + 1)$ est le couple de la question 28c, alors le cardinal de $N(a)$ diminue exactement d'une unité.

29 À ce stade du sujet, on pourra utiliser sans justification le fait que si u_1, u_2, v_1, v_2 sont quatre mots tels que u_1 et u_2 sont S-équivalents, de même que v_1 et v_2 , alors $u_1 v_1$ et $u_2 v_2$ le sont.

30 Tout a été fait aux questions 27 et 29.

31 Mélanger deux mots u et v revient à construire tous les mots w qui contiennent toutes les lettres de u et toutes celles de v (comptées avec multiplicités) en conservant l'ordre des lettres de u et des lettres de v .

33 Implémenter sous la forme d'une fonction récursive la définition 21 de l'énoncé.

34 Remarquer que pour tous ensembles L', M' et tous mots u et v

$$(\{u\} \cup L') \sqcup (\{v\} \cup M') = (u \sqcup v) \cup (\{u\} \sqcup M') \cup (L' \sqcup M)$$

et traduire cette égalité en une fonction récursive.

35 Implémenter sous la forme d'une fonction récursive le résultat admis après la définition 22 de l'énoncé.

I. ÉTUDE DES PARTITIONS NON CROISÉES

1 Pour la partition $\mathcal{P}$, les couples d'entiers distincts appartenant à une même classe sont $(1, 7), (3, 4), (3, 5)$ et $(4, 5)$. Si $a < b < c < d$ vérifie $\text{Cl}(a) = \text{Cl}(c)$ et $\text{Cl}(b) = \text{Cl}(d)$, alors nécessairement $a = 1$, puis $c = 7$. Mais dans ce cas, $\{b, d\}$ est inclus dans $\{3, 4, 5\}$ et $d < c$. Par conséquent,

La partition $\mathcal{P}$ est non croisée.

Pour la partition $\mathcal{Q}$, on peut remarquer que

$1 < 5 < 6 < 7$ avec $\text{Cl}(1) = \text{Cl}(6)$ et $\text{Cl}(5) = \text{Cl}(7)$

ce qui prouve que

La partition $\mathcal{Q}$ n'est pas une partition non croisée.

2 Parmi les quatre exemples proposés, on distingue 2 partitions, dont une seule est non croisée. Plus précisément,

- $\mathcal{P}_1$ est une partition qui n'est pas non croisée.
- $\mathcal{P}_2$ n'est pas une partition.
- $\mathcal{P}_3$ n'est pas une partition.
- $\mathcal{P}_4$ est une partition non croisée.

Si l'énoncé ne demande pas de justification, il ne faut pas chercher à en donner le jour du concours. On n'y gagnera pas grand chose mis à part un peu de bienveillance du correcteur (mais sans garantie), et on risque surtout de perdre du temps. Dans un souci pédagogique, on peut toutefois préciser dans ce corrigé que

- $\mathcal{P}_1$ n'est pas non croisée car $1 < 2 < 3 < 4$ avec 1 et 3 (respectivement 2 et 4) dans la même classe, mais pas dans une classe commune.
- $\mathcal{P}_2$ n'est pas une partition car les parties $\{1, 3\}$ et $\{1, 2\}$ ne sont pas disjointes.
- $\mathcal{P}_3$ n'est pas une partition car $\{1, 3\} \cup \{2, 4\}$ n'est pas égal à $\{1, 2, 3, 4, 5\}$.

3 Une partition $\mathcal{P}$ de $\llbracket 4 \rrbracket$ n'est pas non croisée par définition s'il existe $a < b < c < d$ quatre entiers tels que a et c (respectivement b et d) sont dans une même classe C_1 (respectivement C_2), mais avec $C_1 \neq C_2$. Puisque $\llbracket 4 \rrbracket$ ne contient que 4 éléments, cela impose $(a, b, c, d) = (1, 2, 3, 4)$ et $\mathcal{P} = \{\{1, 3\}, \{2, 4\}\}$. Il n'y a donc qu'une seule partition de cette nature.

Toutes les partitions de $\llbracket 4 \rrbracket$ sont non croisées, à l'exception de $\{\{1, 3\}, \{2, 4\}\}$.

L'énoncé ne précise pas très clairement s'il attend que l'on dresse la liste des partitions non croisées de $\llbracket 4 \rrbracket$. Pour ce faire, on peut ordonner les partitions de cet ensemble en fonction de leur cardinal :

- Il y a une seule partition à une seule partie, c'est $\{\{1, 2, 3, 4\}\}$.
- On distingue parmi les partitions à deux parties celles constituées de deux parties de cardinal 2 et celles constituées de deux parties de cardinal 3 et 1.

Centrale Maths 1 MP 2021 — Corrigé

Ce corrigé est proposé par Angèle Niclas (ENS Lyon) ; il a été relu par Tristan Poullaouec (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

L'objet de ce problème est l'étude de la répartition des valeurs propres des matrices symétriques réelles aléatoires de grande taille et la preuve de la loi du demi-cercle. Cette loi affirme que pour toute matrice symétrique réelle constituée de variables aléatoires indépendantes identiquement distribuées, d'espérance nulle et de variance égale à 1, les valeurs propres normalisées par un facteur $1/\sqrt{n}$ se répartissent suivant un profil en demi-cercle de rayon 2.

Le sujet est composé de quatre parties. Les deux premières sont indépendantes et s'attachent à montrer des résultats intermédiaires utiles dans les deux dernières parties.

- Dans la partie I, on montre l'inégalité de Hoffman-Wielandt liant l'écart entre les valeurs propres de deux matrices symétriques et la norme de la différence des deux matrices. Pour cela, on est amené à minimiser une fonction sur l'ensemble des matrices bistochastiques. Cette partie utilise les chapitres sur la réduction des matrices et sur les espaces vectoriels normés.
- La partie II porte sur le dénombrement des mots bien parenthésés pour retrouver les nombres de Catalan. Si le début fait appel à quelques techniques classiques de dénombrement, il faut surtout maîtriser le cours sur les séries entières.
- Dans la partie suivante, on montre la loi du demi-cercle dans le cas simplifié où les variables sont uniformément bornées. On commence par quelques calculs d'intégrales puis on fait des dénombrements plus complexes que dans la partie II. Après avoir prouvé la loi du demi-cercle pour les polynômes, on l'étend aux fonctions bornées. Cette partie est assez longue et demande une bonne compréhension de la démarche car les questions s'appuient les unes sur les autres.
- La partie IV étend les résultats de la partie III au cas général. Pour cela, on introduit de nouvelles variables tronquées qui vérifient les hypothèses de la partie III, et avec les résultats de la partie I on prouve finalement la loi du demi-cercle dans le cas général.

Ce problème est long, assez difficile pour un sujet de la banque Centrale-Supélec et a dû déconcerter beaucoup de candidats par son originalité. La partie II est une bonne révision sur les séries entières, tandis que les parties II et III permettent de travailler le dénombrement. Le sujet dans sa globalité fait appel à de nombreux chapitres du programme de deuxième année. Pour le réussir, il fallait prendre le temps de comprendre le chemin suivi par l'énoncé pour démontrer la loi du demi-cercle.

INDICATIONS

Partie I

- 2 Appliquer le théorème spectral à A et B, puis utiliser le résultat de la question 1.
- 4 Montrer que $\mathcal{B}_n(\mathbb{R})$ est un compact de $\mathcal{M}_n(\mathbb{R})$. On pourra utiliser la caractérisation séquentielle des fermés puis majorer $\|M\|_F$ pour tout $M \in \mathcal{B}_n(\mathbb{R})$.
- 6 Construire la matrice recherchée par récurrence en annulant un par un les coefficients $m_{i,j}$ et $m_{j,i}$ pour $j \in \llbracket i+1; n \rrbracket$. Pour cela, on pourra se servir de la matrice définie en question 5.
- 7 Pour tout $M \in \mathcal{B}_n(\mathbb{R})$, utiliser la question 6 pour construire par récurrence une suite de matrices $M^{(k)}$ se rapprochant de I_n et vérifiant $f(M^{(k)}) \leq f(M)$.
- 8 Utiliser le résultat des questions 2 et 3 puis montrer que $M = (p_{i,j}^2)_{1 \leq i,j \leq n}$ est une matrice de $\mathcal{B}_n(\mathbb{R})$ pour appliquer le résultat de la question 7.

Partie II

- 10 Utiliser le lemme d'Abel pour minorer le rayon de convergence de la série entière $\sum C_k x^k$.
- 12 Utiliser le résultat de la question 11 et la formule du produit de Cauchy pour développer $(F(x))^2$ en série entière.
- 13 Dériver la relation de la question 12 puis raisonner par l'absurde.
- 14 Utiliser la relation de la question 12 pour reconnaître une équation du second degré en $F(x)$, puis le résultat de la question 13 pour choisir la racine réelle à conserver.
- 15 Utiliser le développement en série entière de $(1+x)^\alpha$ puis lier le produit des nombres impairs au produit des nombres pairs pour faire apparaître les factorielles demandées.
- 16 Développer en série entière chaque membre du résultat de la question 14 en utilisant le développement en série entière de la question 15.

Partie III

- 17 Remarquer que la fonction intégrée est impaire.
- 19 Intégrer la fonction $x \mapsto x\sqrt{4-x^2}$ et dériver la fonction $x \mapsto x^{2k+1}$ pour obtenir une relation liant m_{2k+2} et m_{2k} .
- 20 Utiliser le résultat des questions 18 et 19 pour exprimer m_{2k} en fonction de k puis utiliser le résultat des questions 16 et 17.
- 21 Calculer par récurrence le coefficient $(M_n^k)_{i,j}$ pour tout $(i,j) \in \llbracket 1; n \rrbracket^2$.
- 23 Appliquer le résultat de la question 22 à $\ell = (k+1)/2$.
- 24 Utiliser l'indépendance mutuelle des $(X_{ij})_{1 \leq i \leq j \leq n}$ et le fait que $\mathbb{E}(X_{ij}) = 0$ pour tout $(i,j) \in \llbracket 1; n \rrbracket^2$.
- 25 Remarquer que tout cycle comportant ℓ sommets comporte au moins $\ell - 1$ arêtes distinctes.
- 26 Utiliser le résultat de la question 21 puis séparer la somme entre $\mathcal{A}_k$ et $\mathcal{C}_k$. Utiliser le résultat de la question 24 pour traiter la somme sur $\mathcal{A}_k$ et le résultat des questions 23 et 25 pour la somme sur $\mathcal{C}_k$.

- 28 Étant donné un mot bien parenthésé, remarquer que tout cycle correspondant est associé à une unique liste ordonnée des arêtes distinctes, puis que toute liste ordonnée des arêtes distinctes est associée à une unique liste ordonnée des sommets distincts.
- 29 Réutiliser le raisonnement de la question 26 et séparer les cycles de $\mathcal{B}_k$ selon leur nombre de sommets distincts ℓ . Utiliser le résultat de la question 23 si $\ell < k/2 + 1$ et le résultat de la question 28 si $\ell = k/2 + 1$.
- 30 Utiliser le résultat des questions 20, 26 et 29.
- 32 Appliquer le résultat des questions 29 et 31 puis le résultat de la question 10 pour estimer C_{p+q}/A^{p+2q} quand $q \rightarrow +\infty$.
- 34 Utiliser le résultat des questions 32 et 33.
- 35 Approcher uniformément sur un intervalle $[-A; A]$ la fonction f par un polynôme, puis utiliser le résultat des questions 30 et 34.

Partie IV

- 36 Utiliser la caractérisation séquentielle de la limite et le caractère sommable de la famille $(x\mathbb{P}(X = x))_{x \in X(\Omega)}$.
- 37 Appliquer le résultat de la question 36 à X_{ij} et $(X_{ij})^2$.
- 38 Utiliser le résultat de la question 37 pour montrer que les variables $\widehat{X}_{ij}$ sont bien définies.
- 40 Utiliser le résultat de la question 39 et l'inégalité $(a + b)^2 \leq 2a^2 + 2b^2$ pour tout $(a, b) \in \mathbb{R}^2$. Conclure en utilisant le résultat de la question 36.
- 41 S'aider de la concavité de la fonction $x \mapsto \sqrt{x}$ pour faire apparaître la quantité $\sum_{i=1}^n (\Lambda_{i,n} - \widehat{\Lambda}_{i,n})^2$ puis utiliser le résultat de la question 8.
- 42 Calculer la limite du résultat de la question 41 à l'aide de la question 40. Montrer ensuite que les $\widehat{X}_{ij}$ vérifient toutes les hypothèses de la partie III à l'aide du résultat de la question 38 pour ensuite appliquer le résultat de la question 35.
- 43 Prouver en introduisant une fonction lipschitzienne bien choisie que

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \mathbb{1}_{|\Lambda_{i,n}| \geq 3} \right) \xrightarrow{n \rightarrow \infty} 0$$

Effectuer ensuite un raisonnement similaire à celui de la question 35 en remplaçant le polynôme P par une fonction lipschitzienne qui vaut P sur $[-3; 3]$.

I. INÉGALITÉ DE HOFFMAN-WIELANDT

1 Soient $M \in \mathcal{M}_n(\mathbb{R})$ et $(P, Q) \in \mathcal{O}_n(\mathbb{R})^2$, alors

$$\begin{aligned}\|PMQ\|_F &= \sqrt{\text{Tr}(PMQ(PMQ)^T)} \\ &= \sqrt{\text{Tr}(PMQQ^T M^T P^T)} \quad ((M_1 M_2)^T = M_2^T M_1^T) \\ &= \sqrt{\text{Tr}(PMM^T P^T)} \quad (QQ^T = I_n) \\ &= \sqrt{\text{Tr}(P^T P M M^T)} \quad (\text{Tr}(M_1 M_2) = \text{Tr}(M_2 M_1)) \\ &= \sqrt{\text{Tr}(MM^T)} \quad (P^T P = I_n)\end{aligned}$$

$$\boxed{\|PMQ\|_F = \|M\|_F}$$

2 La matrice A est réelle et symétrique et le théorème spectral permet d'affirmer qu'elle est diagonalisable dans une base orthonormale. De même, la matrice B est réelle et symétrique donc diagonalisable dans une autre base orthonormale. Ainsi, il existe $(Q, R) \in \mathcal{O}_n(\mathbb{R})^2$ tel que

$$A = QD_A Q^T \quad \text{et} \quad B = RD_B R^T$$

Remarquons alors que

$$\begin{aligned}\|A - B\|_F &= \|QD_A Q^T - RD_B R^T\|_F \\ &= \|Q(D_A Q^T - Q^T R D_B R^T)\|_F \quad (Q^{-1} = Q^T) \\ &= \|Q(D_A Q^T R - Q^T R D_B)R^T\|_F \quad ((R^T)^{-1} = R) \\ \|A - B\|_F &= \|D_A Q^T R - Q^T R D_B\|_F \quad (\text{question 1})\end{aligned}$$

Définissons $P = Q^T R$. Comme Q^T et R appartiennent à $\mathcal{O}_n(\mathbb{R})$, alors $P \in \mathcal{O}_n(\mathbb{R})$ et

$$\boxed{\text{Il existe une matrice orthogonale } P \text{ telle que } \|A - B\|_F^2 = \|D_A P - P D_B\|_F^2.}$$

3 Posons $M = D_A P - P D_B$. En utilisant le résultat de la question 2, on a

$$\|A - B\|_F^2 = \|M\|_F^2 = \text{Tr}(MM^T) = \sum_{i=1}^n (MM^T)_{i,i} = \sum_{i=1}^n \sum_{j=1}^n M_{i,j} (M^T)_{j,i} = \sum_{i=1}^n \sum_{j=1}^n M_{i,j}^2$$

Soit $(i, j) \in \llbracket 1; n \rrbracket^2$, remarquons que

$$M_{i,j} = (D_A P - P D_B)_{i,j} = \sum_{k=1}^n [(D_A)_{i,k} p_{k,j} - p_{i,k} (D_B)_{k,j}]$$

Les matrices D_A et D_B sont diagonales : pour tout $(r, s) \in \llbracket 1; n \rrbracket^2$ avec $r \neq s$ alors $(D_A)_{r,s} = (D_B)_{r,s} = 0$. En particulier,

$$M_{i,j} = (D_A)_{i,i} p_{i,j} - p_{i,j} (D_B)_{j,j} = (\lambda_i(A) - \lambda_j(B)) p_{i,j}$$

On conclut que

$$\boxed{\|A - B\|_F^2 = \sum_{1 \leq i, j \leq n} p_{i,j}^2 (\lambda_i(A) - \lambda_j(B))^2}$$

Centrale Maths 2 MP 2021 — Corrigé

Ce corrigé est proposé par Sélim Cornet (professeur agrégé). Il a été relu par Bertrand Wiel (professeur en CPGE) et Gilbert Monna (professeur honoraire en CPGE).

Ce sujet est constitué de deux problèmes indépendants ayant pour thème commun la démonstration d'inégalités de Bernstein. Il s'agit d'inégalités permettant de contrôler les valeurs maximales de la dérivée d'une fonction par les valeurs maximales de cette même fonction.

- Le premier problème est plutôt original. Il propose la démonstration de telles inégalités pour des fonctions polynomiales et des fonctions polynômes trigonométriques. Après quelques questions préliminaires sur la famille des polynômes de Tchebychev, on démontre une première inégalité de Bernstein pour les fonctions polynômes trigonométriques : si f est une combinaison linéaire des fonctions de la famille $(\cos(k\cdot), \sin(k\cdot))_{0 \leq k \leq n}$, alors

$$\sup_{\mathbb{R}} |f'| \leq n \sup_{\mathbb{R}} |f|$$

On en déduit une seconde inégalité pour les fonctions polynomiales :

$$\forall P \in \mathbb{C}_n[X] \quad \sup_{x \in [-1; 1]} |P'(x)| \leq n^2 \sup_{x \in [-1; 1]} |P(x)|$$

- Le second problème est plus classique. Il se place dans le cadre des fonctions intégrables. Deux célèbres outils d'analyse dans des espaces de fonctions intégrables sont tout d'abord introduits : la transformation de Fourier et le produit de convolution définis pour f intégrable et g bornée sur $\mathbb{R}$ par

$$\widehat{f} : x \mapsto \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(\xi) e^{-ix\xi} d\xi \quad \text{et} \quad f * g : x \mapsto \int_{-\infty}^{+\infty} f(x-t)g(t) dt$$

La troisième partie est consacrée à la construction d'une fonction auxiliaire. Enfin, la quatrième partie conduit à la démonstration de l'inégalité proprement dite : il existe $C > 0$ telle que pour toute fonction f intégrable sur $\mathbb{R}$, de classe $\mathcal{C}^1$ et dont la transformée de Fourier est nulle en dehors de $[-\lambda; \lambda]$ on a

$$\sup_{\mathbb{R}} |f'| \leq C \lambda \sup_{\mathbb{R}} |f|$$

Si les thèmes du programme abordés dans ce sujet sont relativement peu nombreux, celui-ci constitue néanmoins un bon sujet d'entraînement, notamment pour qui souhaite pratiquer la trigonométrie ou le raisonnement par récurrence. Les deux problèmes suivent une progression typique d'un sujet de la banque d'épreuves Centrale-Supélec : chacun commence par des questions classiques et proches du cours, puis la difficulté et la technicité des questions augmentent progressivement. Le sujet reste néanmoins abordable.

INDICATIONS

- 1 Conjecturer le degré sur les premiers termes de la suite $(T_n)_{n \in \mathbb{N}}$, puis démontrer la conjecture par récurrence.
- 2 Procéder par récurrence. C'est également le moment de se souvenir des formules de trigonométrie!
- 3 Décomposer P dans la base formée par les $(T_k)_{0 \leq k \leq n}$.
- 4 Tout réel de l'intervalle $[-1; 1]$ peut s'écrire $\cos(\theta)$ pour un certain réel θ .
- 5 Procéder par récurrence pour démontrer l'inégalité donnée en indication dans l'énoncé. Utiliser cette inégalité pour majorer la norme de T_n' . Montrer enfin l'égalité en calculant un développement limité de $T_n'(\cos(\theta))$ lorsque θ tend vers 0.
- 6 Former la décomposition en éléments simples de B/A .
- 8 Reconnaître un taux d'accroissement dans la définition de Q_λ .
- 9 Vérifier que les ω_k sont des racines distinctes de R .
- 10 Appliquer la formule (I.1) avec les polynômes R et Q_λ pour obtenir la première relation. Évaluer ensuite en 1 pour en déduire la seconde relation.
- 11 Appliquer deux fois l'inégalité (I.2): une première fois au polynôme X^{2n} , puis réinjecter le résultat obtenu dans la formule (I.2) appliquée au polynôme P .
- 12 Décomposer f comme somme de fonctions trigonométriques, puis appliquer les formules d'Euler.
- 13 Penser d'abord à la technique de l'angle moitié. Appliquer ensuite la relation démontrée à la question 11 avec $\lambda = e^{i\theta}$.
- 14 Recourir une nouvelle fois à la question 11, appliquée au polynôme $P = 1$.
- 15 Procéder au changement de variable $x = \cos(\theta)$.
- 16 Faire appel au résultat de la question 3 et à des formules de trigonométrie pour montrer que $f \in \mathcal{S}_n$. Remarquer ensuite que $Q(1) = f'(0)$.
- 17 Traiter séparément les cas $t = 0$ et $t \neq 0$. Dans ce dernier cas, un changement de variable pourra être utile.
- 18 Rassembler les résultats des questions 15 et 17.
- 19 Penser aux polynômes de Tchebychev.
- 20 Vérifier les hypothèses du théorème de continuité des intégrales à paramètre.
- 21 Penser au critère de continuité pour les applications linéaires entre espaces normés.
- 22 Procéder à un changement de variable.
- 23 Effectuer là encore un changement de variable.
- 25 Utiliser le théorème de dérivation des intégrales à paramètre.
- 27 Démontrer par récurrence le résultat donné en indication. Faire appel au théorème de prolongement $\mathcal{C}^1$ pour prouver l'hérédité.
- 29 Étudier les variations de θ .
- 30 Commencer par construire une fonction $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ de classe $\mathcal{C}^\infty$ qui vérifie les inclusions $\alpha([-1; 1]) \subset]-\infty; -1]$ et $\alpha(\mathbb{R} \setminus [-2; 2]) \subset [1; +\infty[$.
- 32 Procéder à une double intégration par parties.
- 33 Prouver que $\widehat{f} = \widehat{\lambda f * r_\lambda}$.
- 34 Exprimer f' à l'aide des questions 25 et 33, et exploiter la question 24.

I. INÉGALITÉ POLYNOMIALE DE BERNSTEIN ET APPLICATIONS

1 Remarquons que $\deg(T_0) = 0$, $\deg(T_1) = 1$ et $\deg(T_2) = \deg(2X^2 - 1) = 2$. On conjecture que, pour tout $n \in \mathbb{N}$, le polynôme T_n est de degré n . Montrons donc par une récurrence à deux pas que la propriété

$$\mathcal{P}(n) : \quad \deg(T_n) = n$$

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

- $\mathcal{P}(0)$ est vraie puisque $\deg(T_0) = \deg(1) = 0$.
- $\mathcal{P}(1)$ est vraie puisque $\deg(T_1) = \deg(X) = 1$.
- $[\mathcal{P}(n) \wedge \mathcal{P}(n+1)] \implies \mathcal{P}(n+2)$: soit $n \in \mathbb{N}$ tel que $\mathcal{P}(n)$ et $\mathcal{P}(n+1)$ soient vraies. Montrons que $\mathcal{P}(n+2)$ est vraie. Pour cela, remarquons d'abord que $T_{n+2} = 2XT_{n+1} - T_n$. Or, par hypothèse de récurrence,

$$\deg(2XT_{n+1}) = \deg(2X) + \deg(T_{n+1}) = 1 + n + 1 = n + 2$$

et $\deg(T_n) = n$. Comme $2XT_{n+1}$ et T_n sont de degrés différents, il s'ensuit que

$$\deg(T_{n+2}) = \max(\deg(2XT_{n+1}), \deg(T_n)) = \max(n + 2, n) = n + 2$$

Cela prouve que $\mathcal{P}(n+2)$ est vraie.

- Conclusion : par récurrence, la propriété $\mathcal{P}(n)$ est vraie pour tout $n \in \mathbb{N}$. Ainsi,

Pour tout $n \in \mathbb{N}$, $\deg(T_n) = n$.

La famille de polynômes $(T_k)_{0 \leq k \leq n}$ est échelonnée en degré, elle est par conséquent libre. Or, son cardinal est égal à $n + 1$, soit à la dimension de $\mathbb{C}_n[X]$, d'où

La famille $(T_k)_{0 \leq k \leq n}$ forme une base de $\mathbb{C}_n[X]$.

2 Montrons à nouveau par une récurrence à deux pas que la propriété

$$\mathcal{P}(n) : \quad \forall \theta \in \mathbb{R} \quad T_n(\cos(\theta)) = \cos(n\theta)$$

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

- $\mathcal{P}(0)$ est vraie puisque pour tout $\theta \in \mathbb{R}$, $T_0(\cos(\theta)) = 1 = \cos(0\theta)$.
- $\mathcal{P}(1)$ est vraie puisque pour tout $\theta \in \mathbb{R}$, $T_1(\cos(\theta)) = \cos(\theta)$.
- $[\mathcal{P}(n) \wedge \mathcal{P}(n+1)] \implies \mathcal{P}(n+2)$: soit $n \in \mathbb{N}$ tel que $\mathcal{P}(n)$ et $\mathcal{P}(n+1)$ soient vraies. Montrons que $\mathcal{P}(n+2)$ est vraie. Soit $\theta \in \mathbb{R}$. Calculons, en utilisant la relation de récurrence sur les $(T_n)_{n \in \mathbb{N}}$ et l'hypothèse de récurrence,

$$T_{n+2}(\cos(\theta)) = 2\cos(\theta)T_{n+1}(\cos(\theta)) - T_n(\cos(\theta)) = 2\cos(\theta)\cos((n+1)\theta) - \cos(n\theta)$$

$$\text{Or} \quad \cos((n+1)\theta) = \cos(n\theta + \theta) = \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta)$$

On en déduit

$$\begin{aligned} T_{n+2}(\cos(\theta)) &= 2\cos(\theta)(\cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta)) - \cos(n\theta) \\ &= \cos(n\theta)(2\cos(\theta)^2 - 1) - 2\sin(n\theta)\cos(\theta)\sin(\theta) \end{aligned}$$

puis, grâce aux formules de duplication et d'addition,

$$T_{n+2}(\cos(\theta)) = \cos(n\theta)\cos(2\theta) - \sin(n\theta)\sin(2\theta) = \cos(n\theta + 2\theta) = \cos((n+2)\theta)$$

ce qui prouve que $\mathcal{P}(n+2)$ est vraie.

- Conclusion : $\mathcal{P}(n)$ est vraie pour tout $n \in \mathbb{N}$ et finalement

Pour tout $n \in \mathbb{N}$, pour tout $\theta \in \mathbb{R}$, $T_n(\cos(\theta)) = \cos(n\theta)$.

3 Soient $n \in \mathbb{N}$ et $P \in \mathbb{C}_n[X]$. Comme $(T_k)_{0 \leq k \leq n}$ forme une base de $\mathbb{C}_n[X]$, il existe des nombres complexes $a_0, \dots, a_n$ tels que

$$P = \sum_{k=0}^n a_k T_k$$

Il découle alors du résultat de la question 2 que, pour tout $\theta \in \mathbb{R}$,

$$P(\cos(\theta)) = \sum_{k=0}^n a_k T_k(\cos(\theta)) = \sum_{k=0}^n a_k \cos(k\theta) = a_0 + \sum_{k=1}^n a_k \cos(k\theta)$$

Ainsi,

La fonction $\theta \mapsto P(\cos(\theta))$ est dans $\mathcal{S}_n$.

Les fonctions de $\mathcal{S}_n$ sont appelées des polynômes trigonométriques. Elles constituent notamment la base de la théorie des séries de Fourier, qui consiste à approcher une fonction périodique et régulière par une suite de polynômes trigonométriques.

4 Soient $n \in \mathbb{N}$ et $x \in [-1; 1]$. Soit $\theta = \text{Arccos}(x)$, ainsi $x = \cos(\theta)$. Il s'ensuit

$$|T_n(x)| = |T_n(\cos(\theta))| = |\cos(n\theta)| \leq 1$$

ce qui prouve $\|T_n\|_{L^\infty([-1; 1])} \leq 1$. En outre, remarquons que

$$|T_n(1)| = |T_n(\cos(0))| = |\cos(n \times 0)| = |\cos(0)| = 1$$

Cela permet de conclure que

$$\|T_n\|_{L^\infty([-1; 1])} = 1$$

5 Suivons l'indication de l'énoncé et commençons par prouver, par récurrence, que la propriété

$$\mathcal{P}(n) : \quad \forall \theta \in \mathbb{R} \quad |\sin(n\theta)| \leq n |\sin(\theta)|$$

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

- $\mathcal{P}(0)$ est vraie : en effet, pour tout $\theta \in \mathbb{R}$,

$$|\sin(0 \times \theta)| = |\sin(0)| = 0 \leq 0 |\sin(\theta)|$$

- $\mathcal{P}(n) \implies \mathcal{P}(n+1)$: soit $n \in \mathbb{N}$ tel que $\mathcal{P}(n)$ soit vraie. Montrons $\mathcal{P}(n+1)$. Pour cela, considérons un réel θ quelconque et calculons

$$|\sin((n+1)\theta)| = |\sin(n\theta) \cos(\theta) + \cos(n\theta) \sin(\theta)|$$

Appliquons l'inégalité triangulaire :

$$|\sin((n+1)\theta)| \leq |\sin(n\theta)| |\cos(\theta)| + |\cos(n\theta)| |\sin(\theta)|$$

Les majorations $|\cos(\theta)| \leq 1$, $|\cos(n\theta)| \leq 1$ et l'hypothèse de récurrence entraînent alors

$$|\sin((n+1)\theta)| \leq n |\sin(\theta)| + |\sin(\theta)| = (n+1) |\sin(\theta)|$$

- Conclusion : pour tout $n \in \mathbb{N}$ et pour tout $\theta \in \mathbb{R}$,

$$|\sin(n\theta)| \leq n |\sin(\theta)|$$

Centrale Informatique MP-PC-PSI 2021 — Corrigé

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

Ce sujet a pour objectif la réalisation d'une image en deux dimensions représentant une scène simple, composée de plusieurs sources lumineuses et de plusieurs objets, choisis sphériques pour simplifier le problème. Il permet d'aborder une technique très efficace de fabrication d'images virtuelles et demandant peu de ressources. Les structures manipulées sont des tableaux `numpy` et des fonctions sont données pour en faciliter l'usage.

- La partie I permet la construction des fonctions élémentaires de géométrie et des premiers objets optiques tels que rayons, sphères et intersections des deux.
- Dans la partie II, l'étude évolue vers la problématique de la visibilité d'une source lumineuse depuis un point d'une sphère et de la couleur obtenue compte tenu de la lumière de la source et de l'incidence du rayon lumineux.
- La partie III, indépendante du reste de l'énoncé, demande de traiter l'enregistrement des données relatives à une scène (positions et couleurs des objets et des sources) au sein d'une base de données. C'est l'occasion de poser quatre questions de langage SQL d'un niveau très progressif.
- L'étude entre dans le vif du sujet à la quatrième partie, avec la modélisation du « lancer de rayons », considération physiquement fautive mais compatible avec la réalité qui consiste à imaginer que les rayons lumineux partent de l'œil. On modélise ces rayons et leur première intersection avec les objets de la scène, puis on étudie la couleur vue et la projection de cette même scène sur un écran en deux dimensions.
- La partie V, plus difficile que les précédentes, demande de ne pas se perdre dans les notations et le concept de rayon lumineux se réfléchissant d'une surface à l'autre. Les deux dernières questions portent sur une optimisation qui reste quelque peu théorique par rapport au reste du sujet.

Ce sujet donne une bonne idée de ce que l'on peut faire avec des outils informatiques, sur le thème assez pratique de la réalisation d'images de synthèse. Il est globalement bien mené et progressif, tous les concepts sont parfaitement définis. Il ne contient que de la manipulation assez simple de tableaux `numpy`, accessible dès la première année. Il a le grand intérêt de faire produire quasiment l'ensemble du code nécessaire à la mise en œuvre : le lecteur intéressé pourra rapidement s'amuser à créer lui-même des images. La fin du sujet réserve des questions plus délicates pour les candidats plus à l'aise.

INDICATIONS

Partie I

- 1 Bien lire les paragraphes précédant la question, notamment à propos de la représentation des points et des vecteurs.
- 2 Lire attentivement l'annexe et les fonctions du module `numpy` autorisées.
- 5 Le mot-clé `assert` permet la vérification de la condition donnée et arrête l'exécution si elle n'est pas vérifiée.
- 8 Il faut résoudre l'équation précédente. On peut remarquer que les deux solutions possibles doivent obligatoirement être positives.

Partie II

- 10 Faire un dessin en deux dimensions et regarder les vecteurs présents.
- 11 Il faut chercher les intersections du rayon partant de la source lumineuse et passant par P. Ne pas oublier que les sphères plus éloignées de la source que P n'interviennent pas.
- 12 Le code à produire est très court, il suffit de traduire la relation (2) de l'énoncé. Le cosinus de θ peut être obtenu d'après le schéma par un produit scalaire.
- 13 Faire un schéma en deux dimensions et décomposer $\vec{u}$ sur l'axe de $\vec{N}$ et l'axe orthogonal.

Partie III

- 14 Pour récupérer l'année, utiliser une fonction donnée en annexe.
- 16 Regarder où se trouvent les données permettant la sélection et celles nécessaires à la projection, puis déterminer comment joindre les tables concernées.
- 17 La fonction `OCCULTE` demande deux identifiants d'objets indépendants, il faut donc joindre deux fois la table `Objet` en les renommant.

Partie IV

- 18 Déterminer le lien entre x et j , puis entre y et i . Pour cela, faire un schéma représentant les cases et le centre de la figure, pour une valeur de N faible. Attention, sur la figure de l'énoncé i et j ne sont pas en face de $E(i, j)$.
- 20 Pour retourner le point d'intersection le plus proche et l'indice de l'objet correspondant, il est nécessaire de récupérer toutes les intersections avec la distance à l'œil, le point de contact et l'indice de l'objet afin de conserver les bonnes valeurs.
- 21 Le calcul de couleur diffusée depuis une source par un point de sphère a déjà été fait à la question 12. Il suffit d'ajouter les couleurs correspondant aux différentes sources, sans oublier de vérifier si elles sont bien visibles.

Partie V

- 25 Une boucle `while` ou `for` est possible. À chaque itération, si l'interception du rayon existe, on la stocke et on continue.
- 26 La relation donnée dans l'énoncé implique que l'on doit parcourir le tableau des réflexions en partant de la fin.
- 27 Reprendre la fonction `lancer` et regarder ce que `couleur_perçue` y modifie.
- 29 Créer l'ensemble des triplets `{objr_id, so_id, objo_id}`. Utiliser l'opérateur `in` pour vérifier leur présence dans la liste `risque`.
- 30 Reprendre la fonction `visible` et l'adapter. Attention à l'ordre des arguments.

I. GÉOMÉTRIE

1 La fonction `vec` calcule les composantes du vecteur $\overrightarrow{AB}$ en soustrayant les coordonnées du point A à celles du point B.

```
def vec(A, B):  
    return B - A
```

L'énoncé dit explicitement que retrancher deux vecteurs est réalisé par l'opération `v1 - v2` et que les vecteurs, comme les points, sont représentés par des tableaux. Il fallait donc éviter d'écrire une boucle ici.

2 La fonction `ps` calcule le produit scalaire de deux vecteurs, somme des produits terme à terme de leurs composantes, ce que fait précisément la fonction `np.inner`.

```
def ps(v1, v2):  
    return np.inner(v1, v2)
```

La méthode `inner` du module `numpy` est donnée dans l'annexe et sa description correspond exactement au produit scalaire. Il est néanmoins possible de réaliser cette opération autrement, notamment en faisant la somme des composantes du produit de Hadamard des deux vecteurs :

```
def ps(v1, v2):  
    return np.sum(v1 * v2)
```

La réalisation de ce calcul par une boucle, demandant beaucoup plus d'écriture, a certainement été acceptée par les correcteurs, mais ce n'est visiblement pas la réponse attendue. On peut écrire par exemple

```
def ps(v1, v2):  
    somme = 0  
    for i in range(len(v1)):  
        somme += v1[i] * v2[i]  
    return somme
```

3 La fonction `norme` calcule la racine carrée du produit scalaire du vecteur par lui-même.

```
def norme(v):  
    return math.sqrt(ps(v, v))
```

4 La fonction `unitaire` retourne le vecteur unitaire correspondant à l'argument, en utilisant le produit d'un scalaire et d'un vecteur.

```
def unitaire(v):  
    return (1/norme(v)) * v
```

5 Le mot-clé `assert` n'est pas au programme et sa présence dans un code d'une des premières questions du sujet a pu décontenancer un certain nombre d'étudiants. Cette instruction correspond à la vérification de la condition donnée, à savoir est-ce que `t` est positif ou non. Si c'est le cas, l'instruction ne fait rien. Si non, elle *lève* une erreur, et l'utilisateur obtient alors le message `AssertionError` avant un arrêt immédiat de l'exécution.

La fonction `pt` permet de calculer, pour un rayon lumineux $\mathbf{r} = (S, \vec{u})$ et un entier $\mathbf{t}$ donnés, la position du point M du rayon tel que $\overrightarrow{SM} = t \vec{u}$. Elle ne fonctionne, comme indiqué dans la description donnée dans l'énoncé, que pour $t \geq 0$, car t est la distance séparant M de la source, dans la direction définie par $\vec{u}$.

La fonction `dir` définit le vecteur unitaire associé au vecteur $\overrightarrow{AB}$, donc au rayon passant par les points A et B et allant de A vers B.

La fonction `ra` définit le rayon lumineux partant du point A et passant par B.

6 La fonction `sp` crée la représentation de la sphère de centre A et passant par B.

```
def sp(A, B):
    return A, norme(vec(A, B))
```

7 On considère un point M appartenant à la droite passant par le point A et de vecteur directeur $\vec{u}$. Cela signifie qu'il existe un réel t tel que

$$\overrightarrow{AM} = t \vec{u}$$

Ce point M appartient à la sphère de centre C et de rayon r si et seulement si la distance MC vaut r , soit $\|\overrightarrow{MC}\|^2 = \overrightarrow{MC} \cdot \overrightarrow{MC} = r^2$. Or,

$$\begin{aligned} \overrightarrow{MC} \cdot \overrightarrow{MC} &= (\overrightarrow{CA} + \overrightarrow{AM}) \cdot (\overrightarrow{CA} + \overrightarrow{AM}) \\ &= \overrightarrow{CA} \cdot \overrightarrow{CA} + \overrightarrow{AM} \cdot \overrightarrow{AM} + 2 \overrightarrow{CA} \cdot \overrightarrow{AM} \\ \overrightarrow{MC} \cdot \overrightarrow{MC} &= \|\overrightarrow{CA}\|^2 + t^2 + 2 \overrightarrow{CA} \cdot (t \vec{u}) \end{aligned}$$

On trouve ainsi que $t^2 + 2t \vec{u} \cdot \overrightarrow{CA} + \|\overrightarrow{CA}\|^2 - r^2 = 0$

Cette équation est un trinôme du second degré. Si son discriminant est strictement positif, elle a deux solutions réelles correspondant aux deux intersections de la droite et de la sphère. Un discriminant nul indique une unique solution réelle double, correspondant à une droite tangente à la sphère. Enfin, dans le cas d'un discriminant strictement négatif, les solutions ne sont pas réelles, la droite et la sphère ne sont alors pas sécantes.

8 L'équation déterminée ci-dessus a des solutions réelles si

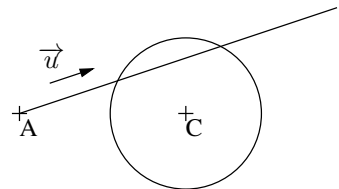
$$\Delta = 4(\vec{u} \cdot \overrightarrow{CA})^2 - 4(\|\overrightarrow{CA}\|^2 - r^2) \geq 0$$

De plus, avec A à l'extérieur de la sphère, le schéma ci-contre montre que les deux solutions doivent être positives. Il faut donc que leur produit et leur somme soient positifs. Dans un trinôme de la forme $ax^2 + bx + c = 0$, le produit des racines vaut c/a et la somme $-b/a$. La condition sur le produit est déjà réalisée si A est à l'extérieur de la sphère. Celle sur la somme rend nécessaire la condition

$$\vec{u} \cdot \overrightarrow{CA} < 0$$

Si c'est le cas, il faut alors retourner la plus petite solution des deux, soit

$$t = -\vec{u} \cdot \overrightarrow{CA} - \frac{\sqrt{\Delta}}{2}$$



Centrale Informatique optionnelle MP 2021

Corrigé

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

Ce sujet porte sur la génération aléatoire de pavages d'un échiquier par des dominos. L'énoncé utilise pour cela une bijection entre ces pavages et les arbres couvrants d'un certain type de graphe. Les parties dépendent les unes des autres et le sujet est de difficulté croissante.

- Dans la partie I, on introduit les fonctions de base qui servent à manipuler et générer des graphes non orientés en forme de quadrillage. Elle comporte quelques questions de réindexation sur lesquelles il convient d'être particulièrement vigilant.
- La partie suivante introduit la notion d'arbre couvrant d'un graphe non orienté. Elle demande d'établir quelques résultats classiques qui caractérisent ces sous-graphes connexes et acycliques, et s'achève par la rédaction d'une fonction qui permet de vérifier si un sous-graphe donné est un arbre.

On notera à cet effet plusieurs questions portant sur la structure de données *union-find* qui permet de manipuler efficacement des partitions d'ensembles.

- La partie III étudie et implémente l'algorithme de Wilson, qui permet de générer un arbre couvrant aléatoire d'un graphe quelconque. La structure utilisée pour représenter les arbres couvrants sera conservée dans tout le reste du sujet.
- Dans la partie IV, on introduit la notion de pavage d'un échiquier par des dominos rectangulaires. On s'intéresse uniquement au cas où l'échiquier est un rectangle de dimensions impaires, privé de son coin inférieur gauche. L'énoncé admet l'existence d'une bijection entre les pavages de ce dernier et les arbres couvrants d'un graphe dont la forme est celle étudiée en première partie. À l'aide d'une représentation simple d'un pavage en Caml, on construit une première fonction bijective associant un arbre couvrant à un pavage.
- La dernière partie, longue et difficile, s'intéresse à la génération aléatoire d'un pavage. On s'intéresse pour cela au problème réciproque de celui de la partie IV : construire un pavage à partir d'un arbre couvrant, ce qui nécessite l'introduction de la notion d'arbre dual des graphes de quadrillage. Cette dernière partie s'achève sur un code de longueur assez considérable, qui combine les fonctions des cinq parties pour aboutir à la génération aléatoire attendue.

Ce sujet est un très bon sujet de révision sur les graphes et sur les structures de données, malgré quelques imprécisions, notamment une définition du dual d'un dual peu claire à la question 28. Sur les 38 questions composant le sujet, 20 demandent d'écrire du code, de difficulté variable. Il est sans doute un peu long pour une épreuve de quatre heures et nécessite de rester concentré jusqu'à la fin.

INDICATIONS

- 1 Utiliser une relation entre le nombre d'arêtes et les degrés des sommets d'un graphe.
- 4 Différencier les cas d'une arête verticale et d'une arête horizontale, puis trouver comment situer l'arête dans le graphe.
- 7 Pour montrer que deux parties sont égales ou disjointes, montrer une double inclusion s'il y a un élément commun à deux parties.
- 8 Montrer l'existence d'un plus court chemin de s à t en considérant l'ensemble des longueurs des chemins de s à t . Montrer ensuite qu'un chemin qui passe deux fois par le même sommet peut être réduit.
- 9 Le caractère acyclique d'un graphe se traduit par le fait qu'un chemin au plus peut relier deux sommets, ce qui signifie que sans une arête $\{x, y\}$, les deux sommets x et y ne sont plus reliés dans G .
- 10 Montrer d'abord qu'ajouter une arête dans un graphe diminue le nombre de composantes connexes ou crée un cycle.
- 11 Un élément est un représentant quand sa case est négative, et un élément qui n'est pas un représentant pointe dans le tableau **parent** soit vers un représentant, soit vers un élément dont on peut chercher le représentant.
- 13 Raisonner en terme d'invariant de la partition ou par récurrence sur le nombre de réunions effectuées pour construire la partition.
- 15 Utiliser la caractérisation de la question 10. Le test de connexité se fait à l'aide de la structure de partition étudiée dans la partie.
- 20 Utiliser les fonctions précédentes pour générer un chemin aléatoire et le greffer pour chaque sommet qui n'est pas dans l'arbre.
- 24 Utiliser une division euclidienne.
- 25 Faire un pattern matching sur les directions possibles.
- 26 Utiliser la fonction de la question précédente pour déterminer le parent de chaque sommet autre que 0.
- 29 Par l'absurde, considérer un cycle de (S_n^*, B^*) et distinguer les faces intérieures et extérieures.
- 30 Réutiliser la caractérisation de la question 10, en remarquant que si B a $n - 1$ arêtes alors B^* a $n^* - 1$ arêtes. Utiliser enfin le résultat de la question 28 pour passer de G^{**} à G .
- 31 Vérifier pour chaque arête si l'un des deux sommets est parent de l'autre.
- 32 Effectuer un parcours en profondeur en partant de la racine pour associer à chaque sommet son parent.
- 35 Convertir l'arbre en son arbre dual en utilisant le tableau de booléens, grâce aux fonctions **vers_parent** et **vers_couple**. Pour construire le dual, dans la représentation par tableau, appliquer la définition.
- 36 Commencer par traiter les sommets noirs, puis les gris. Distinguer les sommets gris au bord de l'échiquier et ceux dans les coins, en utilisant les directions indiquées par les sommets noirs autour lorsqu'un sommet gris a comme père le sommet 0.

I. QUELQUES FONCTIONS AUXILIAIRES

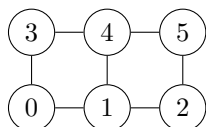
1 Pour un sommet s , notons $d(s)$ le degré de s , à savoir le nombre d'arêtes dont s est une extrémité. Les graphes étant représentés par listes d'adjacence, la quantité $|A|$ peut se calculer à partir des degrés des sommets à l'aide de la formule suivante :

$$2|A| = \sum_{s \in S} d(s)$$

En effet, compter le degré des sommets pour tous les sommets revient à compter les arêtes deux fois : $\{x, y\}$ est comptée une fois lorsque x apparaît comme voisin de y , et une seconde fois lorsque y apparaît comme voisin de x . Cette formule mène au programme suivant :

```
let nombre_aretes g =
  let n = Array.length g in
  let k = ref 0 in
  for i = 0 to n - 1 do
    k := !k + List.length g.(i)
    (*List.length g.(i) représente le degré de i dans g*)
  done;
  !k / 2 ;;
```

2 Commençons par dessiner le graphe $G_{3,2}$:



On liste alors aisément les voisins de chaque sommet, d'où l'instruction suivante :

```
let g = [| [|1;3|]; [|0;2;4|]; [|1;5|]; [|0;4|]; [|1;3;5|]; [|2;4|] |] ;;
```

3 Le programme consiste simplement à construire un tableau dans lequel chaque liste du tableau donné en entrée est convertie en tableau :

```
let adjacence g =
  let n = Array.length g in
  let sortie = Array.make n [|] in
  for i = 0 to n - 1 do
    sortie.(i) <- Array.of_list g.(i)
  done;
  sortie;;
```

Évaluons la complexité de cette fonction :

- les fonctions `Array.length` et `Array.make` s'exécutent en temps constant ;
- la boucle `for` contient n itérations, sa complexité est la somme des complexités des appels à `Array.of_list` ;
- la fonction `Array.of_list` a une complexité linéaire en la taille de la liste.

La complexité de la fonction `adjacence` est donc linéaire en la somme des tailles des listes de son graphe d'entrée, ce qui correspond à la somme des degrés des sommets, valant $2m$. On en conclut que

La complexité de la fonction `adjacence` est en $O(n + m)$.

4 Pour déterminer le rang de $\{s, t\}$, on procède en plusieurs étapes :

- On détermine en premier lieu si l'arête est verticale. Pour ce faire, on sait déjà que s et t sont adjacents, donc $t - s$ vaut 1 ou p , et il suffit ainsi seulement de tester si leur différence vaut p .
- Si l'arête est verticale, il suffit de compter les arêtes qui se situent à gauche ou en-dessous de l'arête donnée. Notons a (resp. b) le numéro de ligne (resp. de colonne) de s en partant du bas (resp. de la gauche).
 - il y a $b \cdot (q-1)$ arêtes à gauche de $\{s, t\}$, car il y a $q-1$ arêtes par colonne ;
 - il y a a arêtes en-dessous de $\{s, t\}$.

L'arête est donc de rang $a + (q-1) \cdot b$.

- De façon similaire, on calcule le rang d'une arête horizontale mais en comptant d'abord par ligne, puis par colonne. Il ne faut bien sûr pas oublier d'ajouter $p \cdot (q-1)$ pour tenir compte de toutes les arêtes verticales comptées avant.

Le code résultant de ce raisonnement est le suivant :

```
let rang (p,q) (s,t) =
  let a, b = s / p, s mod p in
  if t-s=p then
    a + b * (q-1)
  else
    a * (p-1) + b + p * (q-1) ;;
```

5 Comme précédemment, pour déterminer les sommets correspondant au rang i , raisonnons par étapes :

- Déterminons d'abord si l'arête cherchée est verticale ou horizontale : il suffit de regarder si son rang est plus ou moins grand que le nombre d'arêtes verticales du graphe, soit $p \cdot (q-1)$.
- Si l'arête est verticale, on sait qu'elle est précédée de i arêtes. Cherchons alors la valeur de l'indice de s , à laquelle on ajoutera p pour trouver la valeur de l'indice de t . En notant à nouveau a l'indice de ligne de s et b l'indice de colonne de s , on sait que $s = a + b \cdot p$. Pour trouver a et b , on effectue la division euclidienne de i par $q-1$ puisque la question précédente assure que $i = a + b \cdot (q-1)$ avec $0 \leq a < q$.
- Un raisonnement analogue, en enlevant $p \times (q-1)$ au rang de l'arête, permet de trouver l'indice de l'extrémité gauche d'une arête horizontale (on ajoute alors 1 pour obtenir l'indice de l'extrémité droite).

Ce code s'en déduit directement :

```
let sommets (p,q) i =
  if i < p * (q-1) then
    let a, b = i mod (q-1), i / (q-1) in
    let s = a * p + b in
    s, s + p
  else
    let i' = i - p * (q-1) in
    let a, b = i' / (p-1), i' mod (p-1) in
    let s = a * p + b in
    s, s + 1 ;;
```

Mines Maths 1 MP 2021 — Corrigé

Ce corrigé est proposé par Antoine Barrier (ENS Paris-Saclay) ; il a été relu par Guillaume Duboc (ENS Lyon) et Gilbert Monna (professeur honoraire en CPGE).

La limite d'une suite est introduite et étudiée en classes préparatoires principalement dans le cadre des espaces vectoriels normés. Dans un autre contexte, les probabilités, il est également possible de définir la notion de limite d'une suite de variables aléatoires. C'est cette nouvelle notion qu'aborde le sujet : il porte sur une démonstration du théorème de De Moivre-Laplace, qui décrit le comportement limite d'une suite $(X_n)_{n \in \mathbb{N}^*}$ de variables aléatoires suivant des lois binomiales $(X_n \hookrightarrow \mathcal{B}(n; p))$. Il s'agit d'un cas particulier du théorème central limite, un résultat fondamental de la théorie des probabilités.

- Dans la partie I, on établit divers résultats préliminaires, les outils principaux étant des analyses asymptotiques.
- La partie II s'intéresse au comportement asymptotique du maximum p_n de l'ensemble $\{P(X_n = k) \mid k \in \llbracket 0; n \rrbracket\}$. On montre que la suite $(\sqrt{np(1-p)}p_n)_{n \in \mathbb{N}^*}$ converge, ce qui sera utile dans la partie suivante. Pour cela, on s'appuie majoritairement sur les résultats préliminaires et sur la caractérisation de la loi binomiale.
- La partie III démontre le théorème de De Moivre-Laplace. C'est la partie la plus longue et la plus importante du sujet. Les lois binomiales sont renormalisées en variables $(Y_n)_{n \in \mathbb{N}^*}$. On montre que $(P(a \leq Y_n \leq b))_{n \in \mathbb{N}^*}$ converge vers l'intégrale d'une certaine fonction sur le segment $[a; b]$. Les variables aléatoires $(Y_n)_{n \in \mathbb{N}^*}$ étant discrètes, et le terme limite s'exprimant comme une intégrale, on a besoin d'introduire des fonctions auxiliaires permettant de faire le lien entre discret et continu. Le théorème de convergence dominée sera le point crucial de la démonstration.
- Après cela, la quatrième partie s'intéresse à deux applications : une démonstration de la valeur de l'intégrale de Gauss puis une extension du résultat de la partie III au cas où a ou b est infini. Cette extension, qui pourrait sembler immédiate, nécessite une rédaction précise.
- Enfin, la dernière partie porte sur une généralisation du résultat obtenu. On regarde des variables aléatoires $(Z_n)_{n \in \mathbb{N}^*}$ définies en fonction de $(Y_n)_{n \in \mathbb{N}^*}$ et on démontre, sous certaines hypothèses, une convergence du même type que celle obtenue pour $(Y_n)_{n \in \mathbb{N}^*}$. On utilise des résultats d'intégration, notamment le théorème de changement de variable.

Le problème fait appel à une grande partie du programme d'analyse de MP, notamment les chapitres de probabilités discrètes et d'intégration. La maîtrise des comparaisons asymptotiques de suites, vues en première année, est indispensable pour traiter correctement ce sujet. L'énoncé est très progressif, avec une première partie sans difficulté notable, une deuxième de niveau moyen (hormis la difficile question 5) puis une troisième partie plus ambitieuse qui, après quelques questions techniques, reprend les raisonnements mis en place dans la seconde partie de manière plus poussée.

INDICATIONS

Partie I

- 2 Encadrer $\lfloor \lambda x + \mu \rfloor$ et $\lceil \lambda x + \mu \rceil$ par $\lambda x + \mu - 1$ et $\lambda x + \mu + 1$.
 4 Faire un développement limité de $\ln(1+x)$ à l'ordre 2.

Partie II

- 5 Montrer que $k \mapsto P(X_n = k)$ est croissante sur $\llbracket 0; x_n \rrbracket$ et décroissante sur $\llbracket x_n; n \rrbracket$ en comparant le rapport $P(X_n = k+1)/P(X_n = k)$ avec 1.
 6 Utiliser le résultat de la question 2 puis appliquer la formule de Stirling à $n!$, $x_n!$ et $(n - x_n)!$
 8 Vérifier que $\lim_{n \rightarrow +\infty} (x_n - np)/n = 0$ et appliquer la question 4.

Partie III

- 9 Utiliser la relation entre X_n et Y_n ainsi que les propriétés de l'espérance et de la variance.
 11 Commencer par montrer que $k_n = k$ sur $[\tau_{n,k}; \tau_{n,k+1}[$.
 12 Pour le premier résultat, considérer une primitive de Φ . Pour le second, remarquer que f_n est une fonction en escalier constante sur $[\tau_{n,k}; \tau_{n,k+1}[$.
 14 Montrer de façon similaire à la question 2, que $k_n(t) \underset{n \rightarrow +\infty}{\sim} np$.
 16 Procéder comme à la question 8.
 17 Réunir les résultats des questions 13, 14 et 16, en se rappelant que

$$f_n(t) = f_n(\tau_{n,k_n(t)})$$

Pour le second résultat, on appliquera le théorème de convergence dominée en se ramenant, pour l'hypothèse de domination, à l'étude de p_n faite en Partie II.

- 18 Pour la première convergence, montrer que

$$\int_a^b f_n(t) dt - \int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt \xrightarrow{n \rightarrow \infty} 0$$

Ensuite, comparer les événements $(e_n(a) \leq Y_n \leq e_n(b))$ et $(a \leq Y_n \leq b)$.

Partie IV

- 19 Appliquer l'inégalité de Bienaymé-Tchebychev à l'évènement $(|Y_n| \geq T)$.
 20 Commencer par montrer que, pour tout $\varepsilon > 0$,

$$P(Y_n \leq b) \geq \int_{-\infty}^b \Phi(t) dt - \varepsilon$$

pour n suffisamment grand.

Partie V

- 21 Pour l'existence, utiliser le théorème de changement de variable.

RÉSULTATS PRÉLIMINAIRES

1 La formule de Stirling établit que

$$n! \underset{n \rightarrow +\infty}{\sim} \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

Autrement dit,
$$\lim_{n \rightarrow +\infty} \frac{n!}{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n} = 1$$

Posons ainsi, pour $n \in \mathbb{N}^*$, $\varepsilon_n = \frac{n!}{\sqrt{2\pi n} (n/e)^n} - 1$, de sorte que

La suite $(\varepsilon_n)_{n \in \mathbb{N}^*}$ converge vers 0 et est telle que

$$\forall n \in \mathbb{N}^* \quad n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + \varepsilon_n)$$

L'équivalence donnée par la formule de Stirling pouvait aussi se réécrire

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n + \underset{n \rightarrow +\infty}{o} \left(\sqrt{2\pi n} \left(\frac{n}{e}\right)^n \right)$$

ou encore $\forall n \in \mathbb{N}^* \quad n! = \alpha_n \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

pour une suite $(\alpha_n)_{n \in \mathbb{N}^*}$ convergeant vers 1. Dans les deux cas on s'en sortait également en utilisant la définition d'un petit o ou en posant $\varepsilon_n = \alpha_n - 1$.

2 Rappelons que, par définition des parties entières supérieure et inférieure :

$$\forall y \in \mathbb{R} \quad y - 1 < [y] \leq y \quad \text{et} \quad y \leq [y] < y + 1$$

Pour tout $x \in \mathbb{R}_+^*$,

$$\lambda x + \mu - 1 \leq [\lambda x + \mu] \leq \lceil \lambda x + \mu \rceil \leq \lambda x + \mu + 1$$

d'où
$$1 + \frac{\mu - 1}{\lambda x} \leq \frac{[\lambda x + \mu]}{\lambda x} \leq \frac{\lceil \lambda x + \mu \rceil}{\lambda x} \leq 1 + \frac{\mu + 1}{\lambda x} \quad (\lambda x > 0)$$

En faisant tendre x vers $+\infty$, on obtient par théorème d'encadrement

$$\lim_{x \rightarrow +\infty} \frac{[\lambda x + \mu]}{\lambda x} = 1 \quad \text{et} \quad \lim_{x \rightarrow +\infty} \frac{\lceil \lambda x + \mu \rceil}{\lambda x} = 1$$

Autrement dit,
$$[\lambda x + \mu] \underset{x \rightarrow +\infty}{\sim} \lambda x \quad \text{et} \quad \lceil \lambda x + \mu \rceil \underset{x \rightarrow +\infty}{\sim} \lambda x$$

Pour cette question, on ne traite au brouillon qu'un seul calcul. Par contre, au moment de rédiger au propre, remarquer que l'on peut faire les deux preuves simultanément permet de gagner en concision.

3 La fonction Φ est continue et positive sur $\mathbb{R}$. Vérifions son intégrabilité en $+\infty$ et en $-\infty$. Grâce à la parité de Φ , on se restreint à l'étude en $+\infty$. On remarque que

$$t^2 \Phi(t) \xrightarrow[t \rightarrow +\infty]{} 0 \quad \text{ou encore} \quad \Phi(t) = \underset{t \rightarrow +\infty}{o} (t^{-2})$$

En appliquant le critère de Riemann, $\int_1^{+\infty} dt/t^2$ converge donc par comparaison Φ est intégrable en $+\infty$, d'où

$$\text{L'intégrale } \int_{-\infty}^{+\infty} \Phi(t) dt \text{ converge.}$$

4 Utilisons le développement limité à l'ordre 2 de $\ln(1+x)$:

$$\ln(1+x) = x - \frac{x^2}{2} + o_{x \rightarrow 0}(x^2)$$

qui donne

$$\begin{aligned}\zeta(x) &= (x+1) \ln(x+1) \\ &= (x+1) \left(x - \frac{x^2}{2} + o_{x \rightarrow 0}(x^2) \right) \\ &= x^2 + x - \frac{x^2}{2} + o_{x \rightarrow 0}(x^2)\end{aligned}$$

c'est-à-dire

$$\boxed{\zeta(x) = x + \frac{x^2}{2} + o_{x \rightarrow 0}(x^2)}$$

ÉTUDE ASYMPTOTIQUE D'UNE SUITE

5 Fixons $n \in \mathbb{N}^*$. Comme X_n suit une loi binomiale $\mathcal{B}(n; p)$, on sait que

$$\forall k \in \llbracket 0; n \rrbracket \quad \mathbb{P}(X_n = k) = \binom{n}{k} p^k q^{n-k}$$

et notons que, comme $p > 0$ et $q > 0$, $\mathbb{P}(X_n = k) > 0$ pour tout $k \in \llbracket 0; n \rrbracket$.

Regardons les variations de $k \mapsto \mathbb{P}(X_n = k)$. Soit $k \in \llbracket 0; n-1 \rrbracket$. Il vient

$$\begin{aligned}\frac{\mathbb{P}(X_n = k+1)}{\mathbb{P}(X_n = k)} &= \frac{\binom{n}{k+1} p^{k+1} q^{n-(k+1)}}{\binom{n}{k} p^k q^{n-k}} \\ &= \frac{k! (n-k)!}{(k+1)! (n-(k+1))!} \frac{p}{q} \\ \frac{\mathbb{P}(X_n = k+1)}{\mathbb{P}(X_n = k)} &= \frac{n-k}{k+1} \frac{p}{q}\end{aligned}$$

et on peut alors en déduire que

$$\begin{aligned}\frac{\mathbb{P}(X_n = k+1)}{\mathbb{P}(X_n = k)} \leq 1 &\iff (n-k)p \leq q(k+1) \\ &\iff k(p+q) \geq np - q \\ &\iff k \geq np - q & (p+q=1) \\ &\iff k \geq \lceil np - q \rceil & (k \in \mathbb{N})\end{aligned}$$

$$\frac{\mathbb{P}(X_n = k+1)}{\mathbb{P}(X_n = k)} \leq 1 \iff k \geq x_n$$

Autrement dit, $k \mapsto \mathbb{P}(X_n = k)$ est strictement croissante sur $\llbracket 0; x_n \rrbracket$ et décroissante sur $\llbracket x_n; n \rrbracket$, donc atteint son maximum en x_n . Ainsi,

$$\boxed{\text{Le réel } p_n \text{ est le plus grand élément de } \{\mathbb{P}(X_n = k) \mid k \in \llbracket 0; n \rrbracket\}.$$

Mines Maths 2 MP 2021 — Corrigé

Ce corrigé est proposé par Christophe Fiszka (professeur en CPGE) ; il a été relu par Tristan Poullaouec (professeur en CPGE) et Gilbert Monna (professeur honoraire en CPGE).

Ce sujet propose d'étudier les propriétés des matrices symétriques et des fonctions de matrices symétriques.

- Dans la première partie, on introduit les matrices de permutation en tant que cas particuliers de matrices orthogonales.
- La deuxième partie commence avec le théorème spectral et des polynômes de matrices. Ensuite, on définit deux applications dont on étudie les propriétés via une brève incursion en analyse, avec une étude de convergence simple et uniforme sur les ensembles des applications de I dans $\mathbb{R}$ et de $S_n(I)$ dans $\mathbb{R}$.
- La partie 3 montre que la restriction du rayon spectral ρ à l'espace des matrices symétriques réelles définit une norme. Pour rappel,

$$\forall M \in \mathcal{M}_n(\mathbb{C}) \quad \rho(M) = \max_{\lambda \in \text{sp}(M)} |\lambda|$$

- Les deux dernières parties traitent des propriétés de continuité et de convexité des fonctions de matrices symétriques.

La principale originalité (et difficulté) de ce sujet réside dans la compréhension de l'énoncé. Par exemple, l'application u , définie comme application sur des ensembles d'applications matricielles, est difficile à appréhender. Il fallait être rigoureux dans la manipulation des différents objets mathématiques pour éviter les confusions.

Le sujet comporte plusieurs questions classiques (compacité de $O_n(\mathbb{R})$, polynômes interpolateurs de Lagrange...) ce qui peut en faire un sujet de révision sur la réduction des matrices symétriques, les polynômes de matrices et la topologie des espaces matriciels.

INDICATIONS

Matrices de permutations

- 1 Rappelons la formule du produit matriciel. Si $A \in \mathcal{M}_{n,p}(\mathbb{R})$ et $B \in \mathcal{M}_{p,q}(\mathbb{R})$, alors $AB \in \mathcal{M}_{n,q}(\mathbb{R})$ avec $[AB]_{i,j} = \sum_{k=1}^p [A]_{i,k} [B]_{k,j}$.
- 2 Pour vérifier que $\omega(\sigma)$ est orthogonale, il suffit d'établir que ${}^t\omega(\sigma)\omega(\sigma) = I_n$.
- 4 Considérer une permutation σ pour laquelle $d_i' = d_{\sigma(i)}$ pour tout $i \in \llbracket 1; n \rrbracket$.

Fonctions de matrices symétriques

- 5 C'est une conséquence du théorème spectral.
- 6 Penser aux polynômes interpolateurs de Lagrange.
- 7 Utiliser la question précédente et pour $P \in \mathbb{R}[X]$, montrer que

$$P({}^t\Omega \operatorname{diag}((s_i)_i)\Omega) = {}^t\Omega P(\operatorname{diag}((s_i)_i))\Omega = {}^t\Omega \operatorname{diag}((P(s_i))_i)\Omega$$
- 9 La fonction u est injective mais non surjective dès que $n \geq 2$.
- 10 La réciproque est vraie. Appliquer l'égalité à $S = xI_n$ pour $x \in I$.

Norme et convexité

- 12 Se ramener au cas d'une matrice diagonale à l'aide du théorème spectral.
- 13 Utiliser le résultat précédent pour la convexité et l'inégalité triangulaire avec ρ .

Continuité des fonctions symétriques

- 14 L'application déterminant $\det: \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ est une application polynomiale en les coefficients de la matrice.
- 15 Toute suite bornée d'un espace vectoriel de dimension finie admet une valeur d'adhérence.
- 16 À l'aide de la continuité de χ , justifier que $\chi_M = \chi_{\operatorname{diag}(\lim \Lambda_{\alpha(k)})}$.
- 17 Penser à la caractérisation séquentielle de la continuité. En dimension finie, toute suite bornée avec une unique valeur d'adhérence est convergente.
- 18 Montrer que $O_n(\mathbb{R})$ est fermé en tant qu'image réciproque d'une partie fermée par une application continue.
- 19 Utiliser la même démarche que pour la question 17.

Convexité des fonctions symétriques

- 20 De nouveau, se ramener au cas diagonal. Vérifier que $v(f)(S) = \sum_{i=1}^n f(s_i)$.
- 22 Pour la réciproque, utiliser $A = xI_n$ et $B = yI_n$.

MATRICES DE PERMUTATIONS

1 Soit $(\sigma, \sigma') \in B_n^2$. Par définition du produit matriciel, pour $(i, j) \in \llbracket 1; n \rrbracket^2$

$$[\omega(\sigma)\omega(\sigma')]_{i,j} = \sum_{k=1}^n [\omega(\sigma)]_{i,k} [\omega(\sigma')]_{k,j} = \sum_{k=1}^n \delta_{i,\sigma(k)} \delta_{k\sigma'(j)}$$

Or, le terme $\delta_{k\sigma'(j)}$ vaut 1 si $k = \sigma'(j)$ et 0 sinon. La somme précédente se réduit à un unique terme $\delta_{i,\sigma(\sigma'(j))}$. Ainsi

$$[\omega(\sigma)\omega(\sigma')]_{i,j} = \delta_{i,\sigma\circ\sigma'(j)} = [\omega(\sigma\circ\sigma')]_{i,j}$$

Finalement,

$$\boxed{\omega(\sigma)\omega(\sigma') = \omega(\sigma\circ\sigma')}$$

Soit f_σ l'endomorphisme de $\mathbb{R}^n$ de matrice $\omega(\sigma)$ dans la base canonique de $\mathbb{R}^n$.

$$\forall i \in \llbracket 1; n \rrbracket \quad f_\sigma(e_i) = e_{\sigma(i)}$$

où $(e_1, \dots, e_n)$ désigne la base canonique de $\mathbb{R}^n$. C'est l'endomorphisme qui permute les coordonnées : pour $(x_1, \dots, x_n) \in \mathbb{R}^n$,

$$f_\sigma(x_1, x_2, \dots, x_n) = (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)})$$

Avec σ' une seconde permutation, on trouve

$$\begin{aligned} (f_\sigma \circ f_{\sigma'})(x_1, x_2, \dots, x_n) &= f_\sigma(f_{\sigma'}(x_1, x_2, \dots, x_n)) \\ &= f_\sigma(x_{\sigma'(1)}, x_{\sigma'(2)}, \dots, x_{\sigma'(n)}) \\ &= (x_{\sigma(\sigma'(1))}, x_{\sigma(\sigma'(2))}, \dots, x_{\sigma(\sigma'(n))}) \\ (f_\sigma \circ f_{\sigma'})(x_1, x_2, \dots, x_n) &= f_{\sigma\circ\sigma'}(x_1, x_2, \dots, x_n) \end{aligned}$$

Ainsi, $f_\sigma \circ f_{\sigma'} = f_{\sigma\circ\sigma'}$ et on retrouve $\omega(\sigma)\omega(\sigma') = \omega(\sigma\circ\sigma')$.

2 Soit $\sigma \in B_n$. Soit $(i, j) \in \llbracket 1; n \rrbracket^2$. Alors

$$\begin{aligned} [{}^t\omega(\sigma)\omega(\sigma)]_{i,j} &= \sum_{k=1}^n [{}^t\omega(\sigma)]_{i,k} [\omega(\sigma)]_{k,j} \\ &= \sum_{k=1}^n [\omega(\sigma)]_{k,i} [\omega(\sigma)]_{k,j} \\ &= \sum_{k=1}^n \delta_{k,\sigma(i)} \delta_{k,\sigma(j)} \\ [{}^t\omega(\sigma)\omega(\sigma)]_{i,j} &= \delta_{\sigma(i),\sigma(j)} \end{aligned}$$

Précisons que la condition $\sigma(i) = \sigma(j)$ équivaut à $i = j$ puisque σ est une injection. En résumé,

$$\forall (i, j) \in \llbracket 1; n \rrbracket^2 \quad [{}^t\omega(\sigma)\omega(\sigma)]_{i,j} = \delta_{i,j}$$

De ce fait ${}^t\omega(\sigma)\omega(\sigma) = I_n$ et la matrice $\omega(\sigma)$ est orthogonale.

$$\boxed{\omega(B_n) \subset O_n(\mathbb{R})}$$

Avec les notations de la remarque précédente, f_σ transforme la base canonique, qui est orthonormale, en une autre base orthonormale, puisqu'il ne fait que changer l'ordre des vecteurs, donc c'est un endomorphisme orthogonal. Sa matrice dans la base canonique (orthonormale) est aussi orthogonale.

3 Vérifions que pour tout couple d'indices $(i, j) \in \llbracket 1; n \rrbracket^2$,

$$\left[\text{diag}((d_\ell)_\ell) \omega(\sigma) \right]_{i,j} = \left[\omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell) \right]_{i,j}$$

Soit $(i, j) \in \llbracket 1; n \rrbracket^2$. D'une part

$$\begin{aligned} \left[\text{diag}((d_\ell)_\ell) \omega(\sigma) \right]_{i,j} &= \sum_{k=1}^n \left[\text{diag}((d_\ell)_\ell) \right]_{i,k} \left[\omega(\sigma) \right]_{k,j} \\ &= \left[\text{diag}((d_\ell)_\ell) \right]_{i,i} \left[\omega(\sigma) \right]_{i,j} \quad (\text{car } \left[\text{diag}((d_\ell)_\ell) \right]_{i,k} = 0 \text{ si } k \neq i) \\ \left[\text{diag}((d_\ell)_\ell) \omega(\sigma) \right]_{i,j} &= d_i \delta_{i,\sigma(j)} \end{aligned}$$

$$\begin{aligned} \text{D'autre part } \left[\omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell) \right]_{i,j} &= \sum_{k=1}^n \left[\omega(\sigma) \right]_{i,k} \left[\text{diag}((d_{\sigma(\ell)})_\ell) \right]_{k,j} \\ &= \left[\omega(\sigma) \right]_{i,j} \left[\text{diag}((d_{\sigma(\ell)})_\ell) \right]_{j,j} \\ \left[\omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell) \right]_{i,j} &= \delta_{i,\sigma(j)} d_{\sigma(j)} \end{aligned}$$

Si $i \neq \sigma(j)$, alors $\delta_{i,\sigma(j)} = 0$ et $d_i \delta_{i,\sigma(j)} = 0 = \delta_{i,\sigma(j)} d_{\sigma(j)}$. Sinon $\delta_{i,\sigma(j)} = 1$ et

$$d_i = d_{\sigma(j)} \quad \text{puis} \quad d_i \delta_{i,\sigma(j)} = \delta_{i,\sigma(j)} d_{\sigma(j)}$$

$$\text{Dans tous les cas } \left[\text{diag}((d_\ell)_\ell) \omega(\sigma) \right]_{i,j} = \left[\omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell) \right]_{i,j}$$

d'où

$$\boxed{\text{diag}((d_\ell)_\ell) \omega(\sigma) = \omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell)}$$

4 Raisonnons par double implication. Soient $D = \text{diag}((d_i)_i)$ et $D' = \text{diag}((d'_i)_i)$.

- $i) \Rightarrow ii)$: Supposons que les matrices diagonales D et D' aient les mêmes coefficients diagonaux avec les mêmes occurrences. Dans ce cas, il existe une permutation $\sigma \in B_n$ telle que

$$\forall i \in \llbracket 1; n \rrbracket \quad d'_i = d_{\sigma(i)}$$

Par exemple, pour

$$D = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad \text{et} \quad D' = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

On peut considérer les permutations

$$\sigma \begin{cases} 1 \mapsto 2 \\ 2 \mapsto 1 \\ 3 \mapsto 3 \\ 4 \mapsto 4 \end{cases} \quad \text{ou} \quad \sigma \begin{cases} 1 \mapsto 4 \\ 2 \mapsto 1 \\ 3 \mapsto 3 \\ 4 \mapsto 2 \end{cases}$$

Cet exemple montre qu'il n'y a pas unicité de la permutation dès qu'il y a plusieurs occurrences d'un même coefficient.

D'après la question 3,

$$\text{diag}((d_\ell)_\ell) \omega(\sigma) = \omega(\sigma) \text{diag}((d_{\sigma(\ell)})_\ell) = \omega(\sigma) \text{diag}((d'_\ell)_\ell)$$

Mines Informatique MP-PC-PSI 2021 — Corrigé

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

Le sujet se compose de trois exercices indépendants dont le point commun est le mot « marche » :

- La partie I s'intéresse aux randonnées, prétexte à quelques questions de SQL puis à divers calculs avec les données recueillies par GPS lors d'une balade. La seconde sous-partie se compose de questions de programmation, assez progressives et faisant intervenir des classiques du programme de classes préparatoires, comme la recherche d'un maximum dans une liste.
- La brève partie suivante porte sur la marche brownienne d'une petite particule. La thématique abordée est celle du calcul numérique, principalement autour des schémas d'Euler, et propose une approche originale et élégante par vecteurs d'état.
- Le dernier exercice aborde la marche auto-évitante, c'est-à-dire la formation sur un quadrillage d'un chemin qui ne se recoupe jamais. Ceci permet par exemple de modéliser des chaînes de polymères. L'objectif du sujet était de tester la compréhension d'algorithmes, de les programmer et d'étudier leur complexité. La dernière question, plus ouverte, permettait d'évaluer plus finement les candidats.

Ce sujet original peut être abordé dès la fin de la première année car il ne comporte qu'une question portant sur le programme de seconde année (la question 18, sur la complexité d'un tri). Il fait appel à de nombreuses notions de programmation, nécessite de bien maîtriser les algorithmes de base comme la recherche d'un maximum ou le schéma d'Euler, et est très progressif. C'est un très bon sujet de révision, clair et bien posé, qui permettait de trier efficacement les candidats.

INDICATIONS

- 2 Utiliser **GROUP BY** si cette instruction est connue, sinon utiliser des unions de requêtes.
- 3 Utiliser par exemple une requête imbriquée.
- 4 Réaliser une autojointure, c'est-à-dire la jointure d'une table avec elle-même, en prenant soin de renommer les grandeurs dont on a besoin.
- 5 Attention au passage des éléments en flottants, car ce sont des chaînes de caractères lorsqu'ils sont extraits du fichier.
- 8 Ne pas lésiner sur le nombre d'étapes dans la décomposition, le sujet le demande explicitement ! Attention aussi car le rayon terrestre est une variable globale donnée en kilomètres.
- 9 On peut ici utiliser la fonction de la question 8, même si on n'a pas réussi à la programmer.
- 10 Un exemple d'utilisation et de syntaxe de **assert** est donné dans l'annexe de cette partie.
- 11 Il faut regarder chaque composante du vecteur : certaines sont déjà disponibles, d'autres peuvent être déduites de l'équation différentielle.
- 13 Quelles sont les coordonnées des points voisins d'un point fixé ?
- 14 Il faut trouver un chemin qui entoure le plus vite possible un point, afin que celui-ci devienne forcément un chemin non auto-évitant.
- 16 La complexité recherchée dépend de celle de la fonction **positions_possibles**. Il faut donc tout d'abord étudier celle-ci, en fonction de la taille du paramètre d'entrée **atteints**.
- 19 Supposons que le chemin soit trié, quelle propriété a-t-on alors pour la liste associée si le chemin n'est pas un CAE ?
- 20 Un petit dessin permet de définir les relations entre les coordonnées des points utilisés.
- 22 Appliquer directement l'algorithme décrit dans le sujet.

I. RANDONNÉE

1 On peut proposer par exemple

```
SELECT COUNT(*)  
FROM Participant  
WHERE ne >= 1993 AND ne <= 2003;
```

Il aurait été acceptable de proposer, sous réserve de connaître le mot-clef BETWEEN,

```
SELECT COUNT(*)  
FROM Participant  
WHERE ne BETWEEN 1993 AND 2003;
```

2 Cette deuxième requête demande de réunir les enregistrements de même difficulté afin de calculer par agrégation de leurs durées moyennes. On peut donc proposer

```
SELECT diff, AVG(duree)  
FROM Rando  
GROUP BY diff;
```

Notons qu'il n'est pas clair dans le programme que GROUP BY soit autorisé. Par conséquent, en tirant profit du fait qu'il y a cinq difficultés possibles de randonnée, on peut également proposer la requête suivante, qui utilise l'union de plusieurs requêtes. C'est évidemment beaucoup moins élégant.

```
SELECT diff, AVG(duree) FROM Rando WHERE diff = 1  
UNION  
SELECT diff, AVG(duree) FROM Rando WHERE diff = 2  
UNION  
SELECT diff, AVG(duree) FROM Rando WHERE diff = 3  
UNION  
SELECT diff, AVG(duree) FROM Rando WHERE diff = 4  
UNION  
SELECT diff, AVG(duree) FROM Rando WHERE diff = 5;
```

3 On peut proposer une approche consistant à récupérer l'information de la difficulté de la randonnée 42, afin d'utiliser celle-ci comme requête imbriquée dans une seconde requête ayant pour but d'identifier les bons randonneurs.

```
SELECT pnom  
FROM Participant  
WHERE diff_max < (SELECT diff FROM Rando WHERE rid = 42);
```

Toutefois, on peut être tenté d'effectuer une jointure, qui semble moins naturelle car il n'y a aucun lien de jointure entre les deux tables. Par conséquent, et en l'absence de ce lien, cette jointure s'apparente au produit cartésien des deux tables. On retient alors les noms des randonneurs pour lesquels la difficulté maximale supportée est inférieure à celle de la randonnée 42.

```
SELECT pnom  
FROM Participant JOIN Rando  
WHERE rid = 42 AND diff_max < diff;
```

Il est possible toutefois de faire un lien de jointure qui soit directement une inégalité, solution très proche de la précédente.

```
SELECT pnom
FROM Participant JOIN Rando ON diff_max < diff
WHERE rid = 42;
```

- 4 On veut étudier tous les couples de randonnées ayant le même nom, mais pas le même identifiant, autrement dit des clefs primaires différentes. On est donc obligé de réaliser une autojointure, c'est-à-dire une jointure de la table **Rando** avec elle-même. Une fois cette jointure réalisée, et afin de distinguer les randonnées venant de chaque couple, il faut procéder à un renommage de celles-ci avec le mot-clef **AS**. Enfin, le sujet demande de réunir ces informations en supprimant les doublons, ce que l'on peut faire avec le mot-clef **DISTINCT**. Finalement, on peut proposer la requête suivante.

```
SELECT DISTINCT r1.rid
FROM Rando AS r1 JOIN Rando AS r2 ON r1.rnom = r2.rnom
WHERE r1.rid <> r2.rid;
```

Cette question est particulièrement difficile, car il n'est pas clair que le mot-clef **DISTINCT** soit autorisé et la notion d'auto-jointure est sans doute la plus subtile de celles que l'on peut rencontrer en base de données dans le programme d'informatique de classes préparatoires. Bien comprendre cette question est toutefois très bénéfique pour le SQL en général et il est intéressant de passer le temps nécessaire à bien la saisir.

- 5 On utilise pour cette question ce qui est présenté dans le sujet. On commence par ouvrir le fichier, puis on lit la première ligne afin de s'en débarrasser, car elle ne sert pas dans la suite. On récupère alors les lignes les unes après les autres, en prenant soin à chaque étape d'utiliser la méthode **split** afin d'extraire les quatre données utiles, et enfin en n'oubliant pas de changer de type de variable grâce à la fonction **float**, qui convertit une chaîne de caractères en flottant. On n'oublie pas de fermer le fichier.

```
def importe_rando(nom_fichier):
    # Ouverture de nom_fichier en lecture
    fichier = open(nom_fichier, "r")
    # On lit la première ligne, qui ne sert pas
    fichier.readline()
    # On récupère les lignes suivantes
    lignes = fichier.readlines()
    fichier.close() # Fermeture de l'objet fichier
    coords = [ ]
    for l in lignes: # On parcourt chaque ligne du fichier
        # On découpe la ligne par rapport à la virgule
        interm = l.split(",")
        lat, lon, alt, t = interm # On récupère chacun des 4 termes
        # Puis on les stocke en les convertissant en flottants
        coords.append([float(lat), float(lon), float(alt), float(t)])
    return coords
```

La fermeture de fichier peut s'effectuer dès que l'ensemble des informations désirées est constitué, ce qui est le cas après la fin du **readlines**. Cette fermeture aurait tout aussi bien pu s'effectuer juste avant le **return**.

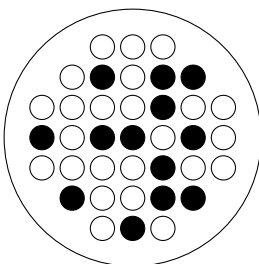
Mines Informatique optionnelle MP 2021

Corrigé

Ce corrigé est proposé par Pacôme Luton (ENS Lyon) ; il a été relu par Titouan Leclercq (ENS Lyon) et Benjamin Monmege (enseignant-chercheur à l'université).

Le sujet est constitué de deux parties indépendantes ayant pour thème la résolution explicite du jeu du solitaire quand cela est possible, et la reconnaissance d'un plateau de jeu résoluble.

- La partie I aborde l'implémentation d'un programme ayant pour objet la résolution du jeu du solitaire sur le tablier européen.



Les dix premières questions utilisent une manière habile de représenter un plateau de jeu avec des entiers, permettant une implémentation efficace du programme. Puis de la question 11 à 17, on utilise des parcours de graphe avec dictionnaires, ce qui permet là aussi une implémentation efficace. Cette partie demande une bonne maîtrise des listes d'OCaml et une bonne compréhension des parcours de graphe.

- La partie II s'intéresse à la caractérisation des parties gagnantes d'une version simplifiée du jeu de solitaire, où l'on joue sur un segment. Contrairement à la première partie, celle-ci est essentiellement théorique. Elle commence par trois questions relativement simples traitant d'un langage local. On étudie ensuite une manière efficace de reconnaître les parties gagnantes à l'aide d'automates locaux. La fin de cette partie est aussi technique que difficile.

Ce sujet aborde des problématiques classiques et importantes en algorithmique de graphe. Il propose aussi une approche originale pour résoudre un problème algorithmique, en construisant pas à pas un automate reconnaissant les parties gagnantes.

INDICATIONS

Partie 1

- 3 Énumérer à l'aide d'une fonction auxiliaire récursive les entiers candidats et regarder un à un s'ils correspondent à un numéro intérieur.
- 6 Parcourir récursivement la liste en entrée et ajouter les encoches au motif. Pour cela s'inspirer du petit exemple donné dans l'énoncé.
- 7 Composer les résultats des questions 3 et 6.
- 8 Décrire l'écriture en base 2 de $\mathbf{m}$ **and** $(\mathbf{m}-1)$ à partir de celle de $\mathbf{m}$.
- 9 Utiliser la fonction **land** pour vérifier si pour un motif m , un certain bit $[m]_z$ est non nul.
- 10 Utiliser la question 4, et ne pas oublier de vérifier si la position candidate appartient bien au plateau de jeu.
- 11 Utiliser la liste **voisins** et remarquer qu'il est facile de transformer la grille après un coup grâce à l'opérateur **lxor**.
- 12 Faire un parcours en profondeur dans le graphe des configurations (m, p) où m est un motif représentant la grille et p le motif ponctuel représentant la position du fichet à déplacer.
- 13 Utiliser une fonction auxiliaire pour énumérer tous les fichets du motif qui peuvent être déplacés, et récupérer la liste des motifs qu'il est possible d'obtenir en déplaçant ce fichet en un coup composé à l'aide de la fonction de la question 12.
- 15 Introduire une fonction qui, à un motif, associe la liste de tous les nouveaux motifs accessibles en un coup composé supplémentaire.
- 16 Énumérer récursivement la liste de tous les motifs qu'il est possible d'obtenir en n étapes.
- 17 Regarder si l'implémentation de la question 16 respecte bien tous les aspects du parcours décrit à la question 1.

Partie 2

- 19 Un automate à deux états suffit.
- 20 Trouver les facteurs de taille 2 possibles et faire un raisonnement par l'absurde.
- 21 Séparer la somme en deux pour faire disparaître les valeurs absolues. Utiliser ensuite la majoration de la somme infinie des ϕ^i qui est donnée par l'énoncé.
- 22 Remarquer que le calcul de $V_k(w_i) - V_k(w_{i+1})$ n'est pas si compliqué, puis faire une disjonction de cas.
- 24 Reasonner par l'absurde et utiliser les questions 21, 22 et 23.
- 25 Comparer la trace de la partie $W = (w_0, w_1, \dots, w_m)$ avec celle de la partie $W' = (w_1, w_2, \dots, w_m)$. Donner une procédure pour passer de W à W' , puis raisonner par récurrence.
- 26 Utiliser la question 24.
- 27 Construire une procédure récursive qui permet de caractériser les facteurs de longueur 2.
- 29 Construire un automate dont la suite d'états après la lecture d'un mot x permet d'explicitier la trace d'une partie où le premier mot de la partie est x et le dernier un mot ponctuel.

I. PARTIE JOUÉE SUR LE TABLIER EUROPÉEN EN UN MINIMUM DE COUPS

1 Pour déterminer la distance entre deux sommets d'un graphe, il faut faire un **parcours en largeur**. La politique de mise en attente des sommets qui caractérise ce parcours est la politique FIFO (First In First Out). On traite en priorité les sommets découverts les plus tôt lors du parcours. On l'implémente à l'aide d'une file.

L'autre politique au programme de CPGE de mise en attente des sommets est LIFO (Last In First Out), qui correspond au parcours en profondeur, que l'on implémente avec une pile.

2 Il suffit de vérifier toutes les conditions données dans la définition de l'ensemble $\mathcal{E}$.

```
let numero_interieur z =
  let x = z mod 8 and y = z/8 in
  x >= 0 && x <= 6 && y >= 0 && y <= 6
  && abs (x - 3) + abs (y - 3) <= 4;;
```

3 La fonction auxiliaire `enumerateur` permet de parcourir récursivement les entiers de 0 à 52 inclus, et les ajoute à la sortie du programme s'ils appartiennent à $\mathcal{E}$.

```
let numeros_europeens =
  let rec enumerateur i = match i with
    | 53 -> []
    | i when numero_interieur i -> i::(enumerateur (i+1))
    | _ -> enumerateur (i+1)
  in enumerateur 0;;
```

4 Considérons l'encoche située à la position (x, y) avec $z = 8y + x$ le numéro associé. Les encoches voisines, dont on suppose l'existence, ont pour coordonnées $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$ et $(x, y + 1)$. Elles sont donc représentées par les numéros

$$z - 1, z + 1, z - 8 \text{ et } z + 8$$

5 Par définition, le motif qui contient seulement l'encoche du numéro z est représenté par l'entier 2^z . Ainsi, la fonction doit juste renvoyer 2^z , que l'on peut obtenir directement à l'aide de la fonction de décalage `lsl` rappelée dans l'énoncé.

```
let numero_vers_ponctuel z = (1 lsl z);;
```

6 La fonction auxiliaire `parcours_liste` parcourt récursivement les éléments de la liste des numéros des encoches, et ajoute leur représentation en tant que motif ponctuel au motif qui sera renvoyé. Si z désigne le numéro d'une encoche à ajouter, l'instruction $2^z \text{ lor } m$ permet de faire passer le z -ième bit de m à 1.

```
let numeros_vers_motif l =
  let rec parcours_liste l = match l with
    | [] -> 0
    | t::q -> (numero_vers_ponctuel t) lor (parcours_liste q)
  in parcours_liste l;;
```

Si on supposait que la liste ne contenait pas de doublons, une somme des puissances de 2 associées aux entiers de la liste conviendrait également. L'utilisation du OU logique permet de ne pas s'embêter avec les doublons car pour tout entier z , $2^z \text{ lor } 2^z = 2^z$. À cela s'ajoute le fait que l'énoncé suggérerait l'utilisation du `lor` pour cette fonction avec la définition de m_0 .

7 Il suffit d'appliquer la fonction `numeros_vers_motif` définie à la question 6 à la variable globale `numeros_europeens` de la question 3.

```
let motif_europeen = numeros_vers_motif numeros_europeens;;
```

8 Une manière de comprendre le test effectué dans la fonction est de le tester à la main sur des exemples de petite taille. Par exemple, en notant f la fonction qui à m associe $m \text{ land } (m - 1)$, on a :

$$f(\overline{10110}) = \overline{10110} \text{ land } (\overline{10110} - 1) = \overline{10110} \text{ land } \overline{10101} = \overline{10100}$$

$$f(\overline{01000}) = \overline{01000} \text{ land } (\overline{01000} - 1) = \overline{01000} \text{ land } \overline{00111} = \overline{00000}$$

On peut alors conjecturer que $f(m)$ passe le bit non nul de poids le plus faible de m à 0. Ce que nous allons à présent formaliser et exploiter.

Soit $m \in \mathbb{Z}$. Si $m \leq 0$, par définition m ne peut pas représenter un motif (donc pas un motif ponctuel). Dans ce cas, le test $m > 0$ s'assure que la fonction `est_ponctuel` renvoie `false`. Considérons à présent $m \in \mathbb{N}^*$. Cet entier admet une unique décomposition de la forme

$$m = \sum_{i \in I} 2^i$$

avec I une partie finie et non vide de $\mathbb{N}$. Par propriété de $\mathbb{N}$, cette partie admet un plus petit élément, que l'on note i_0 . On a alors

$$m - 1 = \sum_{i < i_0} 2^i + \sum_{i \in I \setminus \{i_0\}} 2^i$$

Cette formule n'a rien de bien compliqué, c'est juste l'écriture en binaire du nombre $m - 1$. La soustraction en base 2 fonctionne exactement comme la soustraction classique en base 10 posée à la main.

Notons $n = m \text{ land } (m - 1)$. Soit $i \in \llbracket 0; 62 \rrbracket$. Il y a quatre cas à considérer :

- Si $i < i_0$, $[m]_i = 0$ par minimalité de i_0 et $[m - 1]_i = 1$ d'après l'écriture précédente, soit $[n]_i = 0 \wedge 1 = 0$.
- Si $i = i_0$, $[n]_i = [m]_i \wedge [m - 1]_i = 1 \wedge 0 = 0$.
- Si $i \in I$ et $i > i_0$, $[n]_i = [m]_i \wedge [m - 1]_i = 1 \wedge 1 = 1$.
- Si $i \notin I$ et $i > i_0$, $[n]_i = [m]_i \wedge [m - 1]_i = 0 \wedge 0 = 0$.

On en déduit que
$$m \text{ land } (m - 1) = \sum_{i \in I \setminus \{i_0\}} 2^i$$

Cette somme est nulle si et seulement si $I \setminus \{i_0\} = \emptyset$. Autrement dit si et seulement si I contient exactement un entier. En conclusion,

La fonction `est_ponctuel` m renvoie `true` si m représente un motif ponctuel et `false` sinon.

X/ENS Maths A MP 2021 — Corrigé

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

L'objet de ce sujet est l'étude des sous-groupes finis de $\mathrm{GL}_n(\mathbb{Z})$, l'ensemble des matrices inversibles à coefficients entiers dont l'inverse est encore à coefficients entiers. Il traite plus particulièrement de l'étude des cardinaux possibles pour les sous-groupes finis de $\mathrm{GL}_n(\mathbb{Z})$.

Comme le présente l'énoncé lui-même, le but est de démontrer, pour tout entier $n \in \mathbb{N}^*$, l'existence d'une borne (ne dépendant que de n) pour le cardinal des sous-groupes finis de $\mathrm{GL}_n(\mathbb{Z})$, d'en expliciter une, et d'en donner une majoration raffinée dans le cas d'un p -sous-groupe, c'est-à-dire d'un sous-groupe dont le cardinal est une puissance d'un nombre premier p .

Les trois premières parties sont indépendantes. La quatrième est largement indépendante des autres, mais utilise le résultat de la dernière question de la troisième partie.

- Quelques questions préliminaires de réduction et d'arithmétique utiles tout au long du sujet permettent un début d'épreuve facile.
- Dans la partie I, on s'intéresse aux éléments d'ordre fini de $\mathrm{GL}_n(\mathbb{Z})$. Si $n = 2$, on montre qu'il n'y a que 5 ordres possibles, les entiers 1, 2, 3, 4 et 6. On établit plus généralement que l'ensemble des ordres possibles dans $\mathrm{GL}_n(\mathbb{Z})$ est fini.
- Dans la partie II, on prouve que tout sous-groupe fini de $\mathrm{GL}_n(\mathbb{Z})$ est de cardinal inférieur à 3^{n^2} .
- La troisième partie s'attache à démontrer que la trace de toute matrice d'un p -sous-groupe quelconque de $\mathrm{GL}_n(\mathbb{Z})$ ne peut prendre qu'un nombre fini de valeurs parmi un ensemble décrit en fonction de n et p uniquement :

$$\{n - pj \mid 0 \leq j \leq \lfloor n/(p-1) \rfloor\}$$

- Dans la quatrième et dernière partie, on se sert d'une loi appelée *produit de Kronecker* sur les matrices pour raffiner la majoration de la partie II. On établit notamment, pour tout sous-groupe G de $\mathrm{GL}_n(\mathbb{Z})$ de cardinal p^r avec p premier, et pour tout entier $s \in \mathbb{N}^*$,

$$\sum_{g \in G} (\mathrm{Tr} \, g)^s \text{ est un entier divisible par } p^r$$

On en déduit que le cardinal de tout p -sous-groupe de $\mathrm{GL}_n(\mathbb{Z})$ est majoré par 4^n .

Ce sujet aborde principalement les thématiques d'algèbre linéaire générale, polynômes, groupes, réduction, et arithmétique. Nombre de questions sont très classiques et aucune ne comporte de difficulté insurmontable pour tout candidat solidement préparé sur les thèmes évoqués. Il fallait cependant une grande aisance et du recul sur ces notions d'algèbre pour traiter tout le problème dans le temps imparti.

INDICATIONS

Preliminaires

- P.3.a Expliciter l'ensemble en question pour le mettre en bijection avec un sous-ensemble de $\mathbb{N}$.
- P.3.b Commencer par dénombrer les entiers de $\llbracket 1 ; m \rrbracket$ qui sont multiples de q^i , mais non multiples de q^{i+1} .

Partie 1

- 1.1 La trace d'une matrice est la somme de ses valeurs propres complexes.
- 1.2 Quels nombres complexes de module 1 sont réels ?
- 1.3 Considérer l'expression du polynôme caractéristique pour une matrice carrée de taille 2. On pourra d'abord justifier que $\det g = 1$.
- 1.4 Quelles sont les valeurs propres possibles pour une matrice ayant un polynôme caractéristique parmi ceux de la question 1.3 ?
- 1.5 Utiliser les relations coefficients-racines pour les polynômes scindés.
- 1.6 Se servir du résultat de la question 1.5 en remarquant que toutes les valeurs propres d'une matrice d'ordre fini sont de module 1.
- 1.7 Montrer que l'ordre de tout élément g de $\mathrm{GL}_n(\mathbb{Z})$ d'ordre fini est un diviseur d'un entier ω indépendant de g , construit en utilisant les racines des polynômes de la question 1.6.

Partie 2

- 2.1.a Pour l'inégalité, exprimer les valeurs propres de A en fonction de celles de g , puis appliquer l'inégalité triangulaire gauche.
- 2.1.b Montrer d'abord que $A^k \xrightarrow[k \rightarrow +\infty]{} 0$ pour la norme infinie.
- 2.1.c La matrice g est diagonalisable ; quel est son spectre ?
- 2.2.a Si a et b dans G ont la même réduction modulo m , montrer que $ab^{-1} = I_n$ en utilisant la question 2.1.

Partie 3

- 3.1.a Remarquer en premier lieu que ℓ divise $k! \binom{\ell}{k}$.
- 3.1.b Comment développer une somme de termes qui commutent mise à une puissance entière ?
- 3.2 Utiliser la formule sommatoire du déterminant, et développer un produit du type $\prod_{i=1}^n (a_i + b_i)$.
- 3.3 Commencer par appliquer le résultat de 3.2 avec $R = \mathbb{Z}[X]$ et $P = \sum_{i=0}^n a_i X^i$, puis utiliser le petit théorème de Fermat pour conclure.
- 3.4.a Pour $\ell \geq 3$ premier, alors ℓ est impair : considérer le résultat de la question 3.1.b.
- 3.4.b Se servir du résultat de la question 3.2.
- 3.4.c Relier la trace au polynôme caractéristique et décomposer $\chi_{M^\ell}(X^\ell) - \chi_M(X^\ell)$ en accord avec les résultats des questions 3.3 et 3.4.b.

- 3.5 Utiliser le résultat de la question 3.4.c avec $\ell = p$.
- 3.6 Prouver que $\text{Tr } g \in \llbracket -n ; n \rrbracket$ pour tout $g \in G$.
- 3.7.a Raisonner par l'absurde et distinguer les cas suivant qu'un diviseur ℓ de m vérifiant $\ell \leq 2n$ divise k ou non.
- 3.7.b Noter que $g^m = g^k$ pour tout $g \in G$, puis considérer un diviseur premier de m avec le résultat de la question 3.7.a.
- 3.8.a Remarquer que J_r est un simple intervalle d'entiers privé des multiples de p .
- 3.8.b Écrire la somme à calculer en utilisant le résultat de la question 3.8.a.
- 3.9 Les valeurs propres de g sont des racines p^r -ièmes de l'unité. Utiliser le résultat de la question 3.7.b pour tout $j \in J_r$, et sommer les traces de g^j en se souvenant de la question 3.8.b.
- 3.10 Utiliser le résultat de la question 3.5, puis celui de la question 3.9, en notant que n_0 et n_1 sont inférieurs à n .

Partie 4

- 4.1.a Commencer par calculer γf pour tout $\gamma \in G$.
- 4.1.b Que vaut la trace d'un projecteur ?
- 4.2.i Expliciter la diagonale de $g \otimes h$.
- 4.2.ii Faire le produit par blocs de taille (k, k) pour $(g \otimes h)(g' \otimes h')$.
- 4.2.iii Utiliser le résultat donné par la question 2.ii.
- 4.3.a Distinguer les cas suivant que $\gamma' \in \text{Im } \varphi$ ou non.
- 4.3.b Utiliser la question 3.a en considérant des éléments deux à deux distincts composant $\text{Im } \varphi$. Quel est le cardinal de γH pour tout $\gamma \in G$?
- 4.4.a Raisonner par récurrence en utilisant la question 2.ii. Regrouper ensuite les images égales par φ_s , pour calculer $\sum_{g \in G} \varphi_s(g)$.
- 4.4.b Se servir du résultat démontré à la question 4.3.b.
- 4.5.a Utiliser la question 3.10 et déterminer l'ensemble des $g \in G$ dont la trace est n pour exprimer la somme en fonction de $P(n)$. Ensuite, appliquer le résultat de la question 4.4.b, en examinant à part le cas $s = 0$.
- 4.5.b Calculer $P(n)$ en fonction de a .
- 4.6.a Faire appel à la question préliminaire P.3.b.
- 4.6.b Optimiser la majoration en $p \geq 2$ donnée par la question 4.6.a en étudiant la fonction argument du cardinal p^r mis sous forme exponentielle.

PRÉLIMINAIRES

P.1 Soient $z \in \mathbb{C}$ une racine de l'unité et $d \in \mathbb{N}^*$ tel que $z^d = 1$. Alors d'après la description des racines d -ièmes de l'unité, il existe $k \in \llbracket 0; d-1 \rrbracket$ tel que $z = e^{i2k\pi/d}$ et donc immédiatement

$$\forall z \in \mathbb{C} \quad \forall d \in \mathbb{N}^* \quad (z^d = 1 \implies |z| = 1)$$

P.2 Soit $g \in \mathrm{GL}_n(\mathbb{C})$ d'ordre $d \in \mathbb{N}^*$. L'élément neutre du groupe $\mathrm{GL}_n(\mathbb{C})$ étant I_n , il vient donc $g^d = I_n$, de sorte que $P = X^d - 1$ est un polynôme scindé à racines simples qui annule g . D'après le cours, on en déduit que

$$\text{Si } g \in \mathrm{GL}_n(\mathbb{C}) \text{ est d'ordre fini } d \in \mathbb{N}^*, \text{ alors } g \text{ est diagonalisable et } \mathrm{sp}(g) \subset P^{-1}(\{0\}) = \mathbb{U}_d.$$

P.3.a Soient $m \in \mathbb{N}$ et $q \in \mathbb{N}^*$. Alors

$$\begin{aligned} \{k \in \llbracket 1; m \rrbracket \mid q \text{ divise } k\} &= \{pq \mid p \in \mathbb{N} \text{ et } 1 \leq pq \leq m\} \\ &= \{pq \mid p \in \mathbb{N} \cap \llbracket 1; m/q \rrbracket\} \\ \{k \in \llbracket 1; m \rrbracket \mid q \text{ divise } k\} &= \{pq \mid p \in \llbracket 1; \lfloor m/q \rfloor \rrbracket\} \end{aligned}$$

L'application $p \mapsto pq$ induit donc une surjection de $\llbracket 1; \lfloor m/q \rfloor \rrbracket$ sur l'ensemble

$$\{k \in \llbracket 1; m \rrbracket \mid q \text{ divise } k\}$$

Étant trivialement injective, il vient

$$\forall m \in \mathbb{N} \quad \forall q \in \mathbb{N}^* \quad \mathrm{Card} \{k \in \llbracket 1; m \rrbracket \mid q \text{ divise } k\} = \lfloor m/q \rfloor$$

P.3.b Soit q un nombre premier. Remarquons déjà que $\sum_i \lfloor m/q^i \rfloor$ converge : en effet c'est une somme finie puisque le terme général de cette série est nul pour $q^i > m$, c'est-à-dire $i > \ln m / \ln q$. Soit $i \in \mathbb{N}^*$. D'après le résultat de la question P.3.a, le nombre de multiples de q^i mais non multiples de q^{i+1} dans $\llbracket 1; m \rrbracket$ est

$$n_i = f_i - f_{i+1} \quad \text{avec} \quad f_i = \left\lfloor \frac{m}{q^i} \right\rfloor$$

En effet, en notant $E_i = \{k \in \llbracket 1; m \rrbracket \mid v_q(k) = i\}$ et $F_i = \{k \in \llbracket 1; m \rrbracket \mid v_q(k) \geq i\}$ pour tout $i \in \mathbb{N}$

alors
$$\llbracket 1; m \rrbracket = \bigsqcup_{i=0}^{+\infty} E_i = \bigsqcup_{i=0}^{+\infty} (F_i \setminus F_{i+1}) \quad (\text{union disjointe})$$

Comme pour tout i on a $F_{i+1} \subset F_i$, cela prouve le cardinal annoncé.

Ces entiers multiples de q^i mais pas de q^{i+1} contribuent chacun pour i dans la valuation q -adique de $m!$. Il reste à sommer pour obtenir toutes les contributions q -adiques des entiers entre 1 et m dans la valuation de $m!$: par linéarité pour les sommes suivantes qui sont toutes finies, et changement d'indice, on trouve

X Maths B MP 2021 — Corrigé

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

Ce sujet tourne autour d'une loi de probabilité construite sur la fonction zeta de Riemann.

- Dans la partie I, qui demande d'être au point sur les cours d'arithmétique de MPSI et de probabilités de MP, on étudie une variable aléatoire X suivant la loi zeta de paramètre $s > 1$, caractérisée par

$$\forall k \in \mathbb{N}^* \quad \mathbb{P}(X = k) = \zeta(s)^{-1} k^{-s} \quad \text{où} \quad \zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s}$$

En utilisant deux fonctions arithmétiques multiplicatives particulières (notées g et χ_4 dans le sujet), on établit entre autres la formule du produit eulérien pour la fonction ζ : si $(p_k)_{k \in \mathbb{N}^*}$ est la suite des nombres premiers,

$$\frac{1}{\zeta(s)} = \prod_{k=1}^{+\infty} \left(1 - \frac{1}{p_k^s}\right)$$

- La partie II traite d'une famille de polynômes voisine des polynômes de Tchebychev afin d'établir la formule

$$\sin(\pi x) = \pi x \prod_{k=1}^{+\infty} \left(1 - \frac{x^2}{k^2}\right)$$

Cette partie n'utilise que le programme de MPSI.

- La partie suivante définit la fonction Γ comme limite simple d'une suite de fonctions, et en établit la caractérisation comme l'unique fonction log-convexe telle que $\Gamma(1) = 1$ et satisfaisant l'équation $\Gamma(x+1) = x \Gamma(x)$, afin d'en déduire les formules suivantes :

$$\frac{\Gamma(x) \Gamma(a)}{\Gamma(x+a)} = \int_0^{+\infty} \frac{t^{x-1}}{(1+t)^{x+a}} dt$$

$$\frac{\pi}{\sin(\pi x)} = \int_0^{+\infty} \frac{t^{x-1}}{1+t} dt$$

- La partie IV étudie une suite de rationnels, voisine de celle des nombres de Bernoulli, intervenant dans le développement en série entière de $1/\cos x$ et qui permettent d'obtenir des expressions exactes des sommes de séries

$$\sum_{n=0}^{+\infty} \frac{(-1)^n}{(2n+1)^{2k+1}} \quad \text{avec } k \in \mathbb{N}$$

Ce sujet est trop long pour être traité en totalité dans le temps imparti ; sa difficulté est en adéquation avec la sélectivité du concours.

Il établit une panoplie de résultats très plaisants, tout en balayant une large partie du programme : probabilités, arithmétique dans $\mathbb{Z}$, théorème de convergence dominée et corollaires, calcul algébrique dans $\mathbb{C}$, polynômes, convexité des fonctions de la variable réelle, familles sommables.

Pour ces raisons, il constitue un bon sujet de révision avant les écrits.

INDICATIONS

- 1.a Revenir à la définition de la divisibilité afin d'écrire l'événement $\{n \mid X\}$ comme réunion disjointe dénombrable d'événements de référence pour X .
- 1.b Commencer par rappeler pourquoi si $n = p_{k_1}^{\alpha_1} \cdots p_{k_\ell}^{\alpha_\ell}$ (décomposition en facteurs premiers) et x un entier, alors n divise x si et seulement si pour tout j compris entre 1 et ℓ , $p_{k_j}^{\alpha_j}$ divise x .
- 2.a Les événements $\{p_j^{\alpha_j} \nmid X\}$ sont mutuellement indépendants.
- 2.b Quel est l'unique entier positif qui ne possède aucun diviseur premier ?
- 3.a Calculer pour commencer la probabilité des événements $\{\nu_{p_k}(X) \geq \alpha\}$.
- 3.b Étudier et comprendre ce qui se passe pour les petites valeurs de r .
- 3.c Commencer par constater l'indépendance des événements $\{\nu_{p_{k_j}}(X) \geq n_j\}$.
- 4.a Commencer par établir que l'application $(h, k) \mapsto hk$ induit une bijection de l'ensemble des couples (diviseur de m , diviseur de n) vers l'ensemble des diviseurs de mn .
- 4.b Identifier les diviseurs de p^n , et s'intéresser à leur congruence modulo 4.
- 5 Penser au théorème de la double limite.
- 6.a Faire un calcul de familles sommables.
- 7.b Utiliser les résultats des questions 4.a, 7.a et 5.
- 8.a Utiliser les résultats des questions 3.a et 4.b.
- 8.b Appliquer le théorème de transfert avec la variable aléatoire $\nu_p(X) + 1$ qui suit une loi géométrique.
- 10.a Utiliser la formule du binôme, et séparer la somme selon la parité des indices obtenus.
- 10.b Le polynôme P_n a le bon nombre de racines évidentes. En déduire une factorisation de P_n comprenant le terme $P_n(0)$ que l'on calculera en utilisant l'énoncé de la question 10.a.
- 11.a Que vaut le produit $u_{m,n}(x) v_{m,n}(x)$?
- 11.b Utiliser la concavité de la fonction $\sin$ sur l'intervalle $[0; \pi/2]$.
- 12 Étudier la série de terme général $(\ln \Gamma_n(x) - \ln \Gamma_{n-1}(x))$.
- 13 Déterminer, lorsque n tend vers l'infini, un équivalent à x fixé de $\frac{x \Gamma_n(x)}{\Gamma_n(x+1)}$.
- 14.a Appliquer le théorème de dérivation d'une limite de fonctions à la suite $(\Gamma_n)_{n \in \mathbb{N}}$.
- 14.b Faire un décalage d'indice judicieux (sans oublier que x n'est pas supposé être un entier), ou appliquer le théorème de la double limite.
- 15.a Pour la convexité de S , constater que $S''(x) = S''(x+n)$ pour tout $n \in \mathbb{N}$.
- 16 Utiliser les questions précédentes et une intégration par parties.
- 17 Appliquer le résultat de la question 16 avec $a = 1 - x$, et utiliser la suite de fonctions $(\Gamma_n)_{n \in \mathbb{N}}$.
- 18.a Couper l'intégrale en deux, et appliquer le théorème de convergence dominée aux suites des sommes partielles des séries de fonctions.
- 18.b Appliquer le résultat de la question 18.a avec $x + 1/2$. Pour l'interversion des sommes, traiter à part les indices où $k = 0$.
- 19.a Remarquer que, pour tout x , $v(x) \cos x = 1$, et utiliser l'unicité du développement en série entière.
- 19.b Utiliser les résultats des questions 9.c, 18.c et 19.a.

PARTIE I

1.a | Dans toute la suite, on note $\mathbb{P}$ la probabilité.

Rappelons que si x est un entier, alors n divise x si et seulement si il existe $k \in \mathbb{Z}$ tel que $x = kn$. De plus, si $x > 0$ et $n > 0$, on a alors $k \in \mathbb{N}^*$. Enfin, un tel k est unique ($k = x/n$). Il s'ensuit que l'événement $\{n \mid X\} = \{\omega \in \Omega ; n \mid X(\omega)\}$ s'écrit comme réunion disjointe (dénombrable) d'événements :

$$\{n \mid X\} = \bigcup_{k=1}^{\infty} \{X = kn\}$$

Par σ -additivité, il vient alors, puisque X suit la loi zeta de paramètre s :

$$\mathbb{P}(n \mid X) = \sum_{k=1}^{+\infty} \mathbb{P}(X = kn) = \sum_{k=1}^{+\infty} \frac{1}{\zeta(s)(kn)^s} = \frac{1}{n^s \zeta(s)} \sum_{k=1}^{+\infty} \frac{1}{k^s}$$

soit, en simplifiant,

$$\boxed{\mathbb{P}(n \mid X) = 1/n^s}$$

1.b | Montrer que les événements A_i , $i \in \mathbb{N}^*$, sont mutuellement indépendants, c'est montrer que pour tout $k \in \mathbb{N}^*$ et tout k -uplet $(i_1, i_2, \dots, i_k) \in (\mathbb{N}^*)^k$ tel que $i_1 < i_2 < \dots < i_k$,

$$\mathbb{P}(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k}) = \mathbb{P}(A_{i_1})\mathbb{P}(A_{i_2}) \dots \mathbb{P}(A_{i_k})$$

En notant $J = \{i_1; i_2; \dots; i_k\}$, montrer que les A_i sont mutuellement indépendants revient donc à montrer que pour toute partie finie J de $\mathbb{N}^*$, on a

$$\mathbb{P}\left(\bigcap_{i \in J} A_i\right) = \prod_{i \in J} \mathbb{P}(A_i)$$

On adopte ici cette notation afin de limiter les indices imbriqués.

Soit J une partie finie de $\mathbb{N}^*$. Comme les p_j sont des nombres premiers deux à deux distincts, les $p_j^{\alpha_j}$ sont en particulier deux à deux premiers entre eux. Un résultat du cours d'arithmétique stipule que, si x est un entier positif, alors $\prod_{j \in J} p_j^{\alpha_j}$ divise x si, et seulement si, pour tout $j \in J$, $p_j^{\alpha_j}$ divise x . De fait,

$$\bigcap_{j \in J} \{p_j^{\alpha_j} \mid X\} = \left\{ \prod_{j \in J} p_j^{\alpha_j} \mid X \right\}$$

Il en découle que $\mathbb{P}\left(\bigcap_{j \in J} \{p_j^{\alpha_j} \mid X\}\right) = \mathbb{P}\left(\prod_{j \in J} p_j^{\alpha_j} \mid X\right)$ par ce qui précède

$$= \left(\prod_{j \in J} p_j^{\alpha_j}\right)^{-s} \quad (\text{question 1.a})$$

$$= \prod_{j \in J} (p_j^{\alpha_j})^{-s}$$

$$\mathbb{P}\left(\bigcap_{j \in J} \{p_j^{\alpha_j} \mid X\}\right) = \prod_{j \in J} \mathbb{P}(p_j^{\alpha_j} \mid X) \quad (\text{question 1.a})$$

Ainsi,

$\boxed{\text{Les événements } \{p_j^{\alpha_j} \mid X\} \text{ sont mutuellement indépendants.}}$

2.a Comme les $\{p_j^{\alpha_j} \mid X\}$ sont mutuellement indépendants, la famille des complémentaires $\{p_j^{\alpha_j} \nmid X\}$ est aussi une famille d'événements mutuellement indépendants.

Par conséquent,
$$\mathbb{P}\left(\bigcap_{i=1}^r \{p_i \nmid X\}\right) = \prod_{i=1}^r \mathbb{P}(p_i \nmid X)$$

Or justement, $\{p_i \nmid X\}$ étant l'événement complémentaire de $\{p_i \mid X\}$, on a, en utilisant la question 1.a, $\mathbb{P}(p_i \nmid X) = 1 - \mathbb{P}(p_i \mid X) = 1 - p_i^{-s}$. Finalement,

$$\mathbb{P}\left(\bigcap_{i=1}^r \{p_i \nmid X\}\right) = \prod_{i=1}^r (1 - p_i^{-s})$$

2.b Tout entier strictement supérieur à 1 possède au moins un diviseur premier. Autrement dit, 1 est le seul entier strictement positif n'ayant aucun diviseur premier. Par conséquent :

$$\bigcap_{k \in \mathbb{N}^*} \{p_k \nmid X\} = \{X = 1\}$$

Par continuité décroissante, il vient

$$\prod_{k=1}^n (1 - p_k^{-s}) = \mathbb{P}\left(\bigcap_{k=1}^n \{p_k \nmid X\}\right) \xrightarrow{n \rightarrow +\infty} \mathbb{P}(X = 1) = \frac{1}{\zeta(s)}$$

On a bien

$$\frac{1}{\zeta(s)} = \lim_{n \rightarrow +\infty} \prod_{k=1}^n (1 - p_k^{-s})$$

| Cette formule établie sur la fonction ζ est appelée *produit eulérien*.

3.a Tout d'abord, ν_p étant une application de $\mathbb{N}^*$ vers $\mathbb{N}$, et X étant une variable aléatoire à valeurs dans $\mathbb{N}^*$, $\nu_p(X)$ est une variable aléatoire à valeurs dans $\mathbb{N}$, donc

La variable aléatoire $\nu_p(X) + 1$ est à valeurs dans $\mathbb{N}^*$.

Fixons $\alpha \in \mathbb{N}$. Pour $x \in \mathbb{N}^*$, et p nombre premier, constatons que, d'une part, on a $\nu_p(x) + 1 \geq \alpha$ si et seulement si $p^{\alpha-1}$ divise x . Il s'ensuit que

$$\{\nu_{p_k}(X) + 1 \geq \alpha\} = \{p_k^{\alpha-1} \mid X\}$$

D'autre part, on a également la décomposition en union disjointe suivante

$$\{\nu_{p_k}(X) + 1 \geq \alpha\} = \{\nu_{p_k}(X) + 1 = \alpha\} \cup \{\nu_{p_k}(X) + 1 \geq \alpha + 1\}$$

On a donc $\mathbb{P}(\nu_{p_k}(X) + 1 = \alpha) = \mathbb{P}(\nu_{p_k}(X) + 1 \geq \alpha) - \mathbb{P}(\nu_{p_k}(X) \geq \alpha)$

$$\begin{aligned} &= \mathbb{P}(p_k^{\alpha-1} \mid X) - \mathbb{P}(p_k^\alpha \mid X) \\ &= p_k^{-(\alpha-1)s} - p_k^{-\alpha s} \end{aligned}$$

$$\mathbb{P}(\nu_{p_k}(X) + 1 = \alpha) = (1 - p_k^{-s})(p_k^{-s})^{\alpha-1}$$

Ainsi,

$$\nu_{p_k}(X) + 1 \text{ suit une loi géométrique de paramètre } 1 - p_k^{-s}.$$

X/ENS Informatique B MP-PC-PSI 2021 — Corrigé

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

Ce sujet présente le problème de l'allocation dynamique de mémoire, qui est fondamental dans le fonctionnement des systèmes d'exploitation et de la plupart des langages de programmation. C'est un important domaine de recherche et d'ingénierie : des algorithmes utilisant des idées semblables à celles présentées ici sont utilisés en permanence par les OS modernes et par les applications. Le sujet est de difficulté croissante et à aborder dans l'ordre puisque chaque partie repose sur la précédente.

- En guise d'introduction, la partie I pose trois questions faciles pour réaliser une première implémentation naïve.
- La partie II introduit la notion d'en-têtes à chaque portion de mémoire. Ces métadonnées permettent de recycler la mémoire qui n'est plus utilisée.
- Dans la partie III, les métadonnées sont complexifiées et dupliquées afin de permettre la fusion de portions libres consécutives. L'une des questions d'implémentation de cette partie est particulièrement difficile.
- Enfin, la partie IV introduit une amélioration du système précédent qui consiste à maintenir une chaîne des portions libres permettant de parcourir rapidement ces dernières.

Le sujet ne comporte que 15 questions ; c'est peu, mais les enjeux sont autant la rapidité que la justesse. En effet, l'implémentation des fonctions **reserver** et **liberer** des deux dernières parties est complexe, et propice à de nombreuses erreurs de logique ou de décalage d'indices, surtout sur papier. Néanmoins, le sujet est très guidé et bien posé. Plusieurs pages ne comportent pas de question mais sont à lire attentivement puisqu'elles présentent les principes des algorithmes étudiés ou les fonctions à utiliser pour les implémentations. On pourra d'ailleurs noter que cette pratique d'encapsulation du code derrière des interfaces que l'on peut utiliser sans connaître leur fonctionnement interne est un pilier de la programmation orientée objet, un paradigme de programmation utilisé par exemple dans les langages Python, Java, C++, etc. C'est un très bon sujet d'entraînement, que l'on peut aborder dès la première année pour tester sa capacité à écrire du code sans faute.

INDICATIONS

Partie I

- 3 Ne pas oublier de gérer le cas où la mémoire est trop pleine pour effectuer l'allocation.

Partie II

- 5 Pour parcourir la mémoire, les blocs étant de taille fixe, il est approprié d'utiliser une boucle `for` avec un incrément adapté. Utiliser la fonction `initialiser` de la question 1. Prendre soin de bien gérer les cas particuliers (taille demandée supérieure à la taille d'un bloc ou allocation à la fin impossible par manque de place).

Partie III

- 7 On pourra imaginer la représentation binaire de l'entier dans l'en-tête et le pied de page.
- 8 Utiliser, entre autres, la fonction `marque_reservee` fournie par l'énoncé.
- 9 Commencer par un parcours des portions successives pour en chercher une libre et de taille suffisante. Le cas échéant, on effectue l'allocation, on crée éventuellement une section libre avec le reliquat et on retourne immédiatement. Si on arrive à l'épilogue sans avoir trouvé de portion adaptée, on alloue sur place en décalant l'épilogue après avoir vérifié que la place disponible est suffisante.
- 10 Si la portion suivante est libre, on peut augmenter la taille de la portion libérée. Si la portion précédente est libre, on en augmente la taille et on déplace également le pointeur qui en marque le début.

Partie IV

- 11 Commencer par le cas où la chaîne est initialement vide. Faire un schéma pour voir quelles relations (successeur ou prédécesseur) sont réécrites, et où. Faire de même pour le cas général.
- 12 Observer que pour supprimer une portion de la chaîne, il suffit de raccorder son prédécesseur et son successeur. Vérifier les cas limites lorsque la portion est en entrée ou à la fin de la chaîne.
- 13 Ne pas oublier que la variable globale `PROLOGUE` a été modifiée.
- 14 Les lignes de l'implémentation précédente à modifier sont celles qui effectuent le parcours des portions à la recherche d'une portion recyclable. Il faut également ajouter des appels à `supprime_dans_chaine` et `ajoute_en_entree_de_chaine` pour maintenir la chaîne à jour.
- 15 Ajouter trois lignes à la fonction `liberer` de la question 10 : deux appels à `supprime_dans_chaine` et un à `ajoute_en_entree_de_chaine`.

I. IMPLÉMENTATION NAÏVE

1 La fonction `initialiser` remplit avec le caractère `c` les cases de mémoire situées entre les indices `p` inclus et `p+n` exclu.

```
def initialiser(p, n, c):  
    for i in range(n):  
        mem[p+i] = c
```

La fonction `initialiser` n'est pas une des cinq faisant partie de l'interface du service. C'est donc une fonction auxiliaire, à usage interne (appelée par `reserver`). Il convient ici, comme d'habitude, de lire les questions un peu en avance afin de ne pas déjà implémenter dans `initialiser` ce qui doit l'être dans `reserver`.

2 La fonction `demarrage` doit placer la mémoire dans un état prêt à recevoir des demandes d'allocation. Initialement, la mémoire est vide et la seule case qui sera lue avant d'être écrite est `mem[0]`. On doit donc s'assurer que sa valeur est correcte pour la première utilisation de la mémoire. Cette case indiquant le début de la prochaine portion libre, sa valeur initiale doit être 1.

```
def demarrage():  
    mem[0] = 1
```

3 Dans la fonction `reserver`, on commence par s'assurer que la mémoire disponible est suffisante pour effectuer l'allocation. Si ce n'est pas le cas, on retourne directement `None`, comme demandé. Si tout va bien, on initialise la portion demandée à l'aide de la fonction écrite à la question 1 et on change la valeur de la première case, sans oublier de retourner l'indice de la portion allouée.

```
def reserver(n, c):  
    p = mem[0]  
    if p + n > TAILLE_MEM:  
        return None  
    initialiser(p, n, c)  
    mem[0] += n  
    return p
```

Pour vérifier si le test sur la mémoire disponible doit être implémenté à l'aide d'une inégalité large ou stricte, on peut compter les cases du schéma donné dans l'énoncé. Pour `TAILLE_MEM = 30` et `mem[0] = 19`, on peut allouer au maximum 11 cases, une douzième case serait de trop. Comme $19 + 11 = 30$ et $19 + 12 = 31$, on doit utiliser une inégalité stricte à la troisième ligne.

On aurait pu exploiter le fait que les fonctions Python retournent implicitement `None` lorsque l'exécution arrive à leur fin pour écrire :

```
def reserver(n, c):  
    p = mem[0]  
    if p + n <= TAILLE_MEM:  
        initialiser(p, n, c)  
        mem[0] += n  
        return p
```

C'est une bonne pratique de signaler les erreurs de manière explicite et le plus tôt possible, ce qui permet également de réduire le niveau d'indentation moyen du code et de le rendre plus compréhensible.

Pour calculer la complexité de la fonction, il faut compter le nombre d'opérations élémentaires effectuées par celle-ci. La fonction `reserver` est sans boucle ni récursion, et la seule fonction qu'elle appelle est `initialiser`, qui effectue un nombre d'opérations proportionnel à n . **Par conséquent, la complexité de la fonction est en $O(n)$ et ne dépend pas de la taille totale de la mémoire.**

II. RÉSERVATIONS DE BLOCS DE TAILLES FIXES

4 Pour l'initialisation de la mémoire, il s'agit de s'assurer, comme à la question 2, que la première case pointe au bon endroit. Il est inutile d'initialiser les en-têtes, puisque l'allocation de mémoire à partir de `mem[0]` réserve des blocs neufs, jamais encore réservés, et donc ne lit pas leurs en-têtes. Puisque `mem[0]` ne pointe pas vers l'en-tête d'un bloc, mais vers le début d'une portion de données, sa valeur initiale est ici 2 et non plus 1 comme à la question 2.

```
def demarrage():
    ecrire_prochain(2)
```

5 On peut commencer par vérifier que la taille de mémoire demandée ne dépasse pas d'un bloc. Puis, s'il existe un bloc libre à recycler, on doit l'utiliser. On en recherche donc un, en itérant sur les emplacements des portions, de la première, incluse, à celle désignée par `mem[0]`, exclue, au moyen de la fonction `range` dont le troisième argument indique l'incrément. Si la recherche échoue, après avoir vérifié que la mémoire disponible est suffisante, on effectue l'allocation d'un bloc sur la portion pointée par `mem[0]` et on met à jour la prochaine portion libre si besoin.

```
def reserver(n, c):
    if n + 1 > TAILLE_BLOC:
        return None
    # Recherche d'un bloc libre à recycler
    for p in range(2, lire_prochain(), TAILLE_BLOC):
        if est_libre(p):
            marque_reservee(p)
            initialiser(p, n, c)
            return p
    # Pas de bloc libre trouvé : allocation à la fin si possible
    p = lire_prochain()
    if p + TAILLE_BLOC - 1 > TAILLE_MEM:
        return None
    ecrire_prochain(p + TAILLE_BLOC)
    marque_reservee(p)
    initialiser(p, n, c)
    return p
```

Ici encore, ne pas hésiter à compter les cases à la main et à jouer avec les dimensions pour vérifier que les tests sont corrects.