Annales des Concours

PSI
Mathématiques · Informatique
2018

Sous la coordination de

William Aufort
ENS Lyon
ancien élève de l'École Normale Supérieure (Lyon)

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

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

Par

Virgile Andreani
ENS Ulm

Guillaume Batog
professeur en CPGE

Loïc Devilliers
ENS Paris-Saclay

Hervé Diet
professeur agrégé

Corentin Fierobe
ENS Lyon

Jean-Julien Fleck
professeur en CPGE

Emma Kerinec
ENS Lyon

Thierry Limoges
ENS Paris-Saclay

Romain Panis
ENS Ulm

Cyril Ravat
professeur en CPGE

Sommaire

Mines-Ponts

Polytechnique-ENS

Formulaires

Sommaire thématique de mathématiques

2015 – 2018

	Structures algébriques et arithmétique	Polynômes	Algèbre linéaire générale	Réduction des endomorphismes	Produit scalaire et espaces euclidiens	Topologie des espaces vectoriels normés	Suites et séries numériques	Suites et séries de fonctions	Séries entières	Analyse réelle	Intégration	Équations différentielles	Fonctions de plusieurs variables	Dénombrement et probabilités
CCP MP Maths 1		•	•		•		•	••••	••	•	•••		•	••
CCP MP Maths 2	•	•	••	••••	•••						•	•		•
CCP PC Maths		••	••	••	•	•		•	••		•••			•••
CCP 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		••	•••		••	•				•		••	•	

SESSION 2018 **PSIMA02**

CONCOURS COMMUNS POLYTECHNIQUES

ÉPREUVE SPÉCIFIQUE - FILIÈRE PSI

MATHÉMATIQUES

Lundi 30 avril : 14 h - 18 h

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

Les calculatrices sont interdites

Le sujet est composé de 2 problèmes indépendants.

PROBLÈME 1

Ce problème comporte 3 parties indépendantes.

Notations et définitions

- $\mathbf{N}$ désigne l'ensemble des entiers naturels, $\mathbf{N}^*$ désigne l'ensemble des entiers naturels non nuls.
- $\mathbf{R}$ désigne l'ensemble des nombres réels.
- $\mathbf{R}[X]$ désigne le $\mathbf{R}$-espace vectoriel des polynômes à coefficients réels et, pour tout entier $n \in \mathbf{N}$, on note $\mathbf{R}_n[X]$ le $\mathbf{R}$-espace vectoriel des polynômes à coefficients réels de degré inférieur ou égal à n.
- Si n_1 et n_2 sont deux entiers naturels, on note $[\![n_1, n_2]\!]$ l'ensemble des entiers naturels compris (au sens large) entre n_1 et n_2.

Objectifs

On s'intéresse dans ce problème à l'équation différentielle $x^2y'' + axy' + by = 0$. La **partie I** est une partie d'algèbre linéaire qui traite des solutions polynomiales de cette équation lorsque a et b sont des constantes réelles. Dans la **partie II**, on détermine l'ensemble des solutions de l'équation lorsque a et b sont des constantes réelles. La **partie III** traite des solutions de cette équation lorsque $a = 1$ et b est la fonction carrée.

Partie I - Endomorphismes

Dans toute cette partie, n désigne un entier naturel non nul et a et b des constantes réelles.

Q1. On note Δ l'endomorphisme de $\mathbf{R}[X]$ défini par :

$$\forall P \in \mathbf{R}[X], \quad \Delta(P) = XP.'$$

Calculer, pour tout $k \in [\![0, n]\!]$, $\Delta(X^k)$.

Q2. Montrer que pour tout $P \in \mathbf{R}[X]$, $X^2P'' = \Delta \circ (\Delta - \mathrm{Id})(P)$, où Id désigne l'endomorphisme identité sur $\mathbf{R}[X]$.

Q3. Montrer que si $P \in \mathbf{R}_n[X]$, $\Delta(P) \in \mathbf{R}_n[X]$.

On notera Δ_n l'endomorphisme de $\mathbf{R}_n[X]$ induit par Δ.

Q4. Déterminer la matrice de Δ_n dans la base canonique $(1, X, \cdots, X^n)$ de $\mathbf{R}_n[X]$.

Q5. On définit l'application Φ par :

$$\forall P \in \mathbf{R}[X], \quad \Phi(P) = X^2P'' + aXP'.$$

Montrer que $\Phi = \Delta^2 + (a-1)\Delta$ et en déduire que Φ définit un endomorphisme de $\mathbf{R}[X]$.

Q6. Montrer que Φ induit un endomorphisme Φ_n de $\mathbf{R}_n[X]$.

Q7. Montrer que Φ_n est diagonalisable.

On considère l'endomorphisme φ de $\mathbf{R}[X]$ défini par :

$$\forall P \in \mathbf{R}[X], \quad \varphi(P) = X^2P'' + aXP' + bP.$$

Q8. Montrer que φ induit un endomorphisme de $\mathbf{R}_n[X]$, endomorphisme que l'on notera φ_n. Exprimer φ_n en fonction de Δ_n.

Q9. Exprimer la matrice de φ_n dans la base canonique de $\mathbf{R}_n[X]$.

On considère l'équation :

$$s^2 + (a-1)s + b = 0. \tag{1}$$

Q10. Expliciter le noyau de φ_n lorsque l'équation (1) admet deux racines entières $m_1, m_2 \in [\![0, n]\!]$.

Q11. Expliciter le noyau de φ_n lorsque l'équation (1) admet une unique racine entière $m \in [\![0, n]\!]$.

Q12. Déterminer le noyau de φ. En déduire qu'il est de dimension finie et déterminer sa dimension.

Partie II - Une équation différentielle

On considère dans cette partie l'équation différentielle

$$x^2 y'' + axy' + by = 0, \tag{2}$$

où a et b sont des constantes réelles.

Q13. Que déduit-on du théorème de Cauchy quant à la structure de l'ensemble des solutions de l'équation (2) sur $I =]0, +\infty[$? Et sur $J =]-\infty, 0[$?

Q14. Montrer que si y est une solution de (2) sur I, alors $g = y \circ \exp$ est une solution sur $\mathbf{R}$ de l'équation différentielle linéaire à coefficients constants :

$$u'' + (a-1)u' + bu = 0. \tag{3}$$

Q15. Réciproquement, soit $t \mapsto g(t)$ une solution de (3) sur $\mathbf{R}$. Montrer que la fonction $g \circ \ln$ est solution de (2) sur I.

Q16. Donner les solutions à valeurs réelles de l'équation (3) dans le cas où $a = 3$ et $b = 1$ et dans le cas où $a = 1$ et $b = 4$. En déduire, dans chacun des cas, les solutions à valeurs réelles de l'équation (2) sur l'intervalle I.

On suppose *dans les deux questions suivantes uniquement* que $a = 1$ et $b = -4$.

Q17. Montrer que si y est solution de (2) sur J, alors $h = y \circ (-\exp)$ est solution de (3) sur $\mathbf{R}$.

Q18. Déduire de ce qui précède l'ensemble des solutions de (2) de classe C^2 sur $\mathbf{R}$.

Partie III - Une équation de Bessel

On se propose dans cette partie d'étudier l'équation différentielle :

$$x^2 y'' + xy' + x^2 y = 0. \tag{4}$$

Q19. Rappeler la définition du rayon de convergence d'une série entière.

CCP Maths PSI 2018 — Corrigé

Ce corrigé est proposé par Thierry Limoges (professeur en CPGE) ; il a été relu par Corentin Fierobe (ENS Lyon) et Florian Metzger (docteur en mathématiques).

Ce sujet est composé de deux problèmes indépendants cumulant 44 questions.

Le premier s'intéresse à l'équation différentielle

$$x^2\,y'' + a\,x\,y' + b\,y = 0$$

- La première partie traite des solutions polynomiales lorsque a et b sont des constantes réelles, en utilisant des outils d'algèbre linéaire en dimension finie.
- Ensuite, on détermine l'ensemble des solutions lorsque a et b sont des constantes réelles, en se ramenant à une équation différentielle linéaire à coefficients constants par un changement de fonction.
- La dernière partie est consacrée à l'étude des solutions de l'équation

  $$x^2\,y'' + x\,y' + x^2\,y = 0$$

  On commence par chercher les solutions développables en série entière au voisinage de zéro. C'est un cas particulier d'équation de Bessel. Les fonctions solutions de cette famille d'équations interviennent dans de nombreux problèmes physiques circulaires, comme la propagation d'ondes dans un tuyau dont la section est circulaire. C'est en coordonnées cylindriques que ces problèmes s'expriment le plus simplement.

Le second problème aborde un cas particulier de la loi forte des grands nombres : pour $(\mathrm{X}_i)_{i\in\mathbb{N}^*}$ une suite de variables aléatoires centrées de même loi, la suite de variables aléatoires $(\mathrm{S}_n)_{n\in\mathbb{N}^*}$ définie par

$$\mathrm{S}_n = \frac{\mathrm{X}_1 + \cdots + \mathrm{X}_n}{n}$$

converge presque sûrement vers 0, c'est-à-dire que l'ensemble

$$\Big\{\omega \in \Omega \mid \lim_{n\to+\infty} \mathrm{S}_n(\omega) = 0\Big\}$$

a pour probabilité 1. On y utilise des propriétés sur les événements, union, intersection et complémentaire, ainsi que sur la continuité croissante et décroissante d'une suite d'événements.

Ce sujet est composé de nombreuses questions assez courtes, mais pas toujours évidentes, ce qui en fait un ensemble relativement long. Il permet de balayer de nombreuses parties du programme, y compris de première année : algèbre linéaire en dimension finie, équations différentielles linéaires, séries entières et probabilités.

Indications

Problème 1

7 Déterminer la matrice de φ_n dans la base canonique.

10 Déterminer le rang de φ_n à l'aide de sa matrice, en déduire la dimension du noyau et en trouver une base.

11 Même indication que pour la question 10.

12 Remarquer que $\deg(\varphi(\mathrm{P})) = \deg(\mathrm{P})$ lorsque $\deg(\mathrm{P})$ n'est pas solution de (1). Considérer les éventuelles solutions entières de (1), et raisonner dans $\mathbb{R}_n[\mathrm{X}]$ avec n suffisamment grand.

14 Calculer g' et g'', puis les injecter dans l'équation (3).

15 Calculer $(g \circ \ln)'$ et $(g \circ \ln)''$, puis les injecter dans l'équation (2).

16 Appliquer le cours de première année sur les équations différentielles linéaires à coefficients constants.

17 Calculer h' et h'', puis les injecter dans l'équation (3).

18 Adapter la question 15 pour obtenir les solutions de (2) sur J. Étudier le recollement de classe $\mathscr{C}^2$ en 0 de solutions sur I et sur J.

20 Calculer ${\mathrm{J}_0}'$ et ${\mathrm{J}_0}''$, puis les injecter dans (4) et montrer la formule par récurrence.

21 Appliquer le critère de d'Alembert pour les séries numériques.

22 Écrire une relation de colinéarité entre J_0 et f. Utiliser qu'une fonction continue en 0 est bornée au voisinage de 0.

23 Écrire la formule du produit de Cauchy de deux séries entières.

24 Revenir à la définition du rayon de convergence d'une série entière.

26 Pour $\rho = \dfrac{r}{\mathrm{M}+1}$, montrer que la suite $(\beta_n \rho^n)$ est bornée.

27 Calculer d'une part la dérivée de $x\mathrm{J}_0\lambda'$, d'autre part y' et y'', puis les injecter dans (4).

28 Faire le produit de Cauchy de la série entière qui définit J_0 avec elle-même.

29 Chercher η sous la forme $\eta = \mathrm{J}_0 + \mu$ et utiliser la question 27.

30 Montrer que (J_0, f) avec $f = \eta + \mathrm{J}_0 \ln$ est une base des solutions de (4).

Problème 2

31 Majorer en valeur absolue les termes de la série qui définit l'espérance.

34 Montrer que les événements $\{\mathrm{S}_n \geqslant \varepsilon\}$ et $\left\{\mathrm{e}^{tn\mathrm{S}_n} \geqslant \mathrm{e}^{tn\varepsilon}\right\}$ sont égaux. Utiliser l'inégalité de Markov, puis que l'espérance d'un produit de variables aléatoires mutuellement indépendantes est le produit de leurs espérances.

35 Dériver deux fois g_a, puis appliquer le théorème des accroissements finis sur les intervalles $[\,-1\,;x\,]$ et sur $[\,x\,;1\,]$ pour $x \in \,]\,-1\,;1\,[$.

37 Majorer le terme général de la série qui définit l'espérance de $\mathrm{e}^{t\mathrm{X}}$, et utiliser que la variable aléatoire X est centrée.

38 Développer en série entière les fonctions ch et $t \longmapsto \mathrm{e}^{\frac{t^2}{2}}$.

39 Étudier la fonction f sur $\mathbb{R}$.

40 Montrer que $\{|S_n| \geqslant \varepsilon\} = \{S_n \geqslant \varepsilon\} \cup \{-S_n \geqslant \varepsilon\}$. Prendre $t > 0$ et utiliser les questions 34 et 39, puis considérer $-S_n$.

41 Majorer le terme général de la série par celui d'une série géométrique convergente.

42 Une image réciproque d'un intervalle par une variable aléatoire est un événement et une union ou intersection dénombrable d'événements est aussi un événement. Majorer $P(B_n)$ par le reste d'une série convergente et utiliser la continuité décroissante d'une probabilité.

43 Écrire Ω_k comme union d'intersections d'événements, puis A comme intersection.

44 Utiliser la continuité décroissante d'une probabilité et la question 42.

Problème 1

$\boxed{1}$ Soit $k \in [\![1\,;n]\!]$. On a $\Delta(\mathrm{X}^k) = \mathrm{X}k\mathrm{X}^{k-1} = k\mathrm{X}^k$ et $\Delta(1) = \mathrm{X} \times (1)' = 0$. Finalement,

$$\boxed{\forall k \in [\![0\,;n]\!] \quad \Delta(\mathrm{X}^k) = k\mathrm{X}^k}$$

$\boxed{2}$ Soit $\mathrm{P} \in \mathbb{R}[\mathrm{X}]$. Calculons, par linéarité de la dérivation,

$$\Delta \circ (\Delta - \mathrm{Id})(\mathrm{P}) = \Delta(\mathrm{XP}' - \mathrm{P}) = \mathrm{X}(\mathrm{XP}' - \mathrm{P})' = \mathrm{X}(\mathrm{P}' + \mathrm{XP}'' - \mathrm{P}') = \mathrm{X}^2\mathrm{P}''$$

On obtient $\boxed{\forall\, \mathrm{P} \in \mathbb{R}[\mathrm{X}] \quad \mathrm{X}^2\mathrm{P}'' = \Delta \circ (\Delta - \mathrm{Id})(\mathrm{P})}$

$\boxed{3}$ Soit $\mathrm{P} \in \mathbb{R}_n[\mathrm{X}]$. Si P est constant, alors $\mathrm{XP}' = 0 \in \mathbb{R}_n[\mathrm{X}]$. Sinon, P est non constant, et $\deg(\mathrm{P}') = \deg(\mathrm{P}) - 1$. Comme $\deg(\mathrm{AB}) = \deg(\mathrm{A}) + \deg(\mathrm{B})$ pour tout $(\mathrm{A}, \mathrm{B}) \in \mathbb{R}[\mathrm{X}]^2$, on obtient

$$\deg(\mathrm{XP}') = \deg(\mathrm{X}) + \deg(\mathrm{P}') = 1 + \deg(\mathrm{P}) - 1 = \deg(\mathrm{P}) \leqslant n$$

donc $\boxed{\forall\, \mathrm{P} \in \mathbb{R}_n[\mathrm{X}] \quad \Delta(\mathrm{P}) \in \mathbb{R}_n[\mathrm{X}]}$

$\boxed{4}$ D'après la question 1, on a $\Delta_n(\mathrm{X}^k) = k\mathrm{X}^k$ pour $0 \leqslant k \leqslant n$. La matrice de Δ_n dans la base canonique $(1, \mathrm{X}, \ldots, \mathrm{X}^n)$ est donc la matrice diagonale

$$\boxed{\mathrm{Mat}_{\,(1,\mathrm{X},\ldots,\mathrm{X}^n)}(\Delta_n) = \mathrm{diag}(0, 1, \ldots, n) = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & n \end{pmatrix} \in \mathscr{M}_{n+1}(\mathbb{R})}$$

$\boxed{5}$ D'après la question 2 et par linéarité de Δ, on a

$$\Phi = \Delta \circ (\Delta - \mathrm{Id}) + a\Delta = \Delta^2 - \Delta + a\Delta = \Delta^2 + (a-1)\Delta$$

Ainsi, Φ est une combinaison linéaire d'endomorphismes de $\mathbb{R}[\mathrm{X}]$, d'où

$$\boxed{\text{L'application } \Phi = \Delta^2 + (a-1)\Delta \text{ est un endomorphisme de } \mathbb{R}[\mathrm{X}].}$$

$\boxed{6}$ D'après la question 5, on a $\Phi = \Delta^2 + (a-1)\Delta$. Or $\mathbb{R}_n[\mathrm{X}]$ est stable par Δ, donc par Δ^2 et par $(a-1)\Delta$ soit finalement par Φ.

$$\boxed{\text{L'application } \Phi \text{ induit un endomorphisme } \Phi_n \text{ sur } \mathbb{R}_n[\mathrm{X}].}$$

$\boxed{7}$ D'après les questions 5 et 6, $\Phi_n = {\Delta_n}^2 + (a-1)\Delta_n$. Soit M la matrice de Δ_n définie dans la question 4. Ainsi, la matrice de Φ_n dans la base canonique $(1, \mathrm{X}, \ldots, \mathrm{X}^n)$ de $\mathbb{R}_n[\mathrm{X}]$ est

$$\begin{aligned} \mathrm{M}^2 + (a-1)\mathrm{M} &= \mathrm{diag}(0, 1, \ldots, n)^2 + (a-1)\mathrm{diag}(0, 1, \ldots, n) \\ &= \mathrm{diag}(0, 1^2, \ldots, n^2) + \mathrm{diag}((a-1) \times 0, (a-1) \times 1, \ldots, (a-1) \times n) \\ &= \mathrm{diag}(0, 1 + a - 1, \ldots, n^2 + (a-1)n) \end{aligned}$$

$$\mathrm{M}^2 + (a-1)\mathrm{M} = \mathrm{diag}\left(k^2 + (a-1)k\right)_{0 \leqslant k \leqslant n}$$

CCP Informatique PSI 2018 — Corrigé

Ce corrigé est proposé par Virgile Andreani (ENS Ulm) ; il a été relu par Emma Kerinec (ENS Lyon) et Julien Dumont (professeur en CPGE).

Ce sujet d'une vingtaine de questions traite du fonctionnement du système Hawk-Eye, utilisé au tennis comme assistance à l'arbitrage sur la validité des impacts des balles sur le terrain. On passe en revue une grande partie du système, de l'identification automatique de la balle sur les images des caméras à la visualisation finale de sa trajectoire en 3D.

- Une première partie très courte, proche du cours, demande de programmer un schéma d'Euler pour simuler la trajectoire de la balle à partir de conditions initiales. Cette simulation n'est pas suffisamment précise pour détecter la validité de l'impact d'une balle, de nombreux paramètres (météo, déformation de la balle...) n'étant pas pris en compte.
- La deuxième partie remédie à ce problème en présentant le système Hawk-Eye. Après plusieurs questions portant sur l'architecture d'un ordinateur et la représentation des données en mémoire, on demande d'écrire quelques fonctions de traitement d'image pour repérer la position de la balle sur les films capturés par les caméras. Après la segmentation, le *tracking* est abordé et il faut implémenter un algorithme permettant de suivre la trajectoire de la balle au cours du temps. Puis de l'algèbre linéaire permet de reconstruire la position de la balle en 3D sur le terrain à partir de ses positions dans les champs de vision des caméras. Enfin, on détermine la position de l'impact de la balle sur le sol.
- La dernière partie pose quatre questions faciles sur les bases de données.

Le système étudié est complexe et intéressant. Cependant, pour pouvoir en aborder presque tous ses aspects, le sujet a dû les simplifier à l'extrême et la plupart des questions sont quasiment immédiates : aucune ne requiert de réelle réflexion, que ce soit sur l'algorithmique ou sur la programmation. C'est en conséquence un sujet accessible dès la première année, mais d'un intérêt limité pour les révisions. De plus, il contient plusieurs erreurs, y compris dans les sections de code à commenter, et n'encourage pas de bonnes pratiques de programmation en Python.

Indications

Partie II

1 Ne pas oublier d'exprimer α en fonction des éléments de Y.

Partie III

4 Multiplier la taille en mémoire d'un élément du tableau par le nombre d'éléments de celui-ci.

10 Indice : il y a une erreur dans la première boucle `for`, qui ne se comporte pas comme elle le devrait.

11 Attention, les coordonnées sont stockées « à plat » dans la liste `liste_balle_i` : c'est une liste 1D. Il faut donc faire un peu d'arithmétique pour y accéder.

12 Que se passerait-il si, par malchance, le signal le plus proche du centre de la zone ne correspondait pas à une balle ?

14 Il est facile de se tromper dans cette question. Commencer par écrire les formules de changement de base entre le repère d'origine et le repère translaté, puis celui translaté et tourné une fois, et enfin celui de la caméra (translaté et tourné deux fois).

16 La précision de l'énoncé (le point se situe durant un échange) est importante : elle signifie que le premier impact de la balle sur la terre n'est peut-être pas celui que l'on souhaite étudier. Une hypothèse raisonnable est que le dernier impact est celui qui nous intéresse.

17 Pour rappel, Python permet d'écrire de manière concise une double inégalité : `a <= b <= c` au lieu de `a <= b and b <= c`.

Partie IV

20 Ne pas utiliser le champ `nom`, mais plutôt les deux champs `joueur`.

II. Modélisation de la trajectoire de la balle

1 Il s'agit d'écrire la dérivée de Y en fonction de ses composantes. On a d'abord

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -\frac{1}{2m}\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1\mathrm{V}^2\cos\alpha - \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\mathrm{V}\Omega\sin\alpha$$

ainsi que $$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{\mathrm{d}^2y}{\mathrm{d}t^2} = -g - \frac{1}{2m}\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1\mathrm{V}^2\sin\alpha + \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\mathrm{V}\Omega\cos\alpha$$

Pour faire disparaître la variable α, on remarque que $u(t) = \mathrm{V}\cos\alpha$ et $v(t) = \mathrm{V}\sin\alpha$ avec $\mathrm{V} = \sqrt{u^2+v^2}$, soit

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{1}{2m}\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1u(t)\sqrt{u(t)^2+v(t)^2} - \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\Omega v(t)$$

et $$\frac{\mathrm{d}v}{\mathrm{d}t} = -g - \frac{1}{2m}\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1v(t)\sqrt{u(t)^2+v(t)^2} + \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\Omega u(t)$$

Les dérivées des deux dernières composantes sont données par

$$\frac{\mathrm{d}x}{\mathrm{d}t} = u \qquad \text{et} \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = v$$

Enfin, $$\boxed{\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u \\ v \\ x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{1}{2m}\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1u\sqrt{u^2+v^2} - \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\Omega v \\ -g - f12m\pi\mathrm{R}^2\rho_{\mathrm{air}}\mathrm{C}_1v\sqrt{u^2+v^2} + \frac{1}{m}\rho_{\mathrm{air}}\mathrm{R}^3\mathrm{C}_2\Omega u \\ u \\ v \end{pmatrix}}$$

> Il est impératif d'exprimer toutes les variables en termes de u, v, x et y : en particulier, il ne doit rester ni α ni V dans l'expression finale, car on risquerait de les considérer comme constantes.

2 L'énoncé demande l'implémentation d'un schéma d'Euler explicite. Les conditions initiales doivent être définies dans la première colonne de `Y0`. Dans la boucle, on affecte à la colonne `i` le contenu de la colonne précédente, auquel on ajoute le produit de la dérivée de `Y` par l'intervalle de temps `T/N`.

```
def euler(T, N, F, Y0):
    Y = zeros((len(Y0), N))
    t = arange(0, T, T/N)
    Y[:, 0] = Y0    # Affectation des conditions
                    # initiales à la première colonne
    for i in range(1, N):
        # Une étape du schéma d'Euler explicite
        Y[:, i] = Y[:, i-1] + T/N * F(t[i-1], Y[:, i-1])
    return (t, Y)
```

III. Tracking et reconstruction de la trajectoire de la balle

3 L'expression `len(image1)` retourne la longueur de la première dimension de l'expression `image1`. L'énoncé donnant les dimensions du tableau (`n*m*3`), on a

$$\boxed{\texttt{len(image1) = 1024}}$$

De même, `image1[12]` est le tableau de dimension `m*3` situé à l'indice 12 de l'expression `image1`, et `image1[12][244]` est le tableau de dimension 3 correspondant aux données de couleur, situé à l'indice 244 de `image1[12]`. On a donc

$$\boxed{\texttt{len(image1[12][244]) = 3}}$$

Enfin, le tableau stocke des entiers, qui sont en Python du type `int`. Par suite,

$$\boxed{\texttt{image1[12][244][2]}\text{ est de type }\texttt{int}.}$$

Cet entier représente la couleur 2, le bleu, du pixel de coordonnées $(12, 224)$.

> Vu la question suivante, il y avait bien sûr une erreur dans le sujet : la question ne porte pas sur le type de `len(image1[12][244][2])` mais bien sur celui de `image1[12][244][2]`. Le type de ces deux expressions est ici le même, mais seul le second est utile pour déterminer la taille occupée en mémoire par le tableau.

4 L'ensemble des valeurs entre 0 et 255 peut être représenté sur huit bits, soit un octet. Au minimum, **la taille de ce tableau en mémoire vaut donc le nombre total d'éléments du tableau fois un octet, soit 2,4 Mo** (ou 19 Mb, ou 2,3 Mio, ou 18 Mib).

> Par défaut, Python est loin d'être aussi efficace que cela ; il n'est d'ailleurs pas clair, étant donnée la formulation de la question, de savoir ce que le concepteur du sujet attendait exactement. Python représente les petits entiers, jusqu'à 2^{60}, avec 32 octets. Cependant, il est difficile d'en tirer directement la taille du tableau complet, d'une part parce qu'il a besoin d'une mémoire non nulle pour représenter la structure de liste elle-même, d'autre part parce qu'il est capable d'effectuer des optimisations en « réutilisant » des objets, de sorte que `[10, 11]` est représenté sur 144 octets, alors que `[10, 10]` n'en nécessite que 112. En pratique, dans la mémoire de Python, l'image complète représente environ 82 Mo. On peut forcer une représentation efficace à l'aide de la bibliothèque `numpy` en utilisant le type `uint8`, prévu pour représenter des entiers positifs sur 8 bits : `np.array(image, dtype='uint8')`.
>
> Pour rappel, deux systèmes d'unités cohabitent pour désigner des tailles de données informatiques. On peut les baser tous les deux sur le bit, de symbole b, l'octet o, ou encore sur le byte B. Un bit est une unité d'information pouvant prendre seulement deux valeurs. Un octet est un ensemble de huit bits, et un byte est la plus petite quantité de bits qu'un programme peut lire ou écrire à la fois (généralement huit). Le premier système d'unités utilise les préfixes traditionnels du SI (k, M, G, etc.). Appliqués à l'octet noté o, on a conversions habituelles 10^3 o = 1000 o = 1 ko, 10^3 ko = 1 Mo, etc. Le second système, plus adapté en informatique avec un bit à deux valeurs, utilise des préfixes spécifiques aux puissances binaires : Ki (prononcé kibi), Mi (mébi), Gi (gibi), etc. Le facteur de conversion est alors une puissance de deux soit, avec l'octet, 2^{10} o = 1024 o = 1 Kio, 2^{10} Kio = 1 Mio, etc.

Centrale Maths 1 PSI 2018 — Corrigé

Ce corrigé est proposé par Corentin Fierobe (ENS Lyon) ; il a été relu par Angèle Niclas (ENS Lyon) et Benoit Chevalier (ENS Ulm).

Ce sujet propose d'étudier les matrices cycliques et les matrices de Toeplitz, puis les liens qui existent entre les deux. Les matrices cycliques, hors-programme, sont souvent abordées pendant la prépa car elles possèdent des propriétés fortes en lien avec la réduction. Les matrices de Toeplitz, moins connues, permettent des calculs plus simples dans la résolution de systèmes linéaires. Elles interviennent également dans certaines équations aux dérivées partielles. Un des axes du sujet consiste à montrer que toute matrice cyclique réelle est semblable à une matrice de Toeplitz.

- La partie I établit des résultats qui seront utiles dans la suite, ainsi que des résultats sur les matrices de Toeplitz : par exemple, que toute matrice de taille 2 sur $\mathbb{C}$ est semblable à une matrice de Toeplitz, que les matrices tridiagonales (qui sont de Toeplitz) sont diagonalisables, ainsi que le calcul de leurs éléments propres.
- La partie II aborde les matrices circulantes, qui font partie des matrices de Toeplitz. Elle étudie leur forme et cherche à montrer que toute matrice circulante est diagonalisable et que l'on peut en calculer les éléments propres.
- La partie III, indépendante de la précédente, aborde les matrices cycliques réelles (souvent appelées « matrices compagnons » par les étudiants).
  - III.A étudie les relations entre cyclicité et diagonalisabilité, montre que le commutant d'un endomorphisme cyclique f est réduit à l'algèbre qu'il engendre $\mathbb{C}[f]$, et introduit une matrice N importante pour la suite.
  - III.B définit les coefficients diagonaux d'ordre $k \in \mathbb{Z}$ d'une matrice A (ce sont les coefficients $a_{i,j}$ de A tels que $j - i = k$), puis étudie les matrices dont tous les coefficients sont nuls sauf éventuellement les coefficients diagonaux d'ordre k.
  - Enfin, III.C généralise une propriété de la partie I en prouvant que toute matrice cyclique réelle est semblable à une matrice de Toeplitz.

Le sujet nécessite de bons réflexes en réduction des endomorphismes (réduction des matrices en dimension 2, utilisation de critères classiques de diagonalisabilité en terme de polynômes annulateurs, et de ceux moins classiques concernant la dimension des sous-espaces propres). Les questions ne sont pas insurmontables et se résolvent assez rapidement, exceptées les dernières qui sont plus ardues. La principale difficulté du sujet réside dans sa longueur et dans le grand nombre de questions à traiter. Le sujet est de périmètre restreint, se cantonnant au monde des matrices, avec très peu d'utilisation des endomorphismes associés.

Indications

Partie I

1 Pour gagner du temps sur cette question : ne pas faire une démonstration à la main (point par point, en revenant aux définitions) mais exhiber une application linéaire.

2 Cette question nécessite deux récurrences.

5 On pourra se rappeler que toute matrice de $\mathscr{M}_n(\mathbb{C})$ est trigonalisable.

6 Partir d'une matrice de Toeplitz quelconque $\mathrm{T}(c, a, b)$ et essayer de déterminer a, b et c pour obtenir une matrice qui se diagonalise en

$$\begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$$

Alors $\mathrm{T}\left(\dfrac{\alpha-\beta}{2}, \dfrac{\alpha+\beta}{2}, \dfrac{\alpha-\beta}{2}\right)$ convient.

10 Utiliser $x_0 = x_{n+1} = 0$ pour obtenir deux relations faisant intervenir r_1 et r_2.

11 Factoriser le polynôme intervenant en (I.1). Essayer de calculer $\lambda - a$ à l'aide des relations obtenues et de la question 10 pour définir ℓ.

12 Se rappeler l'expression des x_k en fonction de r_1 et r_2, et utiliser la question 10.

13 Étudier le nombre de valeurs propres obtenues grâce aux questions précédentes.

Partie II

14 Pour les puissances de M_n, deux façons de traiter le problème sont envisageables : on pourra soit trouver une formule de récurrence et la prouver, soit raisonner en terme d'endomorphisme dans une base adaptée.

15 Étudier le polynôme annulateur de la question 14. On pourra chercher des vecteurs propres s'exprimant à l'aide de racines de l'unité (observer la question 16, qui peut donner une idée de leur forme).

18 Le polynôme bien choisi pourra être le polynôme annulateur de la question 14.

20 Utiliser la matrice Φ_n de la question 16 pour diagonaliser les matrices circulantes.

Partie III

22 Calculer les itérées de f_{M} appliquées en u dans la base de vecteurs propres, et chercher à quelle condition cela forme une famille libre.

24 Résultat classique, qui se prouve d'habitude en calculant par récurrence le polynôme caractéristique de la matrice. Mais ici, il suffit de trouver une relation entre toutes les équations données par le système $\mathrm{C}(a_0, \ldots, a_{n-1})\mathrm{X} = \lambda\mathrm{X}$.

32 Remarquons ici deux manières de procéder : par calcul direct, ou bien en se servant du résultat de la question 29.

37 Calculer $\phi(\mathrm{M}) - \mathrm{M}$ en remplaçant P et P^{-1} par leurs expressions en C.

38 Calculer $\phi(\mathrm{N}) - \mathrm{N} - \mathrm{NC} + \mathrm{CN}$ en remplaçant P et P^{-1} par leurs expressions en fonction de C.

39 Se servir des deux questions précédentes pour exprimer B en fonction de $\mathrm{A}, \mathrm{C}, \mathrm{N}$ et N' et d'une certaine matrice $\mathrm{T}' \in \mathrm{H}_{k+1}$.

40 On pourra remarquer que le calcul du noyau a été fait dans le cas complexe à la question 32.

41 Utiliser le résultat de la question 33 pour ne pas refaire des calculs.

42 S'inspirer du chapitre sur les adjoints d'endomorphismes sur un espace euclidien. En particulier on a la formule $\text{Ker}\, f^* = \text{Im}\, f^\perp$ pour f endomorphisme d'un espace euclidien. On pourra également se servir des questions 34 et 36.

43 Décomposer $\text{A}^{(k)}$ dans Δ_k selon une décomposition dont la forme est proposée à la question précédente. Utiliser ensuite le résultat de la question 39.

44 À l'aide d'une récurrence, trouver, pour une matrice cyclique donnée, une suite de matrices semblables les unes aux autres, qui ressemblent de plus en plus à une matrice de Toeplitz, c'est-à-dire dont le nombre de diagonales le long desquelles les coefficients sont égaux croît. On pourra se servir des questions 33, 36 et surtout 43.

I. Généralités et quelques exemples

$\boxed{1}$ Considérons l'application $\mathrm{T} : \mathbb{C}^{2n-1} \to \mathscr{M}_n(\mathbb{C})$ définie par l'énoncé, qui associe à un $(2n-1)$-uplet $(t_{-n+1}, \ldots, t_{n-1})$, la matrice de Toeplitz $\mathrm{T}(t_{-n+1}, \ldots, t_{n-1})$.

Par définition, $\mathrm{Toep}_n(\mathbb{C}) = \mathrm{Im}\,\mathrm{T}$. Pour montrer que c'est un sous-espace vectoriel de $\mathscr{M}_n(\mathbb{C})$, il suffit donc de prouver que T est une application linéaire.

Vérifions-le : si $t = (t_{-n+1}, \ldots, t_{n-1}) \in \mathbb{C}^{2n-1}$ et $t' = (t'_{-n+1}, \ldots, t'_{n-1}) \in \mathbb{C}^{2n-1}$, et si $\lambda \in \mathbb{C}$, alors

$$\mathrm{T}(t+\lambda t') = \mathrm{T}(t_{-n+1} + \lambda t'_{-n+1}, \ldots, t_{n-1} + \lambda t'_{n-1})$$

$$\mathrm{T}(t+\lambda t') = \begin{pmatrix} t_0 + \lambda t'_0 & \ldots & t_{n-1} + \lambda t'_{n-1} \\ \vdots & \ddots & \vdots \\ t_{-n+1} + \lambda t'_{-n+1} & \ldots & t_0 + \lambda t'_0 \end{pmatrix}$$

et en utilisant la définition de l'addition et de la multiplication par un scalaire dans l'espace des matrices, on aboutit à

$$\mathrm{T}(t+\lambda t') = \begin{pmatrix} t_0 & \cdots & t_{n-1} \\ \vdots & \ddots & \vdots \\ t_{-n+1} & \cdots & t_0 \end{pmatrix} + \lambda \begin{pmatrix} t'_0 & \cdots & t'_{n-1} \\ \vdots & \ddots & \vdots \\ t'_{-n+1} & \cdots & t'_0 \end{pmatrix}$$

$$\mathrm{T}(t+\lambda t') = \mathrm{T}(t_{-n+1}, \ldots, t_{n-1}) + \lambda \mathrm{T}(t'_{-n+1}, \ldots, t'_{n-1})$$

ce qui conclut la preuve de la linéarité. Montrons ensuite que T est injective. En effet, si $(t_{-n+1}, \ldots, t_{n-1}) \in \mathbb{C}^{2n-1}$ vérifie $\mathrm{T}(t_{-n+1}, \ldots, t_{n-1}) = 0$ alors matriciellement

$$\begin{pmatrix} t_0 & \cdots & t_{n-1} \\ \vdots & \ddots & \vdots \\ t_{-n+1} & \cdots & t_0 \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

et donc par définition des matrices de Toeplitz,

$$\forall (i,j) \in [\![\, 1\,;n \,]\!]^2 \qquad 0 = [\mathrm{T}(t_{-n+1}, \ldots, t_{n-1})]_{i,j} = t_{j-i}$$

soit $(t_{-n+1}, \ldots, t_{n-1}) = (0, \ldots, 0)$. On conclut alors que T réalise un isomorphisme entre $\mathbb{C}^{2n-1}$ et son image, $\mathrm{Toep}_n(\mathbb{C})$. En particulier,

$$\dim \mathrm{Toep}_n(\mathbb{C}) = \dim \mathbb{C}^{2n-1} = 2n-1$$

et une base est donnée par l'image par T d'une base de C^{2n-1}, par exemple l'image par T de la base canonique. Ainsi,

$\mathrm{Toep}_n(\mathbb{C})$ est un sous-espace vectoriel de $\mathscr{M}_n(\mathbb{C})$ de dimension $2n-1$, dont une base est donnée par les matrices

$$\mathrm{T}(1,0,\ldots,0) = \begin{pmatrix} & & \\ \vdots & (0) & \\ 0 & & \\ 1 & 0 & \cdots \end{pmatrix}, \quad \mathrm{T}(0,1,0,\ldots,0) = \begin{pmatrix} \vdots & & & \\ 0 & (0) & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \cdots \end{pmatrix}, \ldots,$$

$$\mathrm{T}(0,\ldots,0,1,0) = \begin{pmatrix} \cdots & 0 & 1 & 0 \\ & & 0 & 1 \\ & (0) & & 0 \\ & & & \vdots \end{pmatrix} \quad \text{et} \quad \mathrm{T}(0,\ldots,0,1) = \begin{pmatrix} \cdots & 0 & 1 \\ & & 0 \\ (0) & & \vdots \end{pmatrix}$$

Centrale Maths 2 PSI 2018 — Corrigé

Ce corrigé est proposé par Guillaume Batog (professeur en CPGE) ; il a été relu par Florian Metzger (docteur en mathématiques) et Sophie Rainero (professeur en CPGE).

Ce problème porte sur la notion de moment, qui est étudiée pour des variables aléatoires (I), des suites (II) et des fonctions réelles (III).

- La première partie porte sur les probabilités. Le moment $m_n(\mathrm{X})$ d'ordre $n \in \mathbb{N}$ de la variable aléatoire discrète réelle X est défini par $\mathrm{E}(\mathrm{X}^n)$, en cas d'existence.
  - La première moitié des questions est théorique et plutôt difficile pour une entrée en matière. Pour toute variable aléatoire X possédant des moments à tout ordre, on établit l'égalité
  $$\sum_{n=0}^{+\infty} m_n(\mathrm{X})\,\frac{t^n}{n!} = \mathrm{E}(\mathrm{e}^{\,t\mathrm{X}})$$
  où t appartient à un intervalle $]-\mathrm{R}\,;\mathrm{R}\,[$ sur lequel l'un des deux membres est bien défini. On utilise essentiellement les séries entières et un résultat hors-programme d'échange de deux sommes, énoncé en préambule.
  - La seconde moitié des questions exploite la fonction $t \mapsto \mathrm{E}(\mathrm{e}^{\,t\mathrm{X}})$, qui caractérise la loi de la variable X, afin de démontrer deux résultats d'approximation : la loi de Poisson comme « limite » d'une suite de lois binomiales d'une part ; la loi uniforme sur $[\,0\,;1\,]$ comme « limite » d'une suite de lois uniformes discrètes d'autre part.
- La deuxième partie étend la définition des moments aux suites, le but étant de construire une suite non identiquement nulle dont tous les moments sont nuls (ce qui n'est pas possible avec la suite des valeurs d'une variable aléatoire discrète). On démontre d'abord qu'une certaine fonction φ définie par morceaux est de classe $\mathscr{C}^\infty$ (avec le programme de Sup !), puis on la développe en série entière en 0 (encore avec le théorème en préambule). Enfin, on établit (avec des séries entières et des convergences normales) que la suite des coefficients obtenus répond au problème.
- La troisième partie prolonge le problème précédent au cas des fonctions réelles, où le moment d'ordre $p \in \mathbb{N}$ d'une fonction f est défini par l'intégrale sur $\mathbb{R}$ de la fonction $t \mapsto t^p\, f(t)$, en cas d'existence. À nouveau, on établit le caractère $\mathscr{C}^\infty$ d'une certaine fonction, cette fois-ci à valeurs complexes, définie avec l'exponentielle complexe. Ensuite, quelques changements de variables permettent d'exhiber une fonction de classe $\mathscr{C}^\infty$ sur $\mathbb{R}$ non nulle dont tous les moments sont nuls.

Finalement, ce problème est un sujet sur les séries entières et l'analyse locale de première année. Signalons de nombreux calculs de limites, qui nécessitent une bonne dextérité avec les « petits o », « grands O » et équivalents.

Indications

Partie I

2 Adapter la démonstration concernant le cas $n = 2$.

3 Autrement dit, exprimer $m_n(\mathrm{X})$ pour tout $n \in \mathbb{N}$ en fonction de $\mathrm{M_X}$ (et de ses dérivées en 0 d'après le cours).

4 En notant $\{x_i \mid i \in \mathbb{N}\}$ l'ensemble à l'intérieur duquel X prend ses valeurs, poser

$$\forall\,(n,i) \in \mathbb{N}^2 \qquad u_{n,i} = {x_i}^n\,\mathrm{P}(\mathrm{X} = x_i)\,\frac{t^n}{n!}$$

Utiliser le théorème du préambule où n et i jouent respectivement le rôle de p et q. Ne pas s'occuper des sommes W.

5 Reprendre la démonstration de la question 4 avec la même suite double en considérant d'abord la somme en i puis la somme en n.

6 À l'aide de la formule du binôme, justifier que $\mathrm{X} + \mathrm{Y}$ possède un moment à tout ordre $n \in \mathbb{N}$. Ensuite, établir la formule de la question en partant du produit $\mathrm{M_X}(t) \times \mathrm{M_Y}(t)$ et en utilisant les questions 4 et 5 pour retrouver $\mathrm{M_{X+Y}}(t)$.

7 Comparer la série du moment d'ordre $k \in \mathbb{N}$ à une série exponentielle.

8 Calculer plutôt $\mathrm{E}(\mathrm{e}^{t\mathrm{Z}})$.

9 Généraliser rapidement le résultat de la question 6 pour une somme de n variables aléatoires mutuellement indépendantes, puis calculer les moments d'une variable aléatoire de Bernoulli (attention au moment d'ordre 0 !)

Partie II

15 Pour la limite, se ramener à une expression de la forme $\mathrm{X}^3\mathrm{e}^{\mathrm{X}}$ pour $\mathrm{X} \to -\infty$. Pour le caractère $\mathscr{C}^1$ sur $\mathbb{R}$, utiliser le théorème de la limite de la dérivée.

16 Démontrer le résultat par récurrence sur $p \in \mathbb{N}^*$.

17 S'intéresser aux monômes de plus bas degré pour les polynômes P_p et Q_p de la question 16, afin de trouver un équivalent de $\varphi^{(p)}(x)$. Calculer la limite avec les mêmes ingrédients qu'à la question 15.

18 Démontrer par récurrence sur $p \in \mathbb{N}^*$ la propriété $\mathscr{Q}(p)$: « φ est de classe $\mathscr{C}^p$ et $\varphi^{(p)}(1) = 0$ ».

19 Utiliser le développement en série entière de l'exponentielle.

20 Écrire le développement en série entière de $(1+u)^\alpha$ en utilisant les polynômes de Hilbert de l'énoncé. Changer d'indice dans chaque coefficient.

21 Isoler le terme de rang $q = 0$ dans l'expression de φ puis décaler l'indice q.

22 Justifier que $\mathrm{H}_j\big((i-2)/2 + j\big)$ est positif puis calculer la double somme en remontant les calculs des questions 20 et 19.

23 Utiliser les paquets W_n $(n \in \mathbb{N})$ du théorème en préambule.

24 Revenir à la définition du rayon de convergence d'une série entière.

25 Se ramener aux résultats de dérivation de la somme d'une série entière.

26 Penser au théorème des bornes atteintes.

27 Utiliser un théorème d'intégration terme à terme pour calculer l'intégrale.

28 Isoler a_n dans l'égalité de la question 27. Remarquer que le polynôme en n est équivalent à n^p pour $n \to +\infty$.

29 Considérer l'inégalité de la question 28 pour $p = 2$. Pour calculer la somme, invoquer la continuité sur $[\,0\,;1\,]$ de la somme de la série entière $\sum a_n x^n$.

30 Raisonnements identiques à ceux de la question 29.

31 Démontrer par récurrence sur $p \in \mathbb{N}$ la propriété $\mathscr{R}(p)$: « la série $\sum n^k a_n$ converge et sa somme est nulle pour tout $k \in [\![\,0\,;p\,]\!]$. » Au cours de l'hérédité, utiliser la série de la question 30 avec $p+1$ au lieu de p, en décalant l'indice n pour manipuler a_n au lieu de a_{n+p+1} puis en développant le polynôme en n pour faire apparaître $n^{p+1} a_n$.

Partie III

33 Procéder comme à la question 16 pour le calcul des dérivées n-ièmes.

34 En notant $\alpha\,\mathrm{X}^a$ le monôme dominant de P_n, justifier que $\left|\theta^{(n)}(x)\right|$ est négligeable devant une expression de la forme $\mathrm{Y}^{a/2}\mathrm{e}^{-\mathrm{Y}}$ avec $\mathrm{Y} \to +\infty$.

35 Travailler avec les parties réelle et imaginaire de θ.

36 Pour la convergence de l'intégrale, comparer la fonction intégrée avec $t \mapsto \mathrm{e}^{-t^2/2}$, intégrable sur $\mathbb{R}$. Pour la valeur nulle de l'intégrale, considérer le changement de variable $u = t - p\pi$.

37 Appliquer soigneusement le théorème de changement de variable pour les intégrales généralisées.

38 Reconnaître $\mathrm{Im}\;\theta$ dans l'intégrale I_p.

I. Moments d'une variable aléatoire

1 Soient $x \geqslant 0$, $n \in \mathbb{N}^*$ et $k \in [\![1\,;n]\!]$.

- Si $x \in [\,0\,;1\,]$ alors $x^k \leqslant 1$.
- Si $x \in [\,1\,;+\infty\,[$ alors $x^k \leqslant x^n$ par comparaison des fonctions puissances.

Ainsi, $$\forall\, x \geqslant 0 \qquad 0 \leqslant x^k \leqslant \max(1, x^n) \leqslant 1 + x^n \qquad (1 \text{ et } x^n \text{ positifs})$$

Puisque la variable aléatoire X est à valeurs positives,

$$\boxed{\forall\, k \in [\![1\,;n]\!] \qquad 0 \leqslant \mathrm{X}^k \leqslant 1 + \mathrm{X}^n}$$

2 Supposons que X admet un moment d'ordre n ($n \in \mathbb{N}^*$) et qu'elle est à valeurs dans l'ensemble dénombrable $\{x_i \mid i \in \mathbb{N}\}$. Soit $k \in [\![1\,;n-1]\!]$. D'après le théorème de transfert, il suffit de montrer que la série $\sum\limits_i {x_i}^k\, \mathrm{P}(\mathrm{X} = x_i)$ converge pour établir que X^k admet une espérance.

1. Majoration du terme général : d'après la question 1 et comme toute probabilité est positive,
$$\forall\, i \in \mathbb{N} \qquad 0 \leqslant {x_i}^k\, \mathrm{P}(\mathrm{X} = x_i) \leqslant \mathrm{P}(\mathrm{X} = x_i) + {x_i}^n\, \mathrm{P}(\mathrm{X} = x_i)$$
2. Convergence de la série majorante : d'une part, la série $\sum \mathrm{P}(\mathrm{X} = x_i)$ converge car de somme 1 ; d'autre part, la série $\sum {x_i}^n\, \mathrm{P}(\mathrm{X} = x_i)$ converge puisque X^n admet une espérance par hypothèse. D'après la propriété de linéarité pour les séries convergentes, la série $\sum (1 + {x_i}^n)\, \mathrm{P}(\mathrm{X} = x_i)$ converge.

Par comparaison pour les séries à termes positifs, la série $\sum {x_i}^k\, \mathrm{P}(\mathrm{X} = x_i)$ converge.

$$\boxed{\text{Si } m_n(\mathrm{X}) \text{ existe alors } m_k(\mathrm{X}) \text{ existe pour tout } k \in [\![1\,;n-1]\!].}$$

> Le résultat suivant qui figure dans le programme de MP permet de simplifier la réponse à la question 2 : « si $|\mathrm{X}| \leqslant \mathrm{Y}$ et que Y admet une espérance, alors X admet une espérance. » Justement, la question 1 donne $\left|\mathrm{X}^k\right| \leqslant 1 + \mathrm{X}^n$ et la linéarité de l'espérance assure que $1 + \mathrm{X}^n$ admet bien une espérance lorsque X possède un moment d'ordre n.

3 D'après le cours, la somme $\mathrm{M_X}$ de la série entière $\sum \big(m_n(\mathrm{X})/n!\big)\, t^n$ est de classe $\mathscr{C}^\infty$ sur $]-\mathrm{R_X}\,;\mathrm{R_X}\,[$, où $\mathrm{R_X} > 0$ d'après l'énoncé. Toujours d'après le cours, les coefficients de la série entière sont ceux de la série de Taylor associée à la somme :

$$\forall\, n \in \mathbb{N} \qquad \frac{m_n(\mathrm{X})}{n!} = \frac{{\mathrm{M_X}}^{(n)}(0)}{n!}$$

d'où $$\boxed{\forall\, n \in \mathbb{N} \qquad m_n(\mathrm{X}) = {\mathrm{M_X}}^{(n)}(0)}$$

4 Soit $t \in\,]-\mathrm{R_X}\,;\mathrm{R_X}\,[$ fixé. Posons

$$\forall\, (n, i) \in \mathbb{N}^2 \qquad u_{n,i} = {x_i}^n\, \mathrm{P}(\mathrm{X} = x_i)\, \frac{t^n}{n!}$$

où X est à valeurs dans l'ensemble dénombrable $\{x_i \mid i \in \mathbb{N}\}$. Remarquons que

$$\mathrm{M_X}(t) = \sum_{n=0}^{+\infty} \left(\sum_{i=0}^{+\infty} {x_i}^n\, \mathrm{P}(\mathrm{X} = x_i) \right) \frac{t^n}{n!} = \sum_{n=0}^{+\infty} \sum_{i=0}^{+\infty} u_{n,i}$$

Centrale Informatique MP-PC-PSI 2018 — Corrigé

Ce corrigé est proposé par Cyril Ravat (professeur en CPGE) ; il a été relu par Jean-Julien Fleck (professeur en CPGE) et Guillaume Batog (professeur en CPGE).

Ce sujet d'informatique a pour objectif la simulation microscopique d'un gaz parfait. Le mouvement particulaire d'un tel gaz, abordé en physique en première année de prépa, permet de s'appuyer sur un modèle simple et accessible. L'étude est divisée en cinq parties de tailles sensiblement équivalentes.

- La première aborde l'initialisation de la simulation, c'est-à-dire le placement originel et aléatoire des particules dans l'espace. Si les premières questions, faciles, permettaient à tous les candidats d'engranger quelques points, les dernières sont beaucoup plus difficiles à cause d'un cahier des charges complexe ; elles donnaient l'occasion aux meilleurs candidats de se mettre en valeur. On notera une question 12 très difficile et relativement mal posée.
- On traite ensuite l'aspect physique du mouvement des particules, dans un modèle simplifié à une dimension. Les fonctions demandées sont, tout comme leur support physique, assez élémentaires.
- Dans la troisième partie, le problème de la gestion des différents événements pour un système de N particules est posé. Une solution intéressante est fournie et étudiée en détail, notamment au niveau de la complexité des algorithmes mis en jeu. Ces questions, intéressantes, demandent du temps et de l'attention pour être traitées correctement.
- La quatrième partie rassemble les résultats des questions précédentes pour établir les fonctions globales permettant la réalisation de la simulation. Elle demande d'avoir bien compris les différents modèles et outils utilisés précédemment et favorisait donc les candidats ayant pris le temps de réfléchir sur l'ensemble du sujet.
- Enfin, on exploite une base de données pour enregistrer une partie des informations liées à des simulations. Il s'agit de la partie la moins bien écrite du sujet. Les bases de données sont assez mal formées, très partielles pour les informations stockées et avec un formalisme éloigné des canons du domaine. Bien qu'il n'y ait que trois questions, elles sont répétitives.

Ce sujet est progressif et intéressant (la courte partie 5 exceptée). Le contexte utilisé est assez familier et la physique sous-jacente suffisamment simple pour ne pas poser de problème à la plupart des candidats. Les données sont représentées par des tableaux `numpy` car ils permettent l'addition et la multiplication par des scalaires. Il est donc nécessaire de savoir manipuler ces opérations. Il s'agit, pour les quatre premières parties, d'un bon entraînement accessible dès la fin de la première année. Les notions d'ingénierie numérique n'apparaissent pas, mais des questions associées à la représentation des nombres flottants sont présentes.

Indications

Partie I

1 Ne pas confondre la multiplication par un scalaire d'une liste Python ou d'un tableau `np.array`.

10 Il peut être profitable de commencer par un schéma d'une situation quelconque, sans hésiter à créer des positions proches ou éloignées les unes des autres. Puis dérouler l'algorithme en déplaçant les particules sur le schéma. Ne pas oublier de mettre en forme le retour comme une liste de tableaux `np.array`.

12 Le terme « histogramme » est très mal choisi. L'énoncé souhaite que l'on trace la fonction de densité de probabilité. Pour les cas $\mathrm{N} = 1$ et $\mathrm{N} = 5$, la réponse est intuitive. Mais pour le cas $\mathrm{N} = 2$, il faut multiplier des probabilités conditionnelles.

Partie II

17 Les tableaux `np.array` permettent l'utilisation de l'addition et de la multiplication de façon naturelle. La réponse attendue est très courte.

Partie III

20 Il s'agit bien d'une question en une dimension, ce qui simplifie largement l'étude. Trois cas sont possibles en fonction de la valeur précédente de la vitesse.

21 Comme à la question précédente, l'étude est en une dimension. Les deux particules ne se choqueront que si elles sont en train de se rapprocher, c'est-à-dire si leur distance diminue en valeur absolue.

22 Une fois trouvé l'endroit où insérer l'événement dans le catalogue, la méthode `insert`, dont l'énoncé donne l'aide en annexe, est incontournable.

23 La fonction demandée utilise les trois fonctions écrites précédemment. Les fonctions `tr` et `tc` ont un comportement similaire pour leur retour, notamment lorsqu'il n'y a pas d'événement associé. Ce retour est à tester avant la fabrication de l'événement.

26 Il faut bien compter toutes les complexités dans le pire des cas où chaque particule est capable de rencontrer toutes les autres. Le résultat peut être exprimé sous la forme d'une somme d'entiers consécutifs qui se simplifie.

27 Ne surtout pas aller chercher un algorithme trop compliqué ou éloigné du cours.

Partie IV

29 Mettre à jour les positions pour toutes les particules. Ensuite, ne mettre à jour que les vitesses des particules ayant interagi. Toutes les fonctions nécessaires ont déjà été écrites.

30 L'invalidation des événements doit se faire à la main, selon un test d'appartenance long à écrire, mais simple à concevoir.

31 On ne sait pas à l'avance combien d'itérations seront réalisées. À chaque itération, il faut commencer par supprimer les événements invalides en fin de catalogue.

33 L'erreur numérique associée à la représentation des valeurs flottantes est surtout proportionnelle à la valeur représentée.

Partie V

34 Il s'agit d'une requête d'agrégation selon un critère à définir.

35 Cette question est très proche de la précédente. La jointure n'est pas nécessaire.

36 La requête est plus longue à écrire, mais il suffit de traduire ce que dit l'énoncé.

I. Initialisation

> Pour la seconde fois, le concours Centrale-Supélec utilise la syntaxe des définitions de fonctions appelée « annotations ». Elle rend le code plus explicite en précisant les types des arguments et des retours des fonctions. Il est probable que cela devienne une habitude dans ce concours et c'est une bonne idée dont pourraient s'inspirer les autres.

1 La ligne 9 du code proposé crée un tableau `np.array` d'une unique valeur, générée aléatoirement entre 0 inclus et L exclu.

> Il fallait bien lire la documentation de la fonction `np.random.rand`, qui ne fait pas partie des fonctions à connaître.
>
> Attention à ne pas confondre le comportement de la multiplication sur les listes et sur les objets `np.array`. La multiplication d'une liste par un entier provoque une concaténation multiple de cette liste, comme il est rappelé dans l'annexe de l'énoncé, tandis que pour un `np.array` cela multiplie chaque élément du tableau par l'entier ou le flottant multiplicateur. Ce détail aurait pu être rappelé dans l'annexe.

2 L'argument `c` de la fonction `possible` peut être interprété à partir de l'appel à la fonction à la ligne 10. Il s'agit d'un tableau `np.array` à une dimension, contenant la position d'une nouvelle particule à placer parmi celles déjà présentes dans `res`.

3 La ligne 3 évacue les deux cas où le placement de la particule est impossible car trop près des bords de l'espace disponible. Elle conduit la fonction `possible` à renvoyer `False` s'il faut générer une nouvelle position.

4 Les lignes 4 et 5 testent pour chaque particule déjà présente si la nouvelle particule à insérer est trop proche. Elles ont la même conclusion que la ligne 3.

5 La fonction `possible` teste la possibilité de placement de la nouvelle particule. Elle renvoie `True` si c'est possible et `False` s'il faut générer une nouvelle position.

6 Le rejet réalisé à la ligne 3 peut être évité en générant une valeur comprise entre R et L − R, en remplaçant la ligne 9 par

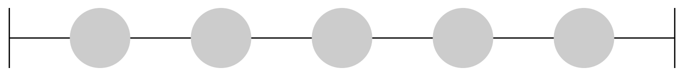
```
p = R + (L-2*R) * np.random.rand(1)
```

7 Dans la configuration proposée, chaque particule est espacée de 0,5 des deux particules adjacentes ou du bord. Il n'y a donc pas de place pour positionner une quatrième particule, et la boucle `while` devient une « boucle infinie ». Ainsi,

$$\boxed{\text{La suite de l'appel à }\texttt{placement1D}\text{ ne termine pas.}}$$

8 Dans le cas où $\mathrm{N} \ll \mathrm{N}_{\max}$, on peut supposer que presqu'aucun échec de placement de particule ne survient. Le contenu de la boucle `while` est alors répété N fois et contient des instructions de complexité constante ainsi qu'un appel à la fonction `possible`. Cette fonction est de complexité linéaire en le nombre d'éléments déjà dans `res`. Puisque $\mathrm{O}(1+2+3+\cdots+\mathrm{N}) = \mathrm{O}(\mathrm{N}^2)$,

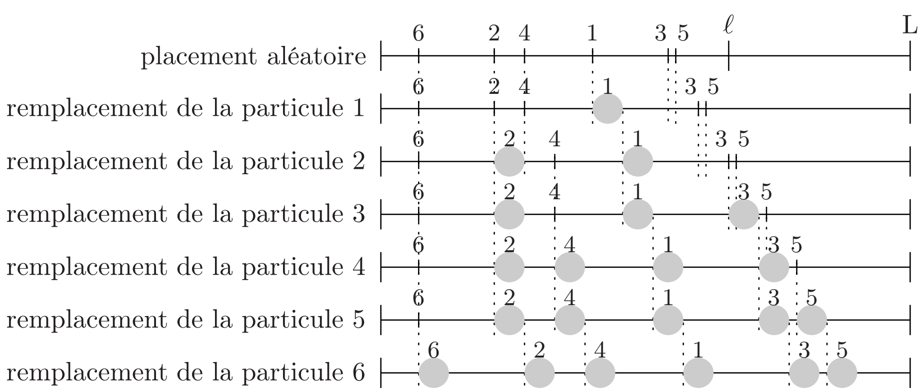
$$\boxed{\text{La complexité de la fonction }\texttt{placement}\text{ est quadratique.}}$$

> La somme des diamètres des particules à placer à la question 7 est pourtant inférieure à la longueur totale du segment. Cela signifie que le nombre N_{max} est certainement inférieur au rapport $L/(2\,R)$. Si l'on souhaite éviter la boucle infinie, il ne faut pas arriver pour autant à la situation où $N-1$ particules sont écartées l'une de l'autre et du bord d'une distance légèrement inférieure à $2\,R$. À la limite de ce raisonnement, la configuration est
>
> 
>
> Alors $$N_{max} = \left\lfloor \frac{L+2\,R}{4\,R} \right\rfloor$$

$\boxed{9}$ Pour recommencer à zéro le placement des particules dès qu'une nouvelle position est impossible, il suffit d'ajouter ce comportement lorsque `possible(p)` vaut `False`.

```
    res = []
    while len(res) < N:
        p = L * np.random.rand(1)
        if possible(p): res.append(p)
        else: res = []
    return res
```

$\boxed{10}$ L'énoncé demande une fonction en trois étapes. La troisième est une étape de déplacements successifs des particules, de l'espace $[\,0\,;\ell\,]$ vers l'espace $[\,0\,;L\,]$. Par exemple, ces déplacements peuvent être représentés par le schéma suivant :



Le placement rapide de N particules peut donc se faire ainsi :

```
def placement1Drapide(N, R, L):
    # Étape 1 : calcul de l'espace libre final
    l = L-2*R*N
    # Étape 2 : placement aléatoire des N particules sur [0; l]
    positions = l * np.random.rand(N)
    # Étape 3 : déplacement des particules
    for i in range(N):
        for j in range(N):  # Déplacement des particules à droite
            if positions[j] > positions[i]:
                positions[j] = positions[j]+2*R
        # Transformation en particule réelle
        positions[i] = positions[i]+R
```

Mines Maths 1 PSI 2018 — Corrigé

Ce corrigé est proposé par Hervé Diet (professeur agrégé) ; il a été relu par Théo Lenoir (ENS Ulm) et Florian Metzger (docteur en mathématiques).

Ce sujet porte sur l'espace de Rademacher, défini comme le sous-ensemble $\Omega_{q,n}$ de $\mathcal{M}_{q,n}(\mathbb{R})$ constitué des matrices à coefficients dans $\{-1, 1\}$. On prouve le théorème de Komlós qui s'énonce ainsi : si $\mathrm{M}^{(n)}$ est une variable aléatoire à valeurs dans $\Omega_{n,n}$ qui suit la loi uniforme, alors

$$\lim_{n\to+\infty} \mathrm{P}(\det(\mathrm{M}^{(n)}) = 0) = 0$$

En d'autres termes, une matrice aléatoire est inversible avec forte probabilité lorsque n est grand.

Le sujet fait largement appel aux matrices et aux probabilités. Il se compose de six parties connectées les unes aux autres.

- La première établit quelques majorations sur les coefficients binomiaux à l'aide de raisonnements classiques (équivalents, récurrences, etc.). Ces majorations permettront de calculer des limites dans la suite du sujet.
- La partie B étudie de manière exhaustive le cas de la dimension 2.
- La partie C donne quelques bornes sur les probabilités qui seront utilisées ultérieurement. Ces résultats sont établis en travaillant sur des espaces vectoriels.
- La partie D porte sur des aspects ensemblistes (cardinal, dénombrement) et sur les anti-chaînes qui sont les sous-ensembles de $\mathcal{P}(\{1, \ldots, n\})$ dont les éléments sont incomparables (ni $\mathrm{A} \subset \mathrm{B}$ ni $\mathrm{B} \subset \mathrm{A}$ si $\mathrm{A} \neq \mathrm{B}$). On démontre alors un théorème d'Erdös-Littlewood-Offord : si $v \in \mathbb{R}^n$ tel que $|v_j| \leqslant 1$ pour tout $j \in [\![1\,; n]\!]$, si J est un intervalle de $\mathbb{R}$ de longueur 2 et si n est assez grand,

  $$\mathrm{P}\left(\langle \mathrm{L}_1^{(n)}, v\rangle \in \mathrm{J}\right) \leqslant \frac{1}{\sqrt{n}}$$

  où $\mathrm{L}_1^{(n)}$ désigne la première ligne de la matrice aléatoire $\mathrm{M}^{(n)}$.
- Ce théorème est exploité dans la partie E. Cela permet de prouver des majorations nécessaires pour la suite.
- La dernière partie est consacrée à la démonstration du théorème de Komlós. On y exploite tous les résultats établis précédemment.

Le sujet est assez difficile pour la filière PC. Il comporte des raisonnements qui peuvent être déstabilisants et la manipulation des probabilités requiert beaucoup de rigueur. Comme le sujet suit une progression linéaire, il n'est pas judicieux de passer une partie puisqu'elle servira à coup sûr pour la suite. Cela en fait un sujet utile en fin de révision car il couvre nombre de thèmes d'algèbre au programme.

Indications

1 Étudier le rapport $\dbinom{n}{k+1} \Big/ \dbinom{n}{k}$. Étendre la propriété à $k > \lfloor n/2 \rfloor$ par symétrie des coefficients binomiaux.

2 Différencier les cas n pair et impair puis utiliser la formule de Stirling pour chercher un équivalent de $\dbinom{n}{\lfloor n/2 \rfloor}$.

3 Traiter le cas $k = 0$ à part. Faire un raisonnement par récurrence sur k.

4 Prouver que $\Omega_{1,n}$ engendre tous les éléments de la base canonique de $\mathbb{R}^n$.

5 Calculer l'espérance de $\mathrm{E}(\mathrm{M}_{1,1})$ puis utiliser les propriétés de l'espérance.

6 Appliquer la formule de transfert.

8 Commencer par déterminer les valeurs de

$$\mathrm{P}(\mathrm{M}_{i,j} = \mathrm{M}_{k,l}) \text{ et } \mathrm{P}(\mathrm{M}_{i,j} = -\mathrm{M}_{k,l}) \text{ avec } (i,j) \neq (k,l)$$

On pourra ensuite poursuivre les calculs et noter que $\det(\mathrm{M}^{(n)}) = 0$ lorsque les colonnes L_1 et L_2 sont liées.

9 Utiliser l'équivalence démontrée pour décomposer l'événement $\big\{\det(\mathrm{M}^{(n)}) = 0\big\}$.

10 Définir une base de $\mathcal{H}^\perp$ et la relier aux $\alpha_{i,j}$.

11 La méthode du pivot de Gauss donne les $n-d$ pivots, et les colonnes sans pivot donnent les indices $1 \leqslant j_1 < j_2 < \cdots < j_d \leqslant n$.

12 Attention à tourner la page du sujet pour voir les indications.

13 Exploiter la méthode d'orthogonalisation de Gram-Schmidt sans normaliser les vecteurs pour construire le vecteur voulu.

15 Étudier la restriction de σ à $\{1, \ldots, |\mathrm{A}|\}$.

17 Utiliser la question précédente pour décomposer l'ensemble $\mathcal{A}$.

19 Étudier la différence $\sum\limits_{j\in\mathrm{B}} v_j - \sum\limits_{j\in\mathrm{A}} v_j$ pour montrer qu'il reste au moins un élément et en déduire une minoration.

20 Prouver d'abord que $\mathcal{A}_\mathrm{J} = \{\mathrm{A} \in \mathcal{P}(\{1, \ldots, n\}) \mid s_\mathrm{A} \in \mathrm{J}\}$ est une anti-chaîne.

21 Écrire la négation de la définition proposée.

23 Trouver une majoration de la probabilité désirée qui ne dépende que de n. Étudier la limite du produit de ce terme et de n pour conclure.

24 Raisonner par l'absurde sur le nombre de coordonnées non nulles de v.

25 Choisir une suite d'indices $1 \leqslant j_1 < j_2 < \cdots < j_k \leqslant n$ tels que l'on puisse ramener le problème de $\Omega_{1,n}$ à $\Omega_{1,m}$ et appliquer le résultat de la question 20.

27 Scinder la somme obtenue en deux morceaux, de 1 à $n - t_n$ puis de $n - t_n + 1$ à $n - 1$ et majorer chaque partie. Choisir une suite (t_n) à croissance moins rapide que $(\sqrt{\ln n})$, par exemple $\big(\lfloor (\ln n)^{1/4} \rfloor\big)$.

A. Coefficients binomiaux

1 Soit $n \in \mathbb{N}^*$ et k un entier tel que $k+1 \leqslant \lfloor n/2 \rfloor$. Le coefficient binomial $\dbinom{n}{k}$ est non nul, ce qui permet de considérer

$$\frac{\dbinom{n}{k+1}}{\dbinom{n}{k}} = \frac{n!}{(k+1)!(n-k-1)!} \times \frac{k!(n-k)!}{n!}$$
$$= \frac{n-k}{k+1} = \frac{n+1}{k+1} - 1$$

La fonction $x \mapsto 1/x$ est décroissante sur $\mathbb{R}_+^*$ et $k+1$ est inférieur à $\lfloor n/2 \rfloor$ donc à $n/2$. On en déduit que

$$\frac{1}{k+1} \geqslant \frac{1}{n/2} = \frac{2}{n}$$

soit $$\frac{\dbinom{n}{k+1}}{\dbinom{n}{k}} \geqslant (n+1)\frac{2}{n} - 1 = 2 + \frac{2}{n} - 1 = 1 + \frac{2}{n} > 1$$

d'où $$\boxed{\text{L'application } k \mapsto \dbinom{n}{k} \text{ est croissante sur } \{0,\ldots,\lfloor n/2 \rfloor\}.}$$

D'après le résultat précédent, on a

$$\forall\, k \in \left\{0,\ldots,\left\lfloor \frac{n}{2} \right\rfloor\right\} \qquad \binom{n}{k} \leqslant \binom{n}{\lfloor n/2 \rfloor}$$

Il suffit alors de remarquer que $\dbinom{n}{k} = \dbinom{n}{n-k}$ donc, si $k > \lfloor n/2 \rfloor$ alors

$$k > \frac{n}{2} \quad\Longrightarrow\quad n-k < \frac{n}{2} \quad\Longrightarrow\quad n-k \leqslant \left\lfloor \frac{n}{2} \right\rfloor$$

car $n-k$ est entier. Par croissance de $k \mapsto \dbinom{n}{k}$ sur $[\![\, 0\,;\lfloor n/2 \rfloor\,]\!]$, on obtient

$$\binom{n}{k} = \binom{n}{n-k} \leqslant \binom{n}{\lfloor n/2 \rfloor} \qquad\qquad \text{si } k > \left\lfloor \frac{n}{2} \right\rfloor$$

La majoration est aussi vraie lorsque $k > \lfloor n/2 \rfloor$. En conclusion,

$$\boxed{\forall k \in \{0,\ldots,n\} \quad \binom{n}{k} \leqslant \binom{n}{\lfloor n/2 \rfloor}}$$

2 Soit $n \in \mathbb{N}$, on va faire l'étude selon la parité de n.

- Considérons le cas où n est pair. Soit $p \in \mathbb{N}$ tel que $n = 2p$. Alors

$$\binom{n}{\lfloor n/2 \rfloor} = \frac{(2p)!}{p!^2}$$

On peut maintenant utiliser la formule de Stirling et écrire

$$\binom{n}{\lfloor n/2\rfloor} \sim \frac{\sqrt{2\pi\times 2p}\left(\dfrac{2p}{\mathrm{e}}\right)^{2p}}{2\pi p\left(\dfrac{p}{\mathrm{e}}\right)^{2p}} = \frac{2^{2p}}{\sqrt{\pi p}}$$

et finalement, lorsque n est pair,

$$\binom{n}{\lfloor n/2\rfloor} \sim \sqrt{\frac{2}{\pi}} \times \frac{2^n}{\sqrt{n}}$$

- Voyons maintenant le cas n impair. Soit $p \in \mathbb{N}$ tel que $n = 2p+1$; alors

$$\left\lfloor \frac{n}{2} \right\rfloor = \left\lfloor p + \frac{1}{2} \right\rfloor = p \qquad \text{d'où} \qquad \binom{n}{\lfloor n/2\rfloor} = \binom{2p+1}{p} = \frac{(2p+1)!}{p!(p+1)!}$$

On peut astucieusement réutiliser le travail précédent en remarquant que

$$\frac{(2p+1)!}{p!(p+1)!} = \frac{2p+1}{p+1}\cdot\frac{(2p)!}{p!^2} \sim 2\cdot\frac{2^{2p}}{\sqrt{\pi p}} = \sqrt{\frac{2}{\pi}}\cdot\frac{2^{2p+1}}{\sqrt{2p}}$$

Ainsi, en remarquant que $\sqrt{2p} \sim \sqrt{2p+1}$, il vient à nouveau

$$\binom{n}{\lfloor n/2\rfloor} \sim \sqrt{\frac{2}{\pi}}\cdot\frac{2^{2p+1}}{\sqrt{2p+1}} = \sqrt{\frac{2}{\pi}} \times \frac{2^n}{\sqrt{n}}$$

En conclusion,

$$\boxed{\binom{n}{\lfloor n/2\rfloor} \underset{n\to+\infty}{\sim} \sqrt{\frac{2}{\pi}} \times \frac{2^n}{\sqrt{n}}.}$$

On a $\sqrt{\dfrac{2}{\pi}} < 1$ puisque $2 < \pi$. L'équivalent précédent implique alors que

$$\boxed{\exists n_0 \in \mathbb{N} \quad n \geqslant n_0 \qquad \Longrightarrow \qquad \binom{n}{\lfloor n/2\rfloor} \leqslant \frac{2^n}{\sqrt{n}}}$$

3 Soit $n \in \mathbb{N}$, la propriété attendue se démontre aisément pour $k = 0$ puisque

$$\binom{n}{0} 2^{-1} = 1 \times \frac{1}{2} = \frac{1}{2} < 1 = n^0$$

Montrons par récurrence sur $k \in \{1, \ldots, n\}$ que la propriété

$$\mathscr{P}(k) : \qquad \binom{n}{k} 2^{k-1} \leqslant n^k$$

est vraie pour tout $k \in \{1, \ldots, n\}$.

- $\underline{\mathscr{P}(1)}$ est vraie car $\dbinom{n}{1} 2^0 = n \times 1 = n = n^1$.
- $\underline{\mathscr{P}(k) \Longrightarrow \mathscr{P}(k+1)}$: supposons que pour tout $k \in [\![\, 1\,;n-1\,]\!]$ la propriété $\mathscr{P}(k)$ est vraie. Alors

$$\binom{n}{k+1} 2^k = \frac{n!}{(k+1)!(n-k-1)!} 2^k = \frac{n!}{k!(n-k)!} 2^{k-1} \times \frac{2(n-k)}{k+1}$$

Mines Maths 2 PSI 2018 — Corrigé

Ce corrigé est proposé par Romain Panis (ENS Ulm) ; il a été relu par Loïc Devilliers (ENS Cachan) et Florian Metzger (docteur en mathématiques).

Ce problème a pour objectif, dans la dernière partie, d'établir un résultat sur la vitesse de convergence dans le théorème central limite. Aucune connaissance sur ce théorème hors programme n'est requise. Pour progresser vers ce résultat, le sujet établit des résultats préliminaires qui utilisent l'analyse (intégration, continuité, dérivabilité,...) et les probabilités (indépendance, espérance,...). De nombreuses questions de cours sont glissées dans le sujet, ce qui permet d'insister sur l'importance de l'apprentissage du cours lors de la préparation du concours des Mines. Le sujet est assez technique et certaines questions (plus spécifiquement les questions 3, 8, 9, 17 et 21) demandent une rédaction précise.

- La première partie établit quelques résultats techniques qui seront utiles dans les parties suivantes.
- La deuxième partie s'intéresse à la transformation d'Ornstein-Uhlenbeck, qui à une fonction f continue sur $\mathbb{R}$ et à croissance lente associe, pour $t \geqslant 0$, l'application
$$\mathrm{P}_t f : x \in \mathbb{R} \longmapsto \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f\left(\mathrm{e}^{-t} x + \sqrt{1 - \mathrm{e}^{-2t}} y\right) \mathrm{e}^{-\frac{y^2}{2}} \,\mathrm{d}y$$
Cette partie repose sur les grands théorèmes du cours d'intégration. On utilise ainsi de nombreuses fois le théorème de convergence dominée et le théorème de continuité des intégrales à paramètre. Il faut vérifier soigneusement les conditions d'application de ces théorèmes ; l'obtention des dominations est parfois délicate (question 9).
- La troisième partie est dédiée à l'introduction de l'opérateur L qui à f de classe $\mathscr{C}^2$ sur $\mathbb{R}$ et à $x \in \mathbb{R}$ associe $\mathrm{L}f(x) = f''(x) - x f'(x)$. Cet opérateur servira en particulier à l'énoncé du théorème 1 dans la partie suivante.
- La quatrième et dernière partie concerne l'obtention d'une vitesse de convergence en $\mathrm{O}\left(1/\sqrt{n}\right)$ dans le théorème central limite pour le cas particulier de variables aléatoires admettant un moment d'ordre 3. Les outils probabilistes utilisés sont classiques (indépendance, calcul d'espérance de produit de variables aléatoires indépendantes,...).

En conclusion, le sujet est bien équilibré, de longueur raisonnable et reste tout à fait abordable sous réserve d'être à l'aise avec les intégrales à paramètre. La dernière partie, plus originale et assez technique, constitue un bon exemple d'énoncé mêlant analyse et probabilités, ce qui semble devenir un incontournable des concours Mines et Centrale.

Indications

Partie I

1. Utiliser le rappel de l'énoncé et effectuer un changement de variable affine.
2. Penser au critère de Riemann.
3. Effectuer le changement de variable proposé par l'énoncé, puis utiliser une formule trigonométrique pour se ramener à une intégrale plus simple.

Partie II

5. Utiliser le fait que $f \in \mathscr{C}_\ell(\mathbb{R})$ pour obtenir une majoration intégrable de la fonction étudiée.
6. Pour le calcul de limite, utiliser le théorème de convergence dominée et la caractérisation séquentielle de la limite.
8. Commencer par calculer la dérivée de ψ_y puis obtenir la majoration souhaitée en distinguant selon les cas. Pour obtenir la majoration dans le cas $|y| > 1$ on cherchera à étudier le terme dans l'exponentielle définissant ψ_y.
9. Pour la première partie de la question utiliser le résultat de 8 et appliquer le théorème de convergence dominée. Pour montrer la continuité de $(\mathrm{P}_t f)'$, utiliser le théorème de continuité des intégrales à paramètre en cherchant une domination intégrable sur un intervalle borné pour le paramètre.
10. Pour l'inégalité distinguer selon le cas $|a| \leqslant |b|$ et $|a| > |b|$.

Partie III

12. Commencer par justifier l'existence des deux intégrales puis penser à effectuer une intégration par parties.
13. Utiliser l'expression de la question 11, puis appliquer la question 12 à f' pour obtenir une expression de ${h_x}'(t_0)$ sans singularité en 0.

Partie IV

14. Utiliser l'écriture $|\mathrm{X}_1| = |\mathrm{X}_1|\mathbb{1}_{\{|\mathrm{X}_1|\leqslant 1\}} + |\mathrm{X}_1|\mathbb{1}_{\{|\mathrm{X}_1|>1\}}$.
17. Partir du terme de droite dans l'égalité et utiliser la question 16.
18. Appliquer le résultat de la question 1 deux fois avec, dans un premier temps $k = 0$, $f = g''$, $b = \mathrm{S}$ et $a = \mathrm{S}^{(i)}$; puis avec $k = 1$, $f = g'$, $b = \mathrm{S}$ et $a = \mathrm{S}^{(i)}$.
20. Partir du lemme 1 appliqué à f pour $k = 2$ puis utiliser le résultat de la question 9.
21. Il y a une erreur d'énoncé, dans le terme de gauche de l'inégalité il ne doit pas y avoir de sup. Utiliser le théorème I et les questions 19 et 20 pour obtenir une majoration du terme de gauche. Ensuite pour obtenir la majoration de l'énoncé utiliser la question 3. Admettre le fait que l'espérance commute avec l'intégrale.

I. Préliminaires

1 Soit f une fonction de classe $\mathscr{C}^{k+1}$ sur $[\,a\,;b\,]$. D'après la formule de Taylor avec reste intégral rappelée par l'énoncé, on a

$$f(b) = \sum_{j=0}^{k} \frac{f^{(j)}(a)}{j!}(b-a)^j + \frac{1}{k!}\int_a^b (b-t)^k f^{(k+1)}(t)\,\mathrm{d}t$$

et en effectuant le changement de variable affine non constant donc licite

$$t = a + \theta(b-a),$$

soit $\theta = \dfrac{t-a}{b-a}$ dans l'intégrale ci-dessus, on obtient

$$\int_a^b (b-t)^k f^{(k+1)}(t)\,\mathrm{d}t = (b-a)^{k+1}\int_0^1 (1-\theta)^k f^{(k+1)}(a+\theta(b-a))\,\mathrm{d}\theta$$

ce qui donne finalement la formule désirée

$$\boxed{f(b) = \sum_{j=0}^{k} \frac{f^{(j)}(a)}{j!}(b-a)^j + \frac{(b-a)^{k+1}}{k!}\int_0^1 (1-\theta)^k f^{(k+1)}(a+\theta(b-a))\,\mathrm{d}\theta}$$

2 Soient $\alpha > 0$ et $n \in \mathbb{N}$. Pour tout $x \in \mathbb{R}$ on a

$$|x^2 g_{n,\alpha}(x)| = |x^2 x^n \mathrm{e}^{-\alpha x^2}| = |x|^{n+2}\mathrm{e}^{-\alpha x^2}$$

Par croissances comparées on a

$$\boxed{|x|^{n+2}\mathrm{e}^{-\alpha x^2} \xrightarrow[|x|\to+\infty]{} 0}$$

La fonction $g_{n,\alpha}$ est continue sur $\mathbb{R}$ et vérifie par le calcul précédent

$$g_{n,\alpha}(x) \underset{|x|\to+\infty}{=} \mathrm{O}\left(\frac{1}{x^2}\right)$$

On en déduit alors par critère de Riemann, puisque $2 > 1$, le résultat recherché

$$\boxed{\text{La fonction } g_{n,\alpha} \text{ est intégrable sur } \mathbb{R}.}$$

3 Commençons par observer que l'intégrale proposée par l'énoncé existe bien. En effet, notons pour $t > 0$

$$h(t) = \frac{\mathrm{e}^{-3t}}{\sqrt{1-\mathrm{e}^{-2t}}}$$

La fonction h est continue sur $]\,0\,;+\infty\,[$ et de signe constant positif. De plus, elle est intégrable au voisinage de l'infini car équivalente à la fonction $t \longmapsto \mathrm{e}^{-3t}$ intégrable au voisinage de l'infini par critère de Riemann, puisque c'est un $\mathrm{O}\left(1/t^2\right)$. Enfin, au voisinage de 0 on a

$$h(t) \underset{t\to0^+}{\sim} \frac{1}{\sqrt{2t}} \underset{t\to0^+}{=} \mathrm{O}\left(\frac{1}{t^{1/2}}\right)$$

Donc par critère de Riemann, h est intégrable en 0 ($1/2 < 1$). Finalement, l'intégrale généralisée est bien définie.

On utilise maintenant l'indication de l'énoncé et on effectue le changement de variable

$$\theta = \text{Arcsin}\,(\mathrm{e}^{-t})$$

qui est bijectif de classe $\mathscr{C}^1$. En effet, $t \longmapsto \text{Arcsin}\,(\mathrm{e}^{-t})$ réalise une bijection $\mathscr{C}^1$ strictement décroissante de $]\,0\,;+\infty\,[$ dans $]\,0\,;\pi/2\,[$: cette fonction est la composée de $t \longmapsto \mathrm{e}^{-t}$, bijection décroissante de $]\,0\,;+\infty\,[$ dans $]\,0\,;1\,[$, avec Arcsin, bijection croissante de $]\,0\,;1\,[$ dans $]\,0\,;\pi/2\,[$. Cela donne $\sin(\theta) = \sin(\text{Arcsin}\,(\mathrm{e}^{-t})) = \mathrm{e}^{-t}$ car $\mathrm{e}^{-t} \in\,]-1\,;1\,[$ si $t > 0$, et

$$\cos(\theta)\,\mathrm{d}\theta = -\mathrm{e}^{-t}\,\mathrm{d}t = -\sin(\theta)\,\mathrm{d}t$$

On a donc, en utilisant le fait que cos est positif sur $[\,0\,;\pi/2\,]$

$$\begin{aligned}\int_0^{+\infty} \frac{\mathrm{e}^{-3t}}{\sqrt{1-\mathrm{e}^{-2t}}}\,\mathrm{d}t &= -\int_{\frac{\pi}{2}}^{0} \frac{\sin^3(\theta)}{\sqrt{1-\sin^2(\theta)}}\,\frac{\cos(\theta)}{\sin(\theta)}\,\mathrm{d}\theta \\ &= \int_0^{\frac{\pi}{2}} \frac{\sin^2(\theta)}{\cos(\theta)}\cos(\theta)\,\mathrm{d}\theta \\ \int_0^{+\infty} \frac{\mathrm{e}^{-3t}}{\sqrt{1-\mathrm{e}^{-2t}}}\,\mathrm{d}t &= \int_0^{\frac{\pi}{2}} \sin^2(\theta)\,\mathrm{d}\theta\end{aligned}$$

or
$$\int_0^{\frac{\pi}{2}} \sin^2(\theta)\,\mathrm{d}\theta = \int_0^{\frac{\pi}{2}} \frac{1-\cos(2\theta)}{2}\,\mathrm{d}\theta = \left[\frac{\theta}{2} - \frac{\sin(2\theta)}{4}\right]_{\theta=0}^{\theta=\frac{\pi}{2}} = \frac{\pi}{4}$$

d'où
$$\boxed{\int_0^{+\infty} \frac{\mathrm{e}^{-3t}}{\sqrt{1-\mathrm{e}^{-2t}}}\,\mathrm{d}t = \frac{\pi}{4}}$$

4 Soit $(m,n) \in \mathbb{N}^2$. On introduit

$$\varphi_{n,m} : \begin{cases} \mathbb{R} \longrightarrow \mathbb{R} \\ x \longmapsto \dfrac{(1+|x|^n)(1+|x|^m)}{1+|x|^{n+m}} \end{cases}$$

Cette application est continue sur $\mathbb{R}$ comme quotient d'applications continues sur $\mathbb{R}$ et dont le dénominateur ne s'annule pas. On dispose de l'équivalent suivant

$$\varphi_{n,m}(x) \underset{|x|\to+\infty}{\sim} \frac{|x|^n|x|^m}{|x|^{n+m}} = 1$$

On en déduit
$$\varphi_{n,m} \xrightarrow[|x|\to+\infty]{} 1$$

Par conséquent, $\boxed{\text{la fonction } \varphi_{n,m} \text{ est bornée sur } \mathbb{R}.}$

De plus, $\varphi_{n,m}$ est strictement positive sur $\mathbb{R}$ et par la remarque précédente, en notant $\mathrm{C}_{n,m} = \sup\limits_{x\in\mathbb{R}} \varphi_{n,m}(x)$, pour tout $x \in \mathbb{R}$,

$$0 < \frac{(1+|x|^n)(1+|x|^m)}{1+|x|^{n+m}} \leqslant \mathrm{C}_{n,m}$$

Mines Informatique MP-PC-PSI 2018 — Corrigé

Ce corrigé est proposé par Jean-Julien Fleck (professeur en CPGE) ; il a été relu par Virgile Andreani (ENS Ulm) et Guillaume Batog (professeur en CPGE).

Le sujet s'inspire d'un problème physique réel (mesures de houle en divers points du globe) et se découpe en cinq grandes parties qui visent chacune à balayer une partie du programme d'informatique commune.

- La première partie s'intéresse au stockage des données récoltées par les instruments de mesure. Elle permet de vérifier que les candidats ont des notions concernant la place occupée en mémoire par un caractère et sur la lecture d'un fichier de données.
- La deuxième partie permet de rentrer plus avant dans le traitement des données en demandant d'implémenter quelques algorithmes simples qui sont au programme de prépa : calcul d'une moyenne, intégration à partir d'une liste de valeurs, recherche d'une particularité dans une liste, calcul de propriétés à partir des spécifications de l'énoncé, etc.
- La troisième partie, après l'implémentation d'une recherche de maximum sur une liste de listes, s'intéresse plus particulièrement au programme de deuxième année avec une petite incursion dans les algorithmes de tri, demandant par exemple de compléter une implémentation préremplie du tri rapide, puis d'écrire entièrement une implémentation d'un tri par insertion. Une analyse de programme est aussi demandée pour y détecter un calcul inefficace et demander de l'améliorer.
- La quatrième partie concerne les bases de données. Il s'agit de survoler rapidement toute une page de présentation de la base de données considérée pour pouvoir écrire les trois requêtes SQL demandées. Là encore, il ne s'agit pas de comprendre *in extenso* tous les tenants et les aboutissants de la base, mais d'identifier les points clés afin de répondre rapidement à la question posée.
- La cinquième partie présente et demande d'implémenter l'algorithme de Cooley-Tukey de transformée de Fourier discrète en imposant une approche récursive (au programme de seconde année). C'est néanmoins tout à fait abordable en première année.

Le sujet est globalement équilibré. Il comporte de nombreuses questions simples de programmation et explore l'ensemble du programme des deux années (avec notamment des questions sur les tris et une implémentation récursive pour le programme de seconde année) mais peut être abordé en grande partie en première année.

Comme chaque année, l'épreuve des Mines nécessite d'être efficace vu sa durée limitée (seulement 1h30). Les questions ne sont pas particulièrement difficiles et il faut savoir aller à l'essentiel pour pouvoir toutes les traiter, quitte à en laisser certaines de côté si elles semblent demander trop de temps de réflexion.

Indications

Partie I

4 L'ouverture d'un descripteur de fichier se fait à l'aide de la commande `open`. La lecture des lignes dans un tableau se fait à l'aide de `readlines`. Ne pas oublier la conversion en flottant à l'aide de la fonction `float` qui gère naturellement les signes et les caractères de fin de ligne.

Partie II

7 La méthode des trapèzes sur un découpage $(x_i)_{i\in[\![\,0\,;\,n\,]\!]}$ en n intervalles revient à approximer

$$\int_{x_0}^{x_n} f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n-1} \left(\frac{f(x_i)+f(x_{i+1})}{2}\right) \times (x_{i+1}-x_i)$$

La moyenne d'une fonction η sur un intervalle de temps t_{tot} se définit comme

$$\langle \eta(t) \rangle = \frac{1}{t_{\text{tot}}} \int_0^{t_{\text{tot}}} \eta(t)\,\mathrm{d}t$$

8 Penser à calculer la moyenne en dehors de la boucle pour éviter de refaire de nombreuses fois le même calcul.

9 Le calcul de la moyenne est déjà de complexité linéaire : il faut que celle-ci soit fournie en argument de la fonction si on veut espérer atteindre une complexité en $\mathrm{O}(1)$ dans le meilleur des cas.

10 Attention, la définition des indices qu'il faut stocker dans la liste des successeurs n'est plus celle des questions 8 et 9 : il y a un décalage d'une position supplémentaire.

11 Penser à utiliser la fonction précédente pour calculer la liste des successeurs et utiliser le *slicing* pour découper en tranches correspondant aux vagues.

12 Utiliser de même la fonction définie à la question précédente pour obtenir les différentes vagues. Attention au fait que la hauteur maximale se calcule à cheval sur deux vagues (maximum de l'une auquel on soustrait le minimum de la suivante). Attention aussi au fait qu'il faut calculer le maximum précédant le premier PND.

Partie III

13 La structure particulière de la liste passée en argument impose d'implémenter la recherche de maximum à la main.

14 Ne pas essayer de comprendre ce que fait le code en détail, il suffit de trouver « par homogénéité » la forme attendue pour le pivot.

16 Penser à la manière dont on trie ses cartes en propageant la place libre jusqu'à atteindre le bon emplacement.

17 Recalculer de nombreuses fois la même chose ne sert à rien : penser à calculer la moyenne une unique fois avant de rentrer dans la boucle.

Partie IV

19 La deuxième requête peut se faire en comptant le nombre d'apparition d'une bouée dans la table `Tempete`, ledit nombre devant être égal à 0 si la bouée n'a pas connu de tempête. On peut aussi utiliser une différence ensembliste à l'aide de `EXCEPT` ou `MINUS`.

La troisième requête se fait assez simplement avec une jointure et une unique fonction d'agrégation.

Partie V

20 On peut estimer la complexité en procédant d'une manière proche du tri fusion : on a $k = \log_2(\mathrm{N})$ étages de divisions par 2 avec une complexité linéaire à chaque étage.

I. Stockage interne des données

1 Hormis la première ligne qu'on demande d'ignorer, chaque ligne est constituée de 8 caractères (5 chiffres, un signe, un point et le caractère de fin de ligne comme le signale l'énoncé) donc occupe 8 octets. 20 minutes correspondent à 1 200 s. En outre, comme il y a deux mesures par seconde, cela correspond à 2 400 mesures au total, soit une taille de $8 \times 2\,400 = 19\,200$ octets $= 19{,}2$ ko.

2 La campagne de mesure fait état d'un fichier récolté toutes les demi-heures pendant 15 jours. Il y a donc $15 \times 24 \times 2 = 720$ fois les informations précédentes collectées, ce qui représente donc une taille totale de 13 824 ko, soit 13,8 Mo.

$$\boxed{\text{Une carte mémoire de 1 Go est largement suffisante.}}$$

3 Ôter un chiffre revient à passer chaque ligne de 8 à seulement 7 octets, d'où une réduction de $1/8 \approx 12\%$ de la taille totale du fichier.

4 La lecture d'un fichier texte peut se faire de la manière suivante

```
with open('donnees.txt') as f:# Ouverture du descripteur de fichier
    L = f.readlines()         # Lecture effective du fichier
    liste_niveaux = []        # Définition du conteneur
    for i in range(1,len(L)): # On saute la première ligne de texte
        liste_niveaux.append(float(L[i])) # et on convertit en flottants
```

Bien sûr, on peut aussi demander à Numpy de faire tout le travail à notre place

```
import numpy as np # Importation de la bibliothèque
# On saute la première ligne et on convertit au vol en liste car np.loadtxt
# renvoie normalement un np.array
liste_niveaux = list(np.loadtxt('donnees.txt',skiprows=1))
```

> La lecture de données numériques dans un fichier est tellement utile en pratique (par exemple en TIPE) qu'il faut que ce soit maîtrisé par les candidats. Ici, l'usage de la fonction `float` n'est pas rappelé, mais on peut partir du principe qu'elle a été conçue de manière à ce que tous les nombres représentés de manière « usuelle » puissent se convertir facilement depuis une représentation en chaîne de caractères. En particulier, le `+` n'est pas gênant en début de nombre, ni le caractère de fin de ligne (ou tout autre espace qui pourrait être présent tout à droite ou tout à gauche de la chaîne de caractères). En revanche, `float` n'est pas capable d'effectuer des opérations de calcul comme « `float("2+2")` », il ne faut tout de même pas trop lui en demander.
>
> La construction avec `with` permet de s'assurer que le descripteur de fichier sera fermé quoi qu'il arrive et que le fichier ne soit pas corrompu (voir par exemple `https://tinyurl.com/with-open-python`), mais la structure suivante sera acceptable de la même manière.
>
> ```
> f = open('donnees.txt')     # Ouverture du descripteur de fichier
> L = f.readlines()           # Lecture effective du fichier
> liste_niveaux = []          # Définition du conteneur
> for i in range(1,len(L)):   # On saute la première ligne de texte
>     liste_niveaux.append(float(L[i])) # et on convertit en flottants
> f.close()                   # Fermeture du descripteur de fichier
> ```

X/ENS Maths PSI 2018 — Corrigé

Ce corrigé est proposé par Loïc Devilliers (ENS Cachan) ; il a été relu par Philippe Bouafia (professeur à l'ENSEA) et Nicolas Martin (professeur agrégé).

Ce sujet propose l'étude qualitative de l'équation différentielle linéaire d'ordre 2

$$-u''(x) + c(x)u(x) = f(x)$$

où c et f sont continues sur $[\,0\,;1\,]$. On impose que la solution u soit nulle en 0 et en 1, sans condition sur la valeur initiale de la dérivée. A priori, c'est donc un problème pour lequel l'existence et l'unicité ne peuvent être garanties, le théorème de Cauchy-Lipschitz ne s'appliquant pas. Plusieurs aspects sont abordés dans ce problème. On montre l'existence et l'unicité de la solution mais aussi son approximation par deux méthodes d'analyse numérique. La première méthode repose sur une discrétisation et fait intervenir de l'algèbre linéaire. La seconde utilise une famille de polynômes dits de Bernstein. Le sujet fait établir que les approximations obtenues par ces deux méthodes convergent bien vers la fonction recherchée. Les parties 2 et 5 sont indépendantes.

- Dans la première partie, on montre l'existence et l'unicité de la solution de l'équation différentielle donnée. On montre également que, si f est positive, la solution u l'est aussi.
- La deuxième partie ne comporte que de l'algèbre. On trouve les éléments propres d'une matrice donnée. On montre également que l'inverse de cette matrice est à coefficients positifs.
- Dans la troisième partie, on revient à l'équation différentielle et on étudie la discrétisation de l'équation différentielle à l'aide des résultats d'algèbre obtenus à la partie précédente.
- Les résultats des parties 1, 2 et 3 sont utilisés dans la quatrième partie pour montrer que l'approximation, décrite à la troisième partie, converge bien vers la solution lorsque le pas de la discrétisation tend vers zéro. La convergence est étudiée grâce à l'introduction d'une norme sur l'espace des matrices.
- Enfin, la cinquième partie propose une autre méthode d'approximation à l'aide de polynômes. De plus, certaines inégalités sont obtenues grâce aux probabilités.

Ce sujet exige donc un assez large spectre de compétences car il mélange l'étude des équations différentielles, l'algèbre linéaire, la topologie ainsi que les probabilités. Il contient plusieurs questions difficiles – réparties dans l'ensemble du sujet – qui demandent de prendre des initiatives. On peut regretter que les indications données par le sujet soient parfois ambiguës.

Indications

Partie 1

2 Poser au brouillon $w_2 = v_\lambda - \lambda w_1$ et trouver une équation satisfaite par w_2.

3 Raisonner par l'absurde en supposant que $w_1(1) = 0$. Considérer (s'il existe) le premier réel non nul où w_1 s'annule. Faire un dessin.

4 Trouver $\lambda \in \mathbb{R}$ tel que $v_\lambda(1) = 0$.

5 Supposer que u prend une valeur strictement négative. Faire également un dessin.

Partie 2

7 Raisonner par l'absurde et montrer que la famille $(|v_i|)_i$ est croissante.

9 Procéder par analyse-synthèse en utilisant les questions 8.a et 8.b.

10 Commencer par montrer que si X possède un coefficient strictement négatif alors BX aussi.

11 En diagonalisant A_n, on pourra montrer que les coefficients de $(\mathrm{A}_n + \varepsilon \mathrm{I}_n)^{-1}$ convergent vers ceux de ${\mathrm{A}_n}^{-1}$ lorsque ε tend vers zéro.

Partie 3

12 Appliquer la formule de Taylor avec reste intégral entre $x_i + h$ et x_i ainsi qu'entre $x_i - h$ et x_i.

13 Réécrire le système (**2**) sous forme matricielle et démontrer que la matrice obtenue est inversible en considérant un vecteur dans son noyau, procéder comme à la question 7.

15 Démontrer que l'inverse de la matrice du système est à coefficients positifs de la même manière qu'à la question 11.

Partie 4

16 Pour montrer que N(A) est égal au maximum de la somme des colonnes en valeur absolue, raisonner par double inégalité.

17.a Noter que $\mathrm{N}((h^{-2}\mathrm{A}_n)^{-1}) = \|(h^{-2}\mathrm{A}_n)^{-1}\mathrm{F}\|_\infty$ où $\mathrm{F} = {}^t(1, 1, \ldots, 1) \in \mathscr{M}_{n,1}(\mathbb{R})$.

17.b De même qu'à la question 17.a, calculer $(h^{-2}\mathrm{A}_n + \mathrm{D}_n)^{-1}\,\mathrm{F}$.

18 Calculer $\mathrm{Z} = (h^{-2}\mathrm{A}_n + \mathrm{D}_n)\,\mathrm{X}$, puis majorer sa norme à l'aide de la question 12.

Partie 5

21 Utiliser l'hypothèse sur f, la question 20 puis la question 19.b.

22 Dériver deux fois et regrouper les sommes obtenues en faisant apparaître le bon coefficient binomial dans chacune d'elles.

23.a Écrire $\chi_{n+1}'' = (f - \mathrm{B}_{n-1}f) + \left(\dfrac{1}{n+1}\mathrm{B}_{n-1}f\right) + \left(\dfrac{n}{n+1}\mathrm{B}_{n-1}f + (\widehat{\mathrm{B}}_{n+1}u)''\right)$, puis utiliser l'inégalité triangulaire.

23.b Trouver trois points où h s'annule et appliquer le théorème de Rolle à h puis à h'.

1. Existence et unicité des solutions de (1)

1 Le problème (**1bis**) est une équation différentielle linéaire d'ordre 2 sous forme résolue avec condition initiale sur la valeur de la fonction et de sa dérivée en 0. Les applications f et c étant continues sur $[\,0\,;1\,]$, le théorème de Cauchy-Lipschitz s'applique. Donc,

$$\boxed{\text{Le problème (\textbf{1bis}) admet une unique solution } v_\lambda \in \mathscr{C}^2([\,0\,;1\,],\mathbb{R}).}$$

2 Même si l'énoncé ne le demande pas explicitement, justifions rapidement l'existence et l'unicité de w_1 : il suffit d'appliquer le résultat de la question 1 avec $f = 0$ et $\lambda = 1$.

> Pour trouver w_2, posons d'abord au brouillon pour $\lambda \in \mathbb{R}$ la fonction $w_2 = v_\lambda - \lambda w_1 \in \mathscr{C}^2([\,0\,;1\,],\mathbb{R})$. Évaluons w_2 et w_2' en 0
>
> $$w_2(0) = v_\lambda(0) - \lambda w_1(0) = 0 \qquad \text{et} \qquad w_2'(0) = v_\lambda'(0) - \lambda w_1'(0) = 0$$
>
> De plus, calculons $-w_2'' + cw_2$, par linéarité de la dérivation
>
> $$-w_2'' + cw_2 = -v_\lambda'' + cv_\lambda - \lambda(-w_1'' + cw_1) = f$$
>
> Ainsi, w_2 est solution du problème (**1bis**) avec $w_2'(0) = 0$. Par unicité de v_0, démontrée à la question 1, on en déduit que $w_2 = v_0$. Une fois qu'on a trouvé w_2 au brouillon, rédigeons proprement la réponse.

Soit $\lambda \in \mathbb{R}$. Posons $w_2 = v_0$ qui ne dépend pas de λ. Définissons $v = \lambda w_1 + w_2$. Montrons que $v = v_\lambda$ en vérifiant que v est solution de (**1bis**).

- Comme w_1, w_2 appartiennent à l'espace vectoriel $\mathscr{C}^2([\,0\,;1\,],\mathbb{R})$, la fonction v appartient également à $\mathscr{C}^2([\,0\,;1\,],\mathbb{R})$;
- $v(0) = \lambda w_1(0) + w_2(0) = 0$;
- Par linéarité de la dérivation, $v'(0) = \lambda w_1'(0) + w_2'(0) = \lambda + 0$;
- Par linéarité de la dérivation, $-v'' + cv = -\lambda(w_1'' - cw_1) - (w_2'' - cw_2) = 0 + f$.

Ainsi, v est solution de (**1bis**). Par unicité de la solution au problème (**1bis**) démontrée à la question 1, $v_\lambda = v = \lambda w_1 + w_2$.

$$\boxed{\begin{array}{l}\text{Il existe une fonction } w_2 \in \mathscr{C}^2([\,0\,;1\,],\mathbb{R}) \text{ telle que pour} \\ \text{tout } \lambda \in \mathbb{R}, \text{ on ait } v_\lambda = \lambda w_1 + w_2. \text{ De plus, } w_2 = v_0.\end{array}}$$

3 Raisonnons par l'absurde : supposons que $w_1(1) = 0$. On va considérer, après avoir montré qu'il existe, le premier temps strictement positif où w_1 s'annule. Effectuons un développement limité à l'ordre 1 en 0 de $w_1 \in \mathscr{C}^1([\,0\,;1\,],\mathbb{R})$

$$w_1(x) = w_1(0) + xw_1'(0) + \mathrm{o}(x) = x + \mathrm{o}(x)$$

En particulier, il existe $\delta \in \,]\,0\,;1\,[$ tel que pour tout $x \in \,]\,0\,;\delta\,]$, $w_1(x) > 0$. Définissons l'ensemble suivant :

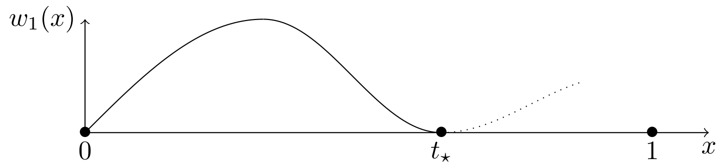
$$\mathrm{T} = \{t \in \,]\,0\,;1\,] \text{ tel que } w_1(t) = 0\}$$

Par hypothèse, on a $1 \in \mathrm{T}$, l'ensemble T étant non vide et minoré par 0, il admet une borne inférieure. Notons $t_\star = \inf \mathrm{T}$. Comme $\mathrm{T} \subset [\,\delta\,;1\,]$, on a $t_\star \geqslant \delta > 0$.

> Attention à ne pas dire directement « soit $t_\star$ le premier temps d'annulation de w_1 strictement positif », car a priori, il pourrait ne pas exister, penser

> à la fonction $f\colon x \longmapsto x\sin(x^{-1})$ prolongée en 0 par continuité. Sur cet exemple, on a $\inf\{t \in]\,0\,;1\,]\,, f(t) = 0\} = 0$.

Il existe $(t_n)_n$ une suite d'éléments de T convergeant vers $t_\star$. Pour tout $n \in \mathbb{N}$, on a $w_1(t_n) = 0$, par continuité de w_1, il en découle que $w_1(t_\star) = 0$. Sur l'intervalle $]\,0\,;t_\star\,[$, la fonction w_1 ne s'annule pas. Comme la fonction w_1 est continue sur $]\,0\,;t_\star\,[$, par la contraposée du théorème des valeurs intermédiaires, on en déduit qu'elle est soit strictement positive sur $]\,0\,;t_\star\,[$, soit strictement négative sur $]\,0\,;t_\star\,[$. En outre, elle est strictement positive sur $]\,0\,;\delta\,[$, par conséquent, elle est strictement positive sur $]\,0\,;t_\star\,[$.



Comme la fonction c est positive, on en déduit que

$$\forall x \in [\,0\,;t_\star\,] \qquad w_1''(x) = c(x)w_1(x) \geqslant 0$$

Par conséquent, w_1' est une fonction croissante sur $[\,0\,;t_\star\,]$. Ainsi,

$$\forall x \in [\,0\,;t_\star\,] \qquad w_1'(x) \geqslant w_1'(0) = 1$$

Par suite, w_1 est croissante sur $[\,0\,;t_\star\,]$. Dès lors, $0 = w_1(t_\star) \geqslant w_1(\delta) > 0$. Ce qui est une contradiction. En conclusion,

$$\boxed{w_1(1) \neq 0}$$

4 D'après la question précédente, $w_1(1) \neq 0$, posons alors $\mu = -w_2(1)/w_1(1) \in \mathbb{R}$ ainsi que $u = v_\mu$.

- la fonction u est une solution du problème (**1bis**) pour la valeur $\lambda = \mu$, en particulier, $-u'' + cu = f$;
- de même, $u(0) = 0$;
- d'après, la question 2, $u = \mu w_1 + w_2$. Ainsi, $u(1) = \mu w_1(1) + w_2(1) = 0$.

La fonction u est donc solution du problème (**1**).

Soit $\widetilde{u}$ une solution du problème (**1**). Posons $\nu = \widetilde{u}'(0) \in \mathbb{R}$, alors $\widetilde{u}$ est solution du problème (**1bis**) pour la valeur $\lambda = \nu$. Par unicité de v_ν, on a $\widetilde{u} = v_\nu$. D'après la question 2, $\widetilde{u} = \nu w_1 + w_2$. En évaluant cette égalité en 1, il vient $0 = \widetilde{u}(1) = \nu w_1(1) + w_2(1)$. Par conséquent, comme on a montré à la question 3 que $w_1(1) \neq 0$, on en déduit que $\nu = -w_2(1)/w_1(1) = \mu$. Ainsi $\widetilde{u} = \mu w_1 + w_2 = u$. Finalement,

$$\boxed{\text{Le problème } (\mathbf{1}) \text{ admet une et une seule solution.}}$$

5 Raisonnons par l'absurde et supposons que u soit strictement négative en un point $x_0 \in [\,0\,;1\,]$. Comme $u(0) = u(1) = 0$, on en déduit que $x_0 \in]\,0\,;1\,[$. On va considérer le plus grand intervalle contenant x_0 sur lequel la fonction u est strictement négative. Pour cela, définissons les deux ensembles suivants

$$\mathrm{A} = \{t \in [\,0\,;x_0\,] \quad u(t) \geqslant 0\} \qquad \text{et} \qquad \mathrm{B} = \{t \in [\,x_0\,;1\,] \quad u(t) \geqslant 0\}$$

X/ENS Informatique B MP-PC-PSI 2018 — Corrigé

Ce corrigé est proposé par Emma Kerinec (ENS Lyon) ; il a été relu par Cyril Ravat (professeur en CPGE) et Vincent Puyhaubert (professeur en CPGE).

Ce sujet regroupe une collection de requêtes SQL élémentaires et leur traduction en algèbre relationnelle, qu'il faut implémenter en Python. Aucune connaissance théorique spécifique n'est requise ; seule la maîtrise primaire de la syntaxe SQL est nécessaire. Le sujet est idéal pour s'entraîner sur des requêtes SQL simples, ainsi que sur les principes basiques des algorithmes et leur implémentation en Python. La première partie doit être abordée avant les deux suivantes, lesquelles peuvent en revanche être traitées indépendamment l'une de l'autre.

- La première partie introduit l'utilisation des listes et des tables afin de définir des fonctions servant de briques de base pour les parties suivantes, notamment la jointure de tables et la sélection d'éléments selon différents critères.
- La deuxième partie met en évidence le lien entre ces fonctions et les principales requêtes SQL ; il faut alors se servir de la partie précédente pour implémenter quelques requêtes SQL simples.
- La dernière partie propose d'améliorer les fonctions définies dans la première partie en exploitant des propriétés de tri concernant les tables ou grâce à l'introduction de dictionnaires.

Si ce sujet est d'une difficulté moyenne, il a sans doute déconcerté les candidats ayant fait l'impasse sur le langage SQL. Il introduit efficacement les concepts utilisés, notamment la structure hors-programme de dictionnaire, ainsi que tous les outils associés auxquels doivent se restreindre les candidats. Malgré l'évocation de tris dans la dernière partie, ce sujet est abordable en fin de première année.

Indications

Partie I

I.5 Observer que les enregistrements d'une table sont indépendants.

I.6 Parcourir les deux tables et utiliser la concaténation.

I.7 Parcourir les deux tables et comparer les bons attributs, ne pas oublier de supprimer l'attribut associé au second indice.

I.9 Comparer chaque enregistrement de la table à tous ceux déjà retenus dans le résultat.

Partie II

II.1 Utiliser la fonction `SelectionConstante` définie dans la partie précédente.

II.2 L'opération demandée est le produit cartésien de `Trajet` et `Véhicule`.

II.3 On peut enchaîner les fonctions élémentaires.

II.4 Attention aux indices des colonnes après jointure.

II.5 Il vaut mieux faire une double jointure suivie d'une unique sélection.

II.6 Penser à réutiliser le résultat précédent, sous forme de jointure.

Partie III

III.1 Parcourir la table en comparant les enregistrements consécutifs.

III.2 Quel est l'algorithme classique de recherche sur les listes triées ?

III.3 Parcourir en parallèle les deux tables et les synchroniser.

I. Implémentation des opérateurs de l'algèbre relationnelle en Python

I.1 On parcourt tous les enregistrements de `table`, et on ajoute dans le résultat ceux dont la valeur pour l'indice `indice` vaut `constante`.

```
def SelectionConstante(table, indice, constante):
    resultat = []
    for enr in table:
        if enr[indice] == constante:
            resultat.append(enr)
    return resultat
```

I.2 Dans la fonction précédente, il n'y a pas d'opération coûteuse en dehors de la boucle `for`. Celle-ci s'exécute `len(table)` fois et on effectue deux opérations élémentaires au plus à chaque étape. Ainsi,

La complexité de la fonction `SelectionConstante` est en $\mathrm{O}(\texttt{len(table)})$.

I.3 Parcourons tous les enregistrements de la table `table` et ajoutons dans le résultat ceux dont la valeur pour l'indice `indice1` est égale à celle pour l'indice `indice2`.

```
def SelectionEgalite(table, indice1, indice2):
    resultat = []
    for enr in table:
        if enr[indice1] == enr[indice2]:
            resultat.append(enr)
    return resultat
```

I.4 Il suffit de parcourir la liste `listeIndices` en créant un nouvel enregistrement auquel on ajoute les valeurs de `enregistrement` pour les indices souhaités, donnés dans `listeIndices`.

```
def ProjectionEnregistrement(enregistrement, listeIndices):
    resultat = []
    for indice in listeIndices:
        resultat.append(enregistrement[indice])
    return resultat
```

I.5 Il suffit d'appliquer la fonction précédente à tous les éléments de `table`, car ceux-ci sont indépendants, et de rassembler les nouveaux enregistrements dans une nouvelle table que l'on renvoie en fin de programme.

```
def ProjectionTable(table, listeIndices):
    resultat = []
    for enr in table:
        proj = ProjectionEnregistrement(enr, listeIndices)
        resultat.append(proj)
    return resultat
```

I.6 L'utilisation de deux boucles `for` permet de créer toutes les concaténations d'un élément de `table1` et d'un élément de `table2`, qui sont successivement ajoutés à une nouvelle table.

```
def ProduitCartesien(table1, table2):
    resultat = []
    for enr1 in table1:
        for enr2 in table2:
            resultat.append(enr1 + enr2)
    return resultat
```

I.7 Comme dans la question précédente, on parcourt les couples d'enregistrements à l'aide de deux boucles. Si les attributs de ces deux enregistrements coïncident, on concatène au premier enregistrement une copie du second dans laquelle seul l'attribut d'indice i_2 n'a pas été recopié. Le résultat de cette concaténation est ensuite ajouté dans la nouvelle table.

```
def JointEnregistrements(enr1, enr2, indice2):
    return (enr1 + enr2[0:indice2] + enr2[indice2 + 1:])
```

> L'utilisation d'une fonction annexe n'est pas obligatoire.

```
def Jointure(table1, table2, indice1, indice2):
    resultat = []
    for enr1 in table1:
        for enr2 in table2:
            if enr1[indice1] == enr2[indice2]:
                jointure = JointEnregistrements(enr1, enr2, indice2)
                resultat.append(jointure)
    return resultat
```

> On peut aussi utiliser une boucle `for` pour la jointure :
>
> ```
> def JointEnregistrements(enr1, enr2, indice2):
>     resultat = enr1[:]
>     for i in range(len(enr2)):
>         if i != indice2:
>             resultat.append(enr2[i])
>     return resultat
> ```
>
> L'astuce `enr1[:]` sert à créer une copie de `enr1` ; il ne faudrait pas que les modifications apportées à `resultat` modifient également l'argument `enr1`.

I.8 La fonction `Jointure` fait intervenir deux boucles. La première boucle `for` s'exécute `len(table1)` fois. Pour chaque étape, une nouvelle boucle `for` s'exécute `len(table2)` fois, en effectuant une lecture de liste et une copie d'un enregistrement de `table2`. Finalement,

La complexité de la fonction `Jointure` est en $\mathrm{O}(\texttt{len}(\texttt{table1}) \cdot \texttt{len}(\texttt{table2}) \cdot \texttt{len}(\texttt{table2}[0]))$.