# PC
## Mathématiques · Informatique
## 2017

Sous la coordination de

Julien Dumont
professeur en CPGE
ancien élève de l'École Normale Supérieure (Cachan)

Vincent Puyhaubert
professeur en CPGE
ancien élève de l'École Normale Supérieure (Cachan)

Par

Virgile Andreani
ENS Ulm

Josselin Giet
ENS Ulm

Quentin Guilmant
ENS Lyon

Émilie Liboz
professeur en CPGE

Thierry Limoges
ENS Cachan

Maël Mevel
professeur agrégé

Tristan Poullaouec
professeur en CPGE

Sophie Rainero
professeur en CPGE

Cyril Ravat
professeur en CPGE

# Sommaire

## Mines-Ponts



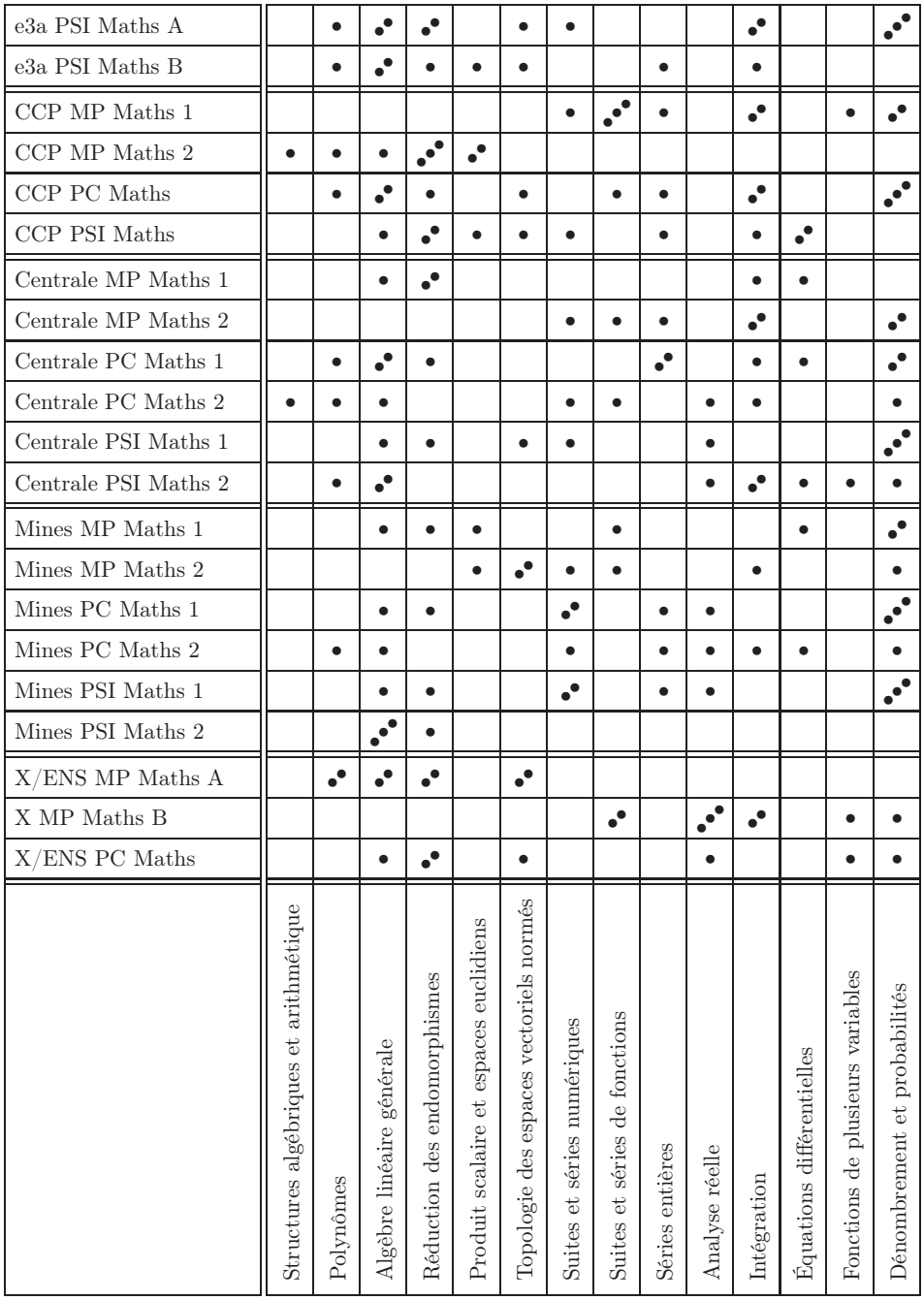
## Polytechnique-ENS



## Formulaires

# Sommaire thématique de mathématiques

## 2015 – 2017



**SESSION 2017** **PCMA002**

**CONCOURS COMMUNS POLYTECHNIQUES**

**EPREUVE SPECIFIQUE - FILIERE PC**

# MATHEMATIQUES

**Mardi 2 mai : 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**

**L'épreuve est constituée d'un problème en cinq parties largement indépendantes.**

Lorsqu'un raisonnement utilise un résultat obtenu précédemment dans le problème, il est demandé au candidat d'indiquer précisément le numéro de la question utilisée.

# PROBLÈME

Soit $p \in ]0,1[$. On pose $q = 1 - p$.

On considère un automate qui génère successivement les lettres C ou P jusqu'à obtenir une certaine séquence prédéfinie.

On suppose que pour tout $n \in \mathbb{N}^*$, l'automate génère la $n$-ième lettre à l'instant $n$ de façon indépendante de toutes les générations précédentes. On suppose également qu'à chaque génération, les lettres P et C ont des probabilités $p$ et $q$ (respectivement) d'être générées. Suivant les parties considérées, on définit différents niveaux que l'automate peut atteindre.

On considère dans tous les cas que l'automate est initialement au niveau 0. On se propose alors d'étudier essentiellement l'existence de l'espérance et de la variance de la variable aléatoire correspondant au temps d'attente de la séquence prédéfinie à travers sa série génératrice.

Pour cette étude probabiliste, on mobilise diverses propriétés analytiques (surtout sur les séries entières) et quelques propriétés d'algèbre linéaire.

Dans les parties **I**, **II** et **V**, on examine le temps d'attente pour les séquences C puis CC, puis CPC et CCPPC. La partie **II** est indépendante de la partie **I** et traite de questions préliminaires sur les séries entières qui seront investies dans les parties **III** et **V**. La partie **IV** est indépendante des parties précédentes et traite les questions préliminaires d'algèbre linéaire qui servent exclusivement dans la partie **V**. La partie **III** ne dépend de la partie **I** que par la question **Q4** et de la partie **II** que par la question **Q10**. La partie **V** utilise seulement la question **Q11** de la partie **II** et la partie **IV**.

Pour $n \in \mathbb{N}^*$, on note $P_n$ l'évènement « l'automate génère la lettre P à l'instant $n$ » et $C_n$ l'évènement « l'automate génère la lettre C à l'instant $n$ ».

## Partie I - Étude d'un cas simple

Dans cette partie, on dit que l'automate passe du niveau 0 au niveau 1 dès qu'il génère la lettre C. Si, en revanche, il génère la lettre P, alors il reste au niveau 0. L'expérience s'arrête dès que l'automate a atteint le niveau 1. On résume l'expérience par la figure 1 suivante :

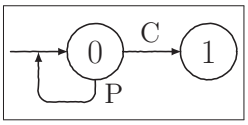



**Figure 1**

On note $Y$ l'instant où, pour la première fois, l'automate atteint le niveau 1. On admet que $Y$ est une variable aléatoire définie sur un espace probabilisé $(\Omega, \mathscr{A}, \mathbf{P})$ telle que $Y(\Omega) \subset \mathbb{N}^*$. On note $G_Y$ la série génératrice de $Y$ et $R_Y$ son rayon de convergence.

On sait alors que $R_Y \geqslant 1$ et que :

$$\forall t \in ]-R_Y, R_Y[, \quad G_Y(t) = \mathbf{E}(t^Y) = \sum_{n=1}^{+\infty} \mathbf{P}(Y = n) t^n.$$

**Q1.** Reconnaître la loi de $Y$ et préciser en particulier $\mathbf{P}(Y=n)$ pour $n \in \mathbb{N}^*$.

**Q2.** Montrer que $R_Y = \dfrac{1}{p} > 1$ et que : $\forall t \in \left]-\dfrac{1}{p}, \dfrac{1}{p}\right[, \quad G_Y(t) = \dfrac{qt}{1-pt}$.

**Q3.** Montrer que $G_Y$ est 2 fois dérivable en 1 et que $G_Y'(1) = \dfrac{1}{q}$ et $G_Y''(1) = \dfrac{2p}{q^2}$.

**Q4.** Donner les valeurs de $\mathbf{E}(Y)$ et de $\mathbf{V}(Y)$.

## Partie II - Séries entières

Soit $z \in \mathbb{C}$ et $a \in \mathbb{C}^*$. Pour $n \in \mathbb{N}$, on pose $u_n(a) = -\dfrac{1}{a^{n+1}}$.

**Q5.** Montrer que $\sum u_n(a) z^n$ est une série entière de rayon de convergence égal à $|a|$.

**Q6.** Montrer que si $|z| < |a|$, on a : $\dfrac{1}{z-a} = \displaystyle\sum_{n=0}^{+\infty} u_n(a) z^n$.

Soit $a$, $b$ et $\lambda$ des nombres complexes non nuls. Dans les questions **Q7** à **Q10**, on suppose que $|a| < |b|$. On définit alors, pour tout $n \in \mathbb{N}$, $v_n = \displaystyle\sum_{k=0}^{n} u_k(a) u_{n-k}(b)$ et pour tout réel $t$ tel que $|t| < |a|$, $f(t) = \dfrac{\lambda t^2}{(t-a)(t-b)}$.

**Q7.** Montrer que l'on a :

$$v_n = \frac{1}{ab^{n+1}} \sum_{k=0}^{n} \left(\frac{b}{a}\right)^k = \frac{1}{b-a}\left(\frac{1}{a^{n+1}} - \frac{1}{b^{n+1}}\right).$$

**Q8.** Trouver un équivalent simple de $v_n$ quand $n$ tend vers $+\infty$.

**Q9.** En déduire que le rayon de convergence de $\sum v_n z^n$ est égal à $|a|$ et que si $|z| < |a|$, alors

$$\frac{1}{(z-a)(z-b)} = \sum_{n=0}^{+\infty} v_n z^n.$$

**Q10.** Justifier que $f$ est développable en série entière au voisinage de 0 et que la série entière qui lui est associée possède un rayon de convergence $R_f$ tel que $R_f = |a|$.

Soit $a$, $b$, $c$ et $\lambda$ des nombres complexes non nuls. On suppose que : $|a| \leqslant |b| \leqslant |c|$.

Pour tout réel $t$ tel que $|t| < |a|$, on pose : $g(t) = \dfrac{\lambda t^3}{(t-a)(t-b)(t-c)}$.

**Q11.** Justifier que $g$ est développable en série entière au voisinage de 0 et que la série entière qui lui est associée possède un rayon de convergence $R_g$ tel que $R_g \geqslant |a|$.

# CCP Maths PC 2017 — Corrigé

Ce corrigé est proposé par Thierry Limoges (ENS Cachan) ; il a été relu par Maël Mevel (professeur agrégé) et Benjamin Monmege (enseignant-chercheur à l'université).

---

Comme dans un jeu de Pile ou Face, on génère aléatoirement une suite de lettres qui peuvent être P ou C. Les tirages sont identiques et indépendants. Un mot écrit avec les lettres P et C étant fixé, on cherche le temps au bout duquel il apparaîtra dans la suite.

Le problème est modélisé par un automate probabiliste. Une variable aléatoire compte le nombre de lettres générées jusqu'à ce que l'automate soit dans son état final, qui correspond à la première apparition du mot voulu. On étudie la loi de cette variable aléatoire à travers sa série génératrice, et on calcule notamment son espérance.

- Dans la partie I, on étudie le cas où le mot voulu est simplement C. Cette partie redémontre le cours sur la loi géométrique.
- La partie II est également élémentaire. Elle porte sur les séries géométriques et leur produit de Cauchy.
- Dans la partie III, on étudie le temps d'attente pour obtenir deux C consécutifs. La méthode repose sur les probabilités conditionnelles et les fonctions polynomiales de degré 2.
- La partie IV établit une méthode de calcul de la série génératrice du temps d'attente à partir de la matrice de transition de l'automate. On utilise le caractère multilinéaire alterné du déterminant, en dimension 4, à travers des polynômes caractéristiques.
- Dans la partie V, on applique la partie IV pour calculer le temps d'attente de la séquence CPC. La dernière question propose d'étendre la méthode pour calculer le temps d'attente de la séquence CCPPC. On s'amusera au passage du fait que le choix inhabituel des lettres P et C au lieu des lettres P (pile) et F (face) s'explique donc essentiellement par un clin d'œil au concours CCP, filière PC.

Ce sujet comporte la bagatelle de 46 questions réparties en 5 parties, mais sa difficulté est modérée et sa longueur se révèle donc raisonnable. Peu de questions nécessitent une prise d'initiative. Il illustre et éclaire les cours sur la loi géométrique et les séries génératrices ; il peut être abordé dès que ces chapitres ont été étudiés.

# Indications

## Partie I

1 Décrire l'évènement $(Y = n)$ à l'aide des $P_k$ et $C_k$.

2 Reconnaître une série géométrique.

4 Rappeler pourquoi $E(Y) = G_Y'(1)$ et $E(Y(Y-1)) = G_Y''(1)$ en écrivant les sommes, puis en déduire $V(Y) = G_Y''(1) + G_Y'(1) - \left[G_Y'(1)\right]^2$.

## Partie II

5 Appliquer la règle de D'Alembert.

8 Montrer que $\dfrac{1}{b^{n+1}} = \underset{n\to+\infty}{o}\left(\dfrac{1}{a^{n+1}}\right)$.

9 Si $|u_n| \underset{n\to+\infty}{\sim} |v_n|$, alors $\sum u_n z^n$ et $\sum v_n z^n$ ont même rayon de convergence.

10 Donner explicitement un développement en série entière de $f$ au voisinage de 0, invoquer l'unicité et étudier son rayon.

## Partie III

14 Appliquer la formule des probabilités totales à $P(Z = n)$ et au système complet d'évènements de la question 13. Remarquer, en particulier, que

$$P(Z = n|P_1) = P(Z = n-1) = p_{n-1}$$

18 Appliquer l'inégalité triangulaire à $\sqrt{\Delta}$ et $-p$. Déterminer le signe de $Q(t)$ entre ses racines et utiliser la question 16.

19 Diviser par $Q(t)$ au voisinage de 0 et invoquer l'unicité du développement en série entière.

21 Montrer l'existence de $G_Z'(1)$ et $G_Z''(1)$.

## Partie IV

25 Écrire la matrice $A - tI_4$ et développer son déterminant, de préférence par rapport à une colonne ou une ligne comportant beaucoup de 0.

27 Utiliser la multilinéarité du déterminant.

29 Montrer que $I_4 - tA$ est inversible pour $t$ au voisinage de 0.

31 Utiliser la multilinéarité et le caractère alterné du déterminant.

32 Calculer explicitement $\det_{\mathscr{B}}(U_1, U_2, U_3, L)$.

34 Écrire que $(x_1, x_2, x_3, x_4)$ est un vecteur propre de ${}^tA$ pour la valeur propre $\lambda$.

35 Diviser par M une égalité bien choisie du système $\mathscr{H}$ pour chaque cas.

37 Utiliser la question 27.

## Partie V

40 Pour tout $k \in [\![\, 0\,;3\,]\!]$, étudier les états possibles au temps $n-1$ pour se retrouver à l'état $k$ au temps $n$.

41 Revenir à la définition des $S_i(t)$ et utiliser la question 40.

43 Utiliser la question 32 en ayant préalablement vérifié ses hypothèses, puis la question 11.

45 Calculer $G_T'(1)$ et remplacer les « $p$ » par « $1-q$ ».

# I. Étude d'un cas simple

**1** Soit $n \in \mathbb{N}^*$. Par construction, l'évènement $(Y = n)$ est

$$\bigcap_{k=1}^{n-1} P_k \cap C_n$$

C'est une intersection d'événements mutuellement indépendants, sa probabilité est le produit

$$P(Y = n) = \prod_{k=1}^{n-1} P(P_k)P(C_n) = p^{n-1}(1-p) = (1-q)^{n-1}q$$

Ainsi, La variable aléatoire Y suit une loi géométrique de paramètre $q$.

**2** Soit $t \in \mathbb{R}$. Le terme général de la série $\sum P(Y = n)t^n$ est

$$(1-q)^{n-1}qt^n = qt((1-q)t)^{n-1} = qt(pt)^{n-1}$$

C'est le terme général d'une série géométrique de raison $pt$. D'une part, si $|t| < 1/p$, on a $|pt| < 1$, et alors cette série converge. D'autre part, si $|t| > 1/p$, on a $|pt| > 1$, et alors cette série diverge grossièrement. Le rayon de $G_Y$ est donc $1/p$, et $R_Y = 1/p > 1$ car $0 < p < 1$. Calculons sa somme. Soit $t \in ]-R_Y\,;R_Y[$. Alors

$$G_Y(t) = \sum_{n=1}^{+\infty} qt(pt)^{n-1} = qt\sum_{n=0}^{+\infty}(pt)^n = \frac{qt}{1-pt}$$

La fonction génératrice $G_Y$ admet pour rayon $R_Y = 1/p > 1$ et pour tout $t \in ]-R_Y\,;R_Y[$,

$$G_Y(t) = \frac{qt}{1-pt}$$

**3** Le rayon de la série entière $G_Y$ est $R_Y = 1/p > 1$, ainsi $G_Y$ est de classe $\mathscr{C}^\infty$ sur $]-1/p\,;1/p[$. Cet intervalle ouvert contient 1, donc $G_Y$ est deux fois dérivable en 1. Pour tout $t \in ]-R_Y\,;R_Y[$,

$$G_Y'(t) = \frac{q(1-pt) - qt(-p)}{(1-pt)^2} = \frac{q}{(1-pt)^2}$$

et

$$G_Y''(t) = \frac{-2q(-p)}{(1-pt)^3} = \frac{2pq}{(1-pt)^3}$$

D'où $$G_Y'(1) = \frac{q}{(1-p)^2} = \frac{1}{q} \quad \text{et} \quad G_Y''(1) = \frac{2pq}{(1-p)^3} = \frac{2p}{q^2}$$

En conclusion, $$\boxed{G_Y'(1) = \frac{1}{q} \quad \text{et} \quad G_Y''(1) = \frac{2p}{q^2}}$$

**4** On a $$E(Y) = \sum_{n=1}^{+\infty} nP(Y = n)1^n = G_Y'(1) = \frac{1}{q}$$

De plus, $$E(Y(Y-1)) = \sum_{n=1}^{+\infty} n(n-1)P(Y = n)1^n = G_Y''(1)$$

Par linéarité de l'espérance, il vient

$$\begin{aligned}
\mathrm{V}(\mathrm{Y}) &= \mathrm{E}(\mathrm{Y}^2) - \mathrm{E}(\mathrm{Y})^2 \\
&= \mathrm{E}(\mathrm{Y}^2 - \mathrm{Y}) + \mathrm{E}(\mathrm{Y}) - \mathrm{E}(\mathrm{Y})^2 \\
&= \mathrm{E}(\mathrm{Y}(\mathrm{Y}-1)) + \mathrm{E}(\mathrm{Y}) - \mathrm{E}(\mathrm{Y})^2 \\
&= \mathrm{G_Y}''(1) + \mathrm{G_Y}'(1) - \left[\mathrm{G_Y}'(1)\right]^2 \\
&= \frac{2p}{q^2} + \frac{1}{q} - \frac{1}{q^2} \\
&= \frac{2p+q-1}{q^2} \\
\mathrm{V}(\mathrm{Y}) &= \frac{p}{q^2}
\end{aligned}$$

Finalement, $\boxed{\mathrm{E}(\mathrm{Y}) = \frac{1}{q} \quad \text{et} \quad \mathrm{V}(\mathrm{Y}) = \frac{p}{q^2}}$

## II. Séries entières

**5** Soit $z \in \mathbb{C}^*$. Comme $a \neq 0$, on peut calculer

$$\left|\frac{u_{n+1}(a)z^{n+1}}{u_n(a)z^n}\right| = \frac{|z|^{n+1}\,|a|^n}{|a|^{n+1}\,|z|^n} = \frac{|z|}{|a|}$$

D'après la règle de D'Alembert, pour $|z| < |a|$, la série $\sum u_n(a)z^n$ converge, tandis que pour $|z| > |a|$, elle diverge grossièrement. Son rayon est donc égal à $|a|$. Ainsi,

$\boxed{\text{La série entière } \sum u_n(a)z^n \text{ a un rayon de convergence égal à } |a|.}$

**6** Soit $z \in \mathbb{C}$ avec $|z| < |a|$. D'après la question 5, la série $\sum u_n(a)z^n$ converge. Comme $z/a \neq 1$, on a

$$\sum_{n=0}^{+\infty} u_n(a)z^n = \sum_{n=0}^{+\infty} \frac{-1}{a}\left(\frac{z}{a}\right)^n = \frac{-1}{a}\frac{1}{1-\dfrac{z}{a}} = \frac{-1}{a-z} = \frac{1}{z-a}$$

$\boxed{\text{Pour tout } z \in \mathbb{C} \text{ avec } |z| < |a|, \text{ on a } \dfrac{1}{z-a} = \displaystyle\sum_{n=0}^{+\infty} u_n(a)z^n.}$

**7** Soit $n \in \mathbb{N}$. Par définition,

$$v_n = \sum_{k=0}^{n} u_k(a)u_{n-k}(b) = \sum_{k=0}^{n} \frac{-1}{a^{k+1}}\frac{-1}{b^{n-k+1}} = \frac{1}{ab^{n+1}}\sum_{k=0}^{n}\frac{1}{a^k b^{-k}} = \frac{1}{ab^{n+1}}\sum_{k=0}^{n}\left(\frac{b}{a}\right)^k$$

# Centrale Maths 1 PC 2017 — Corrigé

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

Le sujet tourne autour de la notion de partitions d'un ensemble à $n$ éléments. Comme tout sujet de combinatoire, c'est l'occasion de faire le lien entre plusieurs domaines classiques des mathématiques, comme les polynômes, les séries entières et, dans cet énoncé, les probabilités.

- La première partie propose l'étude du nombre $\mathrm{S}(n,k)$ de partitions d'un ensemble à $n$ éléments en $k$ parties. Il s'agit ici essentiellement de se familiariser avec la notion et d'établir une formule de récurrence utile pour la suite.
- Le deuxième partie introduit le nombre de Bell $\mathrm{B}_n$ qui compte les partitions d'un ensemble à $n$ éléments ; il est égal à la somme sur $k$ des quantités $\mathrm{S}(n,k)$ manipulées dans la partie précédente. L'objectif de cette partie est d'obtenir une expression simple de la série génératrice exponentielle $\sum_{n\geqslant 0} \mathrm{B}_n\, x^n/n!$ de la suite $(\mathrm{B}_n)_{n\in\mathbb{N}}$.
- La troisième partie fait réapparaître naturellement les nombres $\mathrm{S}(n,k)$ de la première partie dans le cadre des polynômes. On établit en effet qu'il s'agit des coefficients d'une matrice de passage entre la base canonique de $\mathbb{R}_n[\mathrm{X}]$ et la famille bien connue des polynômes de Hilbert (même si l'énoncé ne les nomme pas explicitement). On étudie également les séries génératrices exponentielles des suites $(\mathrm{S}(n,k))_{n\in\mathbb{N}}$ pour obtenir un certain nombre de jolies formules.
- La quatrième partie définit la notion de moment d'une variable aléatoire à valeurs entières, puis relie le nombre de Bell $\mathrm{B}_n$ aux moments d'une variable aléatoire suivant une loi de Poisson.
- La cinquième et dernière partie fait pour finir le lien entre $\mathrm{S}(n,k)$ et les sommes finies $\sum_{k=1}^{n} k^p$ pour $p$ et $n$ entiers et permet d'étudier quelques propriétés notables de ces sommes.

Ce sujet permet de constater une particularité assez fréquente en combinatoire : une suite définie initialement comme le cardinal d'une famille d'objets (en l'occurrence, le nombre de partitions d'un ensemble à $n$ éléments) intervient aussi dans des domaines a priori sans rapport. La combinatoire, relativement peu étudiée en CPGE, constitue ainsi une passerelle entre des domaines très divers. Lorsqu'elle apparaît aux concours, elle donne des énoncés riches faisant intervenir un large spectre du programme de prépa.

# Indications

## Partie I

I.E Utiliser la formule du triangle de Pascal.

## Partie II

II.A Exprimer l'ensemble des partitions comme l'union disjointe des ensembles des partitions en $k$ parties.

II.B Considérer une partition de $[\![\,1\,;n+1\,]\!]$ et isoler la partie contenant l'élément $n+1$.

II.C Procéder par récurrence et utiliser la question II.B.

II.D Comparer la série $\sum \left(\mathrm{B}_n/n\,!\right)x^n$ à la série $\sum x^n$.

II.E Développer en série entière le produit $\mathrm{e}^{\,x}f(x)$ à l'aide d'un produit de Cauchy.

II.F Résoudre l'équation différentielle de la question II.E.

## Partie III

III.B.1 Observer que $\mathrm{H}_k$ divise $\mathrm{H}_{k+1}$ et factoriser l'expression $\mathrm{H}_{k+1}+k\mathrm{H}_k$ par le polynôme $\mathrm{H}_k$.

III.B.2 Procéder par récurrence et utiliser les questions I.C et III.B.1.

III.C.1 Comparer la fonction $f_k$ à la série entière de la question II.D.

III.C.3 Procéder par récurrence sur $k$, dériver la fonction $f_k$ et comparer le problème de Cauchy obtenu à celui que satisfait la fonction $g_k$.

III.D.1 Reconnaître un développement limité usuel.

## Partie IV

IV.A Utiliser le fait que les séries entières de terme général $a_nx^n$ et $na_nx^n$ ont le même rayon de convergence.

IV.B.1 Appliquer les théorèmes de dérivation et d'échange entre limite et série.

IV.B.3 Prouver que la série proposée par l'énoncé converge sur $[\,-1\,;1\,]$ et définir une variable aléatoire dont la série génératrice est de la même forme à une constante multiplicative près.

IV.C.1 Calculer $\mathrm{E}(\mathrm{H}_k(\mathrm{Y}))$ et utiliser la question III.B.2.

## Partie V

V.A Relier $k^n$ à une intégrale entre $k$ et $k+1$ d'un polynôme de degré $n$.

V.C Utiliser la question III.B.2 et faire apparaître une somme télescopique.

V.D.3 Chercher un antécédent par $\Phi$ du polynôme $\mathrm{X}^{2r+1}$.

V.E.1 Utiliser la question V.A.

V.E.2 Écrire $\mathrm{P}_r$ sous une forme développée et calculer $\Phi(\mathrm{P}_r)$ sous forme développée.

# I. Nombre de partitions en $k$ parties

**I.A** Remarquons que toute partition de l'ensemble $[\![1\,;n]\!]$ en $k$ parties est obtenue en choisissant $k$ sous-ensembles deux-à-deux disjoints de $[\![1\,;n]\!]$. Or, il y a $2^n$ sous-ensembles de $[\![1\,;n]\!]$. Ainsi, le nombre de partitions en $k$ parties est majoré par $\binom{2^n}{k}$. En particulier,

Le nombre de partitions de $[\![1\,;n]\!]$ en $k$ parties est fini.

**I.B.1** Une partitions de l'ensemble $[\![1\,;n]\!]$ en $k$ parties ne peut pas contenir strictement plus de $n$ sous-ensembles, car chacun d'entre-eux doit être non vide. Ainsi,

Si $k > n$, alors $\mathrm{S}(n,k) = 0$.

**I.B.2** Pour tout entier $n$ non nul, l'ensemble $\{[\![1\,;n]\!]\}$ est l'unique partition de l'ensemble $[\![1\,;n]\!]$ en 1 partie, de sorte que

$$\forall n \in \mathbb{N}^* \quad \mathrm{S}(n,1) = 1$$

**I.C** Notons $\mathcal{E}_k^n$ l'ensemble des partitions de $[\![1\,;n]\!]$ en $k$ parties. On sépare cet ensemble en deux sous-ensembles disjoints :

- l'ensemble $\mathcal{E}_{k,1}^n$ des partitions en $k$ parties dont l'une d'elles est réduite à l'élément $n$ ;
- l'ensemble $\mathcal{E}_{k,2}^n$ des partitions en $k$ parties dont aucune n'est réduite à $n$.

Ainsi, $$\mathrm{S}(n,k) = |\mathcal{E}_k^n| = |\mathcal{E}_{k,1}^n| + |\mathcal{E}_{k,2}^n|$$

Déterminons maintenant les cardinaux de $\mathcal{E}_{k,1}^n$ et $\mathcal{E}_{k,2}^n$.

- Étant donné un élément de $\mathcal{E}_{k,1}^n$, si l'on supprime la partie réduite à $n$, on obtient une partition de $[\![1\,;n-1]\!]$ en $k-1$ parties. Cette opération est bijective, la réciproque consistant simplement à rajouter la partie $\{n\}$. On en déduit en particulier que
$$|\mathcal{E}_{k,1}^n| = |\mathcal{E}_{k-1}^{n-1}| = \mathrm{S}(n-1,k-1)$$
- Soit maintenant un élément de $\mathcal{E}_{k,2}^n$. Si l'on supprime l'élément $n$ de la partie à laquelle il appartient, on obtient cette fois une partition de $[\![1\,;n-1]\!]$ en $k$ parties. Cependant, l'opération n'est plus bijective, mais surjective, et chaque élément de $\mathcal{E}_k^{n-1}$ admet exactement $k$ antécédents puisque l'on a $k$ choix pour la partie à laquelle on va rajouter $n$. Il vient cette fois que
$$|\mathcal{E}_{k,2}^n| = k \cdot |\mathcal{E}_k^{n-1}| = k \cdot \mathrm{S}(n-1,k)$$

On peut conclure de tout ceci que

$$\mathrm{S}(n,k) = \mathrm{S}(n-1,k-1) + k \cdot \mathrm{S}(n-1,k)$$

**I.D.1** Il s'agit de donner un code qui reprenne tous les cas vus précédemment. Plus précisément, les conventions de l'énoncé et la question I.B donnent les cas de base, la question précédente permet de construire le cas récursif. Il suffit en fait de remarquer que le système de calcul est très proche de la méthode de remplissage du triangle de Pascal. On traitera donc les cas des bordures, c'est-à-dire quand $n = 0$ ou $k = 0$, puis les cas qui n'ont pas de sens et enfin le cas général.

```
def calculS(n,k):
  #Tester les cas "convention".
  if n==0 and k==0:
    #On traite ici le cas n=k=0
    return 1
  elif n==0 or k==0:
    #On traite ici les cas 0=n<k et 0=k<n
    return 0
  elif k>n:
    #On traite ici le cas qui n'a pas de sens k>n
    return 0
  #cas recursif
  else:
    res = calculS(n-1,k-1) + k*calculS(n-1,k)
    return res
```

> On peut remarquer que cette fonction termine. En effet, une quantité strictement décroissante est la somme des deux arguments, et pourtant elle doit rester entière et positive.

**I.D.2** Notons $\mathrm{C}(n,k)$ le nombre d'opérations nécessaires dans l'algorithme précédent pour calculer $\mathrm{S}(n,k)$. En raison des appels récursifs si $1 \leqslant k \leqslant n$, alors

$$\mathrm{C}(n,k) = \mathrm{C}(n-1,k-1) + \mathrm{C}(n-1,k) + 2 \geqslant \mathrm{C}(n-1,k-1) + \mathrm{C}(n-1,k)$$

> Remarquons ici que c'est une formule combinatoire qui correspond au triangle de Pascal.

Si $k > n$, seul l'appel de la fonction est fait et

$$\mathrm{C}(n,k) = 1 \geqslant \binom{n}{k} = 0$$

Maintenant, supposons $k \leqslant n$ et procédons par récurrence. Considérons la propriété

$$\mathscr{P}(n) : \qquad \forall k \in [\![\, 0\,;n \,]\!] \quad \mathrm{C}(n,k) \geqslant \binom{n}{k}$$

- $\underline{\mathscr{P}(0)}$ est vraie car, en considérant le coût de l'appel de la fonction,

$$\mathrm{C}(0,0) = 1 = \binom{0}{0}$$

- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$ : Soit $k \in [\![\, 0\,;n+1 \,]\!]$. Si $k = 0$, alors encore une fois, seul l'appel de la fonction est compté et

$$\mathrm{C}(n+1,0) = 1 = \binom{n+1}{0}$$

Si $1 \leqslant k \leqslant n$, alors

$$\begin{aligned}
\mathrm{C}(n+1,k) &= \mathrm{C}(n,k-1) + \mathrm{C}(n,k) + 2 \\
&\geqslant \binom{n}{k-1} + \binom{n}{k} + 2 && \text{(d'après } \mathscr{P}(n)\text{)} \\
&\geqslant \binom{n}{k-1} + \binom{n}{k} \\
\mathrm{C}(n+1,k) &\geqslant \binom{n+1}{k} && \text{(règle de Pascal)}
\end{aligned}$$

# Centrale Maths 2 PC 2017 — Corrigé

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

Ce sujet traite de la vitesse de convergence des suites réelles et en particulier de majorations de probabilités par des suites de limite nulle, ayant différentes vitesses de convergence, notamment dans le cadre de la loi faible des grands nombres.

- La première partie traite de suites réelles convergentes et sans sous-suite stationnaire. Pour une telle suite $u$ de limite $\ell$, on s'intéresse à l'éventuelle limite $\ell^c$ de la suite de terme général $|u_{n+1}-\ell|/|u_n-\ell|$, définie au moins à partir d'un certain rang. Lorsque $\ell^c$ existe, sa valeur permet de définir la vitesse de convergence de la suite $u$ : lente lorsque $\ell^c = 1$, géométrique lorsque $\ell^c \in ]\,0\,;1\,[$, rapide lorsque $\ell^c = 0$. On démontre des résultats généraux sur la vitesse de convergence et on étudie de nombreux exemples : suites de référence, suite d'intégrales généralisées, suites des sommes partielles de séries de Riemann convergentes, suite des sommes partielles d'une série exponentielle, suite récurrente définie par une relation de la forme $u_{n+1} = f(u_n)$.
- Dans la seconde partie, on établit d'abord trois résultats préliminaires d'analyse, tous indépendants. On s'intéresse ensuite à des variables aléatoires X admettant un moment exponentiel d'ordre $\alpha$ pour $\alpha > 0$, c'est-à-dire telles que la variable aléatoire $\mathrm{e}^{\,\alpha|\mathrm{X}|}$ ait une espérance finie. On étudie dans un premier temps le cas de variables aléatoires suivant des lois du programme : loi de Poisson, loi géométrique, loi binomiale. On considère ensuite des suites $(\mathrm{X}_k)_{k\geqslant 1}$ de variables aléatoires indépendantes et de même loi, vérifiant certaines hypothèses. En notant $\mathrm{S}_n$ la somme $\mathrm{X}_1 + \cdots + \mathrm{X}_n$ pour $n \in \mathbb{N}^*$ et $m$ l'espérance commune des $\mathrm{X}_k$, on majore la probabilité $\mathbb{P}(|\mathrm{S}_n/n - m| \geqslant \varepsilon)$. On fait d'abord appel à l'inégalité donnée par la loi faible des grands nombres, puis on établit une majoration par une suite de limite nulle dont la convergence est plus rapide. Enfin, on traite le cas où $m = 0$, pour lequel on fait appel à la notion de convexité sans la nommer et sans avoir besoin de connaissances préalables sur cette notion qui n'est pas au programme.

À l'exception de la question I.B.3 qui utilise la notion d'intégrale généralisée, la partie I peut être traitée dès la première année de classes préparatoires. La seconde partie nécessite d'avoir étudié les probabilités de deuxième année, ainsi que les séries de fonctions et les séries entières. Ce sujet est un peu long mais ne comporte aucune difficulté majeure. C'est un bon problème d'entraînement qui fait appel à de nombreuses notions d'analyse et probabilité des programmes de PCSI et PC.

## Indications

I.A.3 Construire une suite $u$ de limite nulle telle que $u_{2n} = u_{2n+1}$ pour tout $n \in \mathbb{N}$.

I.A.4 On pourra procéder par l'absurde pour montrer que $\ell^c \leqslant 1$.

I.B.2.b Utiliser le développement asymptotique de la question I.B.2.a pour trouver un équivalent de $v_n - \mathrm{e}$ puis de $v_n^c$ quand $n$ tend vers $+\infty$.

I.B.3.a Appliquer le théorème de convergence dominée.

I.B.3.b Suivre l'indication de l'énoncé puis appliquer encore le théorème de convergence dominée pour trouver un équivalent de $\mathrm{I}_n$ quand $n$ tend vers $+\infty$.

I.B.4.a Penser à une comparaison série/intégrale.

I.B.4.b Trouver un équivalent de $\mathrm{S}_n$ puis de $\mathrm{S}_n^c$ à l'aide de l'encadrement précédent.

I.C.1 Majorer la suite $(u_n^c)_{n\in\mathbb{N}}$ par une suite de limite nulle.

I.C.2.d Procéder par l'absurde.

I.C.3.b Pour montrer que $u \in \mathrm{E}$, vérifier que s'il existe $n \in \mathbb{N}$ tel que $u_n = \ell$, alors $u$ est stationnaire. Écrire ensuite $u_n^c$ à l'aide d'un taux d'accroissement pour étudier sa limite.

I.C.3.c Utiliser les questions I.C.3.b et I.A.4.

I.C.3.d Faire appel à la formule de Taylor-Young.

II.A.2 On peut démontrer à l'aide d'une étude de fonctions que, pour tout réel $t$ dans $[\,0\,;1\,]$, $\mathrm{e}^{(b-a)(1-t)} \leqslant t + (1-t)\mathrm{e}^{b-a}$ et en déduire le résultat voulu.

II.B.2 Se servir du théorème de transfert à la fois pour caractériser l'existence des espérances et pour les calculer.

II.C.1.a Vérifier que, pour $u \geqslant 0$, $u \leqslant \mathrm{e}^{\,u}$ puis majorer $|\mathrm{X}|$ par une variable aléatoire d'espérance finie.

II.C.1.b Justifier que X admet un moment d'ordre 2 à l'aide du développement en série entière de l'exponentielle.

II.C.2.a Utiliser le théorème de continuité des séries de fonctions.

II.C.2.b Intuitivement, on s'attend à ce que la dérivée de la fonction $t \longmapsto \mathbb{E}(\mathrm{e}^{\,t\mathrm{X}})$ soit la fonction $t \longmapsto \mathbb{E}(\mathrm{X}\,\mathrm{e}^{\,t\mathrm{X}})$, c'est-à-dire à $\dfrac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}(\mathrm{e}^{\,t\mathrm{X}}) = \mathbb{E}\Big(\dfrac{\mathrm{d}}{\mathrm{d}t}\mathrm{e}^{\,t\mathrm{X}}\Big)$.
Pour le démontrer rigoureusement, appliquer le théorème de la classe $\mathscr{C}^1$ des séries de fonctions en utilisant la question II.A.3.b pour majorer les dérivées.

II.C.3.b Démontrer que $f$ est strictement décroissante au voisinage de $0^+$.

II.C.4 Utiliser les deux propositions rappelées et admises dans l'énoncé.

II.C.5.a Penser à l'inégalité de Markov.

II.C.5.b Se servir des questions II.C.5.a et II.C.3.b.

II.C.6 Écrire l'ensemble $(|\mathrm{S}_n/n - m| \geqslant \varepsilon)$ comme une réunion et utiliser la question II.C.5.b pour les familles $(\mathrm{X}_k)_{k\geqslant 1}$ et $(-\mathrm{X}_k)_{k\geqslant 1}$.

II.D.2.b Faire appel à la question II.A.2.

II.D.3.a Appliquer l'inégalité de la question II.D.2.b.

II.D.5 Reprendre la démarche de la question II.C.6 et utiliser les questions II.D.3.b et II.D.4. Choisir ensuite le réel $t > 0$ qui donne le meilleur majorant.

II.D.6 Se souvenir qu'une variable aléatoire de loi binomiale de paramètres $n$ et $p$ a même loi que la somme de $n$ variables aléatoires de Bernoulli indépendantes de même paramètre $p$. Introduire une famille de variables aléatoires $(\mathrm{X}_k)_{k\geqslant 1}$ de loi de Bernoulli de paramètre $p$, mutuellement indépendantes, puis la famille $(\mathrm{X}_k - p)_{k\geqslant 1}$.

# I. Vitesse de convergence d'une suite réelle

**I.A.1** Soit $u$ la suite définie par

$$\forall n \in \mathbb{N} \qquad u_n = \frac{1}{n+1}$$

La suite $u$ appartient à $\mathbb{R}^{\mathbb{N}}$, elle converge vers 0 et tous ses termes sont non nuls donc c'est un élément de E. De plus,

$$\forall n \in \mathbb{N} \qquad u_n^c = \left|\frac{u_{n+1}-0}{u_n-0}\right| = \frac{1/(n+2)}{1/(n+1)} = \frac{n+1}{n+2}$$

La suite $(u_n^c)_{n\in\mathbb{N}}$ converge donc vers 1, ce qui implique que $u \in \mathrm{E}^c$. On a par conséquent prouvé que

L'ensemble $\mathrm{E}^c$ est non vide.

**I.A.2** La suite nulle est l'élément nul de $\mathbb{R}^{\mathbb{N}}$. Or, elle n'est pas dans E, puisqu'elle est constante et égale à sa limite, elle n'est donc pas dans $\mathrm{E}^c$. Par conséquent

L'ensemble $\mathrm{E}^c$ n'est pas un sous-espace vectoriel de $\mathbb{R}^{\mathbb{N}}$.

> En notant $u$ la suite introduite à la question I.A.1, on remarque aussi que $u-u$ n'appartient pas à $\mathrm{E}^c$, ce qui prouve que $\mathrm{E}^c$ n'est pas stable par combinaison linéaire.

**I.A.3** Pour établir que $\mathrm{E}^c$ est strictement inclus dans E, il suffit de trouver un élément de E qui n'est pas dans $\mathrm{E}^c$ car, par définition, $\mathrm{E}^c$ est inclus dans E.

> Pour construire un élément de E qui n'est pas dans $\mathrm{E}^c$, on cherche pour simplifier une suite de limite nulle, à termes différents de 0 au moins à partir d'un certain rang, qui ne doit pas être dans $\mathrm{E}^c$, c'est-à-dire que la suite de terme général $u_{n+1}/u_n$ ne doit pas avoir de limite. On peut alors se souvenir des exemples de suites donnés dans le cours pour lesquels la règle de d'Alembert ne s'applique pas, ou bien fabriquer simplement, comme dans l'exemple ci-dessous, une suite de limite nulle pour laquelle les quotients $u_{n+1}/u_n$ ont des valeurs différentes selon la parité de $n$.

Définissons la suite $u$ telle que

$$\forall n \in \mathbb{N} \qquad u_{2n} = u_{2n+1} = \frac{1}{3^n}$$

Les deux suites extraites d'indices pairs et impairs sont alors convergentes et ont toutes deux pour limite 0 ; on en déduit que la suite $u$ converge vers cette limite commune, c'est-à-dire 0. Comme tous les termes de la suite sont non nuls, on peut déjà affirmer que $u$ appartient à E. De plus, pour tout $n \in \mathbb{N}$,

$$u_{2n}^c = \left|\frac{u_{2n+1}-0}{u_{2n}-0}\right| = \frac{u_{2n+1}}{u_{2n}} = 1$$

et

$$u_{2n+1}^c = \left|\frac{u_{2n+2}-0}{u_{2n+1}-0}\right| = \frac{u_{2n+2}}{u_{2n}} = \frac{1}{3}$$

Puisque les deux suites extraites $(u_{2n}^c)_{n\in\mathbb{N}}$ et $(u_{2n+1}^c)_{n\in\mathbb{N}}$ ont des limites différentes, la suite $(u_n^c)_{n\in\mathbb{N}}$ n'est pas convergente et ainsi $u \notin \mathrm{E}^c$. Comme on a construit un élément de E qui n'est pas dans $\mathrm{E}^c$, on peut conclure que

L'ensemble $\mathrm{E}^c$ est strictement inclus dans E.

**I.A.4** Soit $u \in \mathrm{E}^c$, notons $\ell$ sa limite. Par définition,

$$\ell^c = \lim_{n\to+\infty} u_n^c = \lim_{n\to+\infty} \left|\frac{u_{n+1}-\ell}{u_n-\ell}\right|$$

Soit $n_0 \in \mathbb{N}$ tel que, pour tout $n \geqslant n_0$, $u_n \neq \ell$. Pour tout $n \geqslant n_0$, $u_n^c$ est bien défini et $u_n^c \geqslant 0$. Par passage à la limite dans cette inégalité, il vient $\ell^c \geqslant 0$. Démontrons ensuite par l'absurde que $\ell^c \leqslant 1$. Supposons que $\ell^c > 1$. Soit $r \in ]\,1\,;\ell^c\,[$ (qui est non vide puisque $1 < \ell^c$). Comme la suite $(u_n^c)_{n\in\mathbb{N}}$ tend vers $\ell^c$, il existe un rang $n_1 \geqslant n_0$ tel que, pour tout $n \geqslant n_1$, $u_n^c \geqslant r$. Ainsi, pour tout $n \geqslant n_1$,

$$|u_{n+1}-\ell| \geqslant r\,|u_n-\ell|$$

Il en découle, à l'aide d'une récurrence immédiate, que

$$\forall n \geqslant n_1 \qquad |u_n-\ell| \geqslant r^{n-n_1}\,|u_{n_1}-\ell|$$

Or, $r > 1$, donc la suite géométrique $(r^{n-n_1}\,|u_{n_1}-\ell|)_{n\geqslant n_1}$ tend vers $+\infty$ (on sait que $u_{n_1}$ est différent de $\ell$ puisque $n_1 \geqslant n_0$). Par comparaison, il s'ensuit que la suite $(|u_n-\ell|)_{n\geqslant 1}$ tend aussi vers $+\infty$, ce qui est absurde car $(u_n)_{n\in\mathbb{N}}$ tend vers $\ell$. On en conclut que $\ell^c \leqslant 1$ et finalement que

$$\boxed{\text{Si } u \in \mathrm{E}^c, \text{ alors } \lim_{n\to+\infty} u_n^c \in [\,0\,;1\,].}$$

**I.B.1** Soient $k \in \mathbb{N}^*$ et $q \in ]\,0\,;1\,[$. Posons

$$u = \left(\frac{1}{(n+1)^k}\right)_{n\in\mathbb{N}} \qquad v = (n^k\,q^n)_{n\in\mathbb{N}} \qquad \text{et} \qquad w = \left(\frac{1}{n!}\right)_{n\in\mathbb{N}}$$

et étudions la convergence et la vitesse de convergence de ces trois suites.
Observons dans un premier temps que les suites $u$, $v$ et $w$ sont des suites réelles de limite nulle, à termes strictement positifs ; ce sont donc des éléments de E.

- Vitesse de convergence de la suite $u$ : pour tout $n \in \mathbb{N}$,

$$u_n^c = \left|\frac{u_{n+1}-0}{u_n-0}\right| = \frac{u_{n+1}}{u_n} = \frac{1/(n+2)^k}{1/(n+1)^k} = \frac{(n+1)^k}{(n+2)^k} = \left(1-\frac{1}{n+2}\right)^k$$

  La suite $(u_n^c)_{n\in\mathbb{N}}$ converge vers 1, par conséquent $u \in \mathrm{E}^c$ et $\ell^c = 1$.

$$\boxed{\text{La suite } \left(1/(n+1)^k\right)_{n\in\mathbb{N}} \text{ est dans } \mathrm{E}^c \text{ et sa vitesse de convergence est lente.}}$$

- Vitesse de convergence de la suite $v$ : pour tout $n \in \mathbb{N}$,

$$v_n^c = \left|\frac{v_{n+1}-0}{v_n-0}\right| = \frac{v_{n+1}}{v_n} = \frac{(n+1)^k\,q^{n+1}}{n^k\,q^n} = q\left(1+\frac{1}{n}\right)^k$$

  La suite $(v_n^c)_{n\in\mathbb{N}}$ converge vers $q$, par conséquent $v \in \mathrm{E}^c$ et $\ell^c = q$.

$$\boxed{\begin{array}{l}\text{La suite } (n^k\,q^n)_{n\in\mathbb{N}} \text{ est dans } \mathrm{E}^c \text{ et sa vitesse}\\ \text{de convergence est géométrique de rapport } q.\end{array}}$$

- Vitesse de convergence de la suite $w$ : pour tout $n \in \mathbb{N}$,

$$w_n^c = \left|\frac{w_{n+1}-0}{w_n-0}\right| = \frac{w_{n+1}}{w_n} = \frac{1/(n+1)!}{1/n!} = \frac{n!}{(n+1)!} = \frac{1}{n+1}$$

  La suite $(w_n^c)_{n\in\mathbb{N}}$ converge vers 0, par conséquent $w \in \mathrm{E}^c$ et $\ell^c = 0$.

$$\boxed{\text{La suite } (1/n!)_{n\in\mathbb{N}} \text{ est dans } \mathrm{E}^c \text{ et sa vitesse de convergence est rapide.}}$$

# Centrale Informatique PC 2017 — Corrigé

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

Ce sujet d'informatique a pour contexte la mission martienne *Mars Exploration Rovers*. Celle-ci conduit à une étude en trois parties des aspects robotiques de l'exploration, principalement le déplacement du système sur une carte : calculs de distances parcourues et optimisation du chemin reliant tous les points suivant deux approches. De nombreuses questions font intervenir des tableaux `numpy`, donc l'utilisation des fonctions associées, alors que des listes de listes suffiraient la plupart du temps. Un catalogue de ces fonctions est fourni en fin de sujet, avec des explications claires et des exemples.

- La première partie est composée de trois sous-parties indépendantes et de difficultés inégales : la première permet de réaliser deux fonctions élémentaires de génération de carte et de calcul de distance ; la deuxième traite très brièvement de traitement d'image ; la troisième étudie le fonctionnement de la base de données des analyses à effectuer. On regrette la difficulté importante de la première question, qui a dû décourager nombre de candidats, du moins ceux qui ont correctement lu le sujet. Comparativement, les deux questions sur le traitement d'image sont d'une simplicité déconcertante. Enfin, on note que les requêtes SQL demandent l'utilisation d'une syntaxe (`IS NULL`) conceptuellement complexe et certainement peu enseignée.
- La deuxième partie aborde le problème très classique dit « du voyageur de commerce » : il faut optimiser le passage du robot par tous les points d'une liste une seule et unique fois. On met en place ici quelques fonctions élémentaires puis on réalise l'algorithme glouton du plus proche voisin. L'ensemble est équilibré et suffisamment progressif en termes de difficulté.
- La dernière partie continue le travail précédent en proposant pour le même problème l'étude d'un autre type de solution, lui aussi classique : un algorithme génétique. Les questions sont bien détaillées et se suivent parfaitement ; y répondre donne le plaisir de construire petit à petit un algorithme complexe et utilisable sur un cas concret.

Ce sujet laisse au final deux impressions : celle d'un début mal maîtrisé et peu progressif, suivie dans les deux dernières parties d'un sujet classique permettant aux candidats de s'exprimer correctement. Il contient quelques questions de langage SQL mais n'aborde pas la partie ingénierie numérique du programme et ne demande aucune démonstration théorique.

# Indications

## Partie I

I.A.1.a Le plus simple est de continuer à générer des couples *tant que* le résultat n'en contient pas `n` et d'enregistrer ces couples uniquement s'ils ne sont pas déjà présents dans le résultat, à l'aide de l'opérateur `not in`.

I.A.2 Ce genre de matrices est symétrique, on évite de doubler les calculs. La distance entre deux points se calcule à l'aide du théorème de Pythagore.

I.B Il existe une fonction permettant d'obtenir les dimensions d'un tableau `numpy`.

I.C.1 Pour tester si un champ est égal à `NULL`, ce n'est pas l'opérateur `=` mais l'opérateur `IS` qui est nécessaire : `champ IS NULL` ou `champ IS NOT NULL`.

I.C.3 Un seul résultat par exploration, pour plusieurs explorations, car il s'agit d'une requête d'agrégation. Les champs à calculer et la sélection des explorations terminées se font sur deux tables, une jointure est donc nécessaire.

I.C.4 L'énoncé est incomplet, il ne dit pas comment sont stockés les entiers dans la base. On peut imaginer qu'ils sont positifs et enregistrés sur 64 bits.

I.C.5 Il faut réaliser une jointure. Toutes les tables décrites dans l'énoncé ne sont pas utiles. Les champs `EX_NUM` et `TY_NUM` sont présents et équivalents dans trois tables.

## Partie II

II.A.2 Générer une liste de `n` valeurs booléennes permet de savoir rapidement si un point a déjà été visité, sans utiliser la syntaxe `in`.

II.C.1 Il s'agit d'une recherche classique de minimum, mais uniquement sur les points encore disponibles à la $\texttt{i}^{\text{e}}$ itération. Comme en II.A.2, une liste de valeurs booléennes peut aider. Faire attention à l'initialisation du critère minimal recherché.

II.C.3 Les points donnés sont alignés, faire un dessin au brouillon.

## Partie III

III.A Pour générer une liste aléatoire d'entiers, le plus simple est d'utiliser la fonction de permutation donnée en fin d'énoncé, sur la liste complète.

III.B Pour trier les chemins, la méthode `sort` fonctionne immédiatement et il n'est pas utile de la coder à la main. Attention, il faut supprimer directement des éléments de la liste `p`, il est interdit d'écrire `p = p[0 :len(p)//2]` par exemple.

III.C.1 La fonction `random.sample` permet d'obtenir plusieurs valeurs distinctes prises aléatoirement au sein d'une liste.

III.C.2 Penser à utiliser les fonctions `muter_chemin` et `longueur_chemin` réalisées aux questions précédentes.

III.D.1 On peut concaténer deux listes avec l'opérateur `+`.

III.E.1 Attention à la syntaxe des fonctions précédentes, notamment de celles qui ne renvoient pas de valeur.

III.E.2 Où est le meilleur chemin dans une génération ? Est-il modifié ?

III.E.3 Classiquement, une série converge si sa valeur n'évolue plus beaucoup.

# I. Création d'une exploration et gestion des points d'intérêt

La syntaxe des définitions de fonctions choisie pour cet énoncé est une possibilité offerte par Python pour rendre le code plus explicite (les « annotations »). Bien que peu d'élèves l'aient déjà vu, son emploi n'a pas dû être un problème.

**I.A.1.a** Dans la fonction `générer_PI`, on peut choisir de manipuler une liste et d'utiliser la méthode `append` pour ajouter chaque point avant une transformation finale en tableau `numpy`, puisque celle-ci est imposée par l'énoncé :

```
def générer_PI(n:int, cmax:int) -> np.ndarray:
    PI = []
    while len(PI) < n:
        x = random.randrange(0,cmax+1)
        y = random.randrange(0,cmax+1)
        if [x,y] not in PI:
            PI.append([x,y])
    return np.array(PI)
```

La difficulté de cette question réside en grande partie dans le besoin de ne garder que des points distincts. La réflexion pour obtenir une liste de $n$ points aléatoire est très courte, mais celle pour éliminer les doublons l'est beaucoup moins, notamment parce qu'un grand nombre de possibilités existent. Si l'on ajoute à cela le retour sous forme de tableau `numpy` exigé, difficulté supplémentaire pour beaucoup de candidats, alors que la liste de listes est suffisante, cette question est plutôt difficile pour un début d'épreuve.

Il est aussi possible d'utiliser un tableau `numpy` depuis le début de l'algorithme, en l'initialisant à un tableau de `n` zéros puis en le remplissant ligne par ligne, en vérifiant que chaque couple tiré au sort n'est pas déjà présent :

```
def générer_PI(n:int, cmax:int) -> np.ndarray:
    PI = np.zeros((n,2),int) # n lignes, 2 colonnes
    i = 0
    while i < n:
        x = random.randrange(0,cmax+1)
        y = random.randrange(0,cmax+1)
        if [x,y] not in PI[0:i]:
            PI[i] = [x,y]; i = i+1
    return PI
```

Malheureusement, malgré ce qu'indique l'annexe de l'énoncé, la construction en `[x,y] in PI[0:i]` ne donne pas le résultat prévu avec un tableau `numpy` :

```
>>> PI = np.array([[1,2],[3,4]])
>>> [1,2] in PI # Normal
True
>>> [1,5] in PI # Pas normal !
True
```

ce qui n'est pas vraiment le résultat escompté... À n'en pas douter, la fonction proposée plus haut devrait être acceptée par les correcteurs, mais ce code ne fonctionne pas en pratique. Une des possibilités pour résoudre ce problème, très au-delà de ce que l'on peut attendre d'un candidat, est

```
any(np.equal([x,y],PI[0:i]).all(1))
```

qui répond correctement sur notre exemple :

```
>>> any(np.equal([1,2],PI).all(1))
True
>>> any(np.equal([1,5],PI).all(1))
False
```

**I.A.1.b** Les deux nombres doivent **être positifs** et aussi permettre d'obtenir `n` points distincts sur les `cmax+1`, c'est-à-dire

$$\boxed{\texttt{n} \leqslant (\texttt{cmax+1})^2}$$

Les coordonnées ainsi que `cmax` étant notées en millimètres, il est peu probable que `n` s'approche de cette valeur limite. On peut néanmoins noter dans ce cas que l'algorithme donné à la question précédente devient très inefficace. Il convient alors de le remplacer par un algorithme procédant par élimination d'un nombre complémentaire de points, choisis aléatoirement.

**I.A.2** La fonction demandée doit calculer pour chaque couple de points (`i`,`j`) la distance, à l'aide du théorème de Pythagore. Le résultat est donc symétrique et il faut recopier chaque valeur du triangle inférieur gauche vers le triangle supérieur droit. On n'oublie pas d'ajouter la position actuelle en fin de ligne et de colonne.

```
def calculer_distances(PI:np.ndarray) -> np.ndarray:
    n = len(PI)
    pos = position_robot()
    distances = np.zeros((n+1,n+1))
    for i in range(n):
        for j in range(i):
            distances[i,j] = math.sqrt((PI[i,0]-PI[j,0])**2 \
                                       + (PI[i,1]-PI[j,1])**2)
            distances[j,i] = distances[i,j]
        distances[i,n] = math.sqrt((PI[i,0]-pos[0])**2 \
                                   + (PI[i,1]-pos[1])**2)
        distances[n,i] = distances[i,n]
    return distances
```

On peut éviter une des racines en ajoutant la position actuelle directement à la liste des points d'intérêt avant calcul, mais il faut pour cela être un peu familier de la fonction `np.concatenate`, qui nécessite des listes ou des tableaux `numpy` en argument. Les crochets ici sont essentiels, ce qui n'est pas du tout évident...

```
def calculer_distances(PI:np.ndarray) -> np.ndarray:
    points = np.concatenate((PI,[position_robot()]))
    n = len(points)
    distances = np.zeros((n,n))
    for i in range(n):
        for j in range(i):
            distances[i,j] = math.sqrt(                    \
                               (points[i,0]-points[j,0])**2 \
                             + (points[i,1]-points[j,1])**2)
            distances[j,i] = distances[i,j]
    return distances
```

# Mines Maths 1 PC 2017 — Corrigé

Ce corrigé est proposé par Maël Mevel (professeur agrégé) ; il a été relu par Antoine Sihrener (professeur en CPGE) et Céline Chevalier (enseignant-chercheur à l'université).

Ce problème utilise la marche aléatoire d'un rat dans un labyrinthe comme point de départ puis établit quelques propriétés des matrices dites stochastiques (du grec *stokhastês* : « lié au hasard »).

- La partie I propose de démontrer que sous certaines conditions initiales, la loi de la position du rat dans le labyrinthe ne varie pas avec le temps. On associe une matrice au graphe probabiliste présenté dans le sujet, matrice dont il sera montré plus tard que sa transposée est stochastique.
- La partie II s'intéresse à la convergence de la suite $(r_k)$ des moyennes de Césaro des puissances d'un endomorphisme $u$. On justifie notamment que lorsque $u$ est tel que $\|u(x)\| \leqslant \|x\|$ pour tout $x$ dans E, la suite converge vers un projecteur bien particulier. Le point de vue matriciel est également abordé.
- La partie III porte sur l'étude des matrices stochastiques en elles-mêmes ; on montre que les matrices $\mathrm{R}_k$ considérées dans la partie II sont stochastiques, et que ces dernières ont notamment la valeur propre évidente 1 pour valeur propre simple.
- Dans la partie IV, on applique les résultats des parties précédentes à l'étude de la loi de la variable aléatoire $\mathrm{S}_k$ donnant la position du rat à l'instant $k$. On montre en particulier qu'il existe une unique valeur pour $\mathrm{S}_0$ qui rende la suite $(\mathrm{S}_k)_{k\in\mathbb{N}}$ constante.

Ce problème peut être vu comme l'étude d'une chaîne de Markov (à espace d'états finis), c'est-à-dire une marche aléatoire dans laquelle un état au temps $n$ ne dépend que de l'état au temps $n-1$ (et du hasard). Il fait appel aux probabilités bien sûr, mais aussi à l'algèbre linéaire. Quelques questions demandent de savoir bien jongler entre le point de vue des endomorphismes et le point de vue matriciel.

Remarquons enfin que les matrices stochastiques figuraient déjà dans l'épreuve 1 des Mines, filière MP, en 2016, et qu'elles sont présentes dans de nombreux sujets d'oraux. Il s'agit d'un thème très à la mode depuis la réforme introduisant les probabilités en classe préparatoire il y a 3 ans.

# Indications

## Partie I

I.1 Montrer que $(S_k = i)_{i\in[\![1\,;5]\!]}$ est un système complet d'événements.

I.2 Généraliser la question précédente à $\mathbb{P}(S_{k+1} = i)$ pour $i \in [\![1\,;5]\!]$.

I.4 Raisonner par récurrence sur $k$.

I.5 Considérer par exemple $\mathbb{P}(S_0 = 1 \cap S_1 = 1)$.

## Partie II

II.6 S'intéresser à $u^\ell(x)$ pour $x \in \text{Ker}\,(u - I_E)$.

II.7 Justifier qu'il existe $y \in E$ tel que $x = u(y) - y$, puis injecter cela dans $r_k(x)$, et majorer la norme de cette dernière quantité.

II.8 En utilisant les questions II.6 et II.7, étudier la limite de $r_k(x)$ lorsque $x \in \text{Ker}\,(u - I_E) \bigcap \text{Im}\,(u - I_E)$. Conclure ensuite par dimension.

II.9 Traduire ce que signifie, pour un $x \in E$, la décomposition de E en somme directe démontrée précédemment.

II.10 Utiliser les résultats de la question II.9 en jonglant entre endomorphisme et matrice. Penser à une caractérisation de l'application $p$.

## Partie III

III.13 Utiliser la caractérisation séquentielle des fermés.

III.14 En tentant de majorer $\|AX\|_\infty$, faire apparaître $\displaystyle\sum_{j=1}^{n} |a_{i,j}| = \sum_{j=1}^{n} a_{i,j} = 1$.

III.15 Remarquer que la matrice $A^p$ est stochastique, puis raisonner par l'absurde en supposant l'existence d'un indice $i_0$ tel que $x_{i_0} < x_s$ et en utilisant l'égalité $A^pX = X$.

III.16 En exploitant le résultat de la question précédente, montrer que si $AX = X$ alors $A^pX = X$, puis $X \in \text{Vect}\,U$.

III.18 Combiner les résultats des questions III.10 et III.14 pour montrer que $R_k$ converge vers une matrice P telle que $P^2 = P$. Conclure en utilisant les questions III.13, III.16 et III.17.

III.19 S'intéresser aux lignes (ou aux colonnes) de P et exprimer un lien de multiplicité entre ces dernières à l'aide du rang de P. Utiliser la stochasticité de P (condition 4).

III.20 Pour tout $k \in \mathbb{N}^*$, exprimer $R_kA$ en fonction de $R_{k+1}$ et de la matrice identité, puis passer à la limite. En ce qui concerne l'unicité de L, appliquer le théorème du rang à deux matrices bien choisies, en remarquant que ${}^tL \in \text{Ker}\left({}^tA - I_n\right)$, puis utiliser la question III.16.

III.21 Raisonner par l'absurde. Utiliser l'hypothèse sur les coefficients de $A^p$ faite dans le sujet.

III.22 Établir une relation entre le polynôme caractéristique de $u$ et ceux des endomorphismes $u_K$ et $u_I$ induits par $u$ sur les sous-espaces vectoriels $\text{Ker}\,(u - I_E)$ et $\text{Im}\,(u - I_E)$. Etudier alors $\chi_{u_K}$.

## Partie IV

IV.24 L'existence d'une telle loi provient de la partie I. L'unicité est une conséquence de la question III.20.

# I. Premiers pas

**I.1** Soit $k \in \mathbb{N}$. Afin d'utiliser la formule des probabilités totales comme conseillé dans l'énoncé, montrons que la famille $(\mathrm{S}_k = i)_{i\in[\![1\,;5]\!]}$ constitue un système complet d'événements.

Comme le rat se trouve forcément dans une pièce à un instant donné,

$$\bigcup_{i=1}^{5} (\mathrm{S}_k = i) = \Omega$$

et puisque le rat ne peut se trouver en deux pièces au même instant, les événements $(\mathrm{S}_k = i)_{i\in[\![1\,;5]\!]}$ sont deux à deux incompatibles. Ainsi, la famille $(\mathrm{S}_k = i)_{i\in[\![1\,;5]\!]}$ constitue un système complet d'événements.

Appliquons alors la formule des probabilités totales. Il vient

$$\mathbb{P}(\mathrm{S}_{k+1} = 1) = \sum_{i=1}^{5} \mathbb{P}(\mathrm{S}_{k+1} = 1 | \mathrm{S}_k = i)\, \mathbb{P}(\mathrm{S}_k = i)$$

et ainsi

La quantité $\mathbb{P}(\mathrm{S}_{k+1} = 1)$ est une combinaison linéaire des $\big(\mathbb{P}(\mathrm{S}_k = i)\big)_{i\in[\![1\,;5]\!]}$.

> La manière dont cette question est posée est quelque peu étrange : en effet, on demande de montrer qu'un réel $(\mathbb{P}(\mathrm{S}_{k+1} = 1))$ est combinaison linéaire d'autres réels (les $(\mathbb{P}(\mathrm{S}_k = i))_{i\in[\![1\,;5]\!]}$)... Or, c'est toujours vrai dès l'instant où ces derniers ne sont pas tous nuls, ce qui est évident dans le cas présent ! Sans présumer des intentions des auteurs, on peut supposer qu'une formulation plus précise de la question serait « En utilisant la formule des probabilités totales, exprimer $\mathbb{P}(\mathrm{S}_{k+1} = 1)$ en fonction des $\mathbb{P}(\mathrm{S}_k = i)_{i\in[\![1\,;5]\!]}$. »

**I.2** En raisonnant de manière analogue à la question I.1, écrivons pour tout $k \in \mathbb{N}$ et pour tout $i \in [\![1\,;5]\!]$,

$$\mathbb{P}(\mathrm{S}_{k+1} = i) = \sum_{j=1}^{5} \mathbb{P}(\mathrm{S}_{k+1} = i | \mathrm{S}_k = j)\, \mathbb{P}(\mathrm{S}_k = j)$$

ce qui équivaut, en passant à l'écriture matricielle, à l'égalité

$$\mathrm{X}_{k+1} = \mathrm{B}\mathrm{X}_k$$

où les coefficients de la matrice B sont donnés par

$$\forall (i,j) \in [\![1\,;n]\!]^2 \quad b_{i,j} = \mathbb{P}(\mathrm{S}_{k+1} = i | \mathrm{S}_k = j)$$

Enfin, comme le rat se déplace de salle en salle avec équiprobabilité, on en déduit, en notant $n_j$ le nombre de salles adjacentes à la $j^e$ salle, que pour tout $(i,j) \in [\![1\,;n]\!]^2$,

$$b_{i,j} = \mathbb{P}(\mathrm{S}_{k+1} = i | \mathrm{S}_k = j) = \begin{cases} 1/n_j & \text{si les salles } i \text{ et } j \text{ sont adjacentes} \\ 0 & \text{sinon} \end{cases}$$

et ainsi

$$\mathrm{B} = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 1/3 \\ 1/4 & 0 & 1/3 & 0 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 & 0 \\ 1/4 & 0 & 1/3 & 0 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 & 0 \end{pmatrix}$$

> Si l'on souhaite se placer du point de vue des chaînes de Markov évoqué dans la présentation, c'est alors le moment d'évoquer le lien très particulier entre B et ladite chaîne : en effet, B est dite *matrice de transition* de la chaîne de Markov associée, et conjointement à la loi initiale, elle caractérise la chaîne. C'est par l'étude des puissances successives de la matrice de transition que l'on étudie la loi limite de la chaîne, et c'est d'ailleurs en quelque sorte la démarche qui est mise en œuvre dans ce sujet.
>
> De plus, il existe différents moyens de définir cette « matrice de transition » : il est en effet possible de la définir par la relation plus naturelle $\mathrm{B}_{i,j} = \mathbb{P}\left(\mathrm{S}_k = j | \mathrm{S}_{k-1} = i\right)$, ce qui permet d'obtenir directement une matrice stochastique, mais on passera alors d'un état du système à l'autre en multipliant B par le vecteur ligne ${}^t\mathrm{X}_k$ à gauche, ce qui est bien moins pratique.

**I.3** En s'intéressant aux coefficients de B, on remarque que pour tout $j \in [\![ 1\,;5 ]\!]$,

$$\sum_{i=1}^{5} b_{i,j} = 1$$

Ainsi, en notant $\mathrm{V} = {}^t(1,1,1,1,1)$, il vient pour tout $i \in [\![ 1\,;5 ]\!]$,

$$\begin{aligned}
\left({}^t\mathrm{B}\,\mathrm{V}\right)_i &= \sum_{j=1}^{5} \left({}^t\mathrm{B}\right)_{i,j} v_j \\
&= \sum_{j=1}^{5} b_{j,i} \qquad \text{car } v_j = 1 \text{ pour tout j} \in [\![ 1\,;5 ]\!] \\
\left({}^t\mathrm{B}\,\mathrm{V}\right)_i &= 1 \qquad\qquad\qquad\qquad \text{par la remarque précédente}
\end{aligned}$$

Par suite, ${}^t\mathrm{B}\,\mathrm{V} = \mathrm{V}$ et

> Le réel 1 est valeur propre de ${}^t\mathrm{B}$, associée au vecteur propre $\mathrm{V} = {}^t(1,1,1,1,1)$.

> C'est un résultat classique que si la somme de chaque ligne d'une matrice M vaut 1, alors ${}^t(1,1,\ldots,1)$ est un vecteur propre de M pour la valeur propre 1.

**I.4** Les $\mathrm{S}_k$ ont toutes la même loi si pour tout $n \in \mathbb{N}$, $\mathrm{X}_n = \mathrm{X}_0$. Montrons donc par récurrence que la propriété :

$$\mathscr{P}(n) : \quad \mathrm{X}_n = \mathrm{X}_0$$

est vraie pour tout $n \geqslant 0$.

- $\mathscr{P}(0)$ est vraie car $\mathrm{X}_0 = \mathrm{X}_0$.
- $\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)$ : Soit $n \in \mathbb{N}$. Supposons $\mathscr{P}(n)$ vraie et montrons qu'alors $\mathscr{P}(n+1)$ l'est aussi. D'après la question I.2 et l'hypothèse de récurrence,

$$\mathrm{X}_{n+1} = \mathrm{B}\mathrm{X}_n = \mathrm{B}\mathrm{X}_0$$

Calculons alors $\mathrm{B}\mathrm{X}_0$ :

$$\mathrm{B}\mathrm{X}_0 = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 1/3 \\ 1/4 & 0 & 1/3 & 0 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 & 0 \\ 1/4 & 0 & 1/3 & 0 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 & 0 \end{pmatrix} \begin{pmatrix} 1/4 \\ 3/16 \\ 3/16 \\ 3/16 \\ 3/16 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 3/16 \\ 3/16 \\ 3/16 \\ 3/16 \end{pmatrix} = \mathrm{X}_0$$

# Mines Maths 2 PC 2017 — Corrigé

Ce corrigé est proposé par Émilie Liboz (professeur en CPGE) ; il a été relu par Sophie Rainero (professeur en CPGE) et Benjamin Monmege (enseignant-chercheur à l'université).

---

Ce problème développe dans différents cadres un cas particulier de la méthode de Laplace, qui permet l'étude asymptotique de fonctions définies par des intégrales. Elle permet dans cette épreuve d'obtenir des équivalents d'une somme partielle, de l'espérance d'une variable aléatoire, ou encore de redémontrer la formule de Stirling.

Le sujet est découpé en cinq parties qui balayent une grande part du programme de PC puisqu'il est question de séries numériques, d'intégrales, de suites de fonctions et de variables aléatoires discrètes.

La première partie s'intéresse au reste $\mathrm{R}_n(x)$ et à la somme partielle $\mathrm{T}_n(x)$ de la série exponentielle $\displaystyle\sum_{k\geqslant 0}\frac{n^k x^k}{k\,!}$. On étudie le comportement asymptotique de $\mathrm{T}_n(x)$ quand $n$ tend vers $+\infty$ pour $x>0$ différent de 1. Les questions présentes dans cette partie sont assez classiques ; la question 7 est plutôt technique.

Pour obtenir un équivalent de $\mathrm{T}_n(1)$ à la partie 4, on a besoin du résultat suivant, démontré à la partie 2 : si $f$ est une fonction $\mathscr{C}^2$ définie sur $[\,-1\,;1\,]$, à valeurs dans $[\,0\,;1\,]$, et qui atteint son maximum 1 uniquement au point 0 avec la condition $f''(0)=-1$, alors on a l'équivalent

$$\int_{-1}^{1}(f(x))^n\,\mathrm{d}x \underset{n\to+\infty}{\sim}\sqrt{\frac{2\pi}{n}}$$

C'est un cas particulier de la méthode de Laplace qui étudie le comportement asymptotique des fonctions de la forme $t\mapsto\displaystyle\int_a^b \mathrm{e}^{-t\varphi(x)}f(x)\,\mathrm{d}x$, où la fonction $\varphi$ atteint généralement son minimum en un unique point qui ne doit pas être un zéro de $f$. La technique se généralise pour les fonctions de la forme $t\mapsto\displaystyle\int_a^b \mathrm{e}^{-\mathrm{i}t\varphi(x)}f(x)\,\mathrm{d}x$, portant cette fois de nom de méthode de la phase stationnaire. Ces outils s'étendent à des intégrales multiples et à des intégrales de la variable complexe ; elles ont des applications en physique, notamment en mécanique statistique.

On applique ensuite la formule de la partie 2 à une fonction particulière dans les parties 3 et 4 pour démontrer la formule de Stirling puis la formule de Bernstein, $\mathrm{T}_n(1)\underset{n\to+\infty}{\sim}\mathrm{e}^n/2$. Celle-ci permettra dans la partie 5 d'évaluer le comportement asymptotique de l'espérance de la première répétition dans le cas de tirages avec remise d'une boule dans une urne qui en contient $n$ distinctes.

Ce sujet plutôt varié est intéressant car il permet de découvrir et d'appliquer une méthode classique d'analyse numérique dans un cas particulier, tout en restant dans le cadre du programme. De plus, les résultats clés étaient donnés pour permettre aux candidats de ne pas rester bloqués. La partie 2 est la plus difficile et peut être admise dans un premier temps, afin d'observer comment elle s'applique dans la suite du sujet.

# Indications

## Partie I

1 Reconnaître la somme partielle et le reste d'une série usuelle.

3 Penser à la règle de D'Alembert pour conclure.

5 Utiliser un raisonnement par récurrence.

6 Reprendre les résultats des questions 1, 2 et 5.

7 Suivre l'indication de l'énoncé pour majorer $\mathrm{T}_n(x)$.

## Partie II

8 Montrer que $f$ admet un maximum en 0.

9 Distinguer les cas où $\varphi$ est prolongeable ou non par continuité sur $[\,-1\,;1\,]$.

10 Utiliser le développement limité de $f$ en 0 obtenu à la question 8.

11 Penser au théorème de convergence dominée.

## Partie III

12 Effectuer un ou plusieurs changements de variable dans l'intégrale $\mathrm{I}_n + \mathrm{J}_n$ pour se ramener à la question 5.

13 On peut composer l'inégalité par la fonction ln pour la démontrer plus facilement.

15 Montrer à l'aide des questions 13 et 14 que $\mathrm{I}_n + \mathrm{J}_n \underset{n\to+\infty}{\sim} \mathrm{I}_n$.

## Partie IV

17 Appliquer le résultat de la question 11 à une fonction bien choisie.

## Partie V

19 Trouver une issue qui réalise $(\mathrm{X} = k)$.

20 Comparer les événements $(\mathrm{X} > k+1)$ et $(\mathrm{X} > k)$.

21 Démontrer cette formule par récurrence à l'aide de la question 20.

22 Relier les événements $(\mathrm{X} = k)$, $(\mathrm{X} > k+1)$ et $(\mathrm{X} > k)$.

# I. Exponentielle tronquée

**1** $\mathrm{R}_n(x)$ est le reste d'indice $n$ de la série $\displaystyle\sum_{k\geqslant 0}\frac{(nx)^k}{k\,!}$. On reconnaît la série exponentielle de paramètre $nx$, qui converge pour tout $(x,n)\in\,]\,0\,;+\infty\,[\,\times\mathbb{N}$ et dont la somme vaut $\mathrm{e}^{\,nx}$. De plus, $\mathrm{T}_n(x)$ est la somme partielle d'indice $n$ de cette même série. Ainsi

$$\boxed{\mathrm{R}_n(x)\text{ existe}\qquad\text{et}\qquad \mathrm{T}_n(x)+\mathrm{R}_n(x)=\mathrm{e}^{\,nx}.}$$

**2** Soient $x>0$ et $n\in\mathbb{N}$. En appliquant la formule de Taylor avec reste intégral à la fonction $f:t\longmapsto \mathrm{e}^{\,nt}$ qui est de classe $\mathscr{C}^\infty$ sur $[\,0\,;x\,]$, et dont les dérivées successives sont données par $f^{(k)}(t)=n^k\mathrm{e}^{\,nt}$, pour tout $(t,k)\in[\,0\,;x\,]\times\mathbb{N}$, on obtient

$$\mathrm{e}^{\,nx}=\sum_{k=0}^{n}\frac{n^k}{k\,!}x^k+\int_0^x n^{n+1}\mathrm{e}^{\,nt}\frac{(x-t)^n}{n\,!}\,\mathrm{d}t$$

Or, d'après la question 1, $\mathrm{R}_n(x)=\mathrm{e}^{\,nx}-\displaystyle\sum_{k=0}^{n}\frac{n^k}{k\,!}x^k$, d'où

$$\mathrm{R}_n(x)=\frac{n^{n+1}}{n\,!}\int_0^x\mathrm{e}^{\,nt}(x-t)^n\,\mathrm{d}t=\mathrm{e}^{\,nx}\frac{n^{n+1}}{n\,!}\int_0^x\mathrm{e}^{\,n(t-x)}(x-t)^n\,\mathrm{d}t$$

Pour retrouver la formule donnée dans l'énoncé, il suffit de poser le changement de variable $u=x-t$ dans l'intégrale obtenue. La fonction $t\mapsto x-t$ est de classe $\mathscr{C}^1$ sur $[\,0\,;x\,]$ et $\displaystyle\int_0^x\mathrm{e}^{\,n(t-x)}(x-t)^n\,\mathrm{d}t=\int_0^x\mathrm{e}^{\,-nu}u^n\,\mathrm{d}u$. Ainsi

$$\boxed{\mathrm{R}_n(x)=\mathrm{e}^{\,nx}\frac{n^{n+1}}{n\,!}\int_0^x\left(u\mathrm{e}^{\,-u}\right)^n\,\mathrm{d}u}$$

> Rappelons la formule de Taylor avec reste intégral pour une fonction $f$ de classe $\mathscr{C}^{n+1}$ sur un segment $[\,a\,;b\,]$ :
>
> $$f(b)=\sum_{k=0}^{n}\frac{f^{(k)}(a)}{k\,!}(b-a)^k+\int_a^b f^{(n+1)}(t)\frac{(b-t)^n}{n\,!}\,\mathrm{d}t$$

**3** Par définition, $a_n>0$ pour tout $n\in\mathbb{N}$ et

$$\frac{a_{n+1}}{a_n}=\frac{(n+1)^{n+2}}{(n+1)\,!}y^{n+1}\times\frac{n\,!}{n^{n+1}y^n}=\frac{(n+1)^{n+2}}{(n+1)n^{n+1}}y=\left(\frac{n+1}{n}\right)^{n+1}y$$

Or $\displaystyle\left(\frac{n+1}{n}\right)^{n+1}=\frac{n+1}{n}\left(\frac{n+1}{n}\right)^n\underset{n\to+\infty}{\sim}\left(\frac{n+1}{n}\right)^n$ et

$$\left(\frac{n+1}{n}\right)^n=\left(1+\frac{1}{n}\right)^n=\mathrm{e}^{\,n\ln\left(1+\frac{1}{n}\right)}=\mathrm{e}^{\,n\left(1/n+\underset{n\to+\infty}{\mathrm{o}}(1/n)\right)}=\mathrm{e}^{\,1+\underset{n\to+\infty}{\mathrm{o}}(1)}$$

D'où $$\boxed{\lim_{n\to+\infty}\frac{a_{n+1}}{a_n}=\mathrm{e}\,y}$$

Ainsi, si $y<\mathrm{e}^{\,-1}$, alors $\displaystyle\lim_{n\to+\infty}\frac{a_{n+1}}{a_n}<1$ et, d'après la règle de D'Alembert, la série numérique $\sum a_n$ converge donc

$$\boxed{\text{Si } y<\mathrm{e}^{\,-1}\text{ alors }\lim_{n\to+\infty}a_n=0.}$$

**4** La fonction $g : u \longmapsto u\mathrm{e}^{-u}$ est de classe $\mathscr{C}^1$ sur $\mathbb{R}$ de dérivée $g' : u \longmapsto (1-u)\mathrm{e}^{-u}$. Ainsi, pour tout $u \in [\,0\,;1\,[$, $g'(u) > 0$ donc la fonction $g$ est strictement croissante sur l'intervalle $[\,0\,;1\,]$. Par conséquent, si $x \in ]\,0\,;1\,[$, alors $g$ est croissante sur $[\,0\,;x\,]$. Elle admet donc un maximum M atteint en $u = x$. Ainsi,

$$\mathrm{M} = g(x) = x\mathrm{e}^{-x} < g(1) = \mathrm{e}^{-1}$$

On a donc montré que

$$\boxed{\begin{array}{l}\text{Si } x \in ]\,0\,;1\,[\text{, alors la fonction } u \mapsto u\mathrm{e}^{-u} \text{ admet}\\ \text{un maximum M sur } [\,0\,;x\,] \text{ tel que } \mathrm{M} < \mathrm{e}^{-1}.\end{array}}$$

Soit $x \in ]\,0\,;1\,[$. Les inégalités $0 \leqslant u\mathrm{e}^{-u} < \mathrm{M}$ pour tout $u \in [\,0\,;x\,]$ impliquent, par croissance de l'intégrale :

$$0 \leqslant \int_0^x (u\mathrm{e}^{-u})^n \,\mathrm{d}u \leqslant \int_0^x \mathrm{M}^n \,\mathrm{d}u = \mathrm{M}^n\, x \leqslant \mathrm{M}^n$$

d'où, d'après la question 2,

$$0 \leqslant \mathrm{R}_n(x) \leqslant \mathrm{e}^{nx}\frac{n^{n+1}}{n\,!}\mathrm{M}^n$$

La question 3 permet d'affirmer que, comme $\mathrm{M} < \mathrm{e}^{-1}$, $\lim\limits_{n\to+\infty} \dfrac{n^{n+1}}{n\,!}\mathrm{M}^n = 0$ d'où

$$\boxed{\mathrm{R}_n(x) = \underset{n\to+\infty}{\mathrm{o}}\left(\mathrm{e}^{nx}\right) \qquad \text{si } x \in ]\,0\,;1\,[}$$

Mais d'après la question 1, $\mathrm{T}_n(x) = \mathrm{e}^{nx} - \mathrm{R}_n(x) = \mathrm{e}^{nx} + \underset{n\to+\infty}{\mathrm{o}}\left(\mathrm{e}^{nx}\right)$, par conséquent

$$\boxed{\mathrm{T}_n(x) \underset{n\to+\infty}{\sim} \mathrm{e}^{nx} \qquad \text{si } x \in ]\,0\,;1\,[}$$

**5** Soit $n \in \mathbb{N}$. Commençons par justifier la convergence de l'intégrale $\displaystyle\int_0^{+\infty} t^n\mathrm{e}^{-t}\,\mathrm{d}t$ :

- la fonction $t \mapsto t^n\mathrm{e}^{-t}$ est continue sur $[\,0\,;+\infty\,[$ donc intégrable sur $[\,0\,;1\,]$ ;
- de plus, $t^n\mathrm{e}^{-t} = \underset{t\to+\infty}{\mathrm{o}}\left(1/t^2\right)$ et $\displaystyle\int_1^{+\infty} \mathrm{d}t/t^2$ converge donc, par comparaison, $\displaystyle\int_1^{+\infty} t^n\mathrm{e}^{-t}\,\mathrm{d}t$ converge.

Montrons ensuite la relation $\mathscr{P}(n)$ : $n\,! = \displaystyle\int_0^{+\infty} t^n\mathrm{e}^{-t}\,\mathrm{d}t$ par récurrence sur $n \in \mathbb{N}$ :

- $\underline{\mathscr{P}(0)}$ est vraie car $\displaystyle\int_0^{+\infty} \mathrm{e}^{-t}\,\mathrm{d}t = \lim_{x\to+\infty}\int_0^x \mathrm{e}^{-t}\,\mathrm{d}t = \lim_{x\to+\infty}(1-\mathrm{e}^{-x}) = 1$.
- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$ : en posant $u(t) = t^{n+1}$ et $v(t) = -\mathrm{e}^{-t}$, on définit deux fonctions de classe $\mathscr{C}^1$ sur $\mathbb{R}$ qui vérifient $u'(t) = (n+1)t^n$ et $v'(t) = \mathrm{e}^{-t}$. De plus, par croissances comparées $\lim\limits_{t\to+\infty} u(t)v(t) = u(0)v(0) = 0$. On obtient ainsi, par une intégration par parties :

  $$\int_0^{+\infty} t^{n+1}\mathrm{e}^{-t}\,\mathrm{d}t = \int_0^{+\infty} u(t)v'(t)\,\mathrm{d}t = (n+1)\int_0^{+\infty} t^n\mathrm{e}^{-t}\,\mathrm{d}t$$

  Ainsi, si $\mathscr{P}(n)$ est vraie, alors $\displaystyle\int_0^{+\infty} t^{n+1}\mathrm{e}^{-t}\,\mathrm{d}t = (n+1)n\,! = (n+1)\,!$ et $\mathscr{P}(n+1)$ est vraie.
- $\underline{\text{Conclusion}}$ : $\boxed{\forall\, n \in \mathbb{N} \qquad n\,! = \displaystyle\int_0^{+\infty} t^n\mathrm{e}^{-t}\,\mathrm{d}t}$

# Mines Informatique PC 2017 — Corrigé

Ce corrigé est proposé par Virgile Andreani (ENS Ulm) ; il a été relu par Cyril Ravat (Professeur en CPGE) et Julien Dumont (Professeur en CPGE).

Ce sujet aborde la modélisation d'un croisement routier, représenté par deux files de voitures qui s'intersectent.

- Dans la première partie, il pose les bases de la modélisation avec le cas d'une file unique.
- Dans la deuxième, il invite à programmer la dynamique de la file en présence ou non d'un obstacle sur la voie, à l'aide d'une opération élémentaire, fournie par une fonction déjà écrite, qui consiste à faire avancer toute la file d'un coup.
- Dans une troisième partie qui ne comporte que deux questions, on s'intéresse enfin au cas du croisement, en simulant conjointement deux files couplées par une case commune.
- La quatrième partie, courte elle aussi, prépare la suivante en introduisant la notion de transition entre configurations non immédiatement successives.
- L'avant-dernière partie, la moins facile de l'épreuve, est consacrée au codage d'un algorithme de recherche en largeur sur les états du système ce qui permet de déterminer si une configuration est atteignable depuis une autre.
- Enfin, l'épreuve se termine par trois questions utilisant les bases de données.

La difficulté de ce problème est progressive, avec de nombreuses questions très accessibles en début et milieu d'épreuve, suivies d'autres plus délicates en fin d'épreuve, notamment dans la cinquième partie. Il est accessible dès la première année (sauf la question 18). L'ensemble du programme est couvert à l'exception de la simulation numérique.

# Indications

## Partie II

9 La fonction `avancer` peut servir d'opération élémentaire dans la fonction demandée, il faut s'en servir. On peut faire appel à la concaténation de listes au moyen de l'opérateur `+`.

10 Noter que la case $m$ est inoccupée. Que cela implique-t-il sur les voitures à gauche de $m$ ?

11 Étant donné que les voitures immédiatement à gauche de $m$ sont bloquées, où sont les voitures qui ont la possibilité d'avancer ?

## Partie III

12 Bien réfléchir à l'ordre dans lequel appeler les trois fonctions précédemment définies `avancer_debut`, `avancer_debut_bloque` et `avancer_fin`.

## Partie V

17 Noter que la liste est triée : il suffit donc de comparer chaque valeur à la précédente pour déterminer si elle est nouvelle ou non.

20 Chercher dans le code où apparaissent les variables en question et les opérations dans lesquelles elles sont impliquées pour déterminer leur type.

21 Quel est le principal critère de choix d'un algorithme ?

24 Démontrer que le nombre d'éléments uniques contenus dans la liste `espace` est croissant et borné.

# I. Préliminaires

**1** Soit une case est vide, soit elle contient une voiture. On peut donc encoder ces deux états au moyen d'une variable binaire qui vaut `True` si une voiture est présente à cet endroit et `False` sinon. Pour représenter $n$ cases, on utilise une liste de $n$ de ces booléens.

**2** On commence par créer une liste vide, puis on remplit les cases qui doivent l'être :

```
A = 11*[False]
A[0] = True
A[2] = True
A[3] = True
A[10] = True
```

> Il vaut mieux éviter ici de construire la liste directement avec `A = [True, False, True, True, etc.]` pour réduire les risques d'erreur.

**3** Vu la manière dont on a encodé les états des cases, la fonction `occupe` ne renvoie rien d'autre que la valeur booléenne de la case demandée :

```
def occupe(L, i):
    return L[i]
```

> Il faut prendre soin à partir d'ici d'utiliser uniquement cette fonction pour vérifier l'état des cases et éviter d'indexer directement les listes. En effet, si on décidait un jour de changer de représentation pour les états des cases, il suffirait d'adapter cette fonction une fois au lieu de chercher dans le code tous les endroits où l'on indexe une liste.

**4** Chaque case pouvant être dans deux états différents, et ce de manière indépendante des autres cases,

> Le nombre de files différentes est le produit du nombre d'états possibles pour chacune des cases soit $2^n$

.

**5** Deux listes sont égales si et seulement si elles ont la même longueur et chacune de leurs cases égales deux à deux. Par conséquent :

```
def egal(L1, L2):
    if len(L1) != len(L2):
        return False
    for i in range(len(L1)):
        if L1[i] != L2[i]:
            return False
    return True
```

> On fait le test de longueur en premier pour s'assurer que les indices seront valides dans la boucle. À l'intérieur de celle-ci, on peut retourner `False` dès la première paire de valeurs différentes, inutile de poursuivre.
>
> Python permet de comparer directement deux listes avec l'opérateur `==`, mais écrire `L1 == L2` ici annule l'intérêt de la question et ne fait pas apparaître explicitement la complexité.

**6** Si les listes sont de longueur différente, il n'y a pas de boucle et la fonction s'achève en temps constant. Le pire cas correspond à la situation où l'on itère la boucle un maximum de fois, c'est-à-dire quand les listes sont de même taille et que tous leurs éléments sont égaux deux à deux (sauf éventuellement les deux derniers). Dans ce cas,

La complexité est linéaire en la longueur commune des listes

**7** Comme il était demandé à la question 5, cette fonction retourne un booléen.

## II. Déplacement de voitures dans la file

**8** Cette évolution consiste à faire avancer la file deux fois, d'abord sans ajouter de voiture au début, puis en en ajoutant une. On obtient donc `[True, False, True, False, True, True, False, False, False, False, False]`.

**9** Cette étape partielle consiste à faire avancer la deuxième partie de la liste et à laisser la première inchangée. On peut réutiliser la fonction `avancer` pour la deuxième partie et recoller le résultat avec la première partie de la liste au moyen de l'opérateur de concaténation `+`.

```
def avancer_fin(L, m):
    return L[:m] + avancer(L[m:], False)
```

> Pour cette question comme pour les suivantes, l'énoncé incite à réutiliser la fonction `avancer`, qu'il définit mais n'utilise pas ailleurs. Une réponse acceptable sans utiliser cette fonction aurait pu être
>
> ```
> def avancer_fin(L, m):
>     return L[:m] + [False] + L[m:-1]
> ```
>
> ou encore, sans concaténation :
>
> ```
> def avancer_fin(L, m):
>     L1 = L[:m]
>     L1.append(False)
>     for i in range(m, len(L)-1):
>         L1.append(L[i])
>     return L1
> ```

**10** La case $m$ étant inoccupée, les voitures à gauche de $m$ ne sont pas bloquées et peuvent toutes avancer. On peut donc là aussi réutiliser la fonction `avancer` sur la première partie de la liste et recoller le résultat avec la seconde partie.

```
def avancer_debut(L, b, m):
    return avancer(L[:m+1], b) + L[m+1:]
```

**11** Les voitures bloquées immédiatement à la gauche de $m$ ne bougent pas, de la même manière que les voitures à la droite de $m$. Toutes les voitures à la droite de la première case inoccupée à la gauche de $m$ (si elle existe) voient donc leur position inchangée par `avancer_debut_bloque`, seules les voitures à la gauche de cette case

# X/ENS Maths PC 2017 — Corrigé

Ce corrigé est proposé par Tristan Poullaouec (professeur en CPGE) ; il a été relu par Loïc Devilliers (ENS Cachan) et Benoit Chevalier (ENS Ulm).

---

Ce problème d'algèbre linéaire traite du rayon spectral des matrices de $\mathscr{M}_n(\mathbb{C})$ et a pour objet la démonstration d'une version d'un théorème dit de Perron-Frobenius, qui est notamment utilisé en probabilités. Il est composé de trois parties qui peuvent être abordées indépendamment les unes des autres car tous les résultats utiles sont mentionnés dans l'énoncé.

- Dans une première partie, on munit l'espace $\mathscr{M}_n(\mathbb{C})$ d'une norme dont on établit quelques propriétés algébriques et topologiques.
- On s'attaque ensuite au rayon spectral proprement dit dans la deuxième partie. Après avoir étudié quelques exemples simples et des propriétés basiques, on fait le lien avec la norme définie dans la première partie.
- La troisième partie, enfin, permet de démontrer quatre propriétés du rayon spectral d'une matrice à coefficients strictement positifs (c'est en particulier une de ses valeurs propres) et de l'espace propre associé.

Ce sujet ne présente pas de grosse difficulté et utilise peu de résultats du cours : si l'on connaît les définitions d'une norme, d'une borne supérieure, du produit matriciel, d'une valeur propre et d'un vecteur propre, on dispose d'à peu près tous les outils permettant de traiter les questions proposées. Sa longueur permet de plus de le traiter dans le temps imparti. Il permet de vérifier que l'on a bien compris les outils fondamentaux et donne l'occasion de les manipuler sans intervention des gros théorèmes. C'est donc un bon sujet de révision d'algèbre et de topologie, abordable tôt dans l'année et de difficulté mesurée, surtout pour le concours X/ENS.

# Indications

## Première partie

1.a La borne supérieure d'un ensemble est le plus petit de ses majorants.

1.b Utiliser l'équivalence établie à la question 1.a.

3 Montrer, en appliquant la question 1.a, que $\alpha = \underset{1\leqslant j\leqslant n}{\mathrm{Max}} \sum_{i=1}^{n} |a_{ij}|$ majore $\|\mathrm{A}\|$, puis exhiber un vecteur $x \in \mathbb{C}^n$ tel que $\|x\|_1 = 1$ et $\|\mathrm{A}x\|_1 = \alpha$.

5.a Quel est l'effet, sur les lignes et les colonnes d'une matrice, de la multiplication à gauche ou à droite par une matrice diagonale ?

5.b Penser à la continuité de la norme dans un espace vectoriel normé.

5.c Utiliser les résultats des questions 5.a et 2.

## Deuxième partie

7 Pour montrer que les assertions ii) et iii) sont fausses, on pourra prendre des matrices A et B de rayon spectral nul, en s'inspirant de la question 6.

9 Se servir du résultat de la question 5.c.

10.b Raisonner par double inclusion et utiliser les résultats des questions 10.a et 9.

11 On pourra encadrer $\|\mathrm{A}^k\|^{1/k}$ entre $\rho(\mathrm{A})$ et $\rho(\mathrm{A})+\varepsilon$ pour tout $\varepsilon > 0$, en utilisant les propriétés établies aux questions 10.a et 10.b.

12 Prouver d'abord, grâce au résultat de la question 3, que $\|\mathrm{AB}\| \leqslant \|\mathrm{A}_+\mathrm{B}_+\|$ pour toutes matrices A et B $\in \mathscr{M}_n(\mathbb{C})$. Appliquer ensuite la question 11.

## Troisième partie

13 Utiliser le cas d'égalité dans l'inégalité triangulaire, en notant que la colinéarité des vecteurs équivaut à l'existence d'un réel $\theta$ tel que $z_j = \mathrm{e}^{\,\mathrm{i}\,\theta}\,|z_j|$ pour tout $j$.

14 Calculer ${}^t x\,{}^t\mathrm{A}\,y$ de deux manières différentes.

15.a Penser à utiliser le résultat de la question 11.

15.b Trouver $\mu' > \mu$ tel que $\mathrm{A}w \geqslant \mu' w$, puis utiliser le résultat de la question 15.a.

16.a On se servira du résultat de la question 15.c pour montrer que $\mathrm{A}v_0 \leqslant \rho(\mathrm{A})v_0$.

16.b Un calcul direct permet d'établir que $\mathrm{A}v_0 > 0$.

16.c Calculer $|(\mathrm{A}x)_1|$ et $(\mathrm{A}v_0)_1$, puis utiliser la propriété établie à la question 13.

17.b Penser au résultat de la question 14.

18.a S'inspirer de ce qui a été fait à la question 16, en utilisant les résultats des questions 13, 15.b, 16.a et 17.a.

18.b On pourra utiliser une base de F pour écrire matriciellement $\mathrm{A}^k x$, puis utiliser les résultats des questions 18.a et 10.b.

18.c Se servir des propriétés démontrées aux questions 17.a et 18.b.

## Première partie

**1.a** Soient $\mathrm{C} \in \mathbb{R}_+$ et $\mathrm{M} \in \mathscr{M}_n(\mathbb{C})$. Notons

$$\mathcal{E} = \left\{ \frac{\|\mathrm{M}x\|_1}{\|x\|_1} \;\middle/\; x \in \mathbb{C}^n \smallsetminus \{0\} \right\}$$

> L'énoncé demande d'établir la propriété pour $\mathrm{C} > 0$, mais elle est aussi vraie pour $\mathrm{C} = 0$. Cela s'avérera utile au cours de la question 1.b.

Le réel $\|\mathrm{M}\| = \sup(\mathcal{E})$ est le plus petit des majorants de $\mathcal{E}$, de sorte que

$$\begin{aligned} \|\mathrm{M}\| \leqslant \mathrm{C} &\iff \mathrm{C} \text{ est un majorant de } \mathcal{E} \\ &\iff \forall\, x \in \mathbb{C}^n \smallsetminus \{0\},\ \frac{\|\mathrm{M}x\|_1}{\|x\|_1} \leqslant \mathrm{C} \\ &\iff \forall\, x \in \mathbb{C}^n \smallsetminus \{0\},\ \|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1 \end{aligned}$$

$$\|\mathrm{M}\| \leqslant \mathrm{C} \iff \forall\, x \in \mathbb{C}^n,\ \|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1$$

car $\|\mathrm{M}x\|_1$ et $\mathrm{C}\|x\|_1$ sont nuls lorsque $x = 0$. Ceci prouve que

$$\boxed{\forall\, \mathrm{M} \in \mathscr{M}_n(\mathbb{C})\ \ \forall\, \mathrm{C} \geqslant 0 \qquad \|\mathrm{M}\| \leqslant \mathrm{C} \iff \forall\, x \in \mathbb{C}^n,\ \|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1}$$

En particulier avec $\mathrm{C} = \|\mathrm{M}\|$, il vient

$$\forall\, \mathrm{M} \in \mathscr{M}_n(\mathbb{C}) \quad \forall\, x \in \mathbb{C}^n \qquad \|\mathrm{M}x\|_1 \leqslant \|\mathrm{M}\|\,\|x\|_1$$

> On peut aussi démontrer l'équivalence recherchée par double implication.
>
> - Supposons que $\|\mathrm{M}\| \leqslant \mathrm{C}$. Soit $x \in \mathbb{C}^n$ : si $x = 0$, alors $\|\mathrm{M}x\|_1 = 0$ et $\mathrm{C}\|x\|_1 = 0$ d'où $\|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1$. Sinon, il vient par définition de $\|\mathrm{M}\|$
>
> $$\frac{\|\mathrm{M}x\|_1}{\|x\|_1} \leqslant \|\mathrm{M}\| \leqslant \mathrm{C}$$
>
> ce qui mène une fois de plus à $\|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1$.
>
> - Réciproquement, supposons que $\|\mathrm{M}x\|_1 \leqslant \mathrm{C}\|x\|_1$ pour tout $x \in \mathbb{C}^n$. Alors $\|\mathrm{M}x\|_1 \leqslant \mathrm{C}$ pour $\|x\|_1 = 1$ si bien que $\|\mathrm{M}\| \leqslant \mathrm{C}$ puisque la borne supérieure d'un ensemble est le plus petit de ses majorants.

**1.b** Pour être une norme, une application doit vérifier trois propriétés : la séparation, l'homogénéité et l'inégalité triangulaire.

> L'application $x \longmapsto \|x\|_1$ étant une norme sur $\mathbb{C}^n$, elle vérifie ces propriétés. On va pouvoir les utiliser dans ce qui suit.

- *Séparation*

  Soit $\mathrm{M} \in \mathscr{M}_n(\mathbb{C})$ une matrice telle que $\|\mathrm{M}\| = 0$. Pour tout $x \in \mathbb{C}^n$, on a $\|\mathrm{M}x\|_1 = 0$ d'après le résultat ci-dessus, si bien que $\mathrm{M}x = 0$. Ainsi, la matrice M représente l'endomorphisme nul de $\mathbb{C}^n$ et $\mathrm{M} = 0$. Ceci montre que

  $$\forall\, \mathrm{M} \in \mathscr{M}_n(\mathbb{C}) \qquad \|\mathrm{M}\| = 0 \Longrightarrow \mathrm{M} = 0$$

- *Homogénéité*

  Soient $\lambda \in \mathbb{C}$ et $\mathrm{M} \in \mathscr{M}_n(\mathbb{C})$. D'après le résultat de la question 1.a, on a

  $$\forall\, x \in \mathbb{C}^n \qquad \|\lambda \mathrm{M}x\|_1 = |\lambda|\,\|\mathrm{M}x\|_1 \leqslant |\lambda|\,\|\mathrm{M}\|\,\|x\|_1$$

soit $\|\lambda \mathrm{M}\| \leqslant |\lambda| \, \|\mathrm{M}\|$. Si $\lambda = 0$, cette inégalité est évidemment une égalité car les deux membres sont nuls. Si $\lambda \neq 0$, on établit de la même façon que

$$\|\mathrm{M}\| = \left\| \frac{1}{\lambda} (\lambda \mathrm{M}) \right\| \leqslant \frac{1}{|\lambda|} \|\lambda \mathrm{M}\|$$

ce qui conduit à $|\lambda| \, \|\mathrm{M}\| \leqslant \|\lambda \mathrm{M}\|$. On en déduit que

$$\forall \, \lambda \in \mathbb{C} \quad \forall \, \mathrm{M} \in \mathscr{M}_n(\mathbb{C}) \qquad \|\lambda \mathrm{M}\| = |\lambda| \, \|\mathrm{M}\|$$

- *Inégalité triangulaire*

  Soient $\mathrm{M} \in \mathscr{M}_n(\mathbb{C})$ et $\mathrm{N} \in \mathscr{M}_n(\mathbb{C})$. En utilisant une nouvelle fois le résultat de la question 1.a et l'inégalité triangulaire pour la norme $\| \ \|_1$, on obtient

  $$\|(\mathrm{M}+\mathrm{N})x\|_1 = \|\mathrm{M}x + \mathrm{N}x\|_1 \leqslant \|\mathrm{M}x\|_1 + \|\mathrm{N}x\|_1 \leqslant (\|\mathrm{M}\| + \|\mathrm{N}\|) \, \|x\|_1$$

  pour tout $x \in \mathbb{C}^n$, d'où $\|\mathrm{M}+\mathrm{N}\| \leqslant \|\mathrm{M}\| + \|\mathrm{N}\|$. De ce fait,

  $$\forall \, (\mathrm{M}, \mathrm{N}) \in \mathscr{M}_n(\mathbb{C})^2 \qquad \|\mathrm{M}+\mathrm{N}\| \leqslant \|\mathrm{M}\| + \|\mathrm{N}\|$$

Les trois propriétés requises étant vérifiées, on peut affirmer que

$$\boxed{\text{L'application } \mathrm{M} \longmapsto \|\mathrm{M}\| \text{ est une norme sur } \mathscr{M}_n(\mathbb{C}).}$$

**2** Soient A et $\mathrm{B} \in \mathscr{M}_n(\mathbb{C})$. D'après le résultat de la question 1.a,

$$\forall \, x \in \mathbb{C}^n \qquad \|\mathrm{AB}x\|_1 = \|\mathrm{A}(\mathrm{B}x)\|_1 \leqslant \|\mathrm{A}\| \, \|\mathrm{B}x\|_1 \leqslant \|\mathrm{A}\| \, \|\mathrm{B}\| \, \|x\|_1$$

si bien que $\|\mathrm{AB}\| \leqslant \|\mathrm{A}\| \, \|\mathrm{B}\|$. Par conséquent,

$$\boxed{\forall \, (\mathrm{A}, \mathrm{B}) \in \mathscr{M}_n(\mathbb{C})^2 \qquad \|\mathrm{AB}\| \leqslant \|\mathrm{A}\| \, \|\mathrm{B}\|}$$

**3** Posons $\alpha = \underset{1 \leqslant j \leqslant n}{\mathrm{Max}} \displaystyle\sum_{i=1}^{n} |a_{ij}|$. Soit $x \in \mathbb{C}^n$ un vecteur de coordonnées $(x_1, \ldots, x_n)$ : celles de $\mathrm{A}x$ sont alors $(\mathrm{A}x)_i = \displaystyle\sum_{j=1}^{n} a_{ij} x_j$ pour $i \in [\![ 1 \, ; n ]\!]$. De ce fait,

$$\|\mathrm{A}x\|_1 = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \leqslant \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij} x_j| = \sum_{j=1}^{n} \left( \sum_{i=1}^{n} |a_{ij}| \, |x_j| \right)$$

d'où

$$\|\mathrm{A}x\|_1 \leqslant \sum_{j=1}^{n} \left( \sum_{i=1}^{n} |a_{ij}| \right) |x_j| \leqslant \alpha \sum_{j=1}^{n} |x_j| = \alpha \|x\|_1$$

Ainsi, $\|\mathrm{A}x\|_1 \leqslant \alpha \|x\|_1$ pour tout $x \in \mathbb{C}^n$, d'où $\|\mathrm{A}\| \leqslant \alpha$ en vertu de l'équivalence établie à la question 1.a.

Soit maintenant un entier $j \in [\![ 1 \, ; n ]\!]$ tel que $\alpha = \displaystyle\sum_{i=1}^{n} |a_{ij}|$. Considérons le vecteur $x \in \mathbb{C}^n$ défini par $x_j = 1$ et $x_i = 0$ si $i \neq j$ (c'est en fait le $j^{\mathrm{e}}$ vecteur de la base canonique). Alors $\|x\|_1 = 1$ et

$$\forall \, i \in [\![ 1 \, ; n ]\!] \qquad (\mathrm{A}x)_i = \sum_{k=1}^{n} a_{ik} x_k = a_{ij}$$

d'où

$$\|\mathrm{A}x\|_1 = \sum_{i=1}^{n} |a_{ij}| = \alpha$$

ce qui prouve que $\|\mathrm{A}\| \geqslant \alpha$. Il en découle que $\|\mathrm{A}\| = \alpha$ soit

$$\boxed{\forall \, \mathrm{A} \in \mathscr{M}_n(\mathrm{C}) \qquad \|\mathrm{A}\| = \underset{1 \leqslant j \leqslant n}{\mathrm{Max}} \sum_{i=1}^{n} |a_{ij}|}$$

# X/ENS Informatique B (MP/PC) 2017 — Corrigé

Ce corrigé est proposé par Josselin Giet (ENS Ulm) ; il a été relu par Virgile Andreani (ENS Ulm) et Vincent Puyhaubert (professeur en CPGE).

---

Ce sujet présente une solution optimisée à la recherche d'intersection d'ensembles de points à coordonnées entières.

- La première partie permet de définir le problème et d'en donner une solution naïve et guidée, dans le langage Python. Comme toute résolution naïve, elle permet de constater la nécessité de trouver une optimisation.
- La deuxième partie relie cette problématique à l'utilisation d'une base de données. Elle permet d'écrire des exemples de requêtes SQL afin de résoudre le problème.
- La troisième partie introduit la notion de codage de Lebesgue d'un point à coordonnées entières. Il s'agit d'une autre manière de représenter un point en entrelaçant l'écriture binaire de ses coordonnées.
- La quatrième partie introduit la notion d'ordre lexicographique appliquée aux codages de Lebesgue de points ainsi qu'une interprétation visuelle du codage de Lebesgue sous forme de « chemin » pour trouver la position du point.
- La dernière partie définit le codage de Lebesgue compacté d'un ensemble de points et utilise ce procédé pour résoudre le problème du calcul de l'intersection d'ensembles de points de manière optimisée.

Cet énoncé permet d'aborder un problème d'algorithmique original et intéressant. Les remarques tout au long de l'énoncé ainsi que les questions « exemple » permettent de bien comprendre les nouvelles notions introduites. La remarque en début d'énoncé, à propos des opérations sur les listes, interdit d'utiliser un grand nombre d'opérations qui sont rapides à écrire en Python, comme la compréhension de liste.

# Indications

## Partie I

2 Utiliser la fonction de la question précédente.

## Partie II

4 Faire une jointure entre les tables `POINTS` et `MEMBRE`.

5 Utiliser l'opérateur `INTERSECT`.

6 Faire une jointure dont un des arguments est une autre requête complète adéquate.

## Partie III

8 On peut remarquer que chaque élément de la liste peut être calculé indépendamment des autres.

## Partie IV

9 Lire la remarque sur l'ordre lexicographique, ce qui permet de mieux comprendre la définition.

11 Il y a une coquille dans cette question : Il ne faut pas lire le mot « compacte ».

## Partie V

13 Faire attention au cas `k == 0`.

14 Utiliser la fonction de la question précédente.

15 Expliciter ce que signifie une inclusion stricte de deux ensembles au niveau de leur codage de Lebesgue.

16 Utiliser l'ordre lexicographique pour parcourir une seule fois les deux AQL en parallèle.

# I. Une solution naïve en Python

**1** L'algorithme consiste à regarder tous les éléments de `q`, un par un, et à tester si l'un d'entre eux est égal à `p`.

```
def membre(p,q):
    for i in range(len(q)):
        q_i = q[i]
        if p[0] == q_i[0] and p[1] == q_i[1]:
            return True
    return False
```

> Cet algorithme s'arrête quand on a trouvé un élément dans la liste égal à `p` (ce qui correspond au `return True` ligne 5) ou quand on a fini d'énumérer les éléments de `q` (ce qui correspond au `return False` ligne 6).
>
> Le test de la ligne 4 peut s'écrire de nombreuses manières. Par exemple, en remarquant que `q` est en réalité une matrice de taille `len(p)`×2, on peut enlever la ligne 4 et remplacer `q_i[0]` (*resp.* `q_i[1]`) par `q[i][0]` (*resp.* `q[i][1]`), ce qui sera toujours le cas dans la suite.
>
> Enfin, il est possible d'écrire une version purement itérative de cet algorithme (*i.e.* en ne faisant pas de `return` dans une boucle `for` ou `while`). Il faut savoir le faire, car de nombreux langages ne permettent pas de renvoyer le résultat en plein milieu d'une boucle `for`. Pour cela, on utilise une boucle `while` dont le test dépend du nombre d'éléments de `q` testés, ainsi que d'une variable qui dit si un élément de `q` égal à `p` a été trouvé.
>
> ```
> def membre(p,q):
>     res , i = False , 0
>     while (i < len(q)) and not res:
>         if p[0] == q[i][0] and p[1] = q[i][1]:
>             res = True
>         else
>             i += 1
>     return res
> ```

**2** En suivant l'algorithme donné dans l'énoncé, on obtient la fonction ci-dessous.

```
def intersection(p,q):
    res = []
    for i in range(len(p)):
        if membre(p[i],q):
            res.append(p[i])
    return res
```

**3** Notons $|p|$ (*resp.* $|q|$) la taille de `p` (*resp.* de `q`). À chaque passage dans la boucle `for` de la fonction `membre`, on fait deux comparaisons. Ainsi, la complexité d'un appel de la fonction `membre` est $2 \cdot |q|$. Or, on fait $|p|$ appels à la fonction `membre` dans la fonction `intersection`. Par conséquent,

La complexité de l'algorithme est $\mathcal{O}(|p| \cdot |q|)$

## II. Une solution naïve en SQL

**4** Cette requête peut s'écrire en utilisant une jointure.

```
SELECT idensemble
FROM POINTS JOIN MEMBRE ON id = idpoint
WHERE x = a AND y = b
```

> Cette jointure peut aussi s'écrire comme un simple produit cartésien en déplaçant la condition de jointure à la condition de sélection.
>
> ```
> SELECT idensemble
> FROM POINTS JOIN MEMBRE
> WHERE id = idpoint AND x = a AND y = b
> ```
>
> Il faut toutefois noter que cette manière de procéder, bien que parfaitement correcte, est fortement déconseillée, car elle construit toute la table alors que la jointure sélectionne les entrées puis fait les tests de la ligne `WHERE`. Dès lors, dans le premier cas, la requête est généralement plus rapide. Dans la suite, nous utiliserons systématiquement une jointure sur une clé primaire avant de faire des tests.

**5** Cette requête peut se faire au moyen d'une intersection.

```
(   SELECT idpoint
    FROM MEMBRE
    WHERE idensemble = i)
INTERSECT
(   SELECT idpoint
    FROM MEMBRE
    WHERE idensemble = j)
```

> Cette requête n'est pas la seule réponse possible. On peut aussi bien faire une jointure sur la table `MEMBRE`, comme dans l'exemple suivant :
>
> ```
> SELECT t1.idpoint
> FROM MEMBRE AS t1 JOIN MEMBRE AS t2
>     ON t1.idpoint = t2.idpoint
> WHERE t1.idensemble = i AND t2.idensemble = j
> ```
>
> Cette requête crée, en effet, une table relationnelle contenant
>
> `(t1.idpoint, t1.ensemble, t2.idpoint, t2.ensemble)`
>
> mais comme `t1.idpoint = t2.idpoint`, chaque entrée de cette table correspond à un point et à un couple d'ensembles qui contiennent ce point (ce couple peut être deux fois le même ensemble et on retrouve deux fois la même entrée en échangeant `t1.idensemble` et `t2.idensemble`).
>
> On peut écrire la requête plus lisiblement que dans l'exemple précédent en utilisant l'opérateur `IN` (qui n'est pas explicitement au programme).
>
> ```
> SELECT x,y
> FROM POINTS JOIN MEMBRE ON id = idpoint
> WHERE idensemble = i AND idpoint IN
>     (SELECT idpoint
>      FROM MEMBRE
>      WHERE idensemble = j)
> ```