

MP  
Mathématiques · Informatique  
2020

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

William AUFORT  
professeur en CPGE

Guillaume BATOG  
professeur en CPGE

Loïc DEVILLIERS  
professeur en CPGE

Julien DUMONT  
professeur en CPGE

Thierry LIMOGES  
professeur en CPGE

Florian METZGER  
professeur en CPGE

David MICHEL  
ENS Rennes

Benjamin MONMEGE  
enseignant-chercheur à l'université

Matthias MORENO RAY  
professeur en CPGE

Angèle NICLAS  
ENS Lyon

Rémi PELLERIN  
ENS Lyon

Vincent PUYHAUBERT  
professeur en CPGE

# Sommaire thématique de mathématiques

2015 – 2020

e3a MP Maths 1																				
e3a PC Maths 1																				
e3a PSI Maths 1		•	••	••		•	•					••						••		
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
<b>CONCOURS COMMUN INP</b>			
Mathématiques 1	Étude de séries ternaires. <i>séries numériques, suites et séries de fonctions, probabilités, intégration, informatique pour tous</i>	17	22
Mathématiques 2	Étude des similitudes d'un espace vectoriel euclidien. <i>diagonalisation, topologie, espaces euclidiens, réduction</i>	37	41
Informatique optionnelle	Connecteur de Sheffer. Problème de Freudenthal. Mots de Lyndon et de de Bruijn. <i>logique, listes Python, programmation OCaml, récursivité, mots, graphes</i>	53	63
<b>CENTRALE-SUPÉLEC</b>			
Mathématiques 1	Fonctions arithmétiques multiplicatives et applications. <i>algèbre linéaire, séries numériques, séries de fonctions, arithmétique, groupe symétrique</i>	81	86
Mathématiques 2	Espaces à noyau reproduisant. <i>espaces préhilbertiens, équations différentielles linéaires, réduction, intégration, séries entières</i>	115	119
Informatique commune	Photomosaïque. <i>programmation Python, bibliothèque NumPy, représentation des nombres, listes, bases de données</i>	143	151
Informatique optionnelle	Un système de vote. <i>programmation OCaml, graphes, logique</i>	170	174

### MINES-PONTS

Mathématiques 1	Espaces vectoriels d'endomorphismes nilpotents. <i>algèbre linéaire, algèbre bilinéaire, endomorphismes, espaces euclidiens</i>	193	199
Mathématiques 2	Nombre de sites visités par une marche aléatoire. <i>probabilités, séries numériques, séries entières, combinatoire, intégration</i>	221	227
Informatique commune	Images de vagues et de structures. <i>programmation Python, bases de données, complexité, tris, récursivité</i>	247	260
Informatique optionnelle	Énumération des lettres d'un texte et arbres de codage optimaux. <i>programmation OCaml, complexités en temps et en espace, arbres binaires de recherche, programmation dynamique</i>	271	281

### POLYTECHNIQUE-ENS

Mathématiques A	Réels qui sont valeurs propres de matrices symétriques à coefficients réels. <i>algèbre matricielle, produits scalaires, réduction, polynômes, nombres complexes, arithmétique, séries entières</i>	305	310
Mathématiques B	Comportement asymptotique en probabilités et en intégration. <i>probabilités discrètes, intégrales à paramètres, équivalents, développements limités</i>	331	336

### FORMULAIRES

Développements limités usuels en 0	361
Développements en série entière usuels	362
Dérivées usuelles	363
Primitives usuelles	364
Trigonométrie	366

SESSION 2020



MP1M1

---

**ÉPREUVE SPÉCIFIQUE - FILIÈRE MP**

---

**MATHÉMATIQUES 1****Lundi 4 mai : 8 h - 12 h**

---

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

**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.*
- 

<b>Les calculatrices sont interdites</b>
--

**Le sujet est composé d'un problème qui comprend quatre parties indépendantes.**

## Objectifs

L'objectif de la **partie I** est de montrer l'existence d'un développement ternaire propre pour certains nombres réels.

La **partie II** propose l'étude d'une série de fonctions où les coefficients du développement ternaire sont remplacés par une fonction continue.

La **partie III** étudie des développements ternaires aléatoires.

La **partie IV** définit et présente quelques propriétés de la fonction de Cantor-Lebesgue.

## Notations

On note  $T$  l'ensemble des suites réelles  $t = (t_n)_{n \in \mathbb{N}^*}$  à valeurs dans  $\{0; 1; 2\}$  :

$$\forall n \in \mathbb{N}^*, \quad t_n \in \{0; 1; 2\}.$$

On désigne par  $\ell^\infty$  l'ensemble des suites réelles  $u = (u_n)_{n \in \mathbb{N}^*}$  bornées et on pose  $\|u\| = \sup_{n \in \mathbb{N}^*} |u_n|$ .

On note  $\lfloor y \rfloor$  la partie entière d'un réel  $y$ .

## PARTIE I - Développement ternaire

### Étude de l'application $\sigma$

**Q1.** Démontrer que  $\ell^\infty$  est un espace vectoriel réel et que l'application  $u \mapsto \|u\|$  est une norme sur  $\ell^\infty$ .

**Q2.** Pour  $u = (u_n)_{n \in \mathbb{N}^*} \in \ell^\infty$ , démontrer que la série de terme général  $\frac{u_n}{3^n}$  est convergente. On note alors :

$$\sigma(u) = \sum_{n=1}^{+\infty} \frac{u_n}{3^n}.$$

**Q3.** Démontrer que l'application  $\sigma$  est une forme linéaire continue sur  $\ell^\infty$ .

**Q4.** Démontrer que si  $t = (t_n)_{n \in \mathbb{N}^*} \in T$ , alors le réel  $\sigma(t)$  est dans l'intervalle  $[0, 1]$ .

**Q5.** On note  $\tau = (\tau_n)_{n \in \mathbb{N}^*}$  et  $\tau' = (\tau'_n)_{n \in \mathbb{N}^*}$  les éléments de  $T$  définis par :

$$\tau_1 = 1 \text{ et } \forall n \in \mathbb{N}^* \setminus \{1\}, \tau_n = 0 \qquad \tau'_1 = 0 \text{ et } \forall n \in \mathbb{N}^* \setminus \{1\}, \tau'_n = 2.$$

Calculer  $\sigma(\tau)$  et  $\sigma(\tau')$ . L'application  $\sigma$  est-elle injective sur  $T$  ?

### Développement ternaire propre

On fixe  $x \in [0, 1]$ . On définit une suite  $t(x) = (t_n(x))_{n \in \mathbb{N}^*}$  par :

$$\forall n \in \mathbb{N}^*, \quad t_n(x) = \lfloor 3^n x \rfloor - 3 \lfloor 3^{n-1} x \rfloor.$$

**Q6.** Démontrer que  $t(x) \in T$ .

**Q7.** On définit deux suites réelles  $(x_n)_{n \in \mathbb{N}^*}$  et  $(y_n)_{n \in \mathbb{N}^*}$  par :

$$\forall n \in \mathbb{N}^*, \quad x_n = \frac{\lfloor 3^n x \rfloor}{3^n} \quad \text{et} \quad y_n = x_n + \frac{1}{3^n}.$$

Démontrer que les suites  $(x_n)$  et  $(y_n)$  sont adjacentes de limite  $x$ . En déduire que :

$$x = \sum_{n=1}^{+\infty} \frac{t_n(x)}{3^n}.$$

Que peut-on en conclure concernant l'application  $\begin{cases} T \rightarrow [0,1] \\ u \mapsto \sigma(u) \end{cases}$  ?

La suite  $t(x) = (t_n(x))_{n \in \mathbb{N}^*}$  est appelée *développement ternaire propre* de  $x$ .

**Q8.** *Informatique pour tous.* Écrire en langage Python une fonction `flotVersTern(n,x)` d'arguments un entier naturel  $n$  et un flottant  $x$  et qui renvoie sous forme d'une liste les  $n$  premiers chiffres  $t_1(x), \dots, t_n(x)$  définis dans la question précédente du développement ternaire de  $x$ .

Par exemple `flotVersTern(4,0.5)` renvoie `[1,1,1,1]`.

**Q9.** *Informatique pour tous.* Si  $\ell = [\ell_1, \dots, \ell_n]$  est une suite finie d'entiers de  $\{0; 1; 2\}$ , on la complète avec des 0 pour en faire un élément de  $T$  encore noté  $\ell$ .

Écrire en langage Python une fonction `ternVersFlot( $\ell$ )` d'arguments une liste d'entiers  $\ell$ . Cette fonction renvoie en sortie le flottant  $\sigma(\ell)$ .

Par exemple `ternVersFlot([1,1,1,1])` renvoie `0.493827.....`

**Q10.** *Informatique pour tous.* Si  $\ell = [\ell_1, \dots, \ell_n]$  est une suite finie d'entiers de  $\{0; 1; 2\}$ , on lui ajoute un élément égal à  $-1$  si la somme  $\ell_1 + \dots + \ell_n$  est paire et un élément égal à  $-2$  sinon. Ce dernier élément permet alors d'essayer de détecter d'éventuelles erreurs de transmission.

Écrire en langage Python une fonction `ajout( $\ell$ )` qui ajoute à la liste  $\ell$  un élément comme expliqué précédemment et qui renvoie la nouvelle liste.

Écrire en langage Python une fonction `verif( $\ell$ )` qui renvoie `True` si la valeur du dernier élément de  $\ell$  est correcte et `False` sinon.

Par exemple `ajout([1,0,2,1,0])` renvoie `[1,0,2,1,0,-1]` et `verif([1,0,2,1,0,-2])` renvoie `False`.

## PARTIE II - Étude d'une fonction définie par une série

Dans cette partie, on définit une fonction  $\varphi$  à l'aide d'un développement en série analogue au développement ternaire propre d'un réel, mais où la suite  $(t_n)_{n \in \mathbb{N}^*}$  est remplacée par une fonction numérique à valeurs dans l'intervalle  $[0,2]$ .

Pour tout réel  $x$  on pose :

$$\varphi(x) = \sum_{n=1}^{+\infty} \frac{1 + \sin(nx)}{3^n}.$$

## CCINP Maths 1 MP 2020 — Corrigé

Ce corrigé est proposé par David Michel (ENS Rennes) ; il a été relu par Quentin Guilmant (ENS Lyon) et Benjamin Monmege (enseignant-chercheur à l'université).

---

Ce sujet traite de séries dont le terme général est de la forme  $u_n/3^n$  où  $(u_n)$  est une suite numérique, une suite de fonctions ou une suite de variables aléatoires. Il est composé de quatre parties indépendantes.

- Dans la première, on s'intéresse à l'écriture *ternaire* des réels compris entre 0 et 1. On en démontre l'existence ainsi que la non-unicité en général, et on met en œuvre informatiquement le calcul du développement ternaire *propre* d'un flottant entre 0 et 1.
- La partie suivante étudie une fonction définie comme la somme d'une série. Après avoir démontré sa régularité, on calcule son expression et son intégrale sur le segment  $[0; \pi]$ . On en déduit la somme d'une série et la valeur d'une intégrale.
- Dans la troisième partie, on étudie le comportement asymptotique d'une variable aléatoire à valeurs dans l'intervalle  $[0; 1]$  dont l'écriture ternaire est finie.
- La dernière partie demande de construire par récurrence une suite de fonctions. On démontre sa convergence uniforme et certaines propriétés de sa limite, appelée *fonction de Cantor-Lebesgue* ou *escalier du diable*.

Pour réussir ce sujet, il fallait maîtriser les suites et séries de fonctions ainsi que les définitions et inégalités essentielles du programme de probabilités. Des notions élémentaires d'informatique sont également nécessaires. On est amené à utiliser le critère de comparaison des séries à termes positifs à plusieurs reprises, ainsi que les théorèmes de dérivation et d'intégration des séries de fonctions ou encore le théorème de continuité des suites de fonctions. Si le lien entre les parties I, II et III tient à la forme des suites et séries considérées, le lien avec la partie IV est mentionné dans le sujet mais n'est pas traité explicitement. Nous donnons quelques éléments d'explication dans ce corrigé.



## INDICATIONS

- 2 Démontrer la convergence absolue par comparaison à une série géométrique.
- 3 Démontrer la linéarité en raisonnant sur les sommes partielles et en passant à la limite. Se ramener à une série géométrique pour démontrer la continuité.
- 4 Comparer à une série géométrique dont on calculera la somme.
- 6 Montrer que  $t_n(x) \in \mathbb{Z}$  puis que  $-1 < t_n(x) < 3$  en utilisant le fait que

$$\forall x \in \mathbb{R} \quad x - 1 < \lfloor x \rfloor \leq x$$

- 7 Montrer que  $(x_n)_{n \in \mathbb{N}^*}$  est croissante, que  $(y_n)_{n \in \mathbb{N}^*}$  est décroissante et que, pour tout  $n \in \mathbb{N}^*$ ,  $x_n \leq x < y_n$ . Écrire  $x$  comme la somme de la série télescopique de terme général  $x_n - x_{n-1}$ .
- 8 Construire la liste  $t$  à partir de la liste vide en y ajoutant les éléments l'un après l'autre. On pourra utiliser la fonction `floor`.
- 9 Utiliser une boucle `for` pour calculer la somme définissant  $\sigma(\ell)$ .
- 10 Veiller à ne pas prendre en compte le dernier élément de la liste pour le calcul de la somme  $\ell_1 + \dots + \ell_n$  dans la fonction `verif`.
- 11 Appliquer le théorème de dérivation des séries de fonctions. On pourra utiliser des comparaisons à des séries géométriques.
- 12 Justifier que l'on peut séparer la série en deux et utiliser la linéarité de la partie imaginaire. Calculer une expression de  $\varphi(x)$  grâce à la somme d'une série géométrique.
- 13 Utiliser les questions 11 et 12.
- 14 Exprimer d'une part l'intégrale de  $\varphi$  à l'aide de l'intégrale à calculer grâce à la question 12. Utiliser d'autre part le théorème d'intégration des séries de fonctions sur un segment pour calculer la valeur de l'intégrale de  $\varphi$ . En déduire l'égalité de l'énoncé. Enfin, calculer la somme à l'aide du développement en série entière :

$$\forall x \in ]-1; 1[ \quad \ln(1-x) = -\sum_{n=1}^{+\infty} \frac{x^n}{n}$$

- 15 Effectuer le changement de variable  $u = \cos(x)$  puis se ramener à une intégrale de  $v \mapsto (1-v)^{-1}$  par un changement de variable linéaire.
- 16 Justifier que les variables aléatoires  $T_{n,N}$  admettent une espérance et une variance et en déduire que c'est aussi le cas de  $X_N$ .
- 17 Utiliser l'inégalité de Bienaymé-Tchebychev.
- 18 Écrire  $|X_N - 1| \leq |X_N - \mathbb{E}(X_N)| + |\mathbb{E}(X_N) - 1|$   
et montrer qu'il existe  $N_0 \in \mathbb{N}$  tel que pour tout  $N \geq N_0$ ,  $|\mathbb{E}(X_N) - 1| < \varepsilon/2$ .
- 19 Déterminer à la main les valeurs de  $f_1(x)$  et  $f_2(x)$  pour tout  $x \in [0; 1]$  en distinguant les cas. Démontrer par récurrence que  $f_n$  est bien à valeurs dans  $[0; 1]$ .
- 21 Démontrer le résultat par récurrence en distinguant selon la position de  $x$  dans l'intervalle  $[0; 1]$ .
- 22 Utiliser la question 21. On rappelle qu'une suite  $(u_n)_{n \in \mathbb{N}}$  converge si et seulement si la série  $\sum (u_{n+1} - u_n)$  converge.
- 23 Montrer par récurrence que pour tout  $n \in \mathbb{N}$ ,  $f_n$  est continue et croissante, ainsi que les égalités  $f_n(0) = 0$  et  $f_n(1) = 1$  puis passer à la limite. Utiliser le théorème des valeurs intermédiaires pour démontrer la surjectivité de  $f$ .

## I. DÉVELOPPEMENT TERNAIRE

**1** Montrons que  $\ell^\infty$  est un sous-espace vectoriel de l'espace des suites réelles.

- L'ensemble  $\ell^\infty$  est bien inclus dans cet espace.
- La suite nulle, neutre pour l'addition des suites, est bornée, par exemple par 0.
- Soient  $(u, v) \in (\ell^\infty)^2$  et  $\lambda \in \mathbb{R}$ . D'après l'inégalité triangulaire,

$$\forall n \in \mathbb{N}^* \quad |(u + \lambda v)_n| = |u_n + \lambda v_n| \leq |u_n| + |\lambda| |v_n| \leq \|u\| + |\lambda| \|v\|$$

donc  $u + \lambda v \in \ell^\infty$ .

Ainsi,

L'ensemble  $\ell^\infty$  est un  $\mathbb{R}$ -espace vectoriel.

Vérifions à présent que  $\|\cdot\|$  est une norme sur  $\ell^\infty$ .

- L'application  $\|\cdot\|$  est à valeurs dans  $\mathbb{R}_+$  par définition.
- Séparation : si  $u \in \ell^\infty$  et  $\|u\| = 0$ , alors pour tout  $n \in \mathbb{N}^*$ ,  $|u_n| \leq 0$ , d'où  $u = 0$ .
- Homogénéité : si  $u \in \ell^\infty$  et  $\lambda \in \mathbb{R}$ , alors pour tout  $n \in \mathbb{N}^*$ ,  $|\lambda u_n| \leq |\lambda| \|u\|$  donc, en passant à la borne supérieure,  $\|\lambda u\| = |\lambda| \|u\|$ .
- Inégalité triangulaire : comme on l'a vu en démontrant la stabilité de  $\ell^\infty$  par combinaison linéaire, si  $(u, v) \in (\ell^\infty)^2$ , alors

$$\forall n \in \mathbb{N}^* \quad |(u + v)_n| \leq \|u\| + \|v\|$$

d'où, en passant à la borne supérieure,  $\|u + v\| \leq \|u\| + \|v\|$ .

Ainsi,

L'application  $\|\cdot\|$  est une norme sur  $\ell^\infty$ .

**2** Soit  $u \in \ell^\infty$ . Pour tout  $n \in \mathbb{N}^*$ , on a :

$$\left| \frac{u_n}{3^n} \right| \leq \frac{\|u\|}{3^n}$$

Comme  $-1 < 1/3 < 1$ , le membre de droite de cette inégalité est le terme général d'une série géométrique convergente. Par comparaison de séries à termes positifs, on en déduit que la série de terme général  $u_n/3^n$  converge absolument. Par conséquent,

La série de terme général  $u_n/3^n$  est convergente.

**3** D'après la question 2,  $\sigma$  est bien définie sur  $\ell^\infty$  et à valeurs dans  $\mathbb{R}$ . Démontrons sa linéarité. Soient  $(u, v) \in (\ell^\infty)^2$  et  $\lambda \in \mathbb{R}$ . D'après la question 2, les séries de termes généraux  $(u_n + \lambda v_n)/3^n$ ,  $u_n/3^n$  et  $v_n/3^n$  sont convergentes donc on peut passer à la limite dans l'égalité

$$\forall N \in \mathbb{N}^* \quad \sum_{n=1}^N \frac{u_n + \lambda v_n}{3^n} = \sum_{n=1}^N \frac{u_n}{3^n} + \lambda \sum_{n=1}^N \frac{v_n}{3^n}$$

et on obtient

$$\sigma(u + \lambda v) = \sigma(u) + \lambda \sigma(v)$$

Enfin, pour  $u \in \ell^\infty$ , en passant à la limite dans l'inégalité

$$\forall N \in \mathbb{N}^* \quad \left| \sum_{n=1}^N \frac{u_n}{3^n} \right| \leq \|u\| \sum_{n=1}^N \frac{1}{3^n} \leq \|u\| \sum_{n=1}^{+\infty} \frac{1}{3^n} = \|u\| \times \frac{1}{3} \times \frac{1}{1 - 1/3} = \frac{\|u\|}{2}$$

on trouve

$$|\sigma(u)| = \left| \sum_{n=1}^{+\infty} \frac{u_n}{3^n} \right| \leq \frac{\|u\|}{2}$$

Ainsi,

L'application  $\sigma$  est une forme linéaire continue sur  $\ell^\infty$ .

**4** Soit  $t \in \mathbb{T}$ . Par définition, on a

$$\forall n \in \mathbb{N}^* \quad 0 \leq \frac{t_n}{3^n} \leq \frac{2}{3^n}$$

d'où 
$$0 \leq \sigma(t) \leq 2 \sum_{n=1}^{+\infty} \frac{1}{3^n} = 2 \times \frac{1}{3} \times \frac{1}{1 - 1/3} = 1$$

On a donc montré que  $\boxed{\forall t \in \mathbb{T} \quad \sigma(t) \in [0; 1]}$

**5** Seul le premier terme de la série définissant  $\sigma(\tau)$  est non nul et on calcule immédiatement  $\sigma(\tau) = 1/3$ . Par ailleurs,

$$\sigma(\tau') = 2 \sum_{n=2}^{+\infty} \frac{1}{3^n} = \frac{2}{3^2} \times \frac{1}{1 - 1/3} = \frac{1}{3}$$

Ainsi,  $\boxed{\sigma(\tau) = \sigma(\tau') = 1/3}$ . En particulier,  $\sigma$  n'est pas injective sur  $\mathbb{T}$ .

**6** Soit  $n \in \mathbb{N}^*$ . Par définition de la partie entière, on a  $[3^n x] \in \mathbb{N}$  et  $[3^{n-1} x] \in \mathbb{N}$  donc  $t_n(x) \in \mathbb{Z}$ . De plus, d'une part

$$3^n x - 1 < [3^n x] \leq 3^n x$$

et d'autre part

$$3^{n-1} x - 1 < [3^{n-1} x] \leq 3^{n-1} x$$

c'est-à-dire

$$-3^n x \leq -3 [3^{n-1} x] < -3^{n-1} x + 3$$

En additionnant la première et la troisième inégalités, on obtient

$$-1 < t_n(x) < 3$$

Comme  $t_n(x) \in \mathbb{Z}$ , on en déduit que  $t_n(x) \in \{0; 1; 2\}$ . En conclusion,

$$\boxed{t(x) \in \mathbb{T}}$$

**7** Soit  $n \in \mathbb{N}^*$ . D'après la question 6,

$$x_{n+1} - x_n = \frac{[3^{n+1} x]}{3^{n+1}} - \frac{[3^n x]}{3^n} = \frac{t_{n+1}(x)}{3^{n+1}} \geq 0$$

et 
$$y_{n+1} - y_n = x_{n+1} - x_n + \frac{1}{3^{n+1}} - \frac{1}{3^n} = \frac{t_{n+1}(x) - 2}{3^{n+1}} \leq 0$$

Ainsi,  $(x_n)_{n \in \mathbb{N}^*}$  est croissante et  $(y_n)_{n \in \mathbb{N}^*}$  est décroissante. De plus,

$$y_n - x_n = \frac{1}{3^n} \xrightarrow{n \rightarrow \infty} 0$$

donc

$$\boxed{\text{Les suites } (x_n)_{n \in \mathbb{N}^*} \text{ et } (y_n)_{n \in \mathbb{N}^*} \text{ sont adjacentes.}}$$

D'après le théorème des suites adjacentes, elles convergent vers un réel  $y$ . Or, par définition de la partie entière, pour tout  $n \in \mathbb{N}^*$ ,

$$[3^n x] \leq 3^n x < [3^n x] + 1$$

d'où

$$x_n \leq x < y_n$$

En passant à la limite, on obtient  $\boxed{y = x}$

Enfin, on a vu plus haut que pour tout  $n \geq 2$ ,

$$x_n - x_{n-1} = \frac{t_n(x)}{3^n}$$

## CCINP Maths 2 MP 2020 — Corrigé

Ce corrigé est proposé par Angèle Niclas (ENS Lyon) ; il a été relu par David Michel (ENS Rennes) et Benjamin Monmege (enseignant-chercheur à l'université).

---

Ce sujet est composé de deux exercices et d'un grand problème. Chacun porte sur un thème différent du programme.

- L'exercice 1, très calculatoire, est assez proche du cours d'algèbre linéaire. Le sujet demande de diagonaliser une matrice de taille 3, puis de calculer son polynôme minimal et ses puissances.
- L'exercice 2 est une application du cours de topologie. On prouve que  $GL_n(\mathbb{R})$  est un ouvert dense de  $\mathcal{M}_n(\mathbb{R})$  et on propose une application de ce résultat.
- Le problème porte sur l'étude des similitudes d'un espace vectoriel euclidien. Il propose de démontrer plusieurs de leurs caractérisations en lien, entre autre, avec les automorphismes orthogonaux. Il est plus long que les exercices et exige de bien maîtriser le cours sur les espaces euclidiens. On commence par étudier des exemples concrets avant de prouver des résultats de plus en plus théoriques. Les questions sont de difficulté croissante et s'enchaînent naturellement.

Cet énoncé est classique et proche du cours. Le premier exercice constitue une bonne révision des techniques de diagonalisation. Le deuxième permet de vérifier que les notions de topologie sont comprises et maîtrisées. Enfin, le problème regroupe plusieurs questions classiques sur les espaces euclidiens qu'il est bon de savoir traiter.

## INDICATIONS

### Exercice 1

- 1 Utiliser le théorème spectral sur  $A$ .
- 2 Remarquer que  $D$  s'écrit facilement comme le carré d'une matrice.
- 3 Prouver que  $A^n = PD^nP^{-1}$  pour tout  $n \in \mathbb{N}^*$  en utilisant la question 1.
- 4 Utiliser la question 1 et le fait que  $A$  est diagonalisable pour trouver le polynôme minimal de  $A$ , puis effectuer la division euclidienne de  $X^n$  par ce polynôme et évaluer cette formule en 1 et 4.

### Exercice 2

- 5 Étudier la suite de matrices  $M_k = (1/k)I_n$ .
- 6 Utiliser le fait que l'image réciproque d'un ouvert par une application continue est un ouvert.
- 7 Remarquer que  $\det(XI_n - M)$  a au plus  $n$  racines réelles positives, puis considérer une suite de matrices de terme général  $M_k = M - (1/k)I_n$  pour  $k$  bien choisi.
- 8 Commencer par traiter le cas où  $B$  est inversible, puis utiliser la question 7.
- 9 Utiliser le fait que les connexes par arcs de  $\mathbb{R}$  sont les intervalles.

### Problème

- 12 Prouver que tout élément de  $\text{Sim}(E)$  est injectif et utiliser le fait que  $E$  est de dimension finie.
- 13 Utiliser l'identité de polarisation pour prouver que tout automorphisme orthogonal préserve le produit scalaire. Pour trouver la relation caractéristique d'une similitude  $u$  de rapport  $k$ , remarquer que  $u/k$  est un automorphisme orthogonal.
- 14 Utiliser les caractérisations de la question 13.
- 15 Poser  $k$  le rayon de l'image de la sphère de rayon 1 par  $u$  et pour tout  $x \in E$  non nul, remarquer que  $x/\|x\|$  est dans la sphère de rayon 1.
- 16 Remarquer que  $u = k \times u/k$  si  $k$  est le rapport de similitude de  $u$ .
- 17 Utiliser les questions 10 et 16.
- 19 Pour prouver que  $u$  est une similitude, développer  $\|u(x)\|^2$  sur la base  $(e_1, \dots, e_n)$  et remarquer que  $\langle u(e_i) \mid u(e_j) \rangle = 0$  si  $i \neq j$ .
- 20 Montrer que  $\|u(x + \lambda y) - u(x) - \lambda u(y)\|^2 = 0$  pour tous  $x, y \in E$  et  $\lambda \in \mathbb{R}$  en développant et en utilisant l'égalité vérifiée par  $u$ .

## EXERCICE I

**1** La matrice  $A$  est une matrice réelle, et symétrique car  ${}^tA = A$ . Le théorème spectral permet donc d'affirmer qu'elle est diagonalisable dans une base orthonormale et que ses valeurs propres sont toutes réelles.

La matrice  $A$  est diagonalisable.

Calculons ensuite le polynôme caractéristique de  $A$ , noté  $\chi_A$ , en développant le déterminant par rapport à la première colonne.

$$\begin{aligned}\chi_A &= \begin{vmatrix} X-2 & -1 & -1 \\ -1 & X-2 & -1 \\ -1 & -1 & X-2 \end{vmatrix} \\ &= (X-2) \begin{vmatrix} X-2 & -1 \\ -1 & X-2 \end{vmatrix} + \begin{vmatrix} -1 & -1 \\ -1 & X-2 \end{vmatrix} - \begin{vmatrix} -1 & -1 \\ X-2 & -1 \end{vmatrix} \\ &= (X-2)((X-2)^2 - 1) + (2 - X - 1) - (1 + X - 2) \\ \chi_A &= X^3 - 6X^2 + 9X - 4\end{aligned}$$

En remarquant que 1 est une racine de  $\chi_A$  puis en factorisant par  $X - 1$  on obtient que  $\chi_A = (X - 1)(X^2 - 5X + 4)$ . À nouveau, 1 est racine de  $X^2 - 5X + 4$  d'où  $\chi_A = (X - 1)^2(X - 4)$ . Ainsi, les racines de  $\chi_A$  sont 1 et 4. Notons ensuite que

$$A - 4I_3 = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix} \quad A - I_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Soit  $(x, y, z) \in \text{Ker}(A - 4I_3)$ . On a alors

$$\begin{cases} -2x + y + z = 0 \\ x - 2y + z = 0 \\ x + y - 2z = 0 \end{cases} \iff \begin{cases} x + y - 2z = 0 \\ -3y + 3z = 0 \\ 3y - 3z = 0 \end{cases} \iff \begin{cases} y = z \\ x = z \end{cases}$$

donc  $v_1 = (1, 1, 1)$  est un vecteur propre de  $A$  associé à la valeur propre 4. Soit maintenant  $(x, y, z) \in \text{Ker}(A - I_3)$ , alors

$$\begin{cases} x + y + z = 0 \\ x + y + z = 0 \\ x + y + z = 0 \end{cases} \iff x + y + z = 0$$

Ainsi  $v_2 = (1, -1, 0)$  et  $v_3 = (0, 1, -1)$  sont deux vecteurs propres non colinéaires associés à la valeur propre 1. La famille  $(v_1, v_2, v_3)$  est une base de  $\mathbb{R}^3$  et si on note  $P$  la matrice de passage de la base canonique à cette nouvelle base, la formule de changement de base donne

$$A = PDP^{-1} \text{ avec } D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \in \text{GL}_3(\mathbb{R})$$

Il est possible de choisir d'autres vecteurs  $(v_1, v_2, v_3)$  ce qui change alors la matrice de passage  $P$ .

**2** En utilisant la question 1, écrivons  $A = PDP^{-1}$ . Définissons ensuite la matrice

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

Remarquons que  $\Delta^2 = D$ , et posons  $B = P\Delta P^{-1}$ . On a alors

$$B^2 = (P\Delta P^{-1})^2 = P\Delta P^{-1}P\Delta P^{-1} = P\Delta^2 P^{-1} = PDP^{-1} = A$$

Pour expliciter  $B$ , calculons  $P^{-1}$ . Soient  $(x, y, z)$  et  $(a, b, c)$  dans  $\mathbb{R}^3$ . Il vient

$$P \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \iff \begin{cases} x + y = a \\ x - y + z = b \\ x - z = c \end{cases} \iff \begin{cases} x + y = a \\ -2y + z = b - a \\ -y - z = c - a \end{cases}$$

$$\iff \begin{cases} x + y = a \\ y + z = a - c \\ 3z = b + a - 2c \end{cases} \iff \begin{cases} z = \frac{1}{3}a + \frac{1}{3}b - \frac{2}{3}c \\ y = \frac{2}{3}a - \frac{1}{3}b - \frac{1}{3}c \\ x = \frac{1}{3}a + \frac{1}{3}b + \frac{1}{3}c \end{cases}$$

Ainsi,

$$P^{-1} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & -1 \\ 1 & 1 & -2 \end{pmatrix}$$

Finalement, calculons l'expression explicite de  $B = P\Delta P^{-1}$  :

$$B = \frac{1}{3} \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & -1 \\ 1 & 1 & -2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}$$

En conclusion,

La matrice  $B = \frac{1}{3} \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}$  vérifie  $B^2 = A$ .

**3** Montrons que la propriété

$$\mathcal{P}(n) : A^n = PD^n P^{-1}$$

est vraie pour tout  $n \geq 1$ .

- $\mathcal{P}(1)$  est vraie en utilisant le résultat de la question 1.
- $\mathcal{P}(n) \implies \mathcal{P}(n+1)$  : soit  $n \geq 1$  tel que  $\mathcal{P}(n)$  est vraie. D'après  $\mathcal{P}(n)$ , il vient que  $A^n = PD^n P^{-1}$ . Comme on a également  $A = PDP^{-1}$ , alors

$$A^{n+1} = A^n \times A = PD^n P^{-1} PDP^{-1} = PD^n DP^{-1} = PD^{n+1} P^{-1}$$

- Conclusion :  $\forall n \geq 1 \quad A^n = PD^n P^{-1}$

En utilisant l'expression de  $P^{-1}$  de la question 2, on calcule  $A^n$  qui vaut

$$\frac{1}{3} \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 4^n & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & -1 \\ 1 & 1 & -2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 4^n + 2 & 4^n - 1 & 4^n - 1 \\ 4^n - 1 & 4^n + 2 & 4^n - 1 \\ 4^n - 1 & 4^n - 1 & 4^n + 2 \end{pmatrix}$$

En conclusion,

$\forall n \in \mathbb{N}^* \quad A^n = \frac{1}{3} \begin{pmatrix} 4^n + 2 & 4^n - 1 & 4^n - 1 \\ 4^n - 1 & 4^n + 2 & 4^n - 1 \\ 4^n - 1 & 4^n - 1 & 4^n + 2 \end{pmatrix}$

## CCINP Informatique optionnelle MP 2020 — Corrigé

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

---

Le sujet comporte trois parties indépendantes.

- La partie I étudie le connecteur de Sheffer défini par  $x_1 \diamond x_2 = \neg x_1 \vee \neg x_2$ . On montre que toute formule propositionnelle est équivalente à une formule n'utilisant que le connecteur de Sheffer. Cette partie est courte et composée d'applications directes du cours sur les tables de vérité, les équivalences logiques et les démonstrations par induction.
- La partie II implémente en Python une solution d'un problème soumis en 1969 par Hans Freudenthal, didacticien en mathématiques. Un professeur choisit deux entiers  $1 < x < y$  avec  $x + y \leq n$  et confie le produit  $xy$  à un étudiant et la somme  $x + y$  à un autre. À partir d'une conversation en apparence sans contenu mathématique, les deux étudiants peuvent trouver  $x$  et  $y$  sans connaître l'information de l'autre. La résolution théorique est admise et difficile à comprendre en temps limité. Il est conseillé de coder les fonctions décrites sans comprendre le lien avec le problème initial. La plupart des questions encouragent à implémenter des opérations ensemblistes sur des listes d'entiers à coups de nombreux `for x in liste` ou de listes en compréhension.
- La partie III construit des séquences de de Bruijn. Ce sont des mots de longueur minimale qui contiennent tous les mots de taille  $n$  sur un alphabet à  $k$  lettres. Les algorithmes de génération sont donnés : soit on les déroule sur un exemple, soit on les implémente en OCaml à l'aide de chaînes de caractères. Une des méthodes utilise les mots de Lyndon étudiés en début de partie, une autre utilise les graphes. On termine avec une application amusante sur l'ouverture d'une porte munie d'un digicode. Les questions sont nombreuses, très souvent basiques, et l'on passe facilement d'une sous-partie à l'autre. La partie ne comporte que 4 questions de programmation sur 27 au total.

Cette épreuve est du même format que les années passées et aborde une bonne partie du programme d'informatique optionnelle : programmation, algorithmique sur les mots, graphes et logique. Elle contient de nombreuses définitions mathématisées auxquelles il faut s'habituer. Pour la réussir, il vaut mieux ne pas s'arrêter trop longtemps sur les difficultés théoriques afin de traiter un maximum de questions faciles.



## INDICATIONS

### Partie I

- 5 Transformer les écritures des connecteurs à l'aide des lois de De Morgan et de la double négation pour reconnaître le connecteur de Scheffer.
- 6 Une forme normale conjonctive peut être réduite à un seul maxterme.
- 7 Utiliser les lignes de la table de vérité de la question 2.
- 8 Un raisonnement par induction porte ici sur la structure de la formule propositionnelle. Ne pas oublier les cas de base.

### Partie II

- 11 Utiliser deux listes, l'une stockant les produits rencontrés pour la première fois et l'autre stockant ceux rencontrés pour la deuxième fois.
- 12 Tout couple  $(x, y) \in \mathcal{N}_n$  tel que  $x + y = S$  est de la forme  $(x, S - x)$ .
- 13 Construire l'ensemble  $\{S \leq n \mid \text{Prod}(S, n) \subset \text{CoupleProd}(n)\}$ . Coder une fonction testant l'inclusion d'une liste dans une autre.
- 14 Construire l'ensemble suivant :

$$\bigcup_{\substack{S_1 < S_2 \\ \text{dans } \text{Candidat\_S}(n)}} [\text{Prod}(S_1, n) \cap \text{Prod}(S_2, n)]$$

Coder 2 fonctions calculant l'intersection et l'union de 2 listes.

- 15 Construire l'ensemble suivant :

$$\{S \in \text{Candidat\_S}(n) \mid \text{Card}(\text{Prod}(S, n) \cap \text{Double\_P}(n)) = 1\}$$

- 16 Construire la différence ensembliste  $\text{Prod}(S, n) \setminus \text{Double\_P}(n)$ .
- 17 Les entiers  $x$  et  $y$  sont les racines du polynôme  $X^2 - \Sigma X + \Pi$ .

### Partie III

- 18 Utiliser la fonction `String.sub` pour récupérer un mot privé de sa première lettre.
- 20 Réflexivité, symétrie et transitivité sont les trois propriétés à vérifier.
- 22 Tester si le mot est strictement inférieur à tous ses conjugués sauf celui pour  $i = 0$ .
- 27 Écrire une fonction récursive auxiliaire qui simule la boucle « tant que » avec 4 arguments correspondants à  $j$ ,  $k$ ,  $\mathcal{L}$  et  $m$ .
- 31 Dénombrer par paquets d'arcs issus d'un même sommet.
- 34 Chercher un mot de longueur  $n + 1$  que l'on construit facilement à partir de l'arc  $(am, mb)$  du graphe  $\mathcal{B}(k, n)$ .
- 35 Deux arcs adjacents dans le graphe  $\mathcal{B}(k, n)$  fournissent deux sommets dans le graphe  $\mathcal{B}(k, n + 1)$  avec la construction de la question 34. Il reste à trouver l'étiquette de l'arc reliant ces deux sommets.
- 37 Montrer que le graphe  $\mathcal{B}(k, n)$  vérifie les conditions nécessaires et suffisantes du résultat (i) de l'énoncé.
- 39 Pour un mot donné, construire sa classe d'équivalence puis itérer le procédé avec un mot qui n'est pas encore apparu.
- 42 Pour la borne inférieure, donner une inégalité reliant la longueur d'un mot et le nombre de codes distincts de  $n$  chiffres qu'il contient.
- 43 Préférer la technique avec les mots de Lyndon pour construire un mot de de Bruijn pour  $n = 2$  et  $k = 4$ .

## I. LOGIQUE ET CALCUL DES PROPOSITIONS

**1** À permutation des littéraux près, les mintermes sur  $\{x_1, x_2\}$  sont exactement

$$\boxed{x_1 \wedge x_2 \quad \neg x_1 \wedge x_2 \quad x_1 \wedge \neg x_2 \quad \neg x_1 \wedge \neg x_2}$$

et les maxtermes sur  $\{x_1, x_2\}$  sont exactement

$$\boxed{x_1 \vee x_2 \quad \neg x_1 \vee x_2 \quad x_1 \vee \neg x_2 \quad \neg x_1 \vee \neg x_2}$$

**2** La table de vérité du connecteur de Sheffer est

$x_1$	$x_2$	$x_1 \diamond x_2$
0	0	1
0	1	1
1	0	1
1	1	0

**3** D'après les lois de De Morgan,

$$\boxed{x_1 \diamond x_2 = \neg(x_1 \wedge x_2)}$$

L'énoncé ne distingue pas l'équivalence sémantique  $\equiv$  de l'égalité syntaxique  $=$  de deux formules propositionnelles.

**4** D'après la définition du connecteur de Scheffer,

$$\boxed{x_1 \diamond x_1 = \neg x_1 \vee \neg x_1 = \neg x_1}$$

**5** Pour toutes formules propositionnelles  $\mathcal{F}_1$  et  $\mathcal{F}_2$ ,

$$\mathcal{F}_1 \wedge \mathcal{F}_2 = \neg \neg (\mathcal{F}_1 \wedge \mathcal{F}_2) \quad \text{(double négation)}$$

$$= \neg (\mathcal{F}_1 \diamond \mathcal{F}_2) \quad \text{(question 3)}$$

$$\boxed{\mathcal{F}_1 \wedge \mathcal{F}_2 = (\mathcal{F}_1 \diamond \mathcal{F}_2) \diamond (\mathcal{F}_1 \diamond \mathcal{F}_2)} \quad \text{(question 4)}$$

$$\mathcal{F}_1 \vee \mathcal{F}_2 = \neg \neg (\mathcal{F}_1 \vee \mathcal{F}_2) \quad \text{(double négation)}$$

$$= \neg (\neg \mathcal{F}_1 \wedge \neg \mathcal{F}_2) \quad \text{(loi de De Morgan)}$$

$$= \neg \mathcal{F}_1 \diamond \neg \mathcal{F}_2 \quad \text{(question 3)}$$

$$\boxed{\mathcal{F}_1 \vee \mathcal{F}_2 = (\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)} \quad \text{(question 4)}$$

$$\mathcal{F}_1 \Rightarrow \mathcal{F}_2 = \neg \mathcal{F}_1 \vee \mathcal{F}_2 \quad \text{(définition de } \Rightarrow \text{)}$$

$$= (\neg \mathcal{F}_1 \diamond \neg \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2) \quad \text{(encadré précédent)}$$

$$= \neg \neg \mathcal{F}_1 \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2) \quad \text{(question 4)}$$

$$\boxed{\mathcal{F}_1 \Rightarrow \mathcal{F}_2 = \mathcal{F}_1 \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)} \quad \text{(double négation)}$$

Il est possible d'obtenir d'autres résultats beaucoup plus longs lorsque l'on s'y prend mal. Par exemple, dans l'expression de  $\vee$  où l'on utilise exclusivement la question 4 et le résultat pour  $\wedge$ , on obtient de façon tout aussi juste

$$\begin{aligned} \mathcal{F}_1 \vee \mathcal{F}_2 &= \neg(\neg \mathcal{F}_1 \wedge \neg \mathcal{F}_2) = \neg((\mathcal{F}_1 \diamond \mathcal{F}_1) \wedge (\mathcal{F}_2 \diamond \mathcal{F}_2)) \\ &= \neg((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \diamond ((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \\ \mathcal{F}_1 \vee \mathcal{F}_2 &= \left[ ((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \diamond ((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \right] \\ &\quad \diamond \left[ ((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \diamond ((\mathcal{F}_1 \diamond \mathcal{F}_1) \diamond (\mathcal{F}_2 \diamond \mathcal{F}_2)) \right] \end{aligned}$$

**6** L'expression de  $x_1 \diamond x_2$  est réduite à un seul maxterme donc

La formule  $\neg x_1 \vee \neg x_2$  est une forme normale conjonctive de  $x_1 \diamond x_2$ .

**7** Formons les mintermes correspondant aux lignes de la table de vérité de la question 2 qui évaluent  $x_1 \diamond x_2$  à 1, de sorte que

La formule  $(\neg x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)$  est une forme normale disjonctive de  $x_1 \diamond x_2$ .

**8** Considérons le prédicat  $\mathcal{P}$ : « la formule  $\mathcal{F}$  est équivalente à une formule où seul le connecteur  $\diamond$  est utilisé. » Rappelons que l'ensemble  $\{\neg, \vee, \wedge\}$  est un système complet de connecteurs. Soit  $\mathcal{F}$  une formule propositionnelle.

- Si  $\mathcal{F}$  est réduite à  $\top$ ,  $\perp$  ou une variable, elle n'utilise aucun connecteur donc elle vérifie  $\mathcal{P}$ .
- Si  $\mathcal{F} = \neg \mathcal{F}_1$  avec  $\mathcal{F}_1$  vérifiant  $\mathcal{P}$ , alors elle vérifie  $\mathcal{P}$  aussi car  $\mathcal{F} = \mathcal{F}_1 \diamond \mathcal{F}_1$  d'après la question 4, valable également pour les formules propositionnelles.
- Si  $\mathcal{F} = \mathcal{F}_1 \vee \mathcal{F}_2$  ou  $\mathcal{F} = \mathcal{F}_1 \wedge \mathcal{F}_2$  avec  $\mathcal{F}_1$  et  $\mathcal{F}_2$  vérifiant  $\mathcal{P}$ , on conclut que  $\mathcal{F}$  vérifie  $\mathcal{P}$  d'après la question 5.

Par induction structurelle, le prédicat  $\mathcal{P}$  est vérifié pour toute formule. Ainsi,

L'ensemble  $\{\diamond\}$  est un système complet de connecteurs.

Il est possible de procéder par récurrence sur le nombre de connecteurs logiques composant la formule. Attention, une récurrence portant sur le nombre de variables propositionnelles n'aboutit pas, notamment lorsque les ensembles de variables sont identiques de part et d'autre d'un connecteur.

**9** Les négations sont transformées en fin de calcul :

$$x_1 \vee (\neg x_2 \wedge x_3) = \neg x_1 \diamond \neg(\neg x_2 \wedge x_3) \quad (\text{question 5})$$

$$= \neg x_1 \diamond (\neg x_2 \diamond x_3) \quad (\text{question 3})$$

d'où

$$x_1 \vee (\neg x_2 \wedge x_3) = (x_1 \diamond x_1) \diamond ((x_2 \diamond x_2) \diamond x_3) \quad (\text{question 4})$$

## Centrale Maths 1 MP 2020 — Corrigé

Ce corrigé est proposé par Thierry Limoges (professeur en CPGE) ; il a été relu par Matthias Moreno Ray (professeur en CPGE) et Vincent Puyhaubert (professeur en CPGE).

---

Ce sujet d'algèbre est constitué de deux problèmes indépendants. Le premier traite des fonctions arithmétiques multiplicatives et de leurs applications (parties I et III) ; le second porte sur les endomorphismes de permutation (partie II).

- Dans la première partie, on établit des résultats généraux à propos d'une loi de composition interne  $*$  sur l'ensemble  $\mathbb{A}$  des fonctions arithmétiques. Quatre applications suivent, sur la fonction de Möbius  $\mu$ , la fonction indicatrice d'Euler  $\varphi$ , le déterminant de Smith et les séries de Dirichlet. Cette dernière est la seule qui utilise des outils d'analyse.
- La deuxième partie étudie les endomorphismes de permutation. On commence par des propriétés sur les bijections de  $\llbracket 1 ; n \rrbracket$  qui prolongent celles du cours et on obtient une condition nécessaire et suffisante pour que deux permutations soient conjuguées. On applique ensuite ces résultats aux endomorphismes de permutation, ce qui permet de les caractériser par les traces de leurs puissances successives parmi les endomorphismes diagonalisables.
- Enfin, on s'intéresse à la matrice de Redheffer, dont on montre que 1 est valeur propre avec une multiplicité connue en fonction de sa taille. Il s'agit d'un résultat récent (1977).

Ce sujet de 42 questions est long et technique, hormis quelques questions proches du cours. Il permet d'approfondir certains chapitres de seconde année. On utilise les structures algébriques, notamment le groupe symétrique, les déterminants, la réduction des endomorphismes et, dans une moindre mesure, les séries numériques. À l'exception des questions 14, 18 et 19, la première partie est faisable en MPSI et constitue un entraînement sur les sommes doubles, l'arithmétique et les structures algébriques. La partie II.A peut également être abordée en MPSI.

## INDICATIONS

## Partie I

- 1 Calculer  $\delta * g$  et  $g * \delta$  pour  $g \in \mathbb{A}$ .
- 2 Établir une bijection entre  $\mathcal{C}_n$  et  $\mathcal{D}_n$ .
- 3 Montrer puis utiliser que l'application suivante est involutive :

$$\xi: \begin{cases} \mathcal{C}_n & \longrightarrow \mathcal{C}_n \\ (d_1, d_2) & \longmapsto (d_2, d_1) \end{cases}$$

- 4 Prendre la réunion des  $\mathcal{C}_{n/d_3}$  pour  $d_3|n$ .
- 5 Montrer que  $(\mathbb{A}, +, *)$  est un anneau commutatif.
- 6 Montrer que  $f(n) = g(n)$  pour tout  $n \in \mathbb{N}^*$  par récurrence sur le nombre de facteurs premiers distincts de  $n$ .
- 7 Pour l'injectivité, utiliser le lemme de Gauss. Pour la surjectivité, vérifier que si  $d|nm$ , alors  $(d_1, d/d_1)$  où  $d_1 = \text{pgcd}(d, n)$  est son antécédent par  $\pi$ .
- 8 Transformer la somme à l'aide de la question 7.
- 9 Définir  $g$  d'abord sur les  $p^k$  avec  $p$  premier par récurrence sur  $k$ , puis sur  $\mathbb{N}^*$ . Enfin, calculer  $f * g$ .
- 10 Montrer que  $(\mathbb{M}, *)$  est un groupe abélien.
- 11 Revenir à la définition et utiliser la décomposition d'un entier en produit de facteurs premiers.
- 12 Utiliser les questions 8 et 6.
- 13 Utiliser les questions 12 et 1.
- 14 Montrer que  $n = \sum_{d|n} \varphi(d)$  pour tout  $n \in \mathbb{N}^*$ .
- 15 Après le calcul d'un coefficient de  $M'D^T$  avec la formule sommatoire, utiliser une caractérisation du pgcd.
- 16 Remarquer que  $M'$  et  $D$  sont triangulaires.
- 17 Majorer le module du terme général de la série.
- 18 Raisonner par contraposition, considérer

$$k_0 = \min \{k \in \mathbb{N}^* \mid f(k) \neq g(k)\} \quad \text{et} \quad (L_f(s) - L_g(s)) k_0^s$$

puis faire tendre  $s$  vers  $+\infty$  en justifiant proprement le passage à la limite.

- 19 Utiliser une sommation par paquets. L'hypothèse de multiplicativité n'est pas nécessaire.

## Partie II

- 22 Utiliser un élément de  $\mathfrak{S}_n$  qui envoie le support de  $\gamma_1$  sur celui de  $\gamma_2$ .
- 23 Généraliser la construction de la question précédente à deux composées de plusieurs cycles de longueurs deux à deux correspondantes.
- 24 Remarquer que tout cycle de longueur  $\ell$  est conjugué avec le cycle  $(1 \ 2 \ \dots \ \ell)$  puis développer le déterminant du polynôme caractéristique par rapport à la première ligne.

25 Montrer que  $\sigma$  est conjuguée à une permutation de la forme

$$\tau = (1 \cdots \ell_1)((\ell_1 + 1) \cdots (\ell_1 + \ell_2)) \cdots ((\ell_1 + \cdots + \ell_{n-1} + 1) \cdots (\ell_1 + \cdots + \ell_n))$$

où les  $\ell_i$  sont à préciser.

26 Étudier la multiplicité de la racine  $z = e^{2i\pi/q}$  de  $\chi_\sigma$ .

29 Utiliser la même réduction par blocs qu'à la question 25.

31 Trouver une matrice de permutation ayant même polynôme caractéristique que  $u$ .

32 Pour l'implication réciproque, diagonaliser  $u$  puis obtenir des conditions sur la dimension des sous-espaces propres à l'aide de la trace.

Dans le cas  $k = 3$ , montrer que l'équivalence subsiste en s'inspirant de la question 31. Dans le cas  $k = 4$ , trouver un contre-exemple.

33 Pour l'implication directe, utiliser que  $\mathfrak{S}_n$  est un groupe fini.

Pour l'implication réciproque, considérer  $\sigma$  de type cyclique  $(c_1, \dots, c_n)$  en précisant pourquoi elle existe.

34 Introduire la réunion des valeurs propres de  $u$  et  $v$  puis utiliser le déterminant de Vandermonde sur un système défini par les  $\text{Tr}(u^k) = \text{Tr}(v^k)$ .

35 Calculer  $\text{Tr}(u^k)$  dans le cas où la matrice de  $u$  est de la forme  $\Gamma_\ell$ .

### Partie III

36 Dans le calcul des coefficients de  $C_n$ , considérer aussi le cas  $i = 1, j > 1$ . Développer le déterminant de  $C_n$  par rapport à la première ligne.

37 Calculer  $B_n(\lambda)H_n$  en distinguant différents cas comme dans la question 36, puis développer le déterminant  $\det(B_n(\lambda)(\lambda I_n - H_n))$  par rapport à la première ligne.

38 Calculer  $b * 1$ .

40 Établir une relation de récurrence vérifiée par les  $D_k(m)$  puis en déduire que,

$$\forall m \geq 2 \quad b(m) = \sum_{k=1}^{\lfloor \log_2(m) \rfloor} w^k D_k(m)$$

Enfin, utiliser la question 19.

41 Utiliser la formule de l'indication de la question 40 ci-dessus et la question 37, puis intervertir les deux sommes.

42 Commencer par traiter les cas  $n = 1$  et  $n = 2$ .

Pour  $n \geq 3$ , remarquer qu'on peut factoriser  $\chi_n$  sous la forme

$$\chi_n(\lambda) = (\lambda - 1)^{n - \lfloor \log_2(n) \rfloor - 1} Q(\lambda) \quad \text{avec} \quad Q(1) \neq 0$$

## I. QUELQUES RÉSULTATS UTILES

**1** Soit  $g \in \mathbb{A}$ . D'une part, si  $d \geq 2$ , alors  $\delta(d) = 0$ . Donc, pour tout  $n \in \mathbb{N}^*$ ,

$$(\delta * g)(n) = \delta(1)g(n) = g(n)$$

Ainsi  $\delta * g = g$ . D'autre part, pour tout  $n \in \mathbb{N}^*$ , si  $d|n$  et  $d < n$ , alors  $n/d \geq 2$ , donc  $\delta(n/d) = 0$ . D'où

$$(g * \delta)(n) = g(n)\delta(1) = g(n)$$

Ainsi  $g * \delta = g$ . En conclusion,

La fonction  $\delta$  est un élément neutre pour  $*$ .

**2** L'application  $\varphi: \begin{cases} \mathcal{C}_n & \longrightarrow \mathcal{D}_n \\ (d_1, d_2) & \longmapsto d_1 \end{cases}$  est une bijection

de réciproque  $\psi: \begin{cases} \mathcal{D}_n & \longrightarrow \mathcal{C}_n \\ d & \longmapsto (d, n/d) \end{cases}$

Ainsi, pour tout  $n \in \mathbb{N}^*$ ,

$$(f * g)(n) = \sum_{d \in \mathcal{D}_n} f(d)g\left(\frac{n}{d}\right) = \sum_{(d_1, d_2) \in \mathcal{C}_n} f(d_1)g(d_2)$$

soit

$$\forall n \in \mathbb{N}^* \quad (f * g)(n) = \sum_{(d_1, d_2) \in \mathcal{C}_n} f(d_1)g(d_2)$$

**3** Remarquons que pour tout  $(d_1, d_2) \in \mathcal{C}_n$ , on a  $(d_2, d_1) \in \mathcal{C}_n$  et réciproquement. Aussi, l'application

$$\xi: \begin{cases} \mathcal{C}_n & \longrightarrow \mathcal{C}_n \\ (d_1, d_2) & \longmapsto (d_2, d_1) \end{cases}$$

est une bijection involutive de  $\mathcal{C}_n$ . Ainsi, pour tout  $n \in \mathbb{N}^*$ ,

$$(f * g)(n) = \sum_{(d_1, d_2) \in \mathcal{C}_n} f(d_1)g(d_2) = \sum_{(d_2, d_1) \in \mathcal{C}_n} f(d_1)g(d_2) = \sum_{(d_2, d_1) \in \mathcal{C}_n} g(d_2)f(d_1)$$

par commutativité du produit dans  $\mathbb{C}$ . Les variables  $d_1$  et  $d_2$  étant muettes,

$$(f * g)(n) = (g * f)(n)$$

en appliquant le résultat de la question 2 dans lequel on a échangé les rôles de  $f$  et  $g$ . D'où  $f * g = g * f$  et par suite

La loi  $*$  est commutative.

## Centrale Maths 2 MP 2020 — Corrigé

Ce corrigé est proposé par Rémi Pellerin (ENS Lyon) ; il a été relu par Julie Gauthier (professeur agrégé) et Benjamin Monmege (enseignant-chercheur à l'université).

---

Ce sujet traite d'espaces à noyau reproduisant, qui sont des espaces préhilbertiens réels de fonctions d'un intervalle  $I$  vers  $\mathbb{R}$  pour lesquels toutes les applications  $f \mapsto f(x)$  sont des formes linéaires continues et tels que pour tout  $x \in I$ , il existe une fonction  $k_x : I \rightarrow \mathbb{R}$  représentant toutes les fonctions  $f$  par le biais de l'égalité  $f(x) = \langle k_x, f \rangle$ . Il est composé de quatre parties.

- Dans la première, on montre une propriété élémentaire sur les vecteurs propres des endomorphismes autoadjoints. La question 5 sera plusieurs fois utile par la suite.
- La partie II introduit un opérateur  $T$  défini avec une intégrale dont on obtient une expression analytique. Pour cela, le sujet propose une étude exhaustive de ses éléments propres. La dimension de l'espace ambiant est infinie, ce qui rend la chose un peu délicate.
- La partie suivante étudie trois exemples d'espaces à noyau reproduisant et un contre-exemple en utilisant l'opérateur  $T$  de la partie II. Signalons que la dernière question est la plus difficile car elle demande de la créativité.
- La partie IV contient deux sous-parties et propose d'établir des résultats généraux sur les espaces à noyau reproduisant. On commence par montrer que tous les éléments d'un tel espace sont continus, puis on voit comment construire un espace à noyau reproduisant à partir d'une généralisation de l'opérateur  $T$  introduit en partie II.

Ce sujet est très long et assez répétitif. Il était impossible de le terminer dans le temps imparti, même en rédigeant de façon très lapidaire ! Le jour de l'épreuve, il ne fallait pas hésiter à traiter en priorité les questions les plus simples. Les parties, et même les sous-parties, sont assez indépendantes et ne sont pas rangées par difficulté croissante. Les questions 14 et 16 sont longues et calculatoires. Il fallait aussi prendre garde à ne pas confondre les différents produits scalaires et normes manipulés au cours de l'épreuve.

Ce sujet permet de revoir les notions essentielles concernant les espaces préhilbertiens réels et de travailler certaines subtilités entre dimensions finie et infinie. Le chapitre d'intégration est lui aussi à l'honneur car il constitue la base de chaque exemple à l'exception de celui sur les séries entières. Enfin, certaines questions de la partie II demandent une bonne maîtrise des équations différentielles, notamment dans leur aspect calculatoire.



## INDICATIONS

## Partie I

- 2 Développer l'expression de  $\varphi(t)$ .  
 3 Montrer que 0 est un extremum local pour  $\varphi$ .  
 4 Utiliser la formule établie à la question 2.  
 5 Remarquer que  $u$  stabilise  $(\mathbb{R}x_0)^\perp$ .

## Partie II

- 7 Introduire

$$\Delta^- = \{(s, t) \in [0; 1]^2 \mid t < s\} \quad \text{et} \quad \Delta^+ = \{(s, t) \in [0; 1]^2 \mid t > s\}$$

et établir la continuité de  $K$  en tout point de la diagonale  $\{(s, s) \in [0; 1]^2\}$ .

- 10 Commencer par établir que

$$\forall k \in \mathbb{N} \quad \forall s \in [0; 1] \quad T(p_k)''(s) = -s^k$$

- 12 La fonction  $k_s$  n'étant pas dérivable en  $s$ , découper l'intégrale en deux sur  $[0; s[$  et  $]s; 1]$ .  
 14 Raisonner par analyse-synthèse.  
 16 Utiliser la question 13 pour justifier que  $T$  n'admet pas 0 pour valeur propre. Soit  $\lambda \in \mathbb{R}$  une valeur propre de  $T$ . Résoudre l'équation différentielle

$$y'' + \frac{1}{\lambda} y = 0$$

en discutant suivant le signe de  $\lambda$ . Montrer, en utilisant les résultats des questions 11 et 15, que si  $\lambda < 0$  ou que  $\lambda \in \pi\mathbb{N}^*$ , alors  $\lambda$  n'est pas valeur propre.

- 17 Remarquer que  $\forall (f, g) \in E^2 \quad \langle T(f), g \rangle = \langle T(f), T(-g)'' \rangle$   
 18 Utiliser les questions 1, 5 et 17.  
 19 On rappelle que

$$\forall (p, q) \in \mathbb{R}^2 \quad \cos(p) - \cos(q) = -2 \sin\left(\frac{p+q}{2}\right) \sin\left(\frac{p-q}{2}\right)$$

- 21 Utiliser la question 16 pour obtenir une expression de  $T(g_k)$  puis la question 20.  
 22 On rappelle qu'une famille de vecteurs  $(x_i)_{i \in I}$  est totale dans l'espace vectoriel  $E$  si, par définition,

$$\overline{\text{Vect} \{x_i \mid i \in I\}} = E$$

Utiliser la question 8.

### Partie III

- 25 Attention, on n'a pas a priori  $T(f'') = T(f)''$ . Montrer directement en intégrant par parties que  $T(f'') = -f$ .
- 26 La démarche suggérée par l'énoncé, à savoir utiliser le résultat de la question 25, semble vouée à l'échec ou du moins, assez délicate... Calculer plutôt directement  $U(f)(s)$  pour  $f \in E_1$  et  $s \in [0; 1]$  en découpant l'intégrale suivant une subdivision bien choisie contenant  $s$ .
- 28 Construire une suite de fonctions  $(f_n)$  qui converge vers 0 pour la norme associée au produit scalaire mais qui ne converge pas simplement vers la fonction nulle. On pourra par exemple imposer  $f_n(0) = 1$  pour tout  $n \in \mathbb{N}$ .
- 30 Établir que  $\forall (x, y) \in \mathbb{R}^2 \quad x^2 + y^2 \geq 2 |x y|$
- 33 Remarquer que :  $\forall (x, y) \in \mathbb{R}^2 \quad \min(x, \cdot)'(y) = \begin{cases} 1 & \text{si } y < x \\ 0 & \text{si } y > x \end{cases}$
- 34 Chercher un produit scalaire de la forme

$$\langle \cdot, \cdot \rangle : \begin{cases} E_4 & \longrightarrow \mathbb{R} \\ (f, g) & \longmapsto \int_0^a \omega(t) f'(t) g'(t) dt \end{cases}$$

où  $\omega : [0; a] \rightarrow \mathbb{R}$  est une application à déterminer.

### Partie IV

- 35 Commencer par montrer que  $N(V_x) \leq \sqrt{\langle k_x, k_x \rangle}$  pour tout  $x \in I$ . Pour conclure à l'égalité, distinguer les cas  $k_x = 0$  et  $k_x \neq 0$ . On pourra étudier la fonction  $f : x \mapsto k_x / \|k_x\|$ .
- 36 Remarquer que :  $\forall f \in E \quad \forall x \in I \quad f(x) : x \mapsto \langle \cdot, f \rangle \circ k_x$
- 37 Rappelons que si  $F$  est un sous-espace vectoriel de  $E$  de dimension finie, alors  $F^\perp$  est un supplémentaire. Utiliser le théorème du rang.
- 38 Commencer par montrer que pour tout  $x \in [0; 1]$ ,  $A(x, \cdot) \in (\text{Ker } T)^\perp$  où l'orthogonal est pris au sens du produit scalaire  $\langle \cdot, \cdot \rangle$ . Introduire un élément  $f_x \in \text{Im } T$  tel que  $S(f_x) = A(x, \cdot)$ .

## I. PRÉLIMINAIRES

**1** Soit  $x \in F^\perp$ . Montrons que  $u(x) \in F^\perp$ . Pour  $y \in F$ ,

$$\langle u(x), y \rangle = \langle x, u(y) \rangle = 0$$

car  $u(y) \in F$  et  $x \in F^\perp$ . Ainsi,  $u(x) \in F^\perp$ , ceci pour tout  $x \in F^\perp$ . Par conséquent,

Si  $F$  est stable par  $u$  alors  $F^\perp$  l'est aussi.

Un endomorphisme  $u$  vérifiant la propriété

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

est dit *autoadjoint*.

**2** Pour  $t \in \mathbb{R}$ ,

$$\begin{aligned} \varphi(t) &= \langle u(x_0 \cos(t) + y \sin(t)), x_0 \cos(t) + y \sin(t) \rangle \\ &= \langle \cos(t) u(x_0) + \sin(t) u(y), x_0 \cos(t) + y \sin(t) \rangle \quad (u \text{ est linéaire}) \\ \varphi(t) &= \cos^2(t) \langle u(x_0), x_0 \rangle + \sin^2(t) \langle u(y), y \rangle + 2 \cos(t) \sin(t) \langle u(x_0), y \rangle \end{aligned}$$

ceci car  $\langle u(x_0), y \rangle = \langle x_0, u(y) \rangle$  et par bilinéarité du produit scalaire. Ainsi,  $\varphi$  est une combinaison linéaire de fonctions de classe  $\mathcal{C}^1$  donc

L'application  $\varphi$  est de classe  $\mathcal{C}^1$  sur  $\mathbb{R}$ .

**3** Soit  $t \in \mathbb{R}$ . On a alors

$$\begin{aligned} \|\gamma(t)\|^2 &= \langle \gamma(t), \gamma(t) \rangle \\ &= \langle x_0 \cos(t) + y \sin(t), x_0 \cos(t) + y \sin(t) \rangle \\ &= \cos^2(t) \|x_0\|^2 + \sin^2(t) \|y\|^2 \quad (x_0 \perp y) \\ &= \cos^2(t) + \sin^2(t) \quad (x_0 \text{ et } y \text{ étant unitaires}) \\ \|\gamma(t)\|^2 &= 1 \end{aligned}$$

Ainsi,

$$\forall t \in \mathbb{R} \quad \|\gamma(t)\| = 1$$

De ce fait,  $\forall t \in \mathbb{R} \quad \varphi(t) = \langle u(\gamma(t)), \gamma(t) \rangle \leq \langle u(x_0), x_0 \rangle$

ce qui prouve que  $\varphi$  a un maximum global donc local en 0. L'application  $\varphi$  étant dérivable au voisinage de 0 d'après le résultat de la question 2, il s'ensuit que

$$\varphi'(0) = 0$$

**4** Dérivons l'expression de  $\varphi$  obtenue à la question 2. Pour tout  $t \in \mathbb{R}$ ,

$$\begin{aligned} \varphi'(t) &= -2 \cos(t) \sin(t) \langle u(x_0), x_0 \rangle + 2 \cos(t) \sin(t) \langle u(y), y \rangle \\ &\quad - 2 \sin^2(t) \langle u(x_0), y \rangle + 2 \cos^2(t) \langle u(x_0), y \rangle \end{aligned}$$

Par conséquent,  $\varphi'(0) = 2 \langle u(x_0), y \rangle$ . Or,  $\varphi'(0) = 0$  d'après le résultat de la question 3. Il s'ensuit que  $\langle u(x_0), y \rangle = 0$  puis que

Les vecteurs  $y$  et  $u(x_0)$  sont orthogonaux.

## Centrale Informatique MP-PC-PSI 2020 — Corrigé

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

---

Ce sujet étudie la génération de photomosaïques, qui sont des images composées d'autres images plus petites, à la manière d'une mosaïque. Une image source de grande taille est décomposée en petits pavés qui sont remplacés par des vignettes issues d'une banque d'images dont les caractéristiques ressemblent à celles des pavés sources.

- Dans la partie I, le sujet étudie des images constituées de pixels représentés par leurs niveaux de rouge, de vert et de bleu. La partie se termine par une fonction de conversion d'une image colorée en niveaux de gris. Le sujet ne traitera ensuite que des images en niveaux de gris, plus simples à transformer.
- Pour réaliser une mosaïque, il faut redimensionner des images, que ce soit l'image source ou les vignettes de la banque. C'est l'objectif de la partie II. Deux algorithmes y sont étudiés, à base d'interpolation (dans le cas d'une réduction ou d'une augmentation de la taille de l'image) puis de moyenne locale (uniquement pour la réduction). L'énoncé fournit plusieurs implémentations du second algorithme et demande de les comparer.
- La partie III traite du stockage de la banque d'images dans une base de données. Il faut écrire des requêtes pour extraire certaines photographies de la banque, en se fondant sur leur auteur, leur contenu et les mots-clés qui les caractérisent. Deux questions plus ouvertes demandent d'imaginer comment modifier la base de données pour permettre l'internationalisation de ces mots-clés.
- La dernière partie concerne le cœur de la génération de photomosaïques, à savoir le placement des vignettes sur l'image source. La comparaison d'une vignette et du pavé correspondant de l'image source est réalisée à l'aide d'une norme 1. À nouveau, la partie se termine sur deux questions plus ouvertes et délicates, demandant d'améliorer l'algorithme proposé dans l'énoncé.

Outre le thème original et amusant de ce sujet, les différentes parties du programme sont utilisées dans le sujet : représentation des nombres en mémoire, algorithmique des listes, programmation Python (en particulier avec la bibliothèque NumPy) et bases de données (bien que le sujet utilise des opérateurs SQL à la limite du programme qui rend donc la partie III difficile). Le sujet est entièrement réalisable à la fin de la première année, puisqu'il n'utilise ni récursivité ni tri de tableaux (sauf éventuellement dans les deux dernières questions plus ouvertes). Cela en fait donc un excellent sujet de révision, permettant de plus d'aboutir à un code utilisable pour réaliser vos propres photomosaïques !

## INDICATIONS

### Partie I

- 2 Lire le reste de la page 2 de l'énoncé pour trouver de l'inspiration sur la commande à utiliser.
- 4 Ne pas hésiter à utiliser les fonctions NumPy rappelées en fin d'énoncé.
- 6 Pour parcourir les pixels d'une image, on utilise deux boucles `for` imbriquées.

### Partie II

- 8 Il faut utiliser l'opération de division entière pour prendre la partie entière des indices interpolés.
- 10 Commencer par exhiber l'ensemble des entiers `I` et `J` que parcourent les deux boucles imbriquées. Utiliser ensuite le fait que ces ensembles sont respectivement en bijection avec  $\llbracket 0; h-1 \rrbracket$  et  $\llbracket 0; w-1 \rrbracket$ .
- 13 Attention, une solution naïve consistant à parcourir la matrice `S` à remplir et calculer chaque somme indépendamment ne donne pas la complexité linéaire en `N` demandée. Remarquer plutôt que, dès que  $\ell$  et  $c$  sont strictement positifs,  $S(\ell, c)$  peut s'obtenir à partir de  $S(\ell-1, c)$  et des coefficients  $A(\ell-1, j)$  pour  $j \in \llbracket 0; c-1 \rrbracket$ .
- 14 Il faut notamment comprendre pourquoi la variable `X` en ligne 8 contient exactement le numérateur de la moyenne `np.mean(A[I : I+ph, J : J+pw])` calculée dans la fonction `moyenneLocale`.
- 16 Le tableau `sred` calculé en ligne 4 contient les seuls coefficients de la matrice `S` qu'on utilise dans `réductionSommmation1`. Expliquer ensuite pourquoi les lignes 5 et 6 permettent de calculer simultanément toutes les valeurs des coefficients `X` en ligne 8 de `réductionSommmation1`.

### Partie III

- 20 Utiliser l'opérateur d'agrégation `COUNT(*)` combiné à une jointure des tables `Photo` et `Personne`.
- 21 L'information peut être extraite des jointures des tables `Photo`, `Decrit` et `Motcle`.
- 22 Une solution consiste à utiliser l'opérateur `INTERSECT` calculant l'intersection des résultats de deux requêtes.
- 23 L'opérateur `INTERSECT`, couplé avec l'opérateur `EXCEPT`, permet d'obtenir le résultat en combinant trois requêtes quasiment identiques.

### Partie IV

- 26 Traiter différemment le cas où le redimensionnement est une réduction d'un facteur entier (permettant d'utiliser l'algorithme de `réductionSommmation`) du cas général où la fonction `procheVoisin` est incontournable.

## I. PIXELS ET IMAGES

1 Chaque composante RGB d'un pixel étant un entier naturel codé sur 8 bits, elle peut prendre  $2^8 = 256$  valeurs différentes. Puisqu'il y a trois composantes,

Un pixel dont chaque composante RGB est codée sur 8 bits peut prendre  $256^3 = 16\,777\,216$  couleurs différentes.

2 En synthèse additive, le blanc consiste à donner l'intensité maximale aux trois composantes. Sur 8 bits, l'intensité maximale est  $2^8 - 1 = 255$ . On obtient donc un pixel blanc à l'aide de la commande :

```
np.array([255, 255, 255], np.uint8)
```

L'utilisation de types, tels que `np.uint8` pour les entiers non signés sur 8 bits, n'est pas officiellement au programme et peut désarçonner en première lecture. On avait donc tout intérêt, comme d'habitude, à avoir lu le sujet en entier pour trouver des indices sur la marche à suivre. On imagine cependant que la proposition `np.array([255, 255, 255])` aura été considérée avec bienveillance par les correcteurs, même si le paramètre `dtype` manquant est alors interprété par défaut comme un entier signé sur 64 bits.

On peut également utiliser la fonction `full` de la bibliothèque NumPy :

```
np.full(3, 255, np.uint8)
```

3 Même si utiliser `np.uint8` peut être perturbant si on n'y est pas familier, le comportement de NumPy, et donc les réponses attendues, sont en tout point similaires avec le phénomène de dépassement de capacité qui est bien au programme. Si on essaie de coder l'entier naturel 280 en binaire, on obtient 100011000 : en ne conservant que les huit derniers bits, on obtient alors 00011000 qui code bien 24. Ceci revient à calculer la valeur dans l'intervalle  $\llbracket 0; 255 \rrbracket$  qui est congrue à 280 modulo 256.

En ayant posé `a = np.uint8(280)` et `b = np.uint8(240)`, alors

- `a` vaut 24, puisque 280 dépasse la limite de 255 autorisée sur 8 bits et

$$24 \equiv 280 \pmod{256}$$

- `b` vaut 240 ;
- `a+b` vaut 8, puisque  $24 + 240 = 264$  dépasse la limite autorisée sur 8 bits et

$$8 \equiv 264 \pmod{256}$$

- `a-b` vaut 40, puisque  $24 - 240 = -216$ , qui n'est pas un entier non signé, et

$$40 \equiv -216 \pmod{256}$$

- `a//b`, quotient dans la division entière de 24 par 240, vaut 0 ;
- `a/b` vaut le flottant approximant le réel 0,1.

4 On réalise la moyenne des trois composantes d'un pixel à l'aide de la fonction `np.mean`, puis on arrondit ce résultat à l'entier le plus proche avec `round`, qu'on transforme ensuite en entier non signé sur 8 bits :

```
def gris(p):
    return np.uint8(round(np.mean(p)))
```

Sans l'utilisation de `round`, la fonction `np.uint8` renvoie l'entier le plus proche par valeur inférieure, ce qui n'est pas le comportement attendu ici.

5 L'expression `source.shape` renvoie les dimensions du tableau NumPy `source`. L'image contenue dans le fichier `surfer.jpg` a donc une hauteur de 3000 pixels, une largeur de 4000 pixels et chaque pixel est représenté par 3 composantes.

Contrairement à l'usage du langage courant où l'on a tendance à donner la largeur avant la hauteur, c'est bien le nombre de lignes qui apparaît en premier dans le tuple de dimensions en NumPy.

L'expression `source[0,0]` renvoie les informations du pixel en haut à gauche de l'image, qui est donc un pixel avec  $144/255 \approx 56\%$  de rouge,  $191/255 \approx 75\%$  de vert et  $221/255 \approx 87\%$  de bleu.

Il s'agit d'un pixel de couleur bleu ciel, comme on peut effectivement le voir en haut à gauche de l'image du surfeur en figure 1 dans le sujet original en couleur.

6 Parcourons l'image pixel par pixel à l'aide de deux boucles `for` imbriquées, pour appliquer à chaque pixel la fonction `gris` de la question 4. On stocke la nouvelle image dans un tableau NumPy qu'on a préalablement initialisé.

```
def conversion(a):
    h, w, p = a.shape
    image_gris = np.empty((h, w), np.uint8)
    for i in range(h):
        for j in range(w):
            image_gris[i, j] = gris(a[i, j])
    return image_gris
```

On peut proposer une version bien plus efficace n'utilisant pas la fonction `gris`, mais plutôt les opérations matricielles de la bibliothèque NumPy (largement optimisées vis-à-vis des boucles `for`). En appliquant directement la fonction `np.mean` avec un second argument égal à 2 (ce qui revient à faire la moyenne le long du troisième axe du cube de données, soit sur les trois couleurs d'un pixel), on peut calculer un tableau bidimensionnel contenant les moyennes de chaque pixel. L'utilisation de la fonction `np.round` plutôt que `round` permet d'appliquer l'opération d'arrondi sur chaque élément du tableau.

```
def conversion(a):
    return np.uint8(np.round(np.mean(a, 2)))
```

Un candidat qui maîtrise ce genre d'optimisations utilisant la bibliothèque NumPy a tout intérêt à l'employer pour gagner du temps dans la rédaction le jour J ; mais au moindre doute, mieux vaut assurer en utilisant les boucles `for` pour éviter de perdre des points.

# Centrale Informatique optionnelle MP 2020

## Corrigé

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

---

Ce sujet porte sur les systèmes de vote dans lesquels l'électeur classe les candidats sur son bulletin par ordre de préférence. On peut alors modéliser l'ensemble des votes par une matrice traduisant, pour chaque couple de candidats, les préférences des électeurs. Les systèmes de vote par préférence diffèrent selon le mode de désignation du ou des vainqueurs de l'élection.

- La partie I permet de se familiariser avec les notions de vote et de graphe par préférence à travers deux exemples. On y démontre également le théorème de McGarvey, stipulant que toute matrice de préférence dont les coefficients sont pairs correspond aux résultats d'un dépouillement. La preuve proposée est constructive et donne également lieu à des questions de programmation.
- La partie suivante porte sur deux modes de désignation du ou des vainqueurs d'un vote par préférence. On y étudie rapidement la méthode de Condorcet, consistant à comparer les candidats deux par deux pour déclarer vainqueur celui qui est préféré à tous les autres, s'il existe. Cette méthode ne garantit hélas pas la présence d'un vainqueur. Le reste de la partie, qui constitue le cœur du problème, est consacré à la méthode de Schulze, qui est adaptée de l'algorithme de Floyd-Warshall au programme de l'option informatique. On prouve notamment que cette méthode assure la désignation d'un vainqueur.
- Enfin, la partie III étudie un problème de décision (relativement éloigné des deux premières parties) faisant intervenir la notion de vainqueur de Schulze. On y démontre notamment que le problème étudié est au moins aussi difficile que le problème de satisfiabilité d'une formule logique. Cette dernière partie de logique ne peut être convenablement abordée qu'après une bonne compréhension des notions introduites dans la partie précédente.

L'énoncé comporte 28 questions de difficulté globalement croissante. Seul un quart des questions nécessitent de programmer en OCaml, et la plupart d'entre elles sont abordables. Certains passages sont en revanche plus délicats et nécessitent une bonne compréhension, notamment la description de la méthode de Schulze ou encore de l'algorithme étudié dans la partie III. Ce sujet permet de bien s'entraîner sur les graphes, dans leurs aspects tant algorithmiques que théoriques.



## INDICATIONS

### Partie I

- 1 On peut commencer par écrire une fonction examinant un bulletin donné et renvoyant 1 ou  $-1$  selon le candidat classé en premier parmi  $i$  et  $j$ .
- 3 Pour la parité des coefficients non diagonaux, utiliser le nombre total  $p$  de bulletins dans l'urne.
- 5 Considérer par exemple le bulletin

$$b_{i,j,n} = [i, j, 0, 1, \dots, i-1, i+1, \dots, j-1, j+1, \dots, n-1]$$

et construire le second bulletin de sorte que l'urne obtenue soit associée à  $E_{i,j,n}$ .

- 6 On doit obtenir  $M_3 = M_1 + M_2$ .
- 7 Écrire  $M$  sous la forme d'une combinaison linéaire des matrices  $E_{i,j,n}$ , puis utiliser les questions 5 et 6.
- 8 Ne pas hésiter à décomposer le code en plusieurs fonctions. Commencer par exemple par écrire une fonction implémentant la construction effectuée à la question 5. La fonction `List.rev` pourra être utile à cet effet.
- 9 Le code précédent étant composé de plusieurs fonctions, il est nécessaire de commencer par étudier la complexité des fonctions auxiliaires. Attention à la complexité de l'opérateur `@` en OCaml.

### Partie II

- 11 Pour décider si un sommet donné est un vainqueur de Condorcet, il suffit d'examiner une seule ligne de la matrice.
- 13 Considérer un chemin  $\rho$  comportant une boucle et démontrer qu'en retirant cette boucle on obtient un chemin de poids supérieur ou égal au poids de  $\rho$ .
- 14 Introduire deux chemins, de  $i$  à  $k$  et de  $k$  à  $j$ , de poids respectifs  $I[i, k]$  et  $I[k, j]$ , puis construire à l'aide de ces chemins un chemin de  $i$  à  $j$ .
- 15 Adapter la relation de récurrence de l'algorithme de Floyd-Warshall, ainsi que les conditions pour appliquer cette relation.
- 17 Comparer pour les deux algorithmes le contexte d'utilisation et la complexité.
- 18 Pour la parité des coefficients, justifier que ceux de la matrice  $I$  apparaissent tous nécessairement dans  $M$ .
- 22 Dans le cas où  $I[i, j] \leq I[j, k]$ , utiliser le résultat de la question 14 appliqué à  $I[j, i]$ , puis à  $I[i, k]$ . Le cas  $I[i, j] > I[j, k]$  se traite de façon similaire.
- 23 En raisonnant par l'absurde, construire une suite d'entiers naturels  $(i_k)_{k \in \mathbb{N}}$  strictement inférieurs à  $n$  telle que  $S[i_k, i_{k+1}] < 0$  pour tout  $k \in \mathbb{N}$ .

### Partie III

- 25 Les rôles de  $x$  et  $\neg x$  sont symétriques. Dans le cas où le candidat  $b_x$  a été choisi, calculer le coefficient  $S[c, a_2]$  dans la matrice de préférence de Schulze.
- 26 S'il existe une variable  $x$  pour laquelle les sommets  $b_x$  et  $b_{\neg x}$  ont été choisis, justifier que pour une autre variable  $y$ , ni  $b_y$  ni  $b_{\neg y}$  n'ont été choisis.
- 27 Procéder par double implication. Pour le sens direct, il faut choisir quels sommets sélectionner en fonction de l'interprétation de la formule. Pour le calcul de  $I$ , on se limitera à la ligne et la colonne correspondant au champion  $c$ .
- 28 Justifier que l'algorithme de transformation de formule en élection s'exécute en temps polynomial.

## I. VOTE PAR PRÉFÉRENCE

**1** Commençons par écrire une fonction `traite_bulletin` qui compare les candidats  $i$  et  $j$  sur un bulletin donné, et renvoie 1 ou  $-1$  en fonction du candidat classé en premier parmi  $i$  et  $j$ . Pour cela, on parcourt récursivement la liste associée jusqu'à trouver pour la première fois l'entier  $i$  ou l'entier  $j$ .

```
let rec traite_bulletin i j b = match b with
  | [] -> failwith "i et j absents du bulletin"
  | x::q -> if x = i then 1
             else if x = j then -1
             else traite_bulletin i j q;;
```

Une erreur classique consiste à utiliser directement les noms de variable  $i$  et  $j$  dans les motifs du filtrage :

```
| i::q -> 1
| j::q -> -1
| x::q -> traite_bulletin i j q
```

En effet, on ne peut utiliser dans les filtrages en OCaml que des constructeurs (comme `::` sur les listes), des constantes, mais pas des valeurs calculées.

Le résultat de cette fonction est incorrect si  $i = j$ . Ce cas, où il n'y a pas réellement de candidats à comparer, intervient pourtant dans la construction de la matrice d'adjacence à la question 4. Cette situation sera gérée dans la fonction `duel` directement.

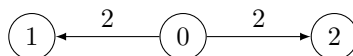
Le résultat de la comparaison des deux candidats s'obtient alors en additionnant les résultats des appels à la fonction `traite_bulletin` sur chaque bulletin de l'urne.

```
let rec duel i j u = match u with
  | _ when i = j -> 0 (* voir remarque precedente *)
  | [] -> 0
  | b::q -> (traite_bulletin i j b) + (duel i j q);;
```

**2** Soit  $M = (m_{ij})_{0 \leq i, j \leq 2}$  la matrice d'adjacence du graphe de préférence de l'urne. Considérons par exemple les candidats 0 et 2 : un seul bulletin (le deuxième) classe 2 avant 0, tandis que les trois autres classent 0 avant 2. On obtient alors  $m_{0,2} = 3 - 1 = 2$  et  $m_{2,0} = 1 - 3 = -2$ . En poursuivant de la même manière pour les autres couples de candidats, on en déduit que

$$M = \begin{pmatrix} 0 & 2 & 2 \\ -2 & 0 & 0 \\ -2 & 0 & 0 \end{pmatrix}$$

En ne représentant que les arcs ayant un poids strictement positif, le graphe de préférence associé à cette urne est le suivant.



**3** Soit  $U$  une urne, et soit  $M = (m_{i,j})_{0 \leq i, j \leq n-1}$  la matrice d'adjacence de son graphe de préférence, où  $n$  désigne le nombre de candidats. Pour tout couple  $(i, j)$  de candidats, notons  $d_{i,j}$  le nombre de bulletins qui placent  $i$  avant  $j$ . Pour tout candidat  $i$ , on a par convention  $m_{i,i} = 0$ . De plus, pour tout couple  $(i, j)$  de candidats distincts, on obtient par définition de la matrice  $M$

$$m_{i,j} = d_{i,j} - d_{j,i} = -(d_{j,i} - d_{i,j}) = -m_{j,i}$$

ce qui prouve que

La matrice d'adjacence d'un graphe de préférence est antisymétrique.

Par ailleurs, si  $p$  désigne le nombre de bulletins dans l'urne, on a  $d_{j,i} = p - d_{i,j}$  pour tout couple  $(i, j)$  de candidats distincts. Dès lors, on obtient avec la même hypothèse  $m_{i,j} = d_{i,j} - (p - d_{i,j}) = 2d_{i,j} - p$ . L'entier  $2d_{i,j}$  étant pair, la parité du coefficient  $m_{i,j}$  est donc indépendante de  $i$  et  $j$ . Autrement dit,

Les coefficients non diagonaux d'une matrice d'adjacence d'un graphe de préférence ont tous la même parité.

On a montré plus précisément que cette parité correspond à la parité du nombre  $p$  de bulletins dans l'urne.

**4** Commençons par créer une matrice nulle, puis utilisons la fonction `duel` de la question 1 pour calculer ses coefficients. On obtient la fonction suivante.

```
let depouillement n u =
  let mat = Array.make_matrix n n 0 in
  for i = 0 to (n-1) do
    for j = 0 to (n-1) do
      mat.(i).(j) <- duel i j u
    done
  done;
  mat;;
```

**5** Supposons que  $i < j$  sans perte de généralité. Définissons les deux listes ordonnées

$$b_{i,j,n} = [i, j, 0, 1, \dots, i-1, i+1, \dots, j-1, j+1, \dots, n-1]$$

et

$$b'_{i,j,n} = [n-1, \dots, j+1, j-1, \dots, i+1, i-1, \dots, 1, 0, i, j]$$

Tous les entiers entre 0 et  $n-1$  sont classés et ce sont les seuls. Dès lors, les deux listes précédentes définissent deux bulletins, et on définit  $U_{i,j,n} = [b_{i,j,n}, b'_{i,j,n}]$  l'urne contenant ces deux bulletins. On pose enfin  $M = \text{Mat}(U_{i,j,n})$  la matrice d'adjacence du graphe de préférence associé à  $U_{i,j,n}$ .

Démontrons à présent que  $M = E_{i,j,n}$ . Ces deux matrices étant antisymétriques, il suffit de montrer que leurs coefficients d'indices  $(k, \ell)$  vérifiant  $k < \ell$  sont égaux deux à deux. Si  $k = i$  et  $\ell = j$ , alors les deux bulletins placent  $i$  avant  $j$ , donc  $m_{i,j} = 2$  et  $m_{j,i} = -2$ , correspondant aux seuls coefficients non nuls de la matrice  $E_{i,j,n}$ . Il reste à montrer que tous les autres coefficients sont nuls. Il y a trois cas :

- Si  $k \in \{i, j\}$  et  $\ell \notin \{i, j\}$ , alors  $b_{i,j,n}$  place  $k$  avant  $\ell$  et  $b'_{i,j,n}$  place  $\ell$  avant  $k$ , donc  $m_{k,\ell} = 0$ .
- De même, si  $\ell \in \{i, j\}$  et  $k \notin \{i, j\}$ ,  $b_{i,j,n}$  place  $\ell$  avant  $k$  et  $b'_{i,j,n}$  place  $k$  avant  $\ell$ , d'où  $m_{k,\ell} = 0$ .
- Si  $k \notin \{i, j\}$  et  $\ell \notin \{i, j\}$ ,  $b_{i,j,n}$  place  $k$  avant  $\ell$  (les candidats différents de  $i$  et  $j$  sont placés dans l'ordre croissant) et  $b'_{i,j,n}$  place  $\ell$  avant  $k$  (les candidats différents de  $i$  et  $j$  sont placés dans l'ordre décroissant), soit  $m_{k,\ell} = 0$ .

## Mines Maths 1 MP 2020 — Corrigé

Ce corrigé est proposé par Matthias Moreno Ray (professeur en CPGE) ; il a été relu par Céline Chevalier (enseignant-chercheur à l'université) et Gilbert Monna (professeur honoraire en CPGE).

---

Soit  $E$  un espace vectoriel euclidien de dimension  $n$ . Un endomorphisme  $u \in \mathcal{L}(E)$  est dit *nilpotent* lorsqu'il existe  $p \in \mathbb{N}^*$  tel que  $u^p = 0$ . Ce problème s'intéresse aux sous-espaces vectoriels  $\mathcal{V}$  de  $\mathcal{L}(E)$  qui ne contiennent que des endomorphismes nilpotents. L'objectif du problème est de montrer qu'un tel espace  $\mathcal{V}$  a toujours une dimension inférieure ou égale à  $n(n-1)/2$ , puis de déterminer les sous-espaces de ce type qui sont de dimension maximale. Plus précisément, on montre que dans ce cas il existe une base  $B$  de  $E$  telle que tout élément de  $\mathcal{V}$  est représenté dans  $B$  par une matrice triangulaire supérieure stricte (c'est-à-dire avec des zéros sur la diagonale). On dit que l'on a fait une « réduction simultanée » des endomorphismes de  $\mathcal{V}$ . La démonstration se fait par récurrence sur la dimension de  $E$ . Il s'agit d'un résultat relativement récent (théorème de Gerstenhaber, 1958).

- Dans la première partie, on montre quelques résultats techniques sur les endomorphismes nilpotents. Les questions de cette partie sont de bons exercices d'algèbre linéaire.
- Dans la deuxième partie, on met en place un mode de représentation des endomorphismes de rang 1 à l'aide d'une opération  $\otimes$  définie à partir d'un produit scalaire. Cet outil sera utilisé dans la démonstration du théorème. Cette partie est plus abstraite que la précédente et demande de la rigueur conceptuelle.
- Dans la troisième partie, on montre deux lemmes qui sont au cœur de la démonstration du théorème : une identité sur les traces des endomorphismes de  $\mathcal{V}$  et une condition suffisante pour avoir un vecteur qui annule tous les endomorphismes de  $\mathcal{V}$ . La difficulté de cette partie provient des notations. On y utilise une généralisation de la formule du binôme de Newton dans le cas de deux endomorphismes qui ne commutent pas.
- La quatrième partie est la plus longue et contient la démonstration du théorème, par récurrence sur la dimension de  $E$ . On montre l'existence d'un vecteur  $x$  qui annule simultanément tous les endomorphismes de  $\mathcal{V}$  et on travaille ensuite avec la restriction de ces endomorphismes sur l'hyperplan  $\text{Vect}(x)^\perp$ . Cette partie utilise tous les résultats des parties précédentes et demande encore une fois d'assimiler de nouvelles notations.

Ce problème étudie principalement des sous-espaces vectoriels de  $\mathcal{L}(E)$  et des applications entre ces sous-espaces, ce qui permet de se confronter à un niveau d'abstraction élevé. Son originalité vient de ce qu'il privilégie les approches spatiale et vectorielle en limitant au maximum l'utilisation des matrices.

Si l'on admet le résultat de la première question, il peut être traité intégralement en fin de première année. Il constitue un bon entraînement sur le programme d'algèbre linéaire de MPSI.

## INDICATIONS

### Partie I

- 1 Montrer que toute matrice  $M$  qui représente  $u$  est trigonalisable dans  $\mathcal{M}_n(\mathbb{C})$ , puis semblable à une matrice de  $T_n^{++}$ .
- 2 Donner un isomorphisme entre  $\mathcal{N}_B$  et  $T_n^{++}(\mathbb{R})$ , puis chercher une base de  $T_n^{++}(\mathbb{R})$  constituée de matrices élémentaires.  
Pour le caractère nilpotent, montrer par récurrence sur  $k \in \llbracket 1; n \rrbracket$  que  $u^k(e_j)$  appartient à  $F_{n-k}$  pour tout  $j \in \llbracket 1; n \rrbracket$ , où  $F_{n-k}$  est l'espace engendré par les  $n - k$  premiers vecteurs de la base  $B$ .
- 3 Pour l'inclusion  $\{\nu(u) \mid u \in \mathcal{N}_B\} \subset \llbracket 1; n \rrbracket$ , utiliser le résultat de la question 4 qui ne dépend pas de ce que l'on démontre ici.  
Pour l'inclusion  $\llbracket 1; n \rrbracket \subset \{\nu(u) \mid u \in \mathcal{N}_B\}$ , considérer l'endomorphisme de  $E$  tel que  $u(e_i) = 0$  si  $i = 1$  ou  $k + 1 \leq i$  et  $u(e_i) = e_{i-1}$  sinon, avec  $k \in \llbracket 1; n \rrbracket$  fixé.
- 4 Appliquer successivement  $u^{p-1}, u^{p-2}, \dots, u$  à une relation linéaire entre les vecteurs de la première famille.  
Pour la deuxième famille, appliquer d'abord  $u^q$  puis poser  $z = u^{p-q}(x)$  et appliquer successivement  $u^{q-1}, \dots, u$ .
- 5 Montrer d'abord que  $\text{Im}(u^{p-1}) \subset \text{Im}(u) \cap \text{Ker}(u)$ . À l'aide de la question 4, prouver ensuite que si  $z \in \text{Im}(u^{p-1}) \setminus \{0\}$ , alors tout vecteur de  $\text{Im}(u) \cap \text{Ker}(u)$  est colinéaire à  $z$ .

### Partie II

- 6 Montrer que  $a \otimes x$  est dans l'ensemble d'arrivée indiqué, pour tout  $a \in E$ . Établir ensuite que  $a \mapsto a \otimes x$  est linéaire et injective. Justifier enfin que l'ensemble d'arrivée a la même dimension que  $E$ .
- 7 Compléter  $x$  en une base de  $E$  et écrire la matrice de  $a \otimes x$  dans cette base.

### Partie III

- 8 Procéder par récurrence pour l'existence et les valeurs de  $f_0^{(k)}$  et  $f_1^{(k)}$ .
- 9 Justifier que  $(u + tv)^p = 0$  et utiliser l'unicité de la relation démontrée à la question 8.
- 10 À l'aide des propriétés de la trace, simplifier  $\text{Tr}(f_1^{(k+1)})$  à partir de sa valeur donnée à la question 8. Montrer ensuite que  $(u + tv)^{k+1}$  est de trace nulle et développer cette expression.
- 11 Établir que  $t \mapsto (a \mid (u + tv)^{p-1}(y))$  est la fonction nulle, puis que c'est une fonction polynomiale en  $t$ . Utiliser l'égalité  $(K(\mathcal{V})^\perp)^\perp = K(\mathcal{V})$ .
- 12 Prouver l'existence par récurrence sur  $k$ . Considérer ensuite le cas  $k = p$ .

### Partie IV

- 13 Montrer que  $v \mapsto v(x)$  et  $u \mapsto \bar{u}$  sont linéaires.
- 14 Appliquer le théorème du rang à  $v \mapsto v(x)$  et  $u \mapsto \bar{u}$  et voir  $\mathcal{V}x, \mathcal{W}, \bar{\mathcal{V}}$  et  $\mathcal{Z}$  comme noyaux ou images de ces applications.
- 15 Justifier que  $\mathcal{Z} \subset \{u \in \mathcal{L}(E) \mid \text{Im}(u) \subset \text{Vect}(x)\}$  et utiliser l'isomorphisme de la question 6.

- 16 Noter  $v = a \otimes x$  et montrer que  $u^k v = a \otimes u^k(x)$ . Appliquer ensuite le lemme A et la question 7.
- 17 Montrer qu'un endomorphisme nilpotent n'a pas de valeur propre non nulle. Obtenir alors que  $\text{Vect}(x)$  et  $\mathcal{V}x$  sont en somme directe.
- 18 Établir la relation par récurrence sur  $k$ .
- 19 Appliquer une partie de l'hypothèse de récurrence à  $\bar{\mathcal{V}}$ , puis utiliser les résultats des questions 14, 15 et 17.
- 20 Combiner les informations obtenues aux questions 14, 15, 17, 18 et 19.
- 21 Appliquer l'hypothèse de récurrence à  $\bar{\mathcal{V}}$  pour obtenir une base  $B'$  de  $H$  et montrer que  $\bar{\mathcal{V}} = \mathcal{N}_{B'}$ . Compléter ensuite  $B'$  par le vecteur  $x$ .
- 22 Avec la question 5, montrer que  $\text{Im}(v^{p-1}) \subset \text{Vect}(v^{k-1}(x))$  où  $k$  est le plus petit entier tel que  $v^k(x) = 0$ . Utiliser ensuite la question 20.
- 23 Justifier que  $t \mapsto (v + tv_0)(x)$  s'annule au plus une fois sur  $\mathbb{R}$  et en déduire que

$$\text{Im}((v + tv_0)^{p-1}) \subset \text{Vect}(x) \oplus \mathcal{V}x$$

Considérer ensuite  $f(t) = (a | (v + tv_0)(y))$  pour  $t \in \mathbb{R}$ ,  $a \in (\text{Vect}(x) \oplus \mathcal{V}x)^\perp$ , et  $y \in E$ .

- 24 Raisonner par l'absurde.

## I. GÉNÉRALITÉS SUR LES ENDOMORPHISMES NILPOTENTS

**1** Soit  $u \in \mathcal{N}(E)$ . Soit  $B$  une base de  $E$  et  $M = \text{Mat}_B(u)$ . Il est possible de voir  $M$  comme un élément de  $\mathcal{M}_n(\mathbb{C})$ . D'après un corollaire du théorème de d'Alembert-Gauss, le polynôme caractéristique  $\chi_M$  de  $M$  est scindé dans  $\mathbb{C}[X]$ . D'après le cours, toute matrice dont le polynôme caractéristique est scindé est trigonalisable. Par suite, la matrice  $M$  est semblable à une matrice triangulaire supérieure  $T \in \mathcal{M}_n(\mathbb{C})$ .

Comme  $M$  et  $T$  sont semblables, les matrices  $M^k$  et  $T^k$  sont semblables pour tout  $k \in \mathbb{N}^*$ . Notons  $(\lambda_1, \dots, \lambda_n)$  les coefficients diagonaux de  $T$ . Comme  $T$  est triangulaire supérieure, les coefficients diagonaux de  $T^k$  sont  $(\lambda_1^k, \dots, \lambda_n^k)$  pour tout  $k \in \mathbb{N}$ . Comme  $u$  est nilpotent, il existe  $p \in \mathbb{N}^*$  tel que  $u^p = 0$ . On a alors  $M^p = 0$  (car  $M^p$  représente  $u^p$ ) donc  $T^p = 0$ , d'où  $\lambda_1^p = \dots = \lambda_n^p = 0$  et  $\lambda_1 = \dots = \lambda_n = 0$ . Ainsi, les coefficients diagonaux de  $T$  sont nuls. Dès lors, pour tout entier naturel  $k$  non nul, comme la trace est un invariant de similitude, il vient

$$\text{Tr}(u^k) = \text{Tr}(M^k) = \text{Tr}(T^k) = \sum_{i=1}^n \lambda_i^k = 0$$

$$\boxed{\forall k \in \mathbb{N}^* \quad \text{Tr}(u^k) = 0}$$

Il est possible de montrer que la réciproque est vraie : si un endomorphisme  $u$  d'un espace vectoriel de dimension finie vérifie  $\text{Tr}(u^k) = 0$  pour tout  $k \in \mathbb{N}^*$ , alors  $u$  est nilpotent. Cette propriété caractérise donc les endomorphismes nilpotents.

Pour démontrer cette réciproque, on raisonne par l'absurde en supposant qu'une matrice  $M$  qui représente  $u$  a au moins une valeur propre complexe non nulle. Notons  $\lambda_1, \dots, \lambda_r$  toutes les valeurs propres complexes non nulles de  $M$  (deux à deux distinctes) et  $n_1, \dots, n_r$  leurs multiplicités respectives. Comme  $M$  est trigonalisable dans  $\mathcal{M}_n(\mathbb{C})$ , il vient

$$\forall k \in \mathbb{N}^* \quad \text{Tr}(M^k) = \text{Tr}(u^k) = \sum_{i=1}^r n_i \lambda_i^k = 0$$

En regardant les  $r$  premières équations ainsi obtenues pour  $k \in \llbracket 1; r \rrbracket$ , on voit que  $(n_1, \dots, n_r)$  est solution d'un système homogène dont le déterminant  $D$  est de type Vandermonde :

$$D = \begin{vmatrix} \lambda_1 & \dots & \dots & \lambda_r \\ \lambda_1^2 & \dots & \dots & \lambda_r^2 \\ \vdots & & & \vdots \\ \lambda_1^r & \dots & \dots & \lambda_r^r \end{vmatrix} = (\lambda_1 \dots \lambda_r) \times \begin{vmatrix} 1 & \dots & \dots & 1 \\ \lambda_1 & \dots & \dots & \lambda_r \\ \vdots & & & \vdots \\ \lambda_1^{r-1} & \dots & \dots & \lambda_r^{r-1} \end{vmatrix}$$

Celui-ci est non nul car les  $\lambda_1, \dots, \lambda_r$  sont tous non nuls et deux à deux distincts. On en déduit que  $(n_1, \dots, n_r) = (0, \dots, 0)$ , ce qui est absurde. Par suite,  $\text{sp}_{\mathbb{C}}(M) = \{0\}$ . Le polynôme caractéristique de  $M$  étant scindé dans  $\mathbb{C}[X]$ , il vaut  $\chi_M = X^n$ , d'où  $M^n = 0$  (Cayley-Hamilton), puis  $u^n = 0$ .

**2** D'après le cours, l'application  $\Phi_B : u \mapsto \text{Mat}_B(u)$  est un isomorphisme d'espaces vectoriels entre  $\mathcal{L}(E)$  et  $\mathcal{M}_n(\mathbb{R})$ . Par définition,  $\mathcal{N}_B = \Phi_B^{-1}(T_n^{++}(\mathbb{R}))$ , donc les ensembles  $\mathcal{N}_B$  et  $T_n^{++}(\mathbb{R})$  sont en bijection par l'isomorphisme  $\Phi_B$ . Montrons que l'ensemble  $T_n^{++}(\mathbb{R})$  est un sous-espace vectoriel de  $\mathcal{M}_n(\mathbb{R})$  de dimension  $n(n-1)/2$ . En notant  $(E_{i,j})_{(i,j) \in \llbracket 1; n \rrbracket^2}$  la base canonique de  $\mathcal{M}_n(\mathbb{R})$ , il vient

$$T_n^{++} = \text{Vect} (E_{i,j} \mid 1 \leq i < j \leq n)$$

## Mines Maths 2 MP 2020 — Corrigé

Ce corrigé est proposé par Rémi Pellerin (ENS Lyon) ; il a été relu par Quentin Guilmant (doctorant en informatique) et Florian Metzger (professeur en CPGE).

---

Ce sujet propose l'étude du nombre de sites visités par une marche aléatoire dans  $\mathbb{Z}^d$ . Il est composé de cinq parties.

- Dans la partie A, on établit des résultats qui seront utilisés dans les parties C et E. Il s'agit de raisonnements combinatoires élémentaires ainsi que d'équivalents de sommes de séries numériques. Les résultats et les méthodes sont proches du cours et doivent être maîtrisés.
- La partie B propose de montrer quelques résultats généraux sur les marches aléatoires dans  $\mathbb{Z}^d$ . Elle est plutôt difficile mais les résultats qui sont utiles pour les parties C et E sont donnés.
- La partie C est consacrée à l'étude du cas  $d = 1$  avec des pas de longueur 1. On y montre qu'il n'y a que lorsque les déplacements sont équiprobables que l'on revient presque sûrement à l'origine en temps fini. On termine en donnant un équivalent du nombre d'entiers visités.
- Dans la partie D, on démontre un résultat sur les suites numériques qui sera exploité à la dernière question.
- Enfin, la dernière partie propose l'étude de la marche aléatoire isotrope (c'est-à-dire lorsque tous les déplacements donnés par la base canonique sont équiprobables) à pas de longueur 1 dans  $\mathbb{Z}^2$ . On y fait la synthèse des résultats précédents. On termine en donnant un bel équivalent du nombre de sites visités ! C'est le théorème d'Erdős-Dvoretzky. Cette partie est un peu plus technique que les autres.

Ce sujet est bien construit et très intéressant. On y montre des résultats puissants et même surprenants ! Il est d'une difficulté et d'une longueur raisonnables comprenant quelques questions assez astucieuses. Il s'agit d'un très bon problème pour s'entraîner en temps limité. Il permet de réviser une grande partie du programme d'analyse, notamment les séries numériques, les séries entières et les probabilités.



## INDICATIONS

### Partie A

- 1 Quel est le coefficient de  $X^n$  dans  $(X + 1)^{2n}$  ?
- 4 Justifier que  $1/\ln$  et  $1/\ln^2$  ne sont pas intégrables sur  $[2; +\infty[$  et utiliser le théorème d'intégration des relations de comparaison.
- 5 On rappelle que

$$\forall \alpha \in \mathbb{R} \quad \forall x \in ]-1; 1[ \quad (1+x)^\alpha = 1 + \sum_{n=1}^{+\infty} \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!} x^n$$

### Partie B

- 7 Remarquer que
- $$\forall (n, k) \in (\mathbb{N}^*)^2 \quad k \geq n \implies (S_n = 0_d) \cap (R = k) = \left( \sum_{i=k+1}^n X_i = 0_d \right) \cap (R = k)$$
- 8 Utiliser un produit de Cauchy.
- 11 Penser à renverser le temps.

### Partie C

- 14 Montrer, en utilisant le résultat de la question 5, que

$$\forall x \in ]-1; 1[ \quad F(x) = \frac{1}{\sqrt{1-4pqx^2}}$$

Pour déterminer la loi de  $R$ , effectuer un développement en série entière de  $G$ .

- 15 On pourra chercher un équivalent du reste  $\sum_{k=n+1}^{+\infty} P(R = 2k)$ .

### Partie D

- 16 Montrer que  $\forall n \in \mathbb{N} \quad a_n B_n \leq 1$
- 18 Prendre  $m_n = \lfloor n \ln(n) \rfloor$  pour  $n \geq 1$  et justifier que

$$B_{m_n} - B_{m_n - n} \underset{n \rightarrow +\infty}{\sim} C \times \sum_{k=m_n - n + 1}^{m_n} \frac{1}{k}$$

### Partie E

- 19 Montrer que la suite de terme général suivant est constante :

$$a_n = \sum_{k=0}^n P(S_k = 0_d) P(R > n - k)$$

- 20 Dans un chemin dans  $\mathbb{Z}^2$  qui termine à son point de départ  $(0, 0)$ , considérer les pas qui sont parallèles aux abscisses et ceux parallèles aux ordonnées.
- 21 Utiliser le résultat de la question 18 afin d'obtenir un équivalent de  $P(R > n)$  et en déduire un équivalent de  $E(N_n)$  en utilisant le théorème de sommation des relations de comparaison ainsi que le résultat de la question 4.

## A. PRÉLIMINAIRES

**1** Procédons comme le suggère l'énoncé et considérons le polynôme  $P = (X + 1)^{2n}$ . La formule du binôme de Newton dans l'anneau commutatif  $\mathbb{R}[X]$  donne

$$P = (X + 1)^{2n} = \sum_{k=0}^{2n} \binom{2n}{k} X^k$$

De ce fait, le coefficient de  $X^n$  dans  $P$  est  $\binom{2n}{n}$ .

Une autre façon de calculer le coefficient de  $X^n$  dans  $P$  est de considérer  $P$  sous la forme du produit de polynômes

$$P = (X + 1)^n (X + 1)^n$$

On a donc

$$\begin{aligned} P &= \left( \sum_{k=0}^n \binom{n}{k} X^k \right) \times \left( \sum_{i=0}^n \binom{n}{i} X^i \right) \\ &= \sum_{k=0}^n \sum_{i=0}^n \binom{n}{k} \binom{n}{i} X^{k+i} \\ P &= \sum_{i,k=0}^n \binom{n}{k} \binom{n}{i} X^{k+i} \end{aligned}$$

Par unicité des coefficients d'un polynôme, le coefficient de  $X^n$  dans  $P$  est donc obtenu pour les indices  $i, k$  tels que  $k + i = n$ . Il vaut ainsi

$$\sum_{k=0}^n \binom{n}{k} \binom{n}{n-k}$$

d'où,

$$\sum_{k=0}^n \binom{n}{k} \binom{n}{n-k} = \binom{2n}{n}$$

Or,

$$\forall n \in \mathbb{N} \quad \forall k \in [0; n] \quad \binom{n}{n-k} = \binom{n}{k}$$

ce qui prouve

$$\boxed{\forall n \in \mathbb{N} \quad \sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}}$$

**2** La formule de Stirling est un équivalent de  $n!$  en  $+\infty$ . D'après le cours,

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

Soit  $n \in \mathbb{N}$ . Si  $a_n \underset{n \rightarrow +\infty}{\sim} b_n$  alors  $a_{2n} \underset{n \rightarrow +\infty}{\sim} b_{2n}$  donc

$$\binom{2n}{n} = \frac{(2n)!}{n! \times n!} \underset{n \rightarrow +\infty}{\sim} \frac{\sqrt{4\pi n} \left(\frac{2n}{e}\right)^{2n}}{2\pi n \left(\frac{n}{e}\right)^{2n}} \underset{n \rightarrow +\infty}{\sim} \frac{4^n}{\sqrt{\pi n}}$$

En posant  $c = 1/\sqrt{\pi} > 0$ , il vient que

$$\boxed{\binom{2n}{n} \underset{n \rightarrow +\infty}{\sim} c \frac{4^n}{\sqrt{n}}}$$

**3** Soit  $\alpha \in ]0; 1[$ . La fonction  $t \mapsto 1/t^\alpha$  est continue et décroissante sur  $[1; +\infty[$ . De ce fait, une comparaison série-intégrale fournit

$$\forall k \geq 1 \quad \int_{k+1}^{k+2} \frac{1}{t^\alpha} dt \leq \frac{1}{(k+1)^\alpha} \leq \int_k^{k+1} \frac{1}{t^\alpha} dt$$

Sommons de 1 à  $n-1$  pour obtenir, avec la relation de Chasles,

$$\int_2^{n+1} \frac{1}{t^\alpha} dt \leq \sum_{k=1}^{n-1} \frac{1}{(k+1)^\alpha} = \sum_{k=2}^n \frac{1}{k^\alpha} \leq \int_1^n \frac{1}{t^\alpha} dt$$

d'où, 
$$\left[ \frac{t^{1-\alpha}}{1-\alpha} \right]_2^{n+1} + 1 \leq \sum_{k=1}^n \frac{1}{k^\alpha} \leq \left[ \frac{t^{1-\alpha}}{1-\alpha} \right]_1^n + 1 \quad (\text{car } \alpha \neq 1)$$

puis 
$$a_n = \frac{(n+1)^{1-\alpha}}{1-\alpha} - \frac{2^{1-\alpha}}{1-\alpha} + 1 \leq \sum_{k=1}^n \frac{1}{k^\alpha} \leq \frac{n^{1-\alpha}}{1-\alpha} - \frac{1}{1-\alpha} + 1 = b_n$$

Comme  $(n+1)^{1-\alpha} \underset{n \rightarrow +\infty}{\sim} n^{1-\alpha} \xrightarrow{n \rightarrow +\infty} +\infty$  car  $1-\alpha > 0$ , il vient que

$$a_n \underset{n \rightarrow +\infty}{\sim} \frac{n^{1-\alpha}}{1-\alpha} \quad \text{et} \quad b_n \underset{n \rightarrow +\infty}{\sim} \frac{n^{1-\alpha}}{1-\alpha}$$

Par encadrement, 
$$\frac{1-\alpha}{n^{1-\alpha}} \sum_{k=1}^n \frac{1}{k^\alpha} \xrightarrow{n \rightarrow +\infty} 1$$

soit 
$$\boxed{\forall \alpha \in ]0; 1[ \quad \sum_{k=1}^n \frac{1}{k^\alpha} \underset{n \rightarrow +\infty}{\sim} \frac{n^{1-\alpha}}{1-\alpha}}$$

On retrouve que  $\sum 1/k^\alpha$  est divergente pour  $\alpha \in ]0; 1[$  ce que l'on sait d'après le critère de Riemann. Au passage, on voit l'équivalence entre la convergence de la série  $\sum 1/k^\alpha$  et l'intégrabilité de  $t \mapsto 1/t^\alpha$  sur  $[1; +\infty[$ .

Notons que les inégalités de comparaison série-intégrale établies plus haut restent valables pour  $\alpha > 1$  car la fonction  $t \mapsto 1/t^\alpha$  est décroissante et continue sur  $[1; +\infty[$ :

$$\forall k \in \mathbb{N}^* \quad \int_{k+1}^{k+2} \frac{1}{t^\alpha} dt \leq \frac{1}{(k+1)^\alpha} \leq \int_k^{k+1} \frac{1}{t^\alpha} dt$$

Puisque  $\alpha > 1$ , le critère de Riemann assure que  $t \mapsto 1/t^\alpha$  est intégrable sur  $[1; +\infty[$  et que la série  $\sum 1/k^\alpha$  converge. Par conséquent, en sommant de  $n$  à  $+\infty$ ,

$$\int_{n+1}^{+\infty} \frac{1}{t^\alpha} dt \leq \sum_{k=n}^{+\infty} \frac{1}{(k+1)^\alpha} \leq \int_n^{+\infty} \frac{1}{t^\alpha} dt$$

d'où 
$$\frac{-(n+1)^{1-\alpha}}{1-\alpha} \leq \sum_{k=n+1}^{+\infty} \frac{1}{k^\alpha} \leq \frac{-n^{1-\alpha}}{1-\alpha}$$

Puis, comme  $(n+1)^{1-\alpha} \underset{n \rightarrow +\infty}{\sim} n^{1-\alpha}$ , il vient, toujours par théorème d'encadrement,

$$\frac{\alpha-1}{n^{\alpha-1}} \sum_{k=n+1}^{+\infty} \frac{1}{k^\alpha} \xrightarrow{n \rightarrow +\infty} 1$$

d'où 
$$\boxed{\sum_{k=n+1}^{+\infty} \frac{1}{k^\alpha} \underset{n \rightarrow +\infty}{\sim} \frac{1}{(\alpha-1)n^{\alpha-1}}}$$

## Mines Informatique MP-PC-PSI 2020 — Corrigé

Ce corrigé est proposé par Julien Dumont (professeur en CPGE) ; il a été relu par William Aufort (professeur en CPGE) et Vincent Puyhaubert (professeur en CPGE).

---

Ce sujet s'intéresse à différents aspects de la fabrication d'images de synthèse, dans le contexte d'une scène contenant des bateaux oscillant à la surface de l'eau dans un canal. L'énoncé comporte trois parties.

- La partie I introduit les notions nécessaires pour la suite du sujet. Elle débute par trois questions assez élémentaires sur les bases de données sans lien avec la suite, puis les volumes des différents objets de la scène sont définis à partir du maillage de leurs surfaces. L'énoncé étudie les fonctions implémentant des opérations géométriques utiles, puis quelques éléments liés à la réduction d'une surface à ses sommets principaux, en enlevant les éventuels doublons dus à de légers écarts en précision.
- La partie suivante prend le prétexte de la génération de vagues pour poser quelques questions sur la mise en forme de données récupérées dans un fichier texte et sur une alternative au stockage brut des données : l'utilisation de matrices creuses.
- La dernière partie cherche à rendre réaliste l'évolution d'un bateau soumis à une vague, grâce à la poussée d'Archimède appliquée à un modèle des parties des bateaux qui sont situées sous l'eau.

Ce sujet assez facile fait appel à de nombreuses notions des deux années de classe préparatoire. La récursivité et le tri fusion font l'objet de quelques questions dans la dernière partie, les premières étant essentiellement concentrées sur les bases de données, la programmation et la complexité.

C'est un bon sujet de révision qui pâtit toutefois de plusieurs imprécisions qui nuisent à la clarté de certaines questions. Mais cela reste un très bon entraînement aux contraintes spécifiques des Mines, à savoir un sujet court durant lequel il faut être efficace, rapide, et ne pas se laisser perturber par un énoncé parfois contradictoire ou une question (Q17) à la très grande limite de ce qui peut être exigé au concours en terme de fonctions.

## INDICATIONS

### Partie I

- 3 Bien identifier dans la lecture des tables à quoi correspond la variable `x` et ce que contient la table issue de la jointure.
- 6 La fonction `ps` n'apparaît pas dans les fonctions fournies, il y a donc eu un renommage.
- 10 Ne pas hésiter à utiliser les fonctions de la bibliothèque fournie.
- 13 Le `return` est manifestement indenté une fois de trop.

### Partie II

- 16 Beaucoup de pièges dans cette question : il faut faire attention aux conversions de types entre les chaînes de caractères du fichier texte et les nombres que l'on veut sauvegarder, en utilisant la fonction de conversion en chaînes de caractères `str`. Par ailleurs, il n'y a ni point-virgule après le dernier caractère de chaque ligne, ni saut de ligne après le dernier caractère de la matrice.
- 17 Pour ouvrir un fichier en écriture, il faut créer un « objet-fichier » en utilisant la commande `open("nomdufichier.txt", "w")`. Il faut aussi le fermer à la fin pour que l'écriture soit effective, grâce à la méthode `close()`.
- 18 Bien détailler les caractères dénombrés selon leur type : entier, flottant, points-virgules, retours à la ligne...
- 19 Faire l'hypothèse que le zéro est codé par un seul caractère.

### Partie III

- 21 La facette est immergée si la hauteur d'eau est au-dessus de la cote du barycentre de la facette.
- 22 Utiliser les deux formules données par l'énoncé.
- 23 Seule la composante de la résultante selon  $\vec{e}_z$  est demandée.
- 24 L'énoncé n'impose rien de particulier comme complexité pour la fusion, une version itérative est toutefois plus simple. Celle-ci peut, par exemple, consister à remplir au fur et à mesure une liste de longueur `len(L1) + len(L2)` avec les éléments pertinents.
- 25 L'énoncé impose une version récursive de l'algorithme. Utiliser la méthode « diviser pour régner » : on coupe la liste en deux, on trie les deux morceaux, puis on fusionne les résultats.
- 26 Attention à inclure la facette médiane.
- 27 L'énoncé est erroné car il n'est pas possible avec des listes (mais cela le serait avec des tableaux NumPy) de s'en sortir avec l'amorce de code des deux lignes à compléter. Réécrire donc totalement ces deux lignes.

## I. CRÉATION D'UN OBJET DANS LA SCÈNE

1 L'information sur le nombre de maillages est contenue intégralement dans la table `maillages_bateau` et s'obtient grâce à la requête

```
SELECT COUNT(*) FROM maillages_bateau
```

Il y a deux solutions alternatives, en changeant l'argument du `COUNT` : `COUNT(id)` ou `COUNT(nom)`. Cela était en effet possible car chaque enregistrement possède une valeur pour ces deux champs.

2 Cette fois-ci, une jointure est nécessaire afin de récupérer les informations recherchées qui sont présentes dans les tables `maillages_bateau` et `faces`. Cette jointure s'effectue en notant la concordance entre les colonnes `maillage` et `id`. Notons enfin que toutes les colonnes ont des noms différents, il n'est donc pas nécessaire de faire des alias. On peut alors proposer par exemple la requête

```
SELECT numero
FROM maillages_bateau JOIN faces ON id = maillage
WHERE nom = "gouvernail"
```

3 L'étude du `FROM ... JOIN ... ON` permet tout d'abord de voir que les trois tables sont nécessaires à cette requête, et que la table issue de cette jointure permet de regrouper les coordonnées de l'ensemble des points du maillage. L'analyse du `WHERE` permet de voir que l'on retient le maillage `coque`. Enfin, le `SELECT` renvoie l'écart entre les deux valeurs extrêmes des abscisses. Autrement dit,

La requête renvoie l'amplitude du maillage de la coque selon la direction  $\vec{e}_x$ .

4 Il faut renvoyer la deuxième coordonnée du premier élément de la première liste, en faisant attention à l'indexation démarrant à 0 en Python et en respectant l'ordre d'extraction. L'expression est ainsi

```
maillage_tetra[0][0][1]
```

5 Comme expliqué à la question précédente, `maillage_tetra[1]` se réfère à la face dont les sommets sont successivement `[0.,0.,0.]`, `[0.,1.,0.]` et `[1.,0.,0.]`, autrement dit les sommets respectifs A, D et B. Ainsi,

L'expression `maillage_tetra[1]` correspond à la face  $S_4$ .

6 La seule fonction appelée dans le code proposé est le produit scalaire, mais sous un autre nom que celui donné dans la documentation. Il y a donc eu un renommage. La première ligne du code est par conséquent

```
from operations_vectorielles import prod_scalaire as ps
```

7 Le code proposé renvoie la racine carrée de la somme des carrés des composantes du vecteur pris en argument d'entrée, par conséquent

La fonction `mystere1` calcule la norme euclidienne d'un vecteur.

**8** Il s'agit d'écrire une fonction qui renvoie un vecteur dont les composantes sont celles du vecteur en argument d'entrée multipliées chacune par une même constante. Le sujet précise de plus en introduction que les vecteurs considérés sont des listes de trois flottants. Par suite, on peut proposer

```
def multiplie_scalaire(a, V):
    return [a*V[0], a*V[1], a*V[2]]
```

On peut proposer le code suivant, plus général car le vecteur d'entrée admet un nombre arbitraire de composantes :

```
def multiplie_scalaire(a, V):
    return [a*v for v in V]
```

**9** Ici, on cherche le barycentre de trois points constituant une face, il s'agit donc de calculer la moyenne de chaque coordonnée des sommets en question. Une première boucle, indiquée dans le code, balaie chaque sommet tandis qu'une seconde permet d'en récupérer les coordonnées (abscisse, ordonnée et cote) et d'ajouter les termes permettant de calculer la moyenne. Comme deux lignes seulement sont proposées, on ne peut faire la somme des coordonnées avant de diviser par trois, il faut plutôt ajouter à chaque étape la contribution de la nouvelle coordonnée balayée divisée par trois. Finalement, le code complet que l'on peut proposer est

```
def barycentre(F):
    G = [0,0,0]
    for i in range(3):
        for j in range(3):
            G[j] = G[j] + F[i][j]/3
    return G
```

Même si c'est souvent assez évident, le sujet est radin en définition des arguments des différentes fonctions. Ici par exemple, il faut comprendre que  $F$  est une facette.

Le code suivant utilise les fonctions proposées par l'énoncé, il serait tout aussi valable s'il n'y avait pas le squelette du code dans le sujet.

```
def barycentre(F):
    somme = addition(F[0], addition(F[1], F[2]))
    return multiplie_scalaire(1/3, somme)
```

**10** On utilise les fonctions de la bibliothèque proposée par l'énoncé, d'abord pour calculer les vecteurs  $\vec{AB}$  et  $\vec{AC}$ , puis pour effectuer leur produit vectoriel. L'usage de la fonction `mystere1` étudiée à la question 7 permet de terminer le code.

```
def normale(F):
    AB = soustraction(F[1], F[0])
    AC = soustraction(F[2], F[0])
    PV = prod_vectoriel(AB, AC)
    norme = mystere1(PV)
    return multiplie_scalaire(1/norme, PV)
```

Notons que l'hypothèse d'aire non nulle — sous réserve que les calculs flottants soient exacts — assure que la quantité `norme` est également non nulle et donc que la division est licite.

## Mines Informatique optionnelle MP 2020 — Corrigé

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

---

Cette épreuve est composée de deux parties indépendantes.

- La première aborde plusieurs façons de compter les occurrences de chaque lettre dans un texte. On cherche des solutions efficaces pour des textes écrits avec un alphabet de grande taille (l'alphabet Unicode, qui comporte de l'ordre du million de caractères différents). Cette partie propose trois implémentations différentes. Elle est assez simple.
- La partie II s'intéresse aux arbres de recherche optimaux, dans un contexte d'encodage de texte. Encoder un texte nécessite d'associer un code binaire à chaque lettre de l'alphabet. L'ensemble des lettres et leur code est stocké sous la forme d'un arbre binaire de recherche. Pour trouver le code d'une lettre, il faut parcourir l'arbre, ce qui est d'autant plus coûteux que la lettre est profonde. Le coût total d'encodage d'un texte dépend donc du nombre d'occurrences et de la profondeur de chaque lettre. On veut alors trouver la forme de l'arbre qui minimise ce coût.

Dans la première sous-partie, plutôt élémentaire, il faut écrire l'algorithme d'encodage et déterminer sa complexité. La deuxième sous-partie, plus difficile, donne une méthode de programmation dynamique qui permet de déterminer un arbre de code optimal en  $O(\lambda^3)$  opérations élémentaires où  $\lambda$  est le cardinal de l'alphabet. La dernière sous-partie démontre un résultat qui permet d'améliorer cette méthode pour construire l'arbre en seulement  $O(\lambda^2)$  opérations élémentaires. Elle est aussi technique que difficile.

Ce sujet aborde des problématiques classiques, importantes et très intéressantes en algorithmique. Hélas, il souffre de deux défauts : la faible qualité de son écriture et la grande hétérogénéité dans la difficulté des questions. Si la majeure partie des questions de la première partie ne devraient pas poser de problème, les questions de la dernière partie nécessitent une finesse et une technicité (magnifique) qu'il est difficile d'acquérir après deux années de classe préparatoire. Par ailleurs, les questions 10 et 11 sont hors-programme.



## INDICATIONS

### Partie I

- 1 Justifier que `fonction1` renvoie un tableau dont la  $i$ -ième case contient  $|t|_{\sigma_i}$ .
- 2 Parcourir le tableau renvoyé par `fonction1` à la recherche des cases contenant des valeurs non nulles.
- 6 Noter qu'un courriel classique contient de 1 000 à 10 000 caractères, et utilise une centaine de lettres différentes.
- 7 Parcourir la liste pour voir si elle contient un couple dont le premier élément est  $u$  auquel cas on incrémente le deuxième. Dans le cas contraire, il suffit de lui ajouter un nouveau couple.
- 8 On peut utiliser le même principe que pour `fonction1` en créant un tableau de  $m$  listes dans le cas d'arrêt et en utilisant `incrimente_repartition` dans le cas général.
- 9 Il suffit de sommer les longueurs des listes dans la table renvoyée par la question 8.
- 10 L'espérance conditionnelle n'est pas au programme des classes préparatoires, cette question n'est pas traitable.
- 11 La notion de complexité en moyenne n'est pas non plus au programme, cette question n'est pas non plus traitable.
- 13 Justifier qu'il est plus judicieux de prendre  $m$  de l'ordre de la valence de  $t$ .
- 14 Vérifier que  $I$  induit une injection sur  $\{\sigma_0, \dots, \sigma_{v-1}\}$  et préciser son image.
- 15 Le sens indirect découle de la question précédente. Pour le sens direct, utiliser (ii) puis appliquer  $A$ .
- 16 Utiliser la question précédente. La terminaison et la correction de cette fonction est triviale, ne pas s'étonner.
- 17 Il s'agit juste de tester si  $u$  est présent grâce à la fonction précédente, puis de mettre à jour les tableaux représentant  $A$  et  $I$  en conséquence.
- 18 On peut encore une fois garder le même schéma de rédaction que `fonction1`, en utilisant cette fois `make_creux` pour le cas d'arrêt et `incrimente_tableaucreux` dans le cas général.
- 21 Les machines actuelles sont capables d'effectuer de l'ordre de  $10^{12}$  opérations élémentaires par seconde.

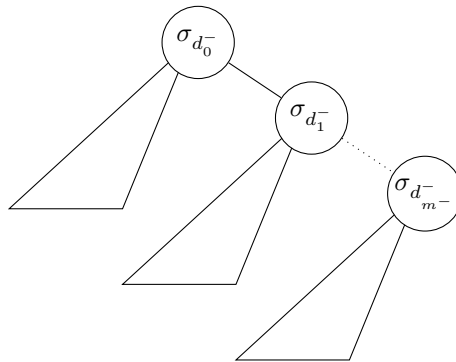
### Partie II

- 22 Implémenter une fonction qui encode une lettre à l'aide d'un parcours récursif de l'arbre représentant le code, comme pour un arbre binaire de recherche.
- 23 Le coût d'une recherche dans un arbre est proportionnel à la profondeur de la valeur cherchée (lorsqu'elle s'y trouve). Une concaténation de deux listes a un coût linéaire en la taille de celle de gauche.
- 24 Dans la définition de la profondeur, séparer la somme en trois en distinguant le nœud contenant la racine et les sous-arbres gauche et droit.
- 25 Démontrer l'égalité

$$\Pi_{u,v} = \sum_{k=u}^v f(\sigma_k) + \min_{\rho \in \llbracket u; v \rrbracket} (\Pi_{u,\rho-1} + \Pi_{\rho+1,v})$$

en utilisant les techniques classiques de programmation dynamique.

- 26 Appliquer les méthodes de programmation dynamique pour calculer les quantités  $(\Pi_{u,v})_{0 \leq u \leq v \leq \lambda-1}$  les unes après les autres. On utilisera la question précédente pour calculer  $(\Pi_{u,u+d})_{0 \leq u \leq \lambda-d-1}$  pour  $d$  croissant de 1 à  $\lambda-1$ .
- 27 Chaque application de la formule de la question 25 nécessite une recherche de minimum qui s'effectue en  $O(v-u)$  opérations élémentaires. Remarquer que  $\mathcal{E}(\lambda-2)$  permet de ramener ce coût à un  $O(r_{u,v+1} - r_{u,v})$  et utiliser un télescopage.
- 28 On rappelle que dans un arbre de code, les lettres apparaissant dans le fils droit d'un nœud sont strictement plus grandes que celle contenue dans le nœud.
- 29 Remarquer que  $\mathcal{A}$  et  $\mathcal{A}'$  ont la même profondeur puis, à partir d'un arbre  $\mathcal{B}'$  de code représentant  $c_{u,v+1}$  quelconque, construire un arbre  $\mathcal{B}$  représentant  $c_{u,v}$  de profondeur inférieure à  $\mathcal{B}'$ .
- 30 Remarquer que si  $\mathcal{A}$  est un arbre de code quelconque représentant  $c_{u,v+1}$ , la formule donnant la profondeur définit une fonction affine de  $f(\sigma_{v+1})$  et déterminer sa pente. Puis, utiliser le fait que  $\Pi_{u,v+1}$  est défini comme le minimum de toutes les profondeurs des arbres de code représentant  $c_{u,v+1}$ .
- 31 Une suite strictement croissante d'entiers majorée est nécessairement finie. Pour la deuxième partie, remarquer que par définition des suites  $(d_k^-)$  et  $(d_k^+)$ , on peut construire un arbre de code optimal de la forme



Faire ensuite le lien entre  $m^-$  et  $m^+$  et la pente de la fonction affine par morceaux  $\pi_{u,v+1}$  à l'aide de la question précédente. Enfin, justifier que, en chaque point où la pente de  $\pi_{u,v+1}$  change, celle-ci ne peut que décroître.

- 32 Cette question est à l'origine de beaucoup de souffrances. Il est conseillé de ne pas chercher à la résoudre, et de se contenter de lire le « corrigé ».
- 33 Raisonner par récurrence sur  $n$ . Pour l'hérédité, l'une des égalités se déduit en faisant le bilan des questions précédentes, afin d'en conclure que  $f(\sigma_{v+1}) \mapsto r_{u,v+1}$  est une fonction croissante sur  $\mathbb{R}_+$  qui prend en 0 une valeur supérieure ou égale à  $r_{u,v}$ .

On utilisera également un argument de symétrie pour l'autre inégalité.

Les fonctions fournies par l'énoncé existent déjà en OCaml (avec des noms quasiment identiques). Dans ce cas de figure, il est toutefois impératif d'utiliser les outils du sujet. Pour ceux qui souhaiteraient tester leur code, il suffit de définir les 4 fonctions données, par exemple de la manière suivante :

```
let list_length = List.length ;;
let list_append l1 l2 = l1@l2 ;;
let array_make = Array.make ;;
let array_length = Array.length ;;
```

Pour simplifier la rédaction, on identifiera directement un texte (qui est ici une suite d'entiers) et le mot associé (qui est une suite de lettres). En particulier, on utilisera directement la notation  $|t|_\alpha$  lorsque  $t$  est un texte et  $\alpha$  une lettre.

## I. FONCTION DE DISTRIBUTION DES LETTRES D'UN TEXTE

**1** La fonction `fonction1` possède un unique argument du type `texte`. Lorsque ce dernier est une liste vide, elle renvoie un tableau d'entiers de  $\lambda = 1114112$  cases initialisées à 0. On en déduit donc que

La fonction `fonction1` est de type `texte -> int array`.

Rappelons que, contrairement à un langage comme Python, les fonctions OCaml doivent renvoyer des résultats ayant le même type. Par conséquent, il suffit d'identifier ce type dans n'importe quel cas d'arrêt de la fonction pour en déduire le type du résultat dans tous les cas.

Justifions maintenant que `fonction1` prend en argument un texte  $\mathbf{t}$  et renvoie un tableau de la forme  $[|t|_{\sigma_0}, |t|_{\sigma_1}, \dots, |t|_{\sigma_{\lambda-1}}]$ , c'est-à-dire un tableau dont la  $i$ -ième case contient la fréquence de la  $i$ -ième lettre de  $\Sigma$  dans  $\mathbf{t}$ . Comme l'impose l'énoncé, on procède par récurrence à l'aide du prédicat suivant :

$\mathcal{P}(k)$  : « Si  $\mathbf{t}$  est un texte de longueur  $k$ , alors `fonction1 t` renvoie un tableau de  $\lambda$  cases dont la  $i$ -ième case contient  $|t|_{\sigma_i}$ . »

- $\mathcal{P}(0)$  est trivialement vrai car si  $k = 0$ , cela signifie que  $\mathbf{t}$  est un texte vide, auquel cas la fonction renvoie un tableau de taille  $\lambda$  dont toutes les cases valent 0.
- $\mathcal{P}(k) \implies \mathcal{P}(k+1)$  : Supposons le résultat vrai pour tout texte de longueur  $k$ . Soit  $\mathbf{t}$  un texte de longueur  $k+1$ . Nécessairement,  $\mathbf{t}$  n'est pas vide car  $k+1$  n'est pas nul. La fonction `fonction1` appelle alors le deuxième cas du filtrage qui lance un appel récursif sur la queue `tprime` de  $\mathbf{t}$  qui est de longueur  $k$  ce qui permet d'appliquer l'hypothèse de récurrence. Le résultat `theta` de cet appel récursif est un tableau de  $\lambda$  cases dont la  $i$ -ième case contient la valeur  $|t|_{\sigma_i}$ . Cet appel est suivi d'une instruction qui incrémente de 1 la case de `theta` correspondant à l'indice de la tête (c'est-à-dire la première lettre) du texte. On en déduit donc clairement qu'à la fin de cette instruction, le tableau `theta` qui est renvoyé est de la forme souhaitée. Ainsi,  $\mathcal{P}(k+1)$  est vraie.
- Conclusion : Pour tout  $k \geq 0$ ,  $\mathcal{P}(k)$  est vrai.

Notamment,

La fonction `fonction1` prend en argument un texte  $\mathbf{t}$  et renvoie un tableau dont la  $i$ -ième case contient la fréquence de la lettre  $\sigma_i$  dans le mot associé à  $\mathbf{t}$ .

## X/ENS Maths A MP 2020 — Corrigé

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

Ce sujet propose l'étude de valeurs propres de matrices symétriques à coefficients rationnels. En particulier, on étudie quels nombres réels peuvent en être valeurs propres. C'est le cas des racines carrées de nombres rationnels positifs, et plus généralement de tout nombre dit *totalelement réel*, c'est-à-dire racine d'un polynôme rationnel scindé sur  $\mathbb{R}$ . D'autres nombres, comme  $\sqrt[3]{2}$ , ne peuvent pas être valeurs propres de telles matrices.

- Dans la partie I, on montre que les nombres algébriques de degré 2 (racines de polynômes à coefficients entiers de degré 2) sont valeurs propres de matrices symétriques à coefficients dans  $\mathbb{Q}$ , de même que les réels  $\cos(2\pi/n)$  avec  $n \in \mathbb{N}^*$ . On montre que ce n'est pas le cas de  $\sqrt[3]{2}$ . Un résultat fondamental pour le problème y est établi : il existe, pour tous rationnels strictement positifs  $q_1, \dots, q_d$ , des matrices symétriques rationnelles  $M_1, \dots, M_d$  commutant deux à deux et telles que  $M_i^2$  soit la matrice d'une homothétie de rapport  $q_i$  pour tout  $i \in \llbracket 1; d \rrbracket$ .
- Dans la partie II, on montre principalement le critère suivant : un polynôme unitaire de degré  $d$  est à coefficients rationnels si, et seulement si, les sommes de Newton de ses racines complexes  $\mu_1, \dots, \mu_d$

$$N_n = \sum_{k=1}^d \mu_k^n$$

sont dans  $\mathbb{Q}$  pour tout  $n \in \mathbb{N}^*$ .

- Dans la partie III, on étudie les nombres totalement réels, ainsi que les nombres totalement positifs qui sont eux les nombres racines d'un polynôme rationnel scindé sur  $\mathbb{R}$  dont toutes les racines sont dans  $\mathbb{R}_+$ . En particulier, ces ensembles sont stables par addition, produit, et inversion d'élément non nul. Le dernier résultat prouvé est l'équivalence entre réalité totale d'un nombre et positivité totale de son carré.
- La quatrième et dernière partie vise à montrer que tout nombre totalement réel est valeur propre d'une matrice symétrique à coefficients rationnels. Elle introduit une forme linéaire sur les nombres totalement réels, strictement positive sur les nombres totalement positifs. Cela permet de construire explicitement la matrice cherchée en fonction du nombre totalement réel.

Le sujet est de difficulté progressive et les questions restent relativement abordables jusqu'à la partie III. La dernière partie demande plus de technique et de subtilité, ainsi qu'une connaissance pointue du cours et du recul pour l'adapter à la situation étudiée. Seule la dernière question est très difficile et demande une synthèse de tout le problème ainsi que beaucoup d'initiative. De nombreux thèmes au programme sont abordés : algèbre matricielle, produit scalaire, réduction, polynômes, nombres complexes, arithmétique, séries entières, ce qui en fait un joli et complet problème de concours.

## INDICATIONS

### Partie I

- 1 Utiliser l'expression du polynôme caractéristique d'une matrice de taille 2.
- 2.a Remarquer que  $(X - \sqrt{3})$  divise  $\chi_M$  et que la trace et le déterminant sont des fonctions polynomiales en les coefficients d'une matrice.
- 2.b Raisonner par disjonction de cas suivant la classe de  $n$  modulo 3.
- 2.c Raisonner par l'absurde et examiner les congruences modulo 3.
- 2.d Utiliser encore l'expression de  $\chi_M$  en fonction de  $\text{Tr } M$  et  $\det M$ .
- 3.a Chercher  $B$  par blocs de taille  $n$  dont un des blocs est  $A$ .
- 3.b Procéder par récurrence sur  $d$  en utilisant la question 3.a.
- 3.c Écrire  $q_i = p_i/p_0$  et utiliser des matrices  $M_i$  données par la question 3.b vérifiant  $M_i^2 = p_i I_n$  pour tout  $i$ .
- 4.a Adapter la classique preuve par l'absurde de l'irrationalité de  $\sqrt{2}$ .
- 5 Trouver une matrice à coefficients entiers annulée par  $X^n - 1$ .

### Partie II

- 6 Prouver d'abord que  $Q$  est bien un polynôme.
- 7 Noter que 0 n'est pas racine de  $P$  et utiliser le développement en série entière de  $u \mapsto 1/(1-u)$ .
- 8.a Penser au lien entre la somme d'une série et ses coefficients. La formule de Leibniz pourra être utile.
- 8.b Remarquer que  $Q'(x) = f(x)Q(x)$  pour tout  $x$  différent d'une racine de  $Q$ , et raisonner par récurrence en utilisant l'expression des coefficients d'une série entière en fonction des dérivées de sa somme.
- 8.c Attention au cas où  $P$  admet pour racine 0.
- 9 Appliquer le résultat de la question 8.c après avoir explicité les sommes associées aux deux polynômes donnés, en se rappelant que

$$\sum_{i,j} a_i b_j = \sum_i a_i \times \sum_j b_j$$

### Partie III

- 10 Utiliser le théorème spectral.
- 11.a Il s'agit de montrer que, si  $\alpha_1, \beta_1$  sont racines de polynômes scindés sur  $\mathbb{R}$  à coefficients rationnels, alors 1,  $\alpha_1 + \beta_1$ ,  $\alpha_1 \beta_1$ ,  $-\alpha_1$  et  $1/\alpha_1$  (pour  $\alpha_1 \neq 0$ ) le sont également.
- 11.b Le résultat de la question 11.a s'adapte très facilement.
- 12 Le sens réciproque est facile. Pour le sens direct, on pourra fixer un polynôme scindé sur  $\mathbb{R}$  à coefficients rationnels dont  $x$  est racine et en déduire un polynôme dont les racines sont les carrées des précédentes.

### Partie III

- 13.a Remarquer que  $B$  est une forme bilinéaire symétrique sur  $\mathbb{Q}^d$  et utiliser sa forme développée. Noter également que  $\lambda x + \mu y \in \mathcal{R}$  pour tous  $x, y \in \mathcal{R}$  et tous  $\lambda, \mu \in \mathbb{Q}$ , de sorte que l'on peut appliquer les propriétés de définition de  $t$ . Pour la non nullité, on pourra se souvenir que  $d$  est le plus petit degré d'un polynôme unitaire à coefficients rationnels annulant  $z$ .
- 13.b Montrer que le noyau de  $S$ , vue comme matrice de  $M_d(\mathbb{Q})$ , est réduit au vecteur colonne nul.

14 Par définition,  $B$  est une forme bilinéaire symétrique sur  $\mathbb{R}^d$ . La question 13.b montre que  $B$  est un produit scalaire sur  $\mathbb{Q}^d$ . Prolonger les propriétés par densité de  $\mathbb{Q}$  dans  $\mathbb{R}$ . On pourra montrer que  $\text{sp } S \subset \mathbb{R}_+^*$  et utiliser le théorème spectral.

- 15.a Adapter le procédé de Gram-Schmidt pour orthogonaliser, pour  $B$ , la base canonique de  $\mathbb{R}^d$ , sans normer les vecteurs afin de garder des composantes rationnelles.
- 15.b Utiliser la matrice de passage de la base canonique à celle construite à la question 15.a. On rappelle que pour toutes matrices  $A, B$  de  $M_d(\mathbb{R})$  :

$$A = B \iff \forall (X, Y) \in M_{d,1}(\mathbb{R})^2 \quad X^T A Y = X^T B Y$$

16 Il s'agit du classique déterminant d'une matrice compagnon d'un polynôme. Montrer que  $\chi_M = Z$  en développant par rapport à la première ligne pour obtenir une formule de récurrence.

- 17.a Montrer que  $(SM)_{i,j} = t(z^{i+j+1})$  pour tous  $i, j$ . On pourra utiliser la relation

$$z^d = \sum_{k=0}^{d-1} a_k z^k$$

- 17.b Remarquer que  $S = R^T R$ .

18 Considérer des matrices  $M_i$  comme à la question 3.c. : elles joueront un peu le rôle de  $\text{Diag}(\sqrt{q_1}, \dots, \sqrt{q_d})$ . Poser  $M' = \text{Diag}(M_1, \dots, M_d) \in M_{nd}(\mathbb{Q})$ . Adapter le raisonnement de la question 17.b en cherchant  $F$  inversible pour qu'en définissant

$$R' = M' F \text{Diag}(P, \dots, P)$$

la matrice  $A = R' \text{Diag}(M, \dots, M)(R')^{-1}$  réponde à la question. Remarquer enfin que  $\chi_A$  s'exprime simplement en fonction de  $\chi_M$ .

## PREMIÈRE PARTIE

**1** Si  $A \in M_2(\mathbb{Q})$ , alors  $\chi_A = X^2 - (\text{Tr } A)X + \det A$ . Il suffit donc d'exhiber une matrice de  $S_2(\mathbb{Q})$  dont le polynôme caractéristique est  $X^2 - 2$ , c'est-à-dire dont la trace est 0 et le déterminant  $-2$ . Par exemple,

La matrice  $M = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \in S_2(\mathbb{Q})$  admet  $\sqrt{2}$  comme valeur propre.

**2.a** Soit  $M \in S_2(\mathbb{Q})$  admettant  $\sqrt{3}$  pour valeur propre. D'après le cours,  $X - \sqrt{3}$  divise (dans  $\mathbb{R}[X]$ ) son polynôme caractéristique

$$\chi_M = X^2 - (\text{Tr } M)X + \det M$$

qui est de degré 2 et unitaire. Fixons  $\alpha \in \mathbb{R}$  tel que  $\chi_M = (X - \sqrt{3})(X - \alpha)$ . Il en résulte que

$$\alpha\sqrt{3} = \det M \quad \text{et} \quad \alpha + \sqrt{3} = \text{Tr } M$$

Or,  $\text{Tr } M \in \mathbb{Q}$  et  $\det M \in \mathbb{Q}$  car  $M \in M_2(\mathbb{Q})$ , d'après les formules polynomiales de la trace et du déterminant en fonction des coefficients. Par suite,

$$\sqrt{3}(\text{Tr } M) = \sqrt{3}(\alpha + \sqrt{3}) = \det M + 3$$

Si  $\text{Tr } M \neq 0$  :

$$\sqrt{3} = (\det M + 3) / \text{Tr } M \in \mathbb{Q}$$

On obtient une contradiction, donc  $\text{Tr } M = 0$  et alors  $\alpha = -\sqrt{3}$ . Finalement,

$$\chi_M = (X - \sqrt{3})(X - \alpha) = X^2 - 3$$

**2.b** Soit  $n \in \mathbb{Z}$ . Distinguons suivant les congruences modulo 3 :

- si  $n \equiv 0 \pmod{3}$ , alors  $n^2 \equiv 0^2 \equiv 0 \pmod{3}$  ;
- si  $n \equiv 1 \pmod{3}$ , alors  $n^2 \equiv 1^2 \equiv 1 \pmod{3}$  ;
- si  $n \equiv 2 \pmod{3}$ , alors  $n^2 \equiv 2^2 \equiv 4 \equiv 1 \pmod{3}$ .

Ainsi,  $\forall n \in \mathbb{Z} \quad n^2 \equiv 0 \pmod{3} \quad \text{ou} \quad n^2 \equiv 1 \pmod{3}$

**2.c** Soient  $x, y$  et  $z$  des entiers avec  $\text{pgcd}(x, y, z) = 1$  et tels que  $x^2 + y^2 = 3z^2$ . Il en découle alors que  $x^2 + y^2 \equiv 0 \pmod{3}$ . Or d'après l'étude réalisée à la question 2.b, le cas  $x^2 + y^2 \equiv 0 \pmod{3}$  ne se produit que pour  $x \equiv 0 \pmod{3}$  et  $y \equiv 0 \pmod{3}$ . Fixons alors  $x' \in \mathbb{Z}$  et  $y' \in \mathbb{Z}$  tels que  $x = 3x'$  et  $y = 3y'$ . Il vient

$$3z^2 = 9(x')^2 + 9(y')^2 \quad \text{donc} \quad z^2 = 3(x')^2 + 3(y')^2 \equiv 0 \pmod{3}$$

Ainsi,  $z^2 \equiv 0 \pmod{3}$ , et donc  $z \equiv 0 \pmod{3}$  d'après la disjonction de cas faite à la question 2.b, donc  $x, y$  et  $z$  sont tous multiples de 3, en contradiction avec l'hypothèse  $\text{pgcd}(x, y, z) = 1$ .

Pour tous entiers  $x, y, z$  tels que  $\text{pgcd}(x, y, z) = 1$ , on a  $x^2 + y^2 \neq 3z^2$ .

## X Maths B MP 2020 — Corrigé

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

Le thème de cette épreuve d'analyse est le comportement asymptotique. On considère d'abord la probabilité d'un événement défini par des variables aléatoires dont le nombre tend vers l'infini. On cherche ensuite des équivalents simples d'intégrales dépendant d'un paramètre tendant vers  $+\infty$ .

- Dans la partie I, on étudie le comportement de  $P[S_n \geq t]$  quand  $n \rightarrow +\infty$  où  $S_n$  est la moyenne de  $n$  variables aléatoires  $(X_1, X_2, \dots, X_n)$  indépendantes et de même loi dite *de Rademacher*. Cette partie est la plus abordable si on maîtrise le cours de probabilités de première année. Seule la question 8b contient une difficulté de rédaction pour la limite.
- Dans la partie II, on établit un équivalent, lorsque  $t \rightarrow +\infty$ , de l'intégrale

$$\int_a^b e^{t f(x)} dx$$

où  $f$  est une fonction définie sur  $[a; b]$  suffisamment régulière. Ce résultat est appelé *méthode du col de Laplace*. Le sujet nous propose, en guise d'application, de redémontrer la fameuse formule de Stirling.

- Dans la troisième partie, proche de la précédente, la fonction exponentielle est remplacée par des fonctions sinusoïdales. On y établit un développement asymptotique, quand  $a \rightarrow +\infty$ , de

$$\int_0^a \sin(t^2) dt$$

mais aussi un équivalent, quand  $t \rightarrow +\infty$ , de

$$\int_0^1 g(x) \sin(t f(x)) dx$$

avec  $f$  et  $g$  des fonctions satisfaisant encore une fois certaines propriétés de régularité.

Ce sujet démontre des résultats classiques, comme la semi-convergence de l'intégrale, dite *de Fresnel*,

$$\int_0^{+\infty} \sin(x^2) dx$$

ainsi que l'écriture de  $n!$  sous forme d'une intégrale. Néanmoins, dans les parties II et III, des questions difficiles requièrent d'avoir plusieurs bonnes idées et initiatives avant d'aboutir.

Les outils de probabilités utilisés sont majoritairement issus du cours de première année. Le chapitre *Intégration sur un intervalle* joue un rôle important pour bien justifier les calculs. Pour les deux dernières parties, il faut également maîtriser les développements limités, la rédaction des limites avec les quantificateurs et le calcul d'intégrales.



## INDICATIONS

## Première partie

- 1 Utiliser l'inégalité de Markov.
- 2 Remarquer que  $f(X_i)$  suit une loi de Bernoulli de paramètre  $1/2$  pour une certaine fonction affine  $f$ . Montrer que si  $B$  est une loi binomiale de paramètres  $(n, 1/2)$ , alors  $P[B \geq n/2] \geq 1/2$ .
- 3 Appliquer le résultat de la question 1 à  $\exp(\lambda n S_n)$ . Justifier que les variables aléatoires  $(\exp(\lambda X_1), \dots, \exp(\lambda X_n))$  sont mutuellement indépendantes.
- 5a Calculer l'espérance comme un produit d'espérances grâce au résultat fourni par l'énoncé. Montrer ensuite que  $E[(X_1 - m(\lambda)) \exp(\lambda X_1)] = 0$ .
- 5b Développer  $(S_n - m(\lambda))^2$  comme une double somme sur  $(i, j) \in \llbracket 1; n \rrbracket^2$ . Traiter à part les cas  $i = j$  et  $i \neq j$ .
- 6 Montrer que  $I_n(\lambda, \varepsilon) \exp(\lambda n(S_n - m(\lambda) - \varepsilon)) \leq I_n(\lambda, \varepsilon)$  et utiliser la croissance de l'espérance.
- 7 Idem que la question 6 avec  $(1 - I_n(\lambda, \varepsilon))\varepsilon^2 \leq (S_n - m(\lambda))^2$ .
- 8a Minorer  $P[S_n \geq m(\lambda) - \varepsilon]$  avec  $E[I_n(\lambda, \varepsilon)D_n(\lambda)]$  et utiliser les questions précédentes.
- 8b Fixer  $\varepsilon > 0$  tel que  $\varepsilon + t \in [0; 1[$ , utiliser la question 4 pour trouver un  $\lambda$  tel que  $m(\lambda) = \varepsilon + t$ . Utiliser les résultats des questions 3 et 8a, pour avoir un encadrement de  $\log P[S_n \geq t]$ . Comme  $u_n(\varepsilon) \xrightarrow{n \rightarrow \infty} 0$ , il existe un rang à partir duquel  $|u_n(\varepsilon)| \leq \varepsilon$ . Majorer alors  $\lambda$  de façon indépendante de  $\varepsilon$ , pour  $\varepsilon \leq \varepsilon_0$  avec un  $\varepsilon_0$  bien choisi.

## Deuxième partie

- 9 Utiliser un développement limité de  $f$  à l'ordre 2.
- 10 Découper, en trois morceaux, l'intégrale

$$\int_a^b e^{t f(x)} dx$$

et démontrer que deux d'entre eux sont négligeables devant  $\int_{x_0 - \delta}^{x_0 + \delta} e^{t f(x)} dx$ .

- 11 Appliquer l'inégalité de Taylor à l'ordre 2 à  $f$ . Majorer le reste en fonction de  $(x - x_0)^2$ . Obtenir un encadrement de  $e^{t f(x)}$ . Faire un changement de variable pour se ramener à une intégrale de  $X \mapsto e^{-X^2}$ .
- 12b Mettre l'intégrande trouvé à la question 12a sous la forme  $x \mapsto e^{n f(x)}$  où  $f$  est une certaine fonction, quitte à faire un changement de variable.

## Troisième partie

- 13 Trouver une partie de  $\mathbb{R}_+$ , la plus grande possible, sur laquelle  $x \mapsto \sin(x^2)$  est plus grande que  $1/2$ , puis comparer les intégrales.
- 15 Intégrer par parties en intégrant  $x \mapsto 2x \sin(x^2)$ . On pourra aussi passer par l'exponentielle complexe.
- 16 Intégrer par parties successivement jusqu'à obtenir une intégrale en  $O(a^{-5})$ .
- 17 Intégrer par parties en intégrant  $x \mapsto t \sin(t f(x))$ .
- 19 Procéder au changement de variable  $u = h(x)/h(1)$ . Appliquer le résultat de la question 17. Procéder alors à un nouveau changement de variable pour utiliser le résultat de la question 16.

## PREMIÈRE PARTIE

**1** Soient  $\lambda > 0$  et  $t \in \mathbb{R}$ . Comme  $\lambda > 0$ , la fonction  $t \mapsto e^{\lambda t}$  est strictement croissante sur  $\mathbb{R}$ . Ainsi, les événements  $(Z \geq t)$  et  $(\exp(\lambda Z) \geq \exp(\lambda t))$  sont égaux. De plus,  $\exp(\lambda Z)$  est une variable aléatoire positive admettant une espérance finie. L'inégalité de Markov s'applique alors à  $\exp(\lambda Z)$  :

$$P[Z \geq t] = P[\exp(\lambda Z) \geq \exp(\lambda t)] \leq \frac{1}{\exp(\lambda t)} E[\exp(\lambda Z)]$$

En conclusion,  $\forall t \in \mathbb{R} \quad \forall \lambda > 0 \quad P[Z \geq t] \leq \frac{1}{\exp(\lambda t)} E[\exp(\lambda Z)]$

**2** Montrons d'abord un résultat sur les variables aléatoires images de variables aléatoires mutuellement indépendantes qui n'est pas au programme de MP mais qui sera utile plusieurs fois dans ce sujet. Soient  $(U_1, U_2, \dots, U_n)$  des variables mutuellement indépendantes et, pour tout  $i \in \llbracket 1; n \rrbracket$ ,  $f_i$  une fonction réelle définie sur  $U_i(\Omega)$ . Pour tout  $i \in \llbracket 1; n \rrbracket$ , posons  $V_i = f_i(U_i)$ . Les variables  $(V_1, V_2, \dots, V_n)$  sont alors mutuellement indépendantes. En effet, soit  $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ ,

$$\begin{aligned} P\left[\bigcap_{i=1}^n (V_i = x_i)\right] &= P\left[\bigcap_{i=1}^n (f_i(U_i) = x_i)\right] \\ &= P\left[\bigcap_{i=1}^n (U_i \in f_i^{-1}(\{x_i\}))\right] \end{aligned}$$

par indépendance mutuelle :

$$\begin{aligned} &= \prod_{i=1}^n P[U_i \in f_i^{-1}(\{x_i\})] \\ &= \prod_{i=1}^n P[f(U_i) = x_i] \end{aligned}$$

$$P\left[\bigcap_{i=1}^n (V_i = x_i)\right] = \prod_{i=1}^n P[V_i = x_i]$$

Soit  $i \in \llbracket 1; n \rrbracket$ . Posons  $Y_i = X_i/2 + 1/2$ . Observons que

$$(Y_i = 1) = (X_i = 1) \quad \text{et} \quad (Y_i = 0) = (X_i = -1)$$

Dès lors,  $Y_i$  suit une loi de Bernoulli de paramètre  $1/2$ . De plus, pour tout entier  $i \in \llbracket 1; n \rrbracket$ ,  $Y_i$  est une variable aléatoire image de  $X_i$ . D'après le résultat démontré en début de question, comme  $(X_1, X_2, \dots, X_n)$  sont des variables aléatoires mutuellement indépendantes,  $(Y_1, Y_2, \dots, Y_n)$  le sont aussi. Il en découle que

$$B = \sum_{i=1}^n Y_i$$

suit une loi binomiale de paramètres  $(n, 1/2)$ . Connaissant la loi de  $B$ , on peut affirmer

$$P[B \geq n/2] = \sum_{n/2 \leq k \leq n} \binom{n}{k} \frac{1}{2^n}$$

tandis que

$$\begin{aligned} P[B \leq n/2] &= \sum_{0 \leq p \leq n/2} \binom{n}{p} \frac{1}{2^n} = \sum_{n/2 \leq k \leq n} \binom{n}{n-k} \frac{1}{2^n} \quad (k = n-p) \\ &= \sum_{n/2 \leq k \leq n} \binom{n}{k} \frac{1}{2^n} \quad \text{car} \quad \binom{n}{n-k} = \binom{n}{k} \end{aligned}$$

$$P[B \leq n/2] = P[B \geq n/2]$$

Il s'ensuit que

$$1 = \mathbb{P}[(B \leq n/2) \cup (B \geq n/2)] \leq \mathbb{P}[B \leq n/2] + \mathbb{P}[B \geq n/2] = 2\mathbb{P}[B \geq n/2]$$

Dès lors,  $\mathbb{P}[B \geq n/2] \geq 1/2$ . Comme  $B = nS_n/2 + n/2$ , on a les égalités d'évènements

$$(B \geq n/2) = (nS_n/2 + n/2 \geq n/2) = (nS_n \geq 0) = (S_n \geq 0)$$

Deux évènements égaux ayant même probabilité, on en conclut que

$$\boxed{\mathbb{P}[S_n \geq 0] \geq \frac{1}{2}}$$

**3** Soient  $t \in \mathbb{R}$  et  $\lambda > 0$ . En tant que variable aléatoire réelle prenant un nombre fini de valeurs,  $\exp(\lambda n S_n)$  est d'espérance finie. D'après la question 1, et en utilisant que l'exponentielle est un morphisme de  $(\mathbb{R}, +)$  vers  $(\mathbb{R}_+, \times)$  et  $n > 0$ ,

$$\mathbb{P}[S_n \geq t] = \mathbb{P}[n S_n \geq t n] \leq e^{-\lambda t n} \mathbb{E}[\exp(\lambda n S_n)] = e^{-\lambda t n} \mathbb{E}\left[\prod_{i=1}^n \exp(\lambda X_i)\right]$$

Comme  $X_1, \dots, X_n$  sont des variables aléatoires mutuellement indépendantes, d'après le résultat démontré au début de la question 2, les variables aléatoires

$$(\exp(\lambda X_1), \exp(\lambda X_2), \dots, \exp(\lambda X_n))$$

sont mutuellement indépendantes. De plus, ces variables aléatoires prenant un nombre fini de valeurs, elles admettent une espérance. En utilisant le résultat fourni par l'énoncé, il vient

$$\mathbb{P}[S_n \geq t] \leq e^{-\lambda t n} \mathbb{E}\left[\prod_{i=1}^n \exp(\lambda X_i)\right] = e^{-\lambda t n} \prod_{i=1}^n \mathbb{E}[\exp(\lambda X_i)]$$

Soit  $i \in \llbracket 1; n \rrbracket$ . À l'aide de la formule de transfert appliquée à  $\exp(\lambda X_i) = f(X_i)$  avec  $f: x \mapsto e^{\lambda x}$ ,

$$\begin{aligned} \mathbb{E}[\exp(\lambda X_i)] &= f(1)\mathbb{P}[X_i = 1] + f(-1)\mathbb{P}[X_i = -1] \\ &= \frac{\exp(\lambda) + \exp(-\lambda)}{2} \\ \mathbb{E}[\exp(\lambda X_i)] &= \operatorname{ch}(\lambda) \end{aligned} \tag{1}$$

En reportant ce résultat dans la majoration précédente de  $\mathbb{P}[S_n \geq t]$ , on trouve

$$\mathbb{P}[S_n \geq t] \leq \exp(-\lambda t n) \operatorname{ch}(\lambda)^n$$

Notons que  $\operatorname{ch}(\lambda) \geq 1 > 0$ . En outre, le logarithme est une fonction strictement croissante de  $\mathbb{R}_+$  vers  $\mathbb{R} \cup \{-\infty\}$ , il vient

$$\log(\mathbb{P}[S_n \geq t]) \leq \log(\exp(-\lambda t n) \operatorname{ch}(\lambda)^n) = -\lambda t n + n \log(\operatorname{ch}(\lambda))$$

On rappelle que  $\log(0) = -\infty$  par convention du sujet. Par conséquent, on n'a pas besoin de séparer les cas  $\mathbb{P}[S_n \geq t] = 0$  et  $\mathbb{P}[S_n \geq t] > 0$ .

En divisant par  $n \geq 1$ , on obtient

$$\frac{1}{n} \log(\mathbb{P}[S_n \geq t]) \leq \psi(\lambda) - \lambda t$$

Et ce pour tout  $\lambda > 0$ . De plus, pour  $\lambda = 0$ , puisque  $\mathbb{P}[S_n \geq t] \leq 1$ ,

$$\frac{1}{n} \log(\mathbb{P}[S_n \geq t]) \leq 0 = \psi(0) - 0t$$