

PSI
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

Philippe BOUAFIA
professeur agrégé en école d'ingénieurs

Céline CHEVALIER
enseignant-chercheur à l'université

Julien DUMONT
professeur en CPGE

Quentin GUILMANT
ENS Lyon

Benjamin MONMEGE
enseignant-chercheur à l'université

Matthias MORENO RAY
professeur en CPGE

Angèle NICLAS
ENS Lyon

Bertrand WIEL
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
CONCOURS COMMUN INP			
Mathématiques	Transformée de Laplace du sinus cardinal et matrices de Kac. <i>intégration, algèbre linéaire, transformée de Laplace, probabilités</i>	17	25
Informatique	Détection d'obstacles par un sonar de sous-marin. <i>programmation Python, listes, bibliothèque NumPy, récursivité</i>	43	63
CENTRALE-SUPÉLEC			
Mathématiques 1	Étude de la gestion des erreurs dans un processus industriel. <i>probabilités, matrices, réduction, séries numériques, séries de fonctions, informatique pour tous</i>	75	81
Mathématiques 2	Les fonctions de Lambert. <i>analyse réelle, probabilités, séries entières, suites de fonctions</i>	99	103
Informatique	Photomosaique. <i>programmation Python, bibliothèque NumPy, représentation des nombres, listes, bases de données</i>	125	133

MINES-PONTS

Mathématiques 1	Espaces vectoriels d'endomorphismes nilpotents. <i>algèbre linéaire, algèbre bilinéaire, endomorphismes, espaces euclidiens</i>	152	158
Mathématiques 2	Caractérisation et exponentielle des matrices normales. <i>matrices, normes, algèbre euclidienne, diagonalisation, séries numériques</i>	180	185
Informatique	Images de vagues et de structures. <i>programmation Python, bases de données, complexité, tris, récursivité</i>	207	220

POLYTECHNIQUE-ENS

Mathématiques	Régularité de fonctions, fonctions concaves et semi-concaves. <i>fonctions lipschitziennes, convexité, calcul différentiel à plusieurs variables</i>	231	238
---------------	---	-----	-----

FORMULAIRES

Développements limités usuels en 0	265
Développements en série entière usuels	266
Dérivées usuelles	267
Primitives usuelles	268
Trigonométrie	270

SESSION 2020



PSI1M

ÉPREUVE SPÉCIFIQUE - FILIÈRE PSI**MATHÉMATIQUES****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.

Les calculatrices sont interdites**Le sujet est composé de deux problèmes indépendants.**

PROBLÈME 1

Autour de la fonction sinus cardinal

Objectifs

Dans ce problème, on détermine dans la **Partie I** la valeur de la transformée de Laplace de la fonction sinus cardinal. On utilise ensuite dans la **Partie II** une variante de la formule de Viète pour exprimer la transformée de Laplace de la **Partie I** comme limite d'une suite d'intégrales.

Partie I - Transformée de Laplace de la fonction sinus cardinal

Pour $x > 0$, on note :

$$F(x) = \int_0^{+\infty} \frac{\sin(t)}{t} e^{-tx} dt, \quad G(x) = \int_0^{+\infty} e^{-tx} \sin(t) dt \quad \text{et} \quad H(x) = \int_0^{+\infty} e^{-tx} \cos(t) dt.$$

- Q1.** Montrer que : $\forall t \in \mathbb{R}^+, |\sin(t)| \leq t$.
- Q2.** Montrer que les fonctions F , G et H sont bien définies sur $]0, +\infty[$.
- Q3.** Montrer que $\lim_{x \rightarrow +\infty} F(x) = 0$.
- Q4.** Montrer que F est de classe C^1 sur $]0, +\infty[$ et exprimer F' à l'aide de la fonction G .
- Q5.** Trouver une expression simple pour G et pour H . On pourra calculer $H(x) + iG(x)$.
En déduire, pour $\alpha \in]0, +\infty[$, la valeur de $\int_0^{+\infty} e^{-t\alpha} \cos(\alpha t) dt$.
- Q6.** En déduire une expression simple pour F . Que vaut $F(1)$?

Partie II - Autour de la formule de Viète

- Q7.** Montrer que pour tout $t > 0$ et pour tout $n \in \mathbb{N}^*$:

$$\prod_{k=1}^n \cos\left(\frac{t}{2^k}\right) = \frac{\sin(t)}{2^n \sin(t/2^n)}.$$

- Q8.** Montrer que pour tout $t > 0$ et pour tout $n \in \mathbb{N}^*$:

$$\prod_{k=1}^n \cos\left(\frac{t}{2^k}\right) = \frac{1}{2^{n-1}} \sum_{k=1}^{2^{n-1}} \cos\left(\frac{2k-1}{2^n} t\right).$$

On pourra raisonner par récurrence et utiliser l'identité :

$$\cos(a) \cos(b) = \frac{1}{2} (\cos(a+b) + \cos(a-b)).$$

Q9. En déduire que pour tout $t > 0$:

$$\frac{\sin(t)}{t} = \lim_{n \rightarrow +\infty} \frac{1}{2^{n-1}} \sum_{k=1}^{2^{n-1}} \cos\left(\frac{2k-1}{2^n}t\right).$$

Q10. Montrer que pour tout $x > 0$:

$$F(x) = \lim_{n \rightarrow +\infty} \frac{1}{2^{n-1}} \sum_{k=1}^{2^{n-1}} \int_0^{+\infty} \cos\left(\frac{2k-1}{2^n}t\right) e^{-tx} dt.$$

On pourra introduire, pour tout $n \in \mathbb{N}^*$, la fonction $f_n :]0, +\infty[\rightarrow \mathbb{R}$ définie par :

$$\forall t \in]0, +\infty[, f_n(t) = \frac{1}{2^{n-1}} \sum_{k=1}^{2^{n-1}} \cos\left(\frac{2k-1}{2^n}t\right) e^{-tx}.$$

Q11. En déduire que :

$$\frac{\pi}{4} = \lim_{n \rightarrow +\infty} 2^{n+1} \sum_{k=1}^{2^{n-1}} \frac{1}{(2k-1)^2 + 2^{2n}}.$$

L'objet des trois questions suivantes est de redémontrer le résultat précédent de façon plus élémentaire.

Q12. Déterminer :

$$\lim_{n \rightarrow +\infty} 2^{n+1} \sum_{k=0}^{2^{n-1}} \frac{1}{4k^2 + 2^{2n}}$$

en écrivant cette quantité à l'aide d'une somme de Riemann.

Q13. Soit $n \in \mathbb{N}^*$. Montrer que pour tout $k \in \llbracket 0, 2^{n-1} \rrbracket$:

$$\left| \frac{1}{4k^2 + 2^{2n}} - \frac{1}{(2k-1)^2 + 2^{2n}} \right| \leq \frac{4 \times 2^{n-1} + 1}{1 + 2^{2n}} \times \frac{1}{4k^2 + 2^{2n}}.$$

Q14. En déduire que :

$$\lim_{n \rightarrow +\infty} 2^{n+1} \sum_{k=0}^{2^{n-1}} \left(\frac{1}{4k^2 + 2^{2n}} - \frac{1}{(2k-1)^2 + 2^{2n}} \right) = 0$$

et retrouver le résultat de la question **Q11**.

CCINP Maths PSI 2020 — Corrigé

Ce corrigé est proposé par Quentin Guilmant (ENS Lyon) ; il a été relu par Benjamin Monmege (enseignant-chercheur à l'université) et Gilbert Monna (professeur honoraire en CPGE).

Le sujet comporte deux problèmes indépendants mêlant analyse réelle, algèbre linéaire et probabilités.

- Le premier problème est composé deux parties. Il établit une stratégie d'approximation de $\pi/4$ de deux façons.
 - La première partie propose de donner une forme simple à la transformée de Laplace de la fonction sinus cardinal.
 - La seconde reprend cette transformée pour l'exprimer sous forme d'une limite de fonctions de façon à pouvoir approximer correctement $\pi/4$. Elle propose ensuite une méthode plus directe pour démontrer à nouveau le résultat.
- Le second problème, plus long, comporte quatre parties. Il traite des matrices de Kac, qui ont plusieurs applications, notamment en physique des particules. La dernière partie présente ainsi un modèle stochastique qui représente certains phénomènes de mécanique statistique.
 - La première partie est une mise en jambes. Elle s'intéresse au cas simple de la dimension 3. Les résultats étudiés dans la suite sont cependant déjà présents.
 - La deuxième introduit un endomorphisme classique, la dérivation, sur un espace de fonctions construites à partir des fonctions cosinus et sinus.
 - Les résultats de la partie 2 sont ensuite utilisés pour étudier les matrices de Kac de dimension quelconque.
 - La dernière partie utilise les matrices de Kac pour modéliser une expérience aléatoire, les urnes d'Ehrenfest.

Le sujet parcourt à peu près l'ensemble des grands thèmes au programme de PSI. Les candidats étaient amenés à utiliser les théorèmes d'intégration, à établir des majorations et à mobiliser leurs connaissances sur les endomorphismes diagonalisables et les probabilités. Ce sujet permet donc de réviser l'ensemble du programme. Bien qu'il s'agisse souvent d'appliquer le cours, quelques questions opposent plus de résistance.

INDICATIONS

Problème 1

- 1 Majorer \sin et $-\sin$ pour conclure.
- 2 Comparer à une intégrale de Riemann.
- 3 Comparer $|\sin t / te^{-tx}|$ à une fonction intégrable connue.
- 7 Utiliser la formule $\sin(2a) = 2 \sin(a) \cos(a)$.
- 9 Utiliser l'équivalent en 0 de la fonction sinus.
- 10 À l'aide de la question 8, montrer que

$$\left| \frac{1}{2^{n-1}} \sum_{k=1}^{2^{n-1}} \cos\left(\frac{2k-1}{2^n}t\right) \right| \leq 1$$

puis utiliser le théorème de convergence dominée.

- 11 Utiliser le changement de variable affine $u = ((2k-1)/2^n)t$.
- 12 Faire apparaître la somme de Riemann sur $[0; 1]$ avec pas de 2^{n-1} pour la fonction G . Utiliser la question 4.
- 13 Utiliser le fait que $k \in \llbracket 0; 2^{n-1} \rrbracket$.

Problème 2

- 19 La matrice D est un produit de matrices d'opérations élémentaires. On peut répondre sans développer tout le produit matriciel.
- 21 Procéder par récurrence et montrer la propriété

$$\mathcal{P}(k) : \quad \forall i \in \llbracket 0; k \rrbracket \quad \lambda_{n-i} = 0$$

lorsque l'on suppose que $\sum_{k=0}^n \lambda_k f_k = 0$.

- 23 Écrire $g_k(x) = e^{ikx} e^{-i(n-k)x}$.
- 29 Se rappeler que $\det(MN) = \det(M) \det(N)$.
- 30 Utiliser la question 29 sous la forme $B_n = -iD_n^{-1}A_nD_n$.
- 36 Introduire une nouvelle variable pour chaque boule indiquant si cette boule a été placée dans l'urne U_1 .
- 38 Montrer que le vecteur de la loi de probabilité Z d'une loi π' satisfaisant la même propriété que π est un vecteur propre de A_n pour la valeur propre n .

Problème 1. AUTOUR DE LA FONCTION SINUS CARDINAL

1 Considérons la fonction $u: \begin{cases} \mathbb{R} \rightarrow \mathbb{R} \\ t \mapsto t - \sin t \end{cases}$

Cette fonction est dérivable sur \mathbb{R}_+ et pour tout $t \in \mathbb{R}_+$, on a

$$u'(t) = 1 - \cos t \geq 0$$

La fonction u est donc croissante. Or $u(0) = 0$ donc pour $t \in \mathbb{R}_+$, on a $u(t) \geq 0$, puis

$$t \geq \sin t$$

Soit maintenant la fonction $v: \begin{cases} \mathbb{R} \rightarrow \mathbb{R} \\ t \mapsto t + \sin t \end{cases}$

dérivable sur \mathbb{R} telle que pour tout $t \in \mathbb{R}_+$, on a

$$v'(t) = 1 + \cos t \geq 0$$

La fonction v est donc croissante. Or $v(0) = 0$ donc pour $t \in \mathbb{R}_+$, on a $v(t) \geq 0$, puis

$$t \geq -\sin t$$

En réunissant les deux inégalité obtenues, on obtient

$$\boxed{\forall t \in \mathbb{R}_+ \quad t \geq |\sin t|}$$

2 Notons $f(x, t), g(x, t)$ et $h(x, t)$ les intégrandes respectifs de $F(x), G(x)$ et $H(x)$. Par composition et produit, ces fonctions sont continues sur $]0; +\infty[$ en chacune de leurs variables. De plus, d'après la question 1, pour tous $x, t > 0$

$$|f(x, t)| \leq e^{-tx}$$

On a aussi $|g(x, t)| \leq e^{-tx}$ et $|h(x, t)| \leq e^{-tx}$

Ainsi, pour tout $x \in \mathbb{R}_+^*$, on a

$$f(x, t) = \underset{t \rightarrow +\infty}{o} \left(\frac{1}{t^2} \right) \quad g(x, t) = \underset{t \rightarrow +\infty}{o} \left(\frac{1}{t^2} \right) \quad h(x, t) = \underset{t \rightarrow +\infty}{o} \left(\frac{1}{t^2} \right)$$

Par comparaison à une intégrale de Riemann,

$$\boxed{\text{Les fonctions } F, G \text{ et } H \text{ sont bien définies sur } \mathbb{R}_+^* .}$$

3 Fixons $x > 0$. Pour tout $t > 0$, d'après la question 1, on a

$$\left| \frac{\sin t}{t} e^{-tx} \right| \leq e^{-tx}$$

La fonction $t \mapsto e^{-tx}$ étant intégrable sur \mathbb{R}_+ par le même argument qu'en question 2, par croissance de l'intégrale, on a

$$|F(x)| \leq \int_0^{+\infty} \left| \frac{\sin t}{t} e^{-tx} \right| dt \leq \int_0^{+\infty} e^{-tx} dt = \left[-\frac{1}{x} e^{-tx} \right]_0^{+\infty} = \frac{1}{x}$$

La valeur absolue étant positive, on en déduit par encadrement que

$$\boxed{\lim_{x \rightarrow +\infty} F(x) = 0}$$

4 La fonction

$$f: \begin{cases} (\mathbb{R}_+^*)^2 \longrightarrow \mathbb{R} \\ (x, t) \longmapsto \frac{\sin t}{t} e^{-tx} \end{cases}$$

est de classe \mathcal{C}^1 sur son domaine de définition en tant que multiplication de fonctions de classe \mathcal{C}^1 . Pour tous $x, t > 0$

$$\frac{\partial f}{\partial x}(x, t) = -e^{-tx} \sin t = -g(x, t)$$

Cette fonction est continue par rapport à chacune de ses variables. De plus, on a

$$\forall (x, t) \in (\mathbb{R}_+^*)^2 \quad \left| \frac{\partial f}{\partial x}(x, t) \right| \leq e^{-tx}$$

Par croissance de l'exponentielle, on obtient donc l'hypothèse de domination

$$\forall a > 0 \quad \forall x > a \quad \forall t > 0 \quad \left| \frac{\partial f}{\partial x}(x, t) \right| \leq e^{-ta}$$

Toutes les hypothèses du théorème de dérivation sont donc vérifiées pour tout $a \in \mathbb{R}_+^*$ et tout $x \in]a; +\infty[$. Ainsi, pour tout $a \in \mathbb{R}_+^*$, la fonction F est de classe \mathcal{C}^1 sur $]a; +\infty[$ et

$$\forall x \in]a; +\infty[\quad F'(x) = \int_0^{+\infty} -e^{-tx} \sin t = -G(x)$$

Le résultat étant vrai pour tout $a \in \mathbb{R}_+^*$,

La fonction F est de classe \mathcal{C}^1 sur \mathbb{R}_+^* et $F' = -G$.

5 On a pour tout $x > 0$,

$$H(x) + iG(x) = \int_0^{+\infty} e^{-tx} (\cos t + i \sin t) dt = \int_0^{+\infty} e^{-t(x-i)} dt$$

Le module de l'intégrande est e^{-tx} qui est intégrable sur \mathbb{R}_+^* . L'intégrande est donc intégrable sur \mathbb{R}_+^* et une de ses primitives est la fonction

$$t \longmapsto -\frac{1}{x-i} e^{-t(x-i)}$$

qui est bien définie car $x \in \mathbb{R}_+^*$ et en particulier $x \neq i$. On en déduit

$$H(x) + iG(x) = \left[\frac{1}{i-x} e^{-t(x-i)} \right]_0^{+\infty} = \frac{1}{x-i} = \frac{x+i}{x^2+1}$$

Ainsi, en prenant les parties imaginaire et réelle on obtient que

$$\forall x \in \mathbb{R}_+^* \quad G(x) = \frac{1}{x^2+1} \quad \text{et} \quad H(x) = \frac{x}{x^2+1}$$

Soit $\alpha > 0$. En faisant un changement de variable linéaire de coefficient directeur non nul $u = \alpha t$, on obtient que pour tout $x > 0$

$$\int_0^{+\infty} e^{-tx} \cos(\alpha t) dt = \frac{1}{\alpha} \int_0^{+\infty} e^{-\frac{x}{\alpha} u} \cos u du = \frac{1}{\alpha} H\left(\frac{x}{\alpha}\right)$$

Par conséquent

$$\forall (\alpha, x) \in (\mathbb{R}_+^*)^2 \quad \int_0^{+\infty} e^{-tx} \cos(\alpha t) dt = \frac{x}{x^2 + \alpha^2}$$

CCINP Informatique PSI 2020 — Corrigé

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

Ce sujet met en œuvre des méthodes de traitement du signal et d'intelligence artificielle pour résoudre un problème de détection d'obstacles par un sonar de sous-marin. Un sonar émet des ondes qui sont réfléchies par les obstacles ; il détecte cette réflexion et l'interprète pour déterminer la dangerosité de l'obstacle.

- La première partie présente le problème et ne comporte aucune question.
- La deuxième est consacrée au traitement des signaux émis et reçus par le sonar. Elle détaille l'algorithme de transformée de Fourier discrète utilisé pour isoler les composantes fréquentielles principales du signal réfléchi par l'obstacle. La plupart des questions demandent de la programmation nécessitant la manipulation de listes et de tableaux NumPy.
- Enfin, la troisième partie propose deux algorithmes d'apprentissage pour classer des données en deux groupes correspondant à deux types d'obstacles. La première méthode utilise un arbre de décision construit à partir d'une base de données d'entraînement où le type d'obstacle est connu. La notion d'arbre n'est pas au programme de l'informatique commune, mais elle est relativement bien expliquée dans l'énoncé. La seconde méthode, dite des forêts aléatoires, utilise plusieurs arbres de décision de taille moins importante afin d'améliorer les performances de la classification.

Comme en 2019, le sujet aborde un algorithme d'intelligence artificielle appliqué à un problème original. Il comporte beaucoup de questions de programmation et de compréhension de code ou de documentation. Quelques questions font appel à la récursivité. Les autres grandes parties du programme (ingénierie numérique, requêtes SQL, tris) ne sont pas abordées. L'énoncé contient beaucoup d'informations, et la longueur de certaines questions (notamment la question 3) peut s'avérer déstabilisante. Néanmoins, ce sujet est d'une difficulté tout à fait raisonnable.

INDICATIONS

Partie II

- 1 Utiliser la fonction `arange` ou la fonction `array`, toutes deux rappelées en annexe.
- 3 Identifier pour chaque argument optionnel de la fonction `scipy.signal.stft` si la valeur par défaut correspond aux exigences de l'énoncé.
- 4 Observer l'intervalle de temps correspondant à la portion représentée sur les spectrogrammes et comparer avec la représentation du signal de la figure 3.
- 7 La fonction ci-dessous normalise les données contenues dans le vecteur `P`:

$$x \mapsto \frac{x - \min(P)}{\max(P) - \min(P)}$$

- 8 Les méthodes `split` et `strip` sont expliquées en annexe. La fonction `float` permet de convertir un objet Python en flottant.
- 9 Un flottant en double précision est codé sur 64 bits.

Partie III

- 12 Générer un à un des nombres aléatoires ; ne les ajouter à la liste à renvoyer que s'ils y sont absents.
- 14 L'entier `n_donnees` est égal à la somme des longueurs des groupes. Utiliser la variable `p` comme un compteur.
- 15 Compter le nombre de données correspondant à chaque valeur.
- 16 Identifier les différents cas de figure où l'on arrête de construire récursivement l'arbre, puis choisir à l'aide des instructions de la fonction `construit` où placer le test correspondant à chacun.
- 18 Identifier les instructions à l'arrivée sur une feuille dans le parcours de l'arbre. Pour les autres, poursuivre récursivement le parcours.
- 19 Commencer par construire une liste contenant la forêt aléatoire, puis identifier pour chaque donnée de `data_test` la classe majoritaire issue des différentes prédictions, à l'aide d'un code ressemblant à celui de la question 15.

II. ANALYSE DES DONNÉES

1 La fonction `arange` de NumPy rappelée dans l'annexe permet de construire des séquences de valeurs uniformément réparties. Cette fonction prend en arguments la première valeur, la dernière valeur (exclue), ainsi que le pas entre deux valeurs consécutives. Dans notre situation, ces valeurs sont respectivement 0, T_f et T_f/N . On obtient donc l'instruction suivante :

```
temps = arange(0, Tf, Tf/N)
```

Comme le pas T_f/N est un flottant, il est possible que des erreurs d'arrondi perturbent le résultat de la fonction `arange`, en particulier à proximité de la valeur exclue. Par exemple, l'appel `arange(0.2, 0.8, 0.1)` renvoie un tableau contenant la valeur 0.8 (plutôt une valeur flottante légèrement supérieure, mais affichée 0.8) qui devrait être exclue.

Une autre possibilité serait de construire une liste Python contenant les valeurs demandées, par exemple en compréhension, puis de convertir cette liste en tableau NumPy à l'aide de la fonction `array` rappelée dans l'annexe. Attention à ne pas oublier cette étape car l'énoncé attendait bien un tableau NumPy. En suivant cette idée, on pourrait obtenir l'instruction

```
temps = array([i*Tf/N for i in range(N)])
```

Cette méthode assure que le tableau renvoyé aura la bonne longueur.

2 Pour plus de lisibilité, on définit à l'intérieur de la fonction `chirp` une fonction `e` calculant $e(t)$. Cette fonction prend comme unique argument la valeur de t , les autres paramètres intervenant dans l'expression de $e(t)$ étant tous des arguments de la fonction `chirp`. Une fois cette fonction définie, on construit le vecteur correspondant au signal émis à l'aide d'une liste en compréhension et de la fonction `array`.

```
def chirp(temps, f0, Deltafe, T, E0):
    def e(t):
        if t <= T:
            fe = f0 + t*Deltafe/T
            return E0 * sin(2*pi*fe*t)
        else:
            return 0
    return array([e(t) for t in temps])
```

Certaines opérations peuvent être appliquées directement sur les tableaux NumPy sans passer par des boucles `for` ; on parle d'opérations vectorialisées. Ces opérations sont en général plus rapides que leurs homologues utilisant des boucles Python. Par exemple, on peut dans certains cas appliquer une fonction `f` directement sur un tableau NumPy `T` avec la syntaxe `f(T)`. Ce n'est hélas pas possible avec l'implémentation proposée à cause de l'instruction conditionnelle présente dans la définition de `e(t)`. On pourrait contourner ce problème en multipliant le résultat à renvoyer par le booléen `(t <= T)`, interprété comme valant 0 (échec du test) ou 1 (succès) dans le produit.

3 D'après la description fournie dans l'énoncé, la fonction `scipy.signal.stft` prend pour argument obligatoire le signal temporel à traiter contenu dans `s`, ainsi que quatre arguments optionnels :

- la fréquence d'échantillonnage `fs` utilisée dans le calcul des transformées de Fourier locales. On l'obtient à partir de la période d'échantillonnage $t_e = T_f/N$ donnée dans l'énoncé par la formule $f_s = 1/t_e$;
- le nom `window` de la fenêtre à utiliser, ici une fenêtre de type Hamming ;
- la longueur `nperseg` de chaque fenêtre, fixée dans l'énoncé à n valeurs ($n//2$ avant la valeur τ choisie et $n//2$ après) ;
- `noverlap`, le nombre de valeurs qui chevauchent deux fenêtres consécutives. Il est fixé par défaut à $n//2$, correspondant à un chevauchement de 50 %, en accord avec les contraintes de l'énoncé.

En conclusion, les seuls paramètres à expliciter dans l'appel de la fonction `stft` sont `s`, `fs`, `window` et `nperseg`. Les trois derniers étant des arguments optionnels, on les fournit ici avec leurs noms respectifs. On obtient finalement l'instruction suivante :

```
f, t, S = scipy.signal.stft(s, fs=N/Tf, window='hamming', nperseg=n)
```

Si les arguments optionnels sont fournis avec leurs noms, l'ordre dans lequel ils apparaissent dans la fonction n'a aucune importance. Dans le cas contraire, l'ordre d'apparition est décisif. Cela donnerait :

```
f, t, S = scipy.signal.stft(s, N/Tf, 'hamming', n)
```

En reprenant la notation $n_f = N/(n//2)$ de l'énoncé, la description de l'algorithme de transformée de Fourier locale nous indique que

Les tableaux `f` et `t` sont de taille n_f , et `S` est un tableau à deux dimensions de taille $n_f \times n_f$.

L'énoncé comporte une erreur. En testant la fonction `stft`, on peut remarquer que la taille du tableau `f` vaut en réalité $n//2 + 1$. Cette différence est due au fait que la fonction calcule par défaut une transformée de Fourier « one-sided », c'est-à-dire dans laquelle seules les fréquences positives sont conservées (les fréquences négatives étant surtout utiles pour des signaux à valeurs complexes). Pour obtenir un tableau `f` de taille n , il faudrait forcer la fonction à calculer une transformée de Fourier « two-sided » en rajoutant l'argument optionnel `return_onesided=False`.

De plus, pour des contraintes techniques (notamment la gestion des arrondis) les tailles des tableaux `t` et `S` sont proches de celles annoncées par l'énoncé, mais pas égales.

4 Le spectrogramme d'un signal est une représentation en niveaux de gris de la matrice `S` renvoyée par la fonction `stft`. Les pixels blancs correspondent à 0 et les noirs aux valeurs des spectres de Fourier calculés de module maximal.

Tout d'abord, remarquons que les deux spectrogrammes présentés, associés respectivement à un signal non bruité et au même signal bruité, sont très similaires. Le spectrogramme semble donc peu sensible au bruit. Plus précisément, les seules fréquences associées à un module non négligeable sont toutes dans l'intervalle de temps $[0,19; 0,32]$, qui correspond au « pic » de la figure 3 représentant le signal d'origine. Autrement dit, seules les composantes fréquentielles importantes du signal

Centrale Maths 1 PSI 2020 — Corrigé

Ce corrigé est proposé par Angèle Niclas (ENS Lyon) ; il a été relu par Philippe Bouafia (professeur agrégé en école d'ingénieurs) et par Florian Metzger (professeur en CPGE).

L'objectif de ce problème est d'étudier la gestion des erreurs dans un processus industriel. Des variables aléatoires X_n modélisent le nombre d'erreurs se produisant à un instant n ; S_n est le nombre total d'erreurs enregistrées jusqu'à l'instant n . On cherche dans tout le sujet à estimer la probabilité $P(S_n > nam)$ où $a > 1$ et m est le nombre moyen d'erreurs enregistrées à chaque instant. On juge que le système fonctionne et corrige ses erreurs si cette probabilité tend vers 0. Le sujet est découpé en trois parties globalement indépendantes, la première et la troisième utilisant les probabilités et la deuxième les matrices et la réduction des endomorphismes.

- Dans la partie I, on se place dans le cas simple où les variables X_n sont mutuellement indépendantes et suivent une loi de Poisson de paramètre $1/2$. À l'aide d'outils usuels de probabilité et de suites de fonctions, on prouve que $P(S_n > n)$ converge vers 0 exponentiellement vite.
- Dans la partie II, on cherche à démontrer partiellement le théorème de Perron-Frobenius : si A est une matrice strictement positive, alors son rayon spectral $\rho(A) = \max \{|\lambda| \mid \lambda \in \text{sp}(A)\}$ est une valeur propre dominante et l'espace propre associé possède certaines propriétés remarquables. Dans cette partie, les questions sont très liées les unes aux autres et demandent une bonne connaissance des outils d'algèbre linéaire.
- Dans la partie III, on considère cette fois que les variables X_n forment une chaîne de Markov, c'est-à-dire que ce qui se passe à l'instant n ne dépend que de ce qui s'est passé à l'instant $n - 1$. On essaie alors de répondre à l'objectif initial du problème. Cette partie est très variée, incluant quelques questions de programmation, des applications numériques et la mise en application des résultats de la partie II.

Ce problème est assez long, avec des questions de difficulté variée dans chaque partie. La partie I reste très classique et constitue avec la partie III un bon sujet de révision de probabilités. La partie II peut être traitée indépendamment pour réviser tout ce qui touche aux matrices. Quant à la partie III, elle s'éloigne des sujets classiques et fait appel à des capacités de synthèse et d'interprétation des résultats. Attention, l'énoncé comporte quelques erreurs.

INDICATIONS

Partie I

1 Introduire $\mathcal{I}_k = \{(i_1, \dots, i_n) \in \mathbb{N}^n \mid i_1 + \dots + i_n = k\}$ et remarquer que

$$(S_n = k) = \bigcup_{(i_1, \dots, i_n) \in \mathcal{I}_k} (X_1 = i_1) \cap \dots \cap (X_n = i_n)$$

4 Utiliser les questions 2 et 3 pour calculer la fonction génératrice de S_n qui caractérise la loi de S_n .

6 Utiliser la définition $n! = \prod_{i=1}^n i$ et l'encadrement $n^k \leq \prod_{i=n+1}^{n+k} i \leq (n+k)^k$.

8 Utiliser la question 7 pour $x = 1/n$ et le théorème de continuité de la somme ou de la double limite.

9 Combiner les informations résultant des questions 5, 6 et 8.

Partie II

13 On peut choisir de raisonner par l'absurde en supposant que $\rho(A) = 0$, puis utiliser le théorème de Cayley-Hamilton.

16 Commencer par prouver que $A^2|x| - A|x| > 0$ et $A|x| > 0$ en utilisant le résultat de la question 12, puis choisir un ε qui convient.

18 Montrer que $\rho(B) < 1$ puis appliquer le résultat de la question 14.

19 Il faut conclure que l'hypothèse $|x| < A|x|$ est fautive. Attention, la négation de $|x| < A|x|$ n'est pas $x \geq A|x|$.

20 On supposera à partir de cette question avoir prouvé en question 19 que $A|x| = |x|$.

21 Montrer que $A|x| = |Ax|$ et utiliser l'indication de la question pour $z_k = a_{1,k}x_k$.

22 Prendre deux vecteurs x, y linéairement indépendants dans $\text{Ker}(A - I_n)$ et étudier le vecteur $z = x - (x_1/y_1)y$.

23 Synthétiser les questions 13, 14, 20, 21 et 22.

25 Décomposer Y selon $\bigoplus_{\lambda \in \text{sp}(A)} E_\lambda(A)$ et utiliser le résultat de la question 24.

Partie III

29 Utiliser le système complet d'événements $(X_n = i)_{0 \leq i \leq \mathbb{N}}$.

30 Remarquer que la loi de X_n est caractérisée par Π_n .

31 Utiliser le résultat de la question 13 et de la proposition 1 sur $A(t)$.

32 Utiliser la généralisation de la proposition 2 pour $A(t)$ et le vecteur $Z(t)$.

34 S'inspirer du résultat démontré à la question 27.

35 Utiliser la convergence uniforme, vers $t \mapsto \ln(\gamma(t))$, de la suite de fonctions

$$\left(t \mapsto \frac{\ln(\mathbb{E}(e^{tS_n}))}{n} \right)_{n \in \mathbb{N}^*}$$

36 Distinguer les cas $t = 0$ et $t > 0$.

37 Utiliser la caractérisation de la borne inférieure en t et l'inégalité démontrée à la question 36.

40 Utiliser l'information de l'énoncé sur le signe de $\lambda^*(x)$ en fonction de m pour trouver un encadrement de m , puis utiliser le résultat de la question 37 pour un ε bien choisi.

I. CAS DE LA LOI DE POISSON

1 Soit $n \in \mathbb{N}^*$. Introduisons pour tout $k \in \mathbb{N}$ l'ensemble \mathcal{I}_k défini par

$$\mathcal{I}_k = \{(i_1, \dots, i_n) \in \mathbb{N}^n \mid i_1 + \dots + i_n = k\}$$

On remarque alors que

$$(S_n = k) = \bigcup_{(i_1, \dots, i_n) \in \mathcal{I}_k} (X_1 = i_1) \cap \dots \cap (X_n = i_n)$$

où l'union est disjointe. Soit $i_{n+1} \in \mathbb{N}$. Pour prouver que S_n et X_{n+1} sont indépendantes, vérifions que

$$P(S_n = k, X_{n+1} = i_{n+1}) = P(S_n = k)P(X_{n+1} = i_{n+1})$$

Notons $C = P(S_n = k, X_{n+1} = i_{n+1})$. On constate que

$$\begin{aligned} C &= P\left(\bigcup_{(i_1, \dots, i_n) \in \mathcal{I}_k} (X_1 = i_1) \cap \dots \cap (X_n = i_n) \cap (X_{n+1} = i_{n+1})\right) \\ &= \sum_{(i_1, \dots, i_n) \in \mathcal{I}_k} P(X_1 = i_1, \dots, X_n = i_n, X_{n+1} = i_{n+1}) \quad (\text{sigma additivité}) \\ &= \sum_{(i_1, \dots, i_n) \in \mathcal{I}_k} P(X_1 = i_1) \cdots P(X_n = i_n)P(X_{n+1} = i_{n+1}) \quad (\text{indépendance}) \\ C &= P(X_{n+1} = i_{n+1}) \sum_{(i_1, \dots, i_n) \in \mathcal{I}_k} P(X_1 = i_1) \cdots P(X_n = i_n) \end{aligned}$$

Par ailleurs, on remarque que

$$\begin{aligned} P(S_n = k) &= P\left(\bigcup_{(i_1, \dots, i_n) \in \mathcal{I}_k} (X_1 = i_1) \cap \dots \cap (X_n = i_n)\right) \\ &= \sum_{(i_1, \dots, i_n) \in \mathcal{I}_k} P(X_1 = i_1, \dots, X_n = i_n) \quad (\text{sigma additivité}) \\ P(S_n = k) &= \sum_{(i_1, \dots, i_n) \in \mathcal{I}_k} P(X_1 = i_1) \cdots P(X_n = i_n) \quad (\text{indépendance}) \end{aligned}$$

On a donc montré l'égalité souhaitée, et ainsi

Pour tout $n \in \mathbb{N}^*$, les variables aléatoires S_n et X_{n+1} sont indépendantes.

2 La variable aléatoire X_1 suit la loi de Poisson de paramètre $1/2$ et

$$\forall k \in \mathbb{N} \quad \forall t \in \mathbb{R} \quad P(X_1 = k)t^k = \frac{e^{-1/2}}{2^k k!} t^k = e^{-1/2} \frac{(t/2)^k}{k!}$$

D'après le cours, la série $\sum (t/2)^k / k!$ est convergente pour tout $t \in \mathbb{R}$, et sa somme vaut $e^{t/2}$. La série $\sum P(X_1 = k)t^k$ est donc convergente pour tout $t \in \mathbb{R}$, et par suite,

$$G_{X_1}(t) = \sum_{k=0}^{+\infty} P(X_1 = k)t^k = e^{-1/2} e^{t/2} = e^{\frac{t-1}{2}}$$

d'où

$$\forall t \in \mathbb{R} \quad G_{X_1}(t) = e^{\frac{t-1}{2}}$$

Si X est une variable aléatoire suivant la loi de Poisson de paramètre λ , on sait d'après le cours que $G_X(t) = e^{\lambda(t-1)}$ pour tout $t \in \mathbb{R}$. Ici, X_1 suit la loi de Poisson de paramètre $1/2$ et on retrouve le résultat précédent.

3 Montrons par récurrence sur $n \geq 1$ que la propriété

$$\mathcal{P}(n) : \quad \forall t \in \mathbb{R} \quad G_{S_n}(t) = (G_{X_1}(t))^n$$

est vraie pour tout $n \geq 1$.

- $\mathcal{P}(1)$ est vraie car $S_1 = X_1$.
- $\mathcal{P}(n) \implies \mathcal{P}(n+1)$: on remarque que $S_{n+1} = S_n + X_{n+1}$. En appliquant le résultat de la question 1, on sait que S_n et X_{n+1} sont indépendantes. On déduit d'après le cours que pour tout $t \in \mathbb{R}$,

$$G_{S_n+X_{n+1}}(t) = G_{S_n}(t)G_{X_{n+1}}(t)$$

Par hypothèse de récurrence, $G_{S_n}(t) = (G_{X_1}(t))^n$. De plus, X_{n+1} suit la même loi que X_1 (la loi de Poisson de paramètre $1/2$), de sorte que $G_{X_{n+1}} = G_{X_1}$. Finalement, pour tout $t \in \mathbb{R}$,

$$G_{S_{n+1}}(t) = G_{S_n}(t)G_{X_{n+1}}(t) = (G_{X_1}(t))^n G_{X_1}(t) = (G_{X_1}(t))^{n+1}$$

- **Conclusion** : $\forall n \geq 1 \quad \forall t \in \mathbb{R} \quad G_{S_n}(t) = (G_{X_1}(t))^n$

4 En utilisant les résultats des questions 2 et 3, on voit que pour tout $t \in \mathbb{R}$,

$$G_{S_n}(t) = (G_{X_1}(t))^n = \left(e^{\frac{t-1}{2}}\right)^n = e^{\frac{n(t-1)}{2}}$$

On reconnaît l'expression de la fonction génératrice de la loi de Poisson de paramètre $n/2$. Comme la fonction génératrice d'une variable aléatoire caractérise sa loi,

La variable aléatoire S_n suit la loi de Poisson de paramètre $n/2$.

5 Pour tout $n \in \mathbb{N}^*$, on écrit que

$$(S_n > n) = \bigcup_{k=n+1}^{+\infty} (S_n = k)$$

où l'union est disjointe. En utilisant le résultat de la question 4 donnant la loi de S_n et la sigma additivité de la probabilité, on remarque que

$$P(S_n > n) = \sum_{i=n+1}^{+\infty} P(S_n = i) = \sum_{i=n+1}^{+\infty} \frac{e^{-n/2} n^i}{2^i i!}$$

En faisant le changement d'indice $k = i - n$, on obtient

$$P(S_n > n) = e^{-n/2} \sum_{k=1}^{+\infty} \frac{n^{k+n}}{2^{k+n} (k+n)!} = e^{-n/2} \left(\frac{n}{2}\right)^n \sum_{k=1}^{+\infty} \frac{n^k}{(n+k)!} \left(\frac{1}{2}\right)^k$$

Finalement $\forall n \in \mathbb{N}^* \quad n! \left(\frac{2}{n}\right)^n P(S_n > n) = e^{-n/2} \sum_{k=1}^{+\infty} \frac{n! n^k}{(n+k)!} \left(\frac{1}{2}\right)^k$

Centrale Maths 2 PSI 2020 — Corrigé

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

Le sujet porte sur les deux branches réelles de la fonction W de Lambert. Il s'agit des fonctions réciproques des restrictions de la fonction $f: x \mapsto xe^x$ sur chacun des intervalles $]-\infty; -1]$ et $[-1; +\infty[$. La fonction W de Lambert ne pouvant être exprimée à l'aide des fonctions usuelles, le problème met en œuvre différentes techniques d'analyse pour la décrire et en donner des propriétés.

- Dans la partie I, on s'intéresse à la définition, la régularité, les variations et l'étude en 0 et $+\infty$ de la branche W et on en trace la courbe représentative. L'objectif est de pouvoir déterminer les solutions de l'inéquation

$$xe^x \leq m \tag{I.2}$$

sous forme d'intervalles dont les extrémités sont fonctions de W et d'une seconde branche, notée V . Bien qu'elle ne mobilise pour l'essentiel que des techniques d'analyse de début de première année, cette partie est assez délicate. En particulier, il faut rédiger soigneusement la résolution des inéquations.

- Dans la partie II, on se ramène à l'inéquation (I.2) en appliquant l'inégalité de Markov à des variables aléatoires qui suivent une loi de Poisson dans un premier cas et une loi binomiale dans un second. Les questions 15 et 19 font intervenir la partie I pour discuter l'existence de solutions optimales. C'est l'occasion pour le candidat de montrer sa bonne maîtrise des calculs sur ces lois.
- La première moitié de la partie III consiste à établir un « théorème binomial d'Abel » à l'aide de calculs polynomiaux bien guidés. La suite permet de vérifier que W est développable en série entière en utilisant le théorème précédent, un produit de Cauchy et les solutions d'une équation différentielle du premier ordre. Les principales techniques sur les séries, séries de fonctions et séries entières sont utilisées.
- La dernière partie établit la convergence uniforme d'une série de fonctions vers W en utilisant une méthode de point fixe. Des techniques d'analyse plus fines sont utilisées à la dernière question afin d'étendre l'intervalle de convergence.

Ce problème est riche : il comporte 39 questions, fait appel aux programmes de première et seconde années d'analyse, probabilités et (légèrement) d'algèbre. L'ensemble est bien guidé et intéressant. La fin de la première partie, ainsi que les questions 19 et 39 demandent plus d'initiative. Plusieurs questions (6, 7, 11, 21, calculs de variance aux questions 12 et 16) ne sont pas nécessaires au reste du problème et semblent destinées à tester les connaissances des candidats. La principale difficulté du sujet vient de la manipulation répétée en début de problème d'inéquations impliquant des fonctions non usuelles. Le candidat doit trouver le bon équilibre entre des raisonnements géométriques efficaces, mais qui peuvent être incomplets, et une rédaction rigoureuse mais difficile à suivre et pouvant faire perdre beaucoup de temps. Ceci est d'autant plus regrettable que les questions les plus faciles et les techniques les plus travaillées en seconde année se trouvent vers la fin du problème.

INDICATIONS

Partie I

- 1 Appliquer le théorème de la bijection monotone à la fonction f .
- 4 Utiliser la relation $f(W(x)) = x$. Au voisinage de 0, noter que $e^{W(x)} \sim 1$. Pour l'équivalent en $+\infty$, justifier que $\ln(x) = W(x) + \ln(W(x))$ pour $x > 0$, puis que $\ln(W(x))$ est négligeable devant $W(x)$ quand $x \rightarrow +\infty$.
- 5 Dédire les tangentes à \mathcal{C}_W de celles de \mathcal{C}_f par symétrie par rapport à la droite d'équation $y = x$.
- 9 Résoudre l'équation sur chaque intervalle $]-\infty; -1[$ et $[-1; +\infty[$ puis réunir les solutions trouvées.
- 10 Procéder par disjonction de cas suivant la valeur du paramètre m .
- 11 Se ramener à une équation de la forme étudiée à la question 9.

Partie II

- 12 Déterminer les probabilités conditionnelles sachant les événements ($N = n$) puis appliquer la formule des probabilités totales.
- 15 Utiliser la question 10 pour établir que si p est solution, la variable $x = -(\lambda p + 1)$ doit appartenir à $[V(-\alpha e^{-1}); W(-\alpha e^{-1})]$.
- 19 Prouver $0 < a < 1$ puis $-e^{-1} < -\alpha a e^a < 0$. En déduire que si p est solution, la variable x doit appartenir à $[V(-\alpha a e^a); W(-\alpha a e^a)]$. Vérifier, à l'aide des variations de V et W , que $-a$ appartient à ce segment et montrer que r existe quel que soit p .

Partie III

- 23 Dédire de la question 22 une expression simple de $A_k^{(j)}$.
- 25 Appliquer le résultat de la question 22 au polynôme $(X + y)^n$.
- 26 Dériver l'expression établie à la question précédente et l'évaluer en $x = 0$.
- 27 Utiliser la règle de d'Alembert.
- 29 Appliquer le théorème de continuité de la somme d'une série de fonctions. Utiliser la formule de Stirling pour obtenir un équivalent simple d'un majorant de la norme.
- 30 Former un produit de Cauchy.
- 33 Montrer à l'aide de la question 32 et des valeurs de $S(0)$ et $S'(0)$, déterminées à la question 28, que $h = f \circ s$ est la fonction identité. Utiliser le résultat de la question 9 et justifier que si $x < 0$ alors $S(x) > -1$.

Partie IV

- 36 Déterminer le maximum de ϕ'_x à l'aide d'une étude de fonction.
- 37 Appliquer l'inégalité des accroissements finis à la fonction ϕ_x .
- 39 La réponse est positive. Utiliser la continuité de W et la question 37 pour majorer uniformément $|w_n(x) - W(x)|$ au voisinage de e . Utiliser ensuite la question 38 pour obtenir une majoration uniforme en dehors de ce voisinage.

I. FONCTIONS DE LAMBERT

1 Appliquons le théorème de la bijection monotone à f sur l'intervalle $[-1; +\infty[$. C'est une application dérivable (et donc continue) sur \mathbb{R} , car produit de fonctions dérivables. Pour tout réel $x > -1$,

$$f'(x) = e^x + xe^x = e^x(1+x) > 0$$

La fonction f est donc strictement croissante sur $[-1; +\infty[$. D'après le théorème de la bijection monotone, elle réalise une bijection de l'intervalle $[-1; +\infty[$ sur son intervalle image. Comme

$$f(-1) = -e^{-1} \quad \text{et} \quad f(x) = xe^x \xrightarrow{x \rightarrow +\infty} +\infty$$

et que f est strictement croissante et continue sur l'intervalle $[-1; +\infty[$, l'intervalle image est $[-e^{-1}; +\infty[$. On a montré

L'application f réalise une bijection de $[-1; +\infty[$ sur $[-e^{-1}; +\infty[$.

Le théorème de la bijection monotone fournit deux autres résultats : la continuité et la croissance stricte de la fonction réciproque sur $[-e^{-1}; +\infty[$. Ils seront utilisés aux questions 2 et 4 et devront être justifiés au moment de la réponse à ces questions.

2 Une conséquence directe de la mise en œuvre du théorème de la bijection monotone à la question précédente est que

La fonction W est continue sur l'intervalle $[-e^{-1}; +\infty[$.

Par ailleurs, l'application f étant de classe \mathcal{C}^∞ sur \mathbb{R} , comme produit de fonctions de classe \mathcal{C}^∞ , et puisque sa dérivée ne s'annule pas sur $] -1; +\infty [$,

La fonction W est de classe \mathcal{C}^∞ sur l'intervalle $] -e^{-1}; +\infty [$.

3 Évaluons l'identité $W \circ f = \text{id}$ en 0. Puisque $f(0) = 0$, il vient

$$W(0) = 0$$

D'après la question précédente, W est dérivable en $0 \in] -e^{-1}; +\infty [$ et d'après le théorème de dérivation des fonctions réciproques

$$W'(0) = \frac{1}{f'(W(0))} = 1$$

4 Pour tout $x \in] -e^{-1}; +\infty [$, on a la relation

$$x = f(W(x)) = W(x)e^{W(x)} \tag{1}$$

La fonction W est continue en 0 et $W(0) = 0$, d'où $e^{W(x)} \underset{x \rightarrow 0}{\sim} 1$ et par quotient d'équivalents, on obtient

$$W(x) \underset{x \rightarrow 0}{\sim} x$$

On aurait aussi pu appliquer le théorème de Taylor-Young à la fonction W en 0. Puisqu'elle est de classe \mathcal{C}^∞ au voisinage de 0, elle admet un développement limité à tout ordre, en particulier à l'ordre 1 :

$$W(x) = W(0) + W'(0)(x - 0) + o_{x \rightarrow 0}(x)$$

ce qui fournit l'équivalent recherché.

Déterminons à présent un équivalent de $W(x)$ en $+\infty$. Soit un réel $x > 0$, la relation (1) justifie $W(x) > 0$. Appliquons la fonction \ln :

$$\ln(x) = \ln(W(x)e^{W(x)}) = \ln(W(x)) + W(x) \quad (2)$$

Comme la fonction W est strictement croissante et non bornée, elle tend vers $+\infty$ quand $x \rightarrow +\infty$. On en déduit par croissances comparées

$$\frac{\ln(W(x))}{W(x)} \xrightarrow{x \rightarrow +\infty} 0$$

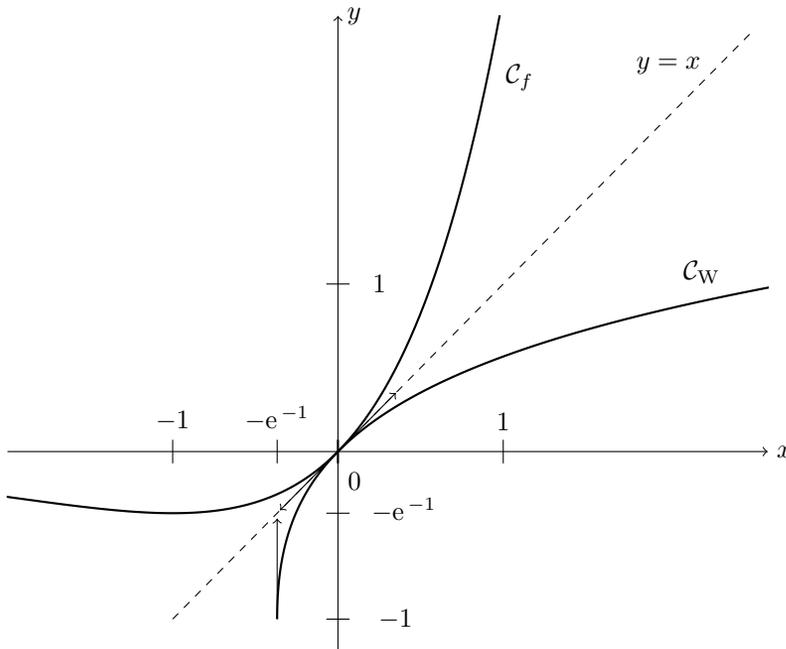
donc $\ln(W(x))$ est négligeable devant $W(x)$ quand $x \rightarrow +\infty$. Finalement, d'après (2) :

$$\boxed{W(x) \underset{x \rightarrow +\infty}{\sim} \ln(x)}$$

5 Les fonctions f et W sont dérivables en 0 et prennent la même valeur en 0, ainsi que leurs dérivées d'après la question 3. Leurs courbes représentatives possèdent donc une tangente commune au point d'abscisse $x = 0$, dont l'équation est :

$$y = f(0) + f'(0)(x - 0) = W(0) + W'(0)(x - 0) = x$$

Les courbes \mathcal{C}_f et \mathcal{C}_W ont comme tangente commune la droite d'équation $y = x$, au point d'abscisse 0.



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 ℓ 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.

Mines Maths 1 PSI 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. On peut montrer qu'un tel espace \mathcal{V} a toujours une dimension inférieure ou égale à $n(n-1)/2$ (ce point est admis par l'énoncé). L'objectif du problème est de déterminer les sous-espaces de ce type qui sont de dimension maximale: 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/PCSI.

INDICATIONS

Partie I

- 1 Utiliser une condition suffisante pour qu'une matrice de $\mathcal{M}_n(\mathbb{C})$ soit trigonalisable, puis montrer que les valeurs propres d'une matrice nilpotente sont nulles.
- 2 Donner un isomorphisme entre \mathcal{N}_B et $T_n^{++}(\mathbb{R})$. Écrire la matrice de u dans la base (e_1, \dots, e_n) et justifier que $u^{n-1} \neq 0$ et $u^n = 0$.
- 3 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 .

- 4 Montrer d'abord que $\text{Im}(u^{p-1}) \subset \text{Im}(u) \cap \text{Ker}(u)$. À l'aide de la question 3, 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

- 5 Utiliser la formule $\dim(\mathcal{L}(E, F)) = \dim(E) \times \dim(F)$. Montrer que $a \mapsto \varphi_a$ est linéaire et injective et conclure avec un argument sur les dimensions.
- 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 A 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 $t \mapsto (a | (u + tv)^{p-1}(y))$ est la fonction nulle, puis que c'est une fonction polynomiale en t . Utiliser l'égalité $(\mathbb{K}(\mathcal{V})^\perp)^\perp = \mathbb{K}(\mathcal{V})$.
- 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 C 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 Combiner les informations obtenues aux questions 14, 15, 17, 18 et le théorème A.
- 20 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 .
- 21 Avec la question 4, 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 19.
- 22 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 $y \in E$.

- 23 Raisonner par l'absurde.

I. GÉNÉRALITÉS SUR LES ENDOMORPHISMES NILPOTENTS

1 D'après un corollaire du théorème de d'Alembert-Gauss, le polynôme caractéristique χ_M de M est scindé dans $\mathbb{C}[X]$. D'après le cours, toute matrice dont le polynôme caractéristique est scindé est trigonalisable. Par conséquent,

La matrice M est semblable à une matrice triangulaire supérieure $T \in \mathcal{M}_n(\mathbb{C})$.

Comme M et T sont semblables, M^k et T^k sont semblables pour $k \in \mathbb{N}^*$. 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, il vient

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

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

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 $M^n = 0$ d'après le théorème de Cayley-Hamilton. Finalement, $u^n = 0$.

Mines Maths 2 PSI 2020 — Corrigé

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

Tout au long de ce problème, une matrice réelle A est dite *normale* si elle commute avec sa transposée, c'est-à-dire si $A {}^t A = {}^t A A$. L'objectif du sujet est la caractérisation de ces matrices et l'étude de leur exponentielle.

- On prouve tout d'abord dans une partie préliminaire que la relation notée **ORTS** est une relation d'équivalence : par définition, une matrice (réelle) A est dite orthogonalement semblable (**ORTS**) à une autre matrice réelle B s'il existe une matrice Q orthogonale vérifiant $B = {}^t Q A Q$.
- Dans les parties II à V, on établit l'équivalence entre les quatre conditions suivantes, pour une matrice réelle A :

(**C**₁) Il existe un polynôme P à coefficients réels tel que ${}^t A = P(A)$.

(**C**₂) La matrice A est normale.

(**C**₃) Pour tout vecteur X réel, $\| {}^t A X \| = \| A X \|$.

(**C**₄) La matrice A est orthogonalement semblable à une matrice diagonale par blocs, dont chaque bloc diagonal est soit de taille 1, soit de taille 2 et de la forme $r \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$ avec $(r, \theta) \in \mathbb{R}_+^* \times \mathbb{R}$.

La partie II permet d'abord de se familiariser avec les quatre conditions citées ci-dessus sur des exemples simples. La partie III établit ensuite les implications (**C**₁) \Rightarrow (**C**₂) \Rightarrow (**C**₃) et on prouve dans la partie IV que (**C**₃) \Rightarrow (**C**₄). Enfin, la partie V conclut cette étude en montrant que (**C**₄) \Rightarrow (**C**₁).

- La dernière partie du sujet définit ensuite l'exponentielle d'une matrice A par l'expression

$$\text{Exp}(A) = \lim_{p \rightarrow +\infty} \sum_{k=0}^p \frac{1}{k!} A^k$$

dont l'existence est prouvée dans la première question. La suite de la partie est consacrée à démontrer que l'image des matrices normales par l'application exponentielle est l'ensemble des matrices réelles B vérifiant deux conditions :

1. les valeurs propres négatives de B sont de multiplicité paire ;
2. il existe une matrice symétrique S à valeurs propres strictement positives et une matrice orthogonale T de déterminant 1 telles que $B = S T = T S$.

Les parties de ce problème sont de difficultés inégales : si les deux premières sont très faciles, les parties IV et VI proposent des questions plus longues et techniques qui, sans être très difficiles, demandent d'avoir bien assimilé les résultats et notations des questions précédentes. Ce sujet fait appel à quelques notions d'algèbre euclidienne, de diagonalisation, de topologie et de séries numériques, mais il donne surtout l'occasion de voir si l'on est à l'aise avec les manipulations de matrices, en particulier quand elles sont écrites par blocs.

INDICATIONS

Partie I

- 1 Montrer que la relation est réflexive, transitive et symétrique.

Partie II

- 2 Pour (\mathbf{C}_4) , utiliser le théorème spectral.
 3 Écrire les normes intervenant dans (\mathbf{C}_3) en termes de produits scalaires.
 4 Pour (\mathbf{C}_1) , calculer le polynôme caractéristique de rT et appliquer le théorème de Cayley-Hamilton.

Partie III

- 6 Les calculs sont similaires à ceux de la question 3.

Partie IV

- 7 Exploiter l'égalité des normes avec les vecteurs suggérés par l'énoncé.
 8 Traduire les égalités de normes en égalités de produits scalaires.
 9 Considérer une valeur propre pour A et un vecteur propre associé, et montrer à l'aide de la question précédente qu'ils sont aussi propres pour tA . Calculer ensuite, pour X_1 et X_2 deux vecteurs propres associés à deux valeurs propres λ_1 et λ_2 distinctes, la valeur tX_1AX_2 de deux façons différentes, faisant intervenir le produit scalaire entre X_1 et X_2 .
 10 Montrer que la matrice est diagonalisable si, et seulement si, elle est symétrique.
 11 Pour la première partie, exprimer les normes associées en termes de produits scalaires. Considérer ensuite l'endomorphisme canoniquement associé à A et utiliser le théorème 1 admis par l'énoncé pour obtenir les blocs de gauche. Appliquer ensuite la condition (\mathbf{C}_3) à des vecteurs bien choisis pour prouver que le bloc en haut à droite est nul, puis que les deux blocs diagonaux vérifient également la condition (\mathbf{C}_3) .
 12 Raisonner par récurrence.

Partie V

- 13 S'inspirer du théorème d'interpolation de Lagrange. Pour montrer que le polynôme P est réel, montrer que \bar{P} est également solution du problème et conclure par unicité.
 14 Après avoir traité le cas $\sin(\theta) = 0$, supposer que $\sin(\theta) \neq 0$. Calculer le polynôme caractéristique χ de $rR(\theta)$ et effectuer la division euclidienne de P par χ . Déterminer son reste à l'aide de la relation $P(re^{i\theta}) = re^{-i\theta}$ puis exploiter le théorème de Cayley-Hamilton.
 15 Écrire la condition (\mathbf{C}_4) , puis appliquer le résultat de la question 13 à l'ensemble des éléments diagonaux de la matrice obtenue et de leurs conjugués. Utiliser alors l'égalité de la question 14 en effectuant les calculs par blocs.

Partie VI

- 16 Majorer brutalement les termes généraux des séries et calculer la somme de la première série et de i fois la deuxième pour retomber sur une série connue.

- 17 Expliciter les coefficients du produit AB .
- 18 La suite converge si, et seulement si, toutes ses composantes (dans la base canonique de \mathcal{M}_n) convergent. Majorer alors la norme du terme général des séries obtenues à l'aide de la question précédente. Utiliser enfin le théorème 2 admis par l'énoncé.
- 19 Montrer que l'application $A \mapsto {}^t A A - A {}^t A$ est continue en considérant ses composantes, puis exploiter le fait que \mathcal{E}_n est l'image réciproque d'un fermé par cette application. Commencer par montrer que si $A \in \mathcal{E}_n$, alors $S_p(A) \in \mathcal{E}_n$ pour tout $p \in \mathbb{N}$.
- 20 Montrer tout d'abord par récurrence que $(R(\theta))^k = R(k\theta)$ pour tout $\theta \in \mathbb{R}$ et $k \in \mathbb{N}$. Calculer ensuite $S_p(R(\theta))$ pour tout $p \in \mathbb{N}$ et appliquer le résultat de la question 18. Pour le sens direct, écrire le fait que A vérifie la condition (C_4) et effectuer les calculs par blocs. Pour le sens réciproque, s'inspirer des calculs tout juste faits pour poser les bonnes valeurs.
- 21 Pour l'inclusion directe, utiliser la question précédente en séparant les éléments réels de la diagonale dans S et les blocs de taille 2 dans T . Pour l'inclusion réciproque, considérer les endomorphismes canoniquement associés à S et T et les diagonaliser simultanément dans une base orthogonale. Si des valeurs propres négatives apparaissent dans la diagonale, les regrouper deux par deux sous la forme d'une matrice multiple de $R(\pi)$ afin d'aboutir à la diagonale sous la forme demandée par l'énoncé.
- 22 Expliciter la matrice donnée dans l'énoncé et calculer son déterminant et ses valeurs propres, en séparant les cas n pair et impair.

I. QUESTION PRÉLIMINAIRE

1 Montrons que la relation **ORTS** est réflexive, transitive et symétrique. Pour cela, fixons $(A, B, C) \in (\mathcal{M}_n)^3$.

- On a $I_n \in \mathcal{O}_n$ et $A = {}^t I_n A I_n$, donc A est **ORTS** à A . La relation est réflexive.
- Si A est **ORTS** à B et B est **ORTS** à C , alors il existe P et Q dans \mathcal{O}_n telles que

$$B = {}^t P A P \quad \text{et} \quad C = {}^t Q B Q$$

Ainsi,
$$C = {}^t Q {}^t P A P Q = {}^t (PQ) A P Q$$

Or, $PQ \in \mathcal{O}_n$ car \mathcal{O}_n est un sous-groupe de $GL_n(\mathbb{R})$. Par suite, A est **ORTS** à C , puis la relation est transitive.

- Si A est **ORTS** à B , alors il existe $P \in \mathcal{O}_n$ telle que

$$B = {}^t P A P$$

Il vient
$$A = ({}^t P)^{-1} B P^{-1} = {}^t (P^{-1}) B P^{-1}$$

puisque passage à l'inverse et transposition commutent. Comme $P^{-1} \in \mathcal{O}_n$ car \mathcal{O}_n est un sous-groupe de $GL_n(\mathbb{R})$, B est **ORTS** à A , puis la relation est symétrique.

La relation **ORTS** est une relation d'équivalence sur \mathcal{M}_n .

II. EXEMPLES

2 Soit $A \in \mathcal{S}_n$. Vérifions successivement les quatre conditions :

- **(C₁)** : comme A est symétrique, ${}^t A = A$ et le polynôme $P(X) = X$ convient.
- **(C₂)** : la symétrie de A assure que $A {}^t A = A^2 = {}^t A A$, donc A est normale.
- **(C₃)** : pour tout $X \in E_n$, ${}^t A X = AX$. Ainsi, leurs normes sont égales.
- **(C₄)** : d'après le théorème spectral, l'endomorphisme canoniquement associé à la matrice A est diagonalisable en base orthogonale, donc la condition est vérifiée avec des blocs de taille 1.

Les éléments de \mathcal{S}_n vérifient les conditions **(C₁)**, **(C₂)**, **(C₃)** et **(C₄)**.

De même, pour $A \in \mathcal{A}_n$,

- **(C₁)** : comme A est antisymétrique, ${}^t A = -A$ et le polynôme $P(X) = -X$ convient.
- **(C₂)** : l'antisymétrie de A assure que $A {}^t A = -A^2 = {}^t A A$, donc A est normale.
- **(C₃)** : pour tout $X \in E_n$, ${}^t A X = -AX$. Ainsi, leurs normes sont égales.

Les éléments de \mathcal{A}_n vérifient les conditions **(C₁)**, **(C₂)** et **(C₃)**.

On va démontrer dans la suite du problème que les quatre conditions sont équivalentes, ce qui vérifiera indirectement que les matrices de \mathcal{A}_n remplissent aussi la condition **(C₄)**.

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.

X/ENS Maths PSI 2020 — Corrigé

Ce corrigé est proposé par Philippe Bouafia (professeur agrégé en école d'ingénieurs); il a été relu par Rémi Pellerin (ENS Lyon) et par Benjamin Monmege (enseignant-chercheur à l'université).

Dans ce problème, on s'attaque à diverses questions portant sur la notion de régularité des fonctions.

- Dans la partie I, on étudie l'opérateur non linéaire de régularisation de fonctions T_ε

$$\forall x \in \mathbb{R}^d \quad (T_\varepsilon h)(x) = \inf \left\{ h(y) + \frac{\|y - x\|^\alpha}{\varepsilon} \mid y \in \mathbb{R}^d \right\}$$

qui est bien défini pour des fonctions $h: \mathbb{R}^d \rightarrow \mathbb{R}$ minorées. On étudie quelques cas particuliers (questions 3, 4c, 4d) où les calculs aboutissent à une fonction $T_\varepsilon h$ « proche » de h . Lorsque $\alpha = 1$, la fonction $T_\varepsilon h$ est en fait la plus grande fonction qui soit à la fois inférieure à h et $1/\varepsilon$ -lipschitzienne. La fin de la première partie est consacrée à la preuve d'un théorème d'approximation : sous certaines conditions, la suite de fonctions $(T_{1/n} f)_{n \in \mathbb{N}^*}$ converge uniformément vers f .

- La partie II est consacrée aux fonctions concaves définies sur \mathbb{R}^d , dont on prouve le caractère continu, et même localement lipschitzien (questions 15d et 15e). Ce résultat n'a rien d'évident, puisque la concavité est une notion globale, tandis que la continuité est une notion locale. Comme le montre l'exemple de la fonction $x \mapsto -\|x\|$, ces fonctions peuvent très bien ne pas être de classe \mathcal{C}^1 sur \mathbb{R}^d . En revanche, on prouve à la fin de la partie que les fonctions qui sont en même temps concaves et convexes sont les fonctions affines.
- La partie III introduit une notion plus souple et plus générale de concavité, la K-semi-concavité, ainsi que la notion analogue de K-semi-convexité. On réutilise des résultats de la partie II pour établir que les fonctions à la fois K-semi-concaves et K-semi-convexes sont de classe \mathcal{C}^1 .

Les parties II et III peuvent être traitées indépendamment de la partie I. Comme assez souvent dans les sujets d'analyse, il ne faut pas avoir peur de la technicité. Cette épreuve est un bon moyen de s'y préparer. À noter que les dernières questions sont d'un niveau plus exigeant.

INDICATIONS

Partie I

- 3 En utilisant les notations de l'indication de l'énoncé, exprimer, à l'aide du théorème de Pythagore, la quantité $\|y\|^2 + \|x - y\|^2/\varepsilon$ en fonction de λ et $\|y_\perp\|^2$, puis trouver les valeurs de λ et $\|y_\perp\|$ qui la minimisent.
- 4c Étudier séparément les cas $\varepsilon \leq 1$ et $\varepsilon > 1$. On rappelle que
- $$\forall (x, y) \in (\mathbb{R}^d)^2 \quad \left| \|x\| - \|y\| \right| \leq \|x - y\| \leq \|x\| + \|y\|$$
- 4d Faire appel au résultat de la question 2.
- 5 Montrer les deux inégalités $T_\varepsilon h \leq |h|_\infty$ et $T_\varepsilon h \geq -|h|_\infty$.
- 7 Justifier l'existence d'un minimum pour la fonction $y \mapsto f(y) + \|y - x\|^\alpha/\varepsilon$, définie sur $A(x)$.
- 10 À $x \in \mathbb{R}^d$ fixé, montrer que l'on a $f(y) + \|y - x\|^\alpha/\varepsilon \geq -\omega_f(r_\varepsilon) + f(x)$ pour tout $y \in A(x)$.

Partie II

- 15a Remarquer que les éléments de $\{-M, M\}^{d+1}$ sont de la forme $(Y, -M)$ ou (Y, M) , avec $Y \in \{-M, M\}^d$.
- 15d Poser le point $z = y + \frac{M(y-x)}{2\|y-x\|}$.
- 16a Considérer la restriction de la fonction $X \mapsto \|Y - X\|$, définie sur C , à une partie fermée et bornée de C .
- 16b Remarquer que la distance de Y à tout point du segment reliant les points Y_0 et X est supérieure ou égale à la distance entre Y et Y_0 .
- 17b Considérer le segment reliant les points X_ε et $(x_*, f(x_*) + \varepsilon)$.
- 17c Appliquer le résultat de la question 16b.
- 17d Appliquer l'inégalité de la question 17c à une valeur bien choisie de x .
- 19 Montrer que la suite $(p_n)_{n \in \mathbb{N}^*}$ admet une sous-suite convergente, puis passer à la limite dans l'inégalité de la question 17c, appliquée à des valeurs de ε se rapprochant de 0, pour obtenir la majoration désirée.
- 20 Montrer qu'il s'agit des fonctions affines, c'est-à-dire de la forme $x \mapsto a + \langle b | x \rangle$, avec $(a, b) \in \mathbb{R} \times \mathbb{R}^d$.

Partie III

- 23a Décomposer la fonction $y \mapsto f(y) - K\|y - x\|^2$ en la somme d'une fonction concave et d'une fonction affine.
- 23b Majorer $\|p_x\|$ par une quantité indépendante de x . Pour cela, appliquer convenablement la question 22.
- 23c Pour des points x et y fixés dans \mathbb{R}^d , considérer l'ensemble des $t \in [0; 1]$ pour lesquels $|f(x + t(y - x)) - f(x)| \leq K't$.
- 24a Appliquer le résultat de la question 22 à f et à $-f$.
- 24b Appliquer l'inégalité de la question 24a à $y = x - t(p_x - q_x)$, où $t > 0$.
- 24d Appliquer plusieurs fois l'inégalité de la question 24a à des valeurs de (x, y) judicieusement choisies.
- 24e Appliquer la question précédente à $h = \|x - y\| \frac{\nabla f(x) - \nabla f(y)}{\|\nabla f(x) - \nabla f(y)\|}$ (dans le cas où $\nabla f(x) \neq \nabla f(y)$).

I. APPROXIMATION PAR DES FONCTIONS LIPSCHITZIENNES

1 Il faut donner un sens à la quantité $\|y - x\|^\alpha$ qui apparaît dans la définition de l'opérateur T_ε . Lorsque $x \neq y$, on rappelle qu'elle vaut $\exp(\alpha \ln \|y - x\|)$. Pour que cette quantité soit également définie dans le cas $x = y$, prenons la décision d'introduire la convention suivante : $0^\alpha = 0$ pour tout $\alpha > 0$. Avec ce choix, on remarque que $\|y - x\|^\alpha \geq 0$ quels que soient $x, y \in \mathbb{R}^d$ et $\alpha > 0$, ce que l'on utilisera de manière implicite dans la suite.

Pour montrer que la fonction $T_\varepsilon h$ est bien définie, il suffit d'établir que, pour tout choix de $x \in \mathbb{R}^d$, l'ensemble non vide

$$\left\{ h(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \mid y \in \mathbb{R}^d \right\}$$

est minoré. Cela provient de l'hypothèse selon laquelle la fonction h est elle-même minorée. En effet, en notant m un de ses minorants, on vérifie que, pour tout $y \in \mathbb{R}^d$, on a bien

$$h(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \geq m$$

Par ailleurs, en considérant la valeur particulière $y = x$, on s'assure que

$$h(x) \in \left\{ h(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \mid y \in \mathbb{R}^d \right\}$$

d'où $T_\varepsilon h(x) = \inf \left\{ h(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \mid y \in \mathbb{R}^d \right\} \leq h(x)$

La fonction $T_\varepsilon h$ est bien définie et vérifie $T_\varepsilon h \leq h$.

2 Notons m_1 et m_2 des minorants respectifs des fonctions h_1 et h_2 . Par construction, la fonction H est minorée par $\min(m_1, m_2)$. Il découle de la question précédente que

La fonction $T_\varepsilon H$ est bien définie sur \mathbb{R}^d .

Fixons à présent $x \in \mathbb{R}^d$ pour le reste de la question. Pour tout $y \in \mathbb{R}^d$, on a l'inégalité $h_1(y) \geq H(y)$. On peut donc écrire

$$\begin{aligned} h_1(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha &\geq H(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \\ &\geq T_\varepsilon H(x) \end{aligned} \quad (\text{définition de } T_\varepsilon H)$$

Ceci est valable quel que soit $y \in \mathbb{R}^d$. En passant à la borne inférieure, on en déduit que $T_\varepsilon h_1(x) \geq T_\varepsilon H(x)$. Puisque h_1 et h_2 jouent des rôles symétriques, il vient aussi que $T_\varepsilon h_2(x) \geq T_\varepsilon H(x)$. Donc $\min(T_\varepsilon h_1, T_\varepsilon h_2)(x) \geq T_\varepsilon H(x)$.

Attaquons-nous à l'inégalité réciproque. Notons que, quel que soit $y \in \mathbb{R}^d$, on a soit $H(y) = h_1(y)$ soit $H(y) = h_2(y)$. Dans le premier cas,

$$H(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha = h_1(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \geq T_\varepsilon h_1(x) \geq \min(T_\varepsilon h_1, T_\varepsilon h_2)(x)$$

et dans le second cas, on obtient la même inégalité. On a alors

$$\forall y \in \mathbb{R}^d \quad H(y) + \frac{1}{\varepsilon} \|y - x\|^\alpha \geq \min(T_\varepsilon h_1, T_\varepsilon h_2)(x)$$

Le minorant $\min(T_\varepsilon h_1, T_\varepsilon h_2)(x)$ ne dépend pas de y . Il ne reste plus qu'à passer une nouvelle fois à la borne inférieure pour parvenir à $T_\varepsilon H(x) \geq \min(T_\varepsilon h_1, T_\varepsilon h_2)(x)$. Finalement,

$$T_\varepsilon H = \min(T_\varepsilon h_1, T_\varepsilon h_2)$$

3 Soit $x \in \mathbb{R}^d$. Comme \mathbb{R}^d est de dimension finie, les espaces $\text{Vect}(x)$ et $\text{Vect}(x)^\perp$ sont supplémentaires. Par conséquent, tout vecteur $y \in \mathbb{R}^d$ s'écrit (de manière unique) sous la forme $y = \lambda x + y_\perp$ avec $\lambda \in \mathbb{R}$ et $y_\perp \in \text{Vect}(x)^\perp$. En appliquant deux fois le théorème de Pythagore, on obtient

$$\begin{aligned} \|y\|^2 + \frac{1}{\varepsilon} \|x - y\|^2 &= \|\lambda x + y_\perp\|^2 + \frac{1}{\varepsilon} \|(1 - \lambda)x - y_\perp\|^2 \\ &= \lambda^2 \|x\|^2 + \|y_\perp\|^2 + \frac{(1 - \lambda)^2}{\varepsilon} \|x\|^2 + \frac{1}{\varepsilon} \|y_\perp\|^2 \\ \|y\|^2 + \frac{1}{\varepsilon} \|x - y\|^2 &= \left(\lambda^2 + \frac{(1 - \lambda)^2}{\varepsilon} \right) \|x\|^2 + \left(1 + \frac{1}{\varepsilon} \right) \|y_\perp\|^2 \end{aligned}$$

On a donc une somme de deux termes positifs. Cette quantité est minimale lorsque chaque terme l'est. Le second terme est minimal pour $y_\perp = 0$. Pour le premier, il suffit d'étudier la fonction

$$\varphi: t \mapsto t^2 + \frac{(1 - t)^2}{\varepsilon} = \left(1 + \frac{1}{\varepsilon} \right) t^2 - \frac{2}{\varepsilon} t + \frac{1}{\varepsilon}$$

C'est une fonction polynomiale du second degré de la forme $t \mapsto at^2 + bt + c$ avec a strictement positif. Elle atteint son minimum en $t_{\min} = -b/2a$, soit $t_{\min} = 1/(1 + \varepsilon)$. Un calcul donne par ailleurs $\varphi(t_{\min}) = 1/(1 + \varepsilon)$. Par conséquent,

$$\forall x \in \mathbb{R}^d \quad \mathbf{T}_\varepsilon g(x) = \frac{\|x\|^2}{1 + \varepsilon} = \frac{g(x)}{1 + \varepsilon}$$

4a Fixons $x, x' \in \mathbb{R}^d$. Pour tout $y \in \mathbb{R}^d$, on a

$$\begin{aligned} \mathbf{T}_\varepsilon h(x) &\leq h(y) + \frac{\|y - x\|}{\varepsilon} && \text{(définition de } \mathbf{T}_\varepsilon h) \\ &\leq h(y) + \frac{\|y - x'\|}{\varepsilon} + \frac{\|x' - x\|}{\varepsilon} && \text{(inégalité triangulaire)} \end{aligned}$$

Autrement dit, on a

$$\forall y \in \mathbb{R}^d \quad \mathbf{T}_\varepsilon h(x) - \frac{\|x' - x\|}{\varepsilon} \leq h(y) + \frac{\|y - x'\|}{\varepsilon}$$

On passe à la borne inférieure, ce qui fournit l'inégalité

$$\mathbf{T}_\varepsilon h(x) - \frac{\|x' - x\|}{\varepsilon} \leq \mathbf{T}_\varepsilon h(x')$$

En inversant les rôles de x et x' dans le raisonnement précédent, on établit que

$$\mathbf{T}_\varepsilon h(x') - \frac{\|x - x'\|}{\varepsilon} \leq \mathbf{T}_\varepsilon h(x)$$

Ces deux inégalités se résument en

$$|\mathbf{T}_\varepsilon h(x') - \mathbf{T}_\varepsilon h(x)| \leq \frac{1}{\varepsilon} \|x' - x\|$$

Ceci étant vrai pour tous x et x' dans \mathbb{R}^d , il vient que

$$\text{La fonction } \mathbf{T}_\varepsilon h \text{ est } \frac{1}{\varepsilon}\text{-lipschitzienne.}$$